**1. Introduction to Web Development and MERN:**

• Overview of the MERN stack and its components

Web development is the process of creating and maintaining websites and web applications. It involves a combination of front-end development (working on the user interface and user experience) and back-end development (handling server-side logic and databases). One popular technology stack used for web development is the MERN stack, which stands for MongoDB, Express.js, React, and Node.js. Each component of the stack plays a specific role in building web applications:

**MongoDB**: A NoSQL database that stores data in a flexible, JSON-like format. MongoDB is known for its scalability and is often used to store and manage the application's data.

**Express**.**js**: A web application framework for Node.js that simplifies building robust and scalable web applications on the server side. It provides tools and middleware for handling routes, requests, and responses.

**React**: A JavaScript library for building user interfaces. React is used on the client side to create dynamic and interactive web pages. It allows developers to build reusable UI components.

**Node**.**js:** A JavaScript runtime environment that allows you to execute JavaScript code on the server side. Node.js is used to build the server, handle HTTP requests, and interact with the database.

**Setting up a Development Environment:**

Before you can start developing web applications with the MERN stack, you need to set up your development environment. Here's a basic guide to getting started:

**Install Node.js and npm:**

Visit the official Node.js website (https://nodejs.org/).

Download and install the appropriate version of Node.js for your operating system.

npm (Node Package Manager) comes bundled with Node.js, so you don't need to install it separately.

Verify Installation:

Open your terminal or command prompt.

Run the following commands to check if Node.js and npm are installed and to see their versions:

node -v

npm -v

Creating a Simple "Hello World" Application using Node.js and Express.js:

Now that you have Node.js and npm installed, you can create a simple "Hello World" application using Node.js and Express.js. Here are the steps:

1. Create a new directory for your project and navigate to it in the terminal:

mkdir hello-world-app

cd hello-world-app

1. Initialize a Node.js project by running the following command and following the prompts:

npm init

1. Install Express.js as a dependency for your project:

npm install express --save

1. Create a JavaScript file (e.g., app.js) in your project directory and add the following code to create a basic Express.js server that responds with "Hello, World!" when accessed:

const express = require('express');

const app = express();

const port = 3000;

app.get('/', (req, res) => { res.send('Hello, World!'); }); app.listen(port, () => { console.log(`Server is running on http://localhost:${port}`);

});

Start the Express.js server by running the following command in your project directory:

node app.js

Open your web browser and navigate to http://localhost:3000. You should see "Hello, World!" displayed on the webpage like below image.

