

## 1. Main Components of Client-Server Architecture and Their Functions

Client-server architecture consists of two main components:

- **Client:** The client is a computer or device that requests services or resources from the server. It could be a web browser, mobile app, or desktop application. The client handles user interactions and sends requests to the server.
- **Server:** The server processes requests from clients, performs computations, accesses databases, and sends back the appropriate responses. It can be a web server (e.g., Apache, Nginx), an application server, or a database server.
- **Network:** The communication medium that connects clients and servers, typically using protocols like HTTP, HTTPS, FTP, or WebSocket.
- **Database:** Often part of the server-side, it stores data required for the application, managed by database management systems (DBMS) like MySQL, PostgreSQL, or MongoDB.

## 2. Career Paths After Completing a Web Development Course

A web development course can open various career opportunities, including:

- **Front-End Developer:** Specializes in UI/UX, working with HTML, CSS, JavaScript, and frameworks like React or Angular.
- **Back-End Developer:** Focuses on server-side logic, databases, and APIs using technologies like Node.js, Python, Java, or PHP.
- **Full-Stack Developer:** Works on both front-end and back-end, handling the entire development process.
- **Web Designer:** Emphasizes the visual aspects and user experience of a website using tools like Figma or Adobe XD.
- **DevOps Engineer:** Manages deployment, CI/CD, and server operations using tools like Docker, Kubernetes, and cloud platforms.
- **Software Engineer:** Expands beyond web development into broader software development.
- **Freelancer/Entrepreneur:** Builds and manages web projects independently or starts a web-based business.

## 3. Role of a Web Browser's Developer Tools in Web Development

Web browser developer tools (DevTools) provide a suite of functionalities to debug, inspect, and optimize web applications. Key roles include:

- **HTML & CSS Inspection:** Allows developers to view and modify the structure and styling of web pages in real-time.
- **JavaScript Debugging:** Helps identify errors and performance issues in scripts using breakpoints and console logs.

- **Network Monitoring:** Tracks HTTP requests, responses, and load times to optimize performance.
- **Performance Profiling:** Analyzes rendering speed, memory usage, and other performance metrics.
- **Security Auditing:** Identifies vulnerabilities such as mixed content and security headers.

## 4. Advantages of Using a Version Control System Like Git

Git provides numerous benefits for development projects:

- **Collaboration:** Enables multiple developers to work on the same project without conflicts.
- **Version Tracking:** Maintains a history of changes, allowing rollback to previous versions.
- **Branching & Merging:** Facilitates working on new features without affecting the main codebase.
- **Backup & Recovery:** Prevents data loss by storing changes in a distributed manner.
- **Open Source & Widely Used:** Supported by platforms like GitHub, GitLab, and Bitbucket for cloud-based repository hosting.

## 5. Comparison of a Text Editor and an IDE

Feature	Text Editor	IDE (Integrated Development Environment)
<b>Definition</b>	A lightweight tool for editing code with minimal features.	A complete environment for coding, debugging, and testing.
<b>Examples</b>	VS Code, Sublime Text, Notepad++	IntelliJ IDEA, Visual Studio, PyCharm
<b>Key Features</b>	Syntax highlighting, extensions, simple UI	Built-in debugger, compiler, version control, advanced tools
<b>Use Case</b>	Quick code edits, lightweight scripting	Full-scale application development, debugging, and testing