The MERN stack is a popular set of technologies used to build full-stack web applications. It stands for:

- **M**ongoDB – NoSQL database
- **E**xpress.js – Web application framework for Node.js
- **R**eact.js – Front-end JavaScript library
- **N**ode.js – JavaScript runtime environment (used for backend)

These tools together allow you to write **entire applications (front-end and back-end)** using **JavaScript**.

□ Why Learn the MERN Stack?

- All components use JavaScript (one language for both client and server).
- Huge community and resources.
- In-demand for modern web development.
- Excellent for building scalable, real-time applications.

□ How to Start Learning the MERN Stack (Step-by-Step for Beginners)

□ Step 1: Learn JavaScript (Core Language)

Before diving into the stack, you must be comfortable with **JavaScript**.

- Concepts to cover: variables, functions, arrays, objects, promises, async/await, ES6+ syntax.

Resources:

- JavaScript.info
- FreeCodeCamp's JavaScript course
- YouTube: The Net Ninja or Programming with Mosh

□ Step 2: Learn Frontend – React.js

React is used to build the UI (User Interface).

What to Learn:

- JSX
- Components (functional and class)
- Props and State
- useEffect and useState hooks
- Routing (React Router)
- Conditional rendering
- Forms and events

Resources:

- React.dev
- YouTube: "React Tutorial for Beginners" by Net Ninja or Traversy Media

---

□ Step 3: Learn Backend – Node.js + Express.js

This part handles APIs, routing, and server-side logic.

What to Learn:

- Node.js basics (npm, modules, creating a server)
- Express.js (routing, middleware, REST API)
- How to connect frontend to backend

Resources:

- Node.js Docs
- Express Docs
- YouTube: Net Ninja Node.js crash course

□ Step 4: Learn Database – MongoDB

MongoDB stores your app data (e.g., user accounts, posts, messages, etc.).

What to Learn:

- MongoDB basics (documents, collections)
- CRUD operations (Create, Read, Update, Delete)
- Mongoose (ODM library for MongoDB in Node.js)
- Connecting MongoDB with Node.js

Resources:

- MongoDB University
- YouTube: "MongoDB Crash Course" by Traversy Media

□ Step 5: Build Projects!

Learning is best with practice. Try building:

- Todo App
- Blog App
- Notes App
- Chat App
- E-commerce site

□ Tools You'll Use

Tool	Purpose
VS Code	Code editor
Postman	Test backend APIs
Git + GitHub	Version control
MongoDB Atlas	Cloud MongoDB database
Vite / CRA	React app setup tools

Here's a step-by-step guide **to build an Express app that displays** 5–6 static pages using `.ejs`:-

1. Prerequisites

Make sure you have Node.js installed.

□ Install Express and EJS:

1)Open your command prompt in windows and type following commands:-

```
D:\>mkdir express-project1

D:\>cd express-project1
```

2) npm init -y

```
D:\express-project1>npm init -y
Wrote to D:\express-project1\package.json:

{
  "name": "express-project1",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
```

3) npm install express ejs

```
D:\express-project1>npm install express ejs
```

4)Here's a **step-by-step guide** to build an Express app that displays **5–6 static pages using `.ejs`**.

□ Goal

Serve pages like:

- Home
- About
- Services
- Contact
- Portfolio
- FAQ

Using **Express.js + EJS (Embedded JavaScript Templates)**.

□ 1. Prerequisites

Make sure you have Node.js installed.

□ Install Express and EJS:

```
npm init -y
npm install express ejs
```

□ 2. Project Structure

```
express-project1/
├── views/
│   ├── home.ejs
│   ├── about.ejs
│   ├── services.ejs
│   ├── contact.ejs
│   ├── portfolio.ejs
│   └── faq.ejs
├── public/
│   └── css/
│       └── style.css
├── app.js
```

□ 3. `app.js` (Main Server File)

```
const express = require('express');
const app = express();
const PORT = 3000;

// Set view engine
app.set('view engine', 'ejs');

// Serve static files like CSS
app.use(express.static('public'));

// Routes
app.get('/', (req, res) => {
    res.render('home');
});

app.get('/about', (req, res) => {
    res.render('about');
});

app.get('/services', (req, res) => {
    res.render('services');
});

app.get('/contact', (req, res) => {
    res.render('contact');
});

app.get('/portfolio', (req, res) => {
    res.render('portfolio');
});

app.get('/faq', (req, res) => {
    res.render('faq');
```

```
});

app.use((req, res) => { res.status(404).render('404'); });

app.listen(PORT, () => {
    console.log(`Server is running at http://localhost:${PORT}`);
});
```

□ 4. Sample EJS Pages

Example: `views/home.ejs`

```
<!DOCTYPE html>
<html>
<head>
    <title>Home</title>
    <link rel="stylesheet" href="/css/style.css">
</head>
<body>
    <h1>Welcome to the Home Page</h1>
    <a href="/about">About</a> |
    <a href="/services">Services</a> |
    <a href="/contact">Contact</a> |
    <a href="/portfolio">Portfolio</a> |
    <a href="/faq">FAQ</a>
</body>
</html>
```

You can copy and modify this for the other pages (`about.ejs`, `services.ejs`, etc.) with their respective content.

views/404.ejs :-

<!DOCTYPE html>

<html>

<head>

<title>Page Not Found</title>

<link rel="stylesheet" href="/css/style.css">

</head>

<body>

<%- include('partials/nav') %>

<h1>404 - Page Not Found</h1>

Sorry, the page you are looking for does not exist.

Go back to Home

</body>

</html>

□ 5. Optional CSS (public/css/style.css)

```
body {
    font-family: Arial, sans-serif;
    background: #f4f4f4;
    margin: 40px;
    color: #333;
}
a {
    margin-right: 10px;
}
```

□ 2. about.ejs

```
<!DOCTYPE html>
<html>
<head>
    <title>About Us</title>
    <link rel="stylesheet" href="/css/style.css">
</head>
<body>
    <%- include('partials/nav') %>

    <h1>About Us</h1>
    <p>We are a team of developers passionate about web development.</p>
</body>
</html>
```

□ 3. services.ejs

```
<!DOCTYPE html>
<html>
<head>
    <title>Our Services</title>
    <link rel="stylesheet" href="/css/style.css">
</head>
<body>
    <%- include('partials/nav') %>

    <h1>Our Services</h1>
    <ul>
        <li>Web Development</li>
        <li>Mobile App Development</li>
        <li>UI/UX Design</li>
        <li>SEO Optimization</li>
    </ul>
    <%- include('partials/nav') %>
</body>
</html>
```

□ 4. contact.ejs

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Us</title>
    <link rel="stylesheet" href="/css/style.css">
</head>
<body>
    <%- include('partials/nav') %>

    <h1>Contact Us</h1>
    <p>Feel free to reach out to us via email or phone.</p>
    <ul>
        <li>Email: contact@example.com</li>
        <li>Phone: +1 123 456 7890</li>
    </ul>
    <%- include('partials/nav') %>
</body>
</html>
```

□ 5. portfolio.ejs

```
<!DOCTYPE html>
<html>
<head>
    <title>Our Portfolio</title>
    <link rel="stylesheet" href="/css/style.css">
```

```
</head>
<body>
    <%- include('partials/nav') %>

    <h1>Our Portfolio</h1>
    <p>Here are some of our recent projects:</p>
    <ul>
        <li>Project Alpha</li>
        <li>Project Beta</li>
        <li>Project Gamma</li>
    </ul>
    <%- include('partials/nav') %>
</body>
</html>
```

□ 6. `faq.ejs`

```
<!DOCTYPE html>
<html>
<head>
    <title>FAQ</title>
    <link rel="stylesheet" href="/css/style.css">
</head>
<body>
    <%- include('partials/nav') %>

    <h1>Frequently Asked Questions</h1>
    <ul>
        <li><strong>Q:</strong> What services do you
offer?<br><strong>A:</strong> Web, Mobile, UI/UX, SEO</li>
        <li><strong>Q:</strong> How can I contact you?<br><strong>A:</strong>
Via email or phone on the contact page.</li>
        <li><strong>Q:</strong> Where are you located?<br><strong>A:</strong>
We work remotely around the world.</li>
    </ul>
    <%- include('partials/nav') %>
</body>
</html>
```

□ `views/partials/nav.ejs` – Reusable Nav Bar

Create a `partials` folder inside `views/`, then create `nav.ejs` for navigation:

```
<nav>
    <hr>
```

```
    <a href="/">Home</a> |
    <a href="/about">About</a> |
    <a href="/services">Services</a> |
    <a href="/contact">Contact</a> |
    <a href="/portfolio">Portfolio</a> |
    <a href="/faq">FAQ</a>
    <hr>
</nav>
```

In each page, we use `<%- include('partials/nav') %>` to insert the nav bar.

□ Basic CSS (Optional)

Add to `public/css/style.css`:

```
body {
    font-family: Arial, sans-serif;
    padding: 30px;
    background-color: #f9f9f9;
    color: #333;
}

nav a {
    text-decoration: none;
    margin-right: 15px;
    color: #007bff;
}

nav a:hover {
    text-decoration: underline;
}
```

□ Done!

Start your server:

```
node app.js
```

Then visit:

- `http://localhost:3000/`
- `/about`

- `/services`
- etc.