# MERN_ASSIGNMENT – 1

## 1. What is Web Development?

**Web development** is the process of designing, building, and maintaining websites and web applications that run on the internet or intranet. It involves creating user-friendly interfaces, handling server-side logic, managing databases, and ensuring secure communication between systems.

Web development is broadly divided into:

- **Frontend development** – what users see and interact with
- **Backend development** – server, database, and application logic

Modern web development uses technologies such as HTML, CSS, JavaScript, frameworks, APIs, and databases to deliver dynamic and scalable applications.

## 2. Difference Between Frontend and Backend (With Examples)

| Aspect | Frontend Development | Backend Development |
|---|---|---|
| Definition | Client-side part of the application | Server-side part of the application |
| Purpose | User interface and user experience | Business logic and data handling |
| Runs On | User's browser | Server |
| Technologies | HTML, CSS, JavaScript, Angular, React | Node.js, Express, Java, Python, PHP |

| Aspect | Frontend Development | Backend Development |
|---|---|---|
| Database Access | No direct access | Direct access |
| Example | Login page UI | Login authentication logic |

**Example:**

- **Frontend**: Angular form where the user enters email and password
- **Backend**: Node.js + Express API that verifies credentials from MongoDB

**3. Explain client-server communication with a neat diagram.**

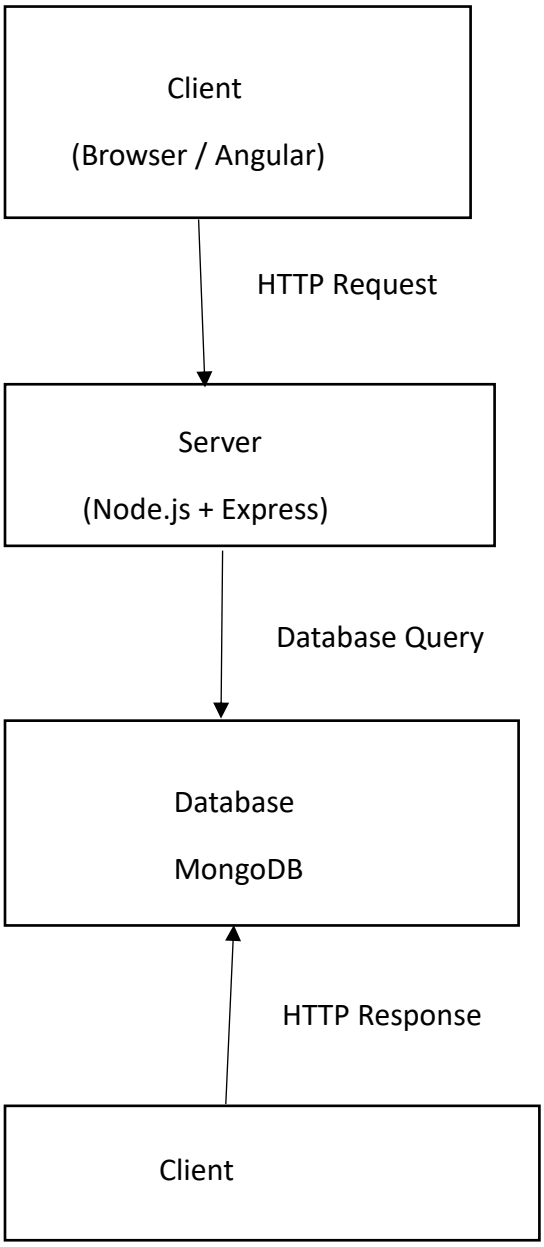
**Client–server communication** is a model where the client (browser or frontend app) sends requests to the server, which processes them and returns responses. It follows a **request-response cycle**.

**Steps:**

1. **Client Sends HTTP Request**
    - The client sends a request to the server using **GET, POST, PUT, or DELETE**.
    - Example: User submits login credentials via a form.
2. **Server Processes the Request**
    - The server validates data, applies business logic, and determines the required database operations.
3. **Server Interacts with Database**
    - Server performs CRUD operations on the database (e.g., MongoDB) as needed.
4. **Server Sends HTTP Response**
    - Server returns a response with **status code** and **data** (usually JSON).

- Example: Successful login returns a JSON with user ID and status message.

5. **Client Updates UI**
    - Client processes the response and updates the interface accordingly.



**4. What is MEAN Stack?**

MEAN Stack is a full-stack JavaScript framework used to develop dynamic web applications. It allows developers to use JavaScript across the entire application, from frontend to backend and database.

Components of MEAN Stack:

| Component | Description |
|---|---|
| M – MongoDB | NoSQL database to store data in JSON-like documents. |
| E – Express.js | Backend web framework for Node.js to handle routing and APIs. |
| A – Angular | Frontend framework to build dynamic, single-page applications. |
| N – Node.js | JavaScript runtime environment to run server-side code. |

Advantages:

- Single language (JavaScript) for client and server
- Scalable and high-performance applications
- Ideal for single-page applications (SPA)
- Supports MVC (Model-View-Controller) architecture

Workflow:

1. Angular (client) sends requests to Express + Node.js (server).
2. Server processes requests and interacts with MongoDB.
3. Server responds with data to Angular to update the UI.

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

**Angular CLI (Command Line Interface)** is a tool that helps developers **create, build, and manage Angular applications** efficiently. It provides commands to generate components, services, modules, and handle project configuration.

**Prerequisites:**

- **Node.js** installed on your system
- **npm (Node Package Manager)** (comes with Node.js)

---

**Step to Install Angular CLI Globally:**

Open a terminal/command prompt and run:

npm install -g @angular/cli

- -g flag installs Angular CLI **globally**, so it can be used in any folder.

---

**Verify Installation:**

ng version

- This displays Angular CLI version and environment details, confirming successful installation.

**6. Draw the MEAN Architecture workflow:**

**Client(Angular) -> Server(Express + Node.js) -> Database(MongoDB)**

The **MEAN Stack** follows a **client–server–database architecture**, where each component has a specific role:

**Workflow:**

1. **Client (Angular)**
    - Handles the **user interface** and **user interactions**.
    - Sends **HTTP requests** (GET, POST, PUT, DELETE) to the server via APIs.

2. **Server (Node.js + Express.js)**
    - Receives and **processes requests** from the client.
    - Applies **business logic** and interacts with the database.
    - Returns **HTTP responses** (usually JSON) to the client.
3. **Database (MongoDB)**
    - Stores and retrieves application data in **JSON-like documents**.
    - Supports CRUD operations requested by the server.

**DIAGRAM: -**