MERN

**Module 1:**

**Introduction to Web Development and the MEAN Stack**

# **Overview of web development**

Web development: what is it?

Web development is the process of designing and maintaining websites or online applications.

Web development falls into one of two categories::

**1.Frontend Development.**

**2.Backend Development.**

**1.Frontend Development**

refers to the area of web development where the user interface is the main focus.Client-side development is another name for front-end development.

**Frontend Roadmap**

**https://media.geeksforgeeks.org/wp-content/cdn-uploads/20220825163009/Front-End-Frameworks-and-Libraries1.png**

**2.Backend Development**

The area of web development that works with the server, databases, and application logic is referred to as backend development, often called server-side development.

### **Backend Roadmap**

https://media.geeksforgeeks.org/wp-content/cdn-uploads/20210309162115/Backend-design-roadmap.jpeg

**Introduction to the MEAN stack**

One of the most widely used web development technologies is MEAN Stack

MEAN Stands for

1. **M** – MongoDB
2. **E** – Express
3. **A** – Angular
4. **N** – Node.js

**1.MongoDB**

a) A NoSQL (Not Just SQL) database is MongoDB.Because MongoDB lacks schemas, it provides flexible data storage without requiring a set structure.

b) It uses a document-oriented approach and stores information in collections in BSON format.

c) The database supports sharding, which distributes data among several servers or clusters, and is built for horizontal scalability.

d) Indexing is included in MongoDB's comprehensive query language, which supports a variety of query types and enhances query efficiency.

**2.Express**

a) Web Application Framework: The Express.js framework for Node.js allows building web applications.

b) Support for Middleware: It allows features like error handling and authentication by supporting middleware functions for jobs that take place within the request-response cycle.

c) Routing: Express makes handling various HTTP methods for certain URLs easier by streamlining the route design process.

d)Template Engines: Express is not limited to any one template engine; it may be used with EJS or Pug, for example, to generate dynamic HTML.

e) RESTful APIs: Express is a popular tool for creating RESTful APIs, enabling programmers to effectively handle HTTP methods, design routes, and control data flow.

**3. Angular**

a) Front-end JavaScript Framework: Google created and maintains the Angular front-end JavaScript framework.

b) Single-Page Application (SPA): It is utilized in the development of dynamic, one-page websites (SPAs).

c) Two-Way Data Binding: Two-way data binding is a feature of Angular that enables automated synchronization between the display and the model.

d)Injecting Dependency: Because of its support for dependency injection, dependencies may be efficiently managed and injected across an application.

e) Components and Modularity: Angular promotes modularity by utilizing components, which are self-contained functional entities that aid in code structure and reuse.

**4.Node.js**

a) JavaScript Runtime: Based on the V8 JavaScript engine, Node.js is a JavaScript runtime.

b)Server-Side JavaScript: This feature enables the creation of scalable and highly performant server-side applications by allowing developers to run JavaScript on the server side.

c) Non-blocking and event-driven I/O: Because Node.js is event-driven and non-blocking, it works well with asynchronous operations and concurrent connections.

d) NPM (Node Package Manager): NPM, which makes it easier to install and manage third-party libraries and tools, is included with Node.js.

d) Flexibility: Web servers, APIs, command-line tools, and real-time apps are just a few of the many kinds of applications that can be built using Node.js due to its versatility and widespread usage.

Setting up the development environment

**1. Set up npm and Node.js:**

Go to the official Node.js website to download and install the framework..

**Link to download :** [**https://nodejs.org/en**](https://nodejs.org/en)

Verify the installation by running the following commands in your terminal or command prompt:

**node -v**

**npm -v**

**2. Install MongoDB:**

Go to the official MongoDB website to download and install MongoDB.

The download may be found at: <https://www.mongodb.com/try/download/community>

Observe the operating system's installation guidelines.

**3. Install Express.js:**

For your project, make a new directory and use your terminal or command prompt to browse to it. Execute the subsequent commands:

**npm init -y**

**npm install express --save**

**4. Create an Express.js Server**:

Make a file (app.js, for example) to configure your Express.js server. With a text editor open, add the following code to the file:

javascript code

**const express = require('express');**

**const app = express();**

**const port = 3000;**

**app.get('/', (req, res) => {**

**res.send('Hello MEAN Stack!');**

**});**

**app.listen(port, () => {**

**console.log(`Server running at http://localhost:${port}`);**

**});**

**5. Install Angular CLI:**

Use this command to install Angular CLI globally:

**npm install -g @angular/cli**

**6. Create Angular App:**

Go to the directory in which you wish to build your Angular application. Execute the subsequent commands:

**ng new my-mean-app**

**cd my-mean-app**

Follow the instructions to configure your Angular application.

**7. Integrate Angular with Express:**

Modify your Express.js server to serve the Angular app. Open the app.js file and add the following code:

javascript code

**const express = require('express');**

**const app = express();**

**const path = require('path');**

**const port = 3000;**

**// Serve the Angular app**

**app.use(express.static(path.join(\_\_dirname, 'my-mean-app/dist/my-mean-app')));**

**// Handle all other routes**

**app.get('\*', (req, res) => {**

**res.sendFile(path.join(\_\_dirname, 'my-mean-app/dist/my-mean-app/index.html'));**

**});**

**app.listen(port, () => {**

**console.log(`Server running at http://localhost:${port}`);**

**});**

**8. Run Your MEAN Stack App:**

Launch the server for Express.js:

Node.js application

To view your MEAN Stack app in action, open your web browser and navigate to http://localhost:3000.