**Lab 01**

Objective: To study web hosting, web servers, frameworks and web technologies.

Web Hosting :

Web hosting is a service that allows individuals and organizations to make their websites accessible on the internet. It involves storing the files, data, and content that make up a website on special computers called servers. These servers are configured to serve the website's content to users who access it through their web browsers.

Populating Platforms for Web Hosting:

AWS (Amazon Web Services): AWS offers a wide range of cloud services, including web hosting options like Amazon Lightsail and Amazon EC2. It's suitable for businesses and developers looking for high scalability.

Google Cloud Platform: Google Cloud offers flexible cloud hosting services, including Google Compute Engine and App Engine, suitable for various applications and projects.

Microsoft Azure: Azure offers a wide range of cloud services, including virtual machines and web apps, making it suitable for businesses using Microsoft technologies.

What is a Web Server?

A web server is a specialized computer or software system that stores, processes, and serves website content to users who request it via their web browsers. It plays a crucial role in delivering web pages, files, and other resources to visitors across the internet. Web servers are the backbone of how the internet functions, as they enable websites to be accessible to users worldwide.

How does it work?

Request from a User: When a user types a website's URL (domain name) into their web browser or clicks on a link, their browser sends a request to the corresponding web server.

Processing the Request: The web server receives the request and interprets it. It figures out which file or resource the user is asking for based on the URL.

Retrieving Resources: If the requested content is a static file (like an image, HTML file, CSS file, etc.), the web server retrieves that file from its storage. If the content is dynamic (like generated web pages from a database), the server processes the necessary data to create the response.

Generating the Response: The web server generates an HTTP response, which includes the requested content and relevant metadata (like headers) in a structured format.

Sending the Response: The web server sends the HTTP response back to the user's browser over the internet.

Rendering in the Browser: The user's browser receives the response and processes the content. It interprets HTML, applies CSS styles, runs JavaScript (if included), and displays the final web page on the user's screen.

What is a Framework?

Frontend and backend frameworks are tools that developers use to build different parts of a web application. Frontend frameworks focus on building the user interface and the user experience, while backend frameworks are used for developing the server-side logic and managing databases and server operations.

Popular Frontend Frameworks:

React: React, developed by Facebook, is a widely used JavaScript library for building user interfaces. It's known for its component-based architecture, which allows developers to create reusable UI components. React makes it easier to manage the state of your application and update the user interface efficiently.

Angular: Angular, maintained by Google, is a full-fledged JavaScript framework for building dynamic web applications. It provides a robust structure for creating complex applications by offering features like two-way data binding, dependency injection, and a modular architecture.

Popular Backend Frameworks:

Express.js: Express.js is a lightweight and flexible backend framework for Node.js. It simplifies building APIs and handling HTTP requests by providing a minimalistic structure. Express is often used to create fast and scalable web applications.

Django: Django is a high-level Python web framework that emphasizes efficiency and developer productivity. It includes an ORM (Object-Relational Mapping) for managing databases, a robust authentication system, and an admin interface, making it suitable for complex web applications.

What are the different web technologies used to create a complete website?

Client-Side (Frontend) Technologies:

HTML (Hypertext Markup Language): HTML is the fundamental building block of the web. It structures content using elements like headings, paragraphs, lists, links, and more.

CSS (Cascading Style Sheets): CSS is used for styling HTML elements, controlling layout, colors, fonts, and visual aspects of a webpage.

JavaScript: JavaScript is a programming language that enables interactivity and dynamic behavior on web pages. It's used to manipulate the DOM (Document Object Model) and create responsive user interfaces.

Frontend Frameworks: Frameworks like React, Angular, and Vue.js help structure and streamline the development of frontend components and user interfaces.

Responsive Design: Techniques like media queries and flexible grids help create websites that adapt and look good on various devices and screen sizes.

Server-Side (Backend) Technologies:

Server-Side Scripting Languages: Languages like Node.js (JavaScript), Python (Django, Flask), Ruby (Ruby on Rails), PHP (Laravel), and Java (Spring Boot) are used to create server-side logic and handle requests.

Databases: Technologies like MySQL, PostgreSQL, MongoDB, and SQLite are used to store, manage, and retrieve data for web applications.

Server Frameworks: Frameworks provide a structured way to build server-side applications. Examples include Express.js (Node.js), Django (Python), Ruby on Rails (Ruby), and Laravel (PHP).

APIs (Application Programming Interfaces): APIs allow different software components to communicate with each other. They're used to fetch data or perform actions from external services.

Authentication and Authorization: Technologies like OAuth, JWT (JSON Web Tokens), and sessions are used to secure and manage user authentication and access.