1. Difference Between Frontend, Backend, and Full-Stack Development

Type	Description	Example
Frontend Development	Purpose Deals with the *visual part* of a website — what users see and interact with i.e User Interface. It involves HTML, CSS, and JavaScript as well as React, Angular	Example: The login form, buttons, navigation bar, and layout you see on Facebook. Responsible for UI Design,UX, Client Side logic
Backend Development	Handles *server-side logic*, databases, and APIs i.e Business logic. It processes user requests and sends responses to the frontend. Technologies include Node.js , Python, Java	Example: When you click “Login” on Facebook, the backend verifies your username/password from the database. Responsible for APIs, data processing, security
Full-Stack Development	Involves both frontend and backend skills. A full-stack developer builds complete web applications.	Example: A developer building both the chat interface (frontend) and chat storage system (backend) for WhatsApp Web.

---

2. Client-Server Model Diagram

You can draw this simple diagram:

+----------------+ +----------------+ +----------------+

| Client | | Internet | | Server |

| (Web Browser) |----->| (Network) |----->| (Website Host) |

+----------------+ +----------------+ +----------------+

| 1. User requests URL | |

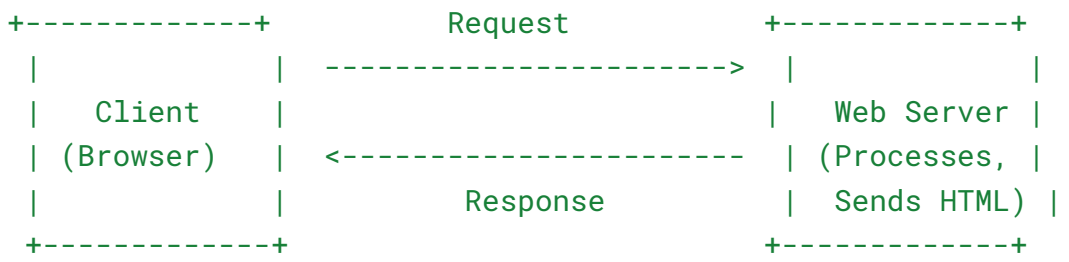
| 2. Browser sends HTTP req | |

| 3. Server sends |

| requested files |

| <---------- |

| 4. Browser renders website |



➞ **Explanation:**
The client (e.g., Chrome browser) sends an HTTP request to the server.
The server processes it and sends an HTML/CSS/JS response that the browser displays.

---

3. How a Browser Requests and Displays a Web Page

Steps:

1. User enters a URL in the browser (e.g., `www.example.com`).
2. The browser converts it into an **HTTP request** and sends it to the **web server**.
3. The **DNS** translates the domain name into the server's IP address.
4. The **server processes** the request and sends back an **HTML file** (and other resources like CSS/JS).
5. The **browser rendering engine** parses HTML, applies CSS, runs JS, and displays the webpage visually to the user.

---

4. Tools Required for Web Development Environment

Tool	Purpose

VS Code	Text editor for writing and managing HTML, CSS, JS, and backend code.
Web Browser (e.g., Chrome)	Used to run and test web pages.
Node.js	Allows JavaScript to run on the server (for backend development).
Git & GitHub	Version control system to save and share code.
Terminal / Command Prompt	For running commands like starting a local server or installing packages.
Live Server Extension (in VS Code)	Automatically refreshes the page when you edit code.

---

5. What is a Web Server?

A **web server** is software or hardware that serves web pages to users upon request.

Function: It receives HTTP requests from browsers and responds with web content (HTML, CSS, images, etc.).

Examples:

- **Apache HTTP Server** — Open-source and widely used.
- **Microsoft IIS** — Used in Windows-based servers.
- **Node.js** — JavaScript runtime that can act as a lightweight server.

---

6. Roles in a Web Project

Role	Responsibilities
Frontend Developer	Designs and develops the user interface using HTML, CSS, JavaScript, React, Angular etc. Convert designs from figma/adobe xd into code. Ensure cross-browser/device compatibility. Optimise performance for faster page loads
Backend Developer	Handles databases, server logic, and API development using Node.js, Python, Java, etc. Handle authentication, authorisation, and security.

	Connect frontend to databases and APIs. Optimise performance and scalability.
Database Administrator (DBA)	Manages, optimizes, and secures the database. Ensures data consistency and backups using SQL, MongoDB, REST APIs.

---

7. VS Code Installation & Configuration

Steps:

1. Download VS Code from https://code.visualstudio.com/
2. Install it.
3. Add the following extensions:
    - *Live Server* (for real-time preview)
    - *Prettier* (for code formatting)
    - *HTML CSS Support* and *JavaScript (ES6) Snippets*

Screenshot (for submission):
Take a screenshot of your VS Code showing:

- The code editor with `index.html`, `style.css`, and `script.js` open.
- Extensions panel visible.

Example caption:

> "VS Code configured for HTML, CSS, and JS development."

---

8. Static vs Dynamic Websites

Type	Description	Example
Static Website	Displays fixed content. No interaction with the database.	Example: A company's brochure website or portfolio (`index.html`).

Dynamic Website	Content changes based on user input or database data.	Example: Facebook, Amazon, YouTube.

---

9. Web Browsers and Rendering Engines

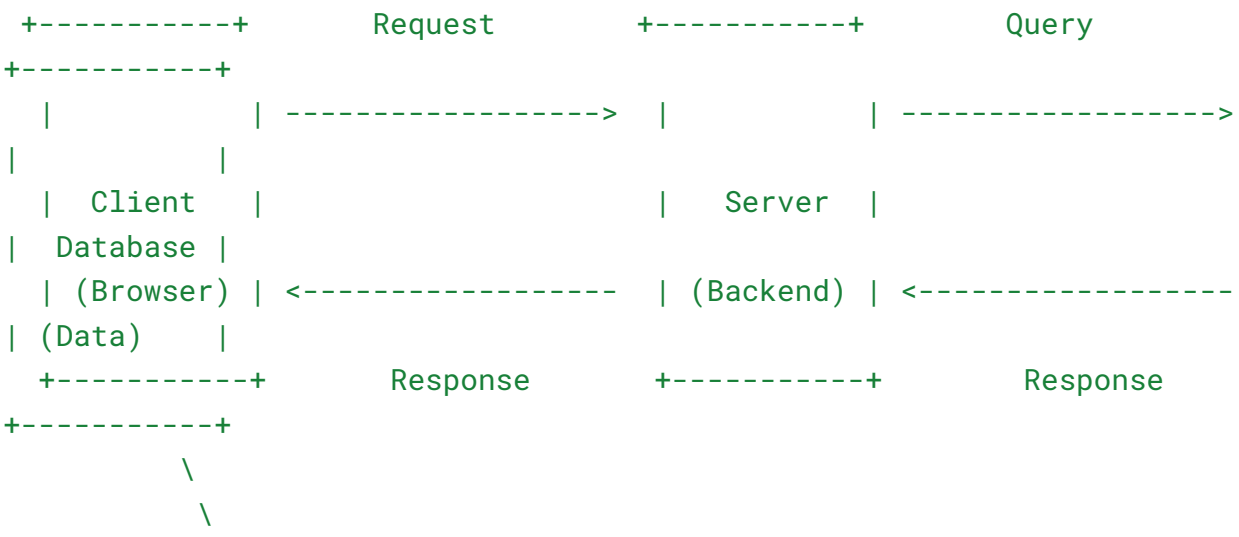
Browser	Rendering Engine	Description
Google Chrome	Blink	Fast and modern engine by Google.
Mozilla Firefox	Gecko	Open-source engine known for strong web standards.
Apple Safari	WebKit	Optimized for Apple devices.
Microsoft Edge	Blink (previously EdgeHTML)	Same engine as Chrome since 2020.
Opera	Blink	Uses same engine as Chrome, with additional built-in tools like VPN.

Rendering Engine:
Responsible for converting HTML, CSS, and JS into the visual webpage you see.

---

10. Basic Web Architecture Flow Diagram

Draw this labeled diagram:



```
\-->   External APIs (for payment, map, weather, etc.)
```

Explanation:

1. Client sends request (e.g., login) to the server.
2. Server interacts with the database and APIs to get data.
3. Response is sent back to the client and displayed in the browser.