

ASSIGNMENT 1 – Web Development Basics

1. Difference between Frontend, Backend, and Full-Stack Development:

- **Frontend Development:** Focuses on the user interface and user experience. It uses technologies like HTML, CSS, and JavaScript. Example: Designing a website's homepage layout and buttons.
- **Backend Development:** Manages the logic, database, and server-side operations. Technologies include Node.js, Python, PHP, and Java. Example: Processing login credentials and fetching user data.
- **Full-Stack Development:** Involves both frontend and backend work. Example: A developer who builds both the website's interface and the server logic.

2. Client-Server Model Diagram:

A simple representation of how clients and servers communicate in web architecture.
Client (Browser) → Internet → Web Server → Database → Web Server → Internet → Client (Browser)

3. Browser Request and Display Process:

1. The user enters a URL in the browser.
2. The browser sends an HTTP/HTTPS request to the web server.
3. The server processes the request and sends back HTML, CSS, and JS files.
4. The browser renders the page using its rendering engine and displays it to the user.

4. Tools Required for Web Development Environment:

- **VS Code:** Code editor for writing and managing code.
- **Browser:** To test and view web pages.
- **Node.js:** To run JavaScript outside the browser.
- **Git:** For version control.
- **Live Server Extension:** For real-time preview of changes.

5. Web Server Explanation:

A web server stores, processes, and delivers web pages to users upon request. Examples: Apache, Nginx, Microsoft IIS, and LiteSpeed.

6. Roles in a Web Project:

- **Frontend Developer:** Designs the interface and ensures responsiveness.
- **Backend Developer:** Handles databases, authentication, and APIs.
- **Database Administrator (DBA):** Manages database design, security, and performance.

7. VS Code Setup:

Install Visual Studio Code and add extensions like Live Server, Prettier, and JavaScript (ES6) snippets. Open an HTML file, right-click, and choose 'Open with Live Server' to test.

8. Static vs Dynamic Websites:

- **Static Website:** Content remains the same for all users. Example: Portfolio website.
- **Dynamic Website:** Content changes based on user interaction or data. Example: E-commerce website like Amazon.

9. Five Web Browsers and Rendering Engines:

1. Google Chrome – Blink
2. Mozilla Firefox – Gecko
3. Microsoft Edge – Blink
4. Safari – WebKit
5. Opera – Blink

Rendering engines interpret HTML, CSS, and JS differently, affecting how web pages appear.

10. Basic Web Architecture Flow:

Client → Web Server → Application Logic → Database

APIs act as intermediaries that allow the frontend to communicate securely with the backend and database.