

Full Stack Developer with AI - Complete Project Setup

Prerequisites Check

Before starting, ensure you have installed:

- Visual Studio Code
- Node.js (latest LTS version)
- Git
- A modern web browser (Chrome/Firefox)

Project Structure Overview

```
ai-lms-project/  
├── client/           # React.js frontend  
├── module3-html-css/ # HTML/CSS/Bootstrap module  
├── module5-javascript/ # JavaScript module  
└── README.md
```

Step 1: Create Main Project Directory

1. Open **Visual Studio Code**
2. Open Terminal (Ctrl + ~ or View → Terminal)
3. Create your main project folder:

```
bash  
  
mkdir ai-lms-project  
cd ai-lms-project
```

Step 2: Set Up HTML/CSS/Bootstrap Module (Module 3-4)

1. Create HTML/CSS module folder:

```
bash  
  
mkdir module3-html-css  
cd module3-html-css
```

2. Create **index.html** file:

```
bash
```

3. **Add the starter code** (copy from your document):

- Paste the provided HTML code with Bootstrap CDN
- The code includes navigation bar, flexbox layouts, and Bootstrap components

4. **Complete the required tasks:**

- Add two more feature boxes: "Progress Tracking" and "Real-time Assessments"
- Add Bootstrap cards section with "HTML Module" and "CSS Module"

5. **Test the page:**

- Right-click on index.html → "Open with Live Server" (install Live Server extension if needed)
- Or simply double-click the file to open in browser

Step 3: Set Up JavaScript Module (Module 5)

1. **Create JavaScript module folder:**

```
bash
cd ..
mkdir module5-javascript
cd module5-javascript
```

2. **Create Javascript.html file:**

```
bash
code Javascript.html
```

3. **Add the starter code** (copy from your document):

- Includes button interactivity, DOM manipulation, and form validation

4. **Complete the required tasks:**

- Finish the `validateEmail()` function
- Add keypress event handling for the goal input

5. **Test the JavaScript functionality:**

- Open Javascript.html in browser
- Test all interactive features

Step 4: Set Up React.js Frontend (Main Application)

1. Navigate back to main project directory:

```
bash  
cd ..
```

2. Create React application:

```
bash  
npx create-react-app client  
cd client
```

3. Install React Router (for navigation):

```
bash  
npm install react-router-dom
```

4. Set up the project structure:

```
bash  
mkdir src/pages src/components src/assets
```

5. Replace src/App.js with the provided code:

```
javascript
```

```

import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './pages/Home';
import Login from './pages/Login';
import Courses from './pages/Courses';
import Chatbot from './pages/Chatbot';

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/login" element={<Login />} />
        <Route path="/courses" element={<Courses />} />
        <Route path="/chatbot" element={<Chatbot />} />
      </Routes>
    </Router>
  );
}
export default App;

```

6. Update src/index.js:

```

javascript

import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);

```

Step 5: Create React Components

1. Create src/components/LMSComponents.js:

- Add the Login component
- Add the CourseRecommender component
- Copy the provided code from your document

2. Create the required pages:

src/pages/Home.js:

javascript

```
import React from 'react';

function Home() {
  return (
    <div style={{ padding: '20px' }}>
      <h1>Welcome to AI-Powered LMS</h1>
      <p>Your learning journey starts here!</p>
    </div>
  );
}

export default Home;
```

src/pages/Login.js:

javascript

```
import React from 'react';
import { Login } from '../components/LMSComponents';

function LoginPage() {
  return <Login />;
}

export default LoginPage;
```

src/pages/Courses.js:

javascript

```
import React from 'react';
import { CourseRecommender } from '../components/LMSComponents';

function Courses() {
  return <CourseRecommender />;
}

export default Courses;
```

src/pages/Chatbot.js:

javascript

```
import React from 'react';

function Chatbot() {
  return (
    <div style={{ padding: '20px' }}>
      <h2>AI Chatbot</h2>
      <p>Chat functionality coming soon!</p>
    </div>
  );
}

export default Chatbot;
```

Step 6: Complete Required React Tasks

1. Create src/components/PasswordStrength.js:

- Implement password strength checker
- Use useState for password and result tracking
- Add validation logic for weak/strong passwords

2. Create src/components/CourseToggle.js:

- Implement toggle functionality for course descriptions
- Use useState for visibility state
- Add conditional rendering

Step 7: Start Development Servers

1. Start React development server:

```
bash
cd client
npm start
```

This will open <http://localhost:3000>

2. For HTML/CSS modules:

- Use Live Server extension in VS Code
- Or open files directly in browser

Step 8: Project Testing Checklist

HTML/CSS Module:

- ☒ Navigation bar displays correctly
- ☒ Flexbox layout works responsively
- ☒ Bootstrap cards appear side by side
- ☒ All styling renders properly

JavaScript Module:

- ☒ Button alerts work
- ☒ DOM manipulation functions
- ☒ Email validation works
- ☒ Keypress events update content

React Application:

- ☒ Routing between pages works
- ☒ Login form validates input
- ☒ Course recommender provides suggestions
- ☒ Password strength checker functions
- ☒ Course toggle shows/hides content

Next Steps for AI Integration

1. Backend Setup (Future modules):

- Node.js/Express server
- Database integration
- AI/ML API connections

2. Advanced Features:

- Real-time chat
- Learning analytics
- Personalized recommendations

Troubleshooting Tips

- If npm install fails: Clear cache with `npm cache clean --force`
- If Live Server doesn't work: Install "Live Server" extension in VS Code
- If React routes don't work: Check React Router installation

- For styling issues: Verify Bootstrap CDN links are working

File Organization

Keep your project organized with this structure:

```
ai-lms-project/  
├── client/src/  
│   ├── components/  
│   ├── pages/  
│   └── assets/  
├── module3-html-css/  
└── module5-javascript/
```