# ASSIGNMENT 1

## 1. Explain the difference between frontend, backend, and full-stack development with suitable real-world examples.

Frontend development involves designing the user interface that users interact with directly using HTML, CSS, and JavaScript. For example, the visible layout of Amazon’s homepage is built by frontend developers.  
  
Backend development focuses on server-side logic, databases, and application functionality using languages like Python, Java, or Node.js. For example, when you log in to Amazon, the authentication process runs on the backend.  
  
Full-stack development combines both frontend and backend work, allowing developers to build entire applications. For instance, a full-stack developer can design the webpage and also manage the server that powers it.

## 2. Create a simple diagram showing how the client-server model works in web architecture.

Simple Client-Server Model:  
  
Client (Browser) → Sends request → Web Server → Processes request → Database (if needed) → Server sends response → Client displays page.

## 3. Describe how a browser requests and displays a web page from a web server.

When a user enters a URL, the browser sends an HTTP/HTTPS request to the web server hosting the website. The server processes the request and returns the corresponding HTML file along with other resources such as CSS, JS, and images. The browser then renders these files to visually display the webpage to the user.

## 4. Identify and list the tools required to set up a web development environment. Explain the purpose of each.

1. Text Editor (e.g., VS Code) – To write and edit code.  
2. Web Browser (e.g., Chrome, Firefox) – To test and view web pages.  
3. Version Control (e.g., Git) – To track and manage code changes.  
4. Local Server (e.g., XAMPP) – To test dynamic websites locally.  
5. Terminal/Command Line – To run scripts and commands efficiently.

## 5. Explain what a web server is and give examples of commonly used servers.

A web server is software that stores, processes, and delivers web pages to clients over the internet using HTTP/HTTPS. Examples include Apache, Nginx, Microsoft IIS, and LiteSpeed.

## 6. Define the roles of a frontend developer, backend developer, and database administrator in a project.

Frontend Developer – Designs and implements the user interface.  
Backend Developer – Manages server-side logic and data processing.  
Database Administrator (DBA) – Oversees database design, maintenance, and security.

## 7. Install VS Code and configure it for HTML, CSS, and JavaScript development. Take a screenshot of the setup.

Steps:  
1. Download and install Visual Studio Code from its official website.  
2. Install extensions: ‘Live Server’, ‘Prettier’, and ‘HTML CSS Support’.  
3. Open a project folder and start coding.  
4. (Insert screenshot of your configured setup here.)

## 8. Explain the difference between static and dynamic websites. Provide an example of each.

Static websites display fixed content and are built with HTML and CSS only (e.g., a personal portfolio page).  
Dynamic websites fetch content from a server or database and change based on user input (e.g., Facebook, YouTube).

## 9. Research and list five web browsers. Explain how rendering engines differ between them.

Five popular browsers:  
1. Google Chrome – Blink engine  
2. Mozilla Firefox – Gecko engine  
3. Microsoft Edge – Blink engine  
4. Apple Safari – WebKit engine  
5. Opera – Blink engine  
  
Rendering engines determine how HTML, CSS, and JS are interpreted and displayed. Different engines may render pages slightly differently based on their processing algorithms.

## 10. Draw a labeled diagram showing the basic web architecture flow — client, server, database, and APIs.

Basic Web Architecture Flow:  
  
Client (Browser) ↔ API Request ↔ Server ↔ Database  
  
The client sends requests via APIs to the server, which retrieves or stores data in the database and sends a response back to the client.