# Lab Exercise

**A-1:** program in git [A1].

**A-2:**

$$\begin{tikzcd}\textbf{Client} \arrow[r, shift left, "\textbf{Internet}"] & \textbf{Server} \arrow[l, shift left]\end{tikzcd}$$

**A-3:**

**HTTP client-server communication**

- There are two sides: **Client** and **Server**.
- The **client** sends a request to the **server.**
- The **server** receives the request, processes it, and sends back a response.

**How it works step-by-step:**

1. The **server** runs continuously, waiting for requests.
2. The **client** sends an HTTP request .
3. The **server** reads the request and prepares a response.
4. The **server** sends the response back to the client.
5. The **client** receives and displays or uses the response.

This process happens over the internet or local network using the **HTTP protocol**, which is just a set of rules for how the client and server should communicate.

## A-4:

**1. Broadband (DSL or Cable) :** Internet through a phone line (DSL) or cable TV line (Cable).

**Pros:**

- Works well for most homes.
- Can watch videos and play games.
- Always connected

**Cons:**

- Slower than fiber.
- Speed can go down if many people use it at the same time.

**2. Fiber (Fiber Optic) :** Internet through tiny glass cables that use light to send data.

**Pros:**

- Very fast (great for streaming, gaming, working).
- More reliable
- Same speed for downloading and uploading.

**Cons:**

- May not be in all places.
- More expensive to install.

**3. Satellite:** Internet from a satellite in space, often used in faraway places.

**Pros:**

- Works in remote areas (no wires needed).
- Can get it almost anywhere.

**Cons:**

- Slower than other types.
- Bad weather can make it stop.

**4. Mobile (3G, 4G, 5G) :** Internet on your phone using mobile data (cell towers).

**Pros:**

- Easy to use anywhere with a signal.
- Fast with 4G or 5G.
- No wires needed.

**Cons:**

- Uses data plan (can cost more).
- Can be slow if signal is weak.

**5. Dial-Up:** Old internet using a phone line. You hear beeping sounds when it connects.

**Pros:**

- Very cheap.
- Can work where nothing else is.

**Cons:**

- Very slow.
- Can’t use phone and internet at the same time.

## A-5:

**Simulating HTTP Requests Using Command Line Tools**

When you use a command line tool like **`curl`**, you can send a request to a server over the internet.

- You can ask a server to **send you a webpage** (this is called a GET request).
- You can also **send data to the server**, like filling out a form (this is usually a POST request).

**Simulating FTP Requests Using Command Line Tools**

FTP is used for transferring files. With a tool like **`curl`**, you can:

- **Download a file** from an FTP server to your computer.
- **Upload a file** from your computer to an FTP server.

## A-6:

### 1. SQL Injection

**Definition:**
A security vulnerability that allows an attacker to interfere with the queries an application makes to its database by injecting malicious SQL code.

### 2. Cross-Site Scripting (XSS)

**Definition:**
A vulnerability that allows attackers to inject malicious scripts into web pages, which then run in the browsers of other users, potentially stealing data or altering site behavior.

### 3. Broken Authentication

**Definition:**
A flaw in the authentication process that allows attackers to gain unauthorized access to user accounts, often due to weak passwords, poor session management, or missing security controls.

## A-7:

**System Software**: Helps run the computer itself

| Application | Explanation |
|---|---|
| Windows Operating System | Manages hardware and runs other software. |
| File Explorer | Helps manage files and folders in the system. |

**Application Software**: Lets you do specific tasks (like browsing, writing, messaging).

| Application | Explanation |
|---|---|
| Google Chrome | Used for browsing the internet. |
| Microsoft Word | Used for creating and editing documents. |

## A-8:

**Presentation Tier (Client/UI Layer) :** This is the topmost layer of the application that interacts directly with the user. It displays information and collects input from the user.

**Example**: Web browser, mobile app.

**Application Tier (Logic Layer) :** This middle layer processes the business logic, rules, and data. It acts as a bridge between the user interface and the database.

**Example**: Web server, backend code (like Java, Python, Node.js).

**Data Tier (Database Layer) :** This bottom layer manages and stores data. It handles database operations such as insert, update, delete, and query.

**Example**: MySQL, MongoDB, PostgreSQL

## A-9:

**1. Presentation Layer :** The presentation layer is the top layer of a software system responsible for displaying information to the user and collecting user input. It provides the user interface and handles interaction between the user and the system.

**Functionality:**

- It is responsible for **interacting with the user**.
- Displays data to the user and collects input.
- Sends user input to the business logic layer.

**2. Business Logic Layer (Application Layer) :** The business logic layer is the middle layer that processes user input, enforces business rules, performs calculations, and manages the flow of data between the presentation and data access layers.

**Functionality:**

- Contains the **core logic and rules** of the application.
- Processes user input, applies rules, and makes decisions.
- Communicates between the presentation and data layers.

**3. Data Access Layer :** The data access layer is responsible for communicating with the database. It handles the retrieval, insertion, updating, and deletion of data, ensuring secure and efficient access to the data storage system.

**Functionality:**

- Responsible for **storing, retrieving, and managing data** from the database.
- Handles all direct communication with the database system.

## A-10:

**1. Development Environment :** The development environment is where software developers write, build, and initially test their code.

- **Purpose**:
  It allows developers to experiment, create new features, and fix bugs without affecting users or other environments.

- **Key Feature:**
  This environment typically includes code editors, debugging tools, and local versions of databases or services to support active development.

**2. Testing Environment :** The testing environment is used to test the software for functionality, bugs, and performance before it's released to users.

- **Purpose:**
  It helps ensure that the application behaves correctly by simulating user actions and running test cases in a controlled setting.

- **Key Feature:**
  It often replicates production like conditions and uses automated or manual testing tools to validate the software's quality and stability.

**3. Production Environment :** The production environment is the live environment where the final software runs and is accessed by real users.

- **Purpose**:
  It delivers the working application to end users, so it must be stable, secure, and optimized for performance.

- **Key Feature:**
  This environment is closely monitored and maintained to handle real traffic, data, and user activity with minimal downtime.

## A-11: Uploaded to github

## A-12:

To commit the changes and push the code to github, here are the steps to be followed :

**Check for Changes:**
First, we use **git status** to review which files have been added, modified, or deleted in the working directory.

**Stage the Changes:**
Next, we prepare the files for commit using **git add**. This tells Git which changes we want to include in the next commit.

**Commit the Changes:**
We then use **git commit -m "commit message"** to record the changes.

**Push to Remote Repository:**
Finally, we use **git push origin branch-name** to upload our committed changes to a remote repository like GitHub. This makes the changes visible to others and keeps the remote branch updated.

**A-13:** Already created

**A-14:**

**System Software :** These manage the hardware and core functions of a computer.

1. Windows Operating System
2. macOS
3. Linux
4. Android OS
5. BIOS (Basic Input Output System)

**Application Software :** These are used to perform specific tasks for the user.

1. Google Chrome (web browsing)
2. Microsoft Word (document editing)
3. WhatsApp (communication)
4. VLC Media Player (media playback)
5. Zoom (video conferencing)

**Utility Software :** These help in system maintenance and performance optimization.

1. Antivirus software (e.g., Avast, Windows Defender)
2. WinRAR / 7-Zip (file compression)
3. CCleaner (cleaning junk files)
4. Disk Cleanup (Windows utility)
5. Backup & Restore tools

## A-15:

1. **Cloning a Repository: C**loning an existing Git repository from a remote platform like GitHub using the **git clone** command. This created a local copy of the entire project on my computer, allowing me to work on it independently.

2. **Creating and Switching Branches:** Created a new branch using **git branch branch-name** and switched to it using **git checkout branch-name**.

3. **Making Changes and Committing:** Making some changes to files, then staged them using **git add**, and committed them with a message using **git commit -m "Commit message"**.

4. **Merging Branches:** After testing the changes, I merged the new branch into the main branch using **git merge branch-name**. This combined the changes into the main project.

## A-16:

Application software refers to programs designed to perform specific tasks for users, helping them complete work more efficiently and effectively.

These software tools are essential in everyday personal, academic, and professional tasks, and they significantly boost productivity by automating processes, organizing information, and enabling communication.

**1. Word Processing Software :** These tools allow users to create, edit, format, and share documents. Features like spell check, templates, and real-time collaboration save time and enhance the quality of writing tasks. **Examples:** Microsoft Word, Google Docs

**2. Spreadsheet Software :** Spreadsheets are used for organizing data, performing calculations, and analyzing information using formulas and charts. They are vital for budgeting, reporting, and data-driven decision-making. **Examples:** Microsoft Excel, Google Sheets

**3. Presentation Software :** Presentation tools help users convey ideas visually and clearly through slideshows. They are widely used in education, business meetings, and training sessions, making information easier to understand and remember. **Examples:** Microsoft PowerPoint, Canva, Google Slides

**4. Database Management Software :** These tools store, retrieve, and manage large sets of structured data efficiently. They help businesses keep records, manage inventory, and support customer relationship management (CRM). **Examples:** Microsoft Access, MySQL

**5. Communication Software :** These tools enable instant communication through text, audio, or video. They are essential for remote work, team collaboration, and staying connected with clients and colleagues. **Examples:** Zoom, Microsoft Teams, WhatsApp

**6. Graphics and Multimedia Software :** These applications allow users to create and edit images, videos, and audio. They are crucial in marketing, design, content creation, and media industries. **Examples:** Adobe Photoshop, CorelDRAW, VLC Media Player

**7. Web Browsers :** Browsers give users access to information, web applications, and online services. They are gateways to online learning, research, and cloud-based work. **Examples:** Google Chrome, Mozilla Firefox

A-17:

```
+------------------------+
|  1. Requirement    |
|     Gathering         |
+-----------+------------+
            |
            V
+------------------------------+
|  2. System Analysis      |
|  & Feasibility Study     |
+-----------+------------------+
            |
            V
+------------------------------+
|  3. System Design       |
+-----------+-----------------+
            |
            V
+----------------------------------+
|  4. Implementation             |
|   (Coding/Development)    |
+-----------+----------------------+
            |
            V
+------------------------+
|  5. Testing            |
+-----------+------------+
            |
            V
```

```
+------------------------+
|  6. Deployment    |
+-----------+------------+
            |
           V
+------------------------+
|  7. Maintenance   |
+------------------------+
```

## A-18:

**For: Simple Library Management System**

### 1. Introduction

This document outlines the functional and non-functional requirements for a simple Library Management System. The system will help manage books, users, and borrowing activities within a library.

### 2. Objectives

- To automate book issue and return processes
- To maintain a database of books and users
- To generate reports on issued/returned books

### 3. Functional Requirements

1. **User Registration and Login**
   Users (students/librarians) should be able to register and log in securely.

2. **Book Management**
   Add, update, delete, and search for books in the system.

3. **Issue/Return Books**
   Librarians can issue or return books to/from registered users and track due dates and overdue books.

4. **User Management**
   View user profiles and their borrowing history.

5. **Search and Filter**
   Users can search books by title, author, genre, or availability.

6. **Report Generation**
   Generate reports for issued books, returned books, and fines.

## 4. Non-Functional Requirements

1. **Usability**
   The system should have a simple and user-friendly interface.

2. **Performance**
   The system should respond quickly (within 2 seconds for search and navigation).

3. **Security**
   Passwords should be encrypted, and access should be role-based (e.g., user vs. librarian).

4. **Scalability**
   The system should handle an increasing number of users and books without performance loss.

## 5. System Users

- **Librarian:** Manages all operations including book and user records.
- **Student/User:** Can view and borrow books.

### 6. Assumptions

- The system is to be used in a school/college library.
- Only registered users can borrow books.
- Internet access is required if hosted online.

## A-19:

### Functional Analysis: Online Shopping System

### 1. Overview

An Online Shopping System allows users to browse products, add them to a cart, make payments, and receive orders. It also includes functions for managing users, products, orders, and payments.

### 2. Main Functionalities

#### Admin Panel

- Manage users, orders, inventory, and payments
- View sales reports and analytics

#### User Management

- User registration and login
- Profile management (address, contact info, password)
- View order history

#### Search and Filter

- Search products by name, category, or brand
- Filter by price range, ratings, or availability

**Product Management**

- Browse products by category, brand, or keyword
- View product details (price, description, reviews)
- Admin can add, update, or delete product listings

**Order Management**

- Place orders from the cart
- Generate order number and confirmation
- Track order status (pending, shipped, delivered)
- Cancel or return an order (based on policy)

**Payment Processing**

- Multiple payment options (credit/debit card, UPI, net banking)
- Secure payment gateway integration
- Generate payment receipt

**Shopping Cart**

- Add/remove items to/from the cart
- View cart items with quantity and total cost
- Update quantity of items

## 3. Supporting Features

- Email/SMS notifications for order confirmation and shipping
- Review and rating system for products
- Wishlist or "Save for later" feature
- Customer support or chatbot integration

## 4. Stakeholders

- **Customers** – End users buying products

- **Admin** – Manages platform operations
- **Delivery Partner** – Handles shipment and delivery
- **Support Team** – Responds to user issues or complaints

## A-20:

**Basic Architecture Layers**

**1. Presentation Layer (Frontend)**

  - Customers (browse, order, pay, track)
  - Restaurants (manage menu & orders)
  - Delivery Agents (view and accept deliveries)

**2. Application Layer (Backend/Business Logic)**

  - Order management
  - Payment processing
  - User authentication
  - Notification system (order updates)
  - Delivery assignment logic

**3. Data Layer (Database)**

  - User data (customers, restaurants, delivery agents)
  - Menu & item details
  - Orders & payment records
  - Ratings & reviews

**4. External Integrations**

- **Payment Gateway** – for secure online payments (e.g., Razorpay, Paytm)
- **Map Services** – for real-time tracking and route optimization (e.g., Google Maps API)
- **Notification Service** – for SMS, email, or app notifications (e.g., Firebase Cloud Messaging)

## A-21:

The calculator performs basic **arithmetic operations** including addition, subtraction, multiplication, and division.

The test cases include a variety of input combinations such as positive numbers, negative numbers, decimal values, zero, and special cases like division by zero.

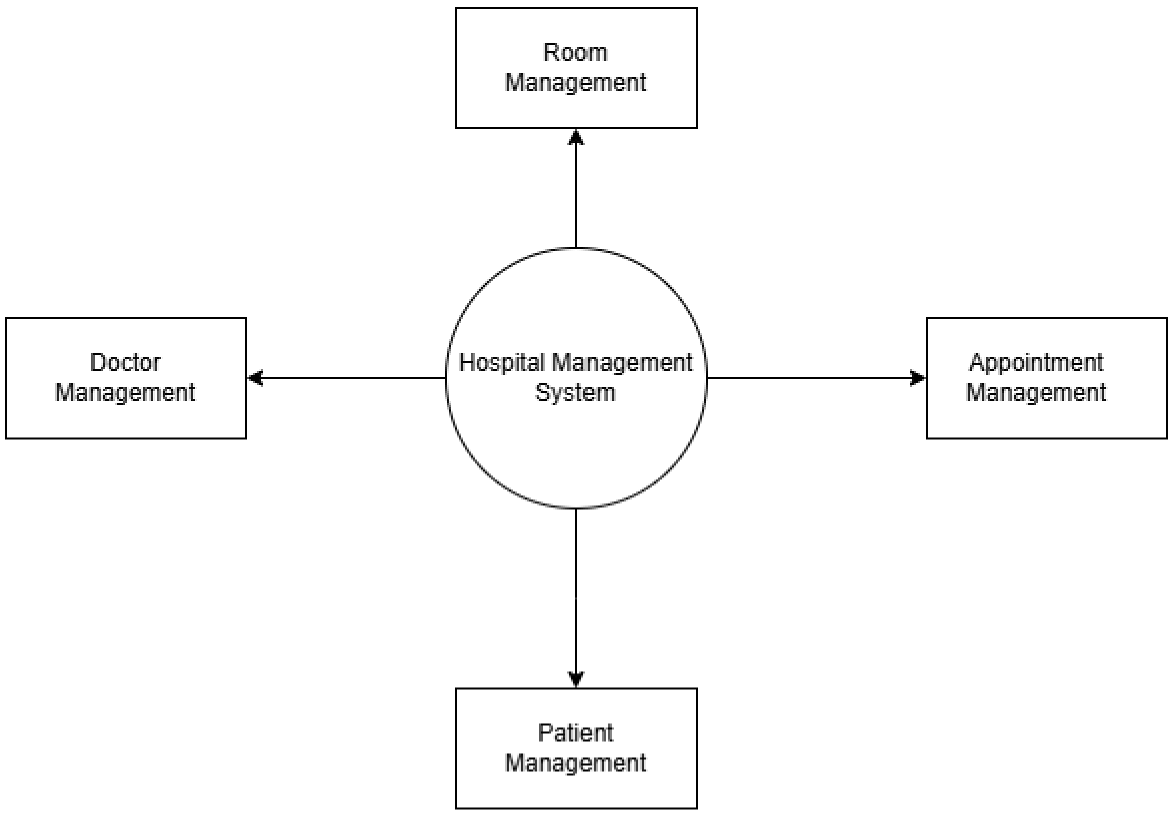
## A-22:

A notable case of critical **software maintenance** occurred with the HealthCare.gov website in 2013. Following its launch, the site experienced severe performance issues, leading to frequent crashes and user access problems.

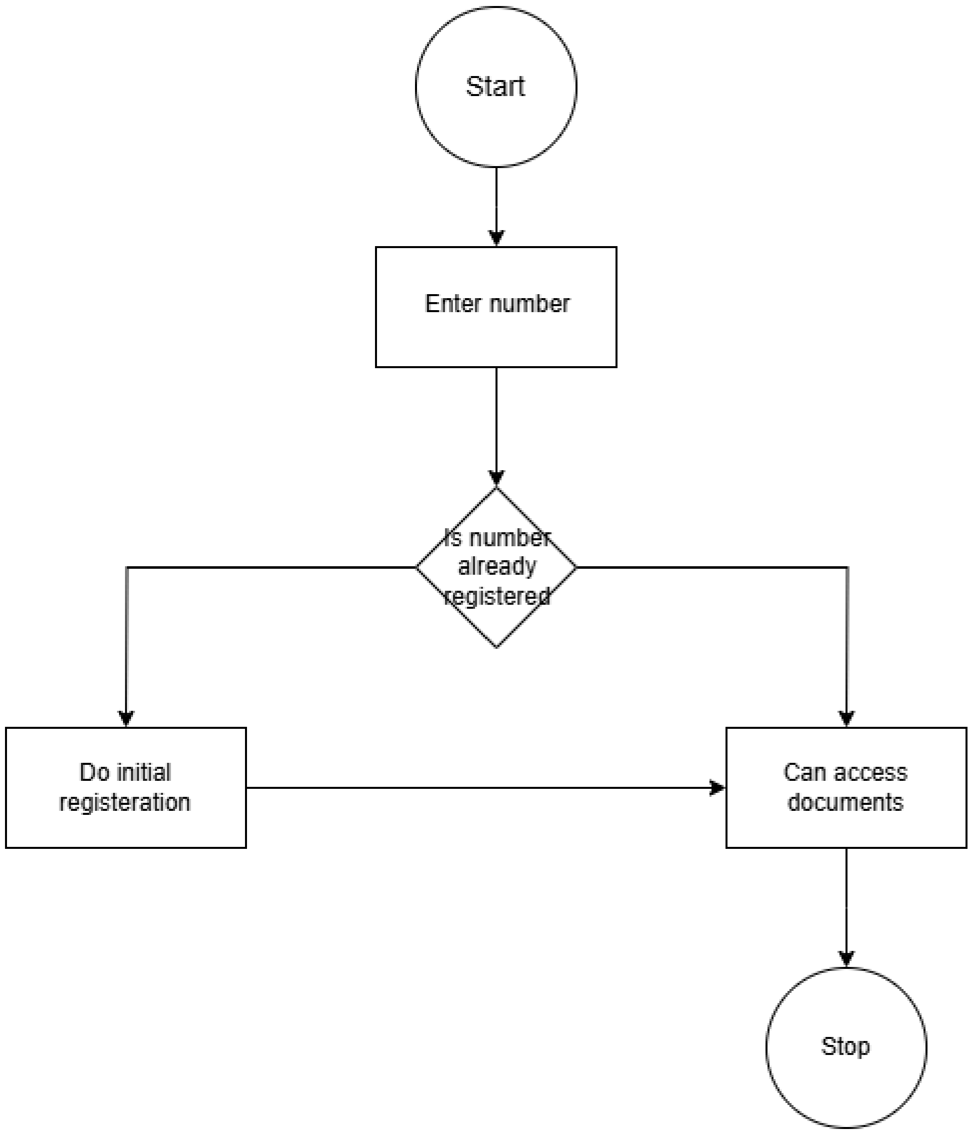
The U.S. government initiated an extensive overhaul involving debugging, performance tuning, and optimizing server capacity. This maintenance was crucial to ensure users could effectively enroll in health insurance plans.

By December 2013, the site performance improved significantly, demonstrating the importance of timely and effective maintenance for software applications, especially in critical public services.

## A-23:


Room Management
Doctor Management
Hospital Management System
Appointment Management
Patient Management

## A-24:


Start
Enter number
Is number already registered
Do initial registeration
Can access documents
Stop