

MERN Stack Installation & Package Setup

1. Frontend Setup (React + Vite + Tailwind)

☒ Step 1: Create React Project using Vite

```
npm create vite@latest
```

- Follow the prompts:
 - Project Name: **your-project-name**
 - Select Framework: **React**
 - Select Variant: **JavaScript** or **TypeScript** (as per your choice)

☒ Step 2: Install Core Dependencies

```
cd your-project-name  
npm install
```

☒ Step 3: Install Tailwind CSS with Vite

```
npm install tailwindcss @tailwindcss/vite
```

☒ Step 4: Update **vite.config.js**

```
import { defineConfig } from 'vite';  
import tailwindcss from '@tailwindcss/vite';  
  
export default defineConfig({  
  plugins: [tailwindcss()],  
});
```

☒ Step 5: Setup Tailwind in CSS

Create or update **src/index.css** or **src/app.css** file:

```
@import "tailwindcss";
```

Make sure this CSS is imported in **main.jsx**:

```
import './index.css'; // or './app.css'
```

☒ Step 6: Run the Project

```
npm run dev
```

☒ Step 7: Install Other Useful Frontend Packages

```
npm install axios react-router-dom react-icons react-redux redux-persist  
@reduxjs/toolkit
```

2. Backend Setup (Node.js + Express + MongoDB)

☒ Step 1: Create Backend Folder & Init

```
mkdir backend  
cd backend  
npm init -y
```

☒ Step 2: Install Backend Packages

```
npm install express mongoose dotenv bcrypt jsonwebtoken nodemailer cors cookie-  
parser
```

☒ Step 3: Install Dev Dependency

```
npm install --save-dev nodemon
```

☒ Step 4: Basic `package.json` Script Setup

Update your `package.json` file:

```
"scripts": {  
  "dev": "nodemon index.js",  
  "start": "node index.js",  
  "test": "echo \"Error: no test specified\" && exit 1"  
}
```

☑ Step 5: Run Backend Server

```
npm run dev
```

📄 Summary of Packages

◇ Frontend

Package	Purpose
vite	Faster React project scaffolding
tailwindcss	Utility-first CSS framework
axios	API calls
react-router-dom	Frontend routing
react-icons	Icon pack
react-redux	State management
redux-persist	Redux state persistence in storage
@reduxjs/toolkit	Simplified Redux setup

◇ Backend

Package	Purpose
express	Web framework for Node.js
mongoose	MongoDB ORM (object data modeling)
dotenv	Manage environment variables
bcrypt	Hashing passwords
jsonwebtoken	Token-based authentication
nodemailer	Sending emails
cors	Cross-Origin Resource Sharing
cookie-parser	Handle HTTP cookies
nodemon (dev)	Auto-restart server during development