Assignment - 1

1. What is web development?

Web development is the process of **building, creating, and maintaining websites and web applications** that run on the internet or an internal network (intranet).

It involves everything from designing how a website looks to developing how it works and managing data behind the scenes.

Key Aspects of Web Development

1. Front-End Development (Client Side)

This focuses on what users **see and interact with** in their browser.

- **Technologies used:**
  - HTML – structure of the webpage
  - CSS – styling and layout
  - JavaScript – interactivity and dynamic behavior
- **Examples:** buttons, forms, animations, responsive design

2. Back-End Development (Server Side)

This handles the **logic, database, and server operations**.

- **Technologies used:**
  - Programming languages: Python, Java, PHP, Node.js
  - Frameworks: Flask, Django, Spring Boot
  - Databases: MySQL, PostgreSQL, MongoDB
- **Examples:** user authentication, data processing, APIs

3. Full-Stack Development

A **full-stack developer** works on both front-end and back-end parts of a website or application.

What Web Development Is Used For

- Business websites
- E-commerce platforms
- Social media applications
- Cloud-based systems
- Dashboards and web portals

2. Difference between frontend and backend with examples.

Difference Between Frontend and Backend (with Examples)

Aspect	Frontend (Client Side)	Backend (Server Side)
Meaning	Frontend is the part of a website that users can see and interact with	Backend is the part that works behind the scenes
Focus	User interface and user experience (UI/UX)	Business logic, data processing, and server management
Runs On	Web browser (Chrome, Firefox, Safari)	Web server
Main Purpose	Displays content and collects user input	Processes requests and sends responses
Languages Used	HTML, CSS, JavaScript	Python, Java, PHP, Node.js
Frameworks	React, Angular, Vue.js	Django, Flask, Spring Boot, Express
Database Access	No direct database access	Directly interacts with databases
Security	Less secure (code visible to users)	More secure (logic hidden on server)
Example Tasks	Designing forms, buttons, layouts	User authentication, data storage

3. Explain client–server communication with a neat diagram.

Client–Server Communication

Client–server communication is a model in which a **client** (such as a web browser or mobile app) sends a **request** to a **server**, and the server processes the request and sends back a **response**.

How Client–Server Communication Works

1. **Client Request**
   The client sends a request to the server (e.g., opening a website, submitting a form).

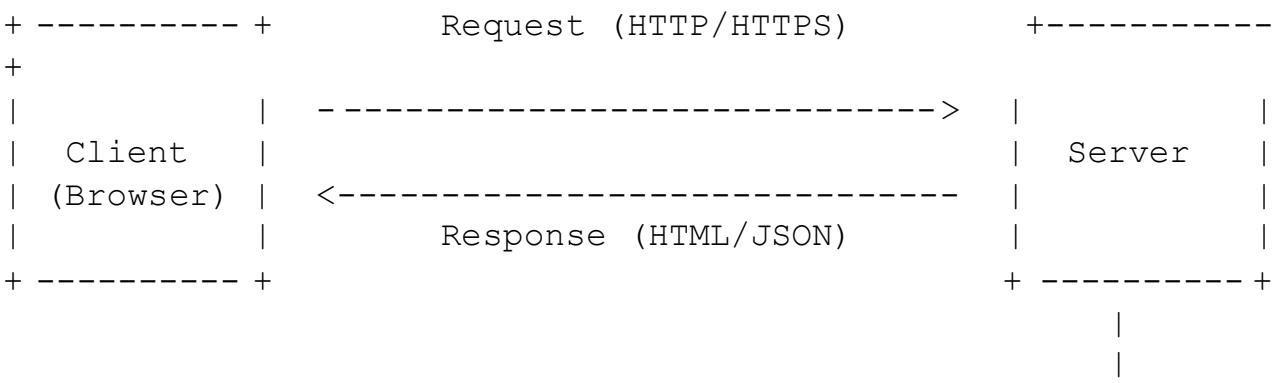
2. **Network/Internet**
   The request travels through the internet using protocols like **HTTP/HTTPS**.

3. **Server Processing**
   The server receives the request, executes business logic, accesses the database if required, and prepares a response.

4. **Server Response**
   The server sends a response back to the client (e.g., web page, JSON data, error message).

5. **Client Display**
   The client displays the received response to the user.

Neat Diagram: Client–Server Communication



4. What is MEAN stack?

MEAN Stack

The **MEAN stack** is a **full-stack web development framework** that uses **JavaScript at every layer** of application development.

The word **MEAN** is an acronym formed from the four core technologies used in the stack.

Components of MEAN Stack

1. **M – MongoDB**
   - A **NoSQL database** used to store data in JSON-like documents
   - Data is stored in a flexible, schema-less format
2. **E – Express.js**
   - A **backend web application framework** for Node.js
   - Handles routing, middleware, and server-side logic

3. **A – Angular**
    - A **frontend framework** developed by Google
    - Used to build dynamic, single-page web applications (SPAs)
4. **N – Node.js**
    - A **JavaScript runtime environment**
    - Executes JavaScript code on the server side

5. Install Angular CLI using npm install -g @angular/cli.

Install Angular CLI using npm

Follow these steps to install **Angular CLI** using **npm**:

Step 1: Check Node.js and npm Installation

Angular CLI requires **Node.js** (LTS version recommended).

Run:

```
node -v
npm -v
```

If not installed, download Node.js from the official site (it installs npm automatically).

Step 2: Install Angular CLI Globally

Run the following command in the terminal:

```
npm install -g @angular/cli
```

- `-g` → installs Angular CLI globally
- `@angular/cli` → Angular Command Line Interface package

Step 3: Verify Installation

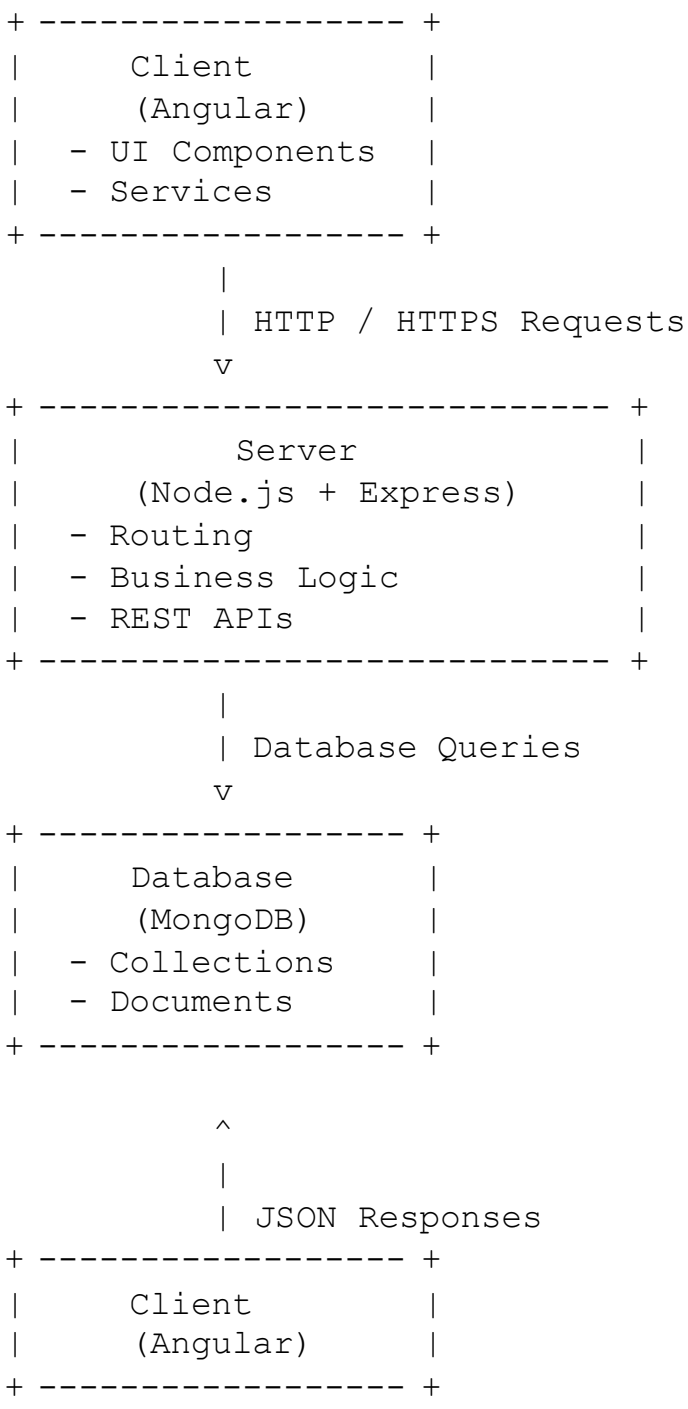
Check if Angular CLI is installed correctly:

```
ng version
```

6. Draw the MEAN architecture workflow: Client (Angular) → Server (Express + Node.js) → Database (MongoDB)

MEAN Architecture Workflow

Below is a **neat and simple diagram** showing the **MEAN stack architecture workflow**:



Explanation (Exam-Friendly)

- **Angular (Client):** Handles user interface and sends requests
- **Node.js + Express (Server):** Processes requests and manages APIs
- **MongoDB (Database):** Stores and retrieves application data
- **Communication:** Uses HTTP/HTTPS and JSON data format