Week 2: Building Full-Stack Skills

Week Overview:

This week focuses on building a full-stack application using **React (frontend)**, **Node.js/Express (backend)**, and **PostgreSQL (database)**. The schedule is reorganized to introduce **Authentication and JWT** earlier for better structuring and smoother implementation throughout the week.

Monday: React Fundamentals

- Introduction to React and component-based architecture
- Setting up a React application with Create React App
- JSX syntax and component rendering
- Functional components and hooks
- State management with useState
- Props and component hierarchies
- Event handling in React
- Conditional rendering

Daily Assignment:

1. Create a new React application using `npx create-react-app task-manager`.
2. Build a `TaskList` component that:
    - Displays at least 3 hardcoded tasks (e.g., ["Walk dog", "Write code", "Read docs"])
    - Uses `useState` to manage the list of tasks
    - Includes a simple form (input + button) to add new tasks to the list
    - Implements a button for each task to remove it from the list

Submission: GitHub repository link with your React application.

Tuesday: React Data Fetching & Effects

- React `useEffect` hook for side effects
- Data fetching in React
- Working with external APIs
- Loading states and error handling
- React Router for multi-page applications
- Navigation and route parameters
- Forms and controlled components
- Styling in React (CSS modules or styled-components)

Daily Assignment: Enhance your Task Manager application to:

- Fetch tasks from a mock API (use JSONPlaceholder: `https://jsonplaceholder.typicode.com/todos`)
- Display loading state while fetching data
- Handle potential errors during fetch operations
- Add React Router with at least two routes:
    - Home page showing the task list
    - About page with information about the app
- Add a "Complete" button for each task that visually indicates completion (UI state only)

Submission: Updated GitHub repository link.

Wednesday: Node.js & Express Backend + Authentication Introduction

- Introduction to backend development with Node.js
- Setting up an Express server
- RESTful API principles
- Building API endpoints (GET, POST, PUT, DELETE)
- Request and response handling
- Middleware in Express
- API testing with Postman
- Environment configuration
- Introduction to JWT and Authentication concepts
- Securing API endpoints with JWT

Daily Assignment: Create a new Express server with the following features:

- Set up a basic Express application (`npm init` and install required packages)
- Create a tasks array to store tasks in memory (no database yet)
- Implement the following endpoints:
  - `GET /api/tasks` - returns all tasks
  - `POST /api/tasks` - adds a new task (accepts JSON with `{text: "Task description"}`)
  - `DELETE /api/tasks/:id` - removes a task by id
- Add JWT authentication middleware for protected routes
- Add basic error handling middleware
- Include CORS middleware to allow requests from your React app

Submission: GitHub repository link for your backend code + Screenshots of Postman tests for each endpoint.

Thursday: Database Integration with PostgreSQL

- Introduction to PostgreSQL
- Database design fundamentals
- Setting up PostgreSQL locally
- Connecting Node.js to PostgreSQL
- CRUD operations with SQL
- Implementing database queries in Express
- Data validation and sanitization
- Error handling for database operations

Daily Assignment: Enhance your Express backend with PostgreSQL:

- Create a `tasks` table in PostgreSQL with fields:
  - `id` (SERIAL, PRIMARY KEY)
  - `text` (VARCHAR, NOT NULL)
  - `completed` (BOOLEAN, DEFAULT false)
  - `created_at` (TIMESTAMP, DEFAULT NOW())
- Connect your Express app to PostgreSQL using `node-postgres` or another ORM

- Update your API endpoints to interact with the database

Submission: Updated backend repository + SQL schema creation script + Screenshot showing rows in your database.

Friday: Full-Stack Integration

- Connecting React frontend to Express backend
- Protected routes in React and Express with JWT
- Advanced state management
- Deployment strategies for full-stack applications
- Code review and best practices
- Performance optimization

Daily Assignment: Connect your React frontend to your Express/PostgreSQL backend:

- Update your React app to fetch tasks from your own API (not JSONPlaceholder)
- Implement forms to add new tasks that save to your database
- Add delete functionality to remove tasks
- Implement task completion toggle (updates database)
- Secure protected routes with JWT in your React app
- Add basic error handling for API operations
- Include loading states for all API operations

Submission:

- Frontend GitHub repository link
- Backend GitHub repository link
- Screenshots or video demo of the working application
- (Optional) Deployed application links if completed