**Q1) Explain the main components of client-server architecture and their functions?**

**Client-Server Architecture:**

Client-server architecture is an architecture in which many clients (users or machines) request services from a single central server. It is being utilised in many web applications, database systems, e-mail services, etc.

**Major Components and Their Functions:**

1. **Client**

* **Definition:** Client refers to a computer or a program that initiates requests to a server.
* **Functions:**
  + It initiates communication with the server.
  + Sends requests for services or information.
  + Receiving and showing responses.
  + Delivers the user interface (i.e., web browsers, mobile apps).

1. **Server**

* **Definition:** Server refers to an extremely capable system that hosts services, information, or resources for clients.
* **Functions:**
  + Awaits for client requests and executes them.
  + Delivers desired content or performs operations.
  + It oversees security, sessions, and access management.
  + Works with various clients simultaneously.

1. **Network (Communication Medium)**

* **Definition:** It is the communication channel through which information is sent from client to server.
* **Functions:**
  + It facilitates communication with protocols such as TCP/IP.
  + Handles transfers of request and responses.
  + Comprises products such as routers, switches, modems, etc.

1. **Protocols**

* **Definition**: Rules governing formatting and processing of communication.
* **Functions:**
  + Manage how servers communicate with their clients.
  + Examples: **HTTP** (Internet webpage), **FTP** (computer files), \*\*
  + Achieve compatibility and effective communication.

1. **DNS (Domain Name System)**

* **Definition:** An internet system that translates domain names (such as www.google.com) to IP addresses.
* **Functions:**
  + Transforms user-friendly domain names to machine-readable IP addresses.
  + Assists the client in finding the appropriate server across the internet.
  + It serves as an address book or phonebook of the internet.
  + Eliminates the requirement to recall numerical IP addresses.

**Q2)Discuss the various career paths available after completing a web development course?**

**1. Front-End Developer**

* **Focus**: User interface (UI) and user experience (UX).
* **Skills**: HTML, CSS, JavaScript, frameworks like React, Angular, or Vue.js.
* **Role**: They design and implement what users see and interact with on a website. They ensure responsiveness, accessibility, and overall visual appeal.

**2. Back-End Developer**

* **Focus**: Server-side logic and databases.
* **Skills**: Programming languages like Java, Python, PHP, Node.js; databases like MySQL, MongoDB.
* **Role**: They handle data storage, retrieval, user authentication, and ensure smooth server operations.

**3. Full-Stack Developer**

* **Focus**: Both front-end and back-end development.
* **Skills**: Combination of front-end and back-end technologies.
* **Role**: They can build entire web applications independently or bridge the gap between front-end and back-end teams, making them versatile and in high demand.

**4. Web Designer**

* **Focus**: Visual design and layout.
* **Skills**: Graphic design tools (Adobe XD, Figma, Photoshop), basic HTML/CSS.
* **Role**: They create the look and feel of websites—designing layouts, color schemes, and typography—ensuring an aesthetically pleasing and user-friendly design.

**Q3)Describe the role of a web browser's developer tools in web development?**

**Ans) Role of Web Browser's Developer Tools in Web Development:**

Developer tools (DevTools) help web developers **inspect, debug, and optimize** websites in real-time. Key functions include:

* **Inspecting HTML/CSS**: Modify structure and styles instantly.
* **Debugging JavaScript**: Set breakpoints, view errors, and monitor variables.
* **Monitoring Performance**: Analyze load times and optimize rendering.
* **Tracking Network Activity**: Inspect API calls and resource loading.
* **Testing Responsiveness**: Simulate different devices and screen sizes.
* **Inspecting Storage**: View/edit local storage, cookies, and more.
* **Using Console**: Run JavaScript and view logs.

**Q4) What are the advantages of using a Version Control System like Git in a development project?**

**Ans) Git** in a development project:

* **Collaboration** – Multiple developers can work on the same project simultaneously.
* **History Tracking** – Keeps a complete history of changes made to the code.
* **Backup** – Acts as a backup by storing code in remote repositories like GitHub.
* **Branching** – Allows experimenting with new features without affecting the main code.
* **Revert Changes** – Easy to undo mistakes and revert to previous versions.
* **Conflict Resolution** – Helps manage and resolve code conflicts between team members.
* **Code Review** – Enables better code review and contribution tracking.

**Q5) Compare and contrast a text editor and an IDE, highlighting their key features and uses?**

**Ans)**

* **Definition**:
  + Text Editor: A lightweight tool for editing plain text or code.
  + IDE (Integrated Development Environment): A full-fledged development environment with many built-in tools.
* **Purpose**:
  + Text Editor: Writing and editing code or text files.
  + IDE: Writing, testing, debugging, and managing software projects.
* **Examples**:
  + Text Editor: VS Code, Sublime Text, Notepad++, Atom.
  + IDE: IntelliJ IDEA, Eclipse, NetBeans, Visual Studio.
* **Speed**:
  + Text Editor: Lightweight and fast.
  + IDE: Heavier and may consume more system resources.
* **Built-in Features**:
  + Text Editor: Basic syntax highlighting, code formatting.
  + IDE: Advanced features like debugging, version control, build tools.
* **Customization**:
  + Text Editor: Highly customizable via plugins/extensions.
  + IDE: Also customizable, but often comes with many features pre-installed.
* **Learning Curve**:
  + Text Editor: Easier to learn and use.
  + IDE: Steeper learning curve due to rich features.
* **Project Management**:
  + Text Editor: Minimal to none.
  + IDE: Strong project and file management capabilities.
* **Use Case**:
  + Text Editor: Small scripts, quick edits, lightweight coding.
  + IDE: Large-scale software development, team projects.