Tech Stack

- Frontend: React.js

- Backend: Node.js with Express.js

- Database: MongoDB

- Styling: Tailwind CSS or Bootstrap

Project Structure

```
mern-portfolio/
backend/
models/
routes/
server.js
frontend/
public/
src/
components/
pages/
App.js
```

Initialize Backend

index.js

README.md

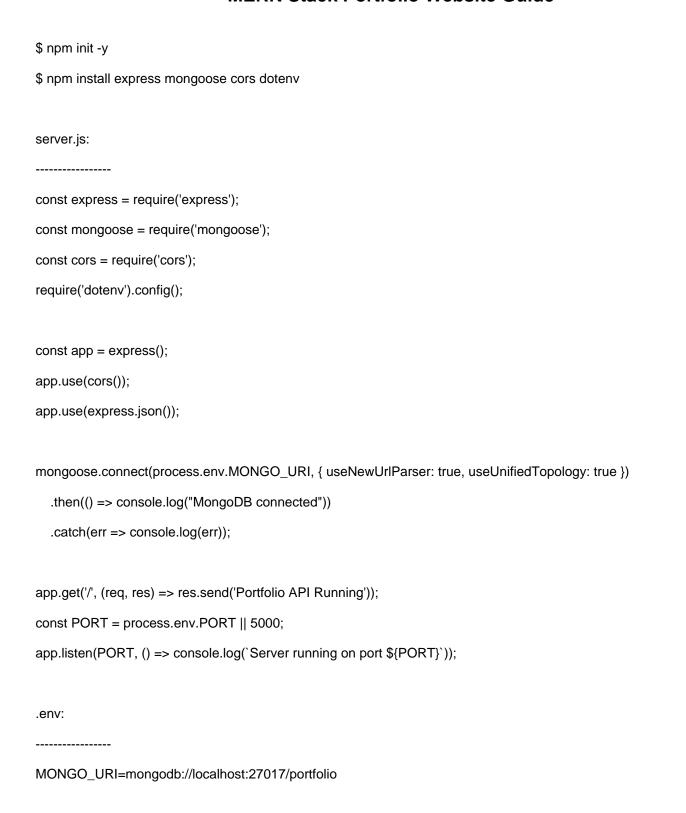
.env

\$ mkdir mern-portfolio

\$ cd mern-portfolio

\$ mkdir backend

\$ cd backend



Create Frontend with React

\$ npx create-react-app frontend

\$ cd frontend

\$ npm install axios react-router-dom

Basic Pages & Routing

```
App.js:
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './pages/Home';
import Projects from './pages/Projects';
import Contact from './pages/Contact';
function App() {
 return (
  <Router>
   <Routes>
     <Route path="/" element={<Home />} />
     <Route path="/projects" element={<Projects />} />
     <Route path="/contact" element={<Contact />} />
   </Routes>
  </Router>
 );
}
export default App;
Home.js:
export default function Home() {
 return (
  <div>
```

```
<h1>Hello, I'm Sarvesh</h1>
MERN Stack Developer
</div>
);
}
```

MongoDB Integration (Project Section)

```
models/Project.js:
const mongoose = require('mongoose');
const ProjectSchema = new mongoose.Schema({
 title: String,
 description: String,
 techStack: [String],
 liveLink: String,
 githubLink: String
});
module.exports = mongoose.model('Project', ProjectSchema);
routes/projectRoutes.js:
const express = require('express');
const router = express.Router();
const Project = require('../models/Project');
router.get('/', async (req, res) => {
 const projects = await Project.find();
 res.json(projects);
});
```

```
module.exports = router;
Mount in server.js:
const projectRoutes = require('./routes/projectRoutes');
app.use('/api/projects', projectRoutes);
Projects.js:
import { useEffect, useState } from 'react';
import axios from 'axios';
export default function Projects() {
 const [projects, setProjects] = useState([]);
 useEffect(() => {
  axios.get('http://localhost:5000/api/projects')
   .then(res => setProjects(res.data));
 }, []);
 return (
  <div>
   <h2>My Projects</h2>
   ul>
     {projects.map((project) => (
      <h3>{project.title}</h3>
       {project.description}
       <a href={project.githubLink}>GitHub</a> | <a href={project.liveLink}>Live</a>
```

```
))}
</div>
);
```

Run the App

Backend:

\$ cd backend

\$ node server.js

Frontend:

\$ cd frontend

\$ npm start

Optional Features

- Contact form (Nodemailer / save to DB)
- Admin login with JWT Auth
- Resume download as PDF
- Dark/Light mode toggle