सूचना प्रौद्योगिकी

Information Technology

सहायक पुस्तिका कक्षा 12 2025-26

राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद

वरुण मार्ग, डिफेंस कॉलोनी, नई दिल्ली - 110025

सूचना प्रौद्योगिकी
(Information Technology)

Class XII

SCERT DELHI

स्वाध्यायान्मा प्रमदः

State Council of Educational Research & Training, Delhi

Varun Marg, Defence Colony, New Delhi - 110024

कक्षा 12 के लिए, विषय "सूचना प्रौद्योगिकी"
ISBN: 978-93-6291-637-2
© राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद, दिल्ली
अप्रैल, 2025

मुख्य सलाहकार
डॉ. रीता शर्मा, निदेशक, राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , दिल्ली
डॉ. नाहर सिंह, संयुक्त निदेशक (शैक्षणिक), राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , दिल्ली

सलाहकार
श्रीमती बिमला कुमारी, डी.डी.ई, वोकेशनल शिक्षा, दिल्ली
श्री राकेश बल, ओ.एस.डी,वोकेशनल ब्रांच, दिल्ली
श्री संजीव कुमार गौड़, ओ.एस.डी,वोकेशनल ब्रांच, दिल्ली

नोडल अधिकारी
श्रीमती रमन अरोड़ा,सहायक प्रोफेसर, राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , दिल्ली
डॉ. अप्सरा अंसारी, सहायक प्रोफेसर, राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , दिल्ली

विषय समन्वयक
डॉ. अनामिका सिंह, प्रिंसिपल, मंडलीय शिक्षा एवं प्रशिक्षण संस्थान, घुमन्हेरा, दिल्ली

लेखक एवं समीक्षक समूह
डॉ. अनामिका सिंह, प्रिंसिपल, मंडलीय शिक्षा एवं प्रशिक्षण संस्थान, घुमन्हेरा, दिल्ली
डॉ. सीमा श्रीवास्तव, प्रोफेसर, के. आई. आई. टी. गुरूग्राम
डॉ. निर्मल सिंह, सेवानिवृत्त डी. एस. ई. यू.
डॉ. अलका योगी, शारदा यूनिवर्सिटी
डॉ. दिव्या मान, सहायक प्रोफेसर, मंडलीय शिक्षा एवं प्रशिक्षण संस्थान, केशवपुरम, दिल्ली
डॉ. राहुल मिश्रा, सहायक प्रोफेसर, मंडलीय शिक्षा एवं प्रशिक्षण संस्थान, राजेंद्र नगर, दिल्ली
श्री विवेक कुमार श्रीवास्तव, व्यावसायिक प्रशिक्षक, रा. प्र. वि. वि., आई. एन. ए. कॉलोनी, दिल्ली
श्रीमती रितु कुमारी, व्यावसायिक प्रशिक्षक, गवर्नमेंट को-एड. एस. एस. एस. सेक्टर 2 द्वारका, दिल्ली
श्री मोहित शर्मा, व्यावसायिक प्रशिक्षक, जी. जी. एस. एस. एस. भाटी माइंस, दिल्ली
श्री महेश कुमार, व्यावसायिक प्रशिक्षक, गवर्नमेंट बॉयज़ सीनियर सेकेंडरी स्कूल, जंगपुरा, दिल्ली

प्रकाशन अधिकारी
डॉ. मुकेश यादव,राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , नई दिल्ली

प्रकाशन दल
श्री दिनेश कुमार शर्मा, (ए.एस.ओ.), राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , नई दिल्ली
सुश्री फ़ौजिया, (बी.आर.पी.), राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद् , नई दिल्ली

प्रकाशित : राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद, दिल्ली
मुद्रित : राज प्रिंटर्स, ए-9, सेक्टर बी-2, ट्रोनिका सिटी, लोनी, गाज़ियाबाद (यू.पी.)

SCERT DELHI

निदेशक का संदेश

Dr. Rita Sharma
Director SCERT

SCERT
स्वाध्यायान्मा प्रमदः

STATE COUNCIL OF EDUCATIONAL RESEARCH and TRAINING
(An autonomous Organisation of GNCT of Delhi)
Varun Marg, Defence Colony, New Delhi-110024
Tel.: +91-11-24331356
E-mail : dir12scert@gmail.com

Date : 29/5/2025

D.O. No. : F.1(1)/DPB/AH Conv/37

संदेश

"राष्ट्रीय शिक्षा नीति 2020" के अंतर्गत स्कूली शिक्षा में व्यावसायिक शिक्षा के एकीकरण पर विशेष बल दिया गया है, जिससे विद्यार्थियों को प्रारंभिक स्तर से ही जीवनोपयोगी और रोजगारोन्मुख कौशलों से जोड़ा जा सके। यह नीति ज्ञान और कौशल के समन्वय से आत्मनिर्भर भारत के निर्माण की दिशा में एक महत्वपूर्ण पहल है।

इसी उद्देश्य की पूर्ति हेतु राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद (SCERT), दिल्ली द्वारा वरिष्ठ माध्यमिक स्तर के विद्यार्थियों के लिए विभिन्न व्यावसायिक विषयों जैसे कि **ऑटोमोटिव, सौंदर्य एवं कल्याण, इलेक्ट्रॉनिक्स और हार्डवेयर, रोज़गार कौशल, वित्तीय बाज़ार प्रबंधन, सूचना प्रौद्योगिकी, खाद्य उत्पाद, स्वास्थ्य देखभाल, विपणन, शारीरिक गतिविधि प्रशिक्षक, रिटेल/खुदरा तथा पर्यटन** के लिए सहायक सामग्री का निर्माण किया गया है।

इन विषयों की सहायक सामग्री इस प्रकार तैयार की गई है कि वह विद्यार्थियों को विषय की मूल अवधारणाओं को समझने, व्यावहारिक रूप से लागू करने और 21वीं सदी के आवश्यक कौशल विकसित करने में सहायता करे। इसमें शिक्षकों के लिए उपयोगी शिक्षण विधियाँ, गतिविधियाँ, मूल्यांकन सुझाव और केस स्टडी जैसे घटकों को शामिल किया गया है, जो शिक्षण को अधिक प्रभावशाली और रोचक बनाते हैं।

आशा है कि यह सहायक सामग्री शिक्षकों एवं विद्यार्थियों के लिए उपयोगी सिद्ध होगी।

रीता शर्मा

(डॉ. रीता शर्मा)
निदेशक

संयुक्त निदेशक का संदेश

स्वाध्यायान्मा प्रमदः

Dr. Nahar Singh
Joint Director (Academic)

State Council of Educational Research and Training
(An autonomous Organisation of GNCT of Delhi)

Tel. : +91-11-24336818, 24331355, Fax : +91-11-24332426
Tel.: +91-11-24331355, Fax : +91-11-24332426
E-mail : jdscertdelhi@gmail.com

Date : 26/05/2025

D.O. No. : F.11(2)JDB/Acad/Misc/SCERT/2025-26/404

संदेश

व्यावसायिक शिक्षा वर्तमान युग की एक अनिवार्य आवश्यकता बन चुकी है, जो विद्यार्थियों को न केवल शैक्षणिक ज्ञान देती है, बल्कि उन्हें व्यावहारिक कौशल और आत्मनिर्भरता की दिशा में भी अग्रसर करती है। **राष्ट्रीय शिक्षा नीति 2020** इसी सोच को साकार करने का एक सशक्त माध्यम है, जो शिक्षा को समग्रता प्रदान करती है।

राज्य शैक्षिक अनुसंधान एवं प्रशिक्षण परिषद, दिल्ली द्वारा व्यावसायिक विषयों के लिए तैयार की गई यह **सहायक सामग्री** शिक्षकों और विद्यार्थियों दोनों के लिए उपयोगी संसाधन है। इसमें पाठ्यवस्तु को सरल और रोचक तरीके से प्रस्तुत किया गया है, जिससे विद्यार्थियों को विषयों की गहरी समझ विकसित करने में सहायता मिलेगी। यह सामग्री न केवल सीखने की प्रक्रिया को प्रभावी बनाएगी, बल्कि छात्रों में आत्मविश्वास और कौशल विकसित करने की दृष्टि से भी महत्वपूर्ण सिद्ध होगी।

यह सामग्री विद्यार्थियों और शिक्षकों के लिए एक दिशा-निर्देशक की तरह कार्य करेगी, जिससे वे विषय को वास्तविक जीवन से जुड़ा हुआ एवं अनुभव आधारित बना सके।

(डॉ. नाहर सिंह)
संयुक्त निदेशक

प्रस्तावना

21वीं सदी को प्रौद्योगिकी में तेजी से हो रही प्रगति द्वारा परिभाषित किया गया है और सूचना प्रौद्योगिकी (आईटी) इस परिवर्तन के केंद्र में है। जैसे-जैसे व्यवसाय, संस्थान और व्यक्ति डिजिटल उपकरणों और प्लेटफ़ॉर्म पर अधिक से अधिक निर्भर होते जा रहे हैं, कुशल आईटी पेशेवरों की मांग बढ़ती जा रही है। छात्रों को इस गतिशील और विकसित होते क्षेत्र के लिए तैयार करने के लिए, उन्हें सैद्धांतिक ज्ञान और व्यावहारिक कौशल दोनों में समृद्ध करना महत्वपूर्ण है। कक्षा XII के लिए यह व्यावसायिक पाठ्यपुस्तक कक्षा XI में शुरू की गई मूलभूत अवधारणाओं की निरंतरता है और इसे उन्नत आईटी अनुप्रयोगों और कार्यस्थल दक्षताओं के बारे में छात्रों की समझ बढ़ाने के लिए प्रकल्पित किया गया है। सामग्री राष्ट्रीय कौशल योग्यता रूपरेखा (NSQF) और उद्योग मानकों के साथ संरेखित होती है ताकि यह सुनिश्चित किया जा सके कि छात्र पूरा होने पर करियर के लिए तैयार हों।

कक्षा 12 के विद्यार्थियों के लिए प्रस्तुत यह पुस्तक सूचना प्रौद्योगिकी के गहन अध्ययन हेतु एक सशक्त माध्यम है। इसका उद्देश्य विद्यार्थियों को तकनीकी दक्षता प्रदान करना है, जिससे वे न केवल उच्च शिक्षा के लिए तैयार हों, बल्कि भविष्य में विभिन्न रोजगारोन्मुखी अवसरों का भी लाभ उठा सकें।

इस पुस्तक में विषय-विशेष कौशल (Subject Specific Skills) को शामिल किया गया है। इस भाग में उन्नत कंप्यूटर अनुप्रयोगों जैसे स्प्रेडशीट में दत्तांश-विश्लेषण (Data analysis), डेटाबेस प्रबंधन प्रणाली, वेब पेज डिज़ाइनिंग, और साइबर सुरक्षा जैसे विषयों को व्यावहारिक उदाहरणों सहित प्रस्तुत किया गया है। पुस्तक में छात्र-केंद्रित दृष्टिकोण अपनाया गया है, जिसमें सैद्धांतिक अवधारणाओं को व्यावहारिक प्रदत्त- कार्यों (Assignment) के साथ जोड़कर आलोचनात्मक सोच, रचनात्मकता और समस्या-समाधान क्षमताओं\ को बढ़ावा दिया गया है। अधिगम को सुदृढ़ करने और आत्म-मूल्यांकन में सक्षम करने के लिए मूल्यांकन, गतिविधियाँ और केस स्टडीज़ को एकीकृत किया गया है।

हमें पूर्ण विश्वास है कि यह पाठ्यपुस्तक छात्रों को उच्च शिक्षा प्राप्त करने या विभिन्न आईटी-संबंधित भूमिकाओं हेतु आवश्यक कौशल, आत्मविश्वास और ज्ञान में सशक्त बनाएगी। यह डिजिटल रूप से कुशल, सक्षम और जिम्मेदार पीढ़ी के निर्माण की दिशा में एक अगला कदम है। पुस्तक की रचना इस प्रकार की गई है कि विद्यार्थी न केवल ज्ञान प्राप्त करें, बल्कि उसे व्यवहार में भी ला सकें। इसमें दिए गए अभ्यास, प्रोजेक्ट कार्य और मूल्यांकन कार्य उन्हें आत्ममूल्यांकन का अवसर प्रदान करते हैं।

डॉ. अनामिका सिंह

SCERT DELHI

विषय-सूची

विवरण		पृष्ठ संख्या
इकाई 1: डेटाबेस अवधारणाएँ – आरडीबीएमएस टूल (Database Concepts – RDBMS Tool)		01
1.	डेटाबेस की बुनियादी जानकारी	01
2.	रिलेशनल डेटाबेस मैनेजमेंट सिस्टम (RDBMS) का परिचय	07
3.	स्ट्रक्चर्ड क्वेरी लैंग्वेज (SQL)	13
4.	SQL क्वेरी	47
इकाई 2: वेब आधारित अनुप्रयोगों का संचालन (Operating Web-Based Applications)		56
5.	वेब आधारित सेवाओं का परिचय	56
6.	ऑनलाइन आरक्षण प्रणाली	61
7.	ऑनलाइन शॉपिंग और बिल भुगतान	74
8.	ई-गवर्नेंस और डिजिटल सेवाएँ	82
9.	परियोजन प्रबंधन	86
इकाई 3: जावा – जावा प्रोग्रामिंग की मूल बातें (JAVA - Fundamentals of Java Programming)		104
10.	जावा का परिचय और इतिहास	104
11.	ऑपरेटर्स और कंट्रोल फ्लो	111
12.	ऐरे, क्लास डिज़ाइन और अपवाद प्रबंधन	124
13.	ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग की अवधारणाएँ	128
14.	असर्शन, थ्रेड्स, रैपर क्लासेस और स्ट्रिंग मैनिपुलेशन	161
इकाई 4: वर्क इंटीग्रेटेड लर्निंग आईटी – डीएमए (Work Integrated Learning IT – DMA)		170
15.	डेटा मैनेजमेंट और विश्लेषण (DMA)	170
16.	आईटी कौशल और व्यावसायिक अनुप्रयोग	171
17.	शॉपिंग वेबसाइट-केस स्टडी	179

SCERT DELHI

डेटाबेस अवधारणाएँ – आरडीबीएमएस टूल

इकाई 1

सीखने के प्रतिफल

विद्यार्थी–

* डेटाबेस की मूलभूत संकल्पनाओं को समझ पाएंगे और डेटाबेस के विभिन्न घटकों (जैसे तालिका, रिकॉर्ड, फ़ील्ड) की पहचान कर सकेंगे।
* DBMS की आवश्यकता और उसके लाभों को समझेंगे तथा RDBMS और DBMS में अंतर जान सकेंगे।
* विभिन्न प्रकार के डेटाबेस मॉडल्स जैसे कि हायरार्किकल, नेटवर्क, रिलेशनल और ऑब्जेक्ट-ओरिएंटेड डेटाबेस के बारे में जानकारी प्राप्त करेंगे।
* SQL (Structured Query Language) की बेसिक कमांड्स (जैसे SELECT, INSERT, UPDATE, DELETE) का उपयोग करके डेटाबेस से जानकारी प्राप्त करने और संशोधित करने की क्षमता विकसित करेंगे।
* डेटाबेस की सुरक्षा और डेटा अखंडता (Integrity) के महत्व को समझेंगे और सुरक्षा उपायों पर विचार करेंगे।
* डेटाबेस का उपयोग करके विभिन्न प्रकार की वास्तविक जीवन की समस्याओं को हल करने की क्षमता विकसित करेंगे, जैसे कि जानकारी का संरचना, संशोधन और पुनर्प्राप्ति।

1.1 परिचय (Introduction)

संगठनों की सफलता (Organizational Success) का प्रमुख कारण प्रभावी निर्णय-निर्माण (effective decision-making) है, जिसके लिए समय पर, प्रासंगिक और सटीक जानकारी की आवश्यकता होती है। इसलिए, वर्तमान प्रतिस्पर्धात्मक वातावरण में जानकारी महत्वपूर्ण भूमिका निभाती है। डेटाबेस प्रबंधन सॉफ्टवेयर (DBMS) डेटा को प्रबंधित करने और उपयोगी जानकारी निकालने का कार्य सरल बना देता है। इस अध्याय में, हम डेटाबेस की मूल अवधारणाओं के बारे में जानेंगे और DBMS का उपयोग कुछ अनुप्रयोगों में कैसे किया जाता है, इसे भी सीखेंगे।

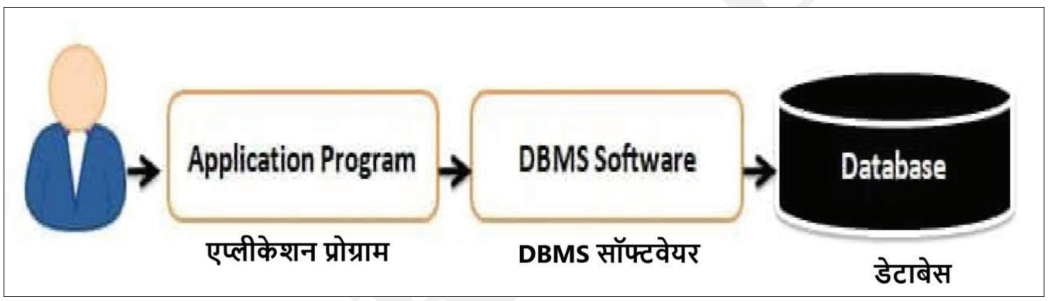
डेटा असंसाधित तथ्यों का एक संग्रह है, जिसे उपयोगी जानकारी में परिवर्तित करने के लिए अभी संसाधित नहीं किया गया है। डेटा को संसाधित कर जानकारी प्राप्त की जाती है, जैसा कि चित्र में दिखाया गया है।

उदाहरण के लिए, यदि एक कक्षा के सभी छात्रों के परीक्षा अंकों का डेटा दिया गया है, तो कक्षा का औसत, अधिकतम और न्यूनतम अंक कक्षा के प्रदर्शन के संकेतक के रूप में उपयोग किए जा सकते हैं।

डेटाबेस का उपयोग हमारे दैनिक जीवन में व्यापक रूप से किया जा रहा है, चाहे वह व्यवसाय हो, इंजीनियरिंग हो, चिकित्सा हो, शिक्षा हो या पुस्तकालय हो।

1.1.1 डेटाबेस (Database) क्या है?

डेटाबेस डेटा का एक संगठित संग्रह है जिसे व्यवस्थित किया गया है और आमतौर पर कंप्यूटर सिस्टम में इलेक्ट्रॉनिक रूप से रखा जाता है। एक डेटाबेस प्रबंधन प्रणाली अक्सर एक डेटाबेस (DBMS) की देखरेख करती है।



चित्र 1. 1 डेटाबेस प्रबंधन प्रणाली

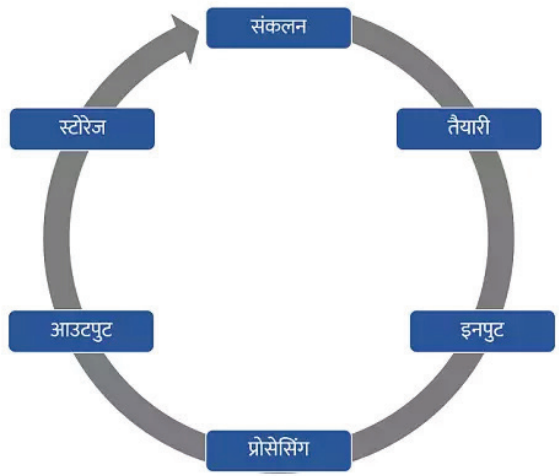
1.1.2 डेटाबेस के गुण:

* डेटाबेस वास्तविक दुनिया के किसी पहलू का प्रतिनिधित्व करता है जिसे मिनीवर्ल्ड भी कहा जाता है। जब भी इस मिनीवर्ल्ड में कोई परिवर्तन होता है तो वह डेटाबेस में भी दिखाई देता है।
* इसे विशिष्ट उद्देश्य के लिए डिजाइन, निर्मित और डेटा से परिपूर्ण किया गया है।
* यह किसी भी आकार और जटिलता का हो सकता है।
* इसका रखरखाव मैन्युअल रूप से किया जा सकता है या इसे कम्प्यूटरीकृत किया जा सकता है।

1.1.3 डेटाबेस की आवश्यकता

डेटाबेस प्रबंधन प्रणाली उपयोगकर्ताओं को पूरे संगठन में सुरक्षित, कुशलतापूर्वक और तेज़ी से डेटा साझा करने में सक्षम बनाती है। डेटा प्रबंधन प्रणाली डेटाबेस अनुरोधों का तुरंत जवाब देकर अधिक

सटीक डेटा तक त्वरित पहुँच प्रदान करती है।



चित्र 1. 2 डेटाबेस की आवश्यकता

1.1.4 डेटाबेस के मुख्य तत्वः

* **डेटाबेस (Database):** डेटाबेस एक संरचित डेटा संग्रह है, जिसमें जानकारी को एकत्रित, संग्रहीत और प्रबंधित किया जाता है। यह कई प्रकार के डेटा को संभाल सकता है, जैसे टेक्स्ट, संख्याएँ, और चित्र।
* **तालिका (Table):** डेटाबेस की मुख्य संरचना होती है, जिसमें डेटा को पंक्तियों और कॉलमों के रूप में व्यवस्थित किया जाता है। प्रत्येक तालिका एक विशिष्ट विषय पर केंद्रित होती है। प्रत्येक तालिका में पंक्तियाँ (rows) और कॉलम (columns) होते हैं। हर पंक्ति एक रिकॉर्ड का प्रतिनिधित्व करती है और प्रत्येक कॉलम उस के एक विशेष गुण को दर्शाता है।
* **पंक्ति (Row):** तालिका में प्रत्येक पंक्ति एक अलग रिकॉर्ड का प्रतिनिधित्व करती है। उदाहरण के लिए, यदि तालिका "छात्र" है, तो प्रत्येक पंक्ति एक अलग छात्र की जानकारी होगी।
* **कॉलम (Column):** कॉलम तालिका में डेटा के एक विशेष प्रकार को दर्शाता है। प्रत्येक कॉलम के पास एक नाम और डेटा प्रकार होता है। उदाहरण के लिए, 'नाम', 'उम्र', 'कक्षा'।
* **डेटा मॉडल (Data Model):** डेटाबेस को संगठित करने के लिए विभिन्न डेटा मॉडल होते हैं, जैसे कि रिलेशनल, नो-एसक्यूएल, ऑब्जेक्ट-ओरिएंटेड आदि।

SCERT DELHI

* **क्वेरीज़ (Queries):** डेटा को खोजने और प्रबंधित करने के लिए SQL (Structured Query Language) का उपयोग किया जाता है। यह उपयोगकर्ताओं को डेटा को जोड़ने, अपडेट करने और हटाने की अनुमति देता है।

* **डेटा इंटीग्रिटी(Data Integrity):** डेटाबेस में डेटा की सटीकता और विश्वसनीयता सुनिश्चित करने के लिए विभिन्न नियमों का पालन किया जाता है।

* **डेटाबेस प्रबंधन प्रणाली (DBMS):** यह एक सॉफ़्टवेयर होता है जो डेटाबेस के निर्माण, प्रबंधन और उपयोग को आसान बनाता है। उदाहरण के लिए, MySQL, Oracle, और Microsoft SQL Server।

1.1.5 डेटाबेस पर किए जाने वाले विभिन्न कार्य निम्नलिखित हैं:

* **डेटाबेस को परिभाषित करना (Defining the Database):** इसमें डेटाबेस में संग्रहीत किए जाने वाले डेटा के प्रकार को निर्दिष्ट करना शामिल है और साथ ही उस डेटा पर लागू किसी भी प्रतिबंध को परिभाषित करना होता है।

* **डेटाबेस को भरना (Populating the Database):** इसमें डेटा को किसी स्टोरेज माध्यम पर संग्रहीत करना शामिल होता है, जिसे DBMS (डेटाबेस प्रबंधन प्रणाली) द्वारा नियंत्रित किया जाता है।

* **डेटाबेस में बदलाव करना (Manipulating the Database):** इसमें डेटाबेस को संशोधित करना, डेटा को पुनः प्राप्त करना या डेटाबेस में क्वेरी करना, रिपोर्ट बनाना आदि शामिल होता है।

* **डेटाबेस को साझा करना (Sharing the Database):** यह विभिन्न उपयोगकर्ताओं को एक साथ डेटाबेस तक पहुंचने की अनुमति प्रदान करता है।

* **डेटाबेस की सुरक्षा (Protecting the Database):** यह सॉफ़्टवेयर या हार्डवेयर विफलताओं और अनधिकृत पहुंच से डेटाबेस की सुरक्षा को सक्षम बनाता है।

* **डेटाबेस का रखरखाव (Maintaining the Database):** यह बदलती आवश्यकताओं के अनुसार डेटाबेस को अनुकूलित करना आसान बनाता है।

कुछ DBMS के उदाहरण हैं – MySQL, Oracle, DB2, IMS, IDS आदि।

SCERT DELHI

1.2 डेटाबेस मैनेजमेंट सिस्टम की विशेषताएँ (Characteristics of Database Management Systems)

* **डेटाबेस सिस्टम की आत्म-वर्णनात्मक प्रकृति (Self-describing Nature of a Database System):** DBMS में न केवल डेटाबेस शामिल होता है बल्कि उस डेटा का विवरण भी होता है जो इसमें संग्रहीत होता है। इस डेटा के विवरण को मेटा-डेटा (Meta-data) कहा जाता है। मेटा-डेटा को डेटाबेस कैटलॉग या डेटा डिक्शनरी में संग्रहीत किया जाता है। इसमें डेटा की संरचना और उस डेटा पर लागू प्रतिबंधों की जानकारी होती है।

* **प्रोग्राम और डेटा के बीच इन्सुलेशन (Insulation Between Programs and Data):** चूंकि डेटा की परिभाषा DBMS में अलग से संग्रहीत होती है, इसलिए डेटा की संरचना में किसी भी परिवर्तन को कैटलॉग में किया जा सकता है और इस तरह उस डेटा को एक्सेस करने वाले प्रोग्राम को संशोधित करने की आवश्यकता नहीं होती है। इस गुण को प्रोग्राम-डेटा स्वतंत्रता (Program-Data Independence) कहा जाता है।

* **डेटा का साझा करना (Sharing of Data):** एक मल्टीयूज़र वातावरण (multiuser environment) में, कई उपयोगकर्ता एक साथ डेटाबेस को एक्सेस कर सकते हैं। इसलिए, DBMS में समवर्तीता नियंत्रण सॉफ़्टवेयर (concurrency control software) शामिल होना चाहिए ताकि डेटाबेस में डेटा का एक साथ उपयोग किया जा सके और किसी भी असंगति की समस्या से बचा जा सके।

यह DBMS की विशेषताएँ इसे प्रभावी और विश्वसनीय बनाती हैं।

1.2.1 DBMS के उपयोगकर्ता और उनके प्रकार (Types of Users of DBMS)

DBMS का उपयोग विभिन्न प्रकार के उपयोगकर्ताओं द्वारा किया जाता है, जिनकी आवश्यकताएं और DBMS के साथ उनकी बातचीत भिन्न होती है। मुख्य रूप से DBMS के चार प्रकार के उपयोगकर्ता होते हैं:

* **एंड यूजर्स (End Users):** ये उपयोगकर्ता अपने कार्य के अनुसार डेटाबेस से क्वेरी, संशोधन और रिपोर्ट बनाने के लिए DBMS का उपयोग करते हैं। इन्हें डेटाबेस के कामकाज और डिज़ाइन की जानकारी की आवश्यकता नहीं होती; वे केवल DBMS का उपयोग करके अपना कार्य पूरा करते हैं

* **डेटाबेस प्रशासक (Database Administrator - DBA):** जैसा कि नाम से पता चलता है, DBA डेटाबेस और DBMS का प्रबंधन करता है। DBA का कार्य डेटा की सुरक्षा,

उपयोग की निगरानी, तकनीकी सहायता प्रदान करना और सॉफ़्टवेयर तथा हार्डवेयर संसाधनों की प्राप्ति सुनिश्चित करना है।

* **एप्लिकेशन प्रोग्रामर्स (Application Programmers):** ये प्रोग्रामर डेटाबेस के साथ इंटरैक्ट करने के लिए एप्लिकेशन प्रोग्राम लिखते हैं। ये प्रोग्राम उच्च स्तरीय भाषाओं और SQL का उपयोग करके डेटाबेस के साथ काम करते हैं।

* **सिस्टम एनालिस्ट (System Analyst):** सिस्टम एनालिस्ट एंड यूजर्स की आवश्यकताओं का निर्धारण करते हैं और फिर उन आवश्यकताओं को पूरा करने के लिए विशिष्टताएं तैयार करते हैं। डेटाबेस डिज़ाइन और तकनीकी, आर्थिक तथा व्यवहारिक पहलुओं में सिस्टम एनालिस्ट की महत्वपूर्ण भूमिका होती है।

1.2.2 DBMS का उपयोग करने के लाभ (Advantages of using DBMS Approach)

DBMS का उपयोग करने की आवश्यकता ही इसके लाभों को दर्शाती है। DBMS के कुछ प्रमुख लाभ निम्नलिखित हैं:

* **डेटा की पुनरावृत्ति में कमी (Reduction in Redundancy):** DBMS में डेटा केंद्रीकृत रूप में संग्रहीत होता है जिससे डेटा की पुनरावृत्ति नहीं होती। डेटा एक ही स्थान पर संग्रहीत होता है, जिससे हार्ड डिस्क या अन्य मेमोरी उपकरणों पर डेटा संग्रहीत करने की लागत भी कम हो जाती है।

* **सुधरी हुई सुसंगतता (Improved Consistency):** डेटाबेस में डेटा की असंगतियों की संभावना भी कम हो जाती है क्योंकि सभी उपयोगकर्ताओं द्वारा एक्सेस या अपडेट किया जाने वाला डेटा केवल एक प्रति में होता है।

* **सुधरी हुई उपलब्धता (Improved Availability):** एक ही जानकारी विभिन्न उपयोगकर्ताओं को उपलब्ध कराई जाती है, जिससे डेटाबेस के विभिन्न उपयोगकर्ताओं द्वारा जानकारी को साझा करना आसान हो जाता है।

* **सुधरी हुई सुरक्षा (Improved Security):** जानकारी की उपलब्धता में सुधार होने के बावजूद, गोपनीय जानकारी तक पहुंच को सीमित करना आवश्यक हो सकता है। पासवर्ड और उपयोगकर्ताओं के एक्सेस अधिकारों को नियंत्रित करके DBA डेटाबेस की सुरक्षा सुनिश्चित कर सकता है।

SCERT DELHI

* **उपयोगकर्ता के अनुकूल (User Friendly):** DBMS का उपयोग करके डेटा को एक्सेस, संशोधित और हटाना बहुत आसान हो जाता है। इसका उपयोगकर्ता के अनुकूल इंटरफ़ेस उपयोगकर्ताओं की कंप्यूटर विशेषज्ञों पर निर्भरता को कम कर देता है।

1.2.3 DBMS का उपयोग करने की सीमाएँ (Limitations of using DBMS Approach)

DBMS का उपयोग करने के दो मुख्य नुकसान हैं:

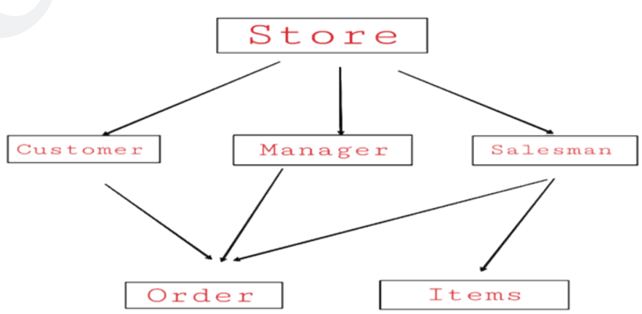
* **उच्च लागत (High Cost):** DBMS प्रणाली को लागू करने की लागत बहुत अधिक होती है। यह एक समय लेने वाली प्रक्रिया भी होती है, जिसमें उपयोगकर्ता की आवश्यकताओं का विश्लेषण करना, डेटाबेस विनिर्देशों को डिज़ाइन करना, एप्लिकेशन प्रोग्राम लिखना और फिर प्रशिक्षण प्रदान करना शामिल होता है।

* **सुरक्षा और पुनर्प्राप्ति की समस्या (Security and Recovery Overheads):** डेटाबेस तक अनधिकृत पहुंच से संगठनों या व्यक्तियों को खतरा हो सकता है, खासकर जब संवेदनशील डेटा संग्रहीत होता है। इसके अलावा, डेटा का नियमित रूप से बैकअप लेना आवश्यक होता है ताकि आग, भूकंप आदि से डेटा की हानि को रोका जा सके।

इसीलिए DBMS का उपयोग आमतौर पर तब नहीं किया जाता जब डेटाबेस छोटा, अच्छी तरह से परिभाषित, कम परिवर्तनशील और कुछ उपयोगकर्ताओं द्वारा ही उपयोग किया जाता हो।

1.3 रिलेशनल डेटाबेस (Relational Database) क्या है?

डेटा को नीचे Shopkart डेटाबेस में अलग-अलग तालिकाओं में व्यवस्थित किया गया है। एक बार तालिकाएँ स्थापित की गई हैं, उन्हें एक साथ जोड़ने के लिए एक संबंध बनाया जा सकता है। ऐसा डेटाबेस जो डेटा को अलग-अलग तालिकाओं में संग्रहीत करके एक सामान्य कॉलम के उपयोग के माध्यम से संबंधित होते हैं, रिलेशनल डेटाबेस कहलाते हैं।



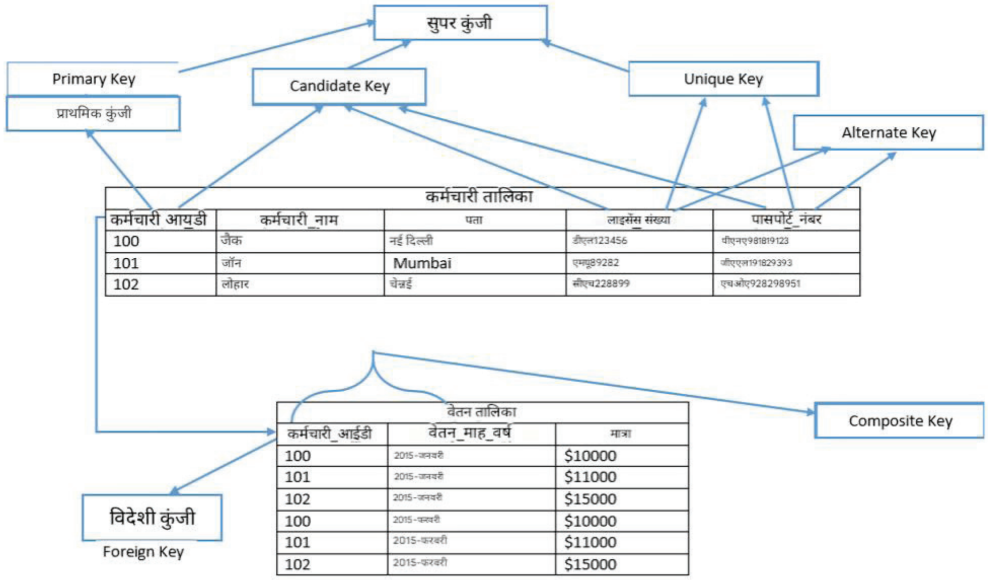
चित्र 1. 3 रिलेशनल डेटाबेस का उदाहरण

1.3.1 आरडीबीएमएस शब्दावली (RDBMS Terminology)

- **प्राइमरी की (Primary Key):** यह एक विशेष कॉलम है जो तालिका में प्रत्येक पंक्ति को अद्वितीय रूप से पहचानता है। प्राइमरी की को NULL मान नहीं हो सकता और इसे किसी भी रिकॉर्ड को पहचानने के लिए प्रयोग किया जाता है।
- **कैंडिडेट की (Candidate Key):** यह एक या एक से अधिक कॉलम (फ़ील्ड्स) का सेट होता है, जो किसी टेबल में सभी रिकॉर्ड्स (rows) को अनन्य रूप से (uniquely) पहचानने के लिए उपयोग किया जा सकता है। किसी भी टेबल में एक से अधिक Candidate Key हो सकती हैं, लेकिन उनमें से केवल एक को Primary Key के रूप में चुना जाता है।
- **फॉरेन की (Foreign Key):** यह एक कॉलम है जो एक तालिका को दूसरी तालिका से जोड़ता है। यह तालिका के बीच संबंध स्थापित करता है, जिससे डेटा को संयोजित करने की सुविधा मिलती है।
- **अलटरनेट की (Alternate Key):** किसी तालिका की कैंडिडेट की (Candidate Key) जिसे प्राथमिक की के रूप में नहीं चुना गया है, उसे अलटरनेट की कहा जाता है।

  उदाहरण: निम्नलिखित तालिका पर विचार करें, रोल नंबर और एडमिशन नो दोनों का उपयोग विशिष्ट रूप से किया जा सकता है इस तालिका में प्रत्येक पंक्ति की पहचान करें, इसलिए दोनों कैंडिडेट की हैं।
- **क्वेरी (Query):** SQL (Structured Query Language) में लिखा गया एक आदेश है, जो डेटा को पुनः प्राप्त करने, जोड़ने, संशोधित करने या हटाने के लिए उपयोग किया जाता है। उदाहरण के लिए, 'SELECT * FROM STUDENT' सभी छात्रों की जानकारी दिखाएगा।
- **डाटा टाइप (Data Type):** यह एक कॉलम में संग्रहीत डेटा के प्रकार को परिभाषित करता है, जैसे INTEGER (पूर्णांक), VARCHAR (पाठ), DATE (तारीख) आदि।

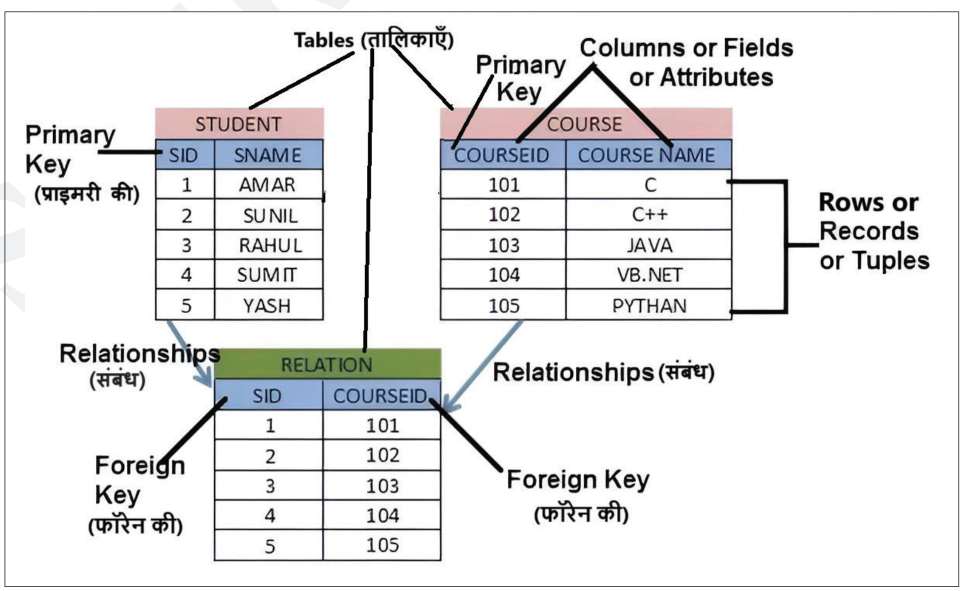
SCERT DELHI



चित्र 1. 4 इन अवधारणाओं के माध्यम से DATABASE को बेहतर ढंग से समझा जा सकता है।

रिलेशनशिप (Relationship):

- यह तालिकाओं के बीच संबंध को दर्शाता है।
- एक-से-एक (One-to-One): एक रिकॉर्ड एक अन्य रिकॉर्ड से जुड़ा होता है।
- एक-से-बहुत (One-to-Many): एक रिकॉर्ड कई अन्य रिकॉर्ड से जुड़ा होता है।
- बहुत-से-बहुत (Many-to-Many): एक रिकॉर्ड कई अन्य रिकॉर्ड से जुड़ा होता है और इसके विपरीत भी।



चित्र 1. 5 इन अवधारणाओं के माध्यम से RDBMS की कार्यप्रणाली और संरचना को बेहतर ढंग से समझा जा सकता है।

1.3.2 RDBMS के मुख्य विशेषताएँ:

* **डेटा की संरचना:** RDBMS में डेटा को तालिकाओं के रूप में व्यवस्थित किया जाता है। प्रत्येक तालिका एक विशेष प्रकार के डेटा को दर्शाती है, जैसे कि ग्राहक, उत्पाद, या ऑर्डर।

* **SQL का उपयोग:** RDBMS में डेटा के साथ संवाद करने के लिए SQL (Structured Query Language) का उपयोग किया जाता है। SQL उपयोगकर्ताओं को डेटा को जोड़ने, खोजने, अपडेट करने और हटाने की अनुमति देता है।

1.4 रिलेशनल मॉडल में प्रतिबंध (Relational Model Constraints)

डेटाबेस में संग्रहीत मानों पर कुछ प्रतिबंध होते हैं, जिन्हें आवश्यकताओं के आधार पर लगाया जाता है। जैसे, अगर हमारे पास एक EMPLOYEE तालिका है, तो उसमें Employee_ID 4 अंकों की संख्या होनी चाहिए, और जन्म तिथि (Date_of_Birth) ऐसी होनी चाहिए कि जन्म वर्ष 1985 से अधिक हो।

रिलेशनल मॉडल में विभिन्न प्रकार के प्रतिबंध निम्नलिखित हैं:

* **डोमेन प्रतिबंध (Domain Constraint):** यह सुनिश्चित करता है कि हर टपल (पंक्ति) में हर एट्रिब्यूट (गुण) का मान उस एट्रिब्यूट के डोमेन (मानों के सेट) से हो। उदाहरण के लिए, Employee_ID 4 अंकों की संख्या होनी चाहिए। इसलिए, मान '12321' या 'A234' इस प्रतिबंध का उल्लंघन करते हैं, क्योंकि '12321' 4 अंकों से अधिक है और 'A234' में अक्षर भी है।

* **की प्रतिबंध (Key Constraint):** इस प्रतिबंध को समझने से पहले हमें सुपर की, की, कैंडिडेट की (candidate key) और प्राइमरी की (primary key) जैसे शब्दों को समझना होगा।

  - **सुपर की (Superkey):** सुपर की गुणों का ऐसा सेट होता है जिसमें किसी भी दो टपल में उन गुणों के मान समान नहीं होते। हर रिलेशन में कम से कम एक सुपर की होती है, जिसमें रिलेशन के सभी गुण शामिल होते हैं। उदाहरण के लिए, EMPLOYEE रिलेशन में कुछ सुपर की निम्नलिखित हैं:

  - {Name, Employee_ID, Gender, Salary, Date_of_birth} - यह सभी गुणों को शामिल करने वाली सुपर की है।

  - {Name, Employee_ID, Date_of_Birth}

- {Employee_ID, Gender, Salary}
- {Name, Employee_ID, Gender}
- {Employee_ID}

लेकिन {Gender, Salary} सुपर की नहीं है, क्योंकि नेहा और हिमानी कर्मचारियों के लिए इन गुणों के मान समान हैं।

* **कैंडीडेट की (Candidate Key):** कैंडीडेट की वह सुपर की होती है, जिसमें यदि किसी भी गुण को हटा दिया जाए, तो वह सुपर की नहीं रहती। उदाहरण के लिए, {Name, Employee_ID, Gender} सुपर की है लेकिन यह की नहीं है, क्योंकि इसमें से Name और Gender को हटाने पर {Employee_ID} बचता है, जो अभी भी एक सुपर की है। अब {Employee_ID} एक की है, क्योंकि यह एक सुपर की है और इसे और घटाया नहीं जा सकता। एक रिलेशन में एक से अधिक की हो सकती है। उदाहरण के लिए, PERSON रिलेशन में {Aadhar_number}, {PAN}, और {Voter_ID_cardno} सभी की हैं, क्योंकि भारत में हर व्यक्ति का आधार, पैन और वोटर आईडी नंबर अलग होता है।

  जैसा कि ऊपर बताया गया है, एक की को कैंडीडेट की कहा जाता है। उदाहरण के लिए, PERSON रिलेशन में तीन कैंडीडेट की हैं: {Aadhar_number}, {PAN}, और {Voter_ID_cardno}।

* **प्राइमरी की (Primary Key):** कैंडीडेट की में से किसी एक को प्राइमरी की के रूप में चुना जाता है। प्राथमिक की का उपयोग रिलेशन में टपल को पहचानने के लिए किया जाता है। यदि रिलेशन में कई कैंडीडेट की हों, तो उस की को प्राइमरी की बनाना उचित होता है, जिसमें कम से कम गुण हों। प्राथमिक की को रिलेशन की स्कीमा में रेखांकित किया जाता है। उदाहरण के लिए, PERSON (Aadhar_number, PAN, Voter_ID_cardno, Name, Date_of_birth, Address) में Aadhar_number प्राइमरी की है।

  इस प्रकार, रिलेशनल मॉडल में विभिन्न प्रकार के प्रतिबंध होते हैं, जो डेटा की अखंडता और संरचना को सुनिश्चित करने में सहायक होते हैं।

* **नल वैल्यू कन्सट्रैन्ट (Null Value Constraint):** कुछ गुणों में NULL मान की अनुमति नहीं होती है। उदाहरण के लिए, यदि हर EMPLOYEE का नाम होना अनिवार्य है, तो Name एट्रिब्यूट में NULL मान नहीं हो सकता।

SCERT DELHI

* **एंटिटी इंटीग्रिटी कन्सट्रैन्ट (Entity Integrity Constraint):** इस प्रतिबंध के अनुसार, किसी तालिका की प्राथमिक की में NULL मान नहीं हो सकता। ऐसा इसलिए क्योंकि प्राथमिक की में कोई भी डुप्लिकेट मान नहीं होना चाहिए, और NULL मान होने से डुप्लिकेट्स की संभावना बढ़ जाती है।

* **रेफरेंशियल इंटीग्रिटी कन्सट्रैन्ट(Referential Integrity Constraint):** यह प्रतिबंध दो तालिकाओं के बीच लागू होता है। इसमें फोरेन की (Foreign Key) का प्रयोग किया जाता है, जो एक तालिका के प्राइमरी की से संदर्भित होती है। उदाहरण के लिए:

  मान लीजिए कि हमारे पास दो तालिकाएँ हैं:

- Department (Dept_Name, Dept_ID, No_of_Teachers)
- Teacher (Teacher_Name, Teacher_ID, Dept_ID, Subject)

यहां, Dept_ID Department तालिका में प्राथमिक की है और Teacher तालिका में यह Dept_ID फोरेन की के रूप में मौजूद है। इसका मतलब है कि Teacher तालिका में हर शिक्षक का Dept_ID Department तालिका में मौजूद होना चाहिए, या फिर NULL हो सकता है। इस प्रकार, रेफरेंशियल इंटीग्रिटी यह सुनिश्चित करती है कि डेटा सुसंगत रहे।

* फोरेन की का नाम प्राइमरी की से अलग हो सकता है। उदाहरण के लिए, Teacher तालिका में Dept_No एक फॉरेन की हो सकती है जो Dept_ID को संदर्भित करती है।

* एक फोरेन की उसी तालिका में भी संदर्भित की जा सकती है। जैसे कि, एक

> **गतिविधि 4.1 (Activity 4.1)**
> एक स्कूल में छात्रों, उनकी कक्षाओं, शिक्षकों, विषयों, और उनके परिणामों को प्रबंधित करने के लिए डेटाबेस डिज़ाइन कीजिये.

Residents तालिका में Neighbor_RID फोरेन की है, जो उसी तालिका में RID (प्राइमरी की) को संदर्भित करती है। इससे हम निवासी के पड़ोसी का विवरण उसी तालिका में रख सकते हैं।

इन प्रतिबंधों का उद्देश्य डेटाबेस में डेटा की संरचना और शुद्धता बनाए रखना है, ताकि डेटा विश्वसनीय और सुसंगत रहे।

1.5 स्ट्रक्चर्ड क्वेरी लैंग्वेज (SQL)

SQL एक भाषा है जिसका उपयोग RDBMS (रेलेशनल डेटाबेस मैनेजमेंट सिस्टम) में डेटा को प्रबंधित करने के लिए किया जाता है। यह दो प्रमुख भागों में बंटी होती है - डेटा डेफिनिशन लैंग्वेज (DDL) और डेटा मैनिपुलेशन लैंग्वेज (DML)।

* **DDL (डेटा डेफिनिशन लैंग्वेज):** इसका उपयोग डेटाबेस में डेटा की संरचना और उस पर लगाई जाने वाली पाबंदियों को परिभाषित करने के लिए किया जाता है।

* **DML (डेटा मैनिपुलेशन लैंग्वेज):** इसका उपयोग डेटाबेस में डेटा को जोड़ने, संशोधित करने और हटाने के लिए किया जाता है।

SQL कमांड्स का उपयोग विभिन्न ऑपरेशनों को पूरा करने के लिए किया जाता है। SQL कमांड्स का अध्ययन करने के लिए, कंप्यूटर पर एक डेटाबेस सिस्टम इंस्टॉल होना चाहिए। इसके लिए कई सॉफ्टवेयर उपलब्ध हैं, और हम यहां MySQL सर्वर का अध्ययन करेंगे।

SQL में रिलेशनल मॉडल के लिए टेबल, रो (पंक्ति) और कॉलम (स्तंभ) जैसे शब्दों का प्रयोग किया जाता है, जो क्रमशः रिलेशन, ट्यूपल और एट्रिब्यूट शब्दों के समान होते हैं।

SQL का अध्ययन करने के लिए हम MySQL Community Server 9.0.1 का उपयोग करेंगे, जो मुफ़्त में डाउनलोड किया जा सकता है। MySQL की नवीनतम संस्करण वेबसाइट पर उपलब्ध है: http://dev.mysql.com/downloads/

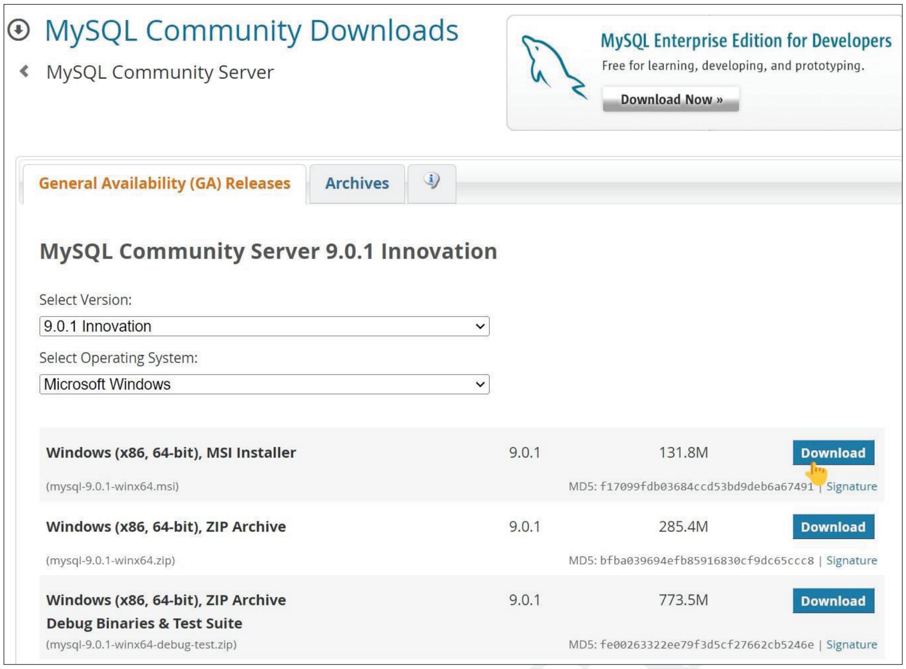
MySQL Community Server 9.0.1 को इंस्टॉल और कॉन्फ़िगर करने के लिए निम्नलिखित चरण हैं:

* URL खोलें: http://dev.mysql.com/downloads/mysql/#downloads

* इस वेबसाइट पर उपलब्ध MySQL Community Server 9.0.1 को डाउनलोड करें। आप अपने ऑपरेटिंग सिस्टम (Windows/Linux) को चुन सकते हैं।

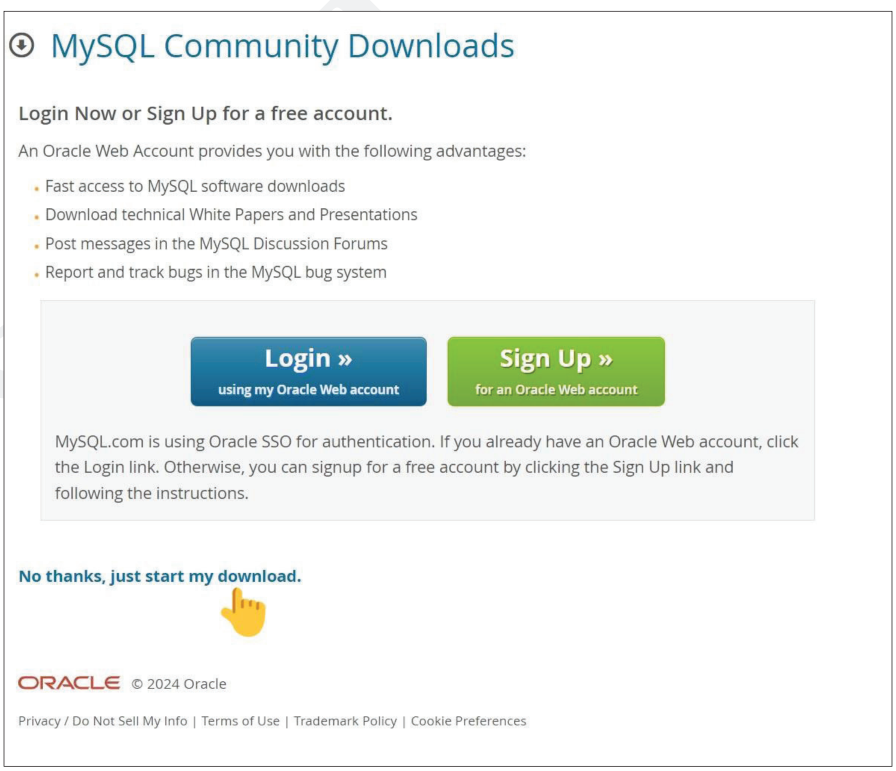
सबसे पहले, अपने ब्राउज़र में जाएं और https://dev.mysql.com/downloads/mysql/ पर जाएं।

आपको डाउनलोड पेज पर 'MySQL Installer for Windows' विकल्प दिखाई देगा।



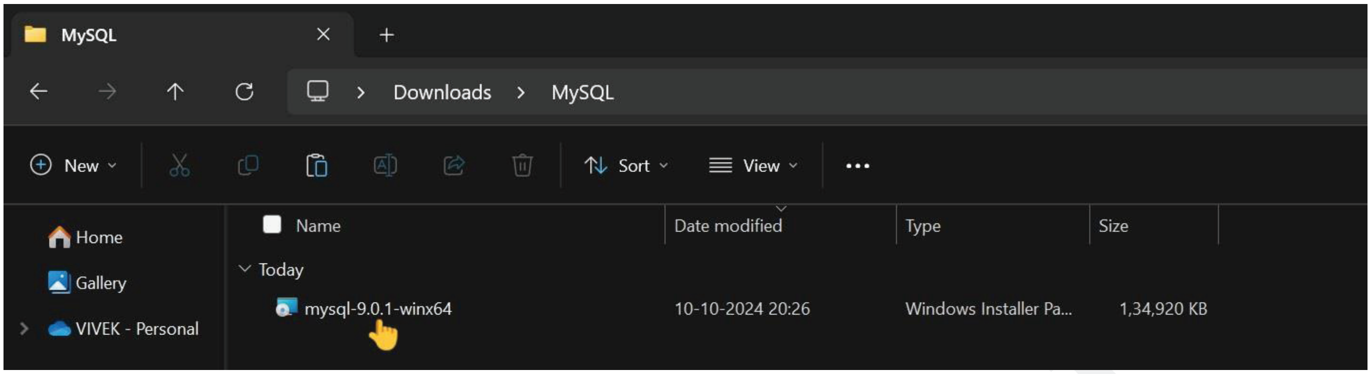
चित्र 1. 6: MySQL कम्युनिटी डाउनलोड का पहला चरण

(ऑपरेटिंग सिस्टम के अनुसार उचित डाउनलोड लिंक चुनें) समर्थित प्लेटफ़ॉर्म (यानी, ऑपरेटिंग सिस्टम जिस पर यह चलेगा) की सूची में से चुनें, जिसमें 32-बिट और 64-बिट विंडोज़, कई अलग-अलग लिनक्स, सोलारिस, मैक ओएस एक्स और कुछ अन्य शामिल हैं।

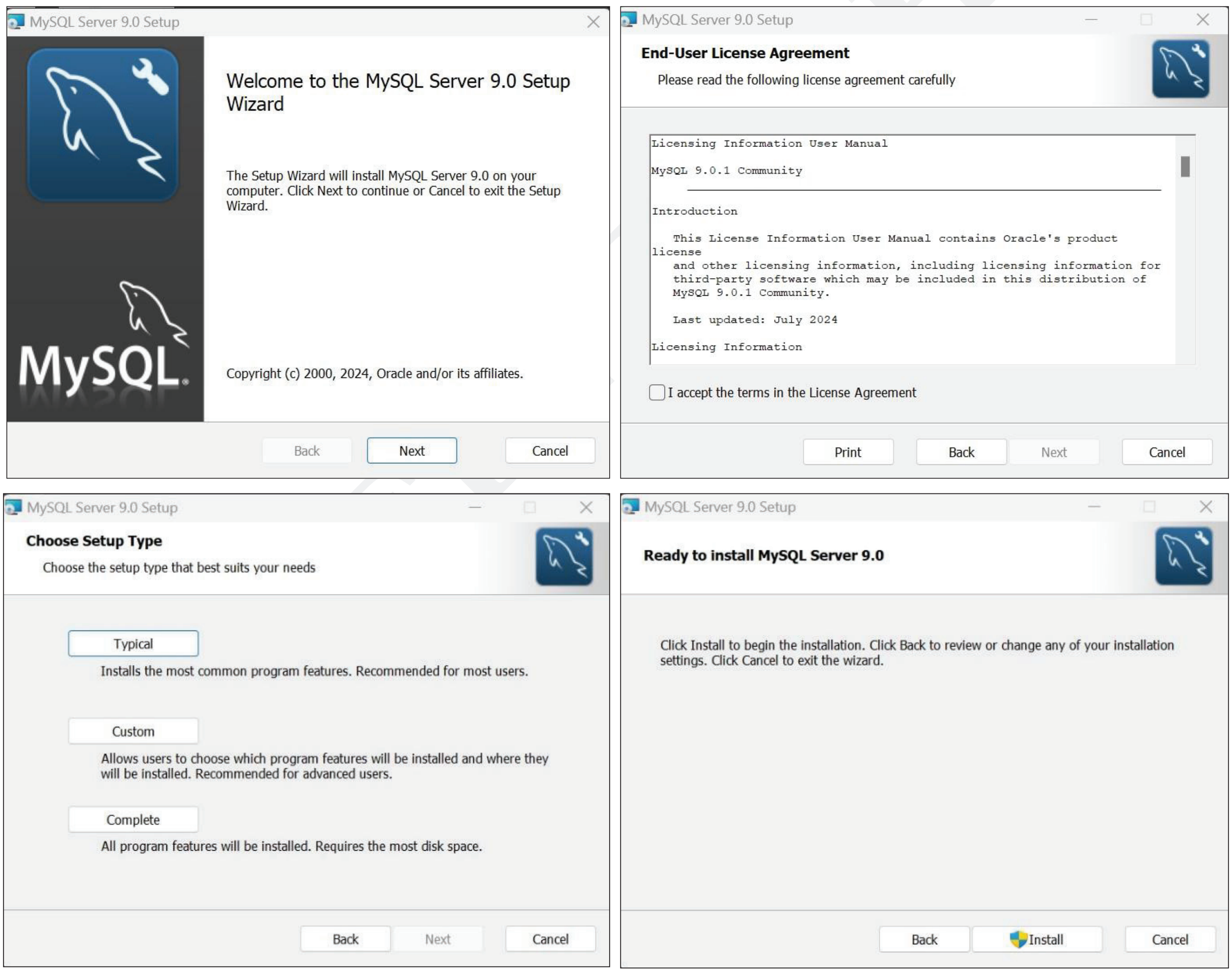


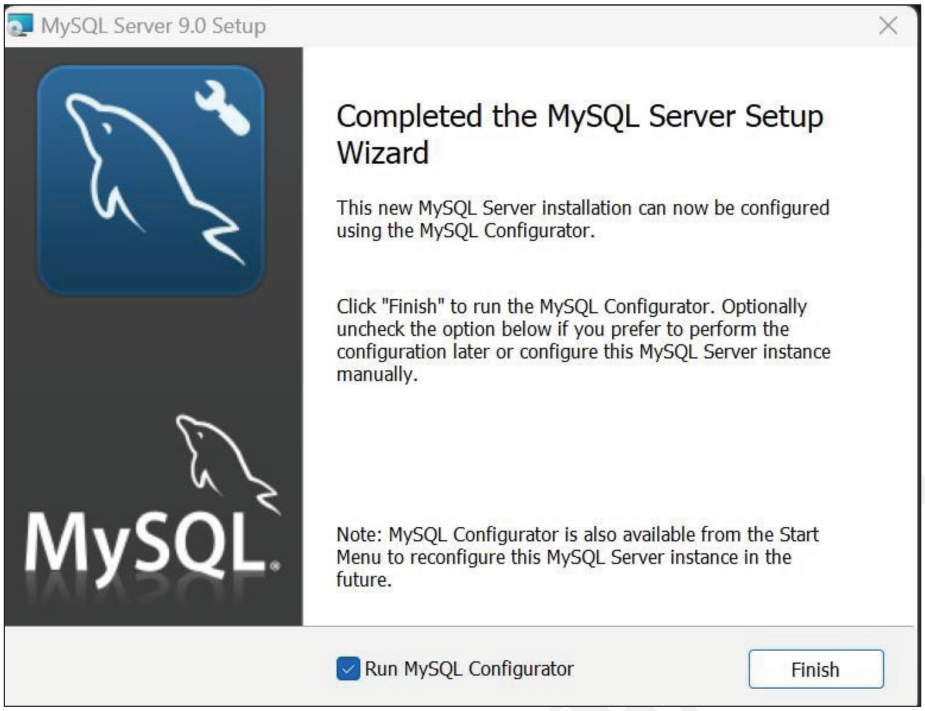
चित्र 1. 7: MySQL कम्युनिटी डाउनलोड का दूसरा चरण

डाउनलोड होने के पश्चात, डाउनलोड के लोकेशन पे जाये.

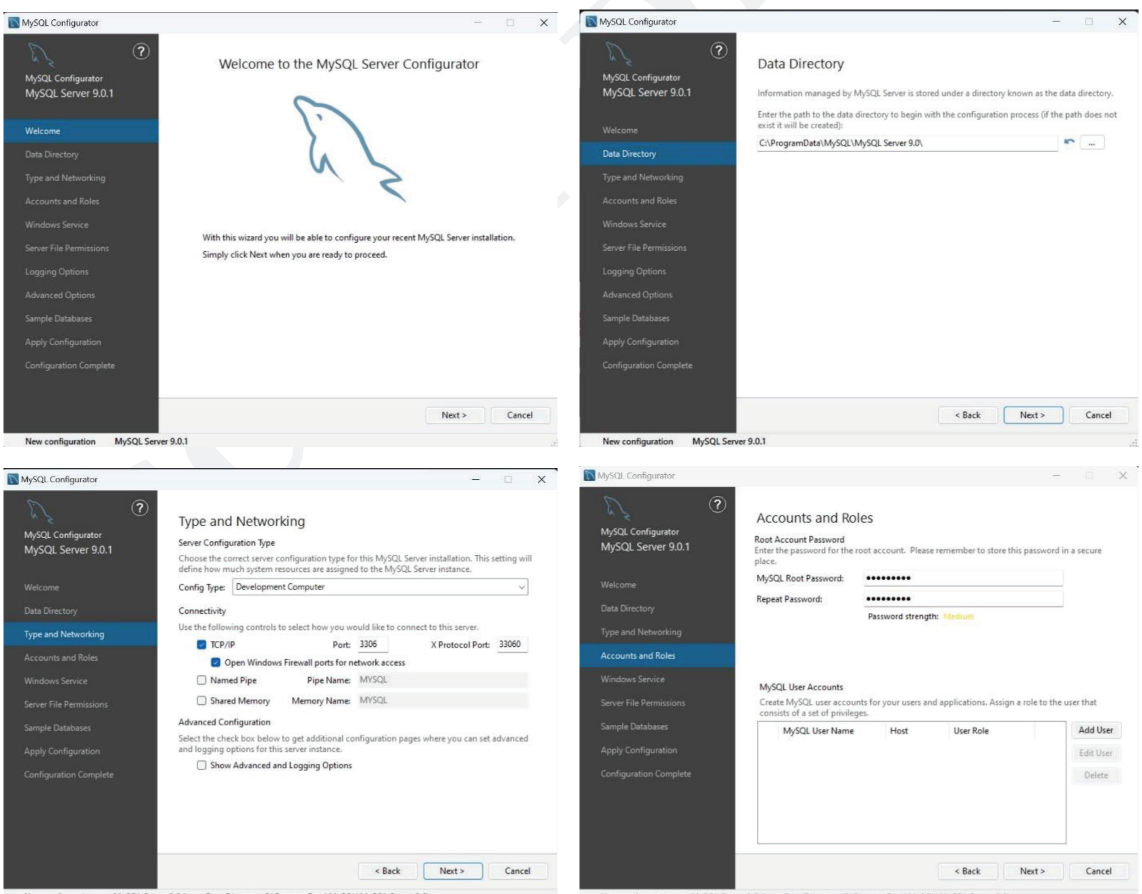


इनस्टॉल करने के लिए विंडो इंस्टालर पैकेज पे डबल क्लिक करे

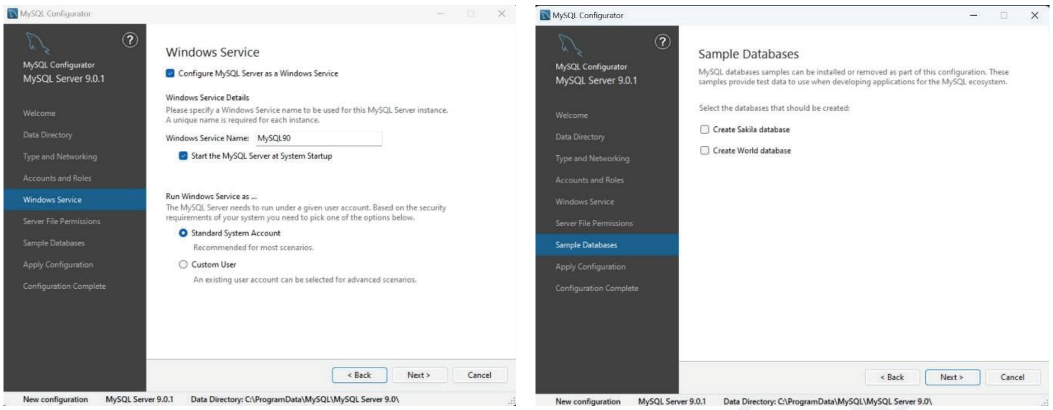




चित्र 1. 8: MySQL सर्वर सेटअप विज़ार्ड

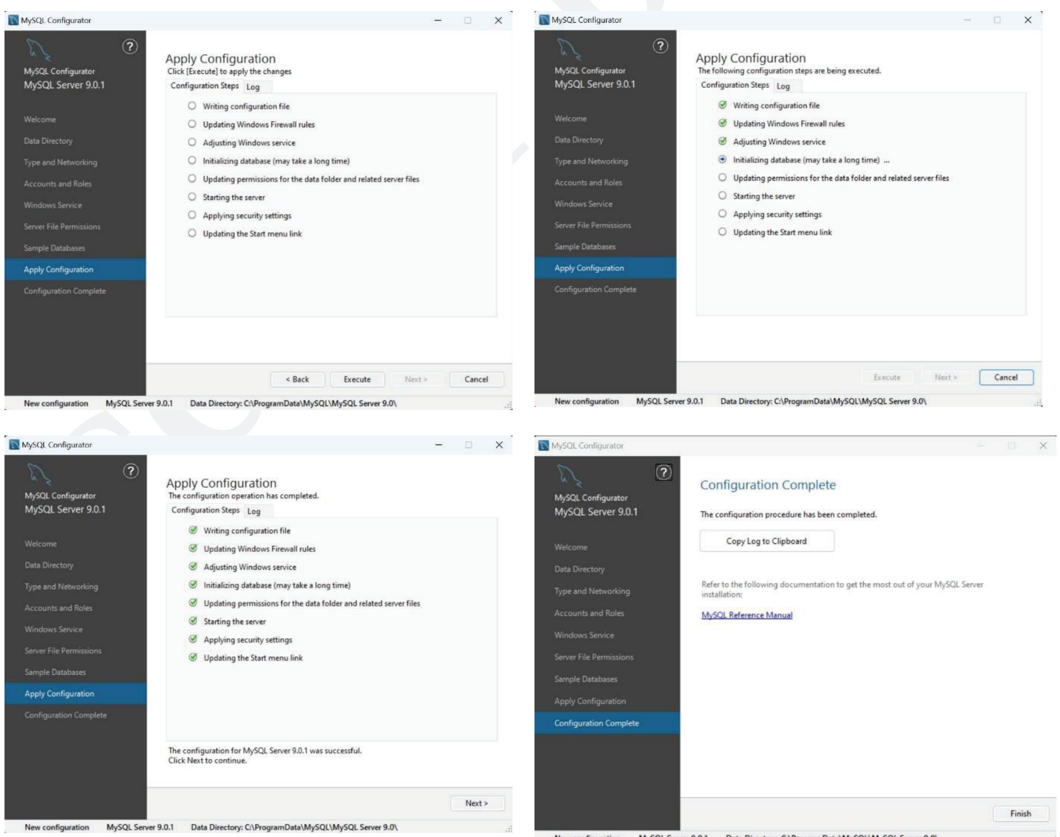


MySQL को कॉन्फ़िगर करने के लिए पहले सेटअप पे नेक्स्ट करे, डाटा डायरेक्टरी के लिए पाथ सेट करे, जहाँ आपको अपना MySQL इनस्टॉल करना है। इसके बाद टाइप्स और नेटवर्किंग को नेक्स्ट करके, एकाउंट्स एंड रोल्स में MySQL रुट पासवर्ड सेट करना होता है।



चित्र 1. 9: विंडो सर्विस नेम & सैंपल डेटाबेस

टिप्पणी (Note): MySQL को कॉन्फ़िगर करने की प्रक्रिया में, पासवर्ड के लिए एक संकेत प्रदर्शित किया जाएगा। यहां, आपको एक पासवर्ड दर्ज करना चाहिए और इसे याद रखना चाहिए, क्योंकि हर बार MySQL शुरू करने पर इसकी आवश्यकता होगी।



चित्र 1. 10: इन प्रोसेसेज के माध्यम से कन्फुगरेशन कम्पलीट हो जायेगा।

टिप्पणीः इसे इनस्टॉल और कॉन्फ़िगर करते समय विंडो में दिए गए डिफ़ॉल्ट विकल्पों का चयन करते रहें। यदि कॉन्फ़िगरेशन में कोई त्रुटि नहीं आती है, तो जानकारी दी जाएगी कि कॉन्फ़िगरेशन फ़ाइल बनाई गई थी, MySQL सर्वर स्थापित और प्रारंभ किया गया था, और सुरक्षा सेटिंग्स लागू की गई थीं।

1.6 SQL में डेटाबेस को परिभाषित और संशोधित करने के लिए कमांड्स

1. क्रिएट टेबल (Create Table) कमांडः

यह कमांड एक नई टेबल या रिलेशन बनाने के लिए प्रयोग किया जाता है। इसका सिंटैक्स निम्नलिखित हैः

CREATE TABLE <टेबल का नाम>

(<कॉलम 1> <डेटा टाइप> [कंस्ट्रेंट] ,

<कॉलम 2> <डेटा टाइप> [कंस्ट्रेंट],

<कॉलम 3> <डेटा टाइप> [कंस्ट्रेंट]);

यहाँ, `[]` विकल्प के लिए है (वैकल्पिक)।

इस कमांड में `CREATE TABLE` कीवर्ड के बाद उस टेबल का नाम लिखा जाता है जिसे आप बनाना चाहते हैं। इसके बाद, कोष्ठकों के अंदर कॉलम का विवरण लिखा जाता है, जिसमें कॉलम का नाम, उसका डेटा टाइप और वैकल्पिक कंस्ट्रेंट होते हैं। आप जितने चाहें उतने कॉलम बना सकते हैं। हर कॉलम के विवरण को अल्पविराम (`,`) से अलग किया जाता है।

SQL के सभी स्टेटमेंट्स का अंत सेमीकोलन (`;`) से होना चाहिए।

डेटा टाइप्स (Data Types)

डेटाबेस में डेटा को स्टोर करने के लिए विभिन्न प्रकार के डेटा टाइप्स का उपयोग किया जाता है। नीचे दिए गए डेटा टाइप्स का मतलब और उदाहरण समझाया गया हैः

SCERT DELHI

डेटा टाइप	अर्थ (Meaning)	उदाहरण (Example)
CHAR(n)	यह एक निश्चित लंबाई का कैरेक्टर स्ट्रिंग होता है। 'n' वह संख्या है जो दर्शाता है कि इसमें कितने कैरेक्टर होंगे।	CHAR(5) : "Ashok", "Vijay"
VARCHAR(n)	यह एक परिवर्तनीय लंबाई का कैरेक्टर स्ट्रिंग होता है। 'n' वह अधिकतम संख्या है, जो स्ट्रिंग में कैरेक्टर की संख्या को निर्धारित करता है।	VARCHAR(15) : "Vijay Kumar", "Ashok Sen"
DATE	यह डेट को YYYY-MM-DD के रूप में स्टोर करता है।	DATE : '2014-03-20'
INTEGER	यह एक पूर्णांक (Integer) संख्या होती है।	INTEGER : 23, 56789
DECIMAL (m, d)	यह एक निश्चित बिंदु संख्या होती है। 'm' उस संख्या के महत्वपूर्ण अंकों की संख्या को दर्शाता है और 'd' दशमलव बिंदु के बाद कितने अंकों को स्टोर किया जा सकता है। अगर 'd' 0 हो या निर्दिष्ट न किया गया हो, तो संख्या में दशमलव भाग नहीं होता।	DECIMAL (5,2) : 999.99, -567.78
DECIMAL (m)	यह भी एक निश्चित बिंदु संख्या होती है, लेकिन इसमें केवल महत्वपूर्ण अंकों की संख्या (m) दी जाती है, और इसमें कोई दशमलव अंक नहीं होता।	DECIMAL (5) : 23456, 99999

तालिका 1: आमतौर पर उपयोग किए जाने वाले डेटा प्रकार

यदि हमें एक स्कूल में काम करने वाले सभी शिक्षकों का डेटाबेस बनाना है, तो हम निम्नलिखित जानकारी को स्टोर करने के लिए एक टेबल बनाएंगे:

Teacher टेबल में निम्नलिखित कॉलम होंगे:

* Teacher_ID: शिक्षक का यूनिक (विशिष्ट) आईडी
* First_Name: शिक्षक का पहला नाम
* Last_Name: शिक्षक का अंतिम नाम
* Gender: शिक्षक का लिंग (पुरुष/महिला)

* Date_of_Birth: शिक्षक की जन्म तिथि
* Salary: शिक्षक का वेतन
* Dept_No: वह विभाग जिसमें शिक्षक काम करता है

SQL कमांड का उपयोग करके इस टेबल को बनाने के लिए हम निम्नलिखित CREATE TABLE कमांड का प्रयोग करेंगे:

```
CREATE TABLE Teacher
(
 Teacher_ID INTEGER,
 First_Name VARCHAR(20),
 Last_Name VARCHAR(20),
 Gender CHAR(1),
 Salary DECIMAL(10,2),
 Date_of_Birth DATE,
 Dept_No INTEGER
);
```

यह कमांड Teacher नाम की एक टेबल बनाएगी, जिसमें ऊपर बताए गए कॉलम होंगे।

* Teacher_ID को एक पूर्णांक (Integer) के रूप में स्टोर किया जाएगा।
* First_Name और Last_Name को 20 वर्णों तक के स्ट्रिंग (VARCHAR) के रूप में स्टोर किया जाएगा।
* Gender को 1 अक्षर (CHAR) के रूप में स्टोर किया जाएगा, जैसे 'M' (पुरुष) या 'F' (महिला)।
* Salary को दशमलव संख्या (DECIMAL) के रूप में स्टोर किया जाएगा, जिससे शिक्षक का वेतन सही तरीके से स्टोर किया जा सके।

SCERT DELHI

* Date_of_Birth को डेट (DATE) के रूप में स्टोर किया जाएगा।
* Dept_No को भी एक पूर्णांक (Integer) के रूप में स्टोर किया जाएगा, जो विभाग का नंबर दर्शाएगा।

इस प्रकार, यह कमांड एक नई टेबल बनाएगा, जिसमें स्कूल के सभी शिक्षकों की जानकारी सुरक्षित होगी।

ऊपर दिए गए टेबल को MySQL में बनाने के लिए निम्नलिखित चरणों का पालन करें:

1. MySQL 9.0.1 Command Line Client को खोलें:

* अपने कंप्यूटर के Start Menu में जाएं।
* MySQL 9.0.1 Command Line Client पर क्लिक करें।

2. पासवर्ड डालें:

* जब Command Line Client खुलता है, तो यह आपसे पासवर्ड पूछेगा। अपने MySQL का पासवर्ड डालें और एंटर दबाएं।

3. डेटाबेस चुनें:

* उस डेटाबेस को चुनें, जिसमें आप टेबल बनाना चाहते हैं। इसके लिए USE database_name; कमांड का उपयोग करें। उदाहरण:

USE school_database;

4. CREATE TABLE कमांड दर्ज करें:

* अब Teacher टेबल बनाने के लिए नीचे दिया गया CREATE TABLE कमांड दर्ज करें और एंटर दबाएं:

```
CREATE TABLE Teacher
(
Teacher_ID INTEGER,
First_Name VARCHAR(20),
Last_Name VARCHAR(20),
Gender CHAR(1),
```

```
Salary DECIMAL(10,2),
Date_of_Birth DATE,
Dept_No INTEGER
);
```

5. सफलता की पुष्टि करें:

1. अगर कमांड सही ढंग से चलता है, तो आपको "Query OK, 0 rows affected" जैसा संदेश मिलेगा, जो इंगित करता है कि टेबल सफलतापूर्वक बन गया है।

अब आपके डेटाबेस में Teacher नामक टेबल तैयार है, जिसमें स्कूल के शिक्षकों की जानकारी संग्रहीत की जा सकती है।

अब अगला कदम एक नया डेटाबेस बनाना है, जिसके लिए हम CREATE DATABASE कमांड का उपयोग करेंगे। इसे करने के लिए निम्नलिखित चरणों का पालन करें:

1. MySQL Command Line Client को खोलें:

- Start Menu में जाकर MySQL 9.0.1 Command Line Client पर क्लिक करें।

2. पासवर्ड दर्ज करें:

- जब Command Line खुलता है, तो अपने MySQL का पासवर्ड दर्ज करें और एंटर दबाएं।

3. CREATE DATABASE कमांड का उपयोग करें:

- एक नया डेटाबेस बनाने के लिए नीचे दिया गया कमांड टाइप करें और एंटर दबाएं:

CREATE DATABASE school_database;

* यहाँ, school_database डेटाबेस का नाम है। आप इसे अपनी आवश्यकता के अनुसार बदल सकते हैं।

4. सफलता की पुष्टि करें:

* अगर कमांड सफलतापूर्वक चलता है, तो आपको "Query OK, 1 row affected" जैसा संदेश मिलेगा, जो दर्शाता है कि डेटाबेस सफलतापूर्वक बन गया है।

SCERT DELHI

5. **डेटाबेस का उपयोग करें:**

* अब इस डेटाबेस में टेबल बनाने के लिए इसे चुनें:

```
USE school_database;
```

अब आपका डेटाबेस school_database सफलतापूर्वक तैयार हो गया है, और आप इस डेटाबेस में टेबल्स बना सकते हैं तथा डेटा स्टोर कर सकते हैं।

अब आप Teacher टेबल बना सकते हैं, जिसे School डेटाबेस के साथ जोड़ा जाएगा। इस प्रक्रिया को पूरा करने के लिए निम्नलिखित चरणों का पालन करें:

1. **डेटाबेस का चयन करें:**

* सबसे पहले, उस डेटाबेस का चयन करें जिसमें आप टेबल बनाना चाहते हैं। इसके लिए कमांड टाइप करें:

USE School;

* यहाँ School आपके डेटाबेस का नाम है।

2. **Teacher टेबल बनाने के लिए CREATE TABLE कमांड का उपयोग करें:**

* अब, नीचे दिए गए SQL कमांड का उपयोग करके Teacher नाम की टेबल बनाएँ:

```
CREATE TABLE Teacher
(
Teacher_ID INTEGER,
First_Name VARCHAR(20),
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2),
Date_of_Birth DATE,
Dept_No INTEGER
```

SCERT DELHI

);

* **इस कमांड में:**

  - Teacher_ID: शिक्षक की एक यूनिक आईडी है।
  - First_Name: शिक्षक का पहला नाम।
  - Last_Name: शिक्षक का अंतिम नाम।
  - Gender: शिक्षक का लिंग ('M' या 'F')।
  - Salary: शिक्षक का वेतन, जिसमें दशमलव (decimal) शामिल है।
  - Date_of_Birth: शिक्षक की जन्म तिथि।
  - Dept_No: शिक्षक जिस विभाग में कार्यरत हैं, उसका विभाग नंबर।

3. **कमांड का निष्पादन करें:**

   * अगर टेबल सफलतापूर्वक बन जाती है, तो आपको "Query OK" का संदेश दिखाई देगा, जो इंगित करता है कि Teacher टेबल अब School डेटाबेस में मौजूद है और डेटा डालने के लिए तैयार है।

इस प्रकार, आपने Teacher टेबल को School डेटाबेस के साथ जोड़कर सफलतापूर्वक बना लिया है।

सत्यापित करने के लिए, आप SHOW TABLES कमांड का उपयोग कर सकते हैं। यह कमांड वर्तमान डेटाबेस में बनाई गई सभी टेबल्स को प्रदर्शित करता है। इस कमांड का उपयोग करके आप देख सकते हैं कि Teacher टेबल सफलतापूर्वक बनाई गई है या नहीं।

कमांड का उपयोग:

SHOW TABLES;

प्रक्रिया:

1. सुनिश्चित करें कि आपने सही डेटाबेस का चयन कर लिया है। इसके लिए पहले USE School; कमांड का उपयोग करें।

SCERT DELHI

2. फिर, SHOW TABLES; टाइप करें और एंटर दबाएँ।

3. यदि Teacher टेबल सही से बनी है, तो इस कमांड के आउटपुट में यह टेबल लिस्ट में दिखाई देगी।

इस प्रकार, आप सुनिश्चित कर सकते हैं कि Teacher टेबल सही डेटाबेस में मौजूद है।

1.7 डेटाबेस प्रतिबंध (Constraints):

DBMS कई तरह के प्रतिबंध लागू कर सकता है ताकि डेटाबेस पर सुचारू रूप से कार्य हो सके। ये प्रतिबंध टेबल बनाने के दौरान ही निर्दिष्ट किए जा सकते हैं, जैसे कि नीचे दिए गए उदाहरण में दिखाया गया है:

a) NOT NULL:

इस प्रतिबंध का उपयोग यह सुनिश्चित करने के लिए किया जाता है कि किसी विशेष कॉलम में NULL मान की अनुमति नहीं है। उदाहरण के लिए, Teacher टेबल में First_Name कॉलम में NULL मान की अनुमति नहीं है, यानी प्रत्येक शिक्षक का नाम होना आवश्यक है। इस स्थिति में NOT NULL प्रतिबंध का उपयोग किया जा सकता है।

```
CREATE TABLE TEACHER
(
Teacher_ID INTEGER,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2),
Date_of_Birth DATE,
Dept_No INTEGER
);
```

ऊपर दिए गए कोड में, First_Name कॉलम पर NOT NULL प्रतिबंध लगाया गया है, जिसका

SCERT DELHI

मतलब है कि इस कॉलम में NULL मान नहीं डाला जा सकता।

b) DEFAULT:

इस प्रतिबंध का उपयोग तब किया जाता है जब उपयोगकर्ता किसी कॉलम में मान नहीं डालता है। ऐसे में डिफ़ॉल्ट मान अपने आप उस कॉलम में भर दिया जाएगा। उदाहरण के लिए, यदि शिक्षक की Salary का मान उपयोगकर्ता द्वारा नहीं दिया गया है, तो डेटाबेस में यह मान अपने आप 40000 हो जाएगा। यह माना गया है कि हर शिक्षक को न्यूनतम वेतन 40000 दिया जाना चाहिए।

```
CREATE TABLE TEACHER
(
Teacher_ID INTEGER,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER
);
```

ऊपर के कोड में, Salary कॉलम में DEFAULT प्रतिबंध का उपयोग करके 40000 का मान निर्धारित किया गया है। यदि किसी शिक्षक का वेतन नहीं दिया गया है, तो अपने आप 40000 भर जाएगा।

इस प्रकार, NOT NULL और DEFAULT प्रतिबंध का उपयोग करके डेटा की अखंडता और आवश्यक शर्तों को सुनिश्चित किया जा सकता है।

MySQL में, यदि आप बनाई गई तालिकाओं की संरचना और विवरण देखना चाहते हैं, तो DESC कमांड का उपयोग किया जा सकता है। ऊपर शिक्षक द्वारा बनाई गई तालिका का विवरण इस प्रकार है:

SCERT DELHI

Field	Type	Null	Key	Default	Extra
Teacher_ID	int	YES		NULL	
First_Name	varchar(20)	NO		NULL	
Last_Name	varchar(20)	YES		NULL	
Gender	char(1)	YES		NULL	
Salary	decimal(10,2)	YES		40000	
Date_of_Birth	date	YES		NULL	
Dept_No	int	YES		NULL	

7 rows in set (0.01 sec)

c) CHECK Constraint:

CHECK constraint का उपयोग किसी attribute (कॉलम) के मानों को एक निश्चित सीमा (range) के भीतर रखने के लिए किया जाता है। उदाहरण के लिए, मान लीजिए कि हमें यह सुनिश्चित करना है कि किसी भी टीचर का Dept_No (विभाग संख्या) 110 से अधिक नहीं हो। इसे हम CHECK constraint का उपयोग करके निर्धारित कर सकते हैं।

उदाहरणः

```
CREATE TABLE TEACHER
(
Teacher_ID INTEGER,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER CHECK (Dept_No <= 110)
```

SCERT DELHI

);

यहां पर Dept_No कॉलम के लिए CHECK constraint का मतलब है कि Dept_No का मान 110 से अधिक नहीं हो सकता।

नोटः MySQL में CHECK constraint को लागू तो किया जा सकता है, लेकिन MySQL इसे पूरी तरह से सपोर्ट नहीं करता है, इसलिए यह एरर नहीं देगा, लेकिन ध्यान रखना कि यह कार्यवाही कुछ मामलों में नहीं होगी।

d) KEY CONSTRAINT:

PRIMARY KEY constraint का उपयोग किसी तालिका में एक अद्वितीय (unique) पहचान बनाने के लिए किया जाता है। यदि किसी तालिका का प्राथमिक की (Primary Key) केवल एक कॉलम से बनी हो, तो उस कॉलम को सीधे PRIMARY KEY के रूप में निर्दिष्ट किया जा सकता है।

उदाहरणः

```
CREATE TABLE TEACHER
(    Teacher_ID INTEGER PRIMARY KEY,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER
);
```

Field	Type	Null	Key	Default	Extra
Teacher_ID	int	YES	PRI	NULL	
First_Name	varchar(20)	NO		NULL	

SCERT DELHI

Last_Name	varchar(20)	YES		NULL	
Gender	char(1)	YES		NULL	
Salary	decimal(10,2)	YES		40000	
Date_of_Birth	date	YES		NULL	
Dept_No	int	YES		NULL	

7 rows in set (0.02 sec)

यहां Teacher_ID को PRIMARY KEY के रूप में घोषित किया गया है। इसका मतलब है कि Teacher_ID का हर मान यूनिक (unique) होगा, और यह तालिका के प्रत्येक रिकॉर्ड को विशिष्ट रूप से पहचानने के लिए प्रयोग किया जाएगा।

संक्षेप में:

* **CHECK Constraint:** इसका उपयोग किसी कॉलम के मानों को सीमित करने के लिए किया जाता है। उदाहरण के तौर पर, Dept_No को 110 से अधिक नहीं होने देना।

* **PRIMARY KEY Constraint:** इसका उपयोग किसी कॉलम को तालिका का अद्वितीय पहचानकर्ता बनाने के लिए किया जाता है, जिससे प्रत्येक रिकॉर्ड को विशिष्ट रूप से पहचाना जा सके।

यहां पर डेटाबेस के बारे में कुछ महत्वपूर्ण बातें दी गई हैं, जिन्हें हम आसान हिंदी में समझाते हैं:

1. NOT NULL और PRIMARY KEY का उपयोग:

* जैसे कि ऊपर दिखाया गया है, First_Name को NOT NULL के रूप में डिफाइन किया गया है, जिसका मतलब है कि First_Name में कोई मान (value) खाली (NULL) नहीं हो सकता।

* Teacher_ID को PRIMARY KEY के रूप में डिफाइन किया गया है, जिसका मतलब है कि Teacher_ID का हर मान यूनिक (विशिष्ट) होना चाहिए और उसे NULL नहीं किया जा सकता।

* Salary के लिए डिफॉल्ट मान 40000 दिया गया है, जिससे यदि कोई शिक्षक वेतन (salary) नहीं देता है, तो 40000 की मान (value) खुद ही भर दी जाएगी।

उदाहरण:

```
CREATE TABLE TEACHER
(
Teacher_ID INTEGER PRIMARY KEY,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER
);
```

2. PRIMARY KEY में कई कॉलम (Attributes):

यदि किसी PRIMARY KEY में एक से अधिक कॉलम (attributes) शामिल हैं, तो आपको उन्हें एक लिस्ट के रूप में परिभाषित करना होगा। जैसे कि अगर Teacher_ID और Date_of_Birth दोनों मिलकर PRIMARY KEY बनाते हैं, तो आप इसे इस प्रकार डिफाइन करेंगे:

उदाहरण:

```
CREATE TABLE TEACHER
(
Teacher_ID INTEGER,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
```

SCERT DELHI

```
Dept_No INTEGER,
PRIMARY KEY (Teacher_ID, Date_of_Birth)
);
```

* PRIMARY KEY द्वारा दिए गए कॉलम खुद-ब-खुद NOT NULL हो जाते हैं, इसलिए आपको NOT NULL constraint को अलग से नहीं लिखने की जरूरत नहीं है।

3. REFERENTIAL INTEGRITY CONSTRAINT:

* FOREIGN KEY का उपयोग REFERENTIAL INTEGRITY सुनिश्चित करने के लिए किया जाता है। इसका मतलब है कि एक टेबल में मौजूद डेटा दूसरे टेबल में मौजूद डेटा से जुड़ा हुआ होना चाहिए।
* उदाहरण के लिए, Dept_No कॉलम को Department टेबल के Dept_ID कॉलम को संदर्भित करने के लिए FOREIGN KEY बनाया जाता है।

उदाहरणः

```
CREATE TABLE Department
(
Dept_ID INTEGER PRIMARY KEY,
Dept_Name VARCHAR(20) NOT NULL
);
CREATE TABLE Teacher
(
Teacher_ID INTEGER PRIMARY KEY,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
```

SCERT DELHI

```
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER,
FOREIGN KEY (Dept_No) REFERENCES Department(Dept_ID)
);
```

* यहाँ, Dept_No कॉलम Department टेबल के Dept_ID कॉलम को FOREIGN KEY द्वारा संदर्भित कर रहा है। इसका मतलब है कि Teacher टेबल में जो भी Dept_No है, वह Department टेबल के Dept_ID के मान के अनुरूप होना चाहिए।

MySQL में, यदि आप बनाई गई तालिकाओं की संरचना और विवरण देखना चाहते हैं, तो DESC कमांड का उपयोग किया जा सकता है। ऊपर शिक्षक द्वारा बनाई गई तालिका का विवरण इस प्रकार है:

Field	Type	Null	Key	Default	Extra
Teacher_ID	int	YES	PRI	NULL	
First_Name	varchar(20)	NO		NULL	
Last_Name	varchar(20)	YES		NULL	
Gender	char(1)	YES		NULL	
Salary	decimal(10,2)	YES		40000	
Date_of_Birth	date	YES		NULL	
Dept_No	int	YES	MUL	NULL	

7 rows in set (0.10 sec)

नोट : DROP और DELETE

DROP और DELETE के बीच अंतर

विषय	DROP	DELETE
प्रयोग	पूरी टेबल, डेटाबेस या ऑब्जेक्ट्स को हटाने के लिए उपयोग किया जाता है।	टेबल की कुछ या सभी पंक्तियों (records) को हटाने के लिए उपयोग किया जाता है।
स्ट्रक्चर पर प्रभाव	DROP कमांड टेबल की संरचना (structure) को पूरी तरह से हटा देता है।	DELETE कमांड टेबल की संरचना को बरकरार रखता है, सिर्फ डेटा हटाया जाता है।
रिकॉर्ड्स पर प्रभाव	टेबल की सभी पंक्तियाँ और उसके साथ टेबल की संरचना भी हट जाती है।	केवल चयनित या सभी पंक्तियों को हटाता है, लेकिन टेबल की संरचना नहीं हटती।
रोलबैक (Undo) संभव	एक बार DROP हो जाने पर इसे वापस नहीं किया जा सकता है।	DELETE के बाद TRUNCATE या ROLLBACK के माध्यम से डेटा वापस किया जा सकता है।
स्पीड	DROP तेज़ी से काम करता है क्योंकि यह पूरी संरचना हटा देता है।	DELETE अपेक्षाकृत धीमा होता है, क्योंकि यह प्रत्येक पंक्ति को एक-एक करके हटाता है।
प्रयोग की स्थिति	जब टेबल या अन्य डेटाबेस ऑब्जेक्ट को हमेशा के लिए हटाना हो।	जब कुछ या सभी डेटा को अस्थायी रूप से हटाना हो, लेकिन संरचना को बनाए रखना हो।
Constraints पर प्रभाव	DROP कमांड से constraints, indexes, और अन्य सभी तालिका संबंधी जानकारी भी हट जाती है।	DELETE कमांड constraints और indexes को प्रभावित नहीं करता है।

1.8 Foreign Key Constraints and Actions in SQL (SQL में Foreign Key Constraints और Actions)

Foreign Key Constraints (फॉरेन की बाधाएँ): Foreign Key का उपयोग दो टेबल्स के बीच संबंध स्थापित करने के लिए किया जाता है। Foreign Key एक टेबल के कॉलम को दूसरे टेबल के Primary Key से जोड़ता है। इससे यह सुनिश्चित होता है कि एक टेबल में दर्ज किया गया डेटा दूसरे टेबल के डेटा से मेल खाता हो।

जब एक Foreign Key एक टेबल में होती है, तो यदि उस टेबल में कोई बदलाव किया जाता है (जैसे किसी डेटा का डिलीट होना या अपडेट होना), तो इसका असर दूसरे टेबल में जुड़ी हुई Foreign Key पर भी पड़ सकता है। इसके लिए हम ON DELETE और ON UPDATE का उपयोग करके यह निर्धारित कर सकते हैं कि बदलाव होने पर क्या कार्रवाई की जानी चाहिए।

1.8.1 Foreign Key Actions (Foreign Key कार्रवाई):

1. **ON DELETE (DELETE पर कार्रवाई):** जब referenced table से डेटा को डिलीट किया जाता है, तो उस डेटा से जुड़ी Foreign Key वाले डेटा पर क्या कार्रवाई करनी है, यह निर्धारित करने के लिए ON DELETE का उपयोग किया जाता है।

2. **ON UPDATE (UPDATE पर कार्रवाई):** जब referenced table के डेटा में कोई बदलाव होता है, तो उस डेटा से जुड़ी Foreign Key वाले डेटा पर क्या बदलाव किया जाएगा, यह निर्धारित करने के लिए ON UPDATE का उपयोग किया जाता है।

Actions (क्रियाएँ):

* **SET NULL:** जब डेटा को डिलीट या अपडेट किया जाता है, तो संबंधित Foreign Key को NULL कर दिया जाता है।

* **CASCADE:** जब डेटा को डिलीट या अपडेट किया जाता है, तो संबंधित डेटा भी डिलीट या अपडेट हो जाता है।

* **RESTRICT:** अगर संबंधित डेटा मौजूद है, तो डिलीट या अपडेट नहीं किया जा सकता।

उदाहरण (Example):

1. **SET NULL:** यदि Department टेबल से कोई विभाग डिलीट किया जाता है, तो Teacher टेबल में उस विभाग से संबंधित डेटा NULL हो जाएगा।

CREATE TABLE Teacher

```
(
Teacher_ID INTEGER PRIMARY KEY,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
```

SCERT DELHI

Gender CHAR(1),

Salary DECIMAL(10,2) DEFAULT 40000,

Date_of_Birth DATE,

Dept_No INTEGER,

FOREIGN KEY (Dept_No) REFERENCES Department (Dept_ID) ON DELETE SET NULL ON UPDATE SET NULL

);

यहां, अगर Department टेबल से Dept_ID डिलीट किया जाता है, तो Teacher टेबल में उस Dept_No से संबंधित सभी डेटा NULL हो जाएंगे।

2. **CASCADE:** अगर Department टेबल से किसी विभाग को डिलीट किया जाता है, तो Teacher टेबल से संबंधित सभी रिकॉर्ड भी डिलीट हो जाएंगे।

CREATE TABLE Teacher

(

Teacher_ID INTEGER PRIMARY KEY,

First_Name VARCHAR(20) NOT NULL,

Last_Name VARCHAR(20),

Gender CHAR(1),

Salary DECIMAL(10,2) DEFAULT 40000,

Date_of_Birth DATE,

Dept_No INTEGER,

FOREIGN KEY (Dept_No) REFERENCES Department (Dept_ID) ON DELETE CASCADE ON UPDATE CASCADE

);

SCERT DELHI

यहां, अगर Department टेबल से Dept_ID डिलीट या अपडेट किया जाता है, तो Teacher टेबल में संबंधित डेटा भी डिलीट या अपडेट हो जाएगा।

3. **RESTRICT:** अगर Teacher टेबल में कोई डेटा है, तो Department टेबल से संबंधित Dept_ID को डिलीट या अपडेट नहीं किया जा सकेगा।

CREATE TABLE Teacher

(

Teacher_ID INTEGER PRIMARY KEY,

First_Name VARCHAR(20) NOT NULL,

Last_Name VARCHAR(20),

Gender CHAR(1),

Salary DECIMAL(10,2) DEFAULT 40000,

Date_of_Birth DATE,

Dept_No INTEGER,

FOREIGN KEY (Dept_No) REFERENCES Department (Dept_ID) ON DELETE RESTRICT ON UPDATE RESTRICT);

यहां, RESTRICT का मतलब है कि अगर Teacher टेबल में उस Dept_No से संबंधित कोई रिकॉर्ड है, तो Department टेबल से उस Dept_ID को डिलीट या अपडेट नहीं किया जा सकेगा।

Self-Referencing Tables (स्व-उल्लेखनीय टेबल्स)

Self-Referencing Table वह टेबल होती है, जो अपने आप से जुड़ी होती है। इसका मतलब है कि किसी टेबल के एक कॉलम को उसी टेबल के किसी अन्य कॉलम के साथ Foreign Key के रूप में जोड़ा जाता है। उदाहरण के लिए, Employee टेबल में Manager_ID कॉलम, जो किसी अन्य कर्मचारी को संदर्भित करता है।

CREATE TABLE Employee

SCERT DELHI

```
(
Employee_ID INTEGER PRIMARY KEY,
Name VARCHAR(30),
Age INTEGER,
Salary DECIMAL(10,2),
Manager_ID INTEGER,
FOREIGN KEY (Manager_ID) REFERENCES Employee (Employee_ID)
);
```

यहां Manager_ID एक Foreign Key है जो Employee_ID से जुड़ी है, जिससे यह साबित होता है कि एक कर्मचारी का मैनेजर भी एक अन्य कर्मचारी हो सकता है।

Naming of Constraints (बाधाओं का नामकरण)

SQL में हम constraints (जैसे Primary Key, Foreign Key) को नाम दे सकते हैं, जिससे उन्हें आसानी से पहचाना और बदला जा सकता है। इसका उपयोग CONSTRAINT कीवर्ड के साथ किया जाता है।

CREATE TABLE Teacher

```
(
Teacher_ID INTEGER,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER,
```

SCERT DELHI

CONSTRAINT TEACHER_PK PRIMARY KEY (Teacher_ID),

CONSTRAINT TEACHER_FK FOREIGN KEY (Dept_No) REFERENCES Department(Dept_ID) ON DELETE SET NULL ON UPDATE SET NULL

);

यहां TEACHER_PK को Primary Key और TEACHER_FK को Foreign Key constraint का नाम दिया गया है।

Drop Table Command (Drop Table कमांड)

अगर आपको कोई टेबल डिलीट करनी हो, तो DROP TABLE कमांड का उपयोग किया जाता है। उदाहरणः

DROP TABLE Teacher CASCADE;

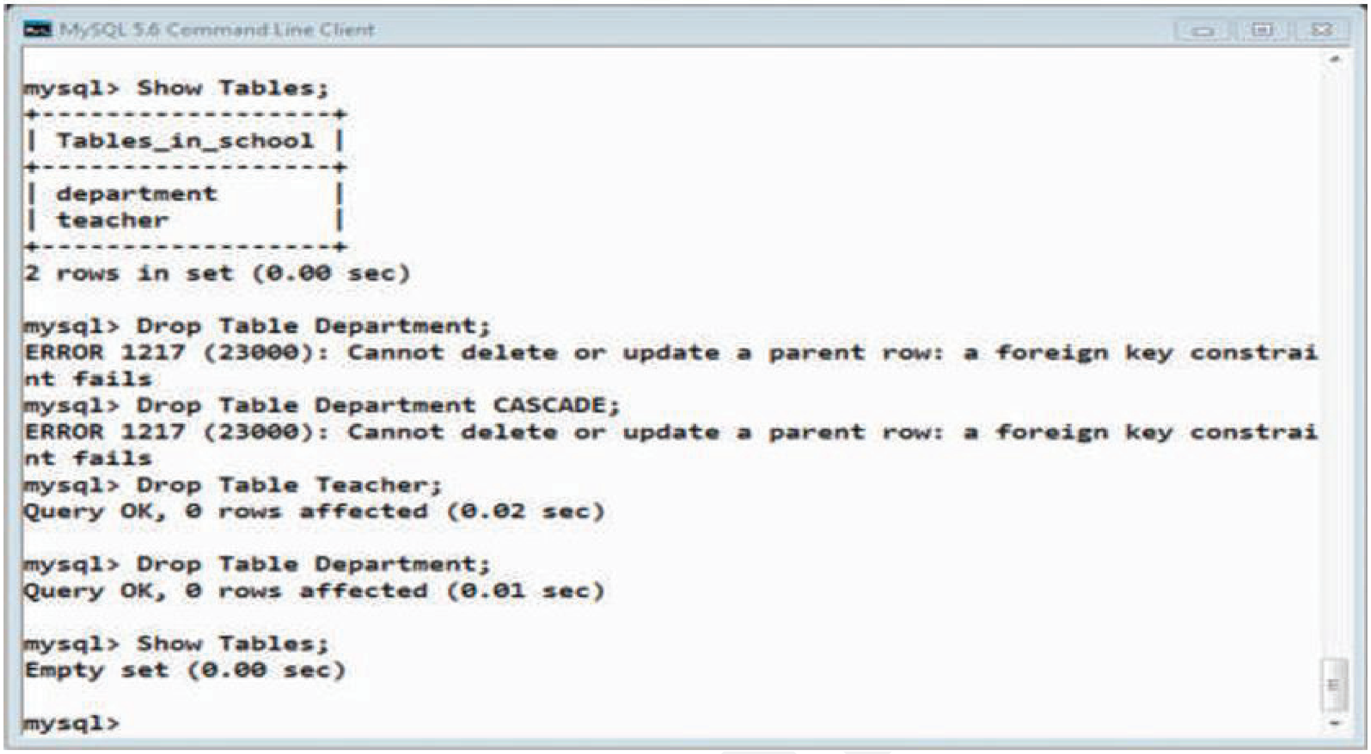
इसमें CASCADE का मतलब है कि इस टेबल से जुड़ी सभी constraints भी डिलीट हो जाएंगी। अगर आप चाहते हैं कि टेबल डिलीट ना हो यदि वह किसी अन्य टेबल में संदर्भित हो, तो RESTRICT का उपयोग किया जाता है।

DROP TABLE Teacher RESTRICT;

यहां, RESTRICT का मतलब है कि अगर कोई दूसरी टेबल इस टेबल को संदर्भित कर रही है, तो इस टेबल को डिलीट नहीं किया जा सकेगा।

नोटः MySQL 9.0.1 में RESTRICT और CASCADE ऑप्शन को सपोर्ट नहीं किया जाता है, हालांकि इन्हें पोर्टिंग को आसान बनाने के लिए अनुमति दी जाती है।

MySQL सर्वर 9.0.1 में, आप डेटाबेस से किसी तालिका को हटाने के लिए बस तालिका नाम के बाद ड्रॉप टेबल टाइप कर सकते हैं। हालाँकि, यदि हटाई जाने वाली तालिका को किसी अन्य तालिका में संदर्भित किया जा रहा है तो यह आपको तालिका को छोड़ने की अनुमति नहीं देगा.



```
MySQL 5.6 Command Line Client
mysql> Show Tables;
+------------------+
| Tables_in_school |
+------------------+
| department       |
| teacher          |
+------------------+
2 rows in set (0.00 sec)

mysql> Drop Table Department;
ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constrai
nt fails
mysql> Drop Table Department CASCADE;
ERROR 1217 (23000): Cannot delete or update a parent row: a foreign key constrai
nt fails
mysql> Drop Table Teacher;
Query OK, 0 rows affected (0.02 sec)

mysql> Drop Table Department;
Query OK, 0 rows affected (0.01 sec)

mysql> Show Tables;
Empty set (0.00 sec)

mysql>
```

जैसा कि ऊपर दिए गए उदाहरण में दिखाया गया है, Department और Teacher दो टेबल्स हैं। Teacher टेबल में एक Foreign Key है जो Department टेबल के Primary Key को संदर्भित करता है। इसका मतलब यह है कि Teacher टेबल में जो Dept_No है, वह Department टेबल के Dept_ID से जुड़ा होता है।

टेबल डिलीट करते समय समस्या:

जब हम Department टेबल को डिलीट करने की कोशिश करते हैं, तो MySQL हमें एक error दिखाता है। इसका कारण यह है कि Department टेबल को Teacher टेबल में संदर्भित किया जा रहा है। यानी Teacher टेबल में Dept_No कॉलम, Department टेबल के Dept_ID कॉलम को संदर्भित करता है, इसलिए Department टेबल को डिलीट नहीं किया जा सकता।

CASCADE ऑप्शन का काम:

आपने देखा होगा कि CASCADE ऑप्शन काम नहीं कर रहा है क्योंकि MySQL Server 5.6.20 में यह ऑप्शन सपोर्ट नहीं करता है। CASCADE ऑप्शन का मतलब होता है कि अगर Department टेबल को डिलीट किया जाए तो Teacher टेबल में उस विभाग से संबंधित डेटा भी डिलीट हो जाए, लेकिन यह इस वर्जन में काम नहीं करता।

समाधानः

Department टेबल को तभी डिलीट किया जा सकता है, जब Teacher टेबल (जो Department टेबल को संदर्भित कर रही है) को पहले डिलीट किया जाए। अगर Teacher टेबल को डिलीट कर दिया जाता है, तो अब Department टेबल को डिलीट करना संभव हो जाता है, क्योंकि अब Department टेबल का कोई संदर्भ नहीं रह गया है।

*   **ALTER TABLE कमांडः**

यह कमांड किसी मौजूदा टेबल की संरचना (structure) को बदलने के लिए उपयोग की जाती है। इसके माध्यम से निम्नलिखित बदलाव किए जा सकते हैंः

(a) कॉलम जोड़ना (Adding a Column):

मान लीजिए हमें Teacher टेबल में एक नया कॉलम "Age" जोड़ना है। इसके लिए निम्नलिखित कमांड का उपयोग किया जा सकता हैः

ALTER TABLE Teacher ADD Age INTEGER;

(b) कॉलम हटाना (Dropping a Column):

किसी कॉलम को हटाने के लिए, उस कॉलम को हटाने से पहले यह निर्दिष्ट करना होता है कि इसे हटाने के बाद क्या करना है, जैसे RESTRICT या CASCADE। यदि कॉलम अन्य टेबल्स में संदर्भित है तो RESTRICT इसे हटाने की अनुमति नहीं देता है, जबकि CASCADE उस कॉलम से जुड़ी सभी constraints (जैसे foreign key) को भी हटा देता है। उदाहरण के लिए, अगर हमें Dept_No कॉलम को Teacher टेबल से हटाना है तोः

ALTER TABLE Teacher DROP Dept_No CASCADE;

यह Dept_No कॉलम को Teacher टेबल से हटा देगा और यदि कोई foreign key constraint (जैसे TEACHER_FK) इस कॉलम को संदर्भित कर रहा है तो उसे भी हटा देगा।

अगर हमें Age कॉलम को हटाना है तोः

ALTER TABLE Teacher DROP Age;

(c) कॉलम बदलना (Altering a Column):

किसी कॉलम की परिभाषा को बदलने के लिए हम इसे बदल सकते हैं, जैसे डिफ़ॉल्ट मान (default value) को हटाना या नया डिफ़ॉल्ट मान सेट करना। उदाहरण के लिए, यदि Teacher टेबल में

SCERT DELHI

Salary कॉलम का डिफ़ॉल्ट मान 40000 है और हम इसे बदलना चाहते हैं:

ALTER TABLE Teacher ALTER Salary DROP DEFAULT;

ALTER TABLE Teacher ALTER Salary SET DEFAULT 30000;

(d) Keys को हटाना (Dropping Keys):

किसी foreign key या primary key को हटाने के लिए ALTER TABLE कमांड का उपयोग किया जाता है। उदाहरण के लिए, अगर हमें Teacher टेबल से TEACHER_FK foreign key को हटाना है:

ALTER TABLE Teacher DROP FOREIGN KEY TEACHER_FK;

प्राइमरी की (primary key) को हटाने के लिए:

ALTER TABLE Teacher DROP PRIMARY KEY TEACHER_PK;

यदि Teacher टेबल का एकमात्र primary key है, तो इसे हटाने का प्रयास करने पर त्रुटि (error) आ सकती है।

(e) Constraint जोड़ना (Adding a Constraint):

यदि हम foreign key constraint को फिर से जोड़ना चाहते हैं, तो निम्नलिखित कमांड का उपयोग किया जा सकता है:

ALTER TABLE Teacher ADD CONSTRAINT TEACHER_FK FOREIGN KEY (Dept_No) REFERENCES Department(Dept_ID) ON DELETE SET NULL ON UPDATE SET NULL;

* **INSERT कमांड:**

यह कमांड एक ट्यूपल (tuple) को तालिका (table) में डालने के लिए उपयोग की जाती है। हमें उस तालिका का नाम और ट्यूपल के मान (values) निर्दिष्ट करने होते हैं। उदाहरण के लिए, Teacher टेबल में एक नया ट्यूपल जोड़ने के लिए:

INSERT INTO Teacher

VALUES (101, 'Shanaya', 'Batra', 'F', 50000, '1984-08-11', 1);

यहां, Teacher टेबल के कॉलम के अनुक्रम (order) के अनुसार मान दर्ज किए गए हैं।

SCERT DELHI

दूसरी विधि में, हम कॉलम नामों के साथ मानों को निर्दिष्ट कर सकते हैं:

INSERT INTO Teacher

(First_Name, Last_Name, Gender, Teacher_ID, Date_of_Birth, Dept_No, Salary)

VALUES ('Shanaya', 'Batra', 'F', 101, '1984-08-11', 1, 50000);

इसमें हम स्पष्ट रूप से कॉलम नामों का उल्लेख करते हैं।

अगर किसी कॉलम का मान निर्दिष्ट नहीं किया गया है, तो उसका default मान उपयोग किया जाएगा। यदि default भी नहीं है, तो NULL मान लिया जाएगा। उदाहरण के लिए:

INSERT INTO Teacher

(First_Name, Last_Name, Gender, Teacher_ID, Date_of_Birth, Dept_No, Salary)

VALUES ("Shanaya", "Batra", 'F', 101, '1984-08-11', 1);

यहां Salary के लिए डिफ़ॉल्ट मान 40000 का उपयोग होगा।

* **UPDATE कमांड:**

यह कमांड एक या एक से अधिक टपल में दिए गए कॉलम के मानों को बदलने के लिए उपयोग की जाती है। उदाहरण के लिए, अगर हमें Teacher_ID = 101 वाले शिक्षक का Salary 55000 करना है:

UPDATE Teacher

SET Salary = 55000

WHERE Teacher_ID = 101;

हम algebraic expression का भी उपयोग कर सकते हैं। उदाहरण के लिए, अगर हमें Shanaya की Salary में 5000 का इज़ाफ़ा करना है:

UPDATE Teacher

SCERT DELHI

```
SET Salary = Salary + 5000
WHERE First_Name = 'Shanaya';
```

* **DELETE कमांडः**

यह कमांड एक या एक से अधिक टपल को हटाने के लिए उपयोग की जाती है। उदाहरण के लिए, अगर हमें Teacher_ID = 101 वाले शिक्षक को हटाना हैः

```
DELETE FROM Teacher
WHERE Teacher_ID = 101;
```

यदि WHERE क्लॉज (clause) नहीं दिया जाता है, तो यह तालिका की सभी पंक्तियों को हटा देगाः

```
DELETE FROM Teacher;
```

* **SELECT कमांडः**

यह कमांड डाटाबेस से जानकारी प्राप्त करने के लिए उपयोग की जाती है। SELECT कमांड का सामान्य रूप (syntax) निम्नलिखित हैः

```
SELECT <attribute list>
FROM <table list>
WHERE <condition>;
```

यहां SELECT के बाद उन कॉलमों के नाम दिए जाते हैं जिन्हें हम प्राप्त करना चाहते हैं, FROM के बाद तालिका का नाम आता है, और WHERE क्लॉज में उस डेटा को फ़िल्टर (filter) करने की शर्तें दी जाती हैं।

उदाहरण के लिए, मान लीजिए कि हमारे पास Department और Teacher टेबल्स हैंः

```
CREATE TABLE Department
(
Dept_ID INTEGER PRIMARY KEY,
```

SCERT DELHI

```
Dept_Name VARCHAR(30) NOT NULL
);
CREATE TABLE Teacher
(
Teacher_ID INTEGER,
First_Name VARCHAR(20) NOT NULL,
Last_Name VARCHAR(20),
Gender CHAR(1),
Salary DECIMAL(10,2) DEFAULT 40000,
Date_of_Birth DATE,
Dept_No INTEGER,
CONSTRAINT TEACHER_PK PRIMARY KEY (Teacher_ID),
CONSTRAINT TEACHER_FK FOREIGN KEY (Dept_No) REFERENCES Department (Dept_ID)
);
```

Teacher टेबल Department टेबल को संदर्भित करती है ताकि यह पता चल सके कि कौन सा शिक्षक किस विभाग में काम कर रहा है।

मान लीजिए कि Department और Teacher टेबल्स में निम्नलिखित डेटा INSERT किया गया है।

Department

Dept_ID	Dept_Name
1	Chemistry
2	Computer Science

SCERT DELHI

3	English
4	Hindi
Physics	
Commerce	
Biology	
Mathematics	
Economics	

TEACHER

Teacher_ID	First_Name	Last_Name	Gender	Salary	Date_of_Birth	Dept_No
101	Shanaya	Batra	F	50000	1984-08-11	1
102	Alice	Walton	F	48000	1983-02-12	3
103	Surbhi	Bansal	F	34000	1985-06-11	4
104	Megha	Khanna	F	38000	1979-04-06	4
105	Tarannum	Malik	F	54000	1978-04-22	5
106	Tarun	Mehta	M	50000	1980-08-21	2
107	Puneet	NULL	M	52500	1976-09-25	3
108	Namit	Gupta	M	49750	1981-10-19	1
109	Neha	Singh	F	49000	1984-07-30	7
110	Divya	Chaudhary	F	39000	1983-12-11	6
111	Saurabh	Pant	M	40000	1982-01-05	8
112	Sumita	Arora	F	40000	1981-10-10	9
113	Vinita	Ghosh	F	51500	1980-09-09	9
114	Vansh	NULL	M	53500	1982-05-04	2

Teacher टेबल में Dept_No कॉलम है जो यह दिखाता है कि हर शिक्षक किस विभाग से संबंधित है।

Dept_No को Department टेबल के Dept_ID के साथ जोड़ा जाता है, जो कि एक foreign key है। इसका मतलब यह है कि Teacher टेबल में हर शिक्षक का Dept_No वही मान होगा जो Department टेबल के Dept_ID कॉलम में होगा।

उदाहरण के लिए, यदि Teacher टेबल में एक शिक्षक है जिसका Dept_No = 1, तो इसका मतलब वह शिक्षक Chemistry विभाग से संबंधित है।

इस प्रकार, Teacher टेबल का Dept_No कॉलम Department टेबल के Dept_ID को संदर्भित करता है ताकि यह पता चले कि कौन सा शिक्षक किस विभाग से संबंधित है।

(a) Query: Teacher के ID=101 के बारे में सभी जानकारी प्राप्त करने के लिए।

इस query में हम Teacher_ID=101 वाले शिक्षक के बारे में सारी जानकारी प्राप्त करना चाहते हैं। इसमें सभी attributes (जैसे First_Name, Last_Name, Salary आदि) को

गतिविधि प्रश्नः MySQL कमांड्स पर आधारित अभ्यास

प्रश्न 1 : जब आप Department टेबल को डिलीट करने की कोशिश करते हैं, तो कौनसी समस्या उत्पन्न होती है? इसके समाधान के लिए क्या कदम उठाए जा सकते हैं?

प्रश्न 2: (a) निम्नलिखित स्थितियों के लिए सही ALTER TABLE कमांड लिखें: (a) Teacher टेबल में एक नया कॉलम "Age" जोड़ें।

(b) Teacher टेबल से Dept_No कॉलम को CASCADE ऑप्शन के साथ हटा दें।

(c) Teacher टेबल से TEACHER_FK foreign key को हटाएँ।

प्रश्न 3 : Teacher टेबल में निम्नलिखित डेटा जोड़ने के लिए सही INSERT कमांड लिखें:

शिक्षक का नाम: "Shanaya Batra",

Teacher_ID: 101,

जन्म तिथि: '1984-08-11',

विभाग नंबर: 1,

वेतन: 50000.

प्रश्न 4 :(a) निम्नलिखित स्थितियों के लिए सही UPDATE कमांड लिखें: (a) Teacher_ID = 101 वाले शिक्षक की Salary को 55000 करें।

(b) Teacher_ID = 101 वाले शिक्षक की Salary में 5000 की वृद्धि करें।

SELECT क्लॉज़ में specify किया गया है। एक आसान तरीका यह है कि हम asterisk (*) का उपयोग करें, जिसका मतलब है 'सभी attributes को select करो'।

1.9 एसक्यूएल क्वेरी (SQL Query)

a) SELECT *

FROM Teacher

WHERE Teacher_ID=101;

Explanation:

* SELECT * का मतलब है कि हम सभी attributes (जैसे Teacher_ID, First_Name, Last_Name, Salary, आदि) को देखना चाहते हैं।
* FROM Teacher का मतलब है कि यह query Teacher टेबल से डेटा लाएगी।
* WHERE Teacher_ID=101 का मतलब है कि हमें केवल उस शिक्षक की जानकारी चाहिए जिसका Teacher_ID 101 है।

Result: जब इस query को execute किया जाएगा, तो Teacher_ID=101 वाले शिक्षक की सारी जानकारी मिलेगी। उदाहरण के तौर पर, परिणाम कुछ इस तरह होगा:

Teacher_ID	First_ Name	Last_ Name	Gender	Salary	Date_of_ Birth	Dept_ No
101	Shanaya	Batra	F	50000	11-08-1984	1

इस table में हमें Teacher_ID=101 वाले शिक्षक का नाम Shanaya Batra, उनकी सैलरी 50000, और जन्मतिथि 1984-08-11 जैसी जानकारी प्राप्त हुई है।

(b) Query: सभी ऐसे शिक्षकों के नाम निकालने के लिए जिनकी सैलरी 50000 से अधिक है।

इस query में हम उन शिक्षकों के First_Name और Last_Name प्राप्त करना चाहते हैं जिनकी सैलरी 50000 से अधिक है। इसके लिए हम SELECT कमांड का उपयोग करेंगे और WHERE

क्लॉज़ में सैलरी की शर्त (salary > 50000) लगाएंगे।

SQL Query:

```
SELECT First_Name, Last_Name
FROM Teacher
WHERE salary > 50000;
```

Explanation:

* SELECT First_Name, Last_Name: इसका मतलब है कि हम सिर्फ शिक्षक के पहले नाम और अंतिम नाम (First_Name, Last_Name) प्राप्त करना चाहते हैं।
* FROM Teacher: यह क्वेरी Teacher टेबल से डेटा लेगी।
* WHERE salary > 50000: हम केवल उन शिक्षकों की जानकारी चाहते हैं जिनकी सैलरी 50000 से अधिक है।

Result: जब इस query को execute किया जाएगा, तो उन शिक्षकों के नाम दिखेंगे जिनकी सैलरी 50000 से अधिक है। उदाहरण के तौर पर, परिणाम कुछ इस प्रकार होगा:

First_Name	Last_Name
Tarannum	Malik
Puneet	Ghosh
Vinita	Vansh

इस table में हमें उन शिक्षकों के First_Name और Last_Name मिल रहे हैं जिनकी सैलरी 50000 से ज्यादा है।

(c) **Query: उन शिक्षकों का Teacher_ID, First_Name, Last_Name और Dept_No दिखाने के लिए जो विभाग संख्या 4 या 7 से संबंधित हैं।**

इस query में हम उन शिक्षकों की जानकारी प्राप्त करना चाहते हैं जो विभाग संख्या 4 या 7 से संबंधित हैं। इसके लिए हम SELECT कमांड का उपयोग करेंगे और WHERE क्लॉज़ में Dept_No को 4 या 7 के बराबर (OR) रखने की शर्त लगाएंगे।

SQL Query:

SCERT DELHI

SELECT Teacher_ID, First_Name, Last_Name, Dept_No

FROM Teacher

WHERE Dept_No = 4 OR Dept_No = 7;

Explanation:

* SELECT Teacher_ID, First_Name, Last_Name, Dept_No: इसका मतलब है कि हम शिक्षक का Teacher_ID, First_Name, Last_Name और Dept_No प्राप्त करना चाहते हैं।
* FROM Teacher: यह query Teacher टेबल से डेटा लेगी।
* WHERE Dept_No = 4 OR Dept_No = 7: हम केवल उन शिक्षकों की जानकारी चाहते हैं जिनका Dept_No (विभाग संख्या) 4 या 7 है। यह शर्त OR के साथ जोड़ी गई है, जिसका मतलब है कि दोनों में से कोई भी शर्त पूरी होने पर डेटा मिलेगा।

Result: जब इस query को execute किया जाएगा, तो उन शिक्षकों का डेटा दिखाई देगा जो विभाग संख्या 4 या 7 से संबंधित हैं। उदाहरण के लिए, परिणाम कुछ इस प्रकार हो सकता है:

Teacher_ID	**First_Name**	**Last_Name**	**Dept_No**
104	Megha	Khanna	4
106	Neha	Arora	7
109	Saurabh	Ghosh	7

इस table में हम देख सकते हैं कि ये शिक्षक विभाग संख्या 4 या 7 में आते हैं।

Result:

जब ऊपर दी गई query को execute किया जाएगा, तो परिणाम कुछ इस प्रकार दिखेगा:

Teacher_ID	**First_Name**	**Last_Name**	**Dept_No**
103	Surbhi	Singh	4
104	Megha	Khanna	4
109	Neha	Bansal	7

इसमें तीन शिक्षक हैं जिनका Dept_No या तो 4 है या 7 है।

Explanation of Boolean Operations:

* जैसा कि आप देख सकते हैं, WHERE क्लॉज़ में OR ऑपरेटर का उपयोग किया गया है। इसका मतलब है कि यदि कोई शिक्षक Dept_No 4 या Dept_No 7 से संबंधित है, तो उसका डेटा परिणाम में शामिल किया जाएगा।

* इसी तरह, AND और OR जैसे Boolean operations का उपयोग भी WHERE क्लॉज़ में किया जा सकता है, ताकि आप दो या दो से अधिक शर्तों के आधार पर डेटा फ़िल्टर कर सकें।

उदाहरणः

* **AND:** यदि आप चाहते हैं कि शिक्षक का Dept_No 4 और Salary 50000 से अधिक हो, तो query कुछ इस प्रकार होगीः

  SELECT Teacher_ID, First_Name, Last_Name, Dept_No

  FROM Teacher

  WHERE Dept_No = 4 AND Salary > 50000;

* OR: जैसा कि आपने देखा, OR का मतलब है कि किसी एक शर्त का सही होना भी पर्याप्त है।

  Boolean operations आपको डेटा को अधिक विशिष्ट रूप से फिल्टर करने की अनुमति देते हैं।

Query के बारे में:

हम इस query में सभी शिक्षकों के नाम और उनके संबंधित विभाग के नाम और नंबर निकालने का प्रयास कर रहे हैं। चूंकि शिक्षक और विभाग के डाटा को अलग-अलग तालिकाओं में संग्रहीत किया गया है, इसलिए इस query को दोनों तालिकाओं — Teacher और Department — की आवश्यकता होगी।

Example Query:

1. **Cross Join (जब कोई WHERE क्लॉज़ नहीं हो):**

   SELECT First_Name, Last_Name, Dept_ID, Dept_Name

   FROM Teacher, Department;

   इस query में कोई WHERE क्लॉज़ नहीं दिया गया है, इसलिए यह Cross Join कहलाता है।

इस query का परिणाम कुछ इस प्रकार होगाः

* परिणाम में प्रत्येक row Teacher तालिका से प्रत्येक row को Department तालिका से जोड़ देगी।

* Teacher तालिका में 14 rows हैं और Department तालिका में 9 rows हैं, इसलिए इस संयोजन में 14 x 9 = 126 rows होंगी।

* यह संयोजन Cartesian Product कहलाता है।

2. Equi Join (संबंधित शिक्षक और विभाग को जोड़ने के लिए):

अब, हम केवल शिक्षक और उनके संबंधित विभाग को जोड़ना चाहते हैं, इसके लिए हम Dept_No (Teacher तालिका में) और Dept_ID (Department तालिका में) का उपयोग करेंगे। इसे Equi Join कहा जाता है और इस प्रकार से जोड़ने के लिए query इस तरह से होगीः

```
SELECT First_Name, Last_Name, Dept_ID, Dept_Name
FROM Teacher, Department
WHERE Dept_ID = Dept_No;
```

इस query में WHERE क्लॉज़ का उपयोग किया गया है ताकि Dept_No और Dept_ID के बीच समानता को परखा जा सके, जिससे केवल वही rows जुड़ेंगी जहां शिक्षक और विभाग का संबंध हो।

उदाहरण परिणामः

First_Name	Last_Name	Dept_ID	Dept_Name
Shanaya	Batra	1	Chemistry
Megha	Khanna	4	Hindi
Neha	Gupta	7	Commerce
Vinita	Arora	9	Mathematics

स्पष्टीकरणः

* यहां, प्रत्येक शिक्षक के Dept_No के आधार पर उसे उसके संबंधित विभाग से जोड़ा गया है।

* इस प्रकार से जुड़ने से हमें केवल उन शिक्षकों के नाम और विभाग की जानकारी प्राप्त होती है जो एक निश्चित विभाग में कार्यरत हैं।

इस प्रकार, Cross Join और Equi Join का उपयोग करके हम विभिन्न तालिकाओं से डेटा जोड़ सकते हैं और केवल वांछित परिणाम प्राप्त कर सकते हैं।

हमारे पास Teacher और Department तालिकाओं में डेटा है। हम चाहते हैं कि प्रत्येक शिक्षक के नाम के साथ उनके संबंधित विभाग का नाम और नंबर भी प्रदर्शित हो। इसके लिए, हम इन तालिकाओं को जोड़ने के लिए Dept_ID (Department तालिका) और Dept_No (Teacher तालिका) का उपयोग करेंगे, ताकि हम केवल उन पंक्तियों को प्राप्त कर सकें जहां शिक्षक और विभाग का संबंध हो।

Query Code:

```
SELECT First_Name, Last_Name, Dept_ID, Dept_Name

FROM Teacher, Department

WHERE Dept_ID = Dept_No;
```

यहाँ पर Dept_ID = Dept_No join condition है। यह join condition दोनों तालिकाओं से केवल उन्हीं पंक्तियों को चुनेगी जहाँ Dept_ID और Dept_No का मान समान हो। इस query का परिणाम नीचे दिया गया है:

Result:

First_Name	Last_Name	Dept_ID	Dept_Name
Shanaya	Batra	1	Chemistry
Megha	Khanna	4	Hindi
Neha	Gupta	7	Commerce
Vinita	Arora	9	Mathematics

स्पष्टीकरण:

* यहां, Dept_ID = Dept_No का उपयोग कर तालिकाओं को जोड़ने पर केवल संबंधित शिक्षक और उनके विभाग से संबंधित जानकारी ही प्रदर्शित होती है।

* इस प्रकार की query को Equi Join भी कहा जाता है क्योंकि इसमें समानता की स्थिति का उपयोग किया गया है।

महत्वपूर्ण प्रश्न

1. MCQs (Multiple Choice Questions)

प्रश्न 1: एक डेटाबेस में डेटा को संगठित करने और संरक्षित करने के लिए किस प्रकार के सिस्टम का उपयोग किया जाता है?

A) RDBMS   B) CMS

C) HTML   D) XML

प्रश्न 2: SQL में, किसी तालिका में नई पंक्ति जोड़ने के लिए किस कमांड का उपयोग किया जाता है?

A) ADD   B) INSERT

C) MODIFY   D) UPDATE

प्रश्न 3: किसी तालिका में Primary Key का उपयोग क्यों किया जाता है?

A) डेटा को समूहित करने के लिए

B) तालिका में प्रत्येक रिकॉर्ड की पहचान करने के लिए

C) तालिका को हटाने के लिए

D) तालिका के आकार को कम करने के लिए

प्रश्न 4: SQL में "LIKE" ऑपरेटर का उपयोग किस प्रकार से किया जाता है?

A) पूरी तरह से समानता जाँच के लिए   B) पैटर्न मिलान के लिए

SCERT DELHI

C) तुलना करने के लिए

D) गणना करने के लिए

प्रश्न 5: RDBMS में दो टेबल्स को जोड़ने के लिए किस की का उपयोग किया जाता है?

A) Composite Key

B) Primary Key

C) Foreign Key

D) Unique Key

2. Fill in the Blanks (रिक्त स्थान भरें)

प्रश्न 1: RDBMS में डेटा को व्यवस्थित और संग्रहीत करने के लिए ________ संरचना का उपयोग किया जाता है।

प्रश्न 2: किसी टेबल में ________ कमांड का उपयोग कर डेटा में संशोधन किया जा सकता है।

प्रश्न 3: डेटाबेस में एक कॉलम की विशेषता जो कि NULL हो सकती है, ________ कहलाती है।

प्रश्न 4: SQL में, ________ ऑपरेटर का उपयोग पैटर्न मिलान के लिए किया जाता है।

प्रश्न 5: एक ________ डेटाबेस में दो टेबल्स के बीच संबंध बनाने के लिए प्रयोग किया जाता है।

3. True or False (सही या गलत)

प्रश्न 1: एक Primary Key एक टेबल में केवल एक ही हो सकती है।

प्रश्न 2: SQL में DELETE कमांड का उपयोग किसी टेबल को पूरी तरह से हटाने के लिए किया जाता है।

प्रश्न 3: Foreign Key का उपयोग किसी अन्य टेबल के Primary Key से संबंध बनाने के लिए किया जाता है।

प्रश्न 4: SQL में ORDER BY का उपयोग डेटा को व्यवस्थित करने के लिए किया जाता है।

प्रश्न 5: SQL में COUNT( ) फंक्शन का उपयोग किसी टेबल में NULL वैल्यू की गणना करने के लिए किया जाता है।

4. Short Answer Questions (संक्षिप्त उत्तर प्रश्न)

प्रश्न 1: RDBMS और DBMS में क्या अंतर है?

प्रश्न 2: Foreign Key की परिभाषा क्या है, और यह कैसे उपयोगी है?

प्रश्न 3: SQL में LIKE ऑपरेटर का क्या उपयोग है?

SCERT DELHI

प्रश्न 4: SQL में COUNT( ), SUM( ) और AVG( ) फंक्शंस क्या करते हैं?

प्रश्न 5: Primary Key को टेबल में कैसे लागू किया जाता है?

5. Long Answer Questions (दीर्घ उत्तर प्रश्न)

प्रश्न 1: SQL में CRUD ऑपरेशन्स (CREATE, READ, UPDATE, DELETE) को विस्तार से समझाइए।

प्रश्न 2: डेटाबेस प्रबंधन प्रणाली (DBMS) की आवश्यकता और इसके प्रमुख लाभों का विस्तार से वर्णन कीजिए। साथ ही, DBMS और RDBMS में अंतर स्पष्ट कीजिए।

प्रश्न 3: डेटाबेस मॉडल्स की विभिन्न प्रकारों की व्याख्या कीजिए, जैसे हायरार्किकल, नेटवर्क, रिलेशनल और ऑब्जेक्ट-ओरिएंटेड डेटाबेस।

प्रश्न 4: RDBMS के विभिन्न प्रकार के Constraints (जैसे कि Primary Key, Foreign Key, Unique, Check, और Default) को विस्तार से समझाइए।

उत्तरः

1. MCQs (Multiple Choice Questions)

1. A) RDBMS
2. B) INSERT
3. B) तालिका में प्रत्येक रिकॉर्ड की पहचान करने के लिए
4. B) पैटर्न मिलान के लिए
5. C) Foreign Key

2. Fill in the Blanks (रिक्त स्थान भरें)

1. तालिका (Table)
2. UPDATE
3. Nullable
4. LIKE
5. Foreign Key

SCERT DELHI

वेब आधारित एप्लिकेशन का संचालन

इकाई 2

सीखने के प्रतिफल

विद्यार्थी-

* वेब आधारित एप्लिकेशन के बारे में समझते है।
* ऑनलाइन आरक्षण प्रणाली समझ पाते हैं।
* बिल भुगतान के बारे में समझ कर पाते हैं।
* ई-गवर्नेंस के बारे में और इसका इस्तेमाल कैसे करे कर पाते हैं।
* परियोजना प्रबंधन के बारे में समझ कर पाते हैं।
* ऑनलाइन कोर्स, ट्यूटोरियल्स के बारे में जान पाते हैं।

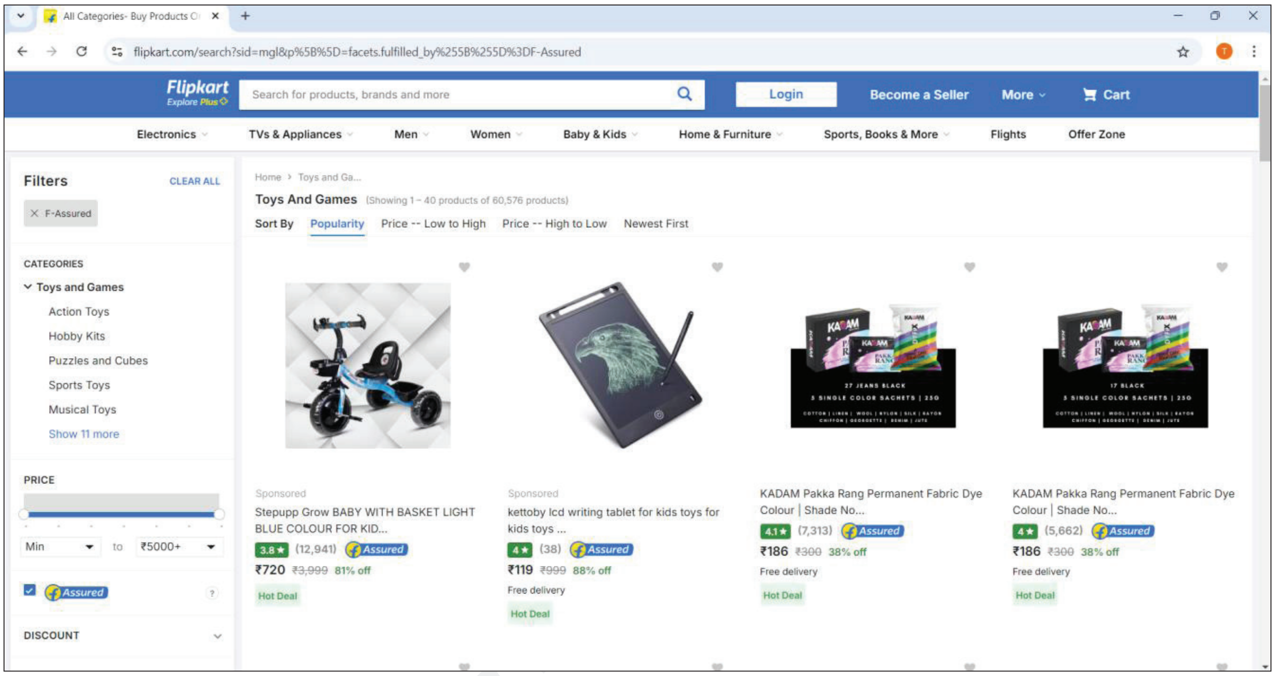
परिचय (Introduction)

चलिए उदाहरण से समझते है आज के इस इंटरनेट के युग में बहुत से चीजों के लिए हम कई प्रकार की एप्लीकेशन/ वेबसाइट पर निर्भर करते है। आपको एक जूते खरीदने है लेकिन जहां आप रहते है वह से बाज़ार बहुत ज्यादा दूर है आप इंटरनेट का इस्तेमाल बहुत अच्छे से करना जानते है। इस परिस्थिति में आप क्या करेगे? आपके दिमाग में इंटरनेट से जूते खरीदने का विचार ज़रूर आया होगा। आप वेब ब्राउज़र पर अमेज़न (AMAZON), फ्लिप्कार्ट (FLIPKART) या कोई और वेबसाइट पर जा कर उससे जुत्ते आर्डर करेगे। इन सब वेबसाइट को वेब आधारित एप्लीकेशन बोला जाता है इन एप्लीकेशन को हम अपने कंप्यूटर में इनस्टॉल न करके सीधे वेब ब्राउज़र के ज़रिये इस्तेमाल करते है। अब हम इन सब को विस्तृत रूप से अध्ययन करेगे।

2.1 वेब आधारित एप्लिकेशन का संचालन (Operating Web Based Applications) क्या है?

वेब आधारित एप्लिकेशन, जिन्हें अक्सर वेब ऐप्स कहा जाता है, ऐसे सॉफ्टवेयर प्रोग्राम होते हैं जिन्हें किसी भी वेब ब्राउज़र (जैसे कि Chrome, Firefox, Safari) के माध्यम से इंटरनेट पर एक्सेस किया जा सकता है। आपको इन ऐप्स को अपने कंप्यूटर या मोबाइल डिवाइस पर इंस्टॉल करने की आवश्यकता नहीं होती है। आजकल, टिकटों के आरक्षण , बुकिंग, ई-गवर्नेंस, ऑनलाइन भुगतान,

ऑनलाइन शॉपिंग और ऑनलाइन ट्यूटोरियल आदि के लिए वेब-आधारित एप्लिकेशन का उपयोग किया जाता है। दुनिया भर में वेब-आधारित एप्लिकेशन 24x7 उपलब्धता की सुविधा प्रदान करता है। इन् का इस्तेमाल घर, कार्यालय, कार आदि में आराम से इंटरनेट के माध्यम से सुविधाएं सेकर सकते हैं।



चित्र 2.1 वेब आधारित एप्लीकेशन का उदहारण

वेब ऐप्स के प्रकार

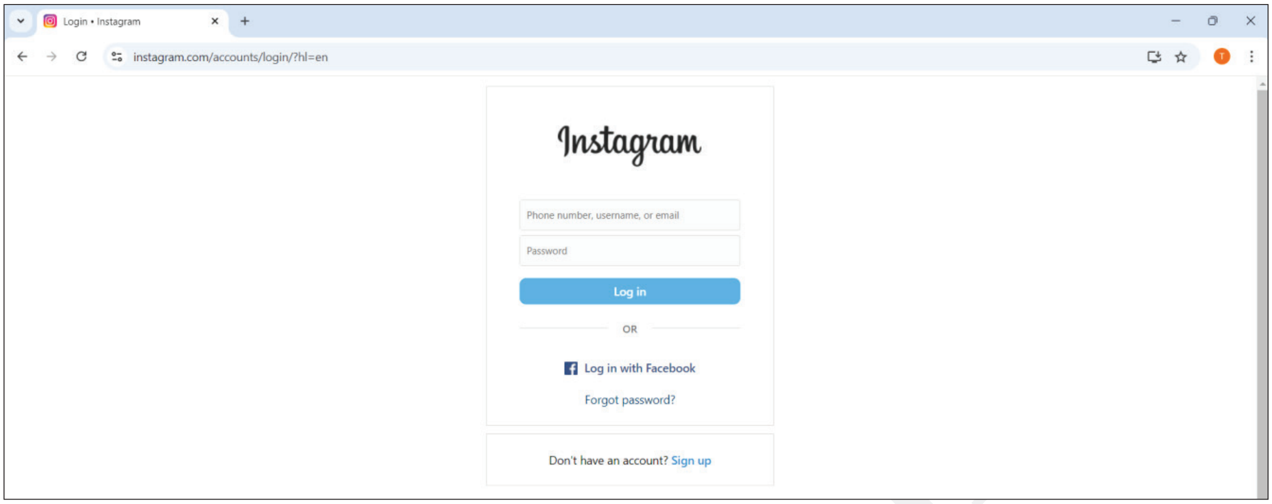
* सामाजिक नेटवर्कः फेसबुक, ट्विटर
* ई-कॉमर्सः अमेज़न, फ्लिपकार्ट
* क्लाउड स्टोरेजः गूगल ड्राइव (Google Drive), ड्राप बॉक्स (Dropbox)
* ऑनलाइन बैंकिंगः नेट बैंकिंग पोर्टल्स
* सॉफ्टवेयर-एज़-ए-सर्विस (SaaS): गूगल डॉक्स(Google Docs), सेल्स फॉर्स (Salesforce)

2.1.1 वेब आधारित एप्लिकेशन के प्रकार (Categories of Web-Based Applications)

वेब आधारित एप्लिकेशन की दुनिया विविध और लगातार विकसित हो रही है। इन एप्लिकेशन को उनके कार्यों, उपयोगकर्ताओं और उद्देश्यों के आधार पर विभिन्न श्रेणियों में वर्गीकृत किया जा सकता है। आइए कुछ प्रमुख श्रेणियों पर एक नज़र डालते हैं:

SCERT

1. सामाजिक नेटवर्किंग एप्लिकेशन (Social Networking Applications)

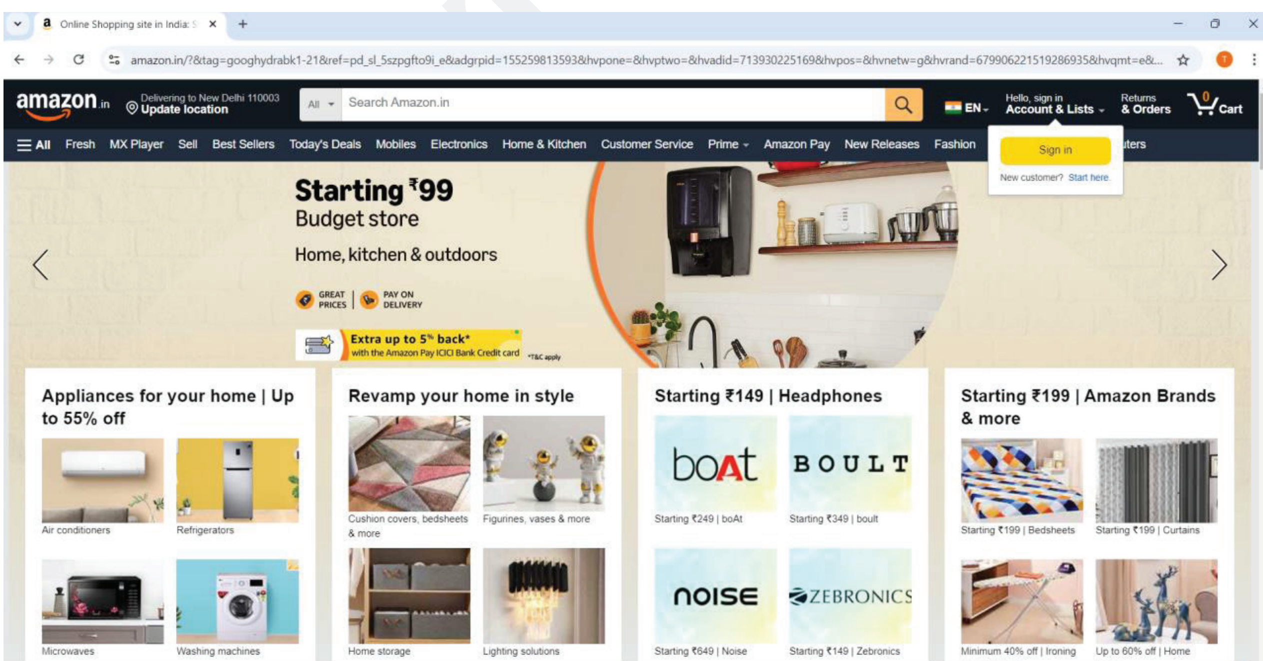


चित्र 2.2 सामाजिक नेटवर्किंग का उदहारण

ये एप्लिकेशन लोगों को एक-दूसरे से जुड़ने, संवाद करने और जानकारी साझा करने की अनुमति देते हैं।

* उदाहरणः फेसबुक (Facebook), ट्विटर (Twitter), इंस्टाग्राम (Instagram)
* विशेषताएंः प्रोफ़ाइल बनाना, दोस्तों को जोड़ना, पोस्ट साझा करना, संदेश भेजना, समूह बनाना।

2. ई-कॉमर्स एप्लिकेशन (E-commerce Applications)

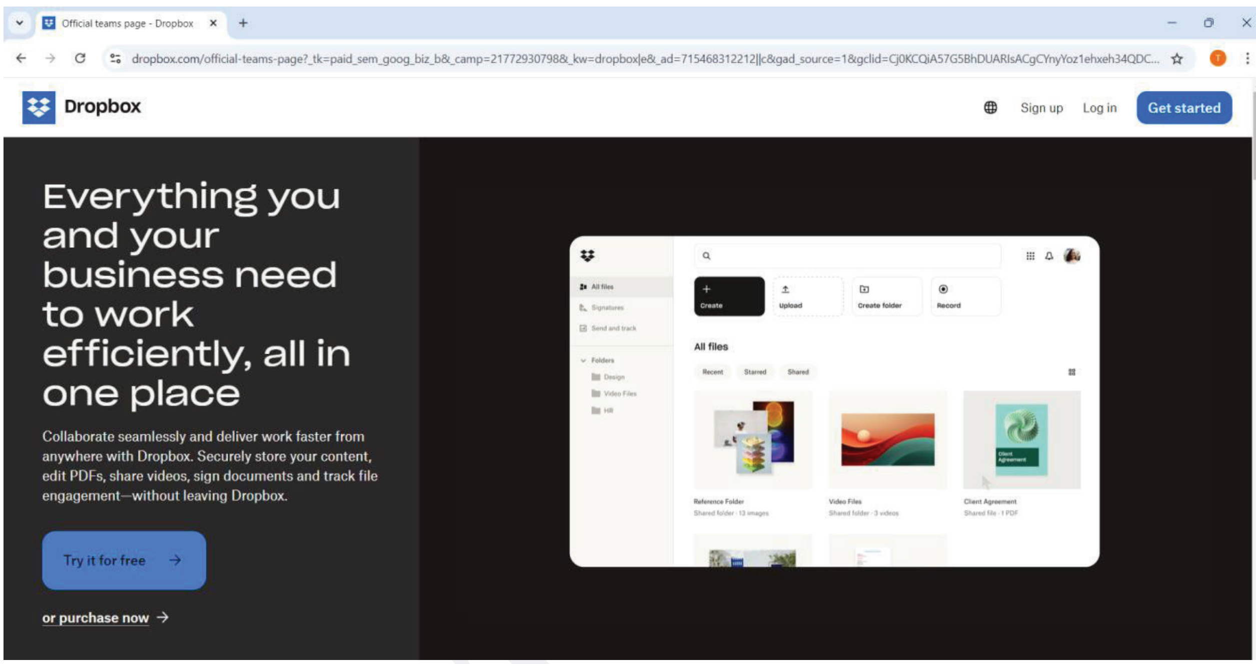


चित्र 2.3 ई-कॉमर्स एप्लिकेशन का उदहारण

ये एप्लिकेशन ऑनलाइन उत्पादों और सेवाओं की खरीद और बिक्री की सुविधा प्रदान करते हैं।

* उदाहरणः अमेज़न (Amazon), फ्लिपकार्ट (Flipkart), अलीबाबा (Alibaba)
* विशेषताएंः उत्पाद ब्राउज़ करना, कार्ट में जोड़ना, भुगतान करना, ऑर्डर ट्रैकिंग

3. क्लाउड स्टोरेज एप्लिकेशन (Cloud Storage Applications)



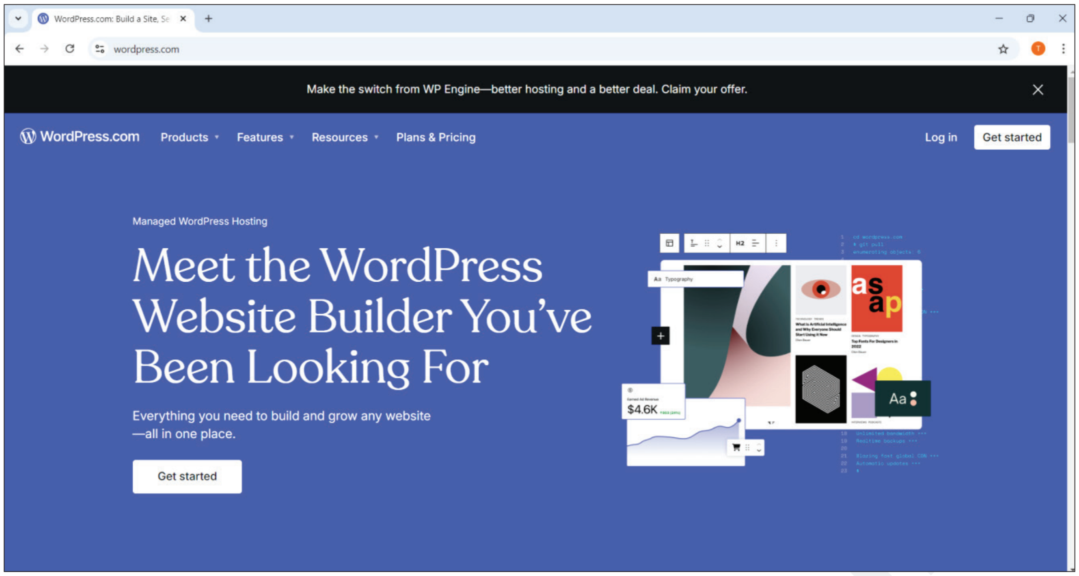
चित्र 2.4 क्लाउड स्टोरेज एप्लिकेशन का उदहारण

ये एप्लिकेशन डेटा को इंटरनेट पर स्टोर करने और एक्सेस करने की अनुमति देते हैं।

* उदाहरणः गूगल ड्राइव (Google Drive), ड्रापबॉक्स (Dropbox), वनड्राइव (OneDrive)
* विशेषताएंः फ़ाइल अपलोड करना, फ़ाइल साझा करना, फ़ोल्डर बनाना

4. ब्लॉगिंग प्लेटफॉर्म (Blogging Platforms)

ये ये एप्लिकेशन उपयोगकर्ताओं को ब्लॉग पोस्ट लिखने और प्रकाशित करने की अनुमति देते हैं।



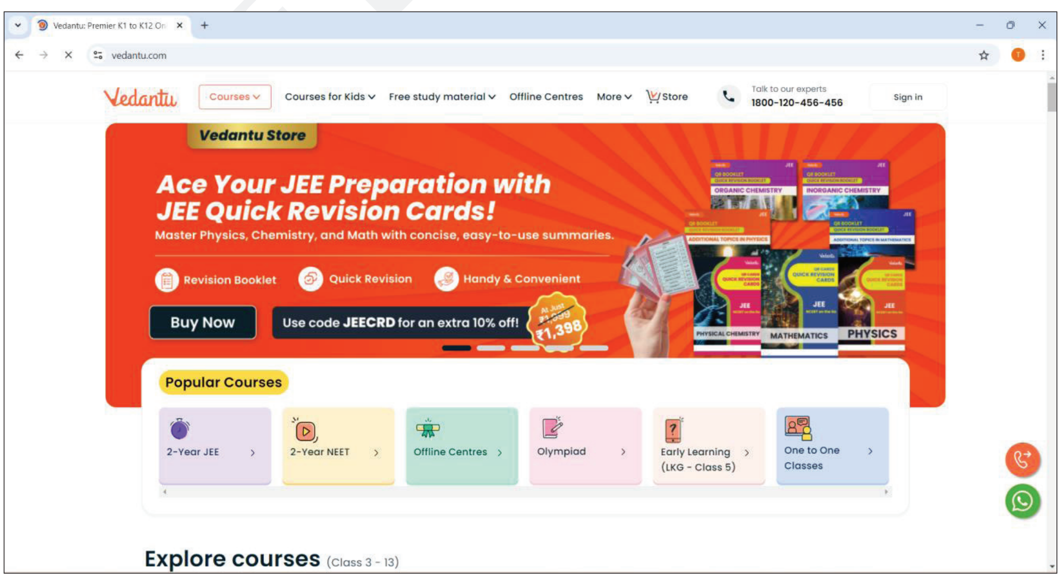
चित्र 2.5 ब्लॉगिंग प्लेटफॉर्म का उदाहरण

* उदाहरणः वर्डप्रेस (WordPress), ब्लॉगर (Bolgger)
* विशेषताएंः पोस्ट बनाना, श्रेणियां बनाना, टैग्स जोड़ना, कमेंट्स का प्रबंधन करना

5. ऑनलाइन शिक्षा प्लेटफॉर्म (Online Education Platforms)

ये एप्लिकेशन ऑनलाइन पाठ्यक्रम और प्रशिक्षण प्रदान करते हैं।

* उदाहरणः ब्ज्युस (Byju's), वेदान्तु (vedantu), माय सीबीएसई गाइड (mycbseguide)
* विशेषताएंः वीडियो लेक्चर, असाइनमेंट, परीक्षा, चर्चा फोरम



चित्र 2.6 ऑनलाइन शिक्षा प्लेटफॉर्म का उदाहरण

अन्य श्रेणियां:

* बैंकिंग एप्लिकेशनः ऑनलाइन बैंकिंग सेवाएं प्रदान करते हैं।
* यात्रा एप्लिकेशनः यात्रा बुकिंग और प्रबंधन की सुविधा प्रदान करते हैं।
* स्वास्थ्य और फिटनेस एप्लिकेशनः स्वास्थ्य और फिटनेस ट्रैकिंग सुविधाएं प्रदान करते हैं।
* न्यूज़ एप्लिकेशनः नवीनतम समाचार और जानकारी प्रदान करते हैं।

वेब आधारित एप्लिकेशन के फायदे

* आसान एक्सेसः उपयोगकर्ता इंटरनेट से किसी भी डिवाइस पर एप्लिकेशन का उपयोग कर सकते हैं।
* केंद्रित डाटाः डेटा एक केंद्रीकृत सर्वर पर स्टोर होता है, जिससे डाटा बैकअप, सुरक्षा और अपडेट्स आसान होते हैं।
* कस्टम अपडेट्सः एप्लिकेशन के अपडेट्स सभी उपयोगकर्ताओं के लिए तुरंत उपलब्ध हो जाते हैं, बिना किसी डाउनलोड की आवश्यकता के।
* कम लागतः क्योंकि एप्लिकेशन को इंस्टॉल करने की आवश्यकता नहीं होती, इसके विकास और रखरखाव की लागत कम हो सकती है।
* सुरक्षाः डेटा सुरक्षा के लिए सर्वर-साइड सुरक्षा उपाय लागू किए जा सकते हैं, जैसे SSL, डेटा एन्क्रिप्शन, और 2FA (Two Factor Authentication)।

2.2 ऑनलाइन आरक्षण प्रणाली (Online Reservation System)

ऑनलाइन आरक्षण प्रणाली (Online Reservation System) एक डिजिटल प्लेटफ़ॉर्म है जिसका उद्देश्य उपयोगकर्ताओं को इंटरनेट के माध्यम से सेवाओं का आरक्षण करने की सुविधा प्रदान करना है। यह प्रणाली उपयोगकर्ताओं को टिकट, कमरे, उत्पाद, या किसी अन्य प्रकार की सेवा को ऑनलाइन बुक करने का मौका देती है, और यह प्रक्रिया त्वरित, सुरक्षित और सुविधाजनक होती है।

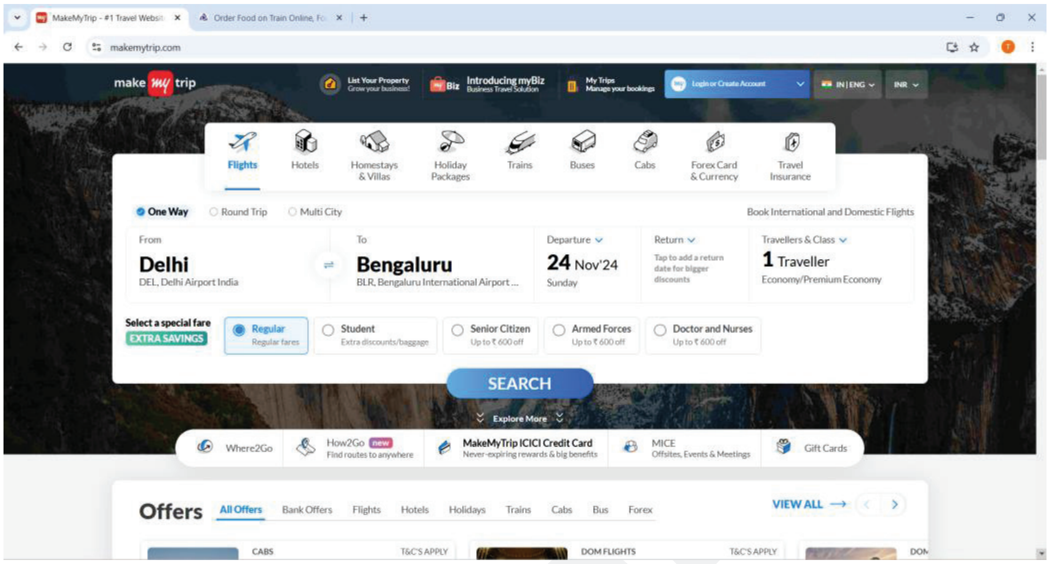
ऑनलाइन आरक्षण प्रणाली को कई प्रकार के उपयोग के मामलों के लिए डिज़ाइन किया जा सकता है, जैसे कि यात्रा, होटल बुकिंग, रेस्तरां आरक्षण, इवेंट्स बुकिंग, चिकित्सा सेवाओं के लिए अपॉइंटमेंट्स, आदि।

2.2.1 ऑनलाइन आरक्षण प्रणाली के प्रकार

1. **होटल और यात्रा आरक्षण (Hotel and Travel reservation) :**

SCERT DELHI

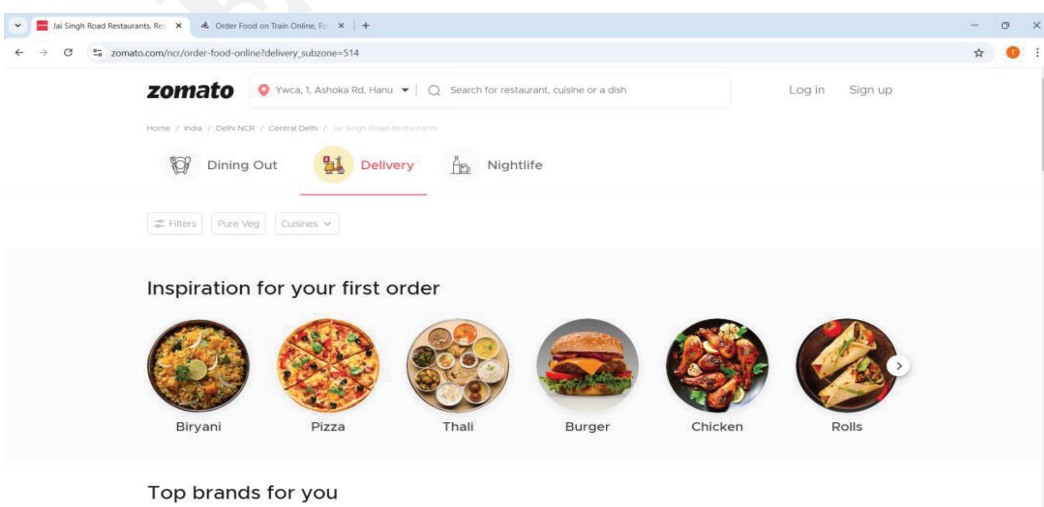
* इन प्रणालियों का उद्देश्य उपयोगकर्ताओं को होटल, फ्लाइट, ट्रेन, बस, और क्रूज जैसी यात्रा सेवाओं का आरक्षण करना है। कुछ प्रसिद्ध उदाहरण हैं:
* बुकिंग.कॉम (Booking.com),एयरबीएनबी(Airbnb),मेकमायट्रिप(Makemytrip)



चित्र 2.7 फ्लाइट आरक्षण का उदहारण

2. रेस्तरां आरक्षण (Restaurant reservation):

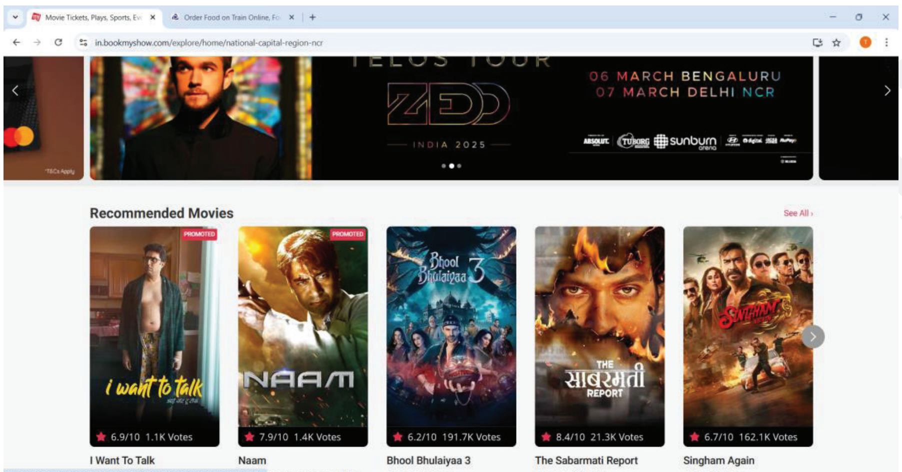
- ये प्रणालियाँ उपयोगकर्ताओं को रेस्तरां में टेबल आरक्षित करने की सुविधा प्रदान करती हैं। उदाहरण:
- ज़ोमाटो(Zomato), डाआइनआउट (DineOut)



चित्र 2.8 रेस्तरां आरक्षण का उदहारण

3. इवेंट बुकिंग (Event booking):

- इवेंट जैसे कॉन्सट्र्स, मूवी टिकट्स, थियेटर शोज, और खेल आयोजनों के लिए आरक्षण प्रणाली। उदाहरण:
- बुकमायशो (BookMyShow), टिकेटमास्टर (Ticketmaster)



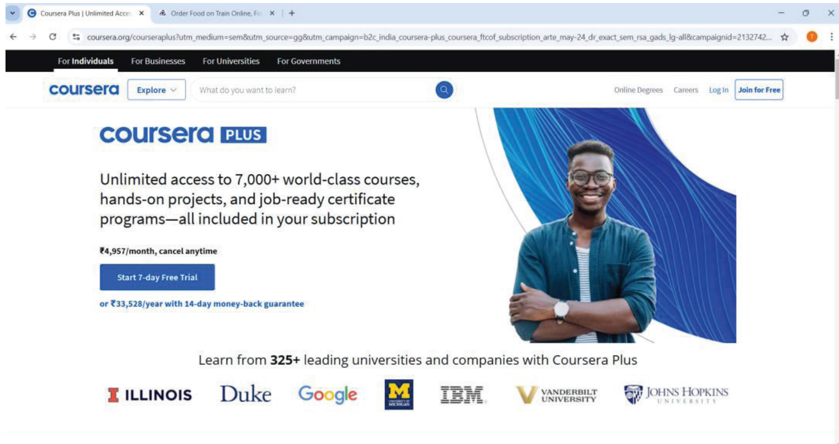
चित्र 2.9 मूवी बूकिंग का उदहारण

4. चिकित्सा आरक्षण (Medical reservation):

- अस्पतालों और क्लीनिकों में डॉक्टर की अपॉइंटमेंट बुक करने के लिए ऑनलाइन आरक्षण प्रणाली। उदाहरण:
- परक्टो(Practo), 1ऍमजी(1mg), हेअल्थिफायेमी (HealthifyMe)

5. ऑनलाइन क्लास और कोर्स आरक्षण (Online class and course reservation):

- शैक्षिक संस्थानों या ऑनलाइन कोर्स प्लेटफार्मों में पाठ्यक्रमों के लिए पंजीकरण और आरक्षण। उदाहरण:



चित्र 2.10 ऑनलाइन क्लास का उदहारण

- कोउर्सेरा (Coursera), उदेमी (Udemy),स्किलशेयर (Skillshare)

गतिविधि 2.1

सभी बच्चे किसी भी वेबसाइट का इस्तेमाल करके होटल या फिर यात्रा आरक्षण करें और इसपर कक्षा में व्यापक रूप से चर्चा करें।

2.2.2 ऑनलाइन आरक्षण प्रणाली के फायदे

* **सुविधाजनक:**
  - उपयोगकर्ता कहीं से भी, किसी भी समय, अपने कंप्यूटर या स्मार्टफोन के माध्यम से आरक्षण कर सकते हैं, जिससे समय और स्थान की सीमाएँ समाप्त हो जाती हैं।
* **तत्काल पुष्टि:**
  - ऑनलाइन आरक्षण प्रणाली में तुरंत पुष्टि प्राप्त होती है, जिससे उपयोगकर्ता को चिंता नहीं रहती कि उनका आरक्षण हुआ या नहीं।
* **समय की बचत:**
  - ऑनलाइन आरक्षण करने से उपयोगकर्ता को लम्बी कतारों या फोन कॉल्स की आवश्यकता नहीं होती, जिससे उनका समय बचता है।
* **लागत प्रभावी:**
  - कई ऑनलाइन आरक्षण प्रणालियाँ विशेष छूट, ऑफ़र, और प्रमोशनल कूपन प्रदान करती हैं, जिससे उपयोगकर्ता को सस्ती दरों पर सेवाएँ मिल सकती हैं।

* **24/7 समर्थन:**

➤ ऑनलाइन आरक्षण प्रणाली में उपयोगकर्ता को 24/7 समर्थन मिलता है, चाहे वे कहीं भी हों, जिससे किसी भी समस्या का समाधान तुरंत किया जा सकता है।

2.2.3 ऑनलाइन रेल टिकट बुकिंग प्रक्रिया (Steps for Booking Train Ticket Online)

आइए अब इंडियन रेलवे कैटरिंग एवम टूरिज्म कारपोरेशन (Indian Railway Catering and Tourism Corporation (IRCTC)) के माध्यम से ऑनलाइन रेल टिकट बुक करने की प्रक्रिया का विश्लेषण करें:

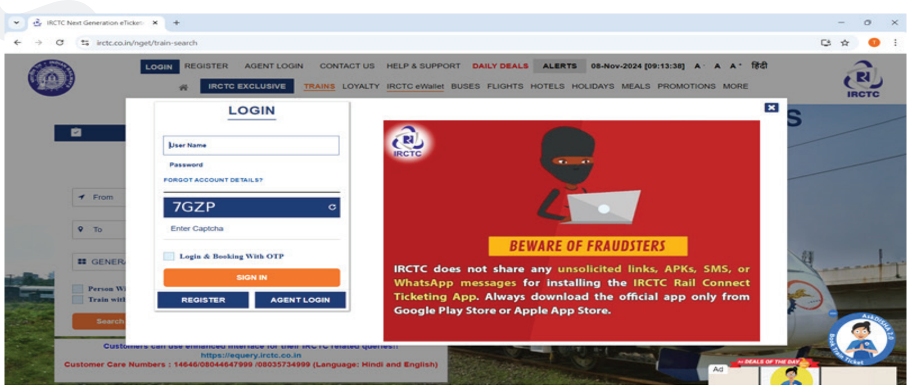
चरण 1: IRCTC की वेबसाइट पर जाएं

उपयोगकर्ता सबसे पहले IRCTC की आधिकारिक वेबसाइट (www.irctc.co.in) या IRCTC मोबाइल ऐप खोलते हैं। अगर उनका पहले से अकाउंट नहीं है, तो उन्हें पहले रजिस्टर करना होगा।

चित्र 2.11 IRCTC की वेबसाइट

चरण 2: उपयोगकर्ता लॉगिन करें

लॉगिन पेज पर उपयोगकर्ता को अपना यूजर आईडी और पासवर्ड डालकर अकाउंट में लॉगिन करना होता है। अगर उपयोगकर्ता के पास अकाउंट नहीं है, तो वह नए अकाउंट के लिए रजिस्टर कर सकता है।

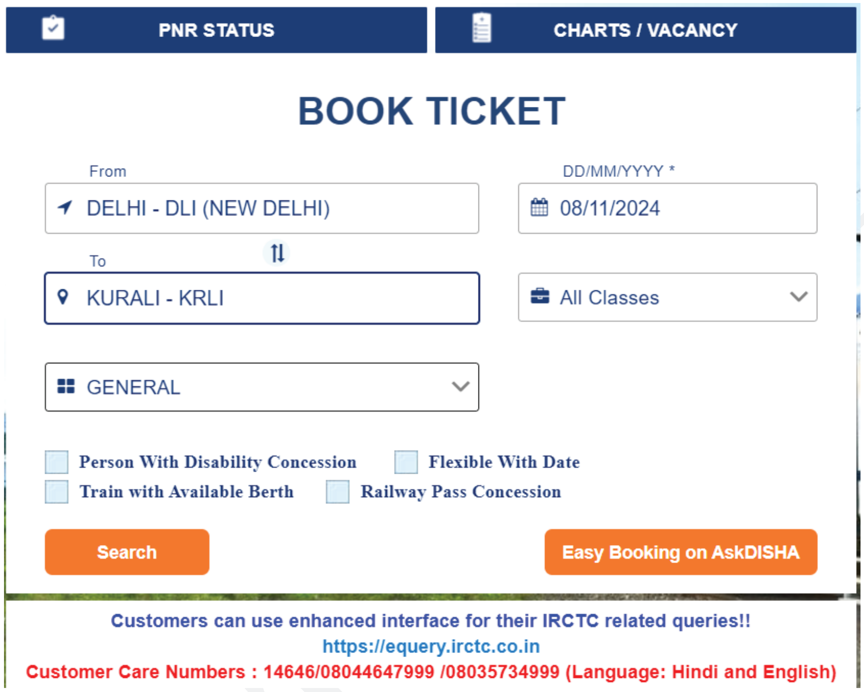


चित्र 2.12 IRCTC की वेबसाइट पर लॉग इन

SCERT DELHI

चरण 3: ट्रेन और मार्ग का चयन करें

लॉगिन करने के बाद, उपयोगकर्ता को ट्रेन बुक करने के लिए प्रस्थान स्थान (Source Station) और गंतव्य स्थान (Destination Station) को चुनना होता है। इसके बाद, उन्हें यात्रा की तारीख और ट्रेन का प्रकार (जैसे एक्सप्रेस, सुपरफास्ट, मेल, पैसेंजर आदि) का चयन करना होता है।

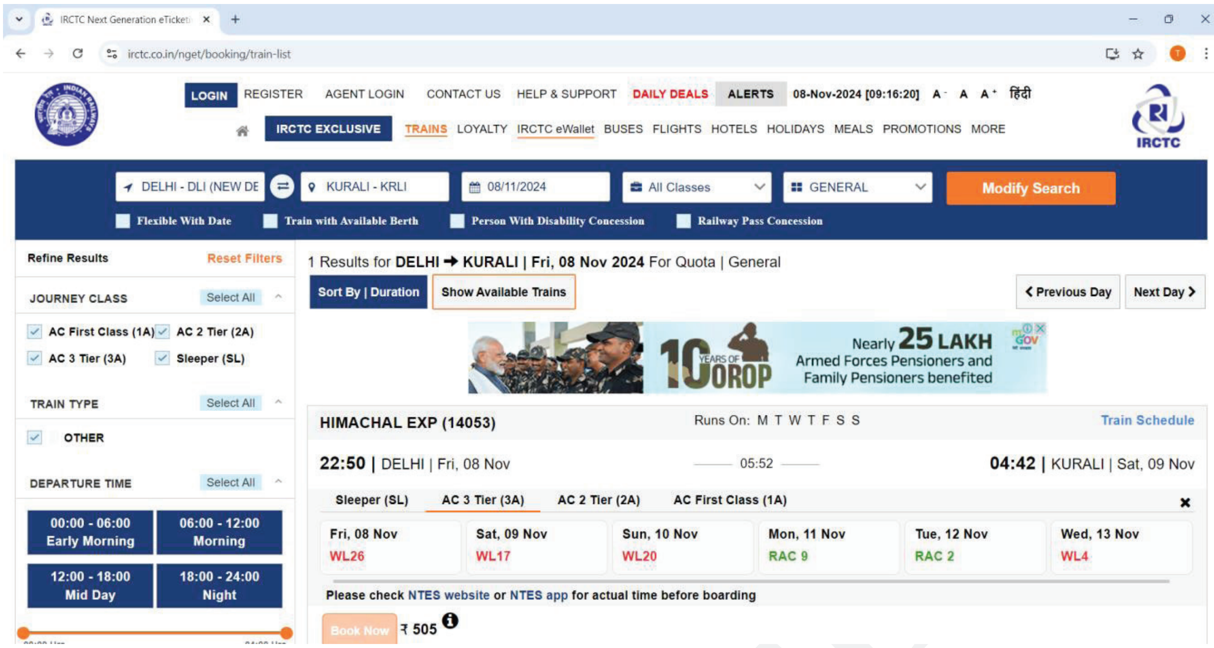


चित्र 2.13 IRCTC की वेबसाइट पर ट्रेन मार्ग का चयन

लॉगिन करने के बाद, उपयोगकर्ता को ट्रेन बुक करने के लिए प्रस्थान स्थान (Source Station) और गंतव्य स्थान (Destination Station) को चुनना होता है। इसके बाद, उन्हें यात्रा की तारीख और ट्रेन का प्रकार (जैसे एक्सप्रेस, सुपरफास्ट, मेल, पैसेंजर आदि) का चयन करना होता है।

चरण 4: ट्रेन का चयन करें और सीट की उपलब्धता देखें

आवश्यक स्टेशन और यात्रा की तारीख डालने के बाद, उपयोगकर्ता को उपलब्ध ट्रेनों की सूची दिखती है। इस सूची में ट्रेनों की श्रेणियाँ, जैसे AC, Sleeper, First Class आदि, और उनके साथ सीट की उपलब्धता (Available, Waiting, Reserved) दिखाई जाती है।



चित्र 2.14 IRCTC की वेबसाइट पर ट्रेन का चयन

चरण 5: सीट और क्लास का चयन करें

उपयोगकर्ता अपनी पसंदीदा ट्रेन और श्रेणी का चयन करता है, और फिर उपलब्ध सीटों को देखता है। वह अपनी पसंदीदा सीट का चयन करता है और आगे बढ़ता है।

चरण 6: यात्री विवरण भरें

अब उपयोगकर्ता को अपनी व्यक्तिगत जानकारी भरनी होती है, जैसे:

* यात्री का नाम
* आयु
* लिंग
* संपर्क नंबर
* पहचान प्रमाण (जैसे आधार कार्ड, पैन कार्ड आदि)

चरण 7: भुगतान करें

सभी जानकारी भरने के बाद, उपयोगकर्ता को भुगतान करना होता है। IRCTC विभिन्न भुगतान विकल्प प्रदान करता है, जैसे:

* क्रेडिट कार्ड/डेबिट कार्ड
* नेट बैंकिंग
* UPI (Unified Payment Interface)
* डिजिटल वॉलेट्स (जैसे Paytm, Google Pay, PhonePe)

चरण 8: टिकट की पुष्टि और प्राप्ति

भुगतान पूरा होने के बाद, उपयोगकर्ता को पुष्टिकरण संदेश प्राप्त होता है। इसके बाद, इलेक्ट्रॉनिक टिकट (e-ticket) उत्पन्न होता है, जिसमें यात्रा की सभी जानकारी होती है, जैसेः

* यात्रा की तारीख और समय
* ट्रेन नंबर और नाम
* यात्री का नाम और सीट नंबर
* PNR (Passenger Name Record) संख्या

गतिविधि 2.2

IRCTC की वेबसाइट का इस्तेमाल करके अपने और अपने परिवार के सभी सदस्यों के लिए अयोध्या जाने की टिकट बुक करे।

यह इलेक्ट्रॉनिक टिकट (e-ticket) उपयोगकर्ता को अपने स्मार्टफोन या कंप्यूटर पर डाउनलोड या प्रिंट करने के लिए मिलता है, और इसे रेलवे स्टेशन पर यात्रा के दौरान दिखाना होता है।

1. URL चेक करेंः हमेशा यह सुनिश्चित करें कि जिस वेबसाइट पर आप लेन-देन कर रहे हैं, उसका URL "https://" से शुरू होता है, न कि सिर्फ "http://"। "s" का मतलब है "secure" (सुरक्षित)।

2. **मजबूत पासवर्ड का उपयोग करें (Use Strong Passwords)**

   * कमजोर पासवर्ड से बचेंः सरल पासवर्ड जैसे "123456" या "password" का उपयोग न करें। यह आसानी से अनुमान लगाए जा सकते हैं।

3. **सार्वजनिक Wi-Fi का उपयोग न करें (Avoid Using Public Wi-Fi)**

   * सार्वजनिक Wi-Fi नेटवर्क पर संवेदनशील जानकारी साझा न करें, जैसे कि बैंक खाता विवरण,

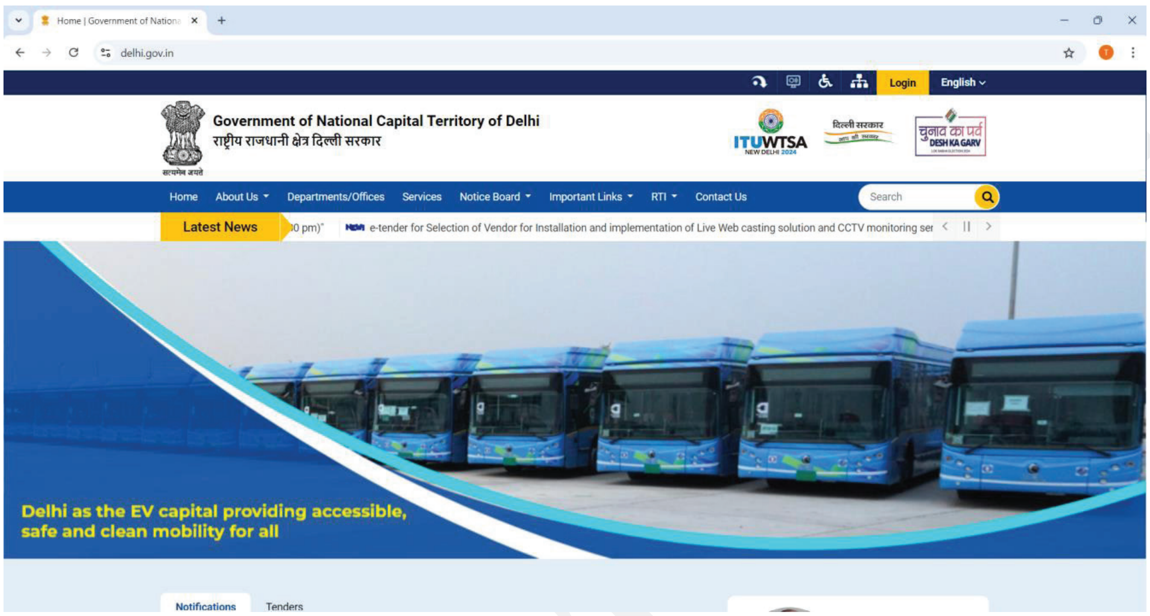
SCERT DELHI

पासवर्ड, या क्रेडिट कार्ड जानकारी।

4. **स्मार्टफोन और कंप्यूटर को सुरक्षित रखें (Secure Your Devices)**

* एंटीवायरस सॉफ़्टवेयर का उपयोग करें: अपने कंप्यूटर और स्मार्टफोन में एक अच्छा एंटीवायरस और ऐंटी-मालवेयर सॉफ़्टवेयर इंस्टॉल करें।

2.3 ई-गवर्नेंस (E-Governance) क्या है?



चित्र 2.15 ई-गवर्नेंस की वेबसाइट

ई-गवर्नेंस का मतलब है सरकार द्वारा सूचना और संचार प्रौद्योगिकी (Information and Communication Technology - ICT) का उपयोग करके नागरिकों, व्यवसायों और अन्य सरकारी एजेंसियों के साथ इंटरैक्ट(interact) करना और सेवाएं प्रदान करना। यह एक ऐसा तरीका है जिसमें सरकार डिजिटल माध्यमों से अधिक पारदर्शी, कुशल और जवाबदेह बन सकती है।

ई-गवर्नेंस (E-Governance) एक प्रक्रिया है, जिसके तहत सरकारी सेवाओं, योजनाओं और प्रक्रियाओं को सूचना प्रौद्योगिकी (Information Technology) के माध्यम से नागरिकों तक पहुंचाया जाता है। इसका उद्देश्य सरकारी कार्यों को अधिक पारदर्शी, सक्षम और त्वरित बनाना है। इसमें इंटरनेट, मोबाइल एप्लिकेशन, डिजिटल प्लेटफॉर्म्स, और अन्य तकनीकी उपकरणों का उपयोग किया जाता है ताकि सरकार अपने नागरिकों को बेहतर सेवाएं प्रदान कर सके।

भारत में, ई-गवर्नेंस के लिए मुख्य जोर NICNET 1987 - राष्ट्रीय उपग्रह-आधारित कंप्यूटर नेटवर्क की शुरूआत द्वारा प्रदान किया गया था। इसके बाद डिस्ट्रिक्ट इनफार्मेशन सिस्टम ऑफ़ दी नेशनल इन्फार्मेटिक्स

सेंटर (District Information System of the National Informatics Centre (DISNIC)) भारत सरकार की एक प्रमुख ई-गवर्नेंस पहल है, जिसे राष्ट्रीय सूचना विज्ञान केंद्र (NIC) द्वारा विकसित और लागू किया गया है। इसका उद्देश्य स्थानीय स्तर पर सरकारी डेटा और सूचना के आदान-प्रदान को सुदृढ़ करना है, जिससे स्थानीय प्रशासन और नागरिकों के बीच सूचना का प्रवाह सरल और सुलभ हो सके। राष्ट्रीय ई-गवर्नेंस योजना (NeGP) का गठन - 2006 राष्ट्रीय ई-गवर्नेंस योजना नेशनल इ-गवर्नेंस प्लान (National e-Governance Plan (NeGP)) भारत सरकार की एक महत्वपूर्ण पहल है, जिसे इलेक्ट्रॉनिक्स और सूचना प्रौद्योगिकी मंत्रालय डिपार्टमेंट ऑफ़ इलेक्ट्रॉनिक्स एंड इनफार्मेशन टेक्नोलॉजी (Department of Electronics and Information Technology (DEITY)) और प्रशासनिक सुधार और लोक शिकायत विभाग (DAR&PG) द्वारा 2006 में गठित किया गया था। इसका उद्देश्य ई-गवर्नेंस के माध्यम से सरकारी सेवाओं का डिजिटलीकरण करना और नागरिकों तक इन सेवाओं को आसानी से पहुंचाना था।

2.3.1 ई-गवर्नेंस साइटें

2.3.1.1 केंद्रीय सरकार पोर्टल:

* **भारत का राष्ट्रीय पोर्टल(The National Portal of India): https://www.india.gov.in/**

चित्र 2.16 भारत का राष्ट्रीय पोर्टल की वेबसाइट

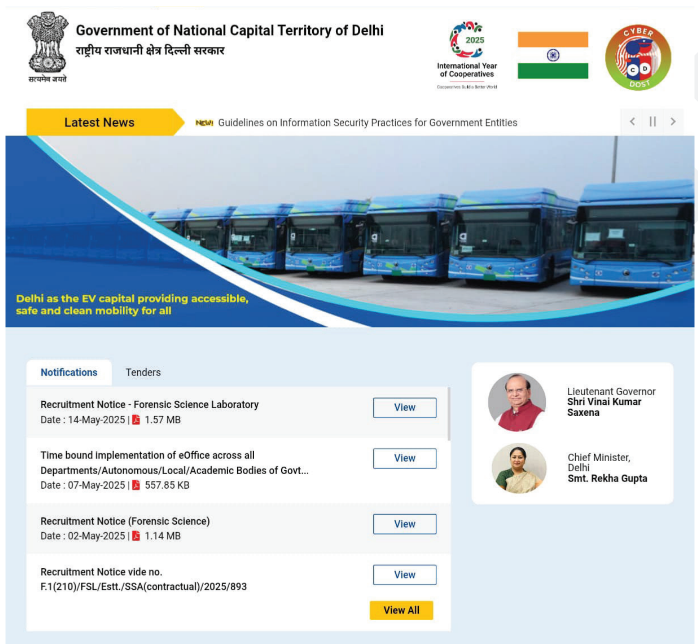
- विभिन्न सरकारी योजनाओं, नीतियों और सेवाओं के बारे में जानकारी प्रदान करता है।
- नागरिकों को सरकारी जानकारी और सेवाओं तक पहुंचने के लिए एक एकीकृत मंच प्रदान करता है।

* **MyGov: https://www.mygov.in/**

- 1. नागरिकों की भागीदारी के लिए एक मंच, नागरिकों को शासन में भाग लेने और प्रतिक्रिया प्रदान करने की अनुमति देता है।
- 2. शिकायत निवारण, सर्वेक्षण और चर्चा जैसी विभिन्न सेवाएं प्रदान करता है।

2.3.1.2 राज्य सरकार पोर्टल

दिल्ली सरकार पोर्टलः https://delhi.gov.in/



चित्र 2.17 दिल्ली सरकार की वेबसाइट

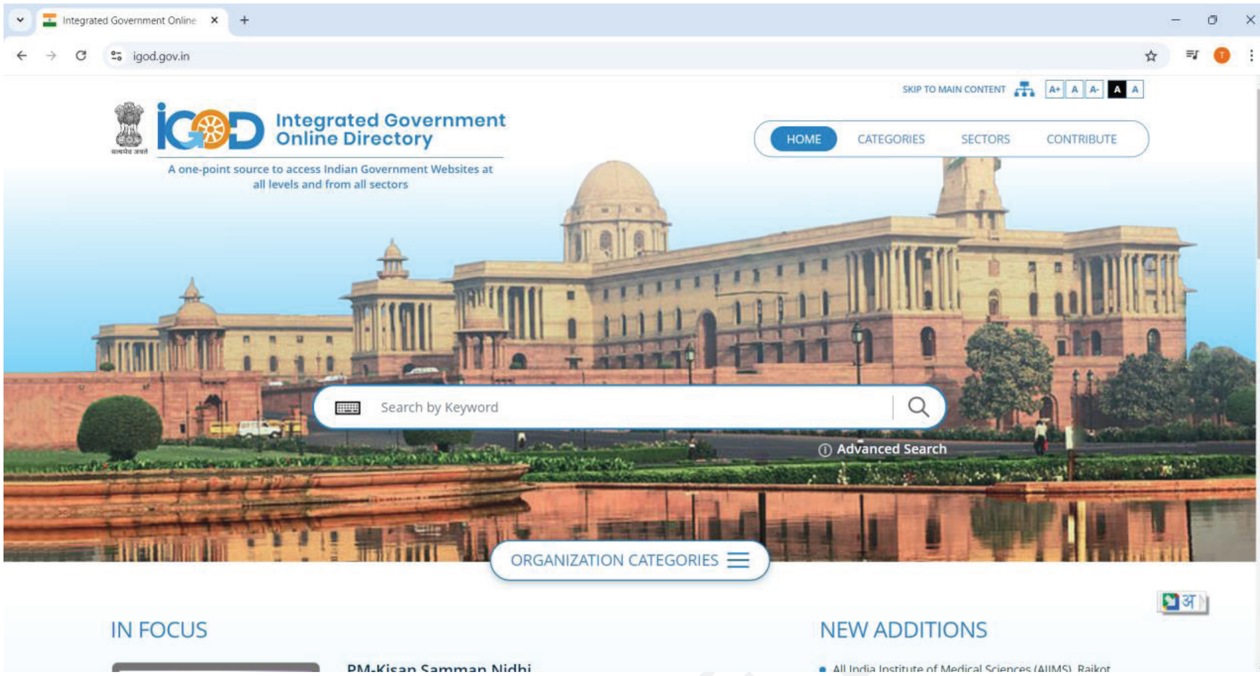
भारत की कुछ प्रमुख ई-गवर्नेंस साइटें इस प्रकार हैं:

1. **भारत का राष्ट्रीय पोर्टल (The National Portal of India) (india.gov.in)**

राष्ट्रीय पोर्टल ऑफ इंडिया भारत सरकार का प्रमुख पोर्टल है, जो नागरिकों को सभी सरकारी सेवाओं, योजनाओं, नीतियों, और दस्तावेजों तक ऑनलाइन पहुंच प्रदान करता है। यह पोर्टल विभिन्न सरकारी विभागों और मंत्रालयों की वेबसाइट्स का लिंक भी प्रदान करता है।

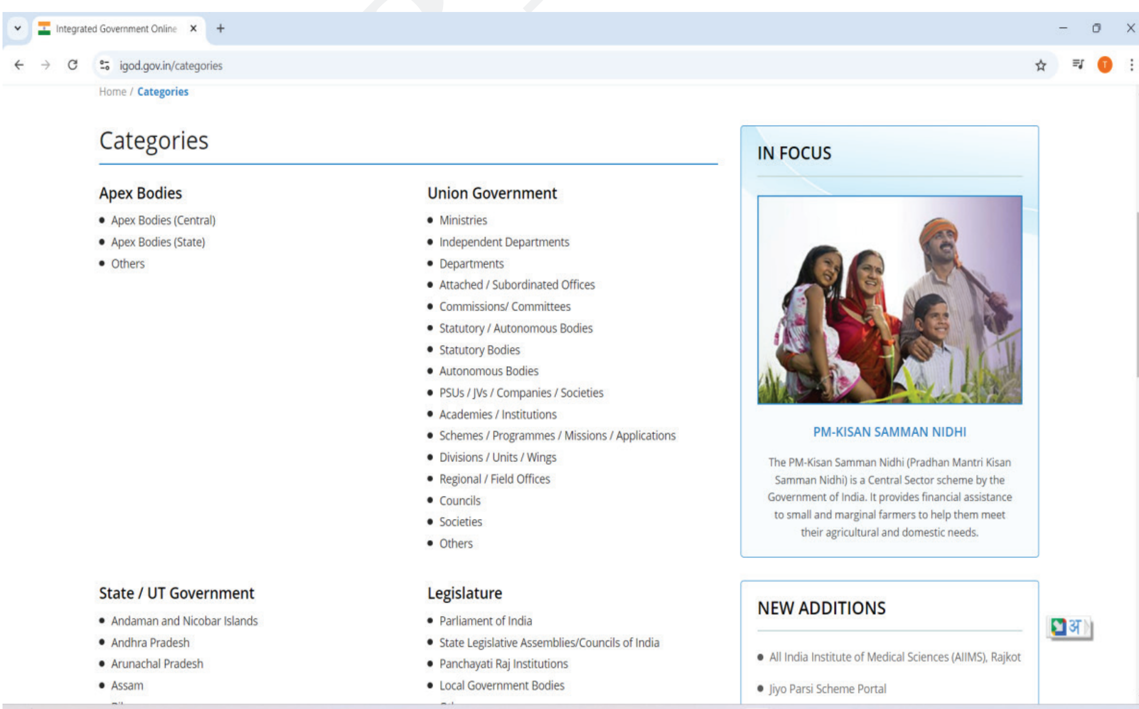
मुख्य उद्देश्यः

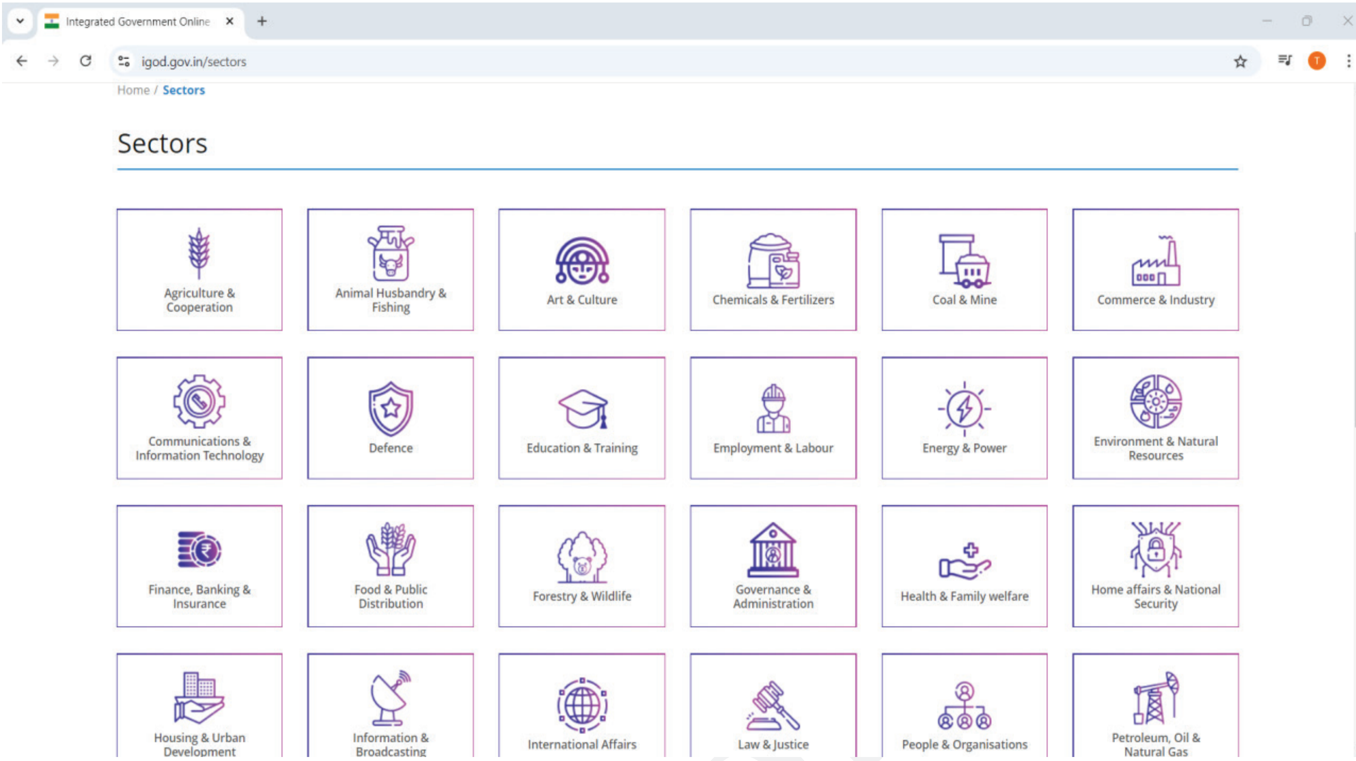
* सभी सरकारी सेवाओं का एकत्रित रूप से डिजिटलीकरण।
* नागरिकों को विभिन्न सरकारी योजनाओं, नीतियों और सेवाओं के बारे में जानकारी प्रदान करना।
* गोडायरेक्टरी.एनआसी.इन goidirectory.nic.in (Government of India Web Directory)



चित्र 2.18 goidirectory.nic.in की वेबसाइट

भारत सरकार द्वारा संचालित एक आधिकारिक ऑनलाइन पोर्टल है, जो सरकारी विभागों, संस्थाओं, और मंत्रालयों के वेबसाइटों और ऑनलाइन सेवाओं की सूची प्रदान करता है। इसका उद्देश्य नागरिकों, व्यवसायों, और सरकारी अधिकारियों को भारत सरकार की वेबसाइटों और सरकारी पोर्टलों तक पहुंच को सुविधाजनक बनाना है।





चित्र 2.19 goidirectory.nic.in की वेबसाइट में श्रेणियाँ

2. e-courts(https://ecourts.gov.in/ecourts_home/)

e-Courts पोर्टल भारत में न्यायिक प्रणाली को डिजिटलीकरण करने के उद्देश्य से विकसित किया गया है। यह पोर्टल नागरिकों को अदालतों में मामले दर्ज करने, फैसलों की जानकारी प्राप्त करने, और अदालत से जुड़ी अन्य सेवाएं ऑनलाइन प्राप्त करने की सुविधा देता है।

* **मुख्य उद्देश्य:**
  * न्यायिक प्रक्रियाओं को सरल और पारदर्शी बनाना।
  * नागरिकों को डिजिटल माध्यम से अदालतों में अपनी स्थिति और फैसलों की जानकारी प्राप्त करने का अवसर देना।

2.3.2 ई-गवर्नेंस नागरिकों को किस प्रकार सशक्त बनाता है?

* **सुलभता और पारदर्शिता:** ई-गवर्नेंस सेवाओं के माध्यम से नागरिकों को सरकारी योजनाओं, नीतियों और सेवाओं तक पहुँच आसान हो जाती है। वे इंटरनेट के जरिए बिना किसी बिचौलिए के सरकारी जानकारी प्राप्त कर सकते हैं, जिससे पारदर्शिता बढ़ती है और भ्रष्टाचार की संभावना कम होती है।

* **समय और स्थान की स्वतंत्रता:** नागरिकों को किसी भी सरकारी काम के लिए लंबी कतारों में खड़े होने या कार्यालयों में जाने की आवश्यकता नहीं होती। वे घर बैठे ऑनलाइन सेवाओं का लाभ उठा सकते हैं, जैसे कि प्रमाण पत्रों की प्राप्ति, बिलों का भुगतान, और आवेदन करना, जिससे समय की बचत होती है।

* **सामाजिक और आर्थिक समावेशन:** ई-गवर्नेंस द्वारा दूरदराज के क्षेत्रों में रहने वाले लोग भी आसानी से सरकारी सेवाओं का लाभ उठा सकते हैं, जिससे उनका सामाजिक और आर्थिक समावेशन बढ़ता है। यह खासकर ग्रामीण और पिछड़े इलाकों में रहने वाले नागरिकों को सशक्त करता है।

गतिविधि 2.3

राष्ट्रीय पोर्टल ऑफ इंडिया की वेबसाइट इस्तेमाल करके किन्ही तीन सरकारी योजना की जानकारी इकट्ठा कर।

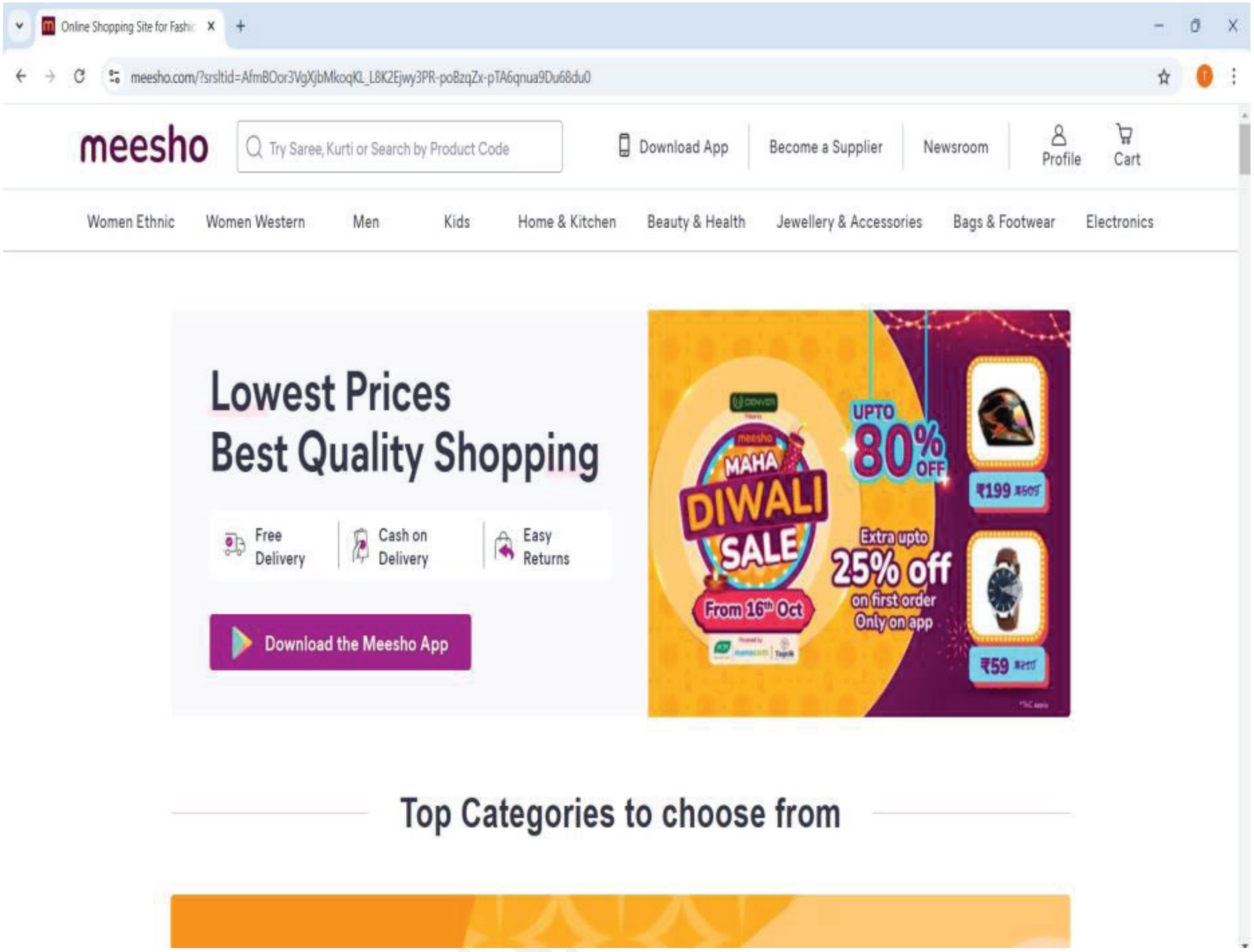
अभ्यास प्रश्न :

1. वेब आधारित एप्लिकेशन क्या है?
2. ई-गवर्नेंस क्या है?
3. ऑनलाइन आरक्षण प्रणाली क्या है ?
4. IRCTC का पूर्ण रूप लिखो और ये किस काम आता है?

2.4 ऑनलाइन शॉपिंग और बिल भुगतान

2.4.1 ऑनलाइन शॉपिंग (Online Shopping)

ऑनलाइन शॉपिंग का मतलब है इंटरनेट के माध्यम से सामान और सेवाओं की खरीदारी करना। इसमें उपभोक्ता विभिन्न ई-कॉमर्स वेबसाइट्स या ऐप्स के जरिए उत्पादों का चयन करते हैं और उन्हें घर बैठे खरीद सकते हैं। यह शॉपिंग का एक आधुनिक तरीका है, जो समय और स्थान की सीमाओं को पार कर देता है। भारत में, ऑनलाइन शॉपिंग ने पारंपरिक शॉपिंग के तरीके को काफी हद तक बदल दिया है, क्योंकि इसके जरिए लोग अपने घरों से ही सब कुछ खरीद सकते हैं।



चित्र 2.20 ऑनलाइन शॉपिंग की वेबसाइट चित्र

2.4.1.1 ऑनलाइन शॉपिंग के लाभ (Advantages of Online Shopping)

* **सुविधाजनक और समय की बचतः**

    * ऑनलाइन शॉपिंग से आपको किसी भी दुकान में जाने की जरूरत नहीं होती। आप घर बैठे या कहीं भी इंटरनेट का इस्तेमाल कर सामान खरीद सकते हैं। इसके लिए न तो लंबी कतारों में खड़ा होना पड़ता है और न ही ट्रैफिक का सामना करना पड़ता है।

* **24x7 उपलब्धताः**

    * ऑनलाइन शॉपिंग साइट्स और ऐप्स हमेशा खुली रहती हैं। आप किसी भी समय शॉपिंग कर सकते हैं, चाहे दिन हो या रात।

* **विविधता और चयनः**
    * ऑनलाइन शॉपिंग के जरिए आपको विभिन्न उत्पादों और ब्रांड्स के बीच चयन करने का मौका मिलता है। बाजार में जो भी नई चीज़ आती है, वह तुरंत ऑनलाइन उपलब्ध हो जाती है।
* **सस्ती कीमतें और छूटः**
    * अक्सर ऑनलाइन शॉपिंग साइट्स पर विभिन्न डिस्काउंट, ऑफर्स और कूपन मिलते हैं, जो पारंपरिक दुकानों से कम कीमतों में खरीदारी करने का मौका देते हैं।
* **घर बैठे डिलीवरीः**
    * आपके द्वारा खरीदे गए सामान को घर तक डिलीवर किया जाता है। इससे आपको सामान लाने के लिए बाहर नहीं जाना पड़ता।
* **उत्पादों की तुलनाः**
    * ऑनलाइन शॉपिंग साइट्स पर विभिन्न उत्पादों की तुलना करना बहुत आसान होता है। आप कीमत, गुणवत्ता और अन्य सुविधाओं के आधार पर एक ही प्रकार के उत्पादों को आपस में तुलना कर सकते हैं।

2.4.1.2 भारत में प्रमुख ऑनलाइन शॉपिंग साइट्स (Popular Online Shopping Websites in India)

* **अमेज़न इंडिया (Amazon india):**
    * अमेज़न इंडिया एक प्रमुख ऑनलाइन शॉपिंग प्लेटफ़ॉर्म है, जो इलेक्ट्रॉनिक्स, फैशन, घर का सामान, किताबें, खिलौने, और बहुत कुछ बेचता है। इसमें अच्छे डिस्काउंट और कस्टमर सर्विस की सुविधा मिलती है।
* **फ्लिप्कार्ट (Flipkart):**
    * फ्लिपकार्ट भारत की एक बड़ी ई-कॉमर्स कंपनी है, जो सामान की एक विस्तृत रेंज ऑफर करती है। यहाँ पर आपको स्मार्टफोन्स, कपड़े, फैशन, इलेक्ट्रॉनिक्स, घरेलू सामान आदि मिलते हैं।

2.4.1.3 ऑनलाइन शॉपिंग के लिए टिप्स (Tips for Online Shopping)

* **समीक्षाएँ पढ़ेंः** उत्पाद खरीदने से पहले उसकी समीक्षाएँ (reviews) और रेटिंग्स ज़रूर पढ़ें, ताकि आप उत्पाद की गुणवत्ता और विक्रेता के बारे में सही जानकारी प्राप्त कर सकें।

SCERT DELHI

* **सेक्योर पेमेंट गेटवे का उपयोग करें:** ऑनलाइन पेमेंट करते समय हमेशा सेक्योर पेमेंट गेटवे का चयन करें और सुनिश्चित करें कि साइट का URL HTTPS से शुरू हो।
* **केवल विश्वसनीय साइट्स से खरीदारी करें:** विश्वसनीय और प्रमाणित ई-कॉमर्स साइट्स से ही सामान खरीदें, जैसे अमेज़न, फ्लिपकार्ट, मिंत्रा आदि।
* **छूट और कूपन का उपयोग करें:** ऑनलाइन शॉपिंग के दौरान विभिन्न ऑफर्स, डिस्काउंट्स, और कूपन का फायदा उठाएं।
* **रिटर्न और एक्सचेंज पॉलिसी पढ़ें:** किसी भी उत्पाद को खरीदने से पहले उसकी रिटर्न पॉलिसी और एक्सचेंज पॉलिसी को ध्यान से पढ़ें।

गतिविधि 2.4

ऑनलाइन शॉपिंग की वेबसाइट का इस्तेमाल करके कोई भी जुटे आर्डर करे।

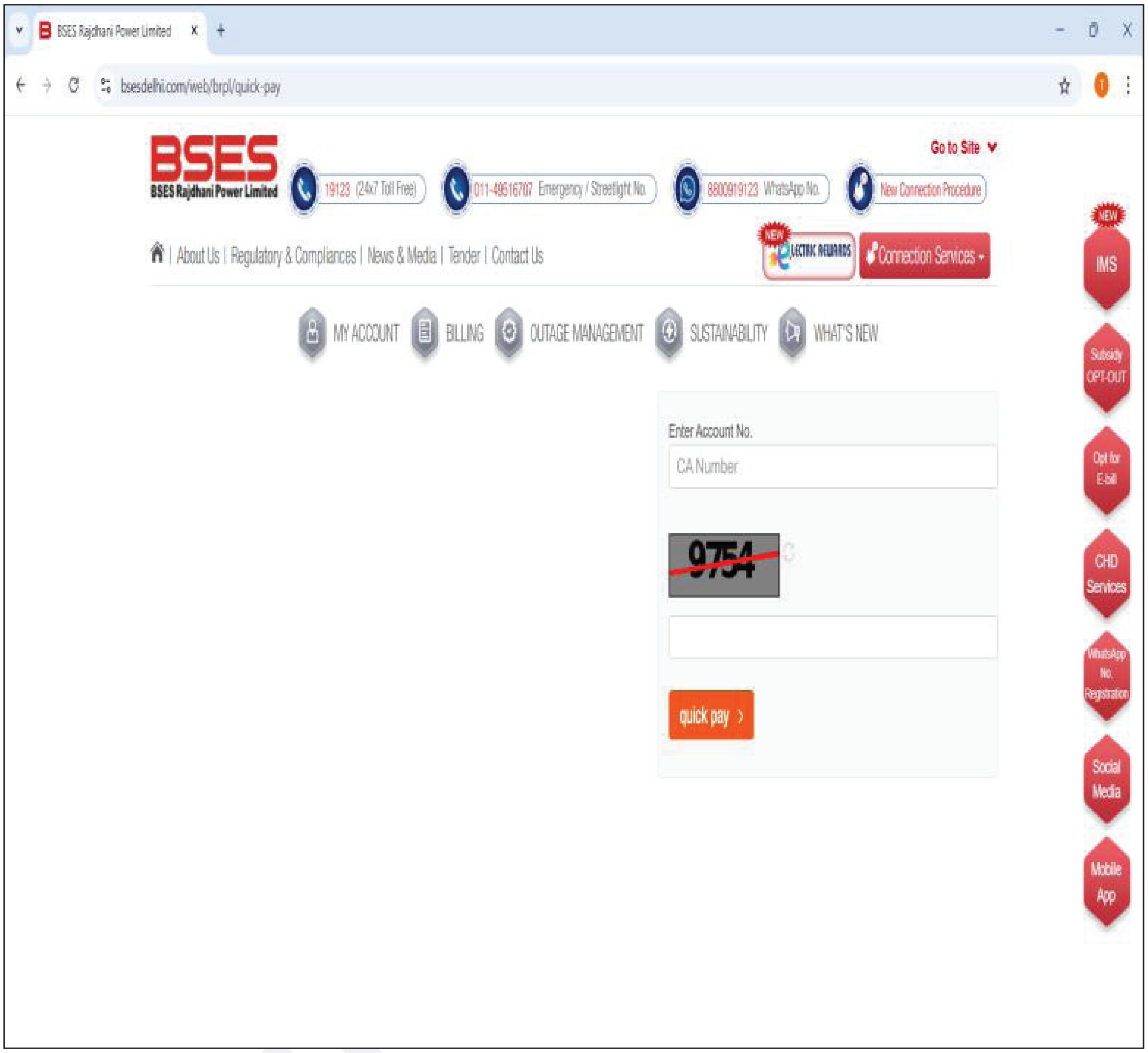
2.4.2 बिल भुगतान (Bill Payments)

बिल भुगतान एक ऐसी प्रक्रिया है जिसमें लोग अपनी मासिक या अन्य प्रकार की बिलों की रकम को समय पर चुकाते हैं। यह आमतौर पर घर के उपयोग से जुड़े बिलों जैसे बिजली, पानी, गैस, मोबाइल, इंटरनेट, टीवी, क्रेडिट कार्ड, और अन्य सेवाओं के भुगतान के रूप में होता है। पहले के समय में, लोग यह भुगतान बैंक में जाकर या पोस्ट ऑफिस में जाकर करते थे, लेकिन अब इंटरनेट और मोबाइल एप्लिकेशन्स के माध्यम से यह प्रक्रिया बहुत आसान और सुविधाजनक हो गई है।

2.4.2.1 बिल भुगतान के तरीके (Methods of Bill Payments)

* **ऑनलाइन बिल भुगतान (Online Bill Payment):**
  * यह सबसे आसान और सुविधाजनक तरीका है। आप विभिन्न वेबसाइट्स या मोबाइल ऐप्स के माध्यम से अपनी बिलों का भुगतान कर सकते हैं। इसमें कई विकल्प होते हैं:
  - नेट बैंकिंग: आपके बैंक खाते से सीधे बिल का भुगतान।
  - डेबिट/क्रेडिट कार्ड: अपने कार्ड से ऑनलाइन भुगतान।
  - यूपीआई (UPI): यूनिफाइड पेमेंट इंटरफेस (UPI) के माध्यम से भुगतान करना।
  - ई-वॉलेट्स: जैसे पेटीएम(Paytm), फ़ोनपे(PhonePe), गूगलपे(Google Pay) आदि से भुगतान।

SCERT DELHI



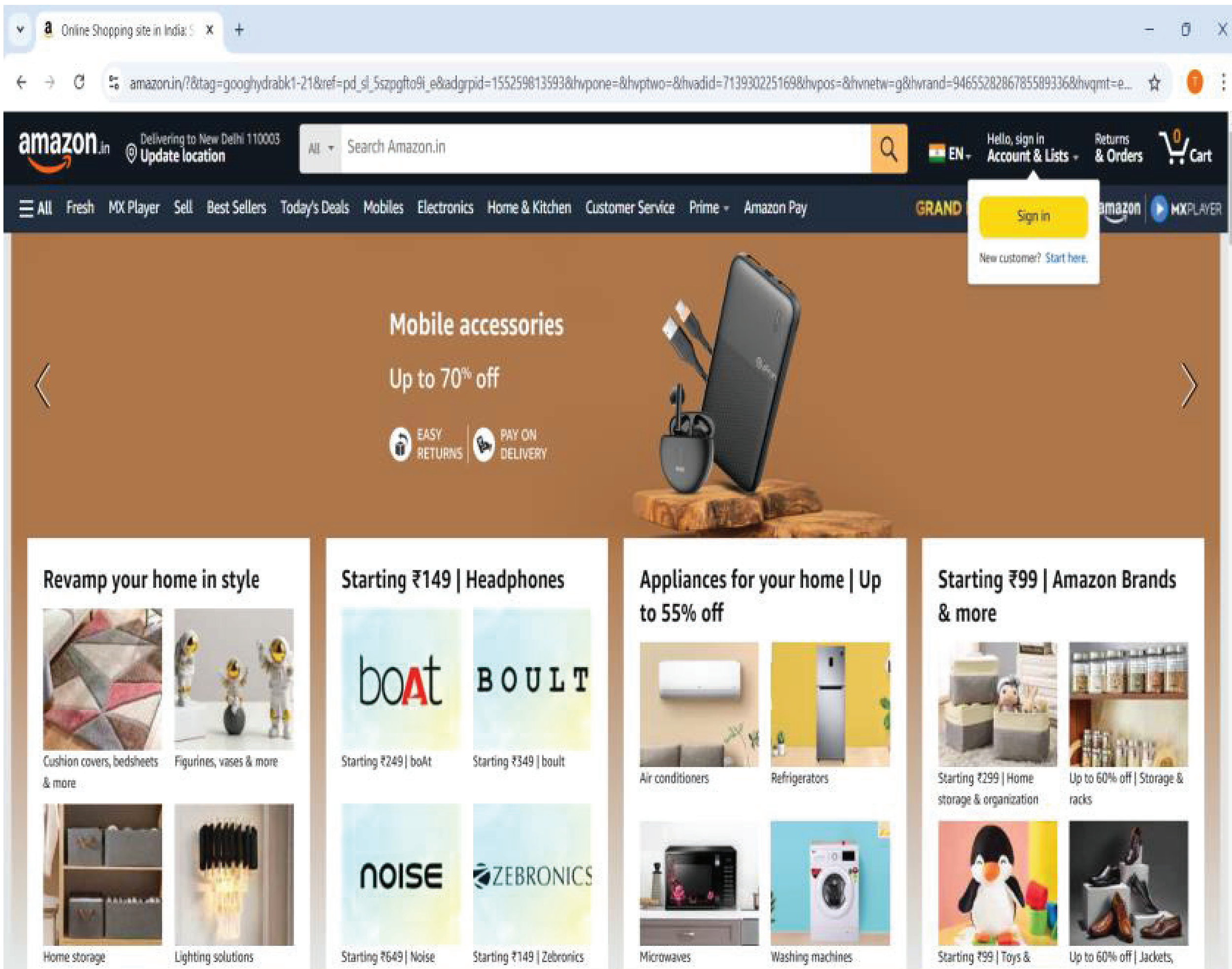
2.21 ऑनलाइन बिल भुगतान

* **ऑटोमेटेड बिल भुगतान (Automated Bill Payment):**
    * इसमें, आपके बैंक खाते से हर महीने स्वतः तय राशि कट जाती है।
* **पेटीएम और अन्य वॉलेट्सः**
    * पेटीएम(Paytm), फ़ोनपे(PhonePe), गूगलपे(Google Pay) जैसे डिजिटल वॉलेट्स के माध्यम से आप विभिन्न प्रकार के बिलों का भुगतान कर सकते हैं।
* **मोबाइल एप्लिकेशन्सः**
    * कई टेलीकॉम कंपनियां जैसे एयरटेल (Airtel), जिओ (Jio), बीएसएनएल (BSNL) और वोडाफोन (Vodafone) अपनी बिल भुगतान सेवा के लिए मोबाइल ऐप्स प्रदान करती हैं,

जिनसे आप आसानी से मोबाइल रिचार्ज, डेटा पैक, और DTH बिल्स का भुगतान कर सकते हैं।

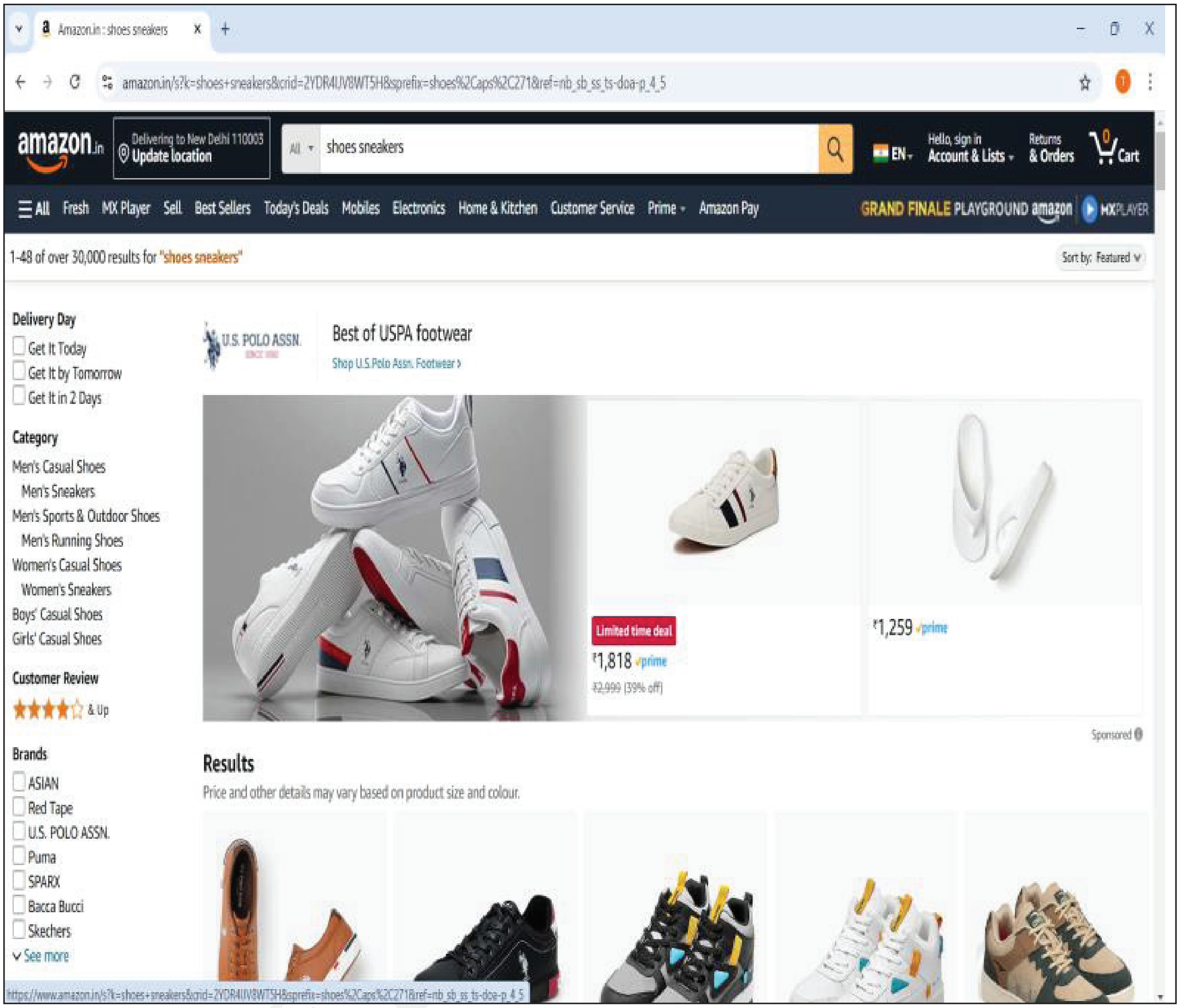
2.4.2.2 केस स्टडी: ऑनलाइन शॉपिंग वेबसाइट का उपयोग करके ऑनलाइन शॉपिंग

* विजिटिंग वेबसाइट: ग्राहक अपने ब्राउज़र में एक ऑनलाइन शॉपिंग वेबसाइट का नाम टाइप करके उसे खोलते हैं। उदाहरण के लिए, ग्राहक Amazon.in या किसी भी एप्लीकेशन पर जा सकते हैं।



2.21 ऑनलाइन शॉपिंग वेबसाइट का उदाहरण

* **खोज और चयन:** वेबसाइट पर विभिन्न श्रेणियां जैसे इलेक्ट्रॉनिक्स, फैशन, किचन और होम अप्लायंसेस, आदि के विकल्प होते हैं। ग्राहक अपनी पसंद के अनुसार प्रोडक्ट का चयन करते हैं और उसे कार्ट में डालते हैं।



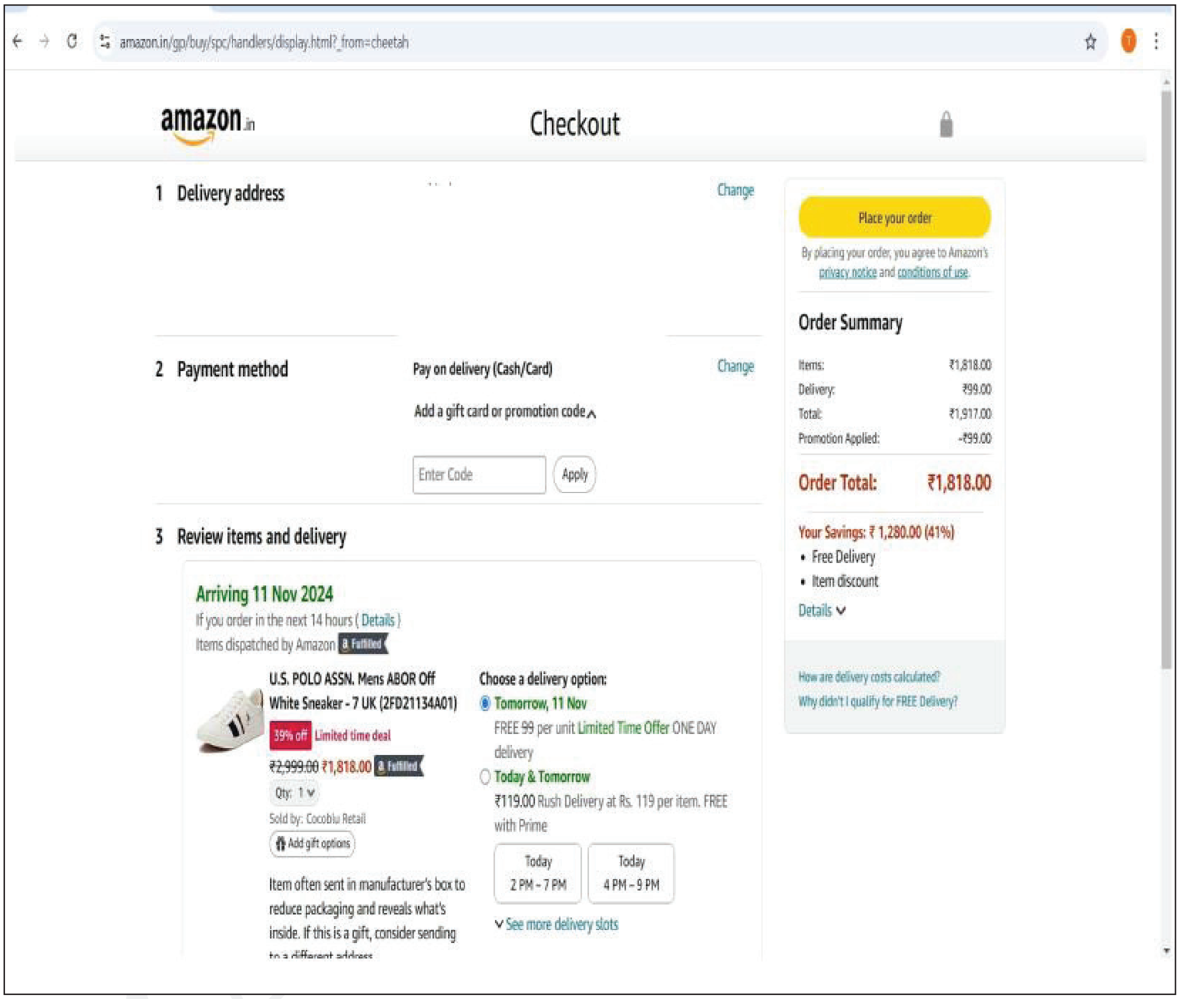
2.22 ऑनलाइन शॉपिंग वेबसाइट का उदाहरण

* **प्रोडक्ट डिटेल्स और रिव्यू:** ग्राहक प्रोडक्ट का विवरण, मूल्य, रिव्यू और रेटिंग्स को ध्यान से पढ़ते हैं ताकि वे सही चयन कर सकें। इससे उन्हें उत्पाद की गुणवत्ता और अन्य ग्राहकों के अनुभव का अंदाजा होता है।

* **ऑर्डर प्लेस करना:** एक बार जब ग्राहक उत्पाद का चयन कर लेता है, तो वह 'Add to Cart' या 'Buy Now' पर क्लिक करता है। फिर वह अपनी डिलीवरी जानकारी, जैसे पता और संपर्क विवरण दर्ज करता है।

* भुगतान विकल्पः भुगतान के विभिन्न विकल्प होते हैं, जैसेः

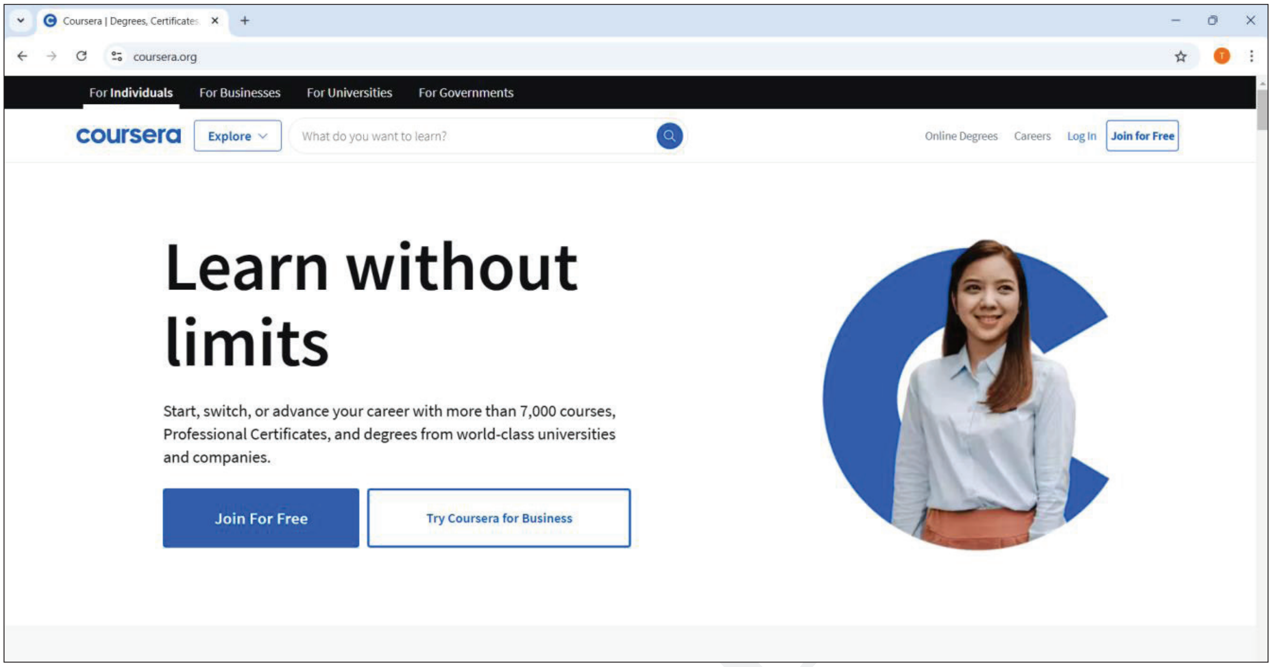
- कैश ऑन डिलीवरी (COD): ग्राहक डिलीवरी के समय भुगतान करते हैं।
- ऑनलाइन भुगतानः क्रेडिट कार्ड, डेबिट कार्ड, नेट बैंकिंग, और UPI जैसी सुविधाओं के माध्यम से भुगतान किया जा सकता है।



2.23 ऑनलाइन शॉपिंग भुगतान का उदाहरण

- **डिलीवरी और ट्रैकिंगः** भुगतान के बाद, वेबसाइट एक अनुमानित डिलीवरी तारीख देती है। ग्राहक अपने ऑर्डर की स्थिति को ट्रैक कर सकते हैं और यह देख सकते हैं कि उनका पैकेज कहाँ तक पहुँचा है।
- **रिटर्न और एक्सचेंजः** यदि ग्राहक को उत्पाद में कोई समस्या होती है या वह संतुष्ट नहीं होते, तो अधिकांश वेबसाइट्स रिटर्न और एक्सचेंज की सुविधा देती हैं। यह ग्राहक को 7-15 दिनों के अंदर रिटर्न करने का मौका देती हैं।

2.5 ऑनलाइन कोर्स, ट्यूटोरियल्स और टेस्ट्स



चित्र 2.24 ऑनलाइन कोर्सेस की वेबसाइट

आज के डिजिटल युग में शिक्षा के क्षेत्र में भारी बदलाव आया है, और ऑनलाइन कोर्सेस, ट्यूटोरियल्स और टेस्ट्स ने सीखने के तरीके को पूरी तरह से बदल दिया है। अब लोग कहीं से भी, कभी भी, और अपनी गति से सीख सकते हैं। चाहे वह एक नया कौशल सीखना हो, किसी विशिष्ट विषय पर गहन ज्ञान प्राप्त करना हो, या परीक्षा की तैयारी करनी हो, ऑनलाइन शिक्षा ने इसे आसान और सुलभ बना दिया है।

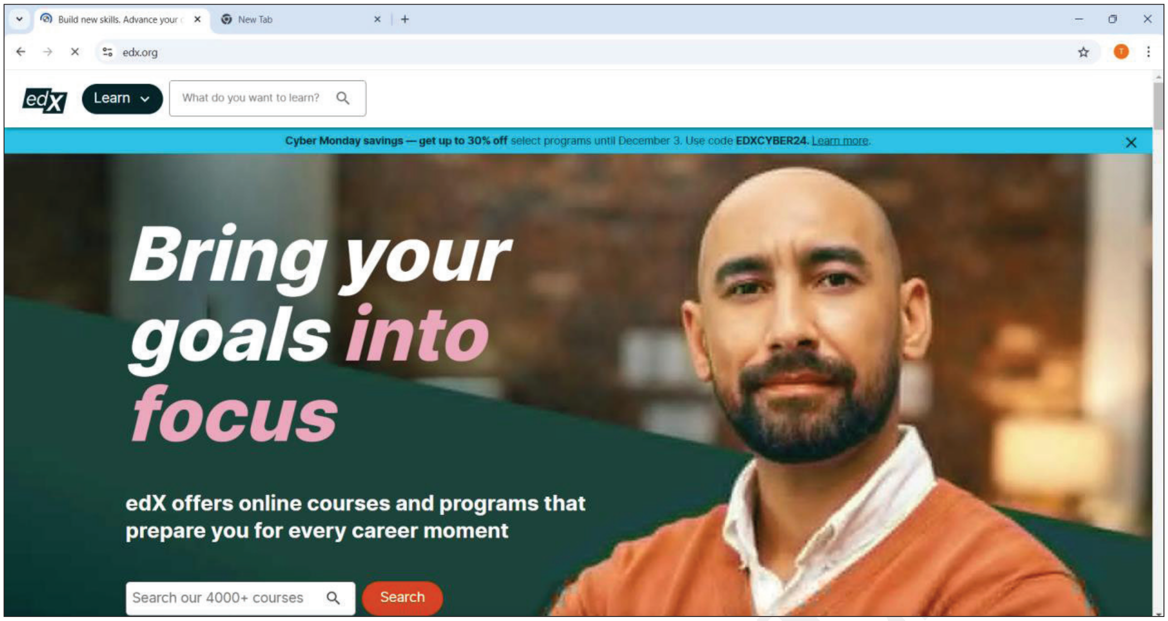
2.5.1 ऑनलाइन कोर्स (Online Courses):

ऑनलाइन कोर्सेस एक structured तरीके से पढ़ाई की पेशकश करते हैं। ये कोर्स किसी भी विषय में हो सकते हैं और इन्हें अक्सर प्रसिद्ध शैक्षिक प्लेटफार्मों द्वारा तैयार किया जाता है, जैसे:

* **कोउर्सेरा (Coursera)** – यहां आपको विश्वविद्यालयों और कॉलेजों के कोर्स मिलते हैं, जिनमें हर विषय से संबंधित कोर्स होते हैं जैसे कंप्यूटर विज्ञान, बिजनेस, आर्ट्स, साइकोलॉजी, और बहुत कुछ।

* **उदेमी (Udemy)** – यह एक प्रमुख प्लेटफ़ॉर्म है, जहां पेशेवर ट्रेनर्स और विशेषज्ञ विभिन्न विषयों पर कोर्स ऑफर करते हैं, जैसे वेब डेवलपमेंट, मार्केटिंग, ग्राफिक डिज़ाइन, आदि।

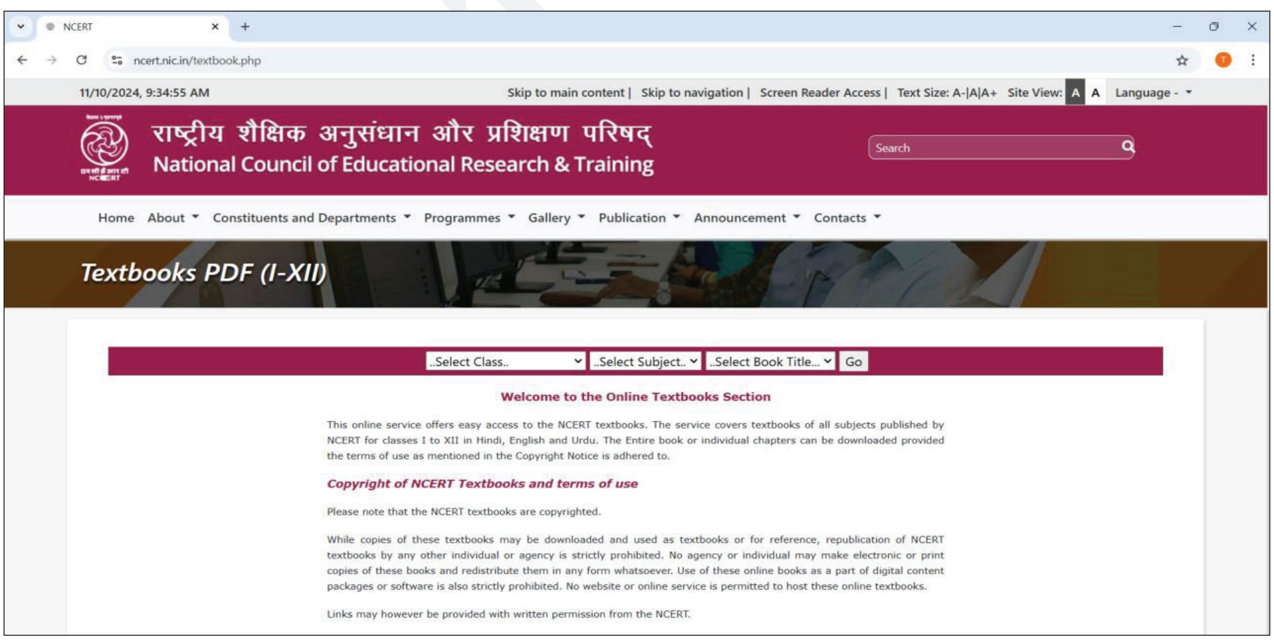
* **इडीएक्स (edX)** – यह एक उच्च शिक्षा प्लेटफार्म है जो प्रमुख विश्वविद्यालयों से ऑनलाइन कोर्स प्रदान करता है।



चित्र 2.25 ऑनलाइन कोर्सेस की वेबसाइट

* **फ्यूचर लर्न (FutureLearn)** – यह प्लेटफ़ॉर्म भी उच्च गुणवत्ता वाले कोर्स प्रदान करता है, विशेष रूप से समाजशास्त्र, विज्ञान, और मानविकी के क्षेत्र में।

* **http://www.ncert.nic.in/index.html**- यह एक एनसीईआरटी का पोर्टल है जो ऑनलाइन ई-पुस्तकें, जर्नल, प्रश्न पत्र, बच्चों की किताबें आदि के रूप में शिक्षण प्रदान करता है ।



चित्र 2.26 NCERT की वेबसाइट

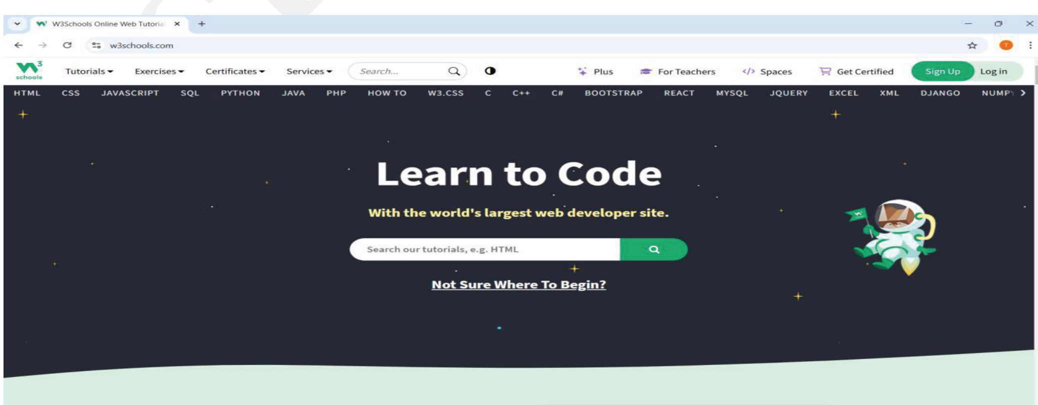
* **जीसीएफलर्नफ्री.ऑर्ग (GCFLearnFree.org):** GCFLearnFree.org का हिंदी में मतलब है जीसीएफलर्नफ्री.ऑर्ग, जो एक प्रसिद्ध ऑनलाइन शैक्षिक प्लेटफ़ॉर्म है, जो मुफ्त में कंप्यूटर, इंग्लिश, गणित, करियर, और अन्य कौशल सिखाता है। यह साइट खासकर उन लोगों के लिए उपयोगी है जो बुनियादी कंप्यूटर और सॉफ़्टवेयर ज्ञान प्राप्त करना चाहते हैं।

ऑनलाइन कोर्स में आमतौर पर निम्नलिखित सुविधाएं होती हैं:

* **वीडियो लेक्चर्स:** जहां शिक्षक वीडियो के माध्यम से सीखाते हैं।
* **असाइनमेंट्स और क्विज़:** जो छात्रों को पाठ्य सामग्री को समझने में मदद करते हैं।
* **प्रमाणपत्र:** कोर्स पूरा करने पर प्रमाणपत्र प्रदान किया जाता है, जो आपके कौशल और ज्ञान को प्रमाणित करता है।
* **समय का लचीलापन:** छात्रों को अपने समय के अनुसार कोर्स करने की स्वतंत्रता होती है।

2.5.2 ऑनलाइन ट्यूटोरियल्स (Online Tutorials):

* ऑनलाइन ट्यूटोरियल्स एक विशेष विषय पर गाइडेड इंस्ट्रक्शन होते हैं, जो किसी को सीखने में मदद करते हैं। ये वीडियो, डॉक्युमेंट्स या वेब पेज के रूप में हो सकते हैं।
* **यूट्यूब (YouTube):** यूट्यूब पर लाखों फ्री ट्यूटोरियल्स उपलब्ध हैं, जो किसी भी विषय पर होते हैं। उदाहरण के लिए, प्रोग्रामिंग, ग्राफिक्स डिज़ाइन, गिट और गिटहब, कंप्यूटर सॉफ्टवेयर, मैथ्स, विज्ञान, और कई अन्य।
* **W3Schools:** वेब डेवलपमेंट से संबंधित ट्यूटोरियल्स के लिए यह एक बहुत अच्छा स्रोत है। HTML, CSS, JavaScript, SQL, आदि पर यहां ट्यूटोरियल्स उपलब्ध हैं।



चित्र 2.27 ऑनलाइन ट्यूटोरियल्स की वेबसाइट

* **खान अकादमी (Khan Academy):** खासकर गणित, विज्ञान, इतिहास, और कला के लिए, यह एक बहुत अच्छा ट्यूटोरियल प्लेटफ़ॉर्म है।

ट्यूटोरियल्स के फायदे:

* **व्यक्तिगत मार्गदर्शन:** यहां विद्यार्थी को हर चरण में मदद मिलती है।

* **विभिन्न रूपों में उपलब्धता:** वीडियो, ब्लॉग, या इन्फोग्राफिक्स के रूप में उपलब्ध होते हैं।

* **स्वतंत्रता और लचीलापन:** छात्र अपनी सुविधानुसार किसी भी समय ट्यूटोरियल देख सकते हैं।

2.5.3 ऑनलाइन टेस्ट्स (Online Tests):

ऑनलाइन टेस्ट्स छात्रों को उनके ज्ञान की माप करने और आत्ममूल्यांकन करने का अवसर प्रदान करते हैं। ये टेस्ट्स शैक्षिक पाठ्यक्रमों, प्रतियोगी परीक्षाओं और अन्य कौशल परीक्षणों के लिए होते हैं।

* **टेस्टबुक (Testbook):** यह प्लेटफॉर्म भारत में प्रतियोगी परीक्षा की तैयारी करने वाले छात्रों के लिए बहुत लोकप्रिय है। यहां पर आप बैंकिंग, SSC, रेलवे, UPSC, आदि के लिए ऑनलाइन मॉक टेस्ट्स और क्विज़ कर सकते हैं।

* **ग्रेडअप (Gradeup):** यह भी एक प्रमुख प्लेटफार्म है जहां आप विभिन्न सरकारी परीक्षाओं के लिए मॉक टेस्ट्स और क्विज़ दे सकते हैं।

ऑनलाइन टेस्ट्स के फायदे:

* **प्रैक्टिस का मौका:** विद्यार्थी इन टेस्ट्स के माध्यम से किसी भी परीक्षा की तैयारी में प्रैक्टिस कर सकते हैं।

* **समय प्रबंधन:** टेस्ट देते समय समय का प्रबंधन करना सिखाया जाता है।

* **स्वचालित मूल्यांकन:** टेस्ट तुरंत मूल्यांकन कर दिए जाते हैं, जिससे छात्र को अपनी कमजोरी का पता चलता है।

* **कई प्रकार के टेस्ट्स:** ऑनलाइन प्लेटफार्म पर विभिन्न प्रकार के मॉक टेस्ट्स, क्विज़, और असाइनमेंट्स होते हैं जो विभिन्न परीक्षाओं की तैयारी के लिए उपयुक्त होते हैं।

2.5.4 ऑनलाइन कोर्सेस, ट्यूटोरियल्स और टेस्ट्स के लाभ:

* **समय और स्थान की स्वतंत्रता:** विद्यार्थी अपनी गति से और अपनी सुविधा के अनुसार अध्ययन कर सकते हैं।

SCERT DELHI

* **कम लागत:** पारंपरिक शिक्षा की तुलना में ऑनलाइन शिक्षा सस्ती होती है।

* **कस्टमाइज्ड लर्निंग:** हर छात्र अपनी आवश्यकता के अनुसार कोर्स और टेस्ट का चयन कर सकता है।

* **24/7 उपलब्धता:** विद्यार्थी किसी भी समय पाठ्य सामग्री और टेस्ट्स का उपयोग कर सकते हैं।

* **ग्लोबल एक्सेस:** इंटरनेट के माध्यम से छात्र दुनिया भर से सीख सकते हैं और विशेषज्ञों से मार्गदर्शन प्राप्त कर सकते हैं।

गतिविधि 2.5
ऑनलाइन वेबसाइट द्वारा अपनी इंग्लिश या हिंदी विषय की बुक ऑनलाइन पढ़े तथा उस वेबसाइट की चर्चा कक्षा में करे ।

अभ्यास प्रश्न :

1. ऑनलाइन कोर्सेस की तीन वेबसाइट के नाम बताये ?
2. ऑनलाइन ट्यूटोरियल्स क्या है?
3. एनसीईआरटी के पोर्टल का क्या नाम है और उसका क्या फयदा है?
4. बिल भुगतान क्या है?

2.6. परियोजना प्रबंधन- वेब आधारित एप्लिकेशन विकास (Project Management – Web Based Application development)

वेब आधारित एप्लिकेशन (Web-based Application) एक प्रकार का सॉफ़्टवेयर एप्लिकेशन होता है जिसे इंटरनेट ब्राउज़र के माध्यम से एक्सेस किया जाता है।वेब आधारित एप्लिकेशन विकास

(Web-based Application development) से हम किसी भी वेब एप्लीकेशन का निर्माण कर सकते है ।इसे क्लाइंट-सर्वर आर्किटेक्चर पर काम करने के लिए डिजाइन किया जाता है।

चित्र 2.28 परियोजना प्रबंधन

वेब एप्लिकेशन्स का उपयोग विभिन्न उद्देश्यों के लिए किया जाता है जैसे ई-कोमर्स वेबसाइट्स, सोशल मीडिया प्लेटफॉर्म्स, ऑनलाइन बैंकिंग सिस्टम्स, और बिजनेस एप्लिकेशन्स। वेब एप्लिकेशन डेवलपमेंट के दौरान प्रोजेक्ट मैनेजमेंट का महत्वपूर्ण भूमिका होती है ताकि प्रोजेक्ट समय पर, बजट के भीतर और उच्च गुणवत्ता के साथ पूरा हो सके। एक वेब एप्लिकेशन विभिन्न प्रकार का हो सकता है, जैसे, प्रश्नोत्तरी (quiz), गेम (game), या एक बिल कैलकुलेटर (bill calculator)।

2.6.1 प्रोजेक्ट (project) क्या है?

चित्र 2.29 प्रोजेक्ट को दर्शाता एक चित्र

एक परियोजना (project) एक ऐसा कार्य है जो एक अद्वितीय उत्पाद(unique product), सेवा(service) या परिणाम बनाने के लिए किया जाता है।एक प्रोजेक्ट की शुरुआत और अंत होता है। किसी प्रोजेक्ट के लिए सीमित संसाधनों की आवश्यकता होती है जो प्रोजेक्ट को पूरा करने के लिए आवश्यक होते हैं। किसी प्रोजेक्ट की एक विशिष्ट समय सीमा होती है। इसकी एक निश्चित शुरुआत और समाप्ति तिथियां हैं।

2.6.2 वेब आधारित एप्लिकेशन प्रोजेक्ट की प्रमुख विशेषताएँ (Key Characteristics of a Web-Based Application Project)

वेब आधारित एप्लिकेशन प्रोजेक्ट एक सॉफ़्टवेयर प्रोजेक्ट होता है, जो इंटरनेट पर आधारित होता है

और वेब ब्राउज़र के माध्यम से एक्सेस किया जाता है। इन प्रोजेक्ट्स की कुछ विशिष्ट विशेषताएँ होती हैं, जो उन्हें अन्य प्रकार के प्रोजेक्ट्स से अलग बनाती हैं। यहां वेब आधारित एप्लिकेशन प्रोजेक्ट की प्रमुख विशेषताओं को बताया गया हैः

1. **स्पष्ट उद्देश्य और आवश्यकताएँ (Clear Objective and Requirements)**

* वेब एप्लिकेशन प्रोजेक्ट का एक स्पष्ट उद्देश्य होता है, जैसे कि कोई नया सॉफ़्टवेयर या सेवा प्रदान करना। इसके अलावा, प्रोजेक्ट की शुरुआत में क्लाइंट या उपयोगकर्ताओं की आवश्यकताओं को सही तरीके से समझना और उनका दस्तावेजीकरण करना महत्वपूर्ण होता है।

* उदाहरणः ई-कॉमर्स वेबसाइट, शिक्षा के लिए ऑनलाइन प्लेटफार्म, या किसी विशेष उद्योग के लिए एप्लिकेशन।

2. **समय सीमा (Time-Bound)**

* जैसे सभी प्रोजेक्ट्स, वेब एप्लिकेशन प्रोजेक्ट्स की भी एक निर्धारित समय सीमा होती है। प्रोजेक्ट के विभिन्न चरणों के लिए स्पष्ट समयसीमा तय की जाती है, जैसे कि डिज़ाइन, विकास, परीक्षण और डिप्लॉयमेंट।

* समय सीमा का पालन करना आवश्यक है ताकि प्रोजेक्ट निर्धारित तिथि पर पूरा हो और ग्राहकों की अपेक्षाएँ पूरी की जा सकें।

3. **विशिष्ट संसाधन (Specific Resources)**

* वेब एप्लिकेशन प्रोजेक्ट के लिए विशेष प्रकार के संसाधनों की आवश्यकता होती है, जैसेः

    * **मानव संसाधनः** वेब डेवलपर्स, डिजाइनर्स, टेस्टर्स, और प्रोजेक्ट मैनेजर।

    * **तकनीकी संसाधनः** सर्वर, डेटाबेस, वेब होस्टिंग, और विकास के उपकरण (जैसे IDEs, CMS, आदि)।

    * **वित्तीय संसाधनः** बजट और अन्य वित्तीय संसाधन।

4. **प्रमुख चरण (Phases of Development)**

* वेब एप्लिकेशन प्रोजेक्ट आमतौर पर निम्नलिखित प्रमुख चरणों में विभाजित होता हैः

    * **आवश्यकताओं का विश्लेषण (Requirement Analysis):** यह चरण प्रोजेक्ट के उद्देश्य और आवश्यकताओं को समझने का होता है।

SCERT DELHI

* **डिज़ाइन (Design):** यूजर इंटरफेस (UI) और यूजर अनुभव (UX) डिज़ाइन किया जाता है, साथ ही साथ एप्लिकेशन के आर्किटेक्चर का डिज़ाइन भी तैयार होता है।

* **विकास (Development):** कोडिंग, डेटाबेस सेटअप, और एप्लिकेशन के सभी कार्यात्मक भागों का विकास किया जाता है।

* परीक्षण (Testing): एप्लिकेशन की गुणवत्ता सुनिश्चित करने के लिए इसे विभिन्न परीक्षणों (जैसे फंक्शनल, सुरक्षा, प्रदर्शन परीक्षण) से गुजरना पड़ता है।

* डिप्लॉयमेंट (Deployment): जब एप्लिकेशन तैयार हो जाता है, तो इसे सर्वर पर डिप्लॉय किया जाता है ताकि उपयोगकर्ता इसका उपयोग कर सकें।

2.7. परियोजना की अनिवार्यताएँ और युक्तियाँ (Project essentials and tips)

हमने पिछले अनुभागों में कई अलग-अलग वेब एप्लिकेशन के बारे में सीखा है। इस में

अनुभाग में वेब के विकास के लिए एक संरचित और सरलीकृत प्रक्रिया का वर्णन किया गया है।

2.7.1 वेब एप्लिकेशन परियोजना के चरण(Phases in a Web Application Project)

वेब एप्लिकेशन प्रोजेक्ट को सफलतापूर्वक पूरा करने के लिए इसे विभिन्न चरणों में विभाजित किया जाता है। प्रत्येक चरण का अपना महत्व होता है और यह अंतिम उत्पाद की गुणवत्ता को सुनिश्चित करता है। वेब एप्लिकेशन के विकास के सामान्य चरण निम्नलिखित होते हैं:

2.7.1.1 वेब एप्लिकेशन परियोजना के चरण

* आवश्यकताएँ परिभाषा चरण (Requirements Definition Phase)

* डिजाइन चरण (Design Phase)

* कार्यान्वयन चरण (Implementation Phase)

* परीक्षण चरण (Testing Phase)

वेब आधारित एप्लिकेशन विकसित करने के लिए, हम उठाए जाने वाले कदमों पर चर्चा करते हैं

विकास के चार चरण इस प्रकार हैं:

चित्र 2.30 वेब एप्लिकेशन परियोजना के चरण

SCERT DELHI

1. आवश्यकताएँ परिभाषा चरण (Requirement Definition Phase):

इस चरण में हम समस्या कथन(problem statement) की पहचान करते हैं जो वेब एप्लिकेशन विकसित किया जाना है।समस्या के दायरे को पहचानें(Identify the scope of the problem)।

a. व्यवहार्यता (Feasibility): क्या परियोजना व्यवहार्य (Feasibility) है?

(i) जांचें कि क्या परियोजना तकनीकी रूप से व्यवहार्य(Feasibility) है (क्या ऐसा करना संभव है?)।

(ii) जांचें कि क्या परियोजना आर्थिक रूप से व्यवहार्य (Feasibility) है (क्या यह लाभदायक है?)।

(iii) यदि संभव हो तो आगे बढ़ें, अन्यथा परियोजना में आगे बढ़ने की कोई आवश्यकता नहीं है।

b. दायरा (Scope): केंद्र (focus) इस बात पर है कि एप्लिकेशन को "क्या" करना चाहिए। एप्लीकेशन का दायरा(scope) तय करे ।

(i) एप्लिकेशन सुविधाओं के स्पष्ट विवरण के साथ एक विस्तृत सूची संकलित करें।

(ii) वे लक्ष्य स्थापित करें जिन्हें समाधान द्वारा अवश्य प्राप्त किया जाना चाहिए।

(iii) परियोजना पर लगाई गई सीमाओं को पहचानना।

2. डिजाइन चरण (Design Phase) : एप्लिकेशन को "कैसे" डिज़ाइन किया जाना है, इस पर ध्यान केंद्रित किया गया है। डिज़ाइन में शामिल है:

a. वेब एप्लिकेशन साइट का मानचित्र: इसमें इसके बारे में आवश्यक जानकारी शामिल है

साइट की संरचना - पृष्ठ और उनके बीच संबंध।

b. डेटाबेस(Database): एप्लिकेशन डेटाबेस डिज़ाइन करें।

(i) उन डेटाबेस तालिकाओं(tables) की पहचान करें जिनकी एप्लिकेशन में आवश्यकता होगी।

(ii) **तालिका संरचना तय करें(Decide the table structures):** प्रत्येक तालिका(table) के लिए हमें इसकी पहचान करने की आवश्यकता है तालिकाओं(tables) की विशेषताएँ, उनके डेटा प्रकार(data TYPE), स्तंभों का आकार तालिकाएँ(size of column of table), तालिकाओं के बीच संबंध(relatioship between tables)।

c. पृष्ठ संरचना (Page Structure): पृष्ठ की संरचना डिज़ाइन करें।

3. **कार्यान्वयन चरण (Implementation Phase):** इसमें एप्लिकेशन या सिस्टम का वास्तविक सॉफ़्टवेयर विकसित किया जाता है। डेवलपर्स कोड लिखते हैं, यूनिट टेस्टिंग करते हैं, और सभी कार्यात्मकताओं को एक साथ जोड़ते हैं। इसमें निम्नलिखित प्रक्रिया शामिल हो सकती हैं:

a. बैक-एंड (Backend Database): डेटाबेस बनाएं और एप्लिकेशन की आवश्यकता के अनुसार SQL कोड से तालिका (table), attribute, relationship परिभाषित करे।

b. फ्रंट-एंड (Frontend): आवश्यकता के अनुसार एप्लिकेशन का फ्रंटएंड विकसित करें। एप्लिकेशन के फ्रंट-एंड में उपयोगकर्ता से इनपुटों की पहचान करें जिन्हें लिया जाना आवश्यक है। तय करें कि आपको किस प्रकार के बटन चाहिए उपयोगकर्ता के टाइपिंग प्रयास को कम करने के लिए, जैसे रेडियो बटन, चेकबॉक्स, सूची और कॉम्बो बॉक्स उपयोगकर्ता को प्रदान किया जाएगा।

4. परीक्षण चरण (Testing Phase) : संपूर्ण एप्लिकेशन (फ्रंट-एंड और बैक-एंड) का परीक्षण करें । सभी एप्लिकेशन बग ढूंढें और उन्हें ठीक करें। अंतिम सत्यापन के बाद, एप्लिकेशन रिलीज़ के लिए तैयार है।

2.8 केस स्टडी - ऑनलाइन गेम (Case Study - Online Game)

परियोजना का नाम: Mythical Battles (एक काल्पनिक ऑनलाइन मल्टीप्लेयर खेल)

उद्देश्य:

इस केस स्टडी का उद्देश्य एक ऑनलाइन मल्टीप्लेयर गेम Mythical Battles के विकास के चरणों और उसमें सामने आने वाली चुनौतियों को समझना है। इस गेम का उद्देश्य खिलाड़ियों को एक काल्पनिक दुनिया में विभिन्न मिथकीय प्राणियों और पात्रों के साथ लड़ने का अनुभव देना था। गेम में खिलाड़ियों को व्यक्तिगत रूप से या एक समूह के रूप में विभिन्न मिशनों को पूरा करना होता है, जिससे वे पुरस्कार और अंक कमा सकते हैं।

1. परियोजना का उद्देश्य और आवश्यकता (Project Objective and Requirements)

उद्देश्य:

Mythical Battles एक एडवेंचर-आधारित ऑनलाइन मल्टीप्लेयर गेम था, जिसमें खिलाड़ियों को विविध प्रकार के मिथकीय प्राणियों (जैसे ड्रेगन, गॉब्लिन्स, टाइटन्स) और शक्तियों का सामना करना होता है। इसका उद्देश्य उपयोगकर्ताओं को रोमांचक और चुनौतीपूर्ण गेमप्ले के अनुभव प्रदान करना था, साथ ही एक सामूहिक प्रतियोगिता और सहयोग का अवसर भी मिलता है।

आवश्यकताएँ:

* प्लेटफॉर्मः वेब और मोबाइल (iOS और Android)
* गति और प्रदर्शनः लोडिंग समय कम, 60 फ्रेम प्रति सेकंड (FPS)
* प्लेटफ़ॉर्म की संगतताः गेम को विभिन्न प्लेटफार्म्स (ब्राउज़र, एंड्रॉयड, iOS) पर उपलब्ध कराना
* सुरक्षाः खिलाड़ियों की व्यक्तिगत जानकारी और गेम डेटा को सुरक्षित रखना
* ग्राफिक्सः उच्च गुणवत्ता वाले 3D ग्राफिक्स और एनीमेशन

2. डिज़ाइन और योजना (Design and Planning)

यूज़र इंटरफेस डिज़ाइन (UI Design):

* गेम का इंटरफेस उपयोगकर्ता के अनुभव को सरल और आकर्षक बनाने के लिए डिज़ाइन किया गया था।
* गेम स्क्रीन पर मुख्यतः एक मैप, नैविगेशन मेन्यू, संपत्ति सूची, चैट बॉक्स, और पावर-अप्स के आइकन होते थे।
* प्रत्येक खिलाड़ी को एक पोर्ट्रेट (Avatar) मिलता था, जिससे उसे खेल के दौरान पहचाना जा सकता था।

यूज़र एक्सपीरियंस डिज़ाइन (UX Design):

* यूज़र अनुभव में गेम के लिए तेज़ और सरल इंटरएक्शन की आवश्यकता थी। इसके लिए फ्री-फ्लो नेविगेशन और स्पीड-अप फिचर्स को लागू किया गया।
* गेम में खिलाड़ी को जितना संभव हो सके कम से कम समय में चुनौतीपूर्ण और मजेदार अनुभव मिल सके, इसके लिए डिज़ाइन की गई थी।

वायरफ्रेम और प्रोटोटाइपः

* सबसे पहले, एक वायरफ्रेम (wireframe) और प्रोटोटाइप तैयार किया गया ताकि गेम की संरचना और इंटरफ़ेस का पहले से ही मूल्यांकन किया जा सके। इसमें गेम के विभिन्न भाग जैसे कि चरणों का चयन, आक्रमण मोड, सामूहिक मिशन आदि शामिल थे।

SCERT DELHI

3. विकास (Development)

फ्रंट-एंड डेवलपमेंट:

* गेम के फ्रंट-एंड को HTML5, CSS3, और JavaScript के साथ विकसित किया गया था।
* Three.js का उपयोग 3D ग्राफिक्स और एनीमेशन के लिए किया गया था, ताकि गेम के पात्रों और पर्यावरण को वास्तविक रूप से प्रस्तुत किया जा सके।
* React Native और Flutter का उपयोग मोबाइल ऐप्स (Android & iOS) के लिए किया गया।

बैक-एंड डेवलपमेंट:

* Node.js और Express.js का उपयोग सर्वर-साइड विकास के लिए किया गया।
* WebSocket का उपयोग किया गया ताकि खेल में वास्तविक समय के मल्टीप्लेयर इंटरएक्शन को संभाला जा सके।
* MongoDB का उपयोग खिलाड़ियों के प्रोफाइल और आँकड़े संग्रहीत करने के लिए किया गया।

सर्वर और होस्टिंग:

* AWS (Amazon Web Services) का इस्तेमाल सर्वर होस्टिंग के लिए किया गया ताकि गेम की प्रदर्शन क्षमता बेहतर हो सके और अधिक खिलाड़ियों को एक साथ संभाला जा सके।
* Cloudflare के माध्यम से गेम को DDoS हमलों से सुरक्षा प्रदान की गई।

4. परीक्षण (Testing)

फंक्शनल टेस्टिंग:

* खेल की सभी प्रमुख कार्यक्षमताओं का परीक्षण किया गया, जैसे कि खिलाड़ी का पंजीकरण, लॉगिन, मैच बनाना, और अन्य खिलाड़ियों के साथ खेलना।
* गेमप्ले की पावर-अप्स, आक्रमण मोड्स, और चैलेंज राउंड्स की भी अच्छी तरह से जांच की गई।

सुरक्षा परीक्षण:

* खिलाड़ी की व्यक्तिगत जानकारी और वित्तीय डेटा को सुरक्षित रखने के लिए SSL एन्क्रिप्शन

SCERT DELHI

और पासवर्ड हैशिंग जैसी तकनीकों का परीक्षण किया गया।

* सुरक्षा परीक्षणों में यह सुनिश्चित किया गया कि खिलाड़ियों के खातों में अनधिकृत प्रवेश या धोखाधड़ी का प्रयास न हो।

प्रदर्शन परीक्षण:

* लोड टेस्टिंग और स्पीड टेस्टिंग की गई ताकि यह सुनिश्चित किया जा सके कि सर्वर उच्च ट्रैफिक के दौरान भी ठीक से काम करता है।

* फ्रेम रेट (60 FPS) और लोड टाइम (3 सेकंड से कम) को टेस्ट किया गया ताकि गेम तेज़ और प्रतिक्रियाशील हो।

5. डिप्लॉयमेंट (Deployment)

* परीक्षण के बाद, गेम को सर्वर पर AWS के माध्यम से डिप्लॉय किया गया और इसे सार्वजनिक रूप से लॉन्च किया गया।

* इसके बाद, गेम को Google Play Store और Apple App Store पर प्रकाशित किया गया ताकि इसे स्मार्टफोन उपयोगकर्ता डाउनलोड कर सकें।

* लॉन्च के दौरान, गेम में उपलब्धता की निगरानी (availability monitoring) और प्रदर्शन की निगरानी (performance monitoring) की गई ताकि किसी भी समस्या को जल्दी से हल किया जा सके।

6. रखरखाव और सुधार (Maintenance and Improvements)

* डिप्लॉयमेंट के बाद, लगातार बग फिक्सिंग और अपडेट्स किए गए। गेम के संतोषजनक संचालन के लिए नई फीचर्स जोड़ी गईं।

2.9 केस स्टडी - ऑनलाइन क्विज़(Case Study - Online Quiz)

समस्या कथन (problem statement): ऑनलाइन क्विज़ वेब एप्लिकेशन कई छात्रों को इसकी अनुमति देता है एक बार में प्रश्नोत्तरी लें, परीक्षा समाप्त होते ही परिणाम प्रदर्शित करेगा। परिणाम खुद ब खुद उत्पन्न होता है। पहली बार उपयोगकर्ता को लॉगिन और पासवर्ड प्राप्त करने के लिए पंजीकरण करना होगा। तब उपयोगकर्ता परीक्षा देने और परिणाम देखने के लिए भी इस लॉगिन और पासवर्ड का उपयोग करता है। एक बार लॉग इन करने के बाद, उपयोगकर्ता उस श्रेणी का चयन कर सकता है जिसके लिए प्रश्नोत्तरी लेनी है। जब प्रश्नोत्तरी पूरी हो जाती है, तो सही उत्तरों के आधार

SCERT DELHI

पर एक रिपोर्ट तैयार की जाती है। व्यवस्थापक (administrator) के पास प्रश्न बनाने,प्रश्नोत्तरी, संशोधित करने और हटाने का विशेषाधिकार है।

1. परियोजना का उद्देश्य और आवश्यकता (Project Objective and Requirements)

a) **व्यवहार्य (Feasibility):** छात्रों को समूहों में चर्चा करने और पहचानने के लिए प्रोत्साहित किया जाता है इस परियोजना के लिए तकनीकी आवश्यकताएँ और मूल्यांकन करें कि क्या यह संभव है इसे स्कूल में उपलब्ध संसाधनों के दायरे में लागू करें। निम्नलिखित कदम इस धारणा पर आधारित हैं कि परियोजना व्यवहार्य है।

b) दायरा(scope): एप्लिकेशन को निम्नलिखित कार्य करना होगा-

(i) उपयोगकर्ता को पंजीकरण करने की अनुमति दें।

(ii) उपयोगकर्ता को लॉगिन और पासवर्ड प्रदान करें।

(iii) उपयोगकर्ता को प्रश्नोत्तरी की एक श्रेणी चुनने की अनुमति दें।

(iv) किसी वैध उपयोगकर्ता को चयनित श्रेणी के लिए प्रश्नोत्तरी प्रश्न प्रदर्शित करें।

(v) प्रश्नोत्तरी का परिणाम प्रदान करने के लिए एक रिपोर्ट तैयार करें।

(vi) व्यवस्थापक(administrator) को प्रश्न बनाने, संशोधित करने और हटाने की अनुमति दें।

2. डिज़ाइन और योजना (Design and Planning)

a) एप्लिकेशन साइट का मानचित्रः छात्रों को डिज़ाइन करने के लिए प्रोत्साहित किया जाता है

इस साइट के पृष्ठों की पहचान करके सेट की संरचना।

b) डेटाबेस (Database): डेटाबेस में निम्नलिखित टेबल्स(Tables)बनाने का निर्णय लिया गया है तालिकाओं की विशेषताएँ, उनके डेटा प्रकार(data type) , टेबल के एट्रिब्यूट (attribute):

* Admi
* Quiz_info
* User_info
* Result

SCERT DELHI

Table: Admi

S. No.	Name	Data Type	Remarks
1	Username	Varchar (30)	व्यवस्थापक(Administrator) लॉगिन नाम (प्राथमिक कुंजी)
2	Password	Varchar (30)	व्यवस्थापक पासवर्ड

Table: Quiz_info

S.No.	Name	Data Type	Remarks
1	Question_number	Varchar(30)	प्रश्न क्रमांक
2	Subject	Varchar(30)	विषय का नाम जैसे हिंदी, इंग्लिश आदि ।
3	Question	Varchar(30)	प्रश्नोत्तरी प्रशन
4	Option 1	Varchar(30)	1st प्रश्नोत्तरी उत्तर का विकल्प
5	Option 2	Varchar(30)	2nd प्रश्नोत्तरी उत्तर का विकल्प
6	Option 3	Varchar(30)	3rd प्रश्नोत्तरी उत्तर का विकल्प
7	Option 4	Varchar(30)	4th प्रश्नोत्तरी उत्तर का विकल्प
8	Result	Varchar(30)	सहीं जवाब 1-4 में से

Table: User_info

S. No.	Name	Data Type	Remarks
1	User Name	Varchar(30)	उपयोगकर्ता का नाम लॉग इन करने के लिए
2	Password	Varchar(30)	उपयोगकर्ता का पासवर्ड लॉग इन करने के लिए
3	Name	Varchar(30)	उपयोगकर्ता का नाम
4	Email Id	Varchar(30)	उपयोगकर्ता का ईमेल
5	Gender	Varchar(30)	उपयोगकर्ता का लिंग
6	DOB	Varchar(30)	उपयोगकर्ता की जन्म की तारिक

Table: Result

S. No.	Name	Data Type	Remarks

1	Username	Varchar(30)	लॉगिन प्रयोजन के लिए उपयोगकर्ता का नाम
2	Marks	Number(30)	उपयोगकर्ता द्वारा प्राप्त अंक
3	Category	Varchar(30)	प्रश्नोत्तरी श्रेणी जिसे उपयोगकर्ता ने प्रयास किया

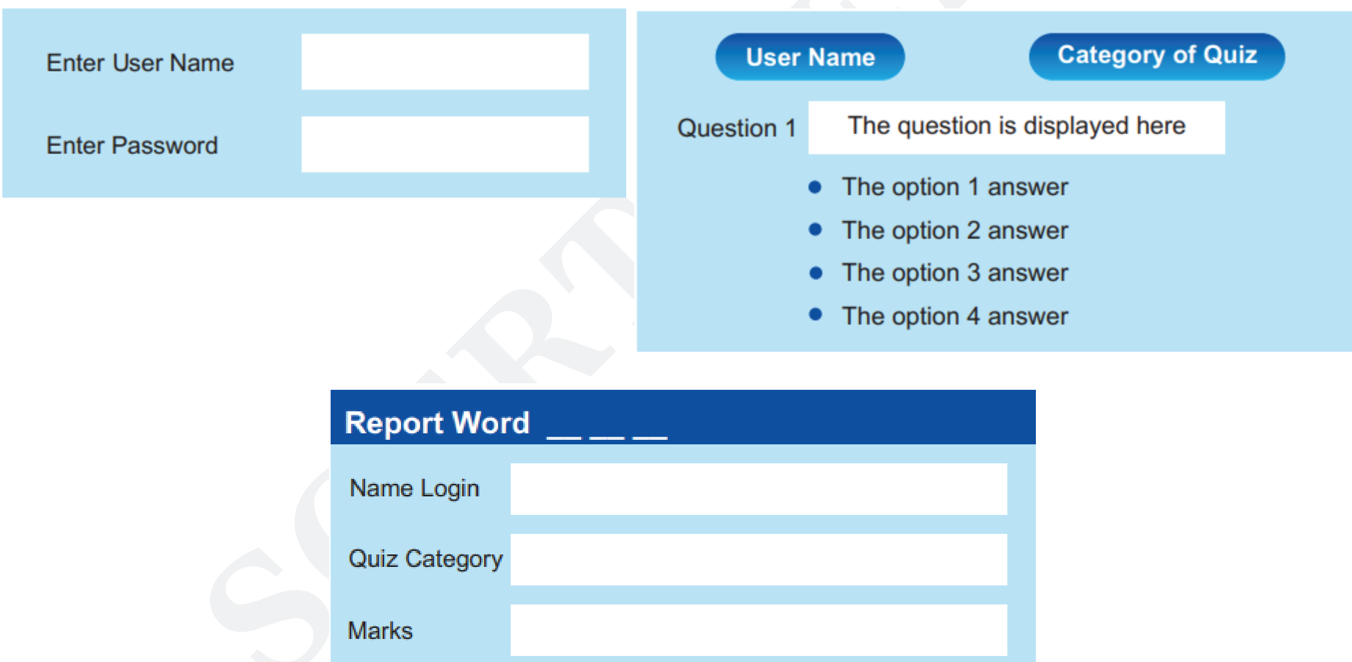
c) **पृष्ठ संरचना:** पृष्ठ की संरचना डिज़ाइन करें यह पाया गया कि उपयोगकर्ता से आवश्यक इनपुट इस प्रकार हैं:

उपयोगकर्ता नाम और पासवर्ड (username and password)

उस श्रेणी का चयन जिसके लिए प्रश्नोत्तरी का प्रयास किया जाना है

प्रश्नोत्तरी उपयोगकर्ता को प्रदर्शित की जाती है। उपयोगकर्ता प्रत्येक सवाल के लिए उत्तर का चयन करता है।

निम्नलिखित फ्रंट-एंड इंटरफ़ेस डिज़ाइन किया जा सकता है



चित्र 2.31 प्रोजेक्ट का फ्रंट-एंड डिज़ाइन इंटरफ़ेस

* **कार्यान्वयन (Implementation):** बैकएंड डेटाबेस, फ्रंटएंड और इनके बीच कनेक्टिविटी बनाएं।

* **परीक्षा (Test):** संपूर्ण एप्लिकेशन लागू करने के बाद, एप्लिकेशन का परीक्षण करें।

यादृच्छिक(random) डेटा(data). एप्लिकेशन की प्रत्येक सुविधा और कार्यक्षमता का परीक्षण करें। यदि बग पाया तो उससे ठीक करें, और पुनः परीक्षण किया।

2.10 केस स्टडी - ऑनलाइन बिल कैलकुलेटर (Case Study – Online Bill Calculator)

पृष्ठभूमिः ऑनलाइन बिल कैलकुलेटर एक डिजिटल टूल है जो उपभोक्ताओं को उनके उत्पाद या सेवाओं के लिए बिलों का सटीक आकलन करने में मदद करता है। यह कैलकुलेटर आमतौर पर व्यवसायों, स्टोरों, और सेवा प्रदाताओं द्वारा वेबसाइटों पर उपलब्ध कराया जाता है ताकि ग्राहक आसानी से अपने खरीदारी के खर्च की गणना कर सकें, टैक्स और डिस्काउंट सहित।

उद्देश्यः इस केस स्टडी का उद्देश्य यह समझना है कि ऑनलाइन बिल कैलकुलेटर कैसे काम करता है, इसके लाभ और समस्याएँ, और इसे कैसे डिजाइन किया जा सकता है ताकि यह उपभोक्ताओं के लिए एक उपयोगी और आकर्षक उपकरण बने।

1. परियोजना का उद्देश्य और आवश्यकता (Project Objective and Requirements)

a) **व्यवहार्य (Feasibility):** छात्रों को समूहों में चर्चा करने और पहचानने के लिए प्रोत्साहित किया जाता है इस परियोजना के लिए तकनीकी आवश्यकताएँ और मूल्यांकन करें कि क्या यह संभव है इसे स्कूल में उपलब्ध संसाधनों के दायरे में लागू करें। निम्नलिखित क द म इस धारणा पर आधारित हैं कि परियोजना व्यवहार्य है।

b) **दायरा (scope):** एप्लिकेशन को निम्नलिखित कार्य करना होगा-

(i) उपयोगकर्ता को पंजीकरण करने की अनुमति दें।

(ii) उपयोगकर्ता को लॉगिन और पासवर्ड प्रदान करें।

(iii) उपयोगकर्ता जानकारी दर्ज करने की अनुमति दें।

(iv) बिल संबंधी जानकारी दर्ज करने की अनुमति दें।

(v) जनरेट किया गया बिल प्रदर्शित करें।

(vi) व्यवस्थापक(administrator) को प्रश्न बनाने, संशोधित करने और हटाने की अनुमति दें।

2. डिज़ाइन और योजना (Design and Planning)

a) एप्लिकेशन साइट का मानचित्रः छात्रों को डिज़ाइन करने के लिए प्रोत्साहित किया जाता है। इस साइट के पृष्ठों की पहचान करके सेट की संरचना।

b) डेटाबेस (Database): डेटाबेस में निम्नलिखित टेबल्स(Tables)बनाने का निर्णय लिया गया

है तालिकाओं की विशेषताएँ, उनके डेटा प्रकार(data type),टेबल के एट्रिब्यूट (attribute):

- User_info
- Bill_info

Table: User_info

S. No.	Name	Data Type	Remarks
1	User Name	Varchar(30)	उपयोगकर्ता का नाम लॉग इन करने के लिए
2	Password	Varchar(30)	उपयोगकर्ता का पासवर्ड लॉग इन करने के लिए
3	Name	Varchar(30)	उपयोगकर्ता का नाम
4	Email Id	Varchar(30)	उपयोगकर्ता का ईमेल
5	City	Varchar(30)	उपयोगकर्ता का शहर
6	State	Varchar(30)	उपयोगकर्ता का राज्य
7	Country	Varchar(30)	उपयोगकर्ता का देश
8	Pin code	Number(15)	उपयोगकर्ता का पिनकोड
9	Mobile Number	Number(15)	उपयोगकर्ता का मोबाइल नंबर

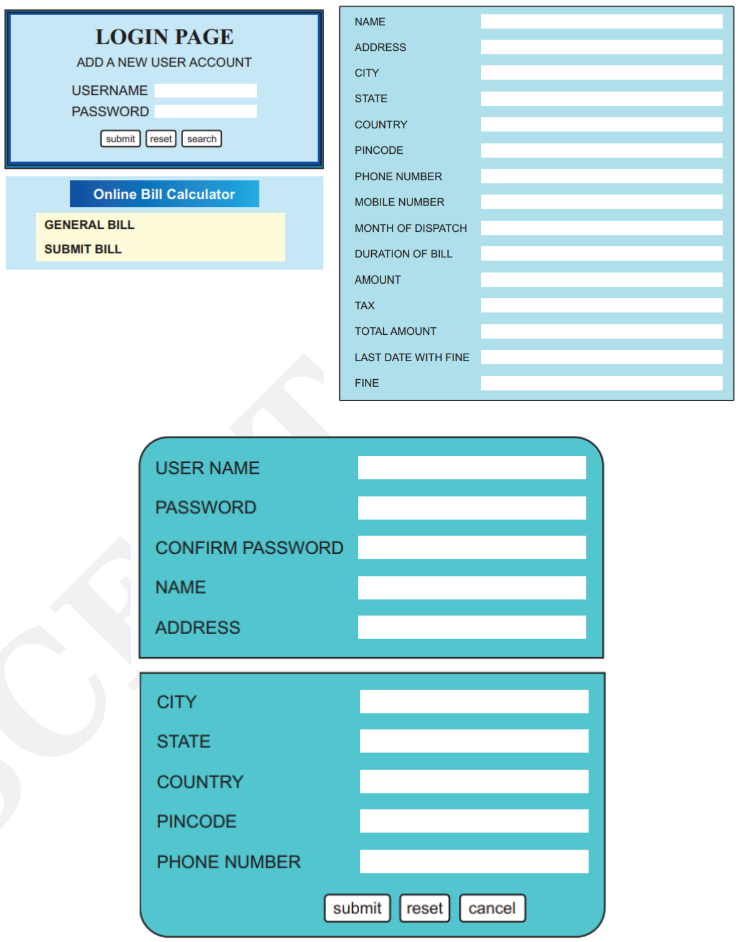
Table: Bill_info

S. No.	Name	Data Type	Remarks
1	User Name	Varchar(30)	उपयोगकर्ता का नाम लॉग इन करने के लिए
2	Name	Varchar(30)	उपयोगकर्ता का नाम
3	Address	Varchar(100)	उपयोगकर्ता का पता
4	City	Varchar(30)	उपयोगकर्ता का शहर
5	State	Varchar(30)	उपयोगकर्ता का राज्य
6	Country	Varchar(30)	उपयोगकर्ता का देश
7	Pin code	Number(15)	उपयोगकर्ता का पिनकोड
8	Month_dispatch	Number(15)	सामग्री प्रेषण का महीना
9	Duration of bill	Number(15)	बिल की अवधि दिनों में
10	Mobile Number	Number(15)	उपयोगकर्ता का मोबाइल नंबर

c) पृष्ठ संरचना: पृष्ठ की संरचना डिज़ाइन करें यह पाया गया कि उपयोगकर्ता से आवश्यक इनपुट इस प्रकार हैं:

* उपयोगकर्ता नाम और पासवर्ड (username and password)
* इंटरफ़ेसबिल बनाने या जमा करने के विकल्प का चयन करने के लिए
* इंटरफ़ेस उपयोगकर्ता जोड़ने के लिए
* इंटरफ़ेस बिल जानकारी जोड़ने के लिए

निम्नलिखित फ्रंट-एंड इंटरफ़ेस डिज़ाइन किया जा सकता है।



चित्र 2.31 प्रोजेक्ट का फ्रंट-एंड डिज़ाइन इंटरफ़ेस

* **कार्यान्वयन (Implementation):** बैकएंड डेटाबेस, फ्रंटएंड और इनके बीच कनेक्टिविटी बनाएं।
* **परीक्षा (Test):** संपूर्ण एप्लिकेशन लागू करने के बाद, एप्लिकेशन का परीक्षण करें।

यादृच्छिक(random) डेटा(data). एप्लिकेशन की प्रत्येक सुविधा और कार्यक्षमता का परीक्षण करें। यदि बग पाया तो उससे ठीक करें, और पुनः परीक्षण किया।

अभ्यास प्रश्न :

1. ऑनलाइन बिल कैलकुलेटर का उद्देश्य क्या था?
2. कार्यान्वयन (Implementation) क्या है?
3. परीक्षा(Test) क्यों जरुरी है?
4. वेब आधारित एप्लिकेशन प्रोजेक्ट की प्रमुख विशेषताएँ क्या है ।

महत्वपूर्ण प्रश्नः -

A. बहुविकल्पीय प्रश्न

1. ऑनलाइन शॉपिंग के लिए निम्नलिखित में से कौन सा सत्य नहीं है?

a) ग्राहक विवरण सहित प्रदर्शित किए गए सामान को देख सकता है।

b) एक बार सामान खरीदने के बाद उसे वापस नहीं किया जा सकता या बदला नहीं जा सकता।

c) फिर सामान ग्राहक द्वारा निर्दिष्ट पते पर वितरित किया जाता है।

d) ग्राहक खरीदे जाने वाले सामान का चयन कर सकता है और उन्हें अपने ऑनलाइन शॉपिंग टोकरी मे स्टोर कर सकता है।

2. निम्नलिखित में से कौन सी ICT-सक्षम सेवाएँ भारतीय सरकार द्वारा स्थापित नहीं की गई हैं?

a) रेलवे टिकटों की बुकिंग,

b) जन्म/मृत्यु प्रमाण पत्र का पंजीकरण

c) मूवी टिकटों की खरीद

d) आरटीआई आवेदन जमा करना

e) जन्म प्रमाण पत्र का पंजीकरण

SCERT DELHI

3. निम्नलिखित में से कौन सी एक परियोजना की विशेषता नहीं है?

a) परियोजना की शुरुआत और अंत है।

b) परियोजना की सीमा परिभाषित है।

c) परियोजना की कोई सीमा नहीं है।

d) परियोजना के लिए सीमित संसाधनों की आवश्यकता है

4. यह वेबसाइट दुनिया भर में विभिन्न विश्वविद्यालय के सहयोग से निःशुल्क उच्च गुणवत्ता वाले ऑनलाइन पाठ्यक्रम प्रदान करती है। यह वेबसाइट कौन सी है?

a) कोउर्सेरा (coursera.org)　　b) खान अकादमी

c) w3schools.org　　d) सीबीएसई (cbse.nic.in)

5. अनिल ऑनलाइन टिकटिंग वेबसाइट makemytrip.com का उपयोग करके रेलवे टिकट बुक कर रही है। जब वह ट्रेन पकड़ने पहुंच तो उसे पता चला कि वह टिकट अपने होटल में ही भूल गई है। वह क्या करे?

a) वह ट्रेन में नहीं चढ़ सकती

b) उसे स्टेशन पर एक नया टिकट बुक करना चाहिए

c) उसे वापस जाना चाहिए और टिकट लेना चाहिए

d) वह वेबसाइट से ई-टिकट डाउनलोड कर सकती है।

6. निम्नलिखित में से कौन सा वेब आधारित अनुप्रयोगों का नुकसान है:

a) उन्हें स्थापित करने की आवश्यकता नहीं है.

b) वे 24X7 उपलब्ध हैं।

c) वे सॉफ़्टवेयर आधारित अनुप्रयोगों की तुलना में धीमे हैं।

d) वे क्रॉस प्लेटफ़ॉर्म अनुकूलता प्रदान करते हैं।

7. DEITY का अर्थ है:

a) Department of Electrical and Information Technology

b) Department of Electronics and Information Technology

c) Department of Electronics and Informatics Technology

d) Department of Electrical and Informatics Technology

SCERT DELHI

B. लघु उत्तरीय प्रश्न

1. ई-गवर्नेंस के लिए किसी एक केंद्रीय पहल का नाम बताइए।
2. ऑनलाइन शॉपिंग के दौरान भुगतान के किन्हीं दो तरीकों का उल्लेख करें।
3. भारत के राष्ट्रीय पोर्टल का नाम बताएं?
4. प्रोजेक्ट क्या है?
5. ई-कॉमर्स एप्लिकेशन क्या है ? किन्ही तीन का उदहारण लिखे।
6. किन्हीं दो शैक्षिक वेबसाइटों के नाम बताइए जो मुफ़्त में ऑनलाइन पाठ्यक्रम प्रदान करती हैं।

C. दीर्घ उत्तरीय प्रश्न

1. अमन ऑनलाइन जूते खरीदना चाहता है। अपनाए जाने वाले विस्तृत चरण/प्रक्रिया लिखें।
2. ऑनलाइन शिक्षण से छात्रों को कैसे लाभ होता है?
3. ऑनलाइन शॉपिंग करते समय किन बातों का ख्याल रखना चाहिए। किन्ही तीन को लिखे।
4. वेब एप्लिकेशन परियोजना के चरणों को विस्तृत रूप में लिखे।
5. अमित एक व्यावसायिक यात्रा पर जाना चाहते हैं। वह एक फ्लाइट बुक करने के बारे में सोच रहे हैं।

a) किसी एक साइट का नाम बताइए जिसे उसे हवाई टिकट बुक करने के लिए ब्राउज़ करना चाहिए।

b) अमित को ऑनलाइन आरक्षण का कोई एक लाभ बताएं?

c) ऑनलाइन प्रदर्शन करते समय किन सावधानियों का पालन किया जाना चाहिए

d) ऑनलाइन आरक्षण प्रदान करने वाली कंपनी को कोई एक लाभ बताएं?

6. 'इंटरनेट लेनदेन धोखाधड़ी के प्रति संवेदनशील हैं'। क्या यह कथन सही है यदि हां तो आप इन धोखाधड़ी से कैसे बच सकते है किन्हें तीन सुझाव लिखे ।

SCERT DELHI

JAVA प्रोग्रामिंग के मूल सिद्धांत, जावा (Java) का परिचय

इकाई 3

सीखने के परिणाम

* जावा को इनस्टॉल करना और NetBeans IDe पर नया प्रोजेक्ट बनाना सीखे। ।
* वेरिएबल्स की आवश्यकता एवं उपयोग को समझना।
* विभिन्न डेटा प्रकारों (DataTypes) और प्रत्येक डेटा प्रकार के उद्देश्य को समझना।
* ऑपरेटरों के उपयोग को समझने के लिए (Arithmetic, Assignment, Comparison, Bitwise Operator) ।
* नियंत्रण (Control Statements) जैसे (if, if else, switch case) कथनों को समझना
* पुनरावृत्ति (Repetition) कथनों (Statements) जैसे (while loop, do while loop, for loop) को समझना।
* Array को समझना।
* यूजर डिफाइन मेथड्स को समझना।
* ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग का परिचय।
* जावा (Java) लाइब्रेरी को समझना।
* एक्सेप्शन (Exception) हैंडलिंग को समझना।
* डेटाबेस कनेक्टिविटी को समझना।
* Assertions, Threads, and Wrapper Classes को समझना।

परिचय (Introduction)

जावा (Java) एक सरल, सुरक्षित, और उपयोगी प्रोग्रामिंग भाषा(Language) है, जिसे 1995 में सन माइक्रोसिस्टम्स ने बनाया। इसकी सबसे खास बात यह है कि जावा(Java) का कोड एक बार लिखने के बाद किसी भी ऑपरेटिंग सिस्टम जैसे Windows, Mac, या Linux पर आसानी से चल सकता है। इसलिए इसे 'प्लेटफॉर्म स्वतंत्र (independent)' भाषा भी कहा जाता है।

जावा में प्रोग्राम लिखना इसलिए भी आसान है क्योंकि यह ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग (OOP) पर आधारित है। इसका मतलब है कि जावा में प्रोग्रामिंग करना वैसा ही है जैसे किसी चीज़ को टुकड़ों में बांट कर बनाना, जिससे कोड को समझना और बाद में बदलना आसान हो जाता है।

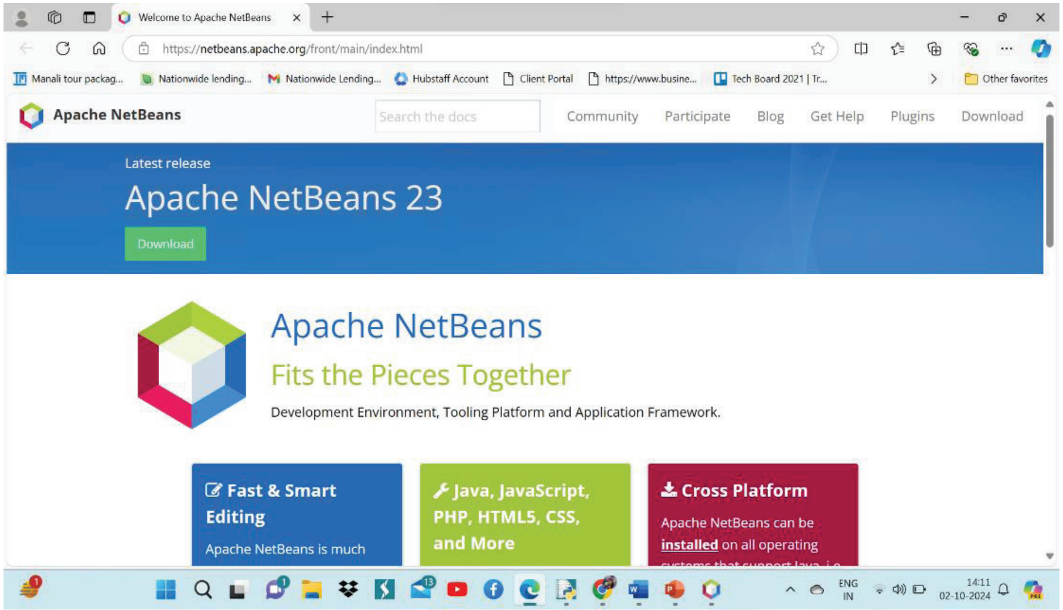
जावा का इस्तेमाल केवल पढ़ाई तक सीमित नहीं है – इससे वेबसाइट्स, मोबाइल एप्लिकेशंस, गेम्स, और यहां तक कि बैंकिंग सिस्टम भी बनाए जाते हैं। इसलिए, जावा सीखना न सिर्फ़ मजेदार है बल्कि भविष्य के लिए उपयोगी भी है।

चित्र 3.1 Java Programming परिचय(Introduction)

3.1 NetBeans इंस्टाल करने का तरीका:

* **डाउनलोड:** सबसे पहले NetBeans की आधिकारिक वेबसाइट पर जाकर अपने ऑपरेटिंग सिस्टम के अनुसार NetBeans का नवीनतम संस्करण डाउनलोड करें।

* **JDK आवश्यक है:** Java डेवलपमेंट के लिए आपको Java Development Kit (JDK) की आवश्यकता होती है। अगर आपके सिस्टम में JDK इंस्टॉल नहीं है, तो इसे पहले इंस्टॉल करें।



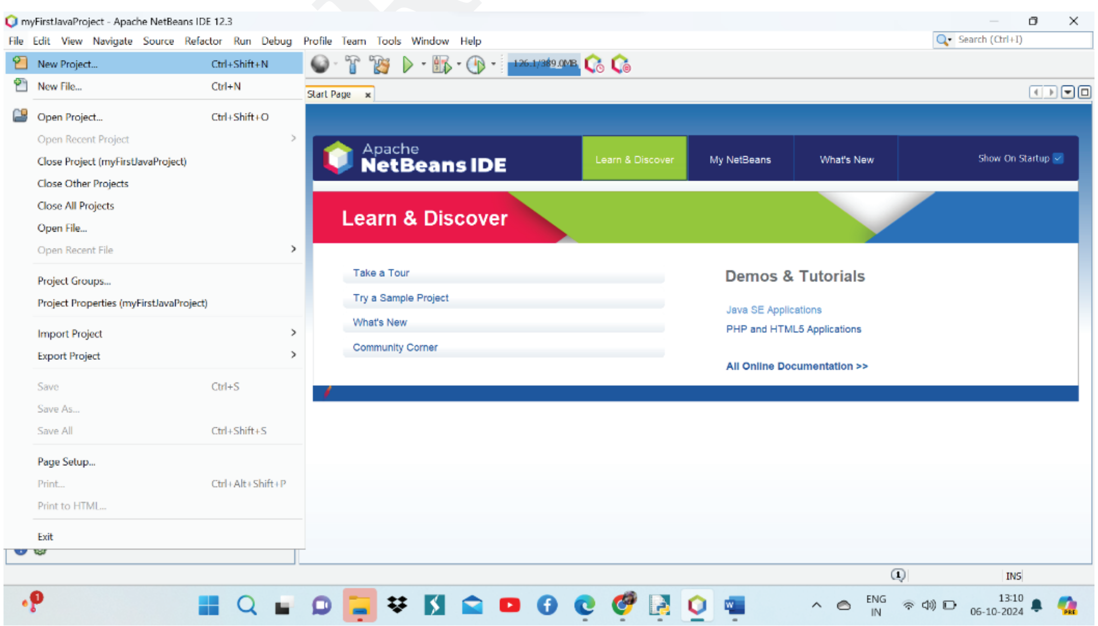
चित्र 3.2 NetBeans इंस्टाल

* **इंस्टालेशन प्रोसेसः** NetBeans को डाउनलोड करने के बाद इंस्टालर को चलाएं और दिए गए स्टेप्स को फॉलो करें।

3.2 NetBeans का उपयोग Java डेवलपमेंट के लिएः

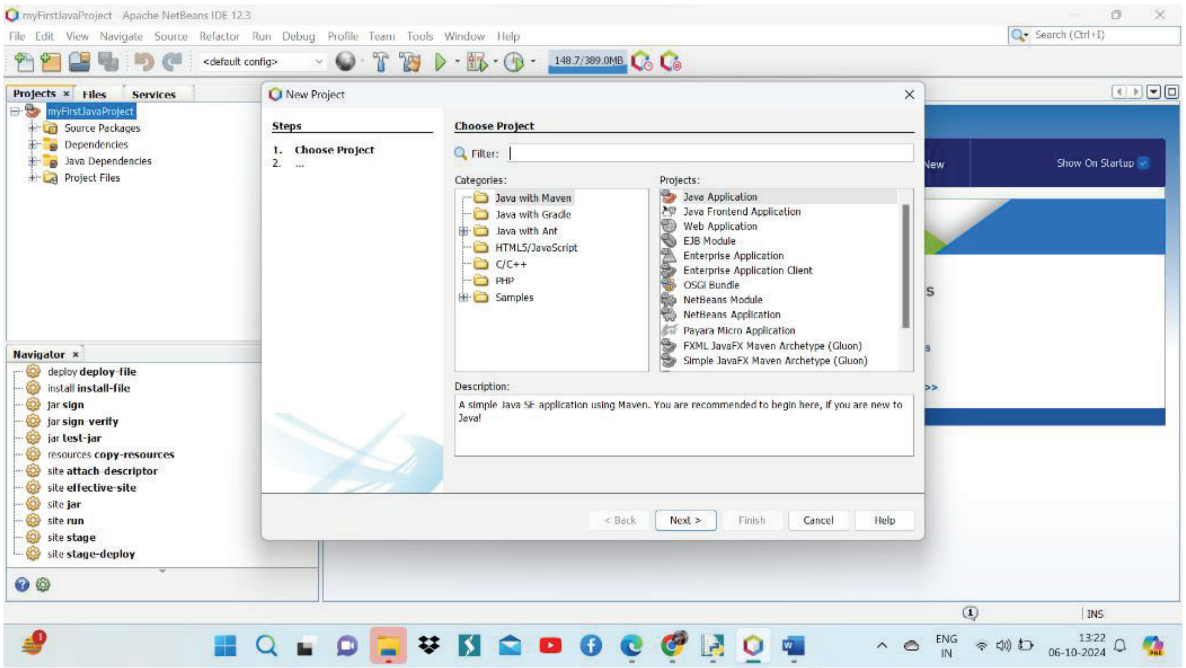
1. नया प्रोजेक्ट बनानाः

   * NetBeans खोलें और File > New Project विकल्प चुनें।



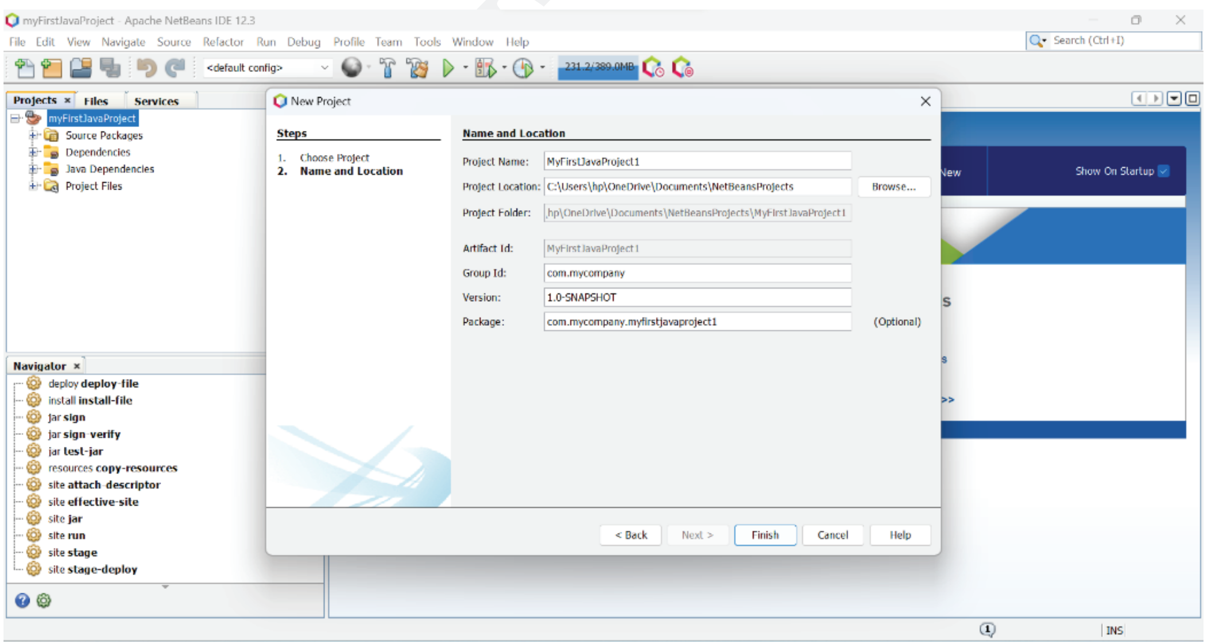
चित्र 3.3 नया प्रोजेक्ट बनाना

* Java Application टेम्पलेट को चुनें।

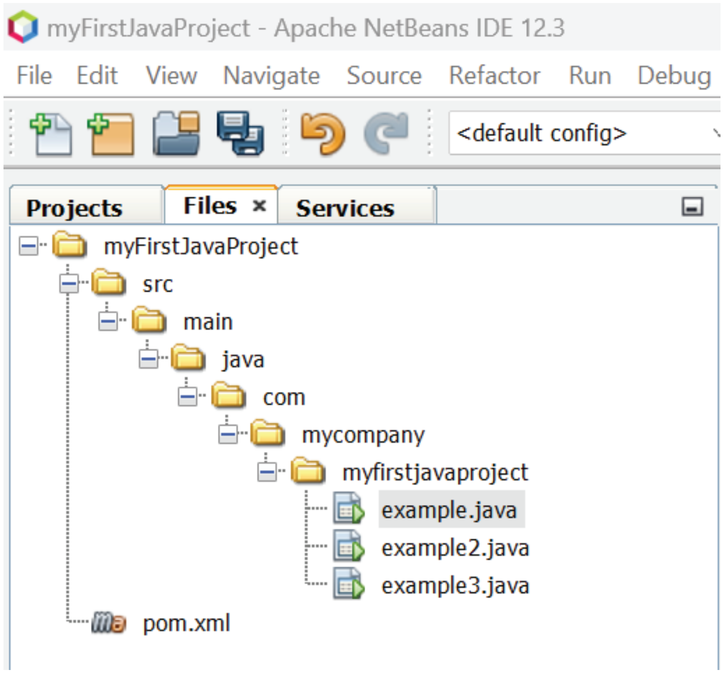


चित्र 3.4 Java Application टेम्पलेट

* प्रोजेक्ट का नाम और लोकेशन निर्धारित करें।



चित्र 3.5 प्रोजेक्ट का नाम और लोकेशन



चित्र 3.6 जावा प्रोजेक्ट का अनुक्रम

आइए किसी प्रोजेक्ट, फॉर्म और घटकों के बीच संबंध को दोबारा समझें। प्रत्येक एप्लिकेशन को इस प्रकार माना जाता है, नेटबीन्स में एक प्रोजेक्ट और प्रत्येक प्रोजेक्ट के एक या एकाधिक रूप हो सकते हैं और यह तथ्य स्पष्ट है प्रोजेक्ट विंडो जैसा कि चित्र 4.4 में दिखाया गया है।

* कोड लिखना:
* NetBeans आपके लिए एक Main.java फाइल अपने आप बना देगा।
* आप इस फाइल में अपना Java कोड लिख सकते हैं, जिसमें आपको कोड कम्प्लीशन और एरर हाइलाइटिंग की सुविधा मिलेगी।

3.3 प्रोजेक्ट को बिल्ड और रन करना:

* जब आप कोड लिख लें, तो Build विकल्प पर क्लिक करके प्रोजेक्ट को बिल्ड कर सकते हैं।
* प्रोजेक्ट को रन करने के लिए Run बटन पर क्लिक करें या F6 दबाएं।
* वेरिएबल्स की आवश्यकता और उपयोग को समझना

Java में, variables का उपयोग डेटा को स्टोर और मैनेज करने के लिए किया जाता है। एक variable एक कंटेनर की तरह होता है, जो एक नाम से पहचाना जाता है और उसमें कोई वैल्यू (जैसे नंबर, कैरेक्टर, या टेक्स्ट) स्टोर की जाती है।

Variables की आवश्यकता:

* डेटा स्टोर करना: प्रोग्रामिंग में किसी भी डेटा को स्टोर करने के लिए variables की आवश्यकता होती है, ताकि हम बाद में उस डेटा का उपयोग कर सकें।
* वैल्यू में बदला व करना: variables का उपयोग करके हम वैल्यू को बदल सकते हैं, अपडेट कर सकते हैं, और विभिन्न गणनाओं में उपयोग कर सकते हैं।

SCERT DELHI

* प्रोग्राम की समझ बढ़ानाः variables को नाम देकर हम प्रोग्राम को अधिक रीडेबल और समझने योग्य बना सकते हैं। जैसे, अगर हमें किसी छात्र की आयु स्टोर करनी है, तो हम age नाम का variable बना सकते हैं।

Variables का उपयोगः

* Variable Declare करनाः पहले हमें variable को declare करना होता है, जैसे कि उसकी data type और नाम देनाः

उदाहरणः

```
int age;
```

* वैल्यू असाइन करनाः variable में वैल्यू असाइन करके हम उसे उपयोग के लिए तैयार करते हैंः

उदाहरणः

```
int age = 18,
```

* कैलकुलेशन में उपयोगः हम variables को गणनाओं में उपयोग कर सकते हैंः

उदाहरणः

```
in total =  age + 5;
```

वेरिएबल (Variable) के प्रकारः

1. स्थायी वेरिएबल (Local Variable):

* ये वेरिएबल्स किसी विशेष ब्लॉक या फंक्शन के भीतर परिभाषित होते हैं।
* इनका उपयोग केवल उस ब्लॉक के भीतर किया जा सकता है।
* जब ब्लॉक समाप्त होता है, तो यह वेरिएबल नष्ट हो जाता है।

इस तरह, variables प्रोग्रामिंग में महत्वपूर्ण भूमिका निभाते हैं क्योंकि वे डेटा को स्टोर, मैनेज, और प्रोसेस करने में मदद करते हैं।

```
void example( ) {
int localVar = 10; // यह स्थायी वेरिएबल है
System.out.println(localVar);
}
```

SCERT Delhi

2. ग्लोबल वेरिएबल (Global Variable):

* ये वेरिएबल्स क्लास के स्तर पर परिभाषित होते हैं और पूरे क्लास में उपलब्ध होते हैं।
* इनका उपयोग किसी भी विधि में किया जा सकता है।

```
class Example {
int globalVar = 20; // यह ग्लोबल वेरिएबल है
void display( ) {
System.out.println(globalVar);
}
}
```

गतिविधि (Activity)

carName नाम से एक वेरिएबल बनाएं और उसे Volvo वैल्यू असाइन करें।

3. विभिन्न डेटा प्रकार और प्रत्येक डेटा प्रकार का उद्देश्य समझना

डेटा टाइप	Description
String	String एक डेटा टाइप है जो टेक्स्ट डेटा, जैसे अक्षरों, शब्दों और वाक्यों का अनुक्रम (sequence), स्टोर करने के लिए उपयोग होता है।
Int	सामान्य integer के लिए। रेंजः -2,147,483,648 से 2,147,483,647।
Float	छोटे दशमलव संख्या के लिए। 19.99 or -19.99
Char	सिंगल कैरेक्टर स्टोर करता है। 'A','b'
Boolean	केवल true या false स्टोर करता है।

Java में डेटा टाइप्स होते हैं, जो साधारण और बेसिक डेटा को स्टोर करने के लिए उपयोग किए जाते हैं:

गतिविधि (Activity)

निम्नलिखित वेरिएबल के लिए सही डेटा प्रकार जोड़ें:

```
________ myNum = 9;
________ myFloatNum = 8.99f;
________ myLetter = 'A';
________ myBool = false;
________ myText = 'Hello World';
```

3.4 ऑपरेटरों का उपयोग समझना (असाइनमेंट, अंकगणितीय, रिश्तेदार, तार्किक, बिटवाइज)

1. Arithmetic Operators (अर्थमेटिक ऑपरेटर्स)

ये ऑपरेटर्स गणितीय ऑपरेशंस (जोड़, घटाव, गुणा, विभाजन) के लिए उपयोग होते हैं।

Operator	नाम	विवरण	उदाहरण
+	Addition (जोड़)	दो वैल्यू को एक साथ जोड़ता है	x + y
-	Subtraction (घटाव)	एक वैल्यू को दूसरे से घटाता है	x - y
*	Multiplication (गुणा)	दो वैल्यू को गुणा करता है	x * y
/	Division (विभाजन)	एक वैल्यू को दूसरे से विभाजित करता है	x / y
%	Modulus (शेष)	विभाजन शेष लौटाता है	x % y
++	Increment	किसी वेरिएबल का वैल्यू 1 से बढ़ा देता है	++x
--	Decrement	किसी वेरिएबल का वैल्यू 1 से कम कर देता है	--x

```
int sum = 5 + 3; // 8
int difference = 10 - 2; // 8
int product = 4 * 2; // 8
int quotient = 16 / 4; // 4
int remainder = 7 % 3; // 1
```

2. Assignment Operators (असाइनमेंट ऑपरेटर्स)

वैल्यू को एक variable में असाइन (assign) करने के लिए उपयोग किए जाते हैं।

Operator	नाम	विवरण	उदाहरण
=	असाइन Assign	वैल्यू को असाइन करता है।	x = y
+=	जोड़ कर असाइन	वैल्यू को जोड़ता है और असाइन करता है।	x +=y
-+	घटा कर असाइन	वैल्यू को घटाता है और असाइन करता है।	x -= y

Operator	नाम	विवरण	उदाहरण
*=	गुणा कर असाइन	वैल्यू को गुणा करता है और असाइन करता है।	x *=y
/=	भाग कर असाइन	वैल्यू को विभाजित करता है और असाइन करता है।	x /=y
%=	शेषफल (remainder) असाइन	वैल्यू का remainder (शेष) असाइन करता है।	X%=y

उदाहरणः

```
int x = 10;  // x में 10 असाइन हुआ
 x += 5;       // x = x + 5; (x अब 15 है)
 x -= 5  //x = x – 5; (x अब 5 है)
```

3. **Relational/Comaprision Operators (रिलेशनल ऑपरेटर्स)** ये ऑपरेटर्स दो वैल्यू की तुलना करने के लिए उपयोग होते हैं और बूलियन (true/false) रिजल्ट देते हैं।

Operator	नाम	विवरण	उदाहरण
==	समानता (Equal to)	यह ऑपरेटर जांचता है कि क्या दो मान समान हैं या नहीं। यदि दोनों मान समान हैं, तो परिणाम true होगा; अन्यथा false।	x == y
!=	असमानता (Not Equal to)	यह ऑपरेटर जांचता है कि क्या दो मान असमान हैं या नहीं। यदि दोनों मान अलग हैं, तो परिणाम true होगा; अन्यथा false।	x != y
>	Greater Than	यह ऑपरेटर जांचता है कि क्या बाएं पक्ष का मान दाएं पक्ष के मान से बड़ा है।	x > y
<	Less Than	यह ऑपरेटर जांचता है कि क्या बाएं पक्ष का मान दाएं पक्ष के मान से छोटा है।	x < y

Operator	नाम	विवरण	उदाहरण
>=	Greater Than Equal to	यह ऑपरेटर जांचता है कि क्या बाएं पक्ष का मान दाएं पक्ष के मान के बराबर या उससे बड़ा है।	x >= y
<=	Less Than Equal to	यह ऑपरेटर जांचता है कि क्या बाएं पक्ष का मान दाएं पक्ष के मान के बराबर या उससे छोटा है।	x <= y

उदाहरणः

```
int a = 5, b = 10;
boolean isEqual = (a == b); // false
boolean isGreater = (a > b); // false
boolean isLess = (a < b); // true
```

4. **Logical Operators (लॉजिकल ऑपरेटर्स)** ये ऑपरेटर्स लॉजिकल स्थितियों पर ऑपरेशन

Operator	नाम	विवरण	उदाहरण
&&	Logical AND	जब दोनों शर्तें सही हों, तभी true देता है।	x < 5 && x < 10
||	Logical OR	जब कोई भी एक शर्त सही हो, true देता है।	x < 5 || x < 4
!	Logical NOT	किसी बूलियन वैल्यू को उलट देता है (अगर true है तो false और अगर false है तो true)।	!(x < 5 && x < 10)

उदाहरण:
```
int a = 5, b = 10;
boolean result = (a < b) && (b > 5); // true (दोनों सही हैं)
boolean anotherResult = (a > b) || (a < 10); // true (एक सही है)
boolean notResult = !(a < b); // false (क्योंकि a < b सही है, NOT इसे
उलट देगा)
```

5. Bitwise Operators (बिटवाइज़ ऑपरेटर्स)

ये ऑपरेटर्स बिट-लेवल पर काम करते हैं और आमतौर पर बाइनरी नंबर पर लागू होते हैं।

Operator	नाम	विवरण	उदाहरण
&	Bitwise AND	AND ऑपरेटर (बिट लेवल पर दोनों बिट्स 1 होने पर 1 देता है)	x & y
|	Bitwise OR	OR ऑपरेटर (बिट लेवल पर कोई एक बिट 1 होने पर 1 देता है)	x | y
^	Bitwise XOR	XOR ऑपरेटर (जब बिट्स अलग होते हैं तो 1 देता है)	x ^ y
₹	Bitwise NOT	NOT ऑपरेटर (बिट्स को उलट देता है)	₹ x
<<	Left Shift	लेफ्ट शिफ्ट (बिट्स को बाईं ओर शिफ्ट करता है)	x << 1
>>	Right Shift	राइट शिफ्ट (बिट्स को दाईं ओर शिफ्ट करता है)	x >> 1

उदाहरण:
```
int x = 5; // बाइनरी: 0101
int y = 3; // बाइनरी: 0011
int andResult = x & y; // बाइनरी: 0001 (आउटपुट: 1)
int orResult = x | y; // बाइनरी: 0111 (आउटपुट: 7)
int xorResult = x ^ y; // बाइनरी: 0110 (आउटपुट: 6)
int notResult = ₹x; // बाइनरी: 1010 (आउटपुट: -6)
```

गतिविधि(Activity)

नीचे दिए गए ऑपरेटर्स के नाम बताये।

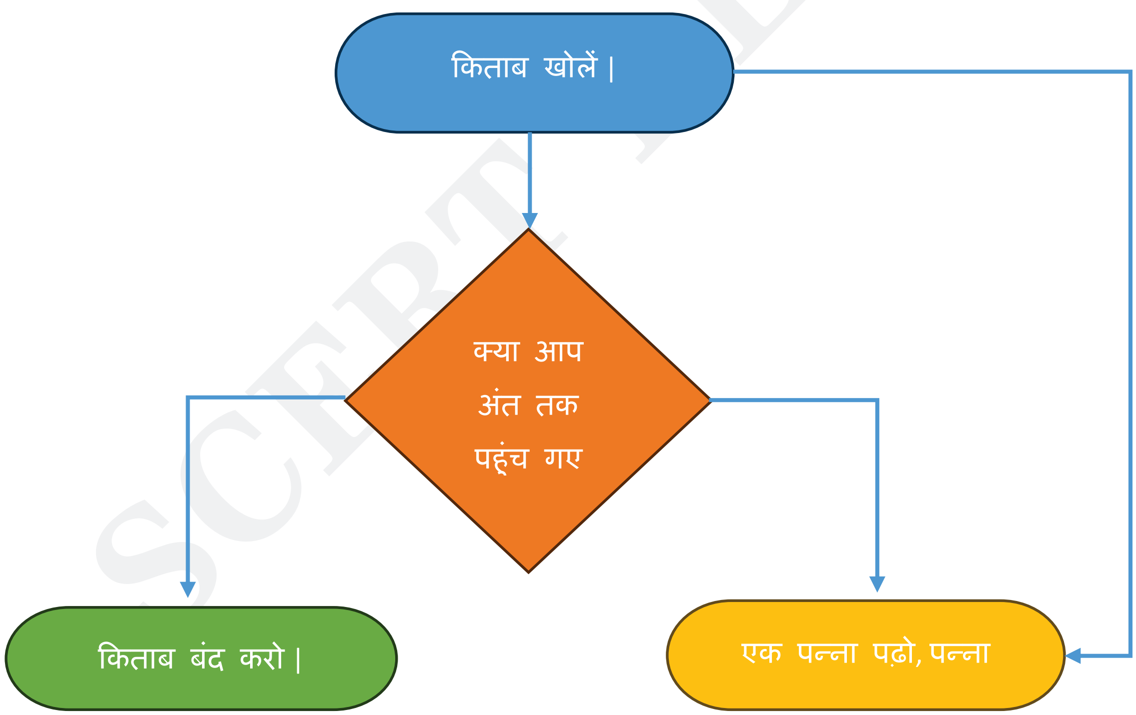
Operator	नाम
+=	

>	
!=	
/=	
%=	
&&	
||	

3.5 चयन वक्तव्यों (if, if else और switch case) का उपयोग कब करें, इसे समझना

वास्तविक जीवन में, आप अक्सर अपने कार्यों का चयन इस आधार पर करते हैं कि कोई स्थिति सत्य है या नहीं

असत्य। उदाहरण के लिए, यदि बाहर बारिश हो रही है तो आप छाता लेकर चलें, अन्यथा नहीं।



चित्र 3.7 चयन वक्तव्यों फ़्लो चार्ट

किसी program में, हमें किसी expression की value के आधार पर program के एक हिस्से को चलाना पड़ सकता है। Java में ऐसे conditionally code को execute करने के लिए दो

statements दिए गए हैं – if else statement और switch statement। इनका उपयोग program में कुछ specific conditions के true या false होने पर code का एक block (curly braces { } में enclosed statements का sequence) चलाने के लिए किया जाता है।

1. if (Condition):

उपयोग: जब आपको एक शर्त (condition) को जांचना हो और अगर वह सही है (true), तो एक कोड का ब्लॉक चलाना हो।

उदाहरण

```
int number = 10;
if (number > 5) {
System.out.println("Number is greater than 5");
}
```

कब उपयोग करें: जब केवल एक ही शर्त को जांचना हो और उसके आधार पर कार्रवाई करनी हो। अगर शर्त गलत है, तो कोई और कार्रवाई नहीं की जाती है।

2. if-else (Condition):

उपयोग: जब आपको एक शर्त को जांचना हो और अगर वह सही है, तो एक कोड ब्लॉक चलाना हो; अगर वह गलत है, तो एक दूसरा कोड ब्लॉक चलाना हो।

उदाहरण

```
int number = 3;
if (number > 5) {
System.out.println('Number is greater than 5');
} else {
System.out.println('Number is less than or equal to 5');
}
```

कब उपयोग करें: जब आपके पास एक वैकल्पिक (alternate) शर्त हो। यदि पहली शर्त सही नहीं है, तो आप दूसरे कोड ब्लॉक को चलाना चाहते हैं।

3. if-else if-else (Condition):

उपयोग: जब आपको कई शर्तों (conditions) को जांचना हो और उनके आधार पर अलग-अलग

कार्रवाइयों का निर्णय लेना हो।

उदाहरण

```
int number = 0;
if (number > 0) {
System.out.println('Positive number');
} else if (number < 0) {
System.out.println('Negative number');
} else {
System.out.println('Number is zero');
}
```

कब उपयोग करें: जब आपके पास एक से अधिक शर्तें हों और आप उनमें से हर एक के लिए अलग-अलग कार्रवाई करना चाहते हों।

4. Switch-case (Condition):

उपयोग: जब आपके पास एक वैरिएबल की वैल्यू के अनुसार कई विकल्प (cases) हों और आपको हर विकल्प के लिए एक अलग कोड ब्लॉक चलाना हो।

उदाहरण

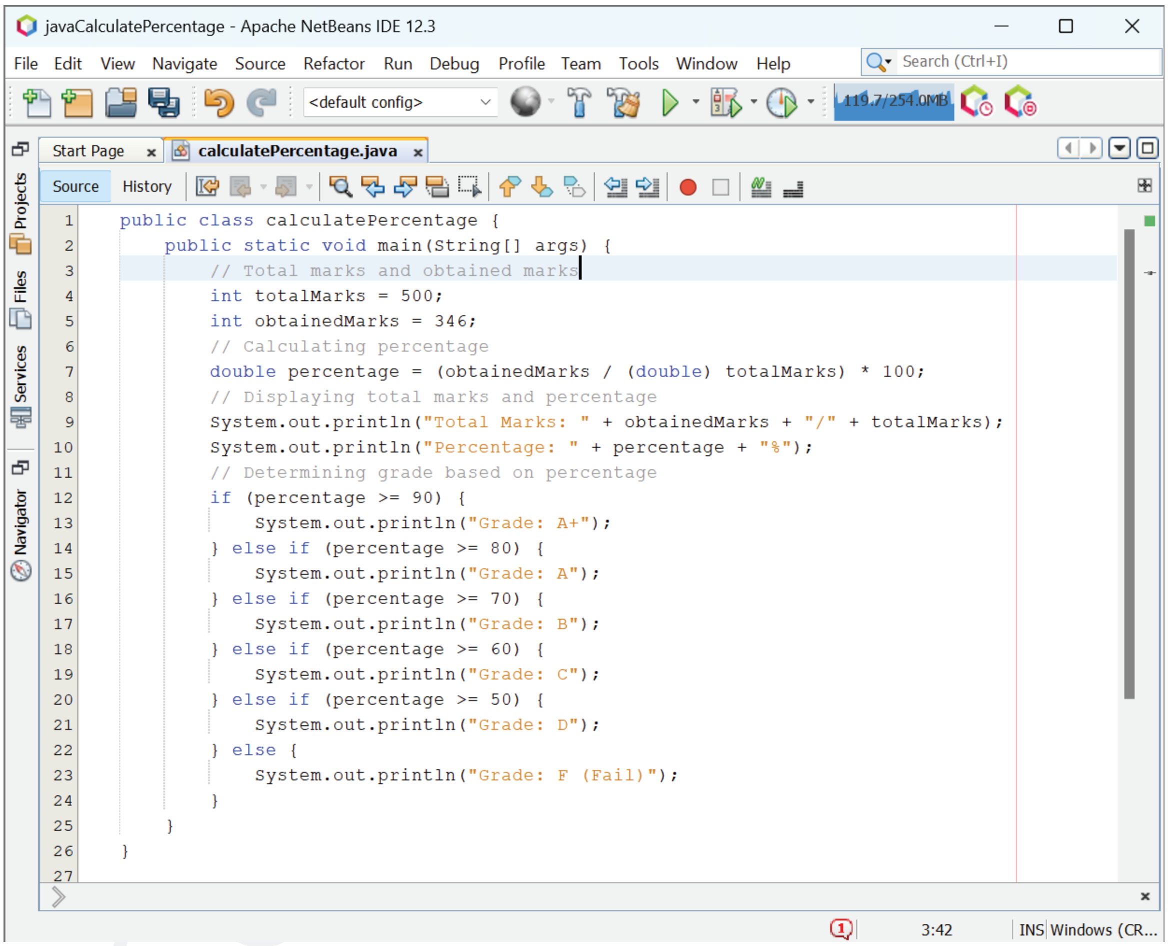
```
int day = 3;
switch (day) {
case 1:
System.out.println("Monday");
break;
case 2:
System.out.println("Tuesday");
break;
case 3:
System.out.println ('Wednesday');
break;
default:
System.out.println ('Invalid day');
}
```

कब उपयोग करें: जब आपके पास एक वैरिएबल की निश्चित वैल्यू के लिए कई विकल्प हों। यह तब उपयोगी होता है जब वैरिएबल के कई पूर्व-निर्धारित मान होते हैं (जैसे दिन, महीने, मेनू विकल्प

आदि)। यह कोड को अधिक स्पष्ट और पढ़ने में आसान बनाता है।

यहाँ एक simple Java program है जो total marks 500 और obtained marks 346 के आधार पर percentage calculate करता है और फिर उस percentage के आधार पर grade दिखाता है। इसमें if-else if-else conditions का उपयोग किया गया है।

Example



चित्र 3.8 चयन वक्तव्यों का उदाहरण

Program Explanation

1. Given Values: Total marks को 500 और obtained marks को 346 set किया गया है।

2. Percentage Calculation: Formula (obtainedMarks / (double) totalMarks) * 100 का उपयोग करके percentage निकाली जाती है।

3. Grade Calculation: if-else if-else statements का उपयोग करके percentage के आधार पर grade assign किया जाता है।

* 90 या उससे अधिक पर A+
* 80 या उससे अधिक पर A
* 70 या उससे अधिक पर B
* 60 या उससे अधिक पर C
* 50 या उससे अधिक पर D
* 50 से कम पर F (Fail)

Example Output

```
Total Marks: 346/500
Percentage: 69.2%
Grade: B
```

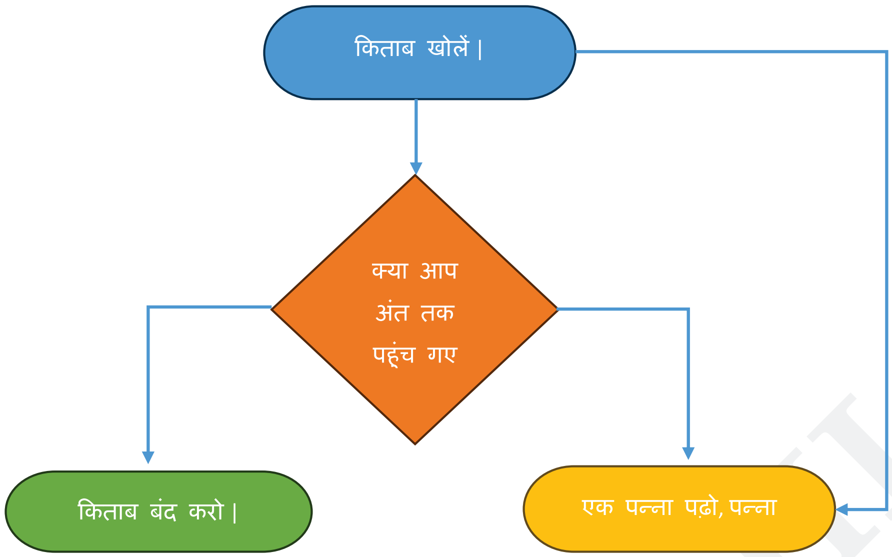
इस program से percentage calculate की जाती है और फिर उस percentage के आधार पर grade assign किया जाता है।

3.6 पुनरावृत्ति (Repetition) कथनों (Statements) जैसे (while loop, do while loop, for loop)

1. दोहराव संरचनाएँ (Repetition Structures)

वास्तविक जीवन में आप अक्सर कोई काम बार-बार करते हैं, उदाहरण के लिए, किसी कार्य पर विचार करें जैसे एक किताब पढ़ते समय, पहले आप किताब खोलें, और फिर बार-बार - एक पृष्ठ पढ़ें; पृष्ठ पलटें - जब तक आप किताब के अंत तक न पहुंच जाएं, तब तक किताब बंद कर दें।

SCERT DELHI



चित्र 3.9: दोहराव संरचनाएँ का फ़्लो चार्ट

Java programming में loops का मतलब है कि हम किसी task को बार-बार repeat कर सकते हैं बिना code को बार-बार लिखे। Loops का use तब किया जाता है जब हमें कोई काम multiple times करना हो, जैसे कि किसी number का table print करना, या 1 से 10 तक के numbers को print करना।

2 Java में तीन प्रकार के loops होते हैं:

* for loop
* while loop
* do-while loop

एक-एक करके इन्हें समझते हैं:

1. for loop

यह loop तब use होता है जब हमें पता हो कि loop कितनी बार चलाना है।

```
for (int i = 1; i <= 5; i++) {
System.out.println(i);
}
```

SCERT DELHI

Explanation:

* int i = 1 से loop शुरू होता है, यानी i की initial value 1 है।
* i <= 5 loop की condition है, यानी जब तक i 5 से कम या बराबर है, loop चलेगा।
* i++ का मतलब है कि हर बार loop के खत्म होने पर i की value 1 बढ़ जाएगी।

Output

```
1
2
3
4
5
```

2. while loop

यह loop तब use होता है जब हमें exact repetition count नहीं पता, लेकिन कोई condition true रहने तक loop को चलाना हो।

```
int i = 1;
while (i <= 5) {
System.out.println(i);
i++;
}
```

Explanation:

* int i = 1 से variable i की initial value set की गई।
* while (i <= 5) से condition दी गई कि जब तक i की value 5 से कम या बराबर है, loop चलता रहेगा।
* हर बार loop के खत्म होने पर i++ से i की value 1 बढ़ेगी।

Output

```
1
2
3
4
5
```

3. do-while loop

यह loop कम से कम एक बार ज़रूर चलेगा, भले ही condition false हो। Condition चेक करने से पहले body execute होती है।

```
int i = 1;
do {
System.out.println(i);
i++;
} while (i <= 5);
```

Explanation:

* do block के अंदर पहले i print होगा, फिर i++ से value बढ़ेगी।
* उसके बाद condition (i <= 5) चेक की जाएगी। अगर true है, तो loop फिर से चलेगा, अन्यथा रुक जाएगा।

Output:

```
1
2
3
4
5
```

यह सभी बिंदु आपको JAVA प्रोग्रामिंग में एक मजबूत आधार बनाने में मदद करेंगे और आपको विभिन्न घटकों और उनके उपयोग को समझने में सहायता करेंगे।

while loop	**do while loop**
Condition पहले check होती है, फिर body execute होती है।	पहले body execute होती है, फिर condition check होती है।
Condition false होने पर body एक भी बार execute नहीं होती।	Condition false होने पर भी body कम से कम एक बार execute होती है।

while (condition) {
  // loop body
}
	do {
	  // loop body
	}
	while (condition);

जब हमें पहले condition check करनी हो और उसी के आधार पर execute करना हो।	जब हमें body को कम से कम एक बार ज़रूर execute करना हो।

* Loops का उपयोग करते समय होने वाली common errors के points:

- **Infinite Loop (अनंत लूप):** Loop कभी खत्म नहीं होता क्योंकि condition कभी false नहीं होती।
- **Off-by-One Error (एक से अधिक या कम का error):** Loop एक बार ज्यादा या कम चल जाता है।
- **Wrong Initialization (गलत initialization):** Loop variable की initial value गलत रखना।
- **Wrong Update Statement (गलत update statement):** Loop variable को सही तरीके से update न करना।
- **Using Wrong Data Type (गलत data type का उपयोग):** Loop variable के लिए गलत data type चुनना।
- **Skipping Loop Body (Loop की body को skip करना):** Loop की body को { } में न रखना।
- **Incorrect Condition (गलत condition):** Loop की condition गलत तरीके से लिखना, जैसे = की जगह == का उपयोग न करना।

इन errors से बचने के लिए loop components को ध्यान से check करें।

सारांश (Summary):

Loops का use हमें repetitive tasks को efficiently करने में मदद करता है। for loop तब अच्छा है जब repetition count पता हो, while loop तब use होता है जब हमें सिर्फ condition पता हो, और do-while तब helpful है जब हमें कम से कम एक बार task run करना हो।

3.7 Array

Java में Array एक data structure है जो एक ही type के कई elements को एक साथ store करने के लिए उपयोग किया जाता है। Array का उपयोग तब किया जाता है जब हमें एक ही तरह के multiple values को एक ही variable में store करना हो।

Array एक contiguous memory location में elements को store करता है, और हर element को एक unique index से access किया जा सकता है।

1. Array की विशेषताएँ:

* Array का size fix होता है, यानी हम इसे run-time में change नहीं कर सकते।
* Array में हम केवल एक ही data type के elements store कर सकते हैं (जैसे int, float, या String)।
* Array के elements का indexing 0 से शुरू होती है, यानी पहला element index 0 पर होता है, दूसरा index 1 पर, और ऐसे ही आगे।

2. Array बनाना और initialize करना

Java में array को declare और initialize करने के कई तरीके हैं।

Example 1: Integer Array

```
int[] numbers = new int[5]; // 5 size का एक integer array बनाया
numbers[0] = 10; // पहला element
numbers[1] = 20; // दूसरा element
numbers[2] = 30; // तीसरा element
numbers[3] = 40; // चौथा element
numbers[4] = 50; // पाँचवाँ element
```

SCERT DELHI

इसमें हमने एक integer array बनाया है जिसका नाम numbers है और size 5 है। इस array में 5 elements store किए जा सकते हैं।

Example 2: Array Initialization करते समय Values Assign करना

```
int[] numbers = {10, 20, 30, 40, 50}; // सीधे values assign कर दीं
```

इस array में भी 5 elements हैं और हमने initialization के समय values assign कर दीं।

3. Array के Elements को Access करना और Print करना

Array के elements को access करने के लिए हम उसकी index का उपयोग करते हैं। जैसे कि numbers[0] हमें पहला element देगा।

```
int[] numbers = {10, 20, 30, 40, 50};
for (int i = 0; i < numbers.length; i++) {
System.out.println(numbers[i]);
}
```

Explanation:

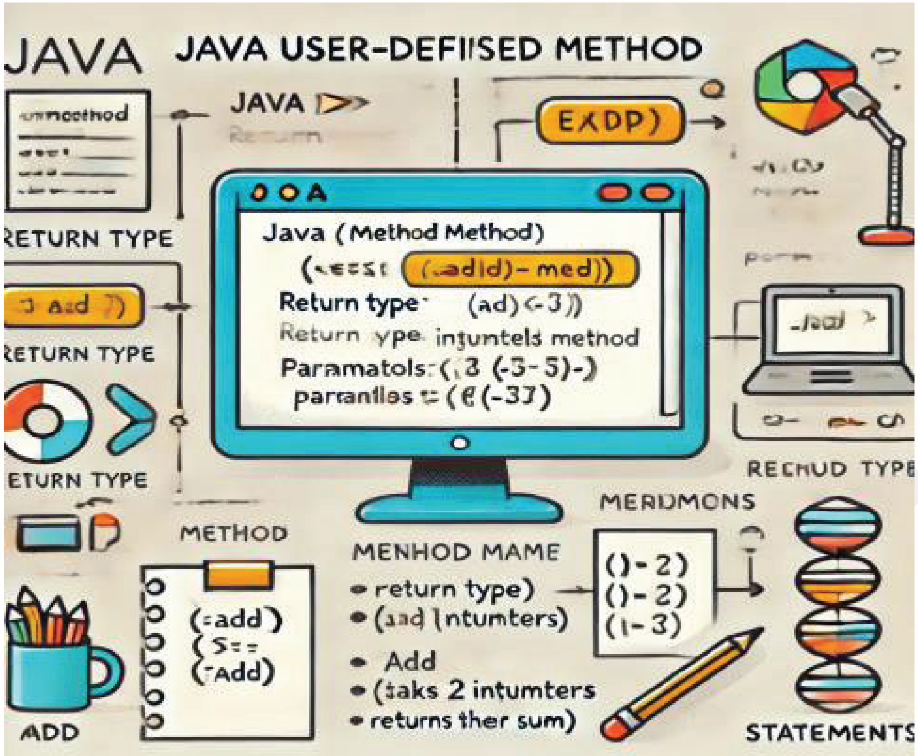
* यहाँ numbers.length array के size को बताता है। numbers array का length 5 है।
* for loop array के हर element को print करेगा।

Output

```
10
20
30
40
50
```

3.8 यूजर-डिफाइनड मेथड (User Define Method)

जावा(Java) में यूजर-डिफाइनड (User-Defined) मेथड्स का मतलब ऐसे मेथड्स से है जिन्हें प्रोग्रामर खुद अपनी ज़रूरत के हिसाब से बनाता है। ये मेथड्स प्रोग्राम को मॉड्यूलर और व्यवस्थित बनाने में मदद करते हैं।



चित्र 3.10 यूजर-डिफाइनड मेथड

1. मेथड (Method) क्या है?

जावा में, मेथड कोड का एक समूह होता है जिसे किसी विशेष काम को पूरा करने के लिए डिज़ाइन किया जाता है। उदाहरण के लिए, अगर हमें बार-बार किसी गणना को करना है, तो हम एक मेथड बना सकते हैं और जब भी जरूरत हो, उसे कॉल कर सकते हैं। इससे कोड को दोहराने की जरूरत नहीं पड़ती है।

2. यूजर-डिफाइनड मेथड क्यों बनाते हैं?

* **कोड का पुनः** उपयोग (Code Reusability): एक बार मेथड बनाकर उसे कई बार उपयोग किया जा सकता है।
* **कोड को पढ़ने और समझने में आसानी (Readability and Modularity):** मेथड्स को प्रोग्राम को छोटे-छोटे भागों में विभाजित कर देते हैं, जिससे प्रोग्राम को पढ़ना और समझना आसान होता है।
* **एरर हैंडलिंग में सहूलियत (Error Handling):** छोटे मेथड्स में एरर को पकड़ना आसान होता है।

3. मेथड की संरचना (Method Sturucture)

जावा में मेथड को निम्नलिखित संरचना में लिखा जाता हैः

```
returnType methodName(parameters) {
// Statements (Statements वह कोड होते हैं जो मेथड में लिखे जाते हैं)
return value; // अगर मेथड कोई वैल्यू लौटाता है
}
```

* **return Type:** यह बताता है कि मेथड किस प्रकार की वैल्यू को वापस करेगा, जैसे int, double, void (अगर मेथड कुछ भी वापस नहीं करता)।
* **method Name:** यह मेथड का नाम होता है, जिसे कॉल करने पर मेथड execute होता है।
* **parameters:** ये मेथड के लिए इनपुट वैल्यूज़ होती हैं, जो मेथड को दिए जाते हैं। अगर इनपुट नहीं चाहिए तो खाली छोड़ा जा सकता है।

4. एक साधारण उदाहरणः जोड़ (Addition) करने का मेथड

```
public class Calculator {
// User-defined method to add two numbers
int add(int a, int b) {
int sum = a + b;
return sum;
}
public static void main(String[] args) {
Calculator calc = new Calculator();
int result = calc.add(10, 20); // मेथड कॉल
System.out.println("Result of add: " + result);
}
}
```

Example program समझें:

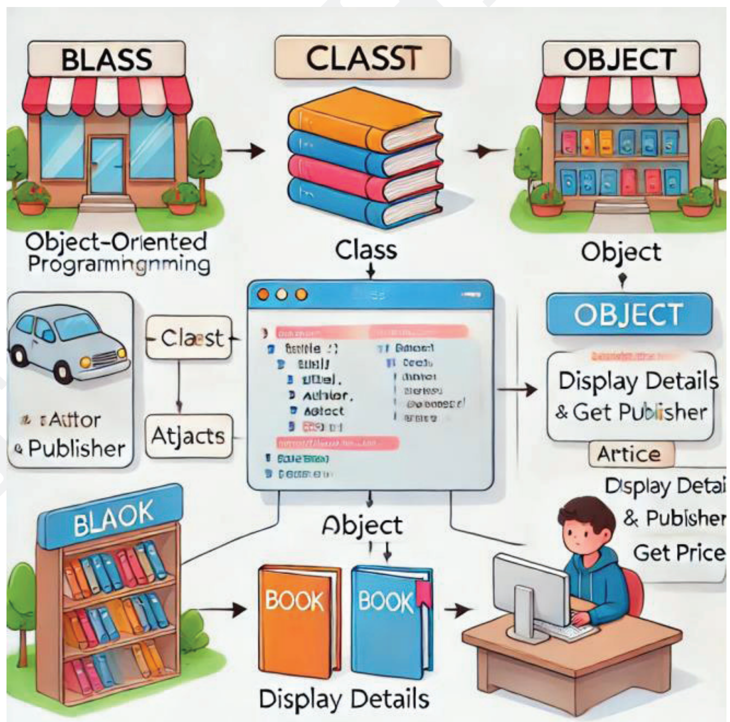
* int add(int a, int b): यह मेथड दो संख्याओं को जोड़ता है और उनका योग sum में स्टोर करता है।

* calc.add(10, 20);: add मेथड को 10 और 20 के साथ कॉल किया जाता है और इसका परिणाम result में स्टोर किया जाता है।

3.9 ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग (OOP)

अब तक आप जावा प्रोग्रामिंग के बारे में काफी कुछ जान चुके हैं। अब हम जावा की सबसे मूलभूत विशेषता – क्लासेज और ऑब्जेक्ट्स को समझना शुरू करेंगे। जावा एक ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग (OOP) भाषा है। OOP भाषा में, एक प्रोग्राम ऑब्जेक्ट्स का संग्रह होता है जो एक-दूसरे के साथ इंटरैक्ट करते हैं ताकि किसी समस्या का समाधान किया जा सके। प्रत्येक ऑब्जेक्ट एक क्लास का उदाहरण होता है।

कल्पना करें कि आप एक बुकस्टोर के लिए एक डेटाबेस एप्लिकेशन बना रहे हैं। आपको स्टोर में सभी किताबों के बारे में डेटा स्टोर करने की आवश्यकता होगी। इस तरह, प्रत्येक किताब आपके प्रोग्राम में एक ऑब्जेक्ट बन जाएगी। आगे, प्रत्येक किताब के कुछ विशेष लक्षण या गुण होंगे जैसे कि उसका शीर्षक, लेखक, प्रकाशक, और मूल्य। आप किसी किताब पर कुछ क्रियाएँ भी करना चाह सकते हैं, जैसे कि उसकी जानकारी कंप्यूटर स्क्रीन पर दिखाना या उसका मूल्य पता करना।



चित्र 3.11 OOPS

OOP भाषा में, जैसे कि जावा, इस उदाहरण में एक किताब जैसी एंटिटी को क्लास कहा जाता है। एक क्लास एक भौतिक या तार्किक एंटिटी होती है जिसमें कुछ गुण होते हैं। "बुक" क्लास में शीर्षक, लेखक, प्रकाशक, शैली (genre) और मूल्य इसके डेटा मेंबर्स होते हैं। किताब की जानकारी दिखाना और उसका मूल्य प्राप्त करना "बुक" क्लास के मेथड मेंबर्स होते हैं। मेथड मेंबर्स क्लास के डेटा मेंबर्स को प्राप्त (get), सेट या अपडेट कर सकते हैं।

जावा में ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग (OOP) का उपयोग करके "Book" क्लास को इस प्रकार से परिभाषित किया जा सकता है। इस उदाहरण में, हम एक बुकस्टोर के लिए "Book" नाम की एक

क्लास बना रहे हैं, जिसमें किताब के बारे में जानकारी को स्टोर करने के लिए डेटा मेंबर्स (गुण) होंगे और उस जानकारी को प्राप्त करने के लिए मेथड्स होंगे।

उदाहरणः Book class

```
public class Book {
// डेटा मेंबर्स (Attributes)
private String title;     // किताब का शीर्षक
private String author;    // लेखक का नाम
private String publisher; // प्रकाशक का नाम
private double price;     // किताब का मूल्य
// कॉन्स्ट्रक्टर (Constructor) - क्लास का उदाहरण बनाने के लिए
public Book(String title, String author, String publisher, double price) {
this.title = title;
this.author = author;
this.publisher = publisher;
this.price = price;
}
// मेथड्स (Methods)
// जानकारी दिखाने के लिए मेथड
public void displayDetails( ) {
System.out.println("Title of Book: " + title);
System.out.println("लेखकः " + author);
System.out.println("प्रकाशकः " + publisher);
System.out.println("मूल्यः " + price);
}
// किताब का मूल्य प्राप्त करने के लिए मेथड
public double getPrice( ) {
return price;
}
}
```

विवरणः

1 **डेटा मेंबर्स (Data Members):** ये Book क्लास की विशेषताएँ हैं, जो हर किताब की जानकारी को स्टोर करते हैंः

* title: किताब का शीर्षक
* author: लेखक का नाम
* publisher: प्रकाशक का नाम
* price: किताब का मूल्य

2. **कन्स्ट्रक्टर (Constructors):** Book क्लास में एक कन्स्ट्रक्टर होता है जो डेटा मेंबर्स को इनिशियलाइज़ करता है। जब कोई नया Book ऑब्जेक्ट बनता है, तो हमें उसकी सभी विशेषताओं के लिए वैल्यू देनी होती है, जो इस कन्स्ट्रक्टर का उपयोग करके सेट की जाती हैं।

3. **मेथड्सः**

* displayDetails( ): यह मेथड किताब की सभी जानकारी को प्रिंट करता है।
* getPrice( ): यह मेथड किताब का मूल्य वापस करता है।

कैसे उपयोग करेंः

```
public class Main {
public static void main(String[] args) {
// Book का नया ऑब्जेक्ट बनाना
Book myBook = new Book("Java Programming", "James Gosling", "Tech
Books", 399.99);
// किताब की जानकारी दिखाना
myBook.displayDetails( );
// किताब का मूल्य प्राप्त करना
System.out.println("किताब का मूल्य हैः " + myBook.getPrice( ));
}
}
```

आउटपुटः

किताब का शीर्षकः Java Programming
लेखकः James Gosling
प्रकाशकः Tech Books
मूल्यः 399.99
किताब का मूल्य हैः 399.99

इस प्रकार, Book क्लास का उपयोग करके हम किताब की जानकारी को व्यवस्थित रूप से स्टोर और प्रबंधित कर सकते हैं।

3.10. जावा (Java) लाइब्रेरी

जावा लाइब्रेरी (Java Library) जावा में तैयार किए गए प्री-बिल्ट कोड का एक संग्रह है, जिसे हम आसानी से अपने प्रोग्राम में इस्तेमाल कर सकते हैं। छात्रों के लिए इसे समझना बहुत महत्वपूर्ण है क्योंकि यह प्रोग्रामिंग को आसान और अधिक प्रभावी बनाता है। आइए इसे आसान शब्दों में समझते हैं।

1. जावा (Java) लाइब्रेरी क्या है?

जावा लाइब्रेरी पहले से तैयार किए गए प्रोग्राम्स, क्लासेज और मेथड्स का संग्रह है। यह ऐसे टूल्स की तरह है जो अलग-अलग कार्यों को जल्दी से पूरा करने में हमारी मदद करते हैं। जब हम जावा प्रोग्राम लिखते हैं, तो हम लाइब्रेरी में उपलब्ध इन टूल्स का उपयोग करके बिना अतिरिक्त मेहनत के अपना काम कर सकते हैं।

उदाहरण के लिए, जैसे आपको किसी दोस्त से चॉकलेट चाहिए, तो आप अपने लिए चॉकलेट बनाने के बजाय, दुकान से मंगवा लेते हैं। उसी तरह, प्रोग्रामिंग में जावा लाइब्रेरी हमारे लिए "दुकान" की तरह काम करती है, जिससे हम पहले से बने मेथड्स और क्लासेज का उपयोग करते हैं।

2. जावा (Java) लाइब्रेरी का महत्व

* **कोडिंग आसान बनाती हैः** पहले से बने कोड का उपयोग करके हमें किसी भी चीज़ को शुरुआत से बनाने की जरूरत नहीं होती।
* **समय की बचतः** बार-बार इस्तेमाल होने वाले कार्यों के लिए जावा लाइब्रेरी से डायरेक्ट कोड का उपयोग करके समय बचता है।

* **एरर फ्री कोडिंग:** जावा लाइब्रेरी के क्लासेज और मेथड्स को पहले से ही अच्छी तरह से टेस्ट किया गया होता है, जिससे एरर की संभावना कम हो जाती है।

3. जावा (Java) लाइब्रेरी के प्रमुख भाग

जावा लाइब्रेरी में कई "पैकेज" (Packages) होते हैं, और हर पैकेज में विभिन्न कार्यों के लिए बनाए गए क्लासेज होते हैं। कुछ महत्वपूर्ण पैकेज ये हैं:

a) java.lang

यह जावा का सबसे बुनियादी पैकेज है, जो हर प्रोग्राम में ऑटोमैटिक शामिल होता है। इसमें कुछ बेहद जरूरी टूल्स और मेथड्स होते हैं:

* **String:** इसे टेक्स्ट को स्टोर करने और उससे जुड़े काम करने के लिए उपयोग किया जाता है।
* **Math:** गणितीय कार्यों के लिए जैसे जोड़, घटाना, गुणा, और स्क्वायर रूट निकालना।
* **System:** प्रोग्राम से बाहर की दुनिया से बातचीत करने के लिए, जैसे प्रोग्राम में डेटा लेना और दिखाना।

b) java.util

यह पैकेज अलग-अलग तरह के डेटा को मैनेज करने के लिए है।

* **ArrayList:** यह एक ऐसा ऐरे(array) है जिसकी साइज को बढ़ाया और घटाया जा सकता है।
* **HashMap:** डेटा को key-value के फॉर्म में स्टोर करने के लिए उपयोग होता है।
* **Date और Calendar:** समय और तारीख को मैनेज करने के लिए।

c) java.io

इसका उपयोग इनपुट और आउटपुट (I/O) कार्यों के लिए होता है, जैसे फाइल से डेटा पढ़ना या उसमें डेटा लिखना।

* **File:** किसी फाइल के साथ काम करने के लिए।
* **BufferedReader और FileWriter:** फाइल में डेटा लिखने और पढ़ने के लिए।

d) java.awt और javax.swing

SCERT DELHI

इनका उपयोग प्रोग्राम में ग्राफिकल यूजर इंटरफेस (GUI) बनाने के लिए होता है।

* Button, TextField, Label जैसी चीजें GUI बनाने में मदद करती हैं।
* javax.swing में एडवांस GUI बनाने के लिए JButton, JTextField, JLabel आदि उपलब्ध हैं।

उदाहरण के माध्यम से समझें

मान लीजिए, हम गणितीय कार्यों के लिए एक साधारण प्रोग्राम बना रहे हैं। इसमें हमें दो संख्याओं का बड़ा मान (maximum) निकालना है। इसके लिए हम Math क्लास का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
int a = 10;
int b = 20;
int max = Math.max(a, b); // Math क्लास का max मेथड उपयोग कर रहे हैं
System.out.println("Maximum value is: " + max);
}
}
```

यहां पर हम Math.max मेथड का उपयोग करके a और b में से बड़ा मान निकाल रहे हैं, जो कि हमें इस पैकेज से आसानी से उपलब्ध हो जाता है।

7. जावा लाइब्रेरी का उपयोग कैसे करें?

जावा (Java) की असली ताकत उन सैकड़ों प्रीबिल्ट क्लासेज में है, जो पहले से जावा में उपलब्ध हैं और जिनका हम अपने प्रोग्राम में इस्तेमाल कर सकते हैं। इन प्रीबिल्ट क्लासेज और उनमें मौजूद मेथड्स का उपयोग करने के लिए हमें केवल import कीवर्ड का उपयोग करना होता है। import का मतलब है कि हम उस क्लास को उस पैकेज से लाकर अपने प्रोग्राम में जोड़ सकते हैं, जहां वह क्लास मौजूद है।

जब हम किसी प्रीबिल्ट क्लास को इंपोर्ट करना चाहते हैं, तो import स्टेटमेंट को हमारे प्रोग्राम में किसी भी क्लास डिफिनिशन से पहले रखना होता है। इस तरीके से हम उन क्लासेज और उनके मेथड्स को अपने प्रोग्राम में सीधे उपयोग कर सकते हैं।

* **पैकेज इंपोर्ट करें:** अगर किसी विशेष पैकेज की आवश्यकता हो, तो हमें उसे इंपोर्ट करना होता है। जैसे:

```
import java.util.ArrayList;
```

* क्लासेज और मेथड्स का उपयोगः इंपोर्ट किए गए क्लासेज और मेथड्स का उपयोग आसानी से प्रोग्राम में किया जा सकता है।

5 Data Input

जावा में डेटा इनपुट का मतलब है कि हम अपने प्रोग्राम में बाहर से डेटा ले सकते हैं, जैसे कि यूजर से इनपुट प्राप्त करना। हम मुख्यतः Scanner क्लास का उपयोग करके डेटा इनपुट करना सीखते हैं, क्योंकि यह आसान और प्रभावी तरीका है।

Scanner क्लास का उपयोग

Scanner क्लास का उपयोग करके हम अलग-अलग प्रकार के डेटा, जैसे कि संख्याएं, स्ट्रिंग (पाठ), फ्लोटिंग-पॉइंट नंबर आदि, को इनपुट कर सकते हैं। यह java.util पैकेज में मौजूद है।

Scanner का सेटअप

पहले हमें Scanner क्लास को इंपोर्ट करना होगा, जिससे हम इसका उपयोग कर सकेंः

```
import java.util.Scanner;
```

इसके बाद, हम Scanner का एक ऑब्जेक्ट बनाते हैं, जो कीबोर्ड (System.in) से इनपुट लेगाः

```
Scanner sc = new Scanner(System.in);
```

उदाहरण 1: संख्यात्मक इनपुट लेना

मान लीजिए, हमें दो संख्याएं यूजर से इनपुट लेकर उनका जोड़ करना हैः

```
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in); // Scanner का ऑब्जेक्ट बनाना
System.out.print('पहली संख्या दर्ज करें:');
int num1 = sc.nextInt( ); // पहली संख्या को इनपुट के रूप में लेना
System.out.print('दूसरी संख्या दर्ज करें:');
int num2 = sc.nextInt( ); // दूसरी संख्या को इनपुट के रूप में लेना
int sum = num1 + num2; // दोनों संख्याओं का जोड़ करना
System.out.println("संख्याओं का जोड़ है: " + sum); // परिणाम दिखाना
```

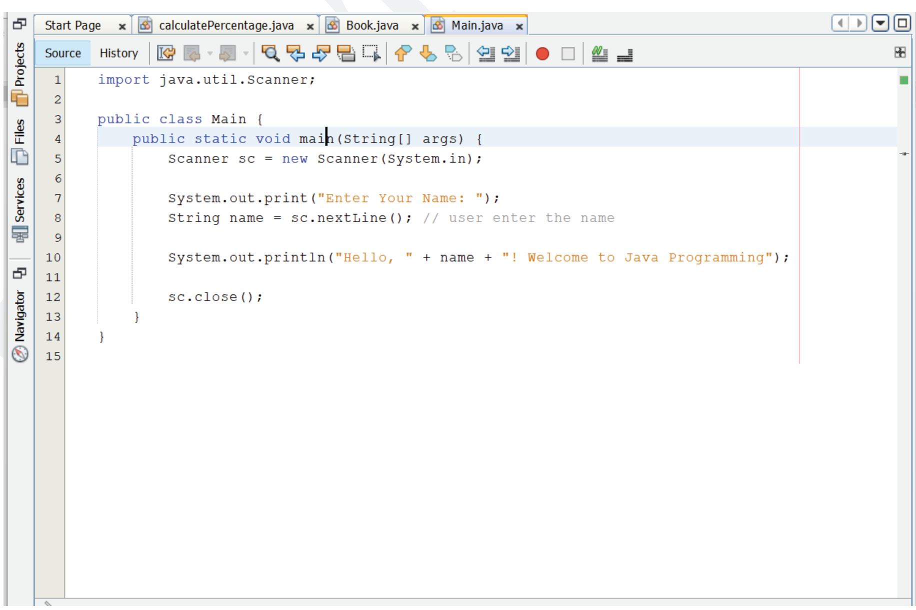
```
sc.close( ); // Scanner को बंद करना
}
}
```

यह प्रोग्राम यूजर से दो संख्याएं लेगा और फिर उनका जोड़ दिखाएगा।

उदाहरण 2: स्ट्रिंग (पाठ) इनपुट लेना

मान लीजिए, हम यूजर का नाम इनपुट लेकर उसे एक संदेश दिखाना चाहते हैं:

```
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter Your Name: ");
String name = sc.nextLine( ); // यूजर से पूरा नाम लेना
System.out.println('Hello,' + name +'! Welcome to Java Programming');
sc.close( );
}
}
```



चित्र 3.12 जावा स्ट्रिंग का उदाहरण

इस प्रोग्राम में nextLine ( ) मेथड का उपयोग करके हमने एक पूरी लाइन (स्ट्रिंग) को इनपुट के

रूप में लिया और यूजर को एक स्वागत संदेश दिखाया।

Output

```
Enter Your Name : Ritu
Hello Ritu! Welcome to Java Programming
```

6 Array Manipulation

जावा (Java) में Array Manipulation का मतलब है कि हम Arrays के साथ विभिन्न प्रकार के कार्य कर सकते हैं जैसे कि उन्हें बदलना, छांटना (sorting), ढूंढना (searching), कॉपी करना, आदि। आइए इसे आसान उदाहरणों के साथ समझते हैं।

* Array को Sort करना

जावा में Arrays को छांटने (sort) के लिए Arrays.sort( ) मेथड का उपयोग किया जा सकता है। यह मेथड छोटे से बड़े क्रम (ascending order) में Array को सॉर्ट कर देता है।

```
import java.util.Arrays;
public class Main {
public static void main(String[] args) {
int[] numbers = {5, 2, 8, 4, 1};
Arrays.sort(numbers); // Array को सॉर्ट करना
System.out.println('Sorted Array: ' + Arrays.toString(numbers));
}
}
```

Output:

```
Sorted Array: [1, 2, 4, 5, 8]
```

यहां Arrays.sort(numbers); मेथड ने Array को बढ़ते क्रम में सॉर्ट कर दिया।

* Array में किसी विशेष एलिमेंट को ढूंढना

Array में किसी विशेष एलिमेंट को खोजने के लिए Arrays.binarySearch( ) का उपयोग किया जा सकता है, लेकिन ध्यान दें कि Array पहले से सॉर्ट होना चाहिए।

```
import java.util.Arrays;
public class Main {
public static void main(String[] args) {
int[] numbers = {10, 20, 30, 40, 50};
int index = Arrays.binarySearch(numbers, 30); // एलिमेंट 30 की खोज
if (index >= 0) {
System.out.println("Index of element 30 is: " + index);
} else {
System.out.println("index is not available of element 30");
}
}

}
```

Output

```
Index of element 30 is: 2
```

यहां Arrays.binarySearch(numbers, 30); मेथड ने 30 को ढूंढकर उसका इंडेक्स लौटाया।

* Array की कॉपी करना

कभी-कभी हमें एक Array की कॉपी बनानी होती है। इसके लिए हम Arrays.copyOf( ) का उपयोग कर सकते हैं।

```
import java.util.Arrays;
public class Main {
public static void main(String[] args) {
int[] original = {1, 2, 3, 4, 5};
int[] copy = Arrays.copyOf(original, original.length); // Array की कॉपी
बनाना
System.out.println("Original Array: " + Arrays.toString(original));
System.out.println("Copied Array: " + Arrays.toString(copy));
}
}
```

Output:

```
Original Array: [1, 2, 3, 4, 5]
Copied Array: [1, 2, 3, 4, 5]
```

यहां Arrays.copyOf(original, original.length); ने original Array की एक कॉपी copy में बनाई।

* Array को भरना (Fill करना)

Arrays.fill( ) मेथड का उपयोग करके हम पूरे Array में एक ही मान भर सकते हैं।

```
import java.util.Arrays;
public class Main {
public static void main(String[] args) {
int[] numbers = new int[5];
Arrays.fill(numbers, 10); // Array में सभी एलिमेंट्स को 10 से भरना
System.out.println('Filled Array: ' + Arrays.toString(numbers));
}
}
```

Output:

```
Filled Array: [10, 10, 10, 10, 10]
```

यहां Arrays.fill(numbers, 10); ने पूरे Array में 10 भर दिया।

* Array को प्रिंट करना

Array को सीधे प्रिंट करने के लिए Arrays.toString( ) मेथड का उपयोग कर सकते हैं।

```
import java.util.Arrays;
public class Main {
public static void main(String[] args) {
int[] numbers = {5, 10, 15, 20, 25};
System.out.println('Array: ' + Arrays.toString(numbers));
}
}
```

Output:

```
Array: [5, 10, 15, 20, 25]
```

संक्षेप में

जावा में Arrays क्लास की मदद से हम Arrays के साथ आसानी से काम कर सकते हैं:

* **Arrays.sort() :** Array को सॉर्ट करने के लिए।
* **Arrays.binarySearch() :** Array में एलिमेंट खोजने के लिए।
* **Arrays.copyOf() :** Array की कॉपी बनाने के लिए।
* **Arrays.fill() :** Array को भरने के लिए।
* **Arrays.toString() :** Array को प्रिंट करने के लिए।

इन आसान उदाहरणों से आप Array Manipulation के विभिन्न तरीके समझ सकते हैं और जावा प्रोग्रामिंग में उनका इस्तेमाल कर सकते हैं।

7. String Manipulation

जावा में स्ट्रिंग मैनिपुलेशन का मतलब है कि हम स्ट्रिंग्स (टेक्स्ट) पर विभिन्न ऑपरेशन्स (क्रियाएं) कर सकते हैं जैसे कि स्ट्रिंग को जोड़ना, काटना, ढूंढना, बदलना आदि। जावा में String क्लास में कई मेथड्स होते हैं, जिनसे हम स्ट्रिंग मैनिपुलेशन कर सकते हैं। आइए इसे आसान उदाहरणों के माध्यम से समझते हैं।

* **दो स्ट्रिंग्स को जोड़ना (Concatenation)**

स्ट्रिंग्स को जोड़ने के लिए हम + ऑपरेटर या concat( ) मेथड का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str1 = "Hello";
String str2 = "World";
// जोड़ना
String result = str1 + " " + str2; // या str1.concat(" ").concat(str2);
System.out.println("Joined string: " + result); // आउटपुट: Hello World
}
}
```

* स्ट्रिंग की लंबाई प्राप्त करना (Length)

स्ट्रिंग की लंबाई पता करने के लिए length( ) मेथड का उपयोग किया जाता है।

```
public class Main {
public static void main(String[] args) {
String str = "Java Programming";
int length = str.length( );
System.out.println("Length of string: " + length); // आउटपुट: 16
}
}
```

* किसी कैरेक्टर को एक्सेस करना (charAt मेथड)

हम किसी विशेष स्थान पर स्थित कैरेक्टर को पाने के लिए charAt( ) मेथड का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str = "Java";
char ch = str.charAt(2); // इंडेक्स 2 का कैरेक्टर प्राप्त करना
System.out.println("character at index 2: " + ch); // आउटपुट: v
}
}
```

* उपस्ट्रिंग निकालना (Substring)

हम स्ट्रिंग के कुछ हिस्से को निकालने के लिए substring( ) मेथड का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str = "Java Programming";
String subStr = str.substring(5, 16); // इंडेक्स 5 से 15 तक का उपस्ट्रिंग
System.out.println("Substring: " + subStr); // आउटपुट: Programming
}
}
```

* स्ट्रिंग को छोटे या बड़े अक्षरों में बदलना (Case Conversion)

स्ट्रिंग को छोटे या बड़े अक्षरों में बदलने के लिए toLowerCase( ) और toUpperCase( ) मेथड का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str = 'Java Programming';
System.out.println('In Capital Letters: ' + str.toUpperCase( )); // आउटपुट:
JAVA PROGRAMMING
System.out.println('In Small Letters: ' + str.toLowerCase( )); // आउटपुट:
java programming
}
}
```

* किसी कैरेक्टर को ढूंढना (Finding a Character)

किसी विशेष कैरेक्टर या उपस्ट्रिंग का पहला या आखिरी स्थान पता करने के लिए indexOf( ) और lastIndexOf( ) का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str = 'Java Programming';
int index = str.indexOf('a'); // पहला 'a' कहां है
int lastIndex = str.lastIndexOf("a"); // आखिरी 'a' कहां है
System.out.println("before ‘a’ index: " + index); // आउटपुट: 1
System.out.println("after ‘a’ index: " + lastIndex); // आउटपुट: 13
}
}
```

* स्ट्रिंग को रिप्लेस करना (Replace)

स्ट्रिंग में किसी कैरेक्टर या उपस्ट्रिंग को दूसरे कैरेक्टर या उपस्ट्रिंग से बदलने के लिए replace( ) का उपयोग कर सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str = 'Java Programming';
String replacedStr = str.replace('Java', 'Python');
System.out.println('Replaced String: ' + replacedStr); // आउटपुटः Python
      Programming
}
}
```

* **स्ट्रिंग को ट्रिम करना (Trim)**

trim( ) मेथड का उपयोग करके हम स्ट्रिंग के शुरू और अंत में मौजूद खाली स्थान (whitespace) को हटा सकते हैं।

```
public class Main {
public static void main(String[] args) {
String str = 'Hello World';
String trimmedStr = str.trim( );
System.out.println('Trimmed String:' + trimmedStr + '); // आउटपुटः 'Hello
World'
}
}
```

* **स्ट्रिंग की तुलना करना (Comparing Strings)**

स्ट्रिंग्स की तुलना करने के लिए equals( ) और equalsIgnoreCase( ) का उपयोग किया जाता है।

```
public class Main {
public static void main(String[] args) {
String str1 = "Java";
String str2 = "java";
```

```
System.out.println("समानता (case-sensitive): " + str1.equals(str2)); //
आउटपुट: false
System.out.println("समानता (case-insensitive): " + str1.
equalsIgnoreCase(str2)); // आउटपुट: true
}
}
```

संक्षेप में

* + या concat( ) : स्ट्रिंग्स जोड़ने के लिए।
* length( ) : स्ट्रिंग की लंबाई प्राप्त करने के लिए।
* charAt( ) : किसी विशेष स्थान पर कैरेक्टर पाने के लिए।
* substring( ) : उपस्ट्रिंग निकालने के लिए।
* toUpperCase( ) और toLowerCase( ) : स्ट्रिंग के अक्षर बदलने के लिए।
* indexOf( ) और lastIndexOf( ) : कैरेक्टर का स्थान पाने के लिए।
* replace( ) : कैरेक्टर या उपस्ट्रिंग को बदलने के लिए।
* trim( ) : खाली स्थान हटाने के लिए।
* equals( ) और equalsIgnoreCase( ) : स्ट्रिंग्स की तुलना के लिए।

ये सभी मेथड्स स्ट्रिंग मैनिपुलेशन के अलग-अलग तरीके हैं, जो हमें जावा में स्ट्रिंग्स के साथ आसानी से काम करने में मदद करते हैं।

स्ट्रिंग डेमो प्रोग्राम (String Demo Program)

```
public class StringDemo {
public static void main(String[] args) {
// कुछ स्ट्रिंग्स बनाना
String str1 = 'Hello';
String str2 = 'World';
```

SCERT DELHI

```
String str3 = 'Java Programming';
// 1. स्ट्रिंग्स जोड़ना (Concatenation)
String combined = str1 + ' ' + str2;
System.out.println('Joined string:' + combined);
// 2. स्ट्रिंग की लंबाई प्राप्त करना (Length)
System.out.println('length of str1 : ' + str1.length( ));
// 3. उपस्ट्रिंग निकालना (Substring)
String subStr = str3.substring(2, 6);
System.out.println('substring: ' + subStr);
// 4. छोटे और बड़े अक्षरों में बदलना (toLowerCase, toUpperCase)
System.out.println('in capital letters: ' + str1.toUpperCase( ));
System.out.println('In small letters: ' + str2.toLowerCase( ));
// 5. ट्रिम करना (Trim)
String trimmedStr = str3.trim( );
System.out.println('trimmed string: '' + trimmedStr + ''');
// 6. स्ट्रिंग को रिप्लेस करना (Replace)
String replacedStr = str1.replace('H', 'J');
System.out.println('replaced string: ' + replacedStr);
// 7. स्ट्रिंग की तुलना करना (Comparing Strings)
System.out.println("is str1 is equal to str2?: " + str1.equals(str2));
}
}
```

इस प्रोग्राम का आउटपुट कुछ इस प्रकार होगाः

```
Joined string: Hello World
Length of str1: 5
Substring: Jav
In Capital letters: HELLO
```

```
In small letters: world
Trimmed string: 'Java Programming'
Replaced String: Jello
Is str1 is equal to str2?: false
```

3.11 एक्सेप्शन (Exception) हैंडलिंग

जावा में Exception Handling का उपयोग प्रोग्राम को रनटाइम पर आने वाली त्रुटियों (errors) से सुरक्षित रखने के लिए किया जाता है। जब प्रोग्राम में कोई त्रुटि आती है, तो वह एक exception उत्पन्न करता है। अगर इन exceptions को सही से हैंडल नहीं किया गया, तो प्रोग्राम अचानक बंद हो सकता है।

जावा में Exception Handling के लिए मुख्य रूप से चार कीवड्र्स होते हैं:

* **try:** कोड के उस भाग को try ब्लॉक में रखा जाता है जहाँ संभावित exceptions हो सकते हैं।
* **catch:** catch ब्लॉक में exception को पकड़ा जाता है और उसे हैंडल किया जाता है।
* **finally:** finally ब्लॉक का उपयोग ऐसे कोड के लिए होता है, जिसे exceptions होने या न होने की स्थिति में भी चलना चाहिए।
* **throw:** इसका उपयोग मैन्युअली exception उत्पन्न करने के लिए किया जाता है।

Example 1: ArithmeticException (0 से विभाजन)

यह उदाहरण दिखाता है कि 0 से विभाजन पर exception कैसे उत्पन्न होता है और उसे कैसे हैंडल किया जाता है।

```
public class ExceptionExample {
public static void main(String[] args) {
try {
int num1 = 10;
int num2 = 0;
int result = num1 / num2; // यहाँ ArithmeticException उत्पन्न होगा
```

```
System.out.println("Result: " + result);
} catch (ArithmeticException e) {
System.out.println("Error: Division by zero is not allowed.");
}
}
}
```

Output:

```
Error: Division by zero is not allowed.
```

Example 2: ArrayIndexOutOfBoundsException

इस उदाहरण में, हम एक ऐसे index को एक्सेस करने की कोशिश करते हैं जो array में मौजूद नहीं है।

```
public class ArrayExceptionExample {
public static void main(String[] args) {
try {
int[] numbers = {1, 2, 3};
System.out.println(numbers[5]); // यहाँ ArrayIndexOutOfBoundsException
        उत्पन्न होगा
} catch (ArrayIndexOutOfBoundsException e) {
System.out.println("Error: Array index is out of bounds.");
}
}
}
```

Output:

```
Error: Array index is out of bounds.
```

Example 3: finally ब्लॉक का उपयोग

finally ब्लॉक का उपयोग उस कोड के लिए किया जाता है जो exception होने या न होने पर भी

अवश्य चलेगा। इसे resources (जैसे फाइलें या डेटाबेस कनेक्शन) को बंद करने के लिए उपयोग किया जा सकता है।

```
public class FinallyExample {
public static void main(String[] args) {
try {
int num = 10 / 2;
System.out.println("Result: " + num);
} catch (ArithmeticException e) {
System.out.println("Error: Division by zero.");
} finally {
System.out.println("This is the finally block.");
}
}
}
```

Output:

```
Result: 5
This is the finally block.
```

Example 4: throw का उपयोग

throw का उपयोग मैन्युअली exception उत्पन्न करने के लिए किया जाता है। मान लीजिए, हमें उम्र चेक करनी है और अगर उम्र 18 से कम है, तो एक custom exception फेंकना है।

```
public class ThrowExample {
public static void checkAge(int age) {
if (age < 18) {
throw new ArithmeticException("Not eligible to vote");
} else {
System.out.println("Eligible to vote.");
}
```

```
}
public static void main(String[] args) {
try {
checkAge(15);
} catch (ArithmeticException e) {
System.out.println("Error: " + e.getMessage());
}
}
}
```

Output:

```
Error: Not eligible to vote
```

संक्षेप में

* **try:** संभावित त्रुटि वाले कोड को इसमें रखें।
* **catch:** exception को कैच और हैंडल करें।
* **finally:** कोड के अंतिम हिस्से को इसमें रखें जो हमेशा चलेगा।
* **throw:** मैन्युअली exception फेंकने के लिए उपयोग करें।

Exception Handling प्रोग्राम को robust और errors से सुरक्षित बनाता है, जिससे प्रोग्राम बिना रुके चल सके और errors को सही से मैनेज किया जा सके।

3.12. डेटाबेस कनेक्टिविटी (Database Connectivity)

Database Connectivity का मतलब है कि हम अपने जावा प्रोग्राम को किसी डेटाबेस के साथ जोड़ सकते हैं ताकि डेटा को पढ़ सकें, जोड़ सकें, अपडेट कर सकें या डिलीट कर सकें (insert, update, select, delete)। इस प्रक्रिया में JDBC (Java Database Connectivity) API का उपयोग किया जाता है।

यहाँ हम MySQL डेटाबेस का उपयोग करेंगे और समझेंगे कि कैसे जावा प्रोग्राम से डेटा बेस को कनेक्ट किया जा सकता है।

1. **JDBC कनेक्शन सेटअप के लिए आवश्यकताएँ**

* **JDBC Driver:** JDBC ड्राइवर एक जावा लाइब्रेरी (जैसे mysql-connector-java.jar) है जो जावा प्रोग्राम और डेटाबेस के बीच संचार करता है। इसे अपने प्रोजेक्ट में जोड़ना आवश्यक है।

* **डेटाबेस और तालिका तैयार करें:** सबसे पहले, आपको MySQL जैसे डेटाबेस में एक डेटाबेस और एक टेबल बनानी होगी। मान लीजिए, हमारे डेटाबेस का नाम school है और उसमें students नामक एक टेबल है।

डेटाबेस में students टेबल बनाना

```
CREATE DATABASE school;
USE school;
CREATE TABLE students (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(50),
age INT,
grade VARCHAR(5)
);
```

उदाहरण: जावा में Database Connectivity

हम यहाँ एक साधारण उदाहरण देखेंगे, जहाँ हम डेटाबेस से कनेक्शन बनाएंगे और students टेबल से डेटा प्राप्त करेंगे।

* **JDBC ड्राइवर को जोड़ना**

अपने प्रोजेक्ट में mysql-connector-java.jar जोड़ें, जो MySQL डेटाबेस के लिए JDBC ड्राइवर का काम करेगा।

* **जावा कोड से डेटाबेस से कनेक्ट करना और डेटा प्राप्त करना**

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
```

```
import java.sql.Statement;
public class DatabaseExample {
public static void main(String[] args) {
// डेटाबेस यूआरएल, यूज़रनेम और पासवर्ड सेट करना
String url = "jdbc:mysql://localhost:3306/school"; // school आपका डेटाबेस नाम है
String username = "root"; // MySQL यूज़रनेम
String password = "password"; // MySQL पासवर्ड
try {
// डेटाबेस से कनेक्शन बनाना
Connection connection = DriverManager.getConnection(url, username, password);
System.out.println("Database connected successfully!");
// SQL स्टेटमेंट बनाना
String query = "SELECT * FROM students";
// स्टेटमेंट बनाना और SQL क्वेरी को एग्जीक्यूट करना
Statement statement = connection.createStatement( );
ResultSet resultSet = statement.executeQuery(query);
// डेटा को प्रिंट करना
while (resultSet.next( )) {
int id = resultSet.getInt("id");
String name = resultSet.getString("name");
int age = resultSet.getInt("age");
String grade = resultSet.getString("grade");
System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age + ", Grade: " + grade);
}
// कनेक्शन को बंद करना
```

```
resultSet.close( );
statement.close( );
connection.close( );
} catch (Exception e) {
e.printStackTrace( );
}
}
}
```

कोड का विवरण

* **Connection Creation:** DriverManager.getConnection( ) का उपयोग करके डेटाबेस से कनेक्ट करते हैं। इसमें डेटाबेस URL, यूज़रनेम और पासवर्ड पास किए जाते हैं।

* **SQL Query Execution:** Statement ऑब्जेक्ट का उपयोग करके SQL क्वेरी बनाई जाती है, और executeQuery( ) मेथड से इसे एग्जीक्यूट किया जाता है।

* **ResultSet Processing:** ResultSet का उपयोग करके क्वेरी के परिणामों को पढ़ा जाता है। resultSet.next( ) का उपयोग करके एक-एक करके हर पंक्ति को एक्सेस करते हैं और डेटा प्रिंट करते हैं।

* **Close Connection:** कनेक्शन और अन्य ऑब्जेक्ट्स को क्लोज करना ज़रूरी है ताकि रिसोर्सेज फ्री हो सकें।

डायग्राम: जावा डेटाबेस कनेक्टिविटी (JDBC) फ्लोचार्ट

आउटपुट

मान लीजिए, students टेबल में डेटा कुछ इस प्रकार है:

id	name	age	grade
1	Raj	16	A
2	Simran	17	B+
3	Aman	16	A-

SCERT DELHI

आउटपुटः

```
Database Connected Successfully!
ID: 1, Name: Raj, Age: 16, Grade: A
ID: 2, Name: Simran, Age: 17, Grade: B+
ID: 3, Name: Aman, Age: 16, Grade: A-
```

संक्षेप में

* डेटाबेस ड्राइवर जोड़ें - mysql-connector-java.jar
* डेटाबेस URL, यूज़रनेम, पासवर्ड सेट करें
* Connection, Statement और ResultSet ऑब्जेक्ट्स का उपयोग करें
* डेटा प्राप्त करें, प्रिंट करें और कनेक्शन बंद करें

जावा (java) में NetBeans IDE का उपयोग करके डेटाबेस से कनेक्ट करना सरल है। यहाँ हम एक उदाहरण के रूप में MySQL डेटाबेस से कनेक्ट करना देखेंगे। इस प्रक्रिया में हम JDBC (Java Database Connectivity) का उपयोग करेंगे।

इस अनुभाग में, हम MySQL डेटाबेस को जावा प्रोग्राम से कनेक्ट करना और वापस प्राप्त करना सीखेंगे डेटाबेस पर SQL क्वेरी निष्पादित करने से परिणाम मिलता है। डेटाबेस को जावा से कनेक्ट करना नेटबीन्स के साथ प्रोग्राम आसान है क्योंकि यह हमें सीधे MySQL सर्वर से जुड़ने की अनुमति देता है।

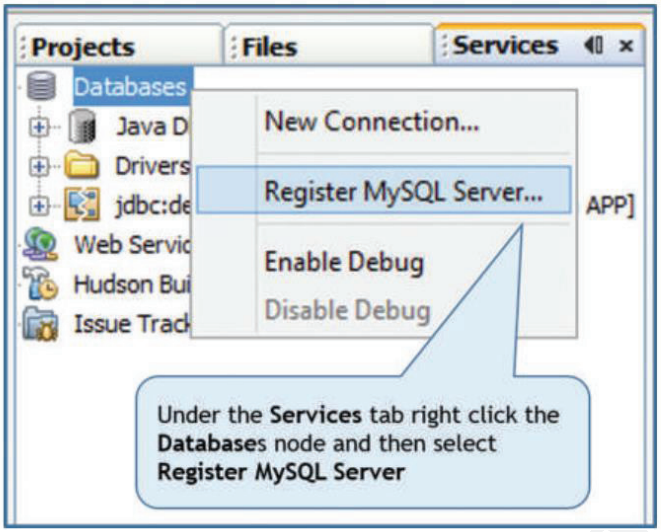
नेटबीन्स को MySQL सर्वर से कनेक्ट करना

हमें सबसे पहले MySQL सर्वर को पंजीकृत(Register) करने और कनेक्ट करने के लिए NetBeans को कॉन्फ़िगर करना होगा।

ऐसा करने के लिए नीचे दिए गए चरण हैं।

चरण 1: नेटबीन्स आईडीई के बाईं ओर स्थित सर्विसेज टैब पर क्लिक करें।

डेटाबेस नोड पर क्लिक करें और रजिस्टर MySQl सर्वर चुनें (3.13)

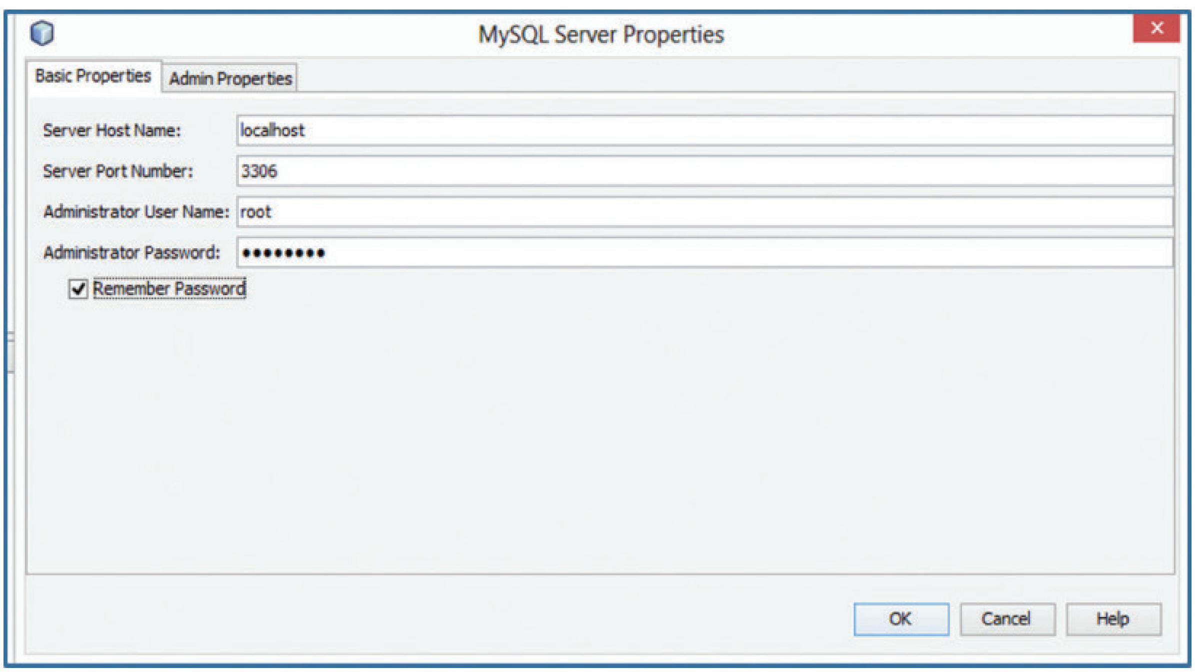


चित्र 3.13 : MySQL सर्वर पंजीकृत करें

चरण 2: खुलने वाले MySQL सर्वर प्रॉपर्टीज डायलॉग बॉक्स में टाइप करें

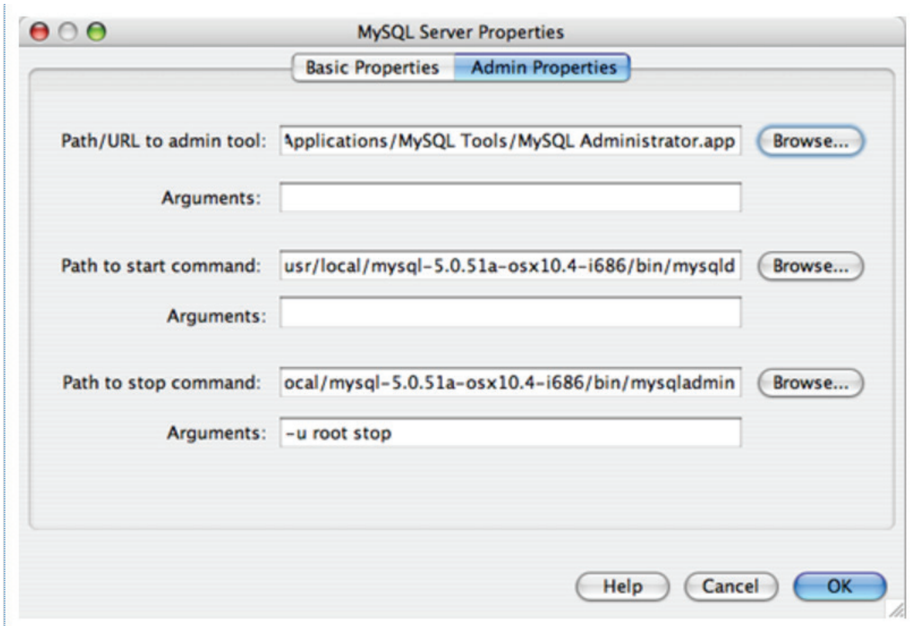
Administrator User नाम (यदि प्रदर्शित नहीं है)। Administrator Password भी टाइप करें अपने MySQlServer के लिए। पासवर्ड याद रखें चेकबॉक्स को चेक करें और ओके

चित्र पर क्लिक करें चित्र 3.14.



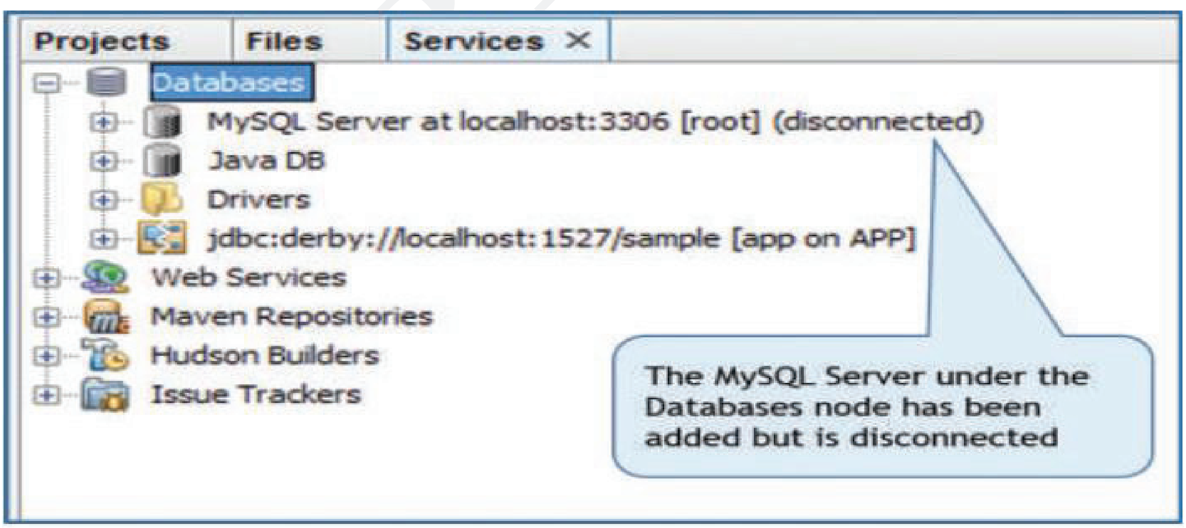
चित्र 3.14 MySQL सर्वर properties विंडो

चरण 3: उसी MySQL सर्वर प्रॉपर्टीज डायलॉग बॉक्स में एडमिन पर क्लिक करें properties टैब चित्र 3.15 के समान दिखना चाहिए।



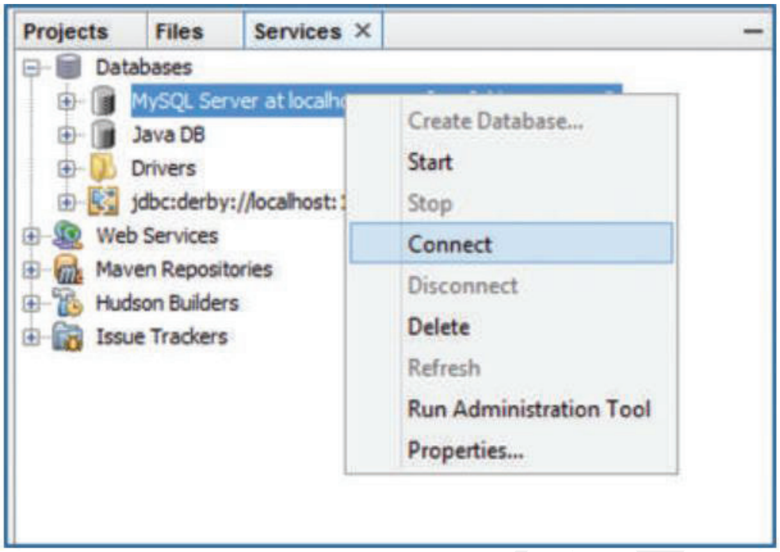
चित्र 3.15 MySQL सर्वर properties विंडो (admin properties टैब)

MySQLServer को अब सर्विसेज टैब में डेटाबेस नोड के अंतर्गत दिखना चाहिए नेटबीन्स आईडीई चित्र 3.16 हालाँकि, इसे डिस्कनेक्ट किया हुआ दिखाया गया है।



चित्र 3.16 MySQL सर्वर डेटाबेस में जोड़ा गया

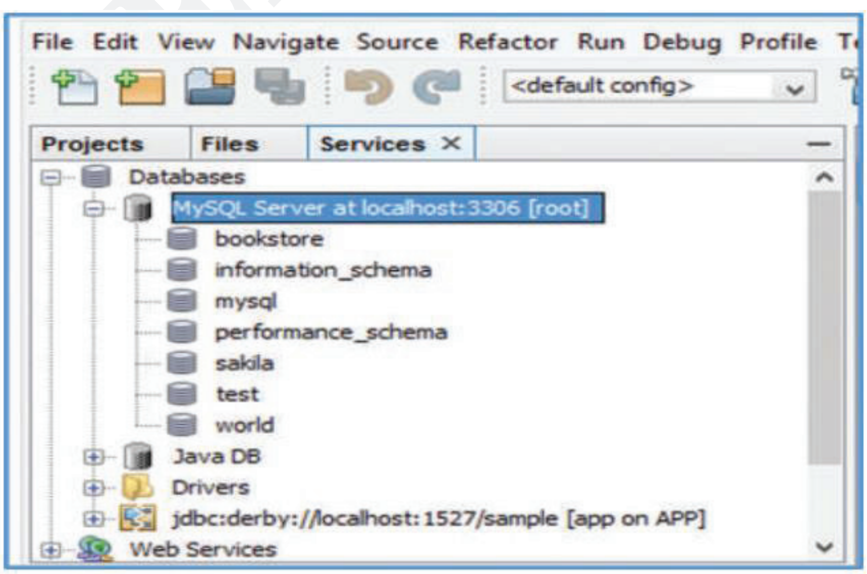
चरण 4: MySQL सर्वर को नेटबीन्स से कनेक्ट करने के लिए, डेटाबेस नोड के नीचे, दाईं ओर Localhost:3306 [रूट] (डिस्कनेक्ट) पर MySQL सर्वर पर क्लिक करें और चुनें



चित्र 3.17 कनेक्ट करें।

3.17 MySQL सर्वर से कनेक्ट करें

चरण 5: जब सर्वर कनेक्ट हो जाता है तो आपको [डिस्कनेक्टेड] को हटा हुआ देखना चाहिए Localhost पर MySQL सर्वर:3306 [रूट] डेटाबेस। आपको भी सक्षम होना चाहिए सभी उपलब्ध देखने के लिए + चिह्न पर क्लिक करके MySQL सर्वर नोड का विस्तार करें MySQL डेटाबेस चित्र 3.18।



चित्र 3.18 MySQL डेटाबेस

नेटबीन्स लाइब्रेरीज़ में MySQL कनेक्टर JAR जोड़ना

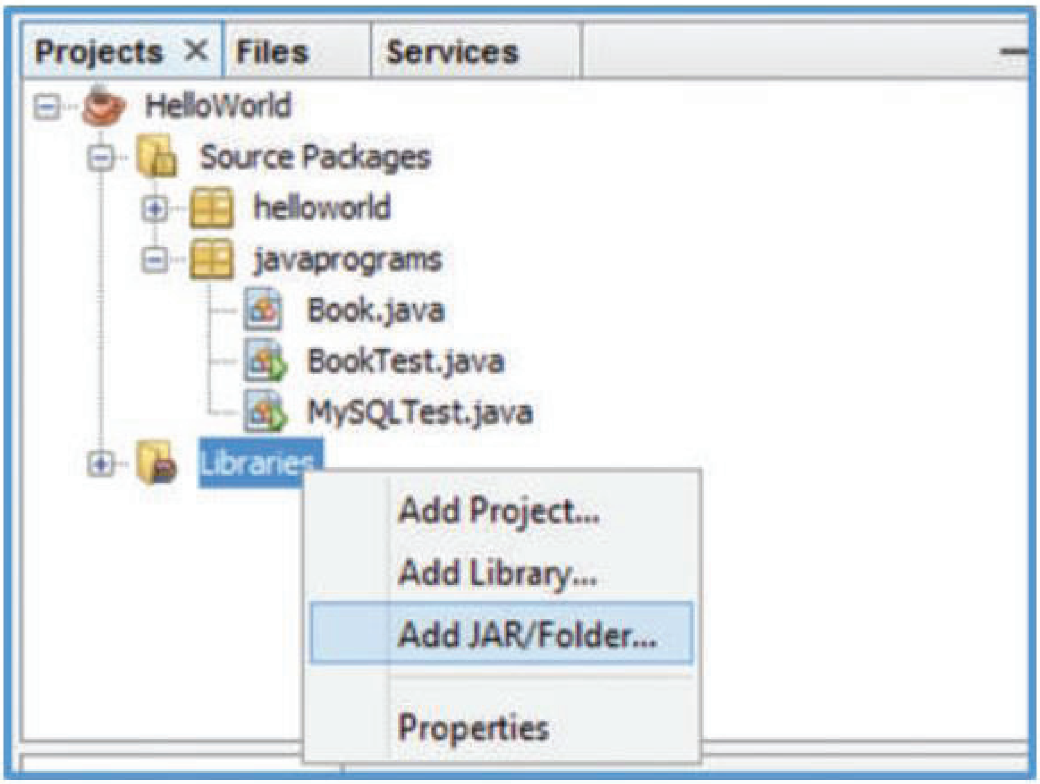
डेटाबेस से कनेक्ट करने के लिए जावा प्रोग्राम लिखने से पहले हमें इसे भी जोड़ना होगा

हमारे प्रोजेक्ट में लाइब्रेरीज़ के लिए mysql कनेक्टर JAR फ़ाइल। निम्नलिखित चरण आपको दिखाते हैं

ऐसा कैसे करें:

चरण 1: प्रोजेक्ट टैब के अंतर्गत, लाइब्रेरीज़ नोड पर राइट क्लिक करें और ADD JAR/ चुनें

फ़ोल्डर चित्र 3.19।

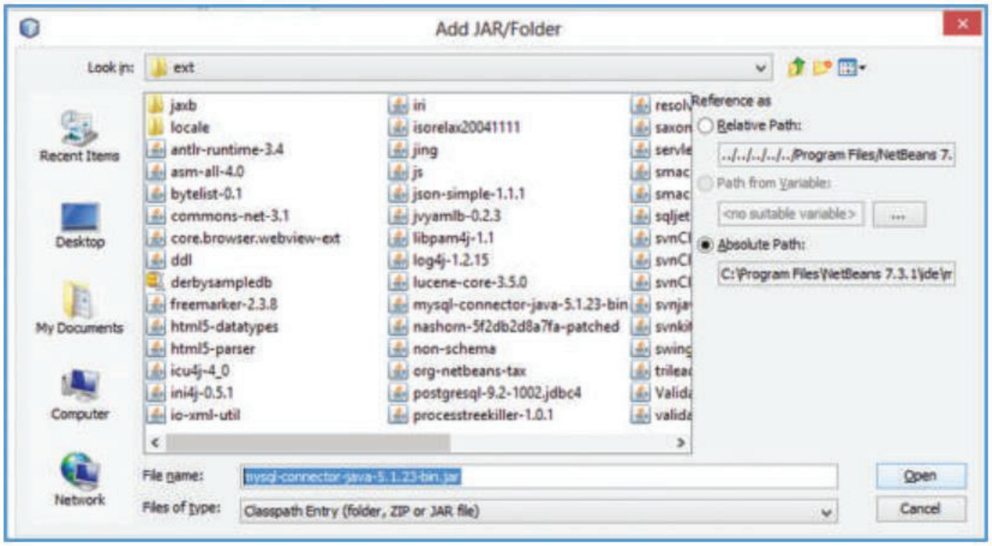


चित्र 3.19 लाइब्रेरीज़ में JAR/फ़ोल्डर जोड़ें

चरण 7: दिखाई देने वाले JAR/फ़ोल्डर जोड़ें संवाद बॉक्स में, अपने पर नेविगेट करें

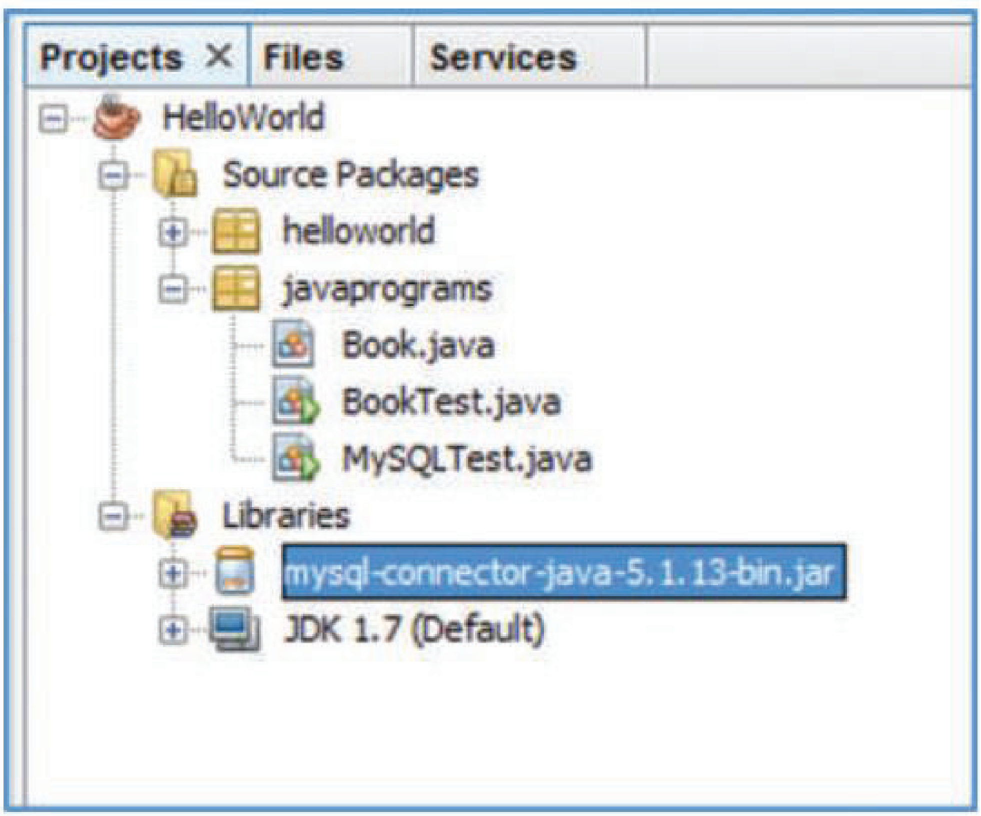
नेटबीन्स इंस्टालेशन फ़ोल्डर। फिर /ide/modules/ext फ़ोल्डर पर जाएँ और

mysql-कनेक्टर-java-5.1.23-bin.jarfile का चयन करें। ओपन चित्र 3.20 पर क्लिक करें।



चित्र 3.20 Mysql-कनेक्टर-java-5.1.23-bin.jar जोड़ें

अब, लाइब्रेरीज़ नोड (इसके बाईं ओर + चिह्न पर क्लिक करें), MySQL कनेक्टर जार का विस्तार करें नेटबीन्स लाइब्रेरीज़ चित्र 3.21 में जोड़ा जाना चाहिए था।



चित्र 3.21 Libraries

Step-by-Step Guide: Java से MySQL डेटाबेस कनेक्टिविटी NetBeans में

1. MySQL Database और टेबल बनाना

पहले MySQL में एक डेटाबेस और टेबल बनाएं। मान लें कि हमारा डेटाबेस का नाम school है और हमारे पास एक टेबल students है:

```
CREATE DATABASE school;
USE school;
CREATE TABLE students (
id INT PRIMARY KEY AUTO_INCREMENT,
name VARCHAR(50),
age INT,
grade VARCHAR(5)
);
```

2. जावा (Java) से डेटाबेस कनेक्शन

अब हम डेटाबेस से कनेक्ट करने और SQL निष्पादित करने के लिए एक जावा प्रोग्राम लिखने के लिए तैयार हैं।

डेटाबेस पर query. हमारे MySQL डेटाबेस में, हमने पहले ही एक डेटाबेस बना लिया है

इसके अंदर एक टेबल होती है जिसे student table कहा जाता है। इस तालिका के कॉलम हैं (id, name, age, grade)। हमने तालिका चित्र 12.10 में 3 पंक्तियाँ(row) भी सम्मिलित की हैं

मान लीजिए, students टेबल में डेटा कुछ इस प्रकार है:

id	name	age	grade
1	Raj	16	A
2	Simran	17	B+
3	Aman	16	A-

3. Java Code लिखना

नीचे एक साधारण कोड है जो डेटाबेस से कनेक्शन बनाता है और students टेबल से डेटा को पढ़ता है।

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class DatabaseConnectionExample {
public static void main(String[] args) {
// MySQL डेटाबेस कनेक्शन यूआरएल, यूज़रनेम और पासवर्ड
String url = "jdbc:mysql://localhost:3306/school"; // school आपका डेटाबेस
        नाम है
String username = "root"; // MySQL यूज़रनेम
String password = "password"; // MySQL पासवर्ड
try {
// कनेक्शन बनाना
Connection connection = DriverManager.getConnection(url, username,
        password);
System.out.println("Database Connected Successfully!");
// SQL क्वेरी बनाना
String query = "SELECT * FROM students";
// स्टेटमेंट बनाना और SQL क्वेरी को एग्जीक्यूट करना
Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery(query);
// डेटा को पढ़ना और प्रिंट करना
while (resultSet.next()) {
int id = resultSet.getInt("id");
String name = resultSet.getString("name");
int age = resultSet.getInt("age");
String grade = resultSet.getString("grade");
```

```
System.out.println("ID: " + id + ", Name: " + name + ", Age: " + age +
        ", Grade: " + grade);
}
// कनेक्शन को बंद करना
resultSet.close( );
statement.close( );
connection.close( );
} catch (Exception e) {
e.printStackTrace( );
}
}
}
```

4. कोड का विवरण

* **URL:** jdbc:mysql://localhost:3306/school - यह MySQL डेटाबेस के लिए URL है। localhost आपके कंप्यूटर का पता है, और school डेटाबेस का नाम है।
* **Username और Password:** अपने MySQL डेटाबेस के यूज़रनेम और पासवर्ड का उपयोग करें।
* **Connection Object:** DriverManager.getConnection( ) का उपयोग करके डेटाबेस कनेक्शन बनाया जाता है।
* **SQL Statement:** SQL क्वेरी को Statement ऑब्जेक्ट का उपयोग करके एग्जीक्यूट किया जाता है।
* **ResultSet:** executeQuery( ) मेथड का उपयोग करके डेटा प्राप्त किया जाता है और resultSet में संग्रहित किया जाता है।

5. प्रोग्राम चलाना

NetBeans में इस कोड को रन करें। अगर सब कुछ सही है, तो आपको कंसोल में students टेबल का डेटा प्रिंट होता दिखाई देगा।

Output (उदाहरण):

```
Database connected successfully
ID: 1, Name: Raj, Age: 16, Grade: A
ID: 2, Name: Simran, Age: 17, Grade: B+
ID: 3, Name: Aman, Age: 16, Grade: A-
```

6. Troubleshooting

* **SQL Exception:** अगर कनेक्शन में समस्या है, तो यह हो सकता है कि आपका यूज़रनेम, पासवर्ड, या डेटाबेस का नाम गलत हो।

* **Driver Not Found:** यह समस्या तब आती है जब mysql-connector-java.jar को सही से जोड़ने में गलती हो।

3.13. Assertions, Threads, and Wrapper Classes

1. Assertions

Assertion जावा में एक प्रोग्रामिंग फीचर है जो यह जांचने के लिए उपयोग होता है कि प्रोग्राम के दौरान कोई असंभावित स्थिति (unexpected condition) उत्पन्न तो नहीं हुई है। यह डिबगिंग के दौरान गलतियों को पकड़ने और प्रोग्राम को अधिक मजबूत बनाने में मदद करता है।

Key Points

* Assertion Syntax:

```
assert condition : "Error Message";
```

- **condition:** Boolean, जिसे true होना चाहिए।
- **Error Message:** Optional, जब condition false हो तो मैसेज दिखाता है।

* **Example:**

```
int age = 15;
assert age >= 18 : "Age must be 18 or above!";
```

- अगर age < 18, तो AssertionError फेंकता है।

* **Enable Assertions:**

Assertions डिफ़ॉल्ट रूप से बंद होती हैं। इसे रन करते समय -ea का उपयोग कर सक्रिय करें:

```
java -ea MyClass
```

* **Use Cases:**

- Debugging के दौरान preconditions और postconditions की जांच करना।
- यह सुनिश्चित करना कि लॉजिक सही तरीके से काम कर रहा है।

* **Limitations:**

- प्रोडक्शन कोड में उपयोग नहीं किया जाना चाहिए।
- उपयोगकर्ता इनपुट वैलिडेशन के लिए उपयुक्त नहीं।

Example

```
public class AssertionExample {
public static void main(String[] args) {
int num = -10;
assert num > 0 : "Number must be positive";
}
}
```

Output (अगर assertion false है):

```
Exception in thread "main" java.lang.AssertionError: Number must be positive
```

सारांश: Assertions का उपयोग डिबगिंग और लॉजिक चेक्स के लिए किया जाता है, लेकिन ये रनटाइम वैलिडेशन के लिए नहीं हैं।

2. Threads in Java

थ्रेड जावा में एक स्वतंत्र कार्यशील इकाई (independent working unit) है। यह प्रोग्राम का वह भाग होता है जो एक समय में कार्य कर सकता है। थ्रेड्स का उपयोग तब किया जाता है जब हमें प्रोग्राम में एक साथ कई कार्य (tasks) करने की आवश्यकता होती है।

Multithreading

मल्टीथ्रेडिंग का मतलब है, एक ही प्रोग्राम में एक साथ कई थ्रेड्स को चलाना। यह CPU के समय और संसाधनों का अधिकतम उपयोग करता है।

Java में थ्रेड क्यों उपयोग करें?

* साथ में कई कार्य करना (Concurrency): उदाहरण के लिए, एक वीडियो चलाते समय बैकग्राउंड में डाउनलोडिंग जारी रह सकती है।
* CPU का उपयोग बढ़ाना: थ्रेड्स के जरिए CPU का अधिकतम उपयोग किया जा सकता है।
* तेजी से प्रोग्राम निष्पादन: थ्रेड्स लंबे समय तक चलने वाले कार्यों को छोटे टुकड़ों में विभाजित करके तेजी से निष्पादित करते हैं।

Java में थ्रेड कैसे बनाते हैं?

जावा में थ्रेड बनाने के दो मुख्य तरीके हैं:

* Thread क्लास को सबक्लास बनाकर।
* Runnable इंटरफ़ेस को लागू करके।

Java में Thread बनाने के तरीके

* Thread क्लास को extends करना

```
class MyThread extends Thread {
public void run( ) {
System.out.println("Thread is running.");
}
}
public class ThreadExample {
public static void main(String[] args) {
MyThread t = new MyThread( );
t.start( ); // थ्रेड शुरू करें
}
}
```

Runnable इंटरफ़ेस को implement करना

```
class MyRunnable implements Runnable {
public void run( ) {
System.out.println("Thread is running.");
}
}
public class RunnableExample {
public static void main(String[] args) {
Thread t = new Thread(new MyRunnable( ));
t.start( ); // थ्रेड शुरू करें
}
}
```

Thread के मुख्य मेथड्स

* start( ): थ्रेड को शुरू करता है।
* run( ): थ्रेड का कोड निष्पादित करता है।
* sleep(ms): थ्रेड को कुछ समय के लिए रोकता है।
* join( ): एक थ्रेड को दूसरे के समाप्त होने तक रोकता है।
* isAlive( ): चेक करता है कि थ्रेड अभी भी रनिंग स्टेट में है या नहीं।

Multithreading Example

```
class FirstThread extends Thread {
public void run( ) {
for (int i = 1; i <= 3; i++) {
System.out.println("First Thread: " + i);
}
}
}
```

```
class SecondThread extends Thread {
public void run( ) {
for (int i = 1; i <= 3; i++) {
System.out.println("Second Thread: " + i);
}
}
}
public class MultiThreadExample {
public static void main(String[] args) {
FirstThread t1 = new FirstThread( );
SecondThread t2 = new SecondThread( );
t1.start( );
t2.start( );
}
}
```

Output (अनियमित हो सकता है):

```
First Thread: 1
Second Thread: 1
First Thread: 2
Second Thread: 2
First Thread: 3
Second Thread: 3
```

सारांश (Summary)

* **थ्रेड:** स्वतंत्र कार्य की इकाई।
* **मल्टीथ्रेडिंग:** एक साथ कई कार्य करना।
* **महत्व:** गति, कुशल संसाधन उपयोग, और बेहतर उपयोगकर्ता अनुभव।

रियल-लाइफ उदाहरण

* एक म्यूजिक प्लेयर में गाना चलना (Thread 1) और लिरिक्स स्क्रॉल करना (Thread 2)।

3 Wrapper Classes in Java

Wrapper Classes जावा में प्रिमिटिव डेटा प्रकारों (जैसे int, float, आदि) को ऑब्जेक्ट के रूप में उपयोग करने की अनुमति देती हैं।

Why Wrapper Classes?

* ऑब्जेक्ट्स का उपयोग करने की ज़रूरत होती है जब हम Collections, Generics, या अन्य ऑब्जेक्ट-आधारित फीचर्स के साथ काम करते हैं।
* **उदाहरणः** int प्रिमिटिव है, लेकिन इसे Integer क्लास द्वारा ऑब्जेक्ट में बदला जा सकता है।

Primitive Types और उनके	Wrapper Classes
Primitive Type	Wrapper Class
int	Integer
char	Character
float	Float
double	Double
long	Long
short	Short
byte	Byte
boolean	Boolean

Features of Wrapper Classes

* Autoboxing: प्रिमिटिव को ऑटोमेटिकली ऑब्जेक्ट में बदलना।

```
int a = 10;
Integer obj = a;  // Autoboxing
```

* Unboxing: ऑब्जेक्ट को प्रिमिटिव में बदलना।

```
Integer obj = 20;
int b = obj;  // Unboxing
```

Example Code

```
public class WrapperExample {
public static void main(String[] args) {
// Autoboxing
int num = 100;
Integer obj = num;  // int से Integer में
System.out.println("Integer Object: " + obj);
// Unboxing
Integer obj2 = 200;
int num2 = obj2;  // Integer से int में
System.out.println("Primitive int: " + num2);
}
}
```

Output:

```
Integer Object: 100
Primitive int: 200
```

Benefits of Wrapper Classes

* Collections के साथ उपयोगी: जैसे ArrayList<Integer> में।
* Generics के साथ आवश्यक: प्रिमिटिव्स Generics में काम नहीं करते।
* Utility Methods: Wrapper Classes में डेटा कन्वर्ज़न के लिए उपयोगी मेथड्स होते हैं।
* **उदाहरण:** Integer.parseInt("123") स्ट्रिंग को int में बदलता है।

अभ्यास:

Q1. जावा बाइटकोड क्या है?

Q2. एक क्लास और एक ऑब्जेक्ट के बीच का अंतर एक उदाहरण के साथ समझाइए।

Q3. कंस्ट्रक्टर क्या है? इसका उपयोग क्यों किया जाता है?

Q4. जावा में अपवाद (Exception) को कैसे हैंडल किया जाता है?

Q5. जावा में थ्रेड्स को कैसे बनाया जा सकता है? एक उदाहरण के साथ संक्षेप में समझाइए।

प्रयोग अभ्यास (Lab Exercises):

Q1. जावा में एक प्रोग्राम लिखें जो निम्न सूत्र को लागू करे:

क्षेत्रफल = लंबाई × चौड़ाई × ऊंचाई

Q2. जावा में एक प्रोग्राम लिखें जो निम्नलिखित अभिव्यक्तियों का परिणाम निकाले:

(मान लें कि a = 20 और b = 30)

(i) a % b

(ii) a /= b

(iii) (a + b * 100) / 10

(iv) a && b

(v) a++

Q3. जावा में एक प्रोग्राम लिखें जो एक ऐरे के हर वैकल्पिक (alternate) संख्या का वर्ग (square) प्रिंट करे।

Q4. जावा में एक प्रोग्राम लिखें जो एक Triangle नामक क्लास बनाए, जिसमें निम्नलिखित डेटा मेंबर्स हों:

* base (आधार), height (ऊंचाई), area (क्षेत्रफल) - इनका प्रकार double हो।
* color (रंग) - इसका प्रकार String हो।

इस क्लास में:

* base, height, और color के लिए getter और setter मेथड्स लिखें।
* एक मेथड compute_area( ) बनाएं, जो क्षेत्रफल की गणना करे।

SCERT DELHI

* Triangle क्लास के दो ऑब्जेक्ट्स बनाएं, उनके क्षेत्रफल की गणना करें और उनकी तुलना करें।

* अगर दोनों ऑब्जेक्ट्स का क्षेत्रफल और रंग समान हो, तो "Matching Triangles" प्रिंट करें, अन्यथा "Non Matching Triangles" प्रिंट करें।

Q5. जावा में एक प्रोग्राम लिखें जो उपयोगकर्ता को शून्य से विभाजन (divide by zero) अपवाद को संभालने में सक्षम बनाए।

❖❖❖

SCERT DELHI

वर्क इंटीग्रेटेड लर्निंग आईटी–डीएमए

इकाई 4

सीखने के प्रतिफल

विद्यार्थी

* डेटाबेस के संभावित कार्य क्षेत्रों की पहचान को समझते हैं।
* केस स्टडी करके एक शॉपिंग वेबसाइट को समझते हैं।
* नेटबीन्स आईडीई(NetBeans IDE) के इंस्टॉलिंग और स्टार्टिंग को समझते हैं।

परिचय (Introduction)

चित्र 4. 1- Data Migration Assistant

डीएमए(Data Migration Assistant) आपके लक्षित(target) वातावरण के लिए प्रदर्शन और विश्वसनीयता में सुधार की सिफारिश करता है और आपको अपने schema, data, और uncontained objects को अपने स्रोत(source) सर्वर से अपने लक्ष्य(target) सर्वर पर ले जाने की अनुमति देता है।

चित्र 4. 2 - DBMS

आज के समय में डेटाबेस का उपयोग लगभग हर छेत्र में हो रहा हैं। जिसके उपयोग से आप डेटा को अर्थपूर्ण(meaningful) ढंग से व्यवस्थित(organize) कर सकते हो।

जरूरत पड़ने पर ये आपको जानकारी तेजी से देने में सहायक होता हैं। अभी तक आपने SQL में DML का उपयोग करके डेटाबेस में डेटा बनाना(create), चुनना(select) और बदलना(modify) सीखा। विभिन्न प्रकार की वेब ऐप्स डेटा प्रबंधन(management) करने के लिए डेटाबेस का उपयोग किया जाता हैं।

इस इकाई में, आप डेटाबेस प्रबंधन सिस्टम(DBMS) का उपयोग करने वाले विभिन्न कार्य के क्षेत्रों के बारे में जानेंगे। इसके अलावा, आप डेटाबेस प्रबंधन सिस्टम(DBMS) और JAVA का उपयोग करके शॉपिंग ऐप(application) बनाएंगे।

4.1 डेटाबेस के उपयोग से संबन्धित विभिन्न कार्य क्षेत्र

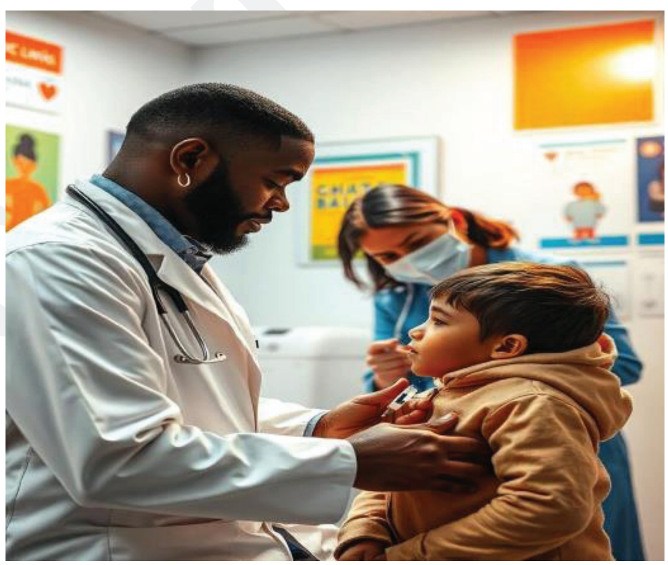
डेटाबेस प्रबंधन(management) सिस्टम का उपयोग कई डोमेन/क्षेत्रों(areas) में होता है जहां डेटा को सुरक्षित रखना और जल्दी निकालना जरूरी है। कुछ ऐसे डोमेन/क्षेत्रों(areas) हैं जहां लोग डेटाबेस का इस्तेमाल करते हैं:

1. ई-कॉमर्स (E-Commerce)

चित्र 4. 3 - ई-कॉमर्स

ऑनलाइन स्टोर उत्पाद सूचियों, ग्राहक विवरण, ऑर्डर और पैसे के आदान-प्रदान पर नज़र रखने के लिए डेटाबेस का उपयोग करते हैं।

2. स्वास्थ्य देखभाल (Healthcare)



चित्र 4. 4 - स्वास्थ्य देखभाल

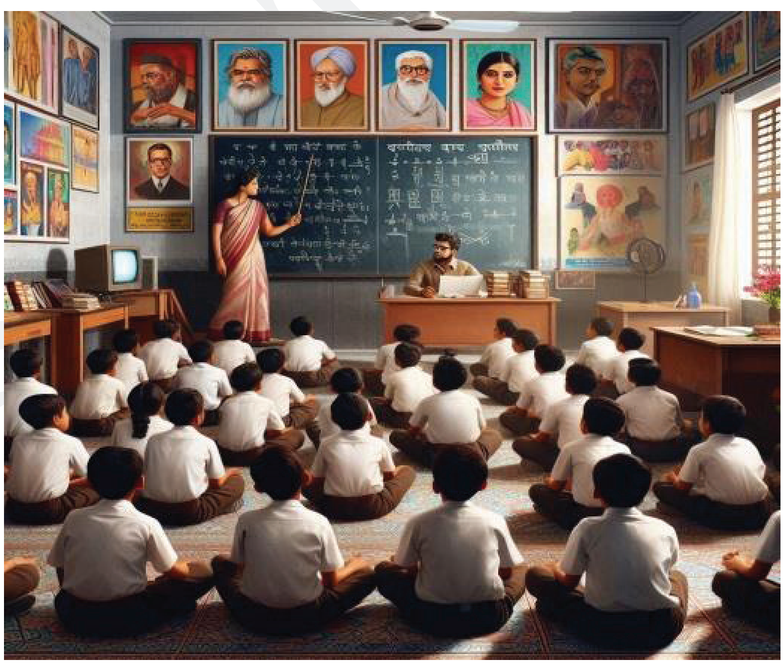
अस्पताल और क्लीनिक मरीज़ों का रिकॉर्ड रखने, नियुक्तियाँ तय करने, बिल संभालने और चिकित्सा आपूर्ति स्टॉक प्रबंधित करने के लिए डेटाबेस का उपयोग करते हैं।

3. वित्त और बैंकिंग (Finance and Banking)

चित्र 4. 5- वित्त और बैंकिंग

बैंक और अन्य धन-संबंधी व्यवसाय ग्राहक खातों, धन की आवाजाही, ऋण अनुरोधों और नियमों का पालन करने के लिए आवश्यक डेटा को प्रबंधित करने के लिए डेटाबेस का उपयोग करते हैं।

4. शिक्षा (Education)



चित्र 4. 6 - शिक्षा(Education)

स्कूल और कॉलेज छात्रों का रिकॉर्ड रखने, कक्षाओं के लिए साइन अप करने, ग्रेड रिकॉर्ड करने और शिक्षक जानकारी संग्रहीत करने के लिए डेटाबेस का उपयोग करते हैं।

5. दूरसंचार (Telecommunications)

चित्र 4. 7 - दूरसंचार(Telecommunications)

फ़ोन और इंटरनेट कंपनियाँ ग्राहक खातों, बिलिंग विवरण, कॉल लॉग और सेवा योजनाओं को संभालने के लिए डेटाबेस का उपयोग करती हैं।

6. सरकार (Government)

चित्र 4. 8 - सरकार(Government)

सरकारी कार्यालय नागरिक रिकॉर्ड, कर जानकारी सार्वजनिक सेवाओं और नियमों का पालन करने के लिए आवश्यक डेटा रखने के लिए डेटाबेस का उपयोग करते हैं।

7. यात्रा और आतिथ्य (Travel and Hospitality)

चित्र 4. 9 - यात्रा और आतिथ्य(Travel and Hospitality)

ट्रैवल एजेंसियां और होटल बुकिंग को संभालने, ग्राहकों को क्या पसंद है यह याद रखने और उनके पास कौन सी सेवाएं हैं, इस पर नज़र रखने के लिए डेटाबेस का उपयोग करते हैं।

8. एयरलाइंस और रेल्वे (Airlines & Railways)

चित्र 4. 10 - एयरलाइंस और रेल्वे (Airlines & Railways)

आगमन समय, प्रस्थान समय, किराया, यात्री क्षमता और बुकिंग की संख्या सहित उड़ान/ ट्रेन विवरण संग्रहीत करने के लिए। ऑनलाइन और ऑफलाइन दोनों तरह के आरक्षण को ट्रैक करने के लिए।

9. कंपनियों (Companies)

चित्र 4. 11 - कंपनियों(Companies)

कंपनियां कर्मचारियों का रिकॉर्ड रखने, वेतन संभालने, नियुक्तियों का प्रबंधन करने और कर्मचारी कितना अच्छा काम कर रहे हैं, इसका रिकॉर्ड रखने के लिए डेटाबेस का उपयोग करती हैं।

10. सोशल मीडिया(Social Media)

चित्र 4. 12 - सोशल मीडिया(Social Media)

सोशल मीडिया प्लेटफ़ॉर्म उपयोगकर्ता प्रोफ़ाइल, पोस्ट, टिप्पणियां और इंटरैक्शन को व्यवस्थित रखने के लिए डेटाबेस पर भरोसा करते हैं।

11. मनोरंजन(Entertainment)

चित्र 4. 13 - मनोरंजन(Entertainment)

स्ट्रीमिंग सेवाएँ और गेमिंग कंपनियाँ उपयोगकर्ता खातों, सामग्री लाइब्रेरी और उपयोगकर्ता इंटरैक्शन को प्रबंधित करने के लिए डेटाबेस का उपयोग करती हैं।

12. कृषि (Agriculture)

चित्र 4. 14 - कृषि (Agriculture)

कृषि संगठन फसल डेटा का प्रबंधन करने, मिट्टी की स्थिति की निगरानी करने, मौसम के पैटर्न का विश्लेषण करने और बाजार की कीमतों को ट्रैक करने के लिए डेटाबेस का उपयोग करते हैं।

विभिन्न एप्लिकेशन का उपयोग करके ऊपर बताए गए कार्य छेत्रों को उनकी आवशयकता अनुसार विकसित किया जा सकता हैं। जैसे -

* प्रबंधन (Management) एप्लिकेशन कॉलेजों और स्कूलों के लिए।
* ऑनलाइन आरक्षण (reservation) एप्लिकेशन एयरलाइंस, रेलवे, बसों, फिल्मों और होटलों के लिए।
* ऑनलाइन ट्यूटोरियल और मूल्यांकन के लिए।
* अस्पतालों के लिए प्रबंधन (Management) एप्लिकेशन।
* कंपनियों के लिए प्रबंधन (Management) एप्लिकेशन।
* प्रबंधन (Management) एप्लिकेशन बैंकिंग सिस्टम के लिए।
* ऑनलाइन शॉपिंग एप्लिकेशन।
* प्रबंधन (Management) एप्लिकेशन बिल भुगतान और सृजन (generation) के लिए।
* प्रबंधन (Management) एप्लिकेशन पुस्तकालय(Library) के लिए।
* प्रबंधन (Management) एप्लिकेशन (रियल एस्टेट के लिए)।
* प्रबंधन (Management) एप्लिकेशन होटल के लिए।
* प्रबंधन (Management) एप्लिकेशन दूरसंचार के लिए।
* प्रबंधन (Management) एप्लिकेशन बीमा के लिए।

अभ्यास प्रश्न 4.1

प्रश्न 1. डेटाबेस क्या है और इसका उपयोग किन-किन क्षेत्रों में होता है?

प्रश्न 2. SQL का उपयोग डेटाबेस में कैसे किया जाता है?

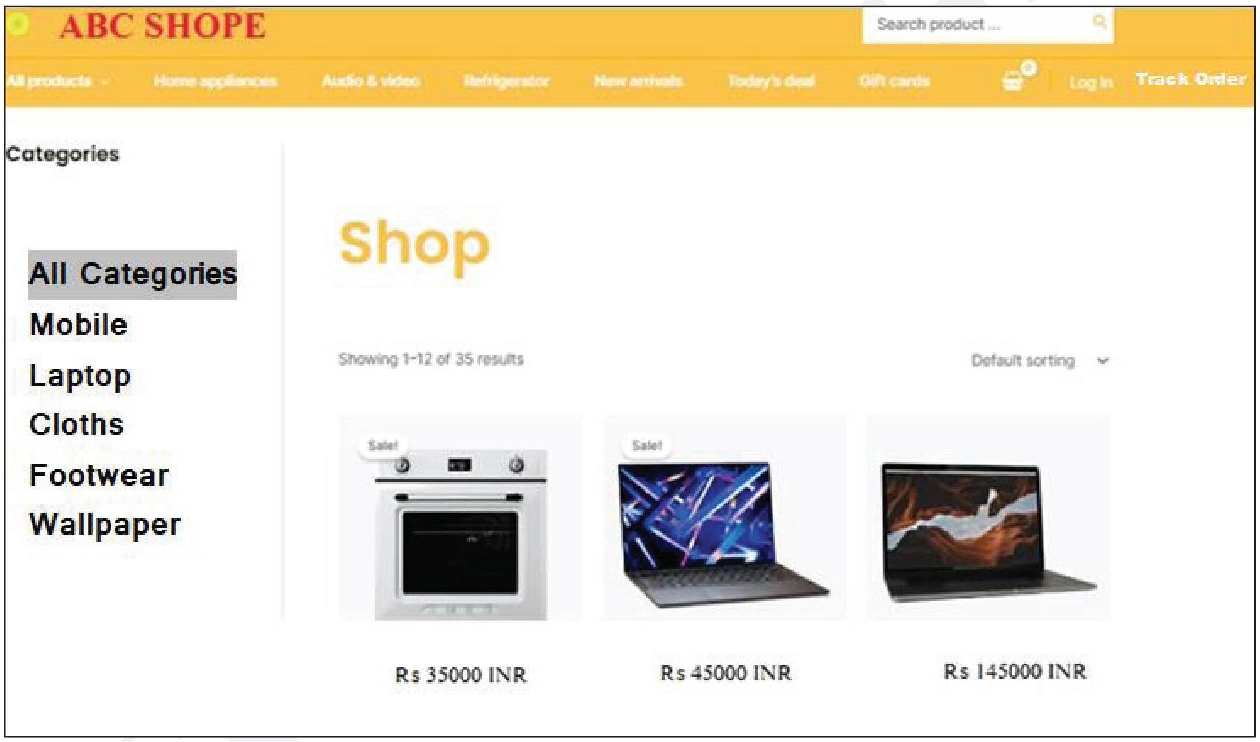
गतिविधि 4.1 (Activity 4.1)

एक सरल डेटाबेस बनाएं जिसमें ग्राहक की जानकारी (जैसे नाम, पता, फोन नंबर) हो। इसे किसी डेटाबेस प्रबंधन प्रणाली (DBMS) में कैसे लागू करेंगे?

SCERT DELHI

4.2 शॉपिंग वेबसाइट – केस स्टडी

इस केस स्टडी का उद्देश्य एक ई-कॉमर्स शॉपिंग वेबसाइट के लिए एक डेटाबेस डिज़ाइन करना है जो यूसरस को उत्पादों(product) को ब्राउज़ करने, अपने कार्ट में आइटम जोड़ने, ऑर्डर प्रबंधित करने और भुगतान संभालने की अनुमति देता है। डेटाबेस को उपयोगकर्ता खातों, उत्पाद(product) प्रबंधन(manage), ऑर्डर प्रोसेसिंग और इन्वेंट्री ट्रैकिंग का समर्थन करना चाहिए। यह शॉपिंग वेबसाइट यूसरस को 24 x 7 ऑनलाइन खरीदारी करने की अनुमित देती है। यह यूसरस को वस्तुओ की विस्तृत श्रृंखला जैसे पुस्तक, कपड़े, स्टेशनरी, रसोई, इलेक्ट्रोनिक आदि प्रदान करती है। ऑर्डर को कार्ट में जोडने के बाद अपने पते को सुनिश्चित करके ऑर्डर का भुगतान विभिन्न विकल्प (कैश ऑन डिलीवरी, क्रेडिट/डेबिट कार्ड या नेट बैंकिंग) के माध्यम पूरा कर सकते है।



चित्र 4. 15 – शॉपिंग वैबसाइट का इंटरफ़ेस

ऊपर दर्शाई गई वेबसाइट के बाएँ(Left) पैनल में विभिन्न श्रेणियों के उत्पादों को दर्शाया गया है। इसके विपरीत पैनल में कुछ उत्पादों की तस्वीरें हैं। सबसे ऊपरी पैनल Todays Deal, Gift Card, Login, Track Order आदि को दिखाता है।

4.2.1 Entitites Involved

शॉपिंग वैबसाइट में डेटाबेस बनाने के लिए शामिल प्रमुख एंटीटीस(entities) के लिए SQL क्रिएट टेबल स्टेटमेंट नीचे दिए गए हैं। प्रत्येक में प्राथमिक कुंजी(Primery Key) को रेखांकित किया गया है।

1. Customer Table

यह table कस्टमर की जानकारी को स्टोर करती है। इसमे कस्टमर की id, first name, email, password, phone number जैसी जानकारी स्टोर होगी।

योजना : Customer (customer_id, first_name, last_name, address, email, phone_number)

```
CREATE TABLE Customer (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    address VARCHAR(100) NOT NULL,
    email VARCHAR (100) UNIQUE NOT NULL,
    phone_number VARCHAR (15)
);
```

चित्र 4. 16 – Customer Table

2. Product Table

यह table प्रॉडक्ट की जानकारी को स्टोर करती है। इसमे प्रॉडक्ट की product id, name, description, price, category, stock quantity जैसी जानकारी स्टोर होगी।

योजना : Product (product_id, name, description, price, category, stock _ quantity)

```
CREATE TABLE Product (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    category_id INT,
    stock_quantity INT NOT NULL
);
```

चित्र 4. 17 – Product Table

3. Order Table

यह table ऑर्डर(order) की जानकारी को स्टोर करती है। इसमे ऑर्डर(order) की order id, customer id, order_date, status, category, total amount जैसी जानकारी स्टोर होगी।

योजना : Order (order_id, customer_id, order_date, status, total_amount)

```
CREATE TABLE `Order` (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(50) NOT NULL,
    total_amount DECIMAL (10, 2) NOT NULL
);
```

चित्र 4. 18 - Order Table

4. Cart Item Table

यह table कार्ट(cart) की जानकारी को स्टोर करती है। इसमे कार्ट(cart) की cart item id, cart id, product id, quantity, category, total amount जैसी जानकारी स्टोर होगी।

योजना : CartItem(cart_item_id, cart_id, product_id, quantity)

```
CREATE TABLE CartItem (
    cart_item_id INT PRIMARY KEY AUTO_INCREMENT,
    cart_id INT,
    product_id INT,
    quantity INT NOT NULL
);
```

चित्र 4. 19 - Cart Item Table

5. Payment Table

यह table पेमेंट(payment) की जानकारी को स्टोर करती है। इसमे पेमेंट(payment) की payment id, order id, amount, payment method, status जैसी जानकारी स्टोर होगी।

योजना : Payment (payment_id, order_id, amount, payment_method, status)

```
CREATE TABLE Payment (
    payment_id INT PRIMARY KEY AUTO_INCREMENT,
    order_id INT,
    amount DECIMAL(10, 2) NOT NULL,
    payment_method VARCHAR(50) NOT NULL,
    status VARCHAR (50) NOT NULL
);
```

चित्र 4. 20 - Payment Table

6. Address Table

यह table एड्रैस(Address) की जानकारी को स्टोर करती है। इसमे एड्रैस(Address) की address id, customer id, street, city, state, postal code, country जैसी जानकारी स्टोर होगी।

योजना : Address (address_id, customer_id, street, city, state, postal_code, country)

```
CREATE TABLE Address (
    address_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    street VARCHAR(255) NOT NULL,
    city VARCHAR(100) NOT NULL,
    state VARCHAR(100) NOT NULL,
    postal_code VARCHAR(20) NOT NULL,
    country VARCHAR(100) NOT NULL
);
```

चित्र 4. 21 – Address Table

7. Category Table

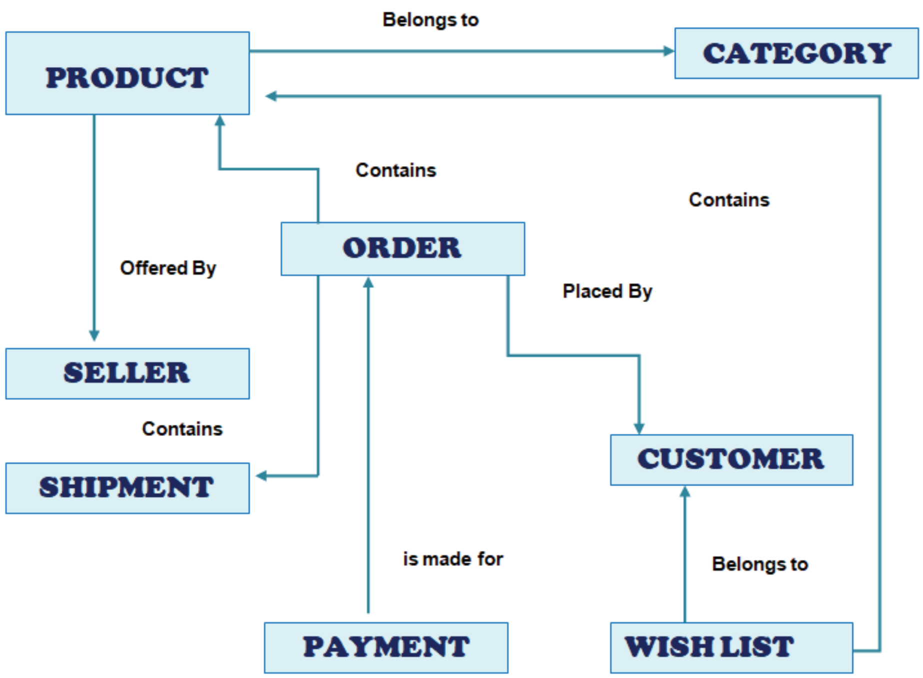
यह table कैटेगरी(category) की जानकारी को स्टोर करती है। इसमे कैटेगरी(category) की category id, category name, description जैसी जानकारी स्टोर होगी।

योजना : Category (category_id, category_name, description)

```
CREATE TABLE Category (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    category_name VARCHAR(100) UNIQUE NOT NULL,
    description VARCHAR(100)
);
```

चित्र 4. 22 – Category Table

अब, आप जांच करें कि ये entities एक-दूसरे से कैसे जुड़ी हुई हैं। यहाँ चित्र के माध्यम से अनौपचारिक (informal) तरीके से इन entities के बीच संबंधों को दर्शाया गया है।]



चित्र 4. 23 - Entities के बीच संबंधों को दर्शाता हुआ चित्र

गतिविधि 4.2 (Activity 4.2)

SQL का उपयोग करके एक टेबल बनाएं जिसमें ऑर्डर की जानकारी हो। इसमें ऑर्डर आईडी, ग्राहक आईडी, ऑर्डर तिथि और कुल राशि शामिल करें।

4.2.3 फंक्शनलिटी (Functionality)

अभी तक आपने entity के बारे में समझा, अब आप देखेगे की शॉपिंग एप्लिकेशन कैसे डेटाबेस का उपयोग और प्रबंधन(manage) कर सकती है?

नीचे कुछ महत्वपूर्ण कार्य दिए गए हैं जो ABC शॉपिंग वेबसाइट द्वारा किए जाते हैं:

सबसे पहले, सभी श्रेणियां(categories) जिनके उत्पाद(product) खरीदारी के लिए उपलब्ध कराए जाने हैं, ABC कंपनी द्वारा तय किए जाते हैं।

1 CATEGORY table में categories को पूरी तरह से स्टोर करने के लिए, class Category के फंकशन(function) insert_category का उपयोग कर सकते हैं। यदि कंपनी (company) ज्यादा categories को table में जोड़ना (add) चाहती हैं, तो वो इसी फंकशन (function) का उपयोग कर सकती हैं। इसी तरह, कंपनी delete_category और modify_category फ़ंक्शंस के माध्यम से श्रेणियों(categories) को हटाने(delete) या संपादित(modify)

करने के विकल्प चुनेगी। यदि ग्राहक वेबसाइट से किसी भी श्रेणी(category) का चयन करता है, तो उस श्रेणी(category) के सभी उत्पादों (products) को दिखाने के लिए display_category फ़ंक्शन(function) निष्पादित(execute) किया जाएगा। CATEGORY table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;
public class Category {
        public void insert_category (String categ_tuple[ ])
        {
                // Add functionality to insert a row of category
                // named categ_tuple.
                // SQL command to be used: insert
        }
        public void delete_category(int categ_id)
        {
                // Add functionality to delete a category with id categ_id.
                // SQL command to be used: delete
        }
        public void modify_category(int categ_id, String attr, String new)
        {
                // Add functionality to change value of attribute attr
                // of category with id categ_id to new.
        }
                // SQL command to be used: update
        public void display_category(int categ_id)
        {
                // Add functionality to display all the products in
                // category with id categ_id.
                // SQL command to be used: select
        }
}
```

चित्र 4. 24

2. अब उत्पादों(products) की सूची(list) के साथ संबंधित विवरण को तय करना, प्रत्येक श्रेणी(category) के लिए और उत्पाद(product) table में डालने के लिए class Product के फंकशन (function) insert_product का उपयोग करते हैं। हालाँकि, कंपनी बाद में समान फ़ंक्शन(function) को कॉल करके अधिक उत्पाद(product) को जोड़ सकती है। Delete_product और Modify_product फ़ंक्शंस (function) का उपयोग उत्पादों(products) को हटाने(delete) या संपादित(modify) करने के लिए आवश्यकता अनुसार किया जा सकता है। यदि ग्राहक वेबसाइट पर कोई उत्पाद(product) चुनता है, फंकशन(function) display_product

का उपयोग उत्पाद(product) के विवरण को प्रदर्शित(display) करने के लिए किया जा सकता है। PRODUCT table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;
    public class Product {
    public void insert_product (String prod_tuple[ ])
    {
        // Add functionality to insert a row of product named
        // prod_tuple.
        // SQL command to be used: insert
    public void delete_product (int prod_id)
    {
        // Add functionality to delete a product with id prod_id.
        // SQL command to be used: delete
    }
    public void modify_product(int prod_id, String attr, String new)
    {
        // Add functionality to change value of attribute attr
        // of product with id prod id to new.
        // SQL command to be used: update
    }
    public void display_product (int prod_id)
    {
        // Add functionality to display details of the products
        // with id prod id
        // SQL command to be used: select
    }
}
```

चित्र 4. 25

3. ऊपर शामिल उत्पादों(products) के लिए उत्पाद(product) की पेशकश(offer) करने वाले विक्रेता(seller) की seller_id है। यह आवशक हो जाता हैं की हम SELLER table में सभी विक्रेताओं(sellers) के बारे में जानकारी शामिल करे। इसको करने के लिए class Seller के फ़ंक्शन (function) insert_seller का उपयोग करते है। Delete_seller और Modify_seller फंकशन (function) का उपयोग विक्रेता (sellers) की जानकारी को हटाने(delete) या संपादित(modify) करने के लिए आवश्यकता अनुसार किया जा सकता है। यदि ग्राहक वेबसाइट पर कोई उत्पाद(product) चुनता है, फंकशन(function) display_seller का उपयोग उत्पाद(product) के विवरण को प्रदर्शित(display) करने के लिए किया जा सकता है। मान लें कि एक ही प्रकार के उत्पाद(product) हैं और एक से अधिक विक्रेताओं(sellers) द्वारा ऑफ़र किया गया हैं। ऐसे मामले में आपको सभी विक्रेताओं(sellers) को उनके विवरण(details) के

साथ सूचीबद्ध(list) करना होगा। यहां Product_id और Seller_id को composite key के रूप में घोषित करने से उद्देश्य पूरा होगा। SELLER table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;
public class Seller {
        public void insert_seller (String seller_tuple[ ]) {
        // Add functionality to insert a row of seller named
        // seller_tuple.
        // SQL command to be used: insert
        }
        public void delete_seller (int seller_id)
        {
        // Add functionality to delete a seller with id seller_id.
        // SQL command to be used: delete
        }
        public void modify_seller (int seller_id, String attr, String new)
        {
        // Add functionality to change value of attribute attr
        // of seller with id seller id to new.
        // SQL command to be used: update
        }
        public void display_seller (int seller_id)
        {
        // Add functionality to display details of the seller
        // with id seller_id
        // SQL command to be used: select
        }
}
```

चित्र 4. 26

4. जब भी कोई ग्राहक(customer) किसी शॉपिंग वेबसाइट पर जाता है, तो किसी भी product को order करने के लिए ग्राहक(customer) को अपना विवरण(details) देना होगा या लॉगिन(login) करना होगा। यदि वह पहले से ही शॉपिंग वैबसाइट पर उपयोगकर्ता(user) है तो वह लॉगिन(login) कर सकता है, इसके लिए चित्र में दिखाए गए इंटरफ़ेस का उपयोग करना होगा। यदि ग्राहक(customer) अभी तक पंजीकृत(registered) नहीं है और पंजीकृत(registered) प्राप्त करना चाहता है तो चित्र में दिए गए इंटरफ़ेस का उपयोग कर सकता है। ध्यान दे यह वैसा ही इंटरफ़ेस है, जैसा पंजीकृत(regis-tered) नहीं होने पर उपयोग किया जाता है। जिसमे ऑर्डर(order) देते समय उपयोगकर्ता अपना विवरण(details) देता हैं। इसमें उपयोगकर्ता(user) द्वारा निर्दिष्ट(specific) विवरण(details) होती हैं जोकि CUSTOMER table में ग्राहक(Customer) की class से फंकशन (function) insert_customer का उपयोग करके saved करता हैं। इन विवरणों

का उपयोग हर बार ग्राहक द्वारा ऑर्डर देने पर किया जाएगा। जैसे किसी उत्पाद(product) की शिपिंग करनी हैं तो निर्दिष्ट (specified) पते(address) का उपयोग डिलीवरी के लिए किया जा सकता है। ग्राहक चाहे तो किसी भी समय अपनी निजी(personal) जानकारी(details) बदलाव कर सकता है। जिसके लिए फ़ंक्शन(function) modify_customer का उपयोग करना होगा।

```
package shopping_application;
    public class Customer {
    public void insert_customer (String customer_tuple[ ])
    {
    // Add functionality to insert a row of customer named
    // customer_tuple.
    // SQL command to be used: insert
    }
    public void modify_customer(int customer_id, String attr, String new)
    {
    // Add functionality to change value of attribute attr
    // of customer with id customer id to new.
    // SQL command to be used: update
    }
}
```

चित्र 4. 27

5. जब भी कोई ग्राहक(customer) किसी उत्पाद(product) को वैबसाइट पर ब्राउज़ करते हैं तो कई उत्पाद(product) ऐसे होते हैं जिन्हे वो बाद में खरीदने(buy) की सोचते हैं। ऐसे में वो इन उत्पाद(product) को wish list में जोड़ सकते हैं। इस लक्ष्य को प्राप्त करने के लिए class Wish List के फ़ंक्शन(function) Insert_wish को कॉल किया जा सकता है। इसके अतिरिक्त, ग्राहक अपनी wish list से कुछ उत्पादों (products) को हटाना चाहता है तो उसे Delete_wish फ़ंक्शन(function) का उपयोग करना पड़ेगा। बाद में ग्राहक (customer) को अपने जोड़े गए उत्पादों(product) को देखना हैं तो वो display_wish फ़ंक्शन का उपयोग करेगे। WISH LIST table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;
        public class WishList{
        public void insert_wish (String wish_tuple[ ])
        {
        // Add functionality to insert a row of wish named
        // wish_tuple.
        // SQL command to be used: insert
        }
        public void delete_wish (int customer_id, String wish)
        {
        // Add functionality to delete a wish of customer with id
        // customer id.
        // SQL command to be used: delete
        }
        public void display_wish(int customer_id)
        {
        // Add functionality to display wish list of customer with
        // id customer_id.
        // SQL command to be used: select
        }
}
```

चित्र 4. 28

6. जब भी कोई ग्राहक(customer) किसी उत्पाद(product) को खरीदना(buy) चाहता हैं तो वो उसे शॉपिंग कार्ट(shopping cart) में जोड़ता है। यदि ग्राहक(customer) लॉगिन(login) नहीं है, तो उसे लॉगिन(login) होना होगा या अपनी विवरण(details) देने के लिए कहा जाएंगा। उसके बाद वह चित्र में दिखाएँ गए किसी भी उपलब्ध भुगतान(payment) मोड में से किसी एक मोड को चुन कर भुगतान कर सकते हैं। इसमें सभी भुगतान(payment) से संबन्धित विवरण(details) स्टोर होती हैं PAYMENT table में class Payment से insert_payment फंकशन (function) का उपयोग करके। यदि ग्राहक डिलीवरी पर नकद भुगतान(payment) करना चुनता है, तो Payment_date विशेषता उस तारीख पर सेट की जाती है जिस दिन उत्पाद(product) वितरित(deliver) किया जाता है। PAYMENT table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;
        public class Payment{
        public void insert_payment(String payment_tuple[ ])
        {
                // Add functionality to insert a row of payment named
                // payment_tuple.
                // SQL command to be used: insert
        }
}
```

चित्र 4. 29

7. सभी ऑर्डर (Order) का विवरण(details) स्टोर होता हैं table ORDER में, class Order के फंकशन(function) insert_order से जिसे चित्र में भी दर्शाया गया हैं। यदि कोई ग्राहक ऑर्डर वापस लेना चाहता है, तो वह ऑर्डर रद्द(cancel order) करने के विकल्प को चुनेगा। इस स्थिति में, फ़ंक्शन delete_order को कॉल(invoked) किया जाएगा। यदि उपयोगकर्ता अपने ऑर्डर का पता(track) लगाना चाहता है, तो वह order_id देकर ऐसा कर सकता है। इसके बाद, फ़ंक्शन(function) display_order ऑर्डर की वर्तमान स्थिति(current status) सहित उसके ऑर्डर के सभी विवरण(details) को प्रदर्शित(display) करेगा। ORDER table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;
        public class Order{
        public void insert_order (String order_tuple[ ])
        {
        // add functionality to insert a row of order named
        // crder_tuple.
        // SQL command to be used: insert
        }
        public void delete_order(int order_id)
        {
        // add functionality to delete order with id order_id.
        // SQL command to be used: delete
        }
        public void display_order (int order_id)
        {
        // Add functionality to display details of the order with
        // id order id.
        // SQL command to be used: select
        }
}
```

चित्र 4. 30

8. अंत में class Shipment के फ़ंक्शन(function) insert_shipment_details का उपयोग करके ऑर्डर shipment से संबंधित जानकारी table SHIPMENT में संग्रहीत(stored) की जाती है। SHIPMENT table को प्रबंधन(manage) करने के लिए class के ढांचे और उससे संबन्धित function को चित्र में दर्शाया गया हैं।

```
package shopping_application;

    public class Shipment{

    public void insert_shipment details (String order shipment[ ])

    {

    // Add functionality to insert a row of shipment named
    // order shipment.
    // SQL command to be used: insert
    }

}
```

चित्र 4. 31

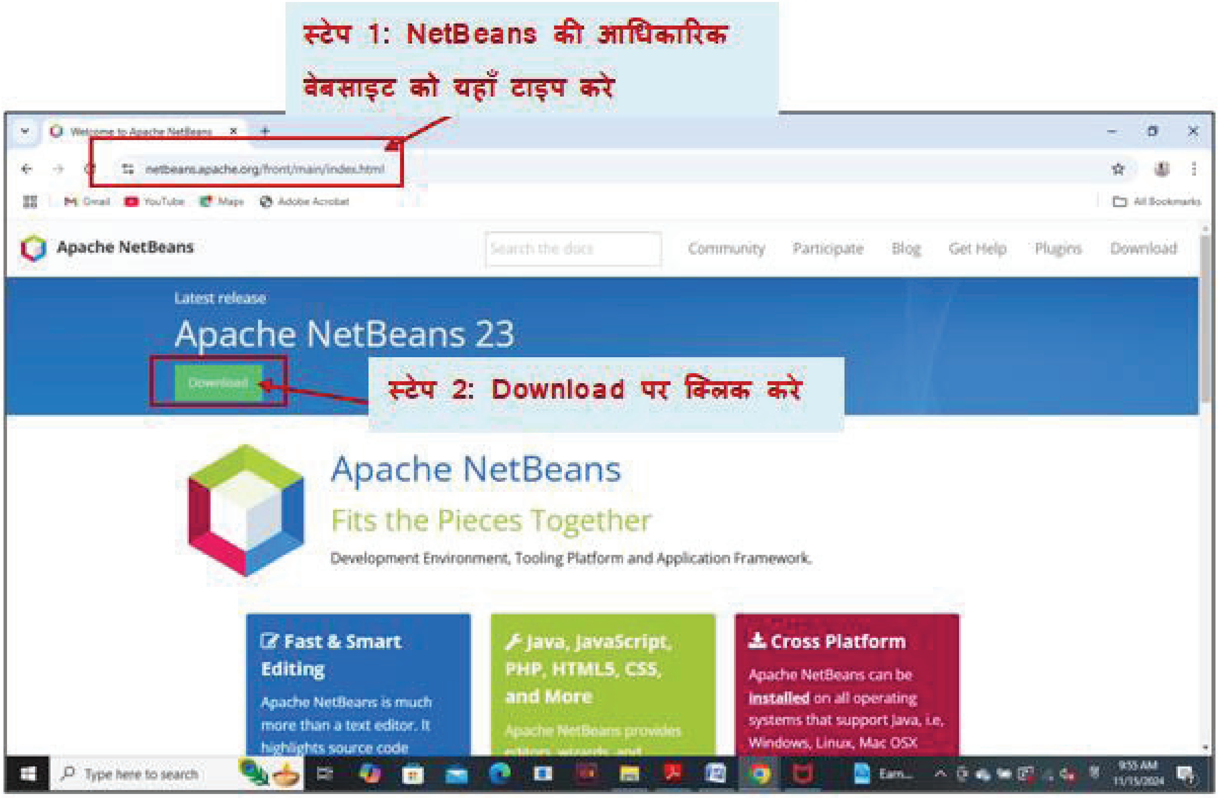
अभ्यास प्रश्न 4.2
प्रश्न 1. "Payment" टेबल में कौन सी जानकारी स्टोर होती है?
प्रश्न 2. शॉपिंग वेबसाइट में एंटिटी रिलेशनशिप क्या दर्शाती है?

4.3 शेषसंग्रह (Appendix) – ए

स्टेप 1: NetBeans IDE डाउनलोड करें

* **वेबसाइट पर जाएं:** अपने वेब ब्राउज़र में NetBeans की आधिकारिक वेबसाइट (https://netbeans.apache.org/front/main/index.html) पर जाएं।

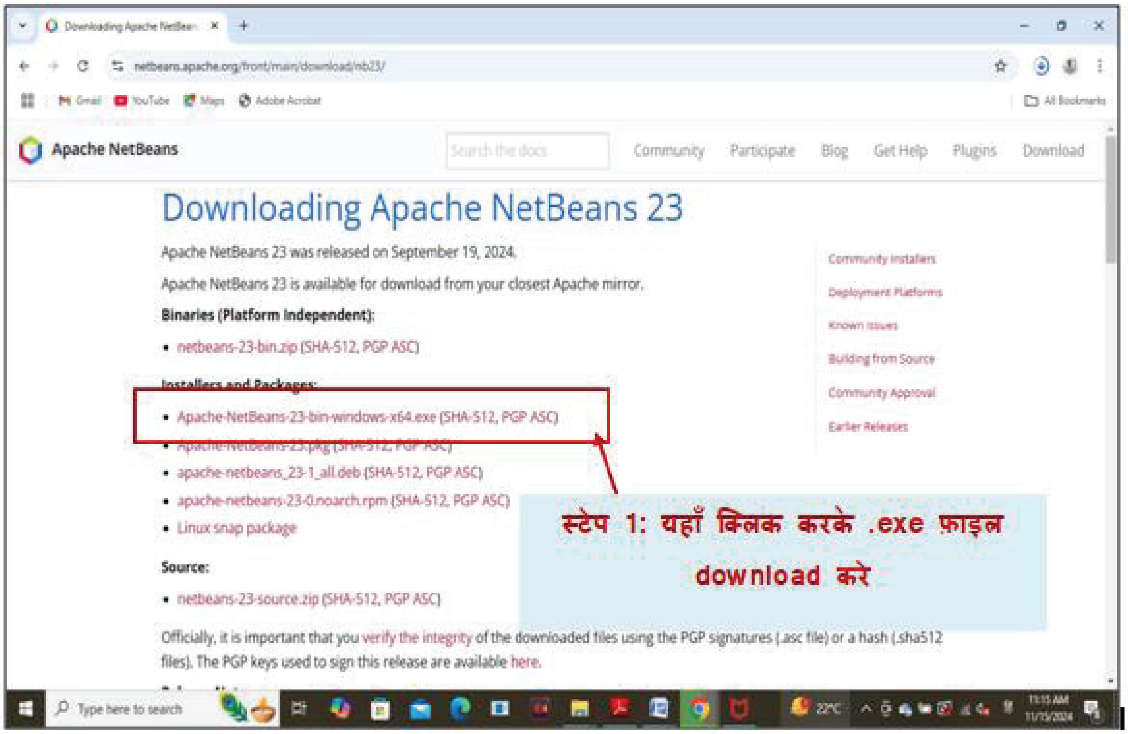
* **डाउनलोड करें:** "Download" सेक्शन में जाएं और अपने ऑपरेटिंग सिस्टम (Windows, macOS, or Linux) के लिए उपयुक्त(suitable) संस्करण(version) को डाउनलोड करें।



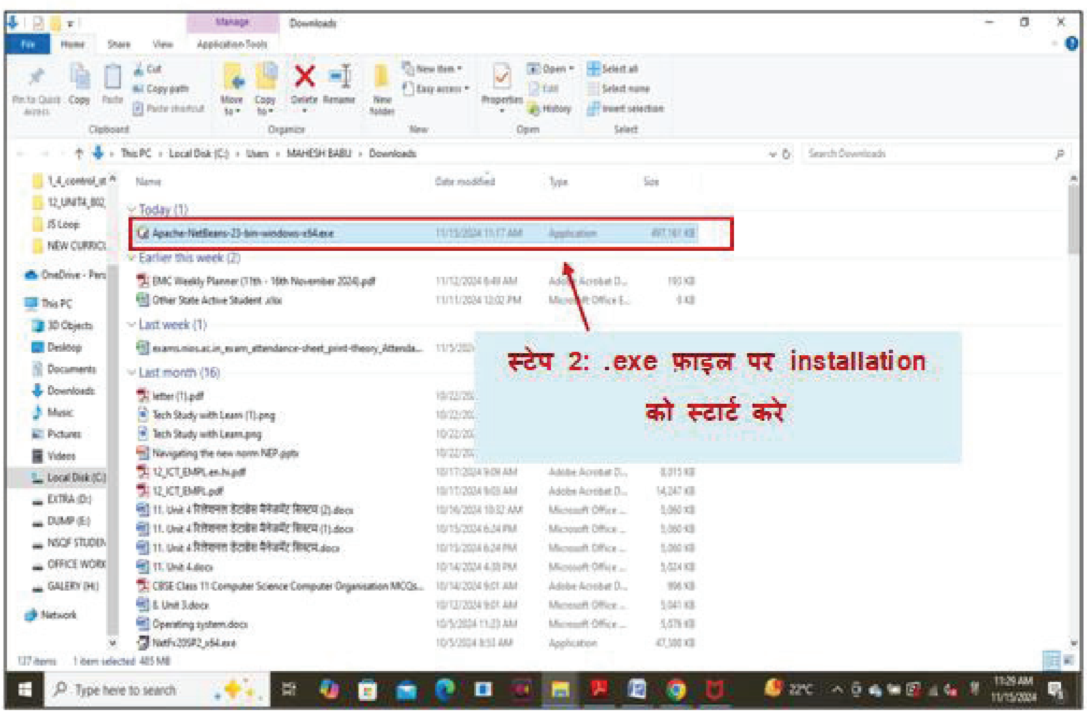
चित्र 4. 32

स्टेप 2: NetBeans IDE स्थापित करें

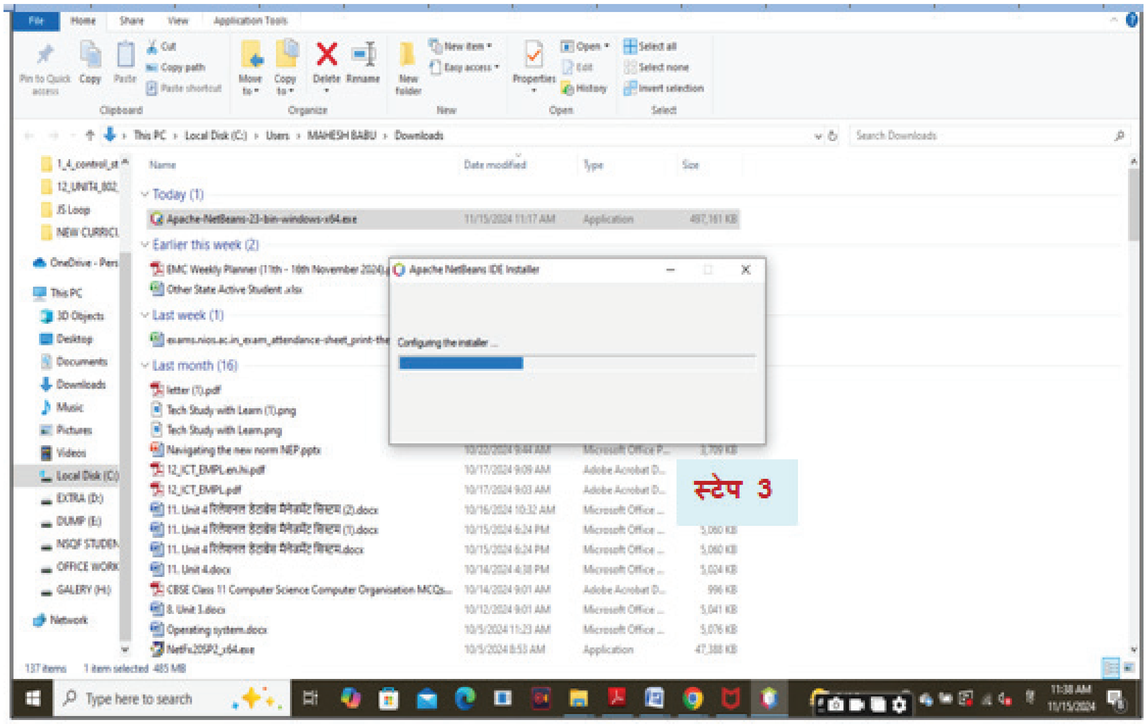
* **इंस्टॉलर चलाएँ:** डाउनलोड की गई फ़ाइल पर डबल-क्लिक करें। यदि आप विंडोज़ का उपयोग कर रहे हैं, तो यह एक .exe फ़ाइल होगी।
* **इंस्टॉलेशन प्रक्रिया:**
  - 216. इंस्टॉलेशन विज़ार्ड खुल जाएगा। "Next" पर क्लिक करें।
  - 217. लाइसेंस समझौते को पढ़ें और "I Accept" पर क्लिक करें।
  - 218. इंस्टॉलेशन स्थान चुनें या डिफ़ॉल्ट स्थान को स्वीकार करें।
  - 219. आवश्यक घटकों का चयन करें (जैसे, Java SE, PHP, C/C++ आदि) और "Next" पर क्लिक करें।
* **इंस्टॉलेशन शुरू करें:** "Install" पर क्लिक करें और प्रक्रिया पूरी होने का इंतजार करें।



चित्र 4. 33

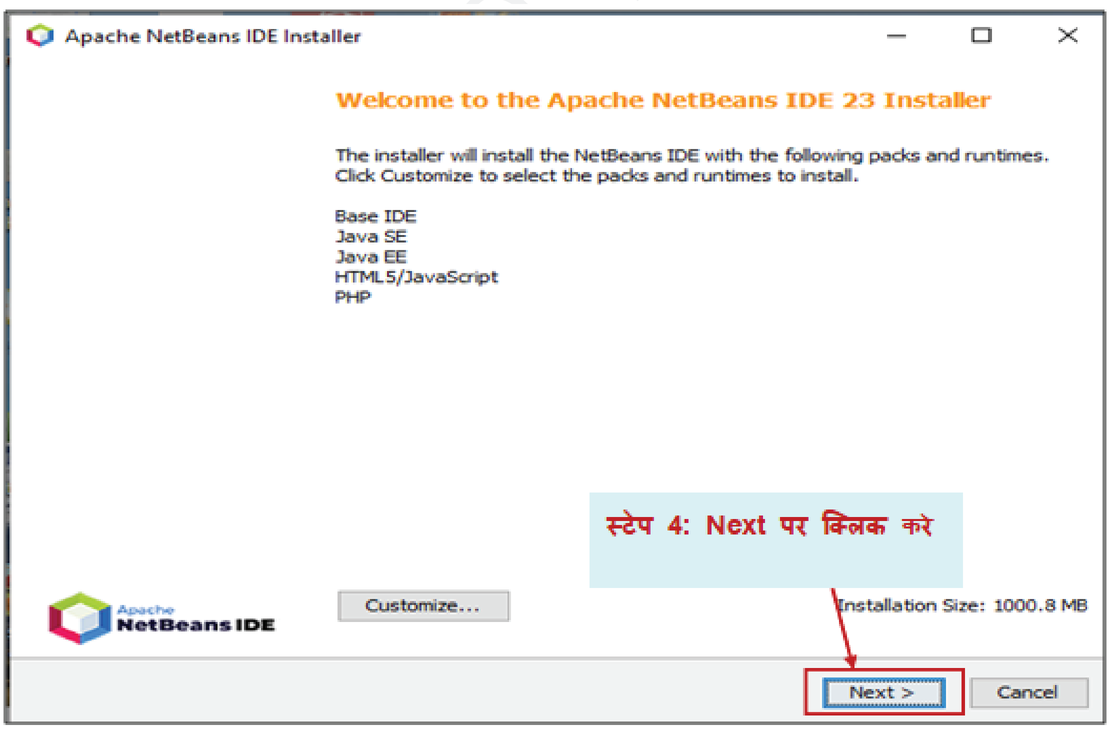


चित्र 4. 34

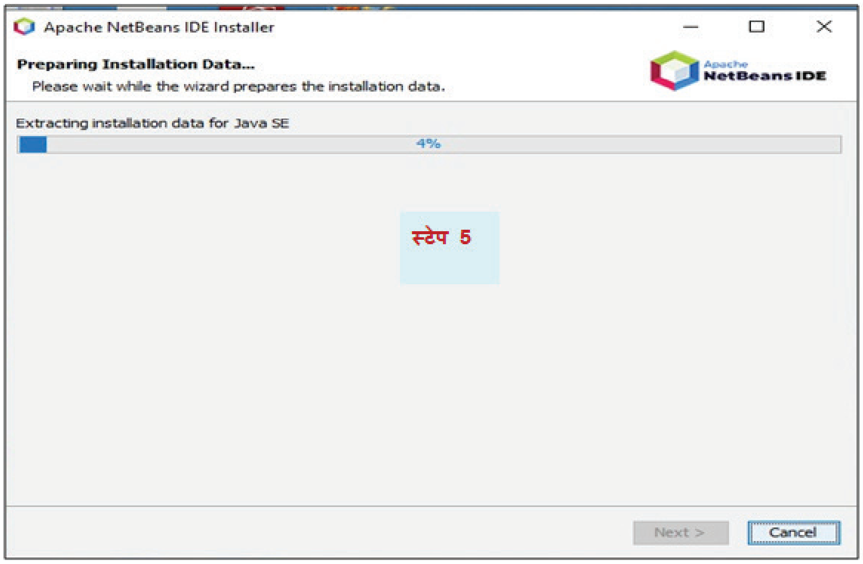


चित्र 4. 35

* **समापनः** इंस्टॉलेशन समाप्त होने के बाद "Finish" पर क्लिक करें।



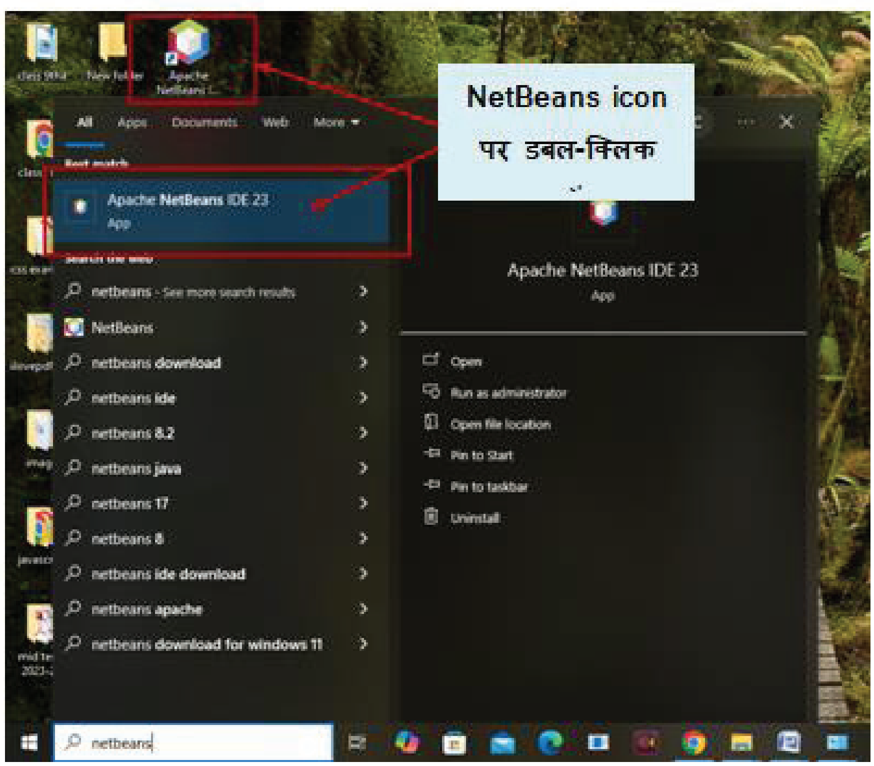
चित्र 4. 36



चित्र 4. 37

स्टेप 3: NetBeans IDE शुरू करें

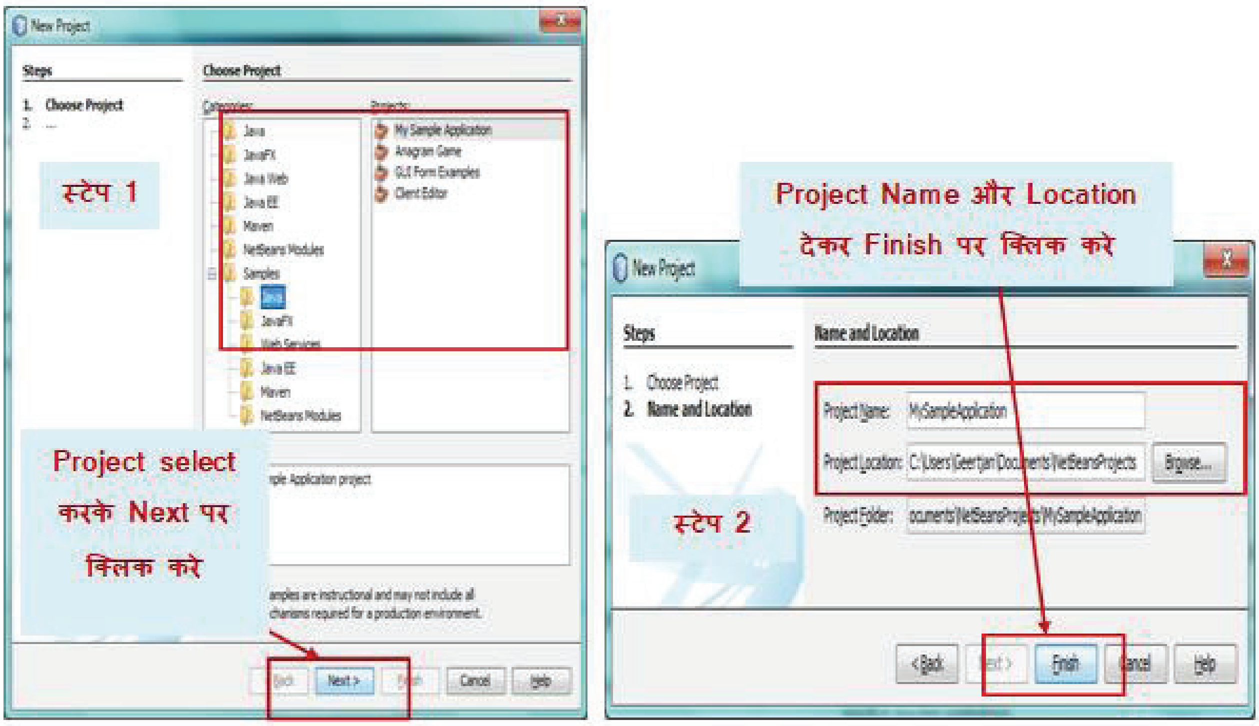
* **NetBeans खोलें:** इंस्टॉलेशन के बाद, आप अपने डेस्कटॉप पर या स्टार्ट मेन्यू में NetBeans आइकन देखेंगे। उस पर डबल-क्लिक करें।



चित्र 4. 38

* **प्रारंभिक (Initial) सेटअप:** पहली बार खोलने पर, आपको कुछ प्रारंभिक सेटिंग्स करने के लिए कहा जा सकता है। इन सेटिंग्स को पूरा करें।

* **नई परियोजना (Project) बनाएँ:** "File" मेन्यू पर क्लिक करें, फिर "New Project" चुनें। अपनी परियोजना के प्रकार का चयन करें और "Next" पर क्लिक करें।



चित्र 4. 39

स्टेप 4: परियोजना का विकास करें(Develop Your Project)

अब आप NetBeans IDE में अपने प्रोजेक्ट पर काम करना शुरू कर सकते हैं। कोड लिखें, रन करें और अपनी परियोजना(project) का परीक्षण(test) करें।

सुझाव (Tips):

सुनिश्चित करें कि आपके पास JDK (Java Development Kit) स्थापित है, क्योंकि NetBeans Java विकास के लिए सबसे अधिक उपयोग किया जाने वाला IDE है।

अभ्यास प्रश्न 4.2

प्रश्न 1. किसी प्रोजेक्ट के लिए NetBeans IDE में एक नई फाइल कैसे बनाई जाती है?

प्रश्न 2. NetBeans IDE में एक प्रोजेक्ट को टेस्ट और रन करने के लिए क्या प्रक्रिया है?

महत्वपूर्ण प्रश्न

A. बहुविकल्पीय प्रश्न (Multiple Choice Questions)

1. निम्नलिखित में से कौन सा डेटाबेस प्रबंधन प्रणाली (DBMS) का उदाहरण नहीं है?

a) MySQL b) Oracle

c) Excel d) PostgreSQL

2. ई-कॉमर्स वेबसाइट में किस टेबल का उपयोग ग्राहक की जानकारी रखने के लिए किया जाता है?

a) Order Table b) Product Table

c) Customer Table d) Payment Table

3. निम्नलिखित में से कौन सा डेटा को जोड़ने के लिए उपयोग किया जाता है?

a) Insert b) Select

c) Delete d) Update

4. निम्नलिखित में से कौन सा क्षेत्र डेटाबेस का उपयोग करता है?

a) स्वास्थ्य देखभाल b) शिक्षा

c) सरकार d) उपरोक्त सभी

5. निम्नलिखित में से कौन सा डेटाबेस का कार्य नहीं है?

a) डेटा को संग्रहीत करना b) डेटा को विश्लेषण करना

c) डेटा को छिपाना d) डेटा को पुनर्प्राप्त करना

उत्तर

1. c) Excel 2. c) Customer Table

3. a) Insert 4. d) उपरोक्त सभी

5. c) डेटा को छिपाना

SCERT DELHI

B. रिक्त स्थान भरें (Fill in the Blanks)

1. SQL का पूरा नाम __________ है।
2. __________ कमांड का उपयोग डेटा को हटाने के लिए किया जाता है।
3. ग्राहक की खरीदारी की सूची को __________ में रखा जाता है।
4. __________ का उपयोग डेटाबेस की संरचना को डिजाइन करने के लिए किया जाता है।
5. __________ में डेटा को टेबलों में संग्रहीत किया जाता है।

उत्तर

1. Structured Query Language
2. Delete
3. Cart
4. Data Modeling
5. Relational Database

C. सही या गलत(True or False)

1. डेटाबेस का मुख्य कार्य डेटा को संग्रहित(stored) करना है।
2. SQL केवल डेटा को देखने के लिए उपयोग किया जाता है।
3. डेटाबेस में डेटा को संरक्षित करने के लिए बैकअप लिया जाना चाहिए।
4. डेटा को संशोधित करने के लिए MODIFY कमांड का उपयोग किया जाता है।
5. एक ई-कॉमर्स वेबसाइट पर ग्राहक को लॉगिन करने की आवश्यकता नहीं होती।

उत्तर

1. सही   2. गलत   3. सही   4. गलत   5. गलत

D. संक्षिप्त उत्तर प्रश्न(Short Answer Questions)

1. ई-कॉमर्स में डेटाबेस का क्या उपयोग है?
2. सरकार किस प्रकार के रिकॉर्ड्स के लिए डेटाबेस का उपयोग करती है?
3. प्रोडक्ट टेबल में कौन-कौन सी जानकारी संग्रहीत होती है?

SCERT DELHI

4. एक शॉपिंग वेबसाइट में कस्टमर टेबल की भूमिका क्या है?

5. "Shipment" टेबल का उद्देश्य क्या है?

E. दीर्घ उत्तर प्रश्न(Long Answer Questions)

1. डेटाबेस मैनेजमेंट सिस्टम (DBMS) का क्या लाभ हैं?

2. डेटाबेस की प्राथमिक कुंजी (Primary Key) और विदेशी कुंजी (Foreign Key) क्या होती हैं?

3. "Data Redundancy" से क्या तात्पर्य है और इसे कैसे कम किया जा सकता है?

4. NetBeans IDE में एक प्रोजेक्ट को टेस्ट और रन करने के लिए क्या प्रक्रिया है?

5. ई-कॉमर्स में विभिन्न भुगतान विधियों के प्रबंधन के लिए डेटाबेस का उपयोग कैसे किया जाता है?

SCERT DELHI

नोट : इस पुस्तक में प्रयुक्त सामग्री एवं चित्र पूर्णतः शैक्षणिक उद्देश्य के लिए है, किसी व्यावसायिक उपयोग के लिए नहीं।

SCERT
स्वाध्यायान्मा प्रमदः
State Council of Educational Research & Training, Delhi
Varun Marg, Defence Colony, New Delhi-110024