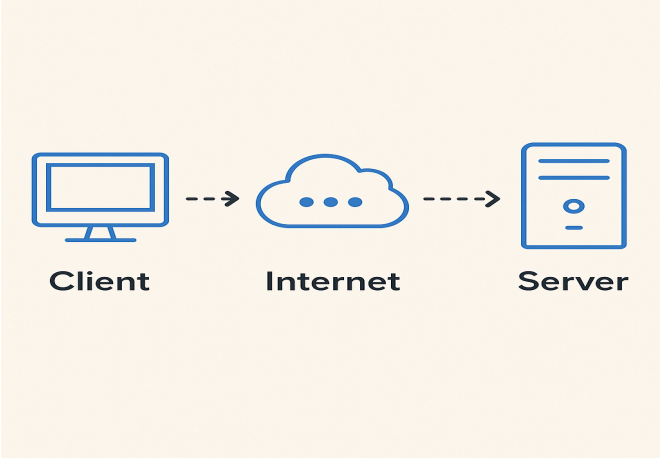
1 - Uploaded on git

2 -



3 - # Definition:

HTTP client-server communication is a method where a **client (usually a browser)** sends an HTTP request to a **server**, and the server processes that request and sends back an **HTTP response** (like a webpage or data).

1. Client:

- A client is a user-side application.
- It initiates a **request** using HTTP (HyperText Transfer Protocol).
- Example: Web browsers like Chrome, Firefox.

2. Server:

- A server is a system that waits for client requests.
- It **processes** the request and sends back a **response**.
- Example: Web servers like Apache, Nginx, or backend code written in Java, Python, etc.

Simple Flow of Communication:

1. **Client enters URL** → `www.example.com`
2. Browser sends **HTTP GET request** to server.
3. Server receives it and **sends back HTML** content.
4. Browser **displays the result**.

(4) 1. Broadband (DSL/Cable)

Definition: Uses telephone lines (DSL) or cable TV lines for internet.

(i) Pros:

- Widely available
- Affordable
- Always-on connection (no dial-up needed)

(ii) Cons:

- Slower than fiber
- Speed drops during peak hours
- Distance from provider affects speed (for DSL)

2. Fiber Optic

Definition: Uses light signals through fiber cables for high-speed internet.

(i) Pros:

- Extremely fast speeds (up to 1 Gbps+)
- Reliable and stable connection
- Not affected by distance or interference

(ii) Cons:

- Expensive to install
- Not available in rural/remote areas

3. Satellite Internet

Definition: Sends signals to and from satellites orbiting Earth.

(i) Pros:

- Works in remote and rural areas
- No need for cables or wires

(ii) Cons:

- High latency (slow response time)
- Weather can affect signal
- Limited data plans and expensive

4. Mobile Data (4G/5G)

Definition: Internet access through mobile networks via SIM cards or hotspots.

(i) Pros:

- Portable and wireless
- High speed (especially 5G)
- Easy setup (no wiring)

(ii) Cons:

- Signal depends on coverage area
- Data limits can be expensive
- Not ideal for heavy downloads/streaming

(5) - Definition of HTTP Request using curl:

HTTP request using curl is a way to communicate with a web server directly from the terminal. The curl tool sends an HTTP request (like GET or POST) to a web server, and the server responds with data (like HTML, JSON, etc.).

Definition of FTP Request using curl:

FTP request using curl allows you to connect to an FTP server using the terminal. You can upload, download, or list files from the FTP server using your login credentials with curl.

(6) - 1. SQL Injection

What is it?

When an attacker inserts malicious SQL code into an input field to access or manipulate the database.

Solution:

- Use **prepared statements** or **parameterized queries**
- Validate and sanitize all user inputs

2. Cross-Site Scripting (XSS)

What is it?

When attackers inject malicious JavaScript into a web page that runs on the user's browser.

Effect:

Can steal cookies, redirect users, or deface the site.

Solution:

- Sanitize and escape user input in HTML
- Use content security policy (CSP)
- Avoid inline JavaScript

3. Broken Authentication

What is it?

Weak login systems that allow attackers to hijack user sessions or guess passwords.

Effect:

Attackers gain unauthorized access to user accounts.

Solution:

- Use **strong password policies**
- Implement **multi-factor authentication (MFA)**
- Use secure session handling and tokens

(7) -

System Software – Definition:

System software is the software that controls and manages computer hardware and provides a platform for other software (like applications) to run.

Example:

- Operating System (Windows, Linux)
- Device Drivers
- Utilities
- BIOS

Application Software – Definition:

Application software is the software designed to help the user perform specific tasks such as browsing, writing, editing, or playing games.

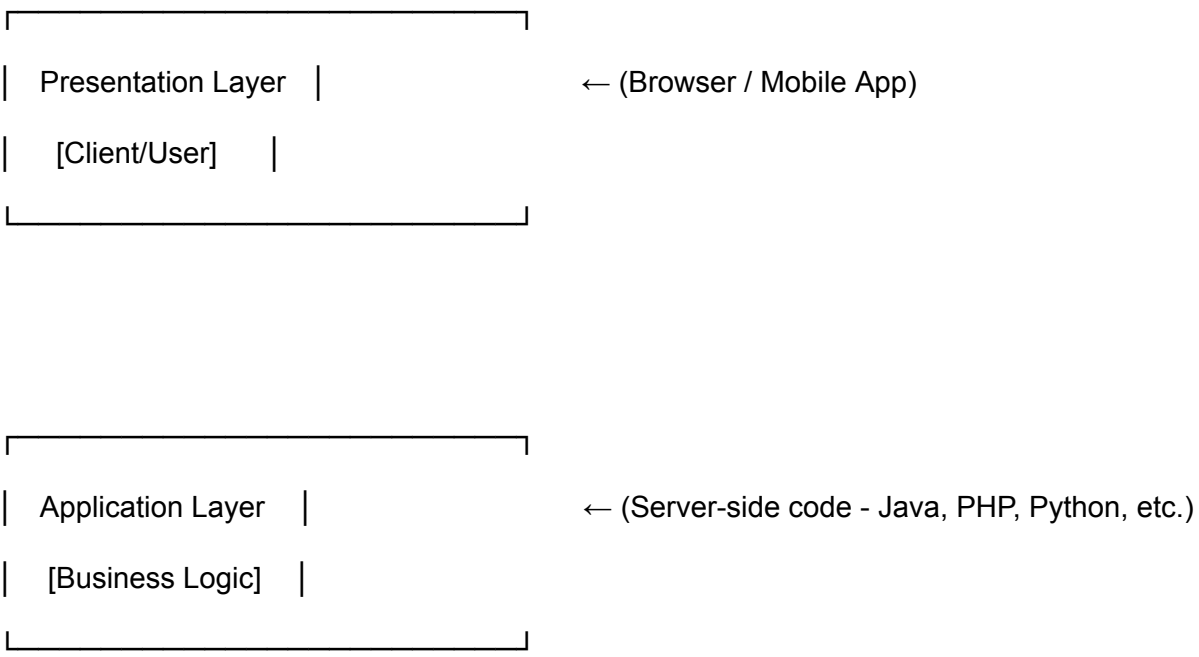
Example:

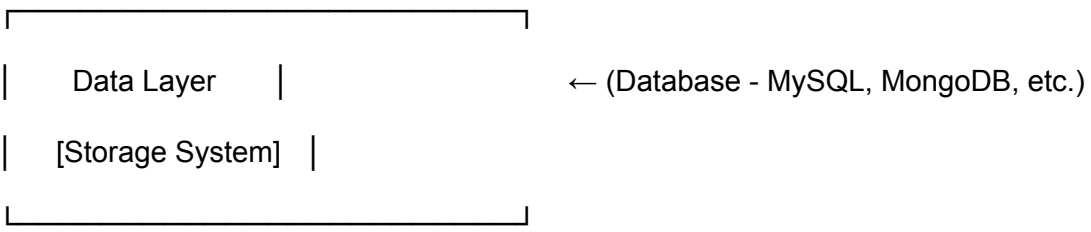
- Google Chrome
- MS Word
- VLC Media Player
- WhatsApp

(8) - Three-Tier Architecture – Definition:

Three-tier architecture separates an application into 3 layers:

1. **Presentation Layer (UI)**
2. **Application Layer (Logic)**
3. **Data Layer (Database)**





(9) Case Study: Online Food Ordering System (like Zomato/Swiggy)

1. Presentation Layer (UI Layer)

Role: Interacts directly with the user.

Example:-

- Login screen
- Menu display (pizzas, burgers, etc.)
- Add to cart / Checkout
- Order confirmation page

Technologies Used:
HTML, CSS, JavaScript, React

2. Business Logic Layer (Application Layer)

Role: Processes user inputs, applies business rules.

What it does:

- Validates login info
- Calculates total order cost
- Applies discount codes

Technologies Used:
Java, Python, Node.js, Spring Boot, Django

Logic Example:
If a user applies the promo code "FIRST50", this layer checks if it's valid and applies the discount.

3. Data Access Layer (Database Layer)

Role: Communicates with the database to **read/write** data.

Stores:

- User profiles
- Food items & prices
- Orders & payment history
- Restaurant info

Technologies Used:
MySQL, MongoDB, PostgreSQL

(10) - Software Environments

1. Development Environment

Where developers write and test code before sharing. To build, experiment, and debug software safely. Code editors, local databases, and debugging tools.

2. Testing Environment

An environment for validating software functionality and fixing bugs. To simulate real use and test software performance. Supports automated/manual testing; mirrors production setup.

3. Production Environment

The live environment where real users access the application. To run the final stable version for end users. Stable, secure, high-performance system with real data and traffic.

(11) - GitHub Task Summary

Task:
Uploaded the project code to GitHub successfully.

(12) - Git Commands to Push Code to GitHub

1. **Check for Changes**
   `git status` is used to see which files are modified, added, or deleted before committing changes.

2. **Stage the Changes**
   `git add` is used to mark which files you want to include in your next commit.

3. **Commit the Changes**
   `git commit -m "your message"` saves your staged changes to the local repository with a message.

4. **Push to Remote Repository**
   `git push origin branch-name` uploads the committed code to GitHub on the specified branch.

(13) - Done

(14) - Types of Software

System Software

System software controls hardware and provides a platform for running application software.
Examples:

- Windows OS
- macOS
- Linux
- Android OS
- BIOS

Application Software

Application software helps users perform specific tasks.
Examples:

- Google Chrome (Web Browsing)
- Microsoft Word (Document Editing)
- WhatsApp (Messaging)
- VLC Media Player (Media Playback)
- Zoom (Video Calling)

Utility Software

Utility software supports system performance, security, and maintenance.
Examples:

- Antivirus (e.g., Avast, Defender)
- WinRAR / 7-Zip (File Compression)
- CCleaner (Clean Junk Files)
- Disk Cleanup (Windows Tool)

(15) - Git Practical Activity Summary

1. **Cloning a Repository**
   `git clone` is used to copy a remote repository (like GitHub) to your local system for editing.

2. **Creating and Switching Branches**
   `git branch branch-name` creates a new branch.
   `git checkout branch-name` switches to the created branch.

3. **Making Changes and Committing**
   Files are edited, staged with `git add`, and saved in Git history using `git commit -m "message"`.

4. **Merging Branches**
   `git merge branch-name` combines the new branch's changes into the main branch.

(16) - 1. Word Processing Software

Examples: Microsoft Word, Google Docs
These applications are used to create, edit, and format text documents. Features like spell check, templates, and real-time collaboration enhance writing quality and save time.

2. Spreadsheet Software

Examples: Microsoft Excel, Google Sheets
Spreadsheets help manage and analyze data using formulas, functions, and charts. They are essential for budgeting, financial planning, and data-driven decision-making.

3. Presentation Software

Examples: Microsoft PowerPoint, Canva, Google Slides
Presentation software is used to create slide-based visual content. It's commonly used in education, corporate meetings, and training to communicate ideas effectively.

4. Database Management Software (DBMS)

Examples: Microsoft Access, MySQL
These tools allow users to store, manage, and retrieve large volumes of structured data. They are used in business environments for tasks like inventory tracking and customer relationship management.

5. Communication Software

Examples: Zoom, Microsoft Teams, WhatsApp
Communication software enables real-time interaction via text, voice, or video. It's crucial for team collaboration, remote work, and staying connected with clients or peers.

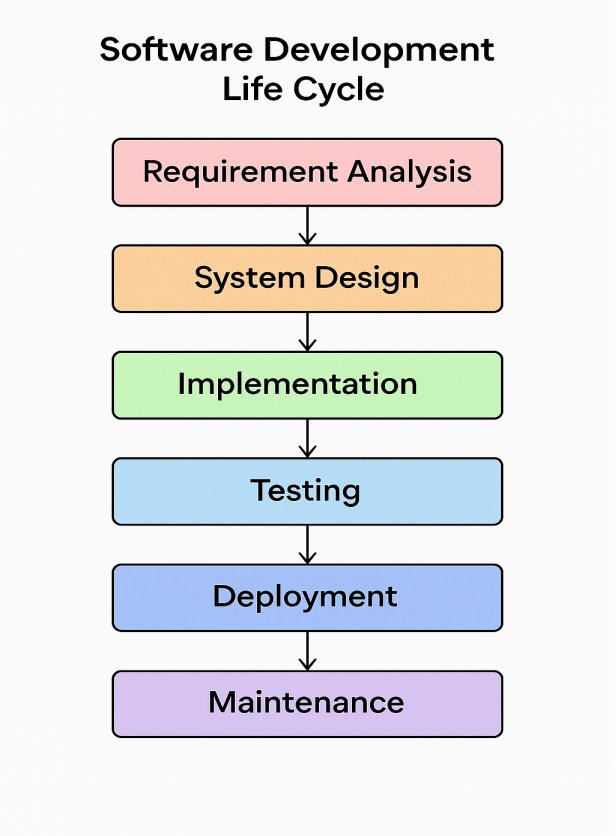
6. Graphics and Multimedia Software

Examples: Adobe Photoshop, CorelDRAW, VLC Media Player
These applications support the creation and editing of visual and audio content. They are widely used in digital design, marketing, video production, and creative media industries.

7. Web Browsers

Examples: Google Chrome, Mozilla Firefox
Web browsers provide access to online content, websites, and cloud-based applications. They are essential for research, learning, communication, and day-to-day web use.

(17) -



(18) - Library management system

1. User Roles

- **Customer: Browses products, places orders, manages profile.**
- **Admin: Manages products, users, and orders.**
- **Seller (optional): Adds and manages their products.**

2. Functional Requirements

A. User Management

- **Register / Login / Logout**
- **Update user profile and change password**
- **View order history and track orders**

B. Product Management

- **View products by categories (e.g., electronics, clothing)**
- **Product search and filtering (by price, rating, etc.)**
- **View product details with images, descriptions, and reviews**

C. Cart Management

- **Add/remove products from cart**
- **View and update product quantity in cart**
- **Calculate total cost dynamically**

D. Checkout & Payment

- **Enter delivery address and contact info**
- **Choose payment method (e.g., card, UPI, COD)**
- **Process payment securely**

E. Order Management

- **Confirm order after payment**
- **Generate order summary/invoice**
- **Track delivery status (pending, shipped, delivered)**

F. Review & Rating System

- **Submit reviews and ratings for purchased products**
- **View reviews left by other customers**

G. Admin Panel

- **Add/update/delete products**
- **View user data and manage accounts**
- **Manage orders (approve, cancel, update status)**

System Features (Non-Functional)

- **Responsive design for mobile/desktop**
- **Secure login with password encryption**
- **Fast search and product load**

(19) - Functional Analysis: Online Shopping System

2. Functionalities

User Management
The system allows users to register and log in with secure credentials. Once logged in, users can manage their profile details such as address, contact number, and password. They can also view their order history to keep track of previous purchases.

Product Management
Users can explore products by browsing through categories, brands, or using keyword-based search. Each product comes with detailed information like price, description, and customer reviews. Administrators have the ability to add new products, update existing listings, or delete products as needed.

Shopping Cart
Users can add products to their shopping cart or remove them at any time. They can view all items currently in the cart, see the total price, and update the quantity of individual items before checking out.

Order Management
Once ready, users can place an order directly from their cart. The system generates a unique order number and provides confirmation. Users can track the status of their order—whether it's pending, shipped, or delivered. If the platform allows, they can also cancel or return an order based on return policies.

Payment Processing
The system offers multiple payment options such as credit/debit cards, UPI, and net banking. Payments are processed through a secure gateway, ensuring safety. After a successful transaction, a receipt is generated and shared with the user.

Search and Filter
Users can search for specific products using names, categories, or brands. Additionally, filters are available to narrow down results based on price range, customer ratings, and stock availability.

Admin Panel

The system includes a dedicated admin panel where administrators can manage users, monitor and update orders, maintain inventory, and handle payment records. It also provides access to sales reports and analytics for better decision-making.

---

3. Supporting Features

The system sends email or SMS notifications to users for order confirmations and shipping updates. Customers can rate and review products they've purchased. There's also a wishlist or "Save for Later" feature, along with options to contact customer support or chat with a chatbot for help.

4. Stakeholders

The key people involved include:

- **Customers, who use the platform to shop and manage orders.**
- **Admins, who oversee operations, products, and user accounts.**
- **Delivery Partners, responsible for shipping and delivering products.**
- **Support Teams, who assist users with complaints, queries, or technical issues.**

(20) - (A)Presentation Layer (Frontend / UI)

Used by customers, delivery agents, and restaurants.

- **Mobile App (iOS/Android)**
    - **Customer App: Browse restaurants, order food**
    - **Delivery App: Accept & track delivery**
    - **Restaurant App: Manage menu & orders**
- **Web Portal**
    - **Admin Dashboard**

(B) Application Layer (Business Logic)

Handles core logic and workflow

- **Order Management Module**
    - **Create, update, cancel orders**
- **Menu & Restaurant Module**
    - **Manage food items and menus**
- **Payment Module**
    - **Payment gateway integration (Razorpay, Stripe, etc.)**
- **User Module**
    - **User registration, login, profiles**
- **Delivery Management**
    - **Assign and track delivery agents**

(C) Data Access Layer (APIs + DB)

Manages communication with the database and external services

- **RESTful APIs (using Node.js / Django / Spring Boot)**
- **Secure Authentication (JWT, OAuth)**
- **External APIs: Maps (for tracking), Payment gateways, SMS/email services**

(D) Database Layer

Stores all data securely

- **Relational DB: PostgreSQL / MySQL**
    - **Users, Orders, Menus, Transactions**
- **NoSQL (Optional): MongoDB**
    - **Real-time chat, logs, analytics**
- **Firebase (Optional): For real-time order tracking**

(E) Admin Layer

Used for management and analytics

- **Admin Panel**
    - **View reports, earnings, feedback**
    - **Manage users/restaurants**

(i) Other Components

- **Notification Service (Push/SMS/Email)**
- **Caching Layer (Redis) – for faster menu loading**
- **CDN – For fast image delivery (food photos, logos)**
- **Logging & Monitoring – Track errors, performance**

(21) - Test Cases

(1) Addition

| TC-01 | Add two positive numbers | 5 + 3 | 8 | - | - |
| TC-02 | Add positive and negative number | 10 + (-4) | 6 | - | - |
| TC-03 | Add two negative numbers | -3 + (-7) | -10 | - | - |
| TC-04 | Add with zero | 0 + 9 | 9 | - | - |

(2) Subtraction

| TC-05 | Subtract two positive numbers | 10 - 4 | 6 | - | - |
| TC-06 | Subtract resulting in negative | 4 - 10 | -6 | - | - |
| TC-07 | Subtract zero | 9 - 0 | 9 | - | - |

(3) Multiplication

| TC-08 | Multiply two positive numbers | 5 × 6 | 30 | - | - |
| TC-09 | Multiply with zero | 9 × 0 | 0 | - | - |
| TC-10 | Multiply with negative number | -3 × 4 | -12 | - | - |

(4) Division

| TC-11 | Divide two positive numbers | 10 ÷ 2 | 5 | - | - |
| TC-12 | Divide with negative | -12 ÷ 3 | -4 | - | - |
| TC-13 | Division by zero | 7 ÷ 0 | Error / Exception | - | - |
| TC-14 | Zero divided by number | 0 ÷ 5 | 0 | - | - |

(5) Invalid Input

| TC-15 | Input is a letter | A + 2 | Error: Invalid Input | - | - |
| TC-16 | Special characters | # × 5 | Error: Invalid Input | - | - |

(22) -

Case: Zoom Security Maintenance (2020)

App: Zoom Video Conferencing
Maintenance Type: Corrective & Adaptive

(i) Problem:

- Security issues during a huge user surge in the COVID-19 pandemic.
- Problems included:
    - Unencrypted meetings
    - Zoombombing
    - Weak privacy settings

(ii) Actions Taken:

- Fixed bugs (corrective maintenance)
- Added features like waiting rooms, passwords, and report button (adaptive)
- Ran 90-day security audit
- Bought Keybase to improve encryption

(iii) Result:

- Security improved
- User trust restored
- Continued platform growth

(23) - DFD – Level 1 for Hospital Management System

External Entities:

1. **Patient**
2. **Doctor**
3. **Admin/Receptionist**
4. **Pharmacy**
5. **Lab**

Processes:

1. **Register Patient**
2. **Schedule Appointment**
3. **Manage Medical Records**
4. **Generate Bill**
5. **Prescribe Medicine**
6. **Lab Test Management**

Data Stores:

- D1: Patient Records
- D2: Appointment Data
- D3: Medical History
- D4: Billing Information
- D5: Prescription Records
- D6: Lab Reports

Flow Summary:

1. **Patient** → fills info → **Register Patient** → stores in **Patient Records**
2. **Patient** requests → **Schedule Appointment** → stores in **Appointment Data**
3. **Doctor** → accesses **Patient Records**, writes to **Medical History**
4. **Doctor** → writes Rx → **Prescribe Medicine** → updates **Prescription Records**
5. **Lab** updates → **Lab Test Management** → stores **Lab Reports**
6. **Admin** → views data → **Generate Bill** → updates **Billing Information**

(24) -