Intro to Web Dev

1.

Frontend

The frontend is the visual, interactive part of a website or app that users see and interact with in their browser. Frontend developers focus on user interface (UI) and user experience (UX) to create responsive, engaging designs.

Example: The Amazon homepage—its product images, search bar, and "Add to Cart" buttons.

Backend

The backend is the behind-the-scenes part of an app. It handles servers, databases, and logic that users never see. Backend developers manage data storage, process requests, and maintain APIs to keep the app running smoothly and securely.

Example: On Amazon, the backend processes your payment, updates your cart, and checks inventory when you make a purchase.

Full-stack

Full-stack development means mastering both frontend and backend technologies. Full-stack developers can handle every aspect of a web app—from designing the interface to managing server logic and databases.

Example: A developer at a small startup who builds the app's interface, creates the server logic to process orders, and manages the database storing user information.

2.

3.

How Browsers Load Web Pages

When you visit a website, your browser follows these steps:

1. **DNS Lookup:** Translates the domain name into an IP address to locate the server.
2. **HTTP Request:** Sends a request to the server for the web page.

3. **Server Processing:** The server processes the request and retrieves necessary data.
4. **HTTP Response:** Server sends back HTML, CSS, JavaScript, and other assets.
5. **Rendering:** Browser parses HTML, applies styles, executes scripts, and loads resources.
6. **Display:** The page appears on your screen, ready for interaction.

4.

To set up a web development environment, you need a **text editor or IDE** (like VS Code) to write code efficiently, a **web browser** (Chrome, Firefox) to view and test pages, and **version control** (Git) to track changes. A **command line/terminal** is used to run scripts and manage files, while a **package manager** (npm, Yarn) handles project dependencies. A **local server or runtime** (Node.js, XAMPP) runs server-side code, and a **database** (MySQL, MongoDB) stores application data. **Browser developer tools** help debug and analyze web pages.

5.

A **web server** is software or hardware that delivers web pages to clients (browsers) over the internet using the **HTTP/HTTPS** protocol. When a browser requests a web page, the web server processes the request, retrieves the requested files (like HTML, CSS, JS), and sends them back to the browser for display.

Common examples of web servers:

- **Apache HTTP Server** – Open-source, widely used, highly customizable.
- **Nginx** – High-performance server, also used as a reverse proxy and load balancer.
- **Microsoft IIS** – Windows-based server for hosting websites and applications.
- **LiteSpeed** – Lightweight, fast server often used for high-traffic websites.

It basically acts as the bridge between users’ browsers and the website’s content

6.

In a web development project:

- A **frontend developer** designs and implements the **user interface** and experience using HTML, CSS, and JavaScript, ensuring the website is visually appealing and interactive.
- A **backend developer** builds and maintains the **server-side logic**, APIs, and application functionality, handling data processing, authentication, and integration with the frontend.
- A **database administrator (DBA)** manages the **database**, including designing schemas, optimizing queries, ensuring data security, integrity, and backups, and supporting efficient data access for the application.

7.

8.

A **static website** consists of fixed web pages with content that **does not change unless manually updated**. It is simple, fast, and usually built with HTML, CSS, and sometimes JavaScript.

Example: A personal portfolio site.

A **dynamic website** generates content **on-the-fly** based on user interaction or server-side logic, often using databases and server-side languages like PHP, Node.js, or Python.

Example: An e-commerce site like Amazon.

9.

Here are five popular **web browsers** and their rendering engines:

1. **Google Chrome** – Uses **Blink** engine, known for speed and supporting modern web standards.
2. **Mozilla Firefox** – Uses **Gecko** engine, emphasizes flexibility, security, and standards compliance.

3. **Microsoft Edge** – Uses **Blink** (Chromium-based), optimized for Windows integration and performance.
4. **Apple Safari** – Uses **WebKit**, designed for macOS/iOS with strong energy efficiency and smooth animations.
5. **Opera** – Uses **Blink**, similar to Chrome, but includes additional features like built-in VPN and ad blocker.

10.