Day 2: Backend Integration and UI Prototyping, based on JWT Authentication and UI Prototyping in React.

Problem Statement

Build a Contacts Management Application: You are required to create a contacts management application that allows users to register, log in, and manage their contacts securely. The application should utilize JWT for user authentication and demonstrate UI prototyping principles.

User Stories

1. **As a user**, I want to register for a new account so that I can manage my contacts.
2. **As a user**, I want to log in to my account using JWT authentication to access my contacts securely.
3. **As a user**, I want to view a list of my contacts after logging in so that I can manage them.
4. **As a user**, I want to add new contacts to my list so that I can keep track of important people.
5. **As a user**, I want to edit existing contacts so that I can update their information.
6. **As a user**, I want to delete contacts from my list so that I can manage my contacts efficiently.
7. **As a developer**, I want to prototype UI designs and translate them into functional React components.

Assignment Structure

1. Setup (10 Minutes)

- Create a new React application using `create-react-app`.
- Set up a backend service (Node.js/Express) for handling user authentication and contact management.
- Install necessary dependencies such as `jsonwebtoken`, `axios`, and any UI component libraries you prefer (e.g., Material-UI).

2. Build the Backend (20 Minutes)

- **User Registration and Authentication**
  - Implement API endpoints for user registration and login.
  - Use JWT for creating and validating user sessions.
  - Set up middleware to protect routes and verify JWTs.

- **Contacts Management API**
  - Implement API endpoints for adding, editing, viewing, and deleting contacts.
  - Ensure that these endpoints are protected and only accessible to authenticated users.

3. Build the Frontend (20 Minutes)

- **Authentication UI**
  - Create a registration form where users can enter their details to sign up.
  - Create a login form that uses JWT for authentication and stores the token in local storage.
- **Contacts UI**
  - Implement a contacts management UI that displays a list of contacts.
  - Use reusable UI components for forms (for adding/editing contacts) and lists (for displaying contacts).
- **UX Design Integration**
  - Utilize a prototyping tool (like Figma or Adobe XD) to design the UI before implementation.
  - Ensure that the React components match the designed prototypes.

4. Final Touches (10 Minutes)

- **Add Functionality**
  - Ensure that users can add, edit, and delete contacts securely.
  - Display success and error messages based on user actions.

Submission Requirements

- Submit the code via a Git repository link.
- Include a README.md file that provides:
  - Instructions on how to run both the frontend and backend applications locally.
  - An overview of the application architecture, including how JWT is used.
  - Screenshots of the UI prototypes used and explanations of how they were implemented.

Evaluation Criteria

- **Functionality**: All user stories should be implemented correctly, with a smooth user experience.
- **Code Quality**: Clean, organized code following best practices.
- **JWT Authentication**: Proper implementation of JWT for user authentication and route protection.
- **UI Prototyping**: Effective translation of UI designs into functional React components.

- **User Interface**: The application should be user-friendly and visually appealing based on UX principles.

Final Note

This coding assignment is designed to test your skills in integrating backend services with frontend React applications, focusing on JWT authentication and UI prototyping. Please manage your time effectively to ensure all components are completed within the allotted 60 minutes. Good luck!