SparkINN Novice Solution Pvt. Ltd

## 1st Weekend Task: Building a Feature-Rich Personal Dashboard

**Objective:** Create a personal dashboard application that utilizes the React concepts learned throughout the course. This project will combine various features, demonstrating your understanding of React components, state management, event handling, and more.

**Problem Statement:** Develop a "Personal Dashboard" React application that includes the following functionalities:

1. **Welcome Section:**
   - Display a dynamic welcome message based on the time of day (e.g., "Good Morning, [User Name]!").
   - Use a functional component to render this greeting.
2. **User Profile Component:**
   - Create a user profile section that displays the user’s name, profile picture, and a short bio.
   - Use props to pass the user’s information into the profile component.
3. **Task Manager:**
   - Implement a task manager where users can add, view, and delete tasks.
   - Use `useState` to manage the list of tasks.
   - Use `map` to render the list of tasks dynamically.
4. **Weather Widget:**
   - Create a weather widget that fetches and displays the current weather of the user’s location.
   - Use the `useEffect` hook to fetch weather data from an API when the component mounts.
5. **Theme Switcher:**
   - Implement a theme switcher that toggles between light and dark themes.
   - Use state to manage the current theme and apply conditional styling.
6. **User Authentication (Optional):**
   - Add a login form that allows users to "log in" using a username and password.
   - Display a personalized message and allow access to the dashboard only after successful login.
7. **Event Handling:**
   - Implement event handling for button clicks, form submissions, and user interactions.
   - Ensure proper binding of event handlers in class components or use arrow functions in functional components.
8. **Conditional Rendering:**
   - Use conditional rendering to show different components or messages based on user interactions (e.g., show a "No tasks available" message if the task list is empty).
9. **List and Keys:**
   - Use `map` to render lists of items (e.g., tasks) and ensure that each list item has a unique `key` prop.

10. **Forms Handling:**
    - Create a form that allows users to add new tasks.
    - Use controlled components for form inputs and handle form submission to update the task list.
11. **Styling and CSS Basics:**
    - Apply custom styling to the components using CSS modules or styled-components.
    - Ensure the app is responsive and visually appealing.

**Deliverables:**

- A fully functional React app that demonstrates your understanding of React concepts.
- A GitHub repository link with the project files.
- A brief README file explaining how to set up and run the project, along with a description of each feature.

**Evaluation Criteria:**

- Proper use of React components (functional and class components).
- Effective use of state (`useState`) and hooks (`useEffect`).
- Clean and maintainable code structure.
- Proper handling of events and form inputs.
- Responsive design and good user experience.
- Extra credit for implementing user authentication.

**Submission Deadline:** Sunday night, 11:59 PM.