User Management Dashboard

A **responsive ReactJS dashboard** to display and manage user information. Supports **searching users**, **adding users**, **deleting users**, and **light/dark theme toggle** using **Tailwind CSS** and **Redux Toolkit**.

Features

- **Responsive Design:** Works on desktop, tablet, and mobile devices.
- **User Listing:** Fetches users from JSONPlaceholder API and displays them in cards.
- **Search/Filter:** Filter users by name, email, or company.
- **Add User:** Add new users through a modal form with validation.
- **Delete User:** Remove users directly from the list.
- **Theme Toggle:** Light and dark themes with global context.
- **State Management:** Uses Redux Toolkit to manage user data.
- **Error Handling:** Shows loading states and handles API errors gracefully.

Tech Stack

- **Frontend:** React (Functional Components + Hooks)
- **State Management:** Redux Toolkit
- **Styling:** Tailwind CSS with class-based dark mode
- **API:** Axios / Fetch API
- **Build Tool:** Vite

Project Structure

```
src/
├── components/
│   ├── common/
│   │   ├── Button.jsx
│   │   ├── Card.jsx
│   │   └── Modal.jsx
│   └── users/
│       ├── UserCard.jsx
│       ├── UserList.jsx
│       └── AddUserForm.jsx
├── store/
│   ├── index.js
│   └── slices/
│       └── userSlice.js
├── themes/
│   └── ThemeProvider.jsx
├── utils/
│   └── constants.js
```

```
├── App.jsx
└── main.jsx
```

Installation

1. **Clone the repository**

```
git clone https://github.com/Niteshsharma831/Internship-Task.git
```

2. **Navigate to the User Management Dashboard folder**

```
cd "Internship-Task/TechX-React-Assignment/User-Management-Dashboard"
```

3. **Install dependencies**

```
npm install
```

4. **Start the development server**

```
npm run dev
```

5. **Open in browser** Visit http://localhost:5173 to view the app.

Usage

- **View Users:** See all users displayed in cards.
- **Search Users:** Type in the search bar to filter users by name, email, or company.
- **Add User:** Click + Add User, fill in the details in the modal, and submit.
- **Delete User:** Click the delete button on any user card to remove it.
- **Toggle Theme:** Click 🌙 Dark or ☀ Light to switch between light and dark mode.

Approach

1. **Layout & Design:** Used CSS Grid/Flexbox for a responsive layout and Tailwind CSS for consistent styling.
2. **API Integration:** Fetched user data from JSONPlaceholder API with loading and error handling.
3. **Reusable Components:** Created components like **Button, Card, Modal, UserCard** for reusability.
4. **Redux:** Managed user data globally, with actions to fetch, add, and delete users.

5. **Theme Management:** Implemented **ThemeProvider** for light/dark theme toggle.
6. **Fallback Data:** Used static local data if API fails.

Challenges & Solutions

- **Dynamic Dark Mode:** Initially, UI didn't update on toggle.

    - **Solution:** Added `.dark` class on `<html>` and `transition-colors` for smooth changes.

- **API Failure Handling:** Added fallback static data to ensure the app works if API is down.

- **Responsive Layout:** Adjusted Grid and Flex layouts for mobile and tablet screens.

Author

Nitesh Sharma Email: niteshkumarsharma831@gmail.com
Phone: +91 9572861917 LinkedIn: https://www.linkedin.com/in/nitesh-kumar-sharma-2894a1185/
Portfolio: https://devcraftnitesh.vercel.app/

Previous Projects

- **E-Commerce Website:** Shopizo Online
- **Job Seeking Platform:** HireOn WorkBridge