(1) Write a simple "Hello World" program in two different programming languages ofyour choice. Compare the structure and syntax.

Ans:

- **Php**

```
<?php
    echo "Hello, World!";
?>
```

- **java**

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Comparison: PHP vs Java

Feature	PHP	Java
Syntax Simplicity	Simple and minimal	Verbose and more structured
Execution Type	Interpreted by a PHP engine (e.g., server-side)	Compiled into bytecode, runs on JVM
File Structure	No class or method needed	Must have a class and a `main()` method
Printing Output	`echo "Hello, World!";`	`System.out.println("Hello, World!");`
Semicolons	Required at the end of statements	Required at the end of statements
Language Type	Loosely typed	Strongly typed
Usage Environment	Web-based, mostly server-side scripting	General-purpose, often for desktop/mobile

(2) Research and create a diagram of how data is transmitted from a client to a serverover the internet.

Ans:

Data Transmission: Client to Server Over the Internet

1. Application Layer (Client)

- The user initiates a request using an application (e.g., browser).
- The browser generates an HTTP/HTTPS request (e.g., for a web page).

2. Transport Layer (TCP/UDP)

- The request is broken into **segments**.
- TCP adds headers (for sequencing, acknowledgment, error checking).

3. Internet Layer (IP)

- Segments are encapsulated into **IP packets**.
- Source and destination IP addresses are added.

4. Network Access Layer

- IP packets are converted into **frames** (Ethernet or Wi-Fi).
- MAC addresses are included for local delivery.

5. Physical Layer

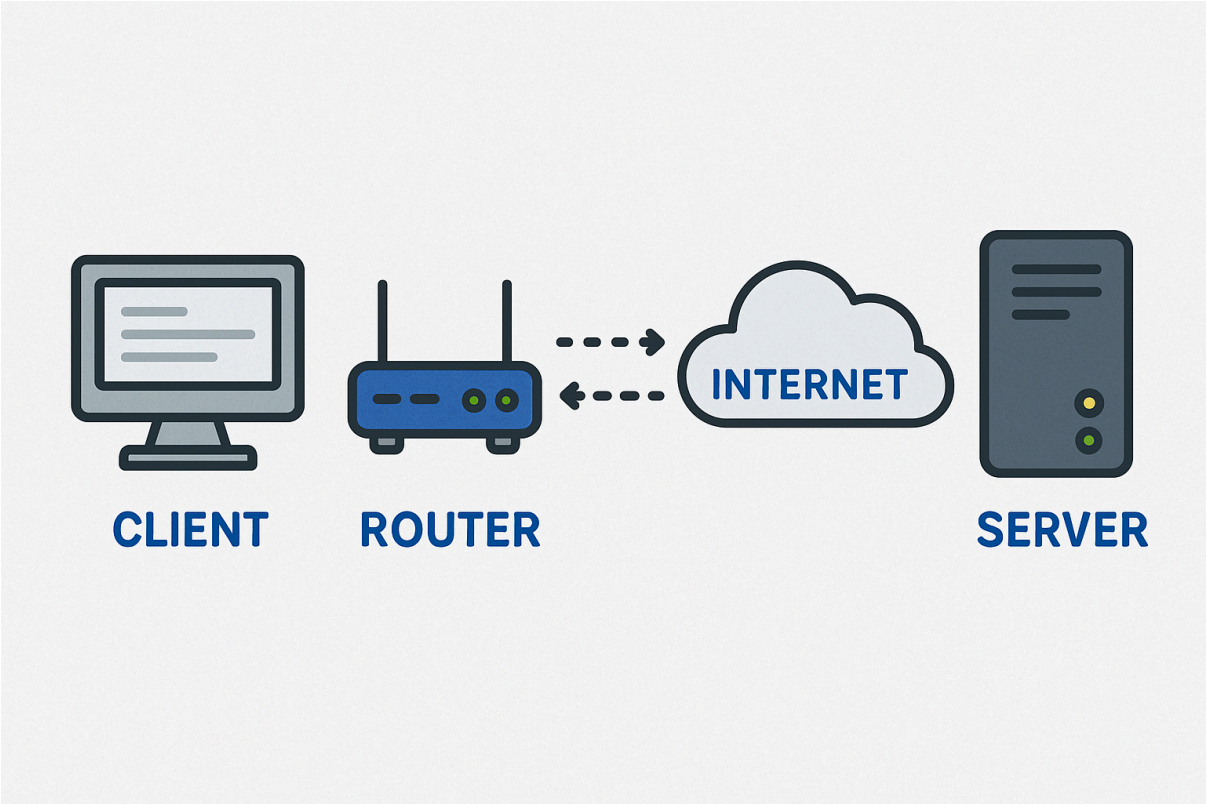
- Frames are converted into **electrical/light/radio signals**.
- Data is transmitted over physical media (cables, wireless).

6. Intermediate Devices

- **Routers** forward packets across networks using IP addresses.
- **Switches** operate at the data link layer to route frames locally.

7. Destination Server

- Server's NIC receives the data.
- Data is reconstructed through the protocol layers back to the application.



(3) Design a simple HTTP client-server communication in any language.

Ans:-

```
<?php
// server.php

if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    header("Content-Type: text/plain");
    echo "Hello from the PHP server!";
} else {
    http_response_code(405); // Method Not Allowed
    echo "Only GET is supported.";
}
?>
```

```
<?php
// client.php

$url = "http://localhost:8080";

$response = file_get_contents($url);

if ($response !== false) {
    echo "Server says: " . $response . PHP_EOL;
} else {
    echo "Failed to connect to server." . PHP_EOL;
}
?>
```

(4) Research different types of internet connections (e.g., broadband, fiber, satellite)and list their pros and cons.

Ans:-

1. DSL (Digital Subscriber Line)

Pros:

- Uses existing telephone lines — widely available.
- Affordable for most users.

- Reliable and stable for basic usage (browsing, email).

Cons:

- Slower speeds compared to cable and fiber.
- Performance drops with distance from the provider's station.
- Not ideal for streaming or gaming.

2. Cable Internet

Pros:

- Faster than DSL.
- Suitable for streaming, online gaming, and video conferencing.
- Available in most urban and suburban areas.

Cons:

- Bandwidth is shared with neighbors — can slow down during peak hours.
- More expensive than DSL.
- Not always available in rural areas.

3. Fiber-Optic Internet

Pros:

- Extremely fast speeds (up to 1 Gbps or more).
- Low latency — ideal for gaming and video conferencing.
- Very reliable with consistent performance.

Cons:

- Limited availability (mainly in cities and suburbs).
- Expensive infrastructure and installation costs.

- Not yet widespread in rural areas.

4. Satellite Internet

Pros:

- Available in remote and rural areas where other options aren't.
- Doesn’t require wired infrastructure.
- Easy to install and access.

Cons:

- High latency — poor for gaming and video calls.
- Affected by weather conditions (rain, snow).
- Expensive and often comes with data limits.

5. Mobile Broadband (4G/5G)

Pros:

- Wireless and portable — can be used anywhere with signal.
- 5G offers competitive speeds for urban users.
- Easy setup via mobile hotspot or USB dongle.

Cons:

- Data limits or throttling common.
- Signal strength and speed vary by location.
- Can be costly for heavy internet users.

(5): Simulate HTTP and FTP requests using command line tools (e.g., curl).

Ans:-

1. Simulate HTTP Requests Using `curl`

a) GET Request

curl http://example.com

b) POST Request

curl -X POST -d "username=admin&password=1234" http://example.com/login

c) Custom Headers

curl -H "Authorization: Bearer YOUR_TOKEN" http://example.com/api/data

d) Save Output to File

curl http://example.com -o homepage.html

2. Simulate FTP Requests Using `curl`

a) Download a File

curl -u anonymous: ftp://ftp.example.com/file.txt -O

b) Upload a File

curl -T upload.txt -u username:password ftp://ftp.example.com/uploads/

c) List Directory Contents

curl -u username:password ftp://ftp.example.com/

(5) Identify and explain three common application security vulnerabilities. Suggest Possible solutions.

Ans:-

1. SQL Injection

Explanation:

SQL Injection occurs when an application allows users to enter input directly into a SQL query without proper validation or escaping. This can allow attackers to execute arbitrary SQL commands, potentially accessing or destroying database data.

Example:

SELECT * FROM users WHERE username = '$username' AND password = '$password';

Solution:

- Use Prepared Statements (Parameterized Queries) instead of string concatenation.

Example in PHP (PDO):

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE username = ? AND password = ?");

$stmt->execute([$username, $password]);
```

- Validate and sanitize all user inputs.
- Use ORM tools (e.g., Eloquent in Laravel) that abstract SQL queries securely.

2. Cross-Site Scripting (XSS)

Explanation:

XSS occurs when attackers inject malicious scripts into web pages that are viewed by other users. These scripts can steal cookies, session tokens, or manipulate DOM.

Types:

- Stored XSS: Malicious script stored on the server.
- Reflected XSS: Script comes from the current request (e.g., URL).
- DOM-based XSS: Manipulation of the DOM environment in the browser.

Solution:

- Escape output before rendering data into HTML using frameworks (e.g., {{ data }} in Blade or JSX).
- Use Content Security Policy (CSP) headers to restrict script execution.
- Validate and sanitize input using libraries (e.g., DOMPurify for HTML).
- Avoid innerHTML and use textContent or safer DOM methods.

3. Cross-Site Request Forgery (CSRF)

Explanation:

A user is logged into a banking site. An attacker sends a malicious link:

Solution:

- Use CSRF Tokens:
    - Include a unique token in every state-changing request and verify it on the server.
    - Example in Laravel: @csrf directive in forms.
- Use SameSite cookies to restrict when cookies are sent.
- Require re-authentication for sensitive actions.

(6)Identify and classify 5 applications you use daily as either system software orapplication software.

Ans:-

- **Google Chrome** – *Application Software*
  (Used for browsing the internet)
- **Microsoft Word** – *Application Software*
  (Used for creating and editing documents)
- **Windows 10** – *System Software*
  (Operating system that manages hardware and software resources)
- **File Explorer** – *System Software*
  (Utility program for managing files and folders)

- **Spotify** – *Application Software*
  (Used for streaming music and podcasts)

Explanation:

- **System Software**: Enables the computer hardware to function and provides a platform for application software. Examples include operating systems, device drivers, and utilities.

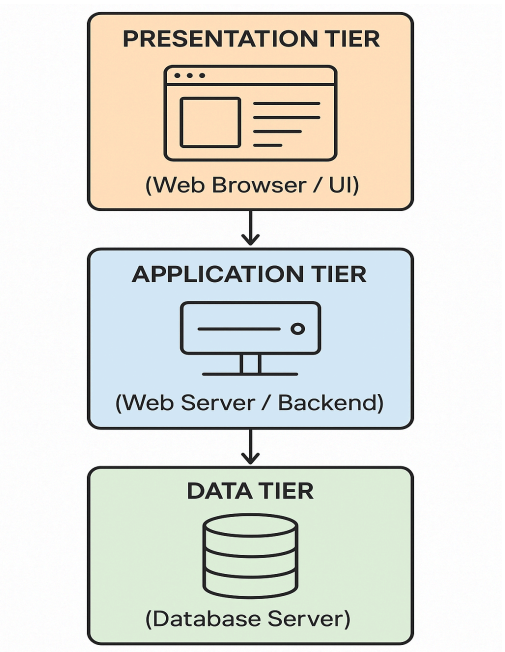
- **Application Software**: Designed for end-users to perform specific tasks like writing documents, browsing the internet, or listening to music.

(7)Design a basic three-tier software architecture diagram for a web application

Ans:-

- **Here's a simple three-tier software architecture diagram for a web application, represented in text format. The three tiers are**:

- Presentation Tier (Client)
- Application Tier (Server)
- Data Tier (Database)



Description:

- Presentation Tier: The front-end interface users interact with (browser or app UI).
- Application Tier: The middle layer that processes requests, contains the business logic, and interacts with the database.
- Data Tier: Where data is stored, queried, and managed.

(8)Create a case study on the functionality of the presentation, business logic, and dataaccess layers of a given software system.

Ans:-

Overview:

This case study explores the three-tier architecture of an Online Banking System, focusing on the roles of the Presentation Layer, Business Logic Layer, and Data Access Layer.

1. Presentation Layer (UI Layer):

Functionality:

- This is the user-facing interface where customers interact with the application via a web browser or mobile app.
- Technologies: HTML, CSS, JavaScript, React Native, or Flutter.
- Key Features:
    - Login and authentication form.
    - Dashboard displaying account balance and recent transactions.
    - Forms for fund transfer, bill payments, and account settings.
- Example Action:
    - A user fills out the "Transfer Funds" form and clicks the "Submit" button.

2. Business Logic Layer (Application Layer):

Functionality:

- Acts as the core of the system where all business rules and operations are implemented.
- Technologies: Java, .NET, Python (Django/Flask), Node.js.
- Key Features:
    - Validates user input (e.g., check if the amount is valid and within the account balance).
    - Executes business rules (e.g., applying daily transfer limits).
    - Manages sessions, authorization, and workflow logic.
- Example Action:
    - After the "Transfer Funds" form is submitted, this layer verifies the user's identity, checks fund availability, and processes the transaction.

3. Data Access Layer (Database Layer):

Functionality:

- Manages communication between the application layer and the database.
- Technologies: SQL, ORM (Object Relational Mapping) like Hibernate or Entity Framework.
- Key Features:
    - Handles CRUD operations: Create, Read, Update, Delete.
    - Ensures data integrity and security.
    - Connects to the relational (MySQL, PostgreSQL) or NoSQL (MongoDB) databases.

- Example Action:
    - Executes a SQL query to deduct the transferred amount from the sender's account and credit the receiver's account.
    - Logs the transaction in the audit trail.

(9)Explore different types of software environments (development, testing, production).Set up a basic environment in a virtual machine.

Ans:-

Exploring Different Types of Software Environments

In the **Software Development Lifecycle (SDLC)**, different environments are used to ensure that software is developed, tested, and deployed efficiently and safely. Here are the three main types:

1. Development Environment

- **Purpose:** Where developers write and initially test code.
- **Tools/Software:** Code editors (VS Code, IntelliJ), compilers, debuggers, local servers.
- **Example:** Running a Node.js app on localhost (`http://localhost:3000`) with a local database.

2. Testing Environment

- **Purpose:** Used by QA teams or automated systems to test the application for bugs, performance, and security.
- **Tools:** Testing frameworks (JUnit, Selenium, Jest), CI/CD tools (Jenkins, GitHub Actions).

- **Example:** A staging server that mirrors production to simulate real-world usage.

3. Production Environment

- **Purpose:** The live environment where end-users access the software.
- **Features:** High security, full performance, uptime monitoring, and backups.
- **Example:** The deployed application at `https://yourapp.com`.

Set Up a Basic Environment in a Virtual Machine (VM)

Step-by-Step Setup (Ubuntu in VirtualBox)

Requirements:

- Oracle VirtualBox (or VMware)
- Ubuntu ISO image (download from ubuntu.com)

Step 1: Install VirtualBox

- Download and install from: https://www.virtualbox.org/

Step 2: Create a New Virtual Machine

- Open VirtualBox → Click "New"
- Name: `DevVM`
- Type: `Linux`

- Version: `Ubuntu (64-bit)`
- Memory: Allocate at least 2048 MB
- Create a Virtual Hard Disk (~20 GB)

Step 3: Install Ubuntu

- Start VM → Choose Ubuntu ISO as startup disk
- Follow Ubuntu setup steps (language, region, user name, etc.)

Step 4: Set Up the Development Environment

- Update packages:

  sudo apt update && sudo apt upgrade

- Install essential tools:

  sudo apt install git curl build-essential

- Install Node.js:

  curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -

  sudo apt install -y nodejs

- Clone a sample project:

  git clone https://github.com/sample/project.git

  cd project

  npm install

  npm start

(10)Create a list of software you use regularly and classify them into the following categories: system, application, and utility software.

Ans:-

1. System Software

These manage the hardware and basic system operations.

- **Windows 10 / 11** – Operating System
- **macOS / Linux (Ubuntu, Fedora)** – Operating System
- **Device Drivers** – Enable hardware like printers, network cards, etc.
- **BIOS / UEFI** – Low-level system software for booting the computer

2. Application Software

These help users perform specific tasks.

- **Microsoft Word / Excel / PowerPoint** – Office productivity
- **Google Chrome / Mozilla Firefox** – Web browsing
- **Spotify / VLC Media Player** – Media streaming and playback
- **Zoom / Microsoft Teams** – Video conferencing
- **Photoshop / Canva** – Graphic design

3. Utility Software

These support system performance and maintenance.

- **WinRAR / 7-Zip** – File compression
- **Antivirus Software (e.g., Norton, Windows Defender)** – Security

- **CCleaner** – Disk cleanup and optimization
- **Backup Software (e.g., Acronis, Windows Backup)** – Data backup
- **Disk Management Tools** – Partitioning and formatting drives

(11) Write a report on the various types of application software and how they improve productivity

Ans:-

Introduction

Application software refers to programs designed to perform specific tasks for users. These applications help individuals and organizations to complete work efficiently, communicate effectively, manage data, and automate tasks. This report explores various types of application software and highlights their role in improving productivity.

1. Word Processing Software

- **Examples:** Microsoft Word, Google Docs, LibreOffice Writer
- **Purpose:** Used to create, edit, and format textual documents.
- **Productivity Benefit:** Speeds up document creation, offers spell check, templates, and collaborative editing tools.

2. Spreadsheet Software

- **Examples:** Microsoft Excel, Google Sheets
- **Purpose:** Allows users to perform calculations, data analysis, and chart generation.
- **Productivity Benefit:** Automates complex calculations, helps in financial modeling, and supports data-driven decision-making.

3. Presentation Software

- **Examples:** Microsoft PowerPoint, Google Slides, Prezi
- **Purpose:** Used to create slideshows and visual presentations.

- **Productivity Benefit:** Enhances communication of ideas and proposals through visual storytelling and slide templates.

4. Database Management Software

- **Examples:** Microsoft Access, MySQL, Oracle Database
- **Purpose:** Helps store, retrieve, and manage large sets of structured data.
- **Productivity Benefit:** Improves data organization, reduces redundancy, and ensures efficient data access and reporting.

5. Communication Software

- **Examples:** Microsoft Teams, Zoom, Slack, WhatsApp
- **Purpose:** Facilitates real-time communication and collaboration.
- **Productivity Benefit:** Enables remote work, instant messaging, video conferencing, and project coordination.

6. Multimedia Software

- **Examples:** Adobe Photoshop, VLC Media Player, Canva
- **Purpose:** Used for creating and editing images, videos, and audio.
- **Productivity Benefit:** Supports creative tasks such as marketing design, content creation, and multimedia presentations.

7. Project Management Software

- **Examples:** Trello, Asana, Microsoft Project
- **Purpose:** Helps plan, track, and manage projects and tasks.
- **Productivity Benefit:** Improves team collaboration, deadline tracking, and resource allocation.

8. Web Browsers

- **Examples:** Google Chrome, Mozilla Firefox, Microsoft Edge
- **Purpose:** Provides access to the internet and web applications.
- **Productivity Benefit:** Supports research, cloud applications, and online collaboration tools.

(12)Create a flowchart representing the Software Development Life Cycle (SDLC).

Ans:-

(13)Write a requirement specification for a simple library management system

Ans:-

1. Introduction

1.1 Purpose

The purpose of this document is to outline the functional and non-functional requirements for a simple Library Management System. The system will help library staff manage books, members, and lending operations efficiently.

1.2 Scope

The system will allow:

- Librarians to add, update, delete, and search for books.
- Librarians to register and manage library members.
- Issuing and returning of books with due date tracking.
- Fine calculation for overdue books.

The system will be used by library staff and administrators.

2. Overall Description

2.1 Product Perspective

The LMS will be a standalone desktop or web application. It will store information in a local or central database and feature a graphical user interface for ease of use.

2.2 User Classes and Characteristics

- **Administrator**: Has full access to manage users, books, and settings.
- **Librarian**: Manages books and member records, handles lending operations.
- **Member (optional)**: Can search book availability and view personal lending history.

3. Functional Requirements

3.1 Book Management

- FR1: Add new books with title, author, ISBN, publisher, and quantity.
- FR2: Edit or delete existing book records.
- FR3: Search for books by title, author, or ISBN.
- FR4: Check book availability.

3.2 Member Management

- FR5: Register new members with name, contact, and ID.
- FR6: Edit or remove member records.
- FR7: View member borrowing history.

3.3 Book Lending

- FR8: Issue books to members.
- FR9: Return books and update the inventory.
- FR10: Track due dates and generate overdue notifications.
- FR11: Calculate fines for late returns.

3.4 Authentication and Authorization

- FR12: Login/logout functionality for staff.
- FR13: Role-based access (admin vs. librarian).

4. Non-Functional Requirements

4.1 Usability

- The system should have an intuitive and user-friendly interface.

4.2 Performance

- System should respond to user actions within 2 seconds.

4.3 Security

- User credentials must be encrypted.
- Access control must prevent unauthorized access to admin functions.

4.4 Reliability

- The system should maintain data integrity in case of failures.

4.5 Portability

- Should run on Windows and Linux systems or be browser-accessible.

5. Assumptions and Constraints

- Internet connectivity is required only for web-based deployments.
- The maximum number of simultaneous users will not exceed 10.
- System backups will be performed manually by administrators.

6. Future Enhancements (Optional)

- Online member self-service portal.
- Barcode scanning for faster operations.
- Email/SMS notifications for due dates.

(14)Perform a functional analysis for an online shopping system.

Ans:-

1. Introduction

An online shopping system allows users to browse products, place orders, and manage their accounts. This analysis identifies and describes the key functions the system must perform to meet user and business needs.

2. Actors Involved

1. **Customer/User**

2. **Admin**
3. **Payment Gateway**
4. **Delivery Service (optional third-party integration)**

3. Core Functionalities

3.1 User Management

- **Register**: New users can create accounts by providing personal information.
- **Login/Logout**: Users can securely log in and out of the system.
- **Profile Management**: Users can view and update their profile details (name, address, contact).
- **Password Recovery**: Forgotten password functionality via email or SMS.

3.2 Product Catalog Management

- **Product Listing**: View all available products with filters by category, price, rating, etc.
- **Product Details**: View detailed information for each product (name, image, description, price, stock).
- **Search**: Search for products using keywords or filters.

3.3 Shopping Cart

- **Add to Cart**: Users can add items to their cart.
- **View Cart**: Display all selected items with quantity and price.
- **Update Cart**: Modify quantities or remove items.
- **Cart Total**: Display subtotal, taxes, shipping, and total amount.

3.4 Checkout and Payment

- **Billing and Shipping Information**: Enter or select saved address.

- **Select Payment Method**: Choose from credit/debit card, net banking, digital wallets, etc.
- **Process Payment**: Integrate with a payment gateway to complete transactions.
- **Order Confirmation**: Generate confirmation page and send email receipt.

3.5 Order Management

- **Order Tracking**: Track current status (Pending, Shipped, Delivered, Cancelled).
- **Order History**: View all past orders with details.
- **Cancel/Return Order**: Initiate cancellation or return process (if allowed by policy).

3.6 Product Reviews and Ratings

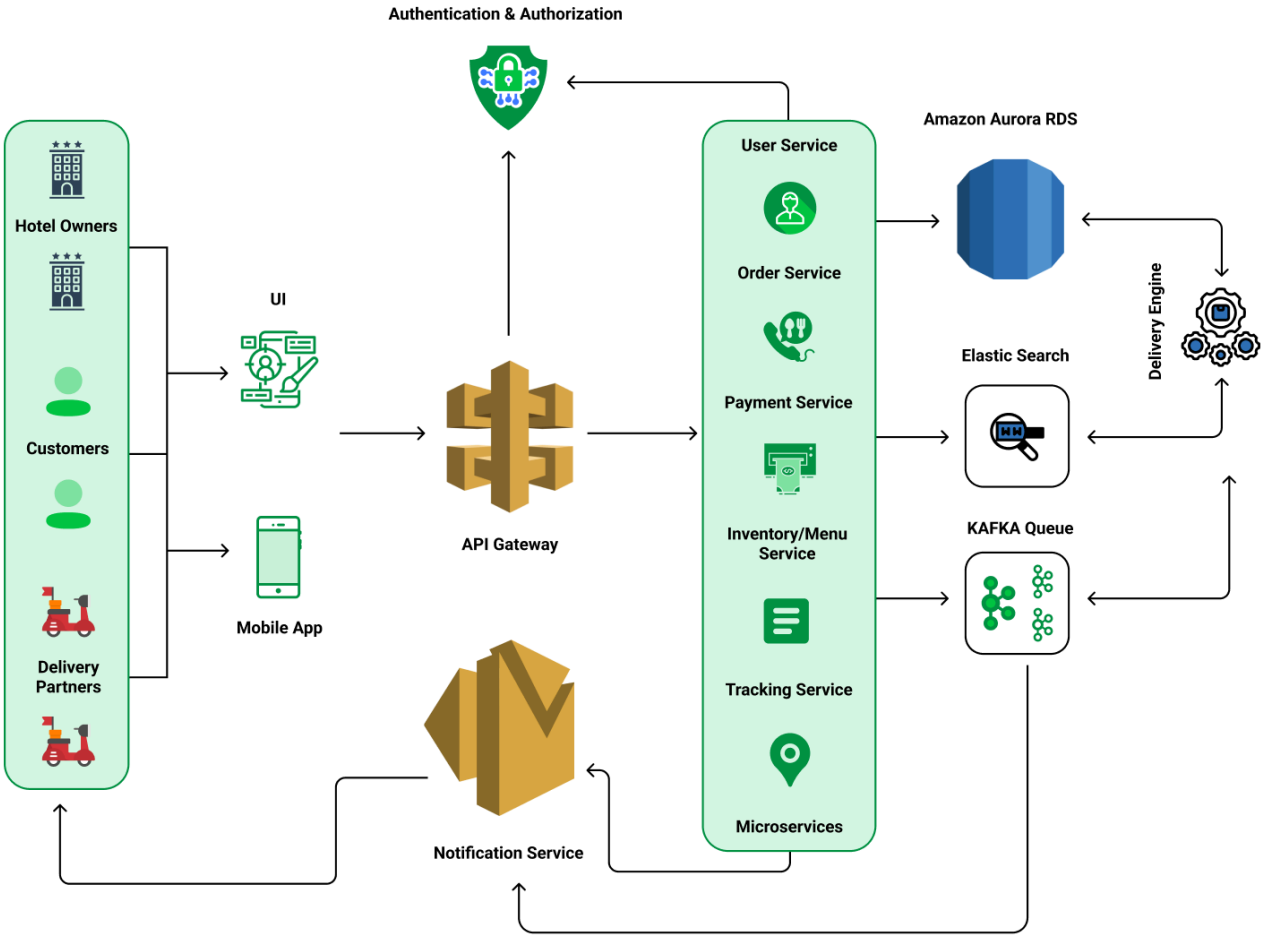
- **Submit Review**: Users can rate and review purchased products.
- **View Reviews**: See other customer reviews on product pages.

3.7 Admin Panel

- **User Management**: Add/remove users, assign roles.
- **Product Management**: Add/edit/delete products and categories.
- **Inventory Management**: Update stock quantities and monitor low-stock alerts.
- **Order Management**: View and update order statuses.
- **Analytics and Reports**: Generate reports on sales, user activity, and inventory.

(15)Design a basic system architecture for a food delivery app

Ans:-



(16)Develop test cases for a simple calculator program

Ans:-

Test Case ID	Description	Input	Expected Output	Actual Output	Result
TC001	Add two positive numbers	5 + 3	8	8	Pass
TC002	Subtract smaller from larger	10 - 4	6	6	Pass
TC003	Multiply two positive numbers	6 * 7	42	42	Pass
TC004	Divide two positive numbers	20 / 5	4	4	Pass
TC005	Add positive and negative number	-5 + 10	5	5	Pass
TC006	Subtract larger from smaller	3 - 8	-5	-5	Pass
TC007	Multiply with zero	9 * 0	0	0	Pass
TC008	Division by zero	7 / 0	Error/Exception	Error	Pass

TC009	Add large numbers	999999 + 1	1000000	1000000	Pass
TC010	Subtract decimal numbers	5.5 - 2.2	3.3	3.3	Pass
TC011	Multiply decimals	2.5 * 4	10.0	10.0	Pass
TC012	Divide resulting in decimal	7 / 2	3.5	3.5	Pass
TC013	Non-numeric input	"a" + 2	Error/Invalid Input	Error	Pass
TC014	Empty input	[empty]	Error/Prompt	Error	Pass
TC015	Invalid operator	3 $ 4	Error/Invalid Op	Error	Pass
TC016	Multiple operations (order of ops)	2 + 3 * 4	14	14	Pass
TC017	Parentheses used in expression	(2 + 3) * 4	20	20	Pass

(17) Document a real-world case where a software application required criticalmaintenance.

Ans:-

Case Study: Critical Maintenance of WhatsApp Outage – October 2022

Background:

- On **October 25, 2022**, **WhatsApp**, the widely used messaging application owned by **Meta (formerly Facebook)**, experienced a **global outage**. Users worldwide were unable to send or receive messages for over 2 hours.

Problem Overview:

- **Duration:** ~2 hours
- **Impact:** Over **2 billion users** globally
- **Symptoms:** Messages not sending or receiving, inability to update status or group chats
- **Platforms Affected:** Android, iOS, Web, and Desktop apps

Cause of the Issue:

Meta officially reported that the outage was due to **a technical configuration change** in their **server infrastructure**.

- Likely involved **backend server misconfiguration** or **deployment failure**
- Affected **message delivery services**, which are critical in real-time messaging apps

Critical Maintenance Actions Taken:

1. **Immediate rollback** of the faulty configuration.
2. **Isolated affected services** to prevent further cascading failures.
3. **Monitored system health** using internal dashboards and diagnostic tools.
4. **Gradual restoration** of services region by region.
5. **Post-outage review and fix**: The engineering team implemented **fail-safes** to prevent similar issues in the future.

Impact of the Outage:

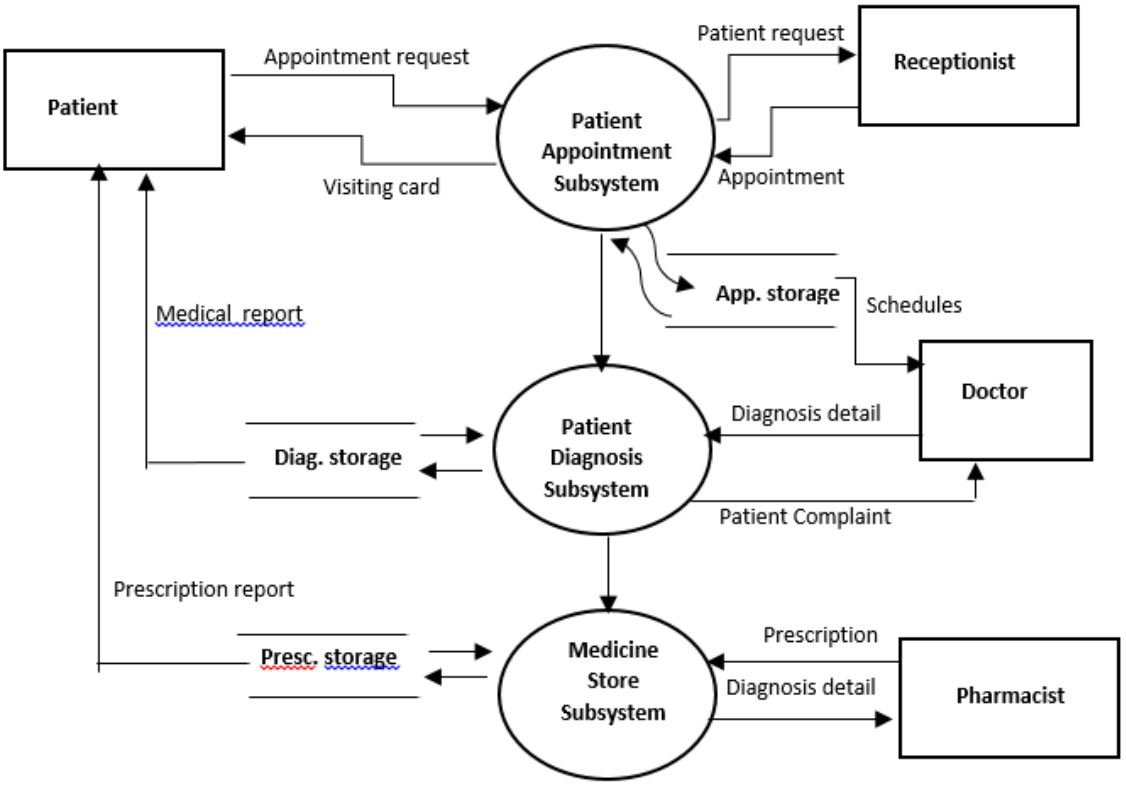
- Business and emergency communications were disrupted globally.
- Social media platforms were flooded with user complaints.
- Raised questions on **reliability and resilience** of globally-used communication apps.

Lessons Learned:

- **Rigorous testing** is needed before deploying changes to critical infrastructure.
- **Automated rollback systems** are crucial in large-scale applications.
- Importance of **real-time monitoring** and **incident response planning**.
- Highlighted the need for **redundancy and distributed failover systems**.

(18)Create a DFD for a hospital management system.

Ans:-



(19)Draw a flowchart representing the logic of a basic online registration system

Ans:-