Assignment-1

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

Answer.

Frontend Development:

- Focuses on the part of a website or application that users interact with directly.
- Uses technologies like **HTML, CSS, JavaScript, React, Angular**.
- Example: The layout, buttons, menus, and forms on **Amazon's homepage**.

Backend Development:

- Handles server-side logic, databases, and application functionality.
- Uses technologies like **Node.js, Django, PHP, SQL**.
- Example: Processing orders, storing user data, or authenticating login credentials on **Amazon's server**.

Full-Stack Development:

- Combines both frontend and backend development skills.
- A full-stack developer can build **both the user interface and server logic**.
- Example: A developer building a blog platform that allows users to create posts (frontend) and saves posts in a database (backend)

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

Answer.

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

Answer.

1. **User Request:**
   - The user enters a URL in the browser or clicks a link.
   - The browser identifies the server using the domain name (DNS lookup converts it to an IP address).
2. **HTTP Request:**
   - The browser sends an **HTTP/HTTPS request** to the web server requesting the web page.
3. **Server Processing:**
   - The server receives the request, processes it, and may fetch data from a database or run backend logic.
4. **HTTP Response:**
   - The server sends back an **HTTP response** containing HTML, CSS, JavaScript, and other resources (images, videos, etc.).
5. **Rendering the Page:**
   - The browser parses the HTML, applies CSS styles, executes JavaScript, and renders the final web page on the screen.
6. **Additional Requests:**
   - The browser may send additional requests for images, scripts, or API data needed for the page.

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

Answer.

Code Editor (e.g., Visual Studio Code):

- Purpose: To write, edit, and organize code efficiently. Provides syntax highlighting, extensions, and debugging tools.

Web Browser (e.g., Google Chrome, Firefox, Microsoft Edge):

- Purpose: To test and view web pages; browsers render HTML, CSS, and execute JavaScript.

Version Control System (Git):

- Purpose: To track changes in code, collaborate with others, and manage versions of the project.

Terminal / Command Line:

- Purpose: To run commands for Git, Node.js, package managers, and server scripts.

Package Manager (e.g., npm or Yarn):

- Purpose: To install and manage libraries, frameworks, and dependencies for the project.

Web Server / Local Server (e.g., Node.js, XAMPP):

- Purpose: To run and test web applications locally before deploying to production.

Browser Developer Tools:

- Purpose: To debug and inspect web pages, check network requests, and monitor performance.

Database (Optional for full-stack development, e.g., MySQL, MongoDB):

- Purpose: To store and manage data for dynamic web applications.

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

Answer.

- **Web Server:**
  A web server is a software or hardware system that **hosts websites** and **delivers web content** (like HTML pages, images, CSS, JavaScript) to clients (browsers) over the internet using **HTTP/HTTPS protocols**.
  It handles requests from clients, processes them, and sends back the appropriate responses.
- **Commonly Used Web Servers:**
  1. **Apache HTTP Server** – Open-source, widely used.
  2. **Nginx** – High-performance, often used for load balancing and serving static content.
  3. **Microsoft IIS (Internet Information Services)** – Web server for Windows servers.
  4. **LiteSpeed** – Optimized for high traffic websites.
  5. **Caddy** – Modern web server with automatic HTTPS.

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

Answer.

Frontend Developer:

- Builds the **user interface** of the application that users interact with.
- Works with **HTML, CSS, JavaScript**, and frontend frameworks like React or Angular.
- Ensures the website is **responsive, visually appealing, and user-friendly**.

Backend Developer:

- Develops the **server-side logic** of the application.
- Works with **databases, APIs, server-side frameworks** (Node.js, Django, PHP).
- Handles **data processing, authentication, business logic, and server communication**.

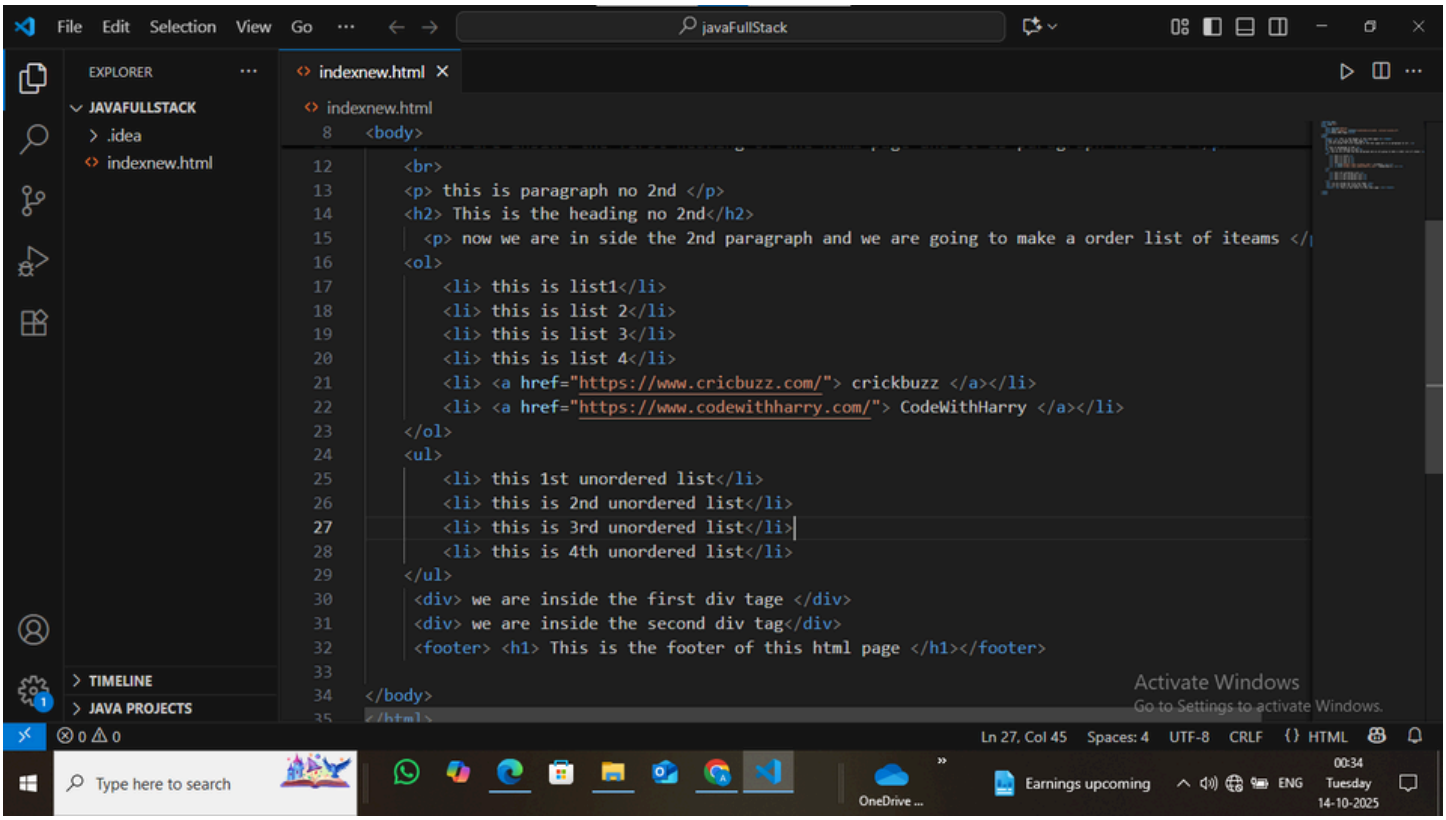
Database Administrator (DBA):

- Manages and maintains **databases** used in the project.
- Ensures **data integrity, security, backup, and performance optimization**.
- Works closely with backend developers to design and maintain efficient database structures.

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

Answer.

1. **Download and Install VS Code:**
   - Go to https://code.visualstudio.com/
   - Download the installer for your operating system (Windows/Mac/Linux).
   - Run the installer and complete the installation.
2. **Install Extensions for Web Development:**
   - Open VS Code → Click on **Extensions (Ctrl+Shift+X)**.
   - Install the following extensions:
     - **HTML CSS Support** – Provides autocomplete for HTML and CSS.
     - **JavaScript (ES6) code snippets** – For JavaScript shortcuts.
     - **Live Server** – Launch a local development server to preview HTML pages.
3. **Configure Settings (Optional):**
   - Enable **auto-save** and **format on save** for better workflow.
   - Customize themes or keyboard shortcuts as needed.
4. **Test the Setup:**
   - Create a new HTML file (index.html).
   - Write simple HTML/CSS/JS code.
   - Use **Live Server** to preview the webpage in the browser.
5. **Take a Screenshot:**
   - Capture the VS Code window showing your installed extensions and a sample HTML page.
   - Save it as VSCode-Setup.png and place it in your assignment folder.



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

Answer.

Static Websites:

- Content is fixed and does not change unless manually updated by the developer.
- Simple HTML, CSS, and sometimes JavaScript files.
- Faster to load and easier to host.
- **Example:** Personal portfolio website or a company's "About Us" page.

Dynamic Websites:

- Content is generated on-the-fly based on user interaction, server-side logic, or database queries.
- Requires server-side programming languages like PHP, Node.js, or Python and a database.
- **Example:** Facebook news feed, Amazon product pages, or an online banking portal.

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

Answer.

Five Popular Web Browsers and Their Rendering Engines:

Browser	Rendering Engine
Google Chrome	Blink
Mozilla Firefox	Gecko
Microsoft Edge	Blink
Safari	WebKit
Opera	Blink

Explanation:

- A **rendering engine** interprets HTML, CSS, and JavaScript to display web pages correctly in the browser.
- Different engines may **render the same page slightly differently**, affecting layout, style, and performance.
- **Blink** is used by Chrome, Edge, and Opera, so they often display pages similarly.
- **Gecko** (Firefox) and **WebKit** (Safari) may have slight differences in rendering behavior.

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

Answer.