

Assignment – 1

Introduction to Web Development

1. Explain main components of client server architecture and their functions.

Ans: Introduction to client server architecture:

- A client is like your computer or smartphone that you use to access websites or apps.
- It sends requests to the server asking for information or services. For example, when you open a web browser and type in a website address, your computer (the client) asks the server to send the website data.
- A server is a powerful computer that stores data and applications, and it responds to requests from clients.
- It processes the requests from clients and sends back the required information or services. For instance, when you type in a website address, the server sends the website's pages back to your browser.
- A network is the connection between the client and server, like the internet or a local network in a school or office.

2. Discuss the various career paths available after completing a web development course.

Ans: Introduction to various career paths:

- Front End Web Developer:
 - a) What you'll do: Focus on the visual and interactive parts of websites and applications. You'll use languages like HTML, CSS, and JavaScript to build the user interface.
 - b) Skills needed: Strong design sense, good knowledge of user experience (UX), proficiency in front-end frameworks like React, Angular, or Vue.js.
- Backend Web Developer:
 - a) What you'll do: Work on the server side of web applications. You'll handle databases, server logic, and application integration.
 - b) Skills needed: Understanding of server-side languages like Node.js, Python, Ruby, or PHP, and knowledge of databases like MySQL, MongoDB, or PostgreSQL.

- **Full Stack Web Developer:**
 - a) **What you'll do:** Combine both front-end and back-end development. You'll handle everything from the server to the user interface.
 - b) **Skills needed:** Proficiency in both front-end and back-end technologies, understanding of how web applications work end-to-end.

3. Describe the role of a web browser's developer tools in web development.

Ans: Introduction to role of web browser's developer tools:

- Web browser developer tools, often called "dev tools," are essential for anyone involved in web development. They help developers inspect, debug, and analyze the HTML, CSS, and JavaScript code that makes up a web page. Here's a simple breakdown of their role:
- **Inspecting Elements:** Dev tools allow you to see the HTML structure and CSS styles of a webpage. You can click on any part of the page and see the code that defines it, which helps you understand how the page is built.
- **Debugging JavaScript:** When your JavaScript isn't working as expected, dev tools let you set breakpoints and step through your code line by line to find where things are going wrong.
- **Performance Analysis:** Dev tools can show you how fast your page loads and runs. You can see which parts of your code or which resources (like images or scripts) are slowing things down.
- **Network Monitoring:** You can check what resources are being loaded by the page, how long they take to load, and if there are any errors. This helps in optimizing the loading times and fixing issues related to resource loading.
- **Console:** The console is a powerful tool for logging information and running JavaScript code on the fly. It's useful for testing snippets of code and debugging.

4. What are the advantages of using a Version Control System like Git in a development project?

Ans: Introduction to Version Control System:

- A Git repository, also called as a repo is a directory where all your project files are stored. Git tracks changes to these files, giving you complete control over your project development's lifecycle.
- Each commit is a snapshot of your repository at a specific time, with a unique ID that allows you to revisit it later if needed.
- It allows you to collaborate with others while keeping your work separate by creating branches to test new features or bug fixes.
- In Git, branching allows you to work on separate parts of your project's codebase without affecting its main code. Once you are ready to integrate, you can merge your branch back into the main "master" branch.
- In Git, switch to the new branch with the git branch command.
- After creating the branch, make the changes needed for the feature or fix. Commit your work with git commit.
- Git allows for a collaborative development approach using pull requests. A pull request notifies a team member that feature is ready for review and merging.

5. Compare and contrast a text editor and an IDE, highlighting their key features and uses.

Ans: Introduction to text editor and an IDE:

- A text editor is a program that lets you write and edit plain text. Basic text editors are straightforward and easy to use. Examples include Notepad on Windows or TextEdit on MacOS.
- They use minimal system resources, which means they run quickly and don't slow down your computer.
- Many text editors, like Sublime Text or Visual Studio Code, offer plugins or extensions to add extra features.
- An IDE is a software suite that provides comprehensive tools for software development. Just like a text editor, but usually with more

advanced features like syntax highlighting, code completion, and error checking.

- It helps you find and fix errors in your code by allowing you to run your code step-by-step and inspect variables.
- Integration with version control systems like Git to manage changes to your code over time.
- It provides tools to organize and manage files and resources in larger projects.