

# Blog Application Assignment (Frontend + Backend)

**Objective:** Develop a comprehensive blog application using React.js for the frontend with Material-UI (MUI) for styling, Redux for state management, and a rich text editor to create and edit blog posts. The application should also include a theme switcher for toggling between light and dark modes. Additionally, implement a backend to manage CRUD operations for blog content, user authentication, users can share blogs with others, specifying permission levels (view/edit) for shared access.

## Frontend Deliverables:

### 1. Core Features:

- **Blog Management:** Implement CRUD operations to allow users to create, read, update, and delete blog posts.
- **Rich Text Editor:** Integrate a rich text editor for blog creation and editing.
- **Theme Switcher:** Implement a theme switcher to toggle between light and dark modes.
- **Responsive UI:** Ensure the UI is responsive and works seamlessly across different screen sizes.

### 2. Pages and Navigation:

- **HomePage:** Display a list of featured blogs with their titles, excerpts, and publication dates. Include a sidebar or navbar with categories (e.g., Technology, Travel, Food, Lifestyle).
- **Category Pages:** Create separate pages for each category, displaying blogs specific to that category.
- **Filtering and Search:** Implement filtering by category and a search option for users to find specific blogs.
- **Blog Details Page:** Clicking on a blog title or excerpt should navigate the user to a detailed view of that blog, displaying the full content, image, author, publication date, and metadata.

### 3. State Management:

- Use Redux for centralized state management, ensuring efficient actions and reducers.

## Backend Deliverables:

### 1. Technology Stack:

- Select any backend technology stack (e.g., Node.js with Express, Python with Django) for implementing CRUD operations and data storage.

### 2. Database Setup:

- Set up any database (e.g., PostgreSQL, MongoDB) to store blog content, images, user information, and permissions.
- Define schemas to manage blog posts, users, and permissions.

### 3. User Authentication and Authorization:

- Implement user authentication and authorization to secure API endpoints.
- Use JWT or session-based authentication to ensure CRUD operations are restricted to authenticated users.

### 4. Blog Sharing and Permissions:

- Enable users to share their blogs with others by specifying another user's ID.

- Implement access control to define permissions (view or edit) for shared blogs.
- 5. **CRUD Operations:**
  - Implement CRUD operations for managing blog posts (Create, Read, Update, Delete).
- 6. **Error Handling:**
  - Implement error handling for API requests and responses.
  - Provide meaningful error messages to users in case of failures.
- 7. **User Dashboard:**
  - After login, users should be able to see their personal blogs.
  - Users can create, edit, and delete their blogs.
  - Users can share their blogs with others and define access levels (view or edit).

#### Evaluation Criteria:

- **Functionality:** The application should meet all specified requirements and function correctly without bugs.
- **Code Quality:** The code should be clean, well-organized, and follow best practices for React, Redux, and backend technologies.
- **User Interface:** The UI should be intuitive, responsive, and visually appealing.
- **State Management:** Proper use of Redux for state management with efficient actions and reducers.
- **CRUD Operations:** Effective implementation of CRUD operations for blog management.
- **Theme Management:** Effective implementation of the theme switcher with persistence across sessions.

#### Submission:

- Create a private GitHub repository and share it with prime@vecros.com and abhay\_gond@vecros.com.
- Include a README.md with a description, features, setup instructions, and the deployment URL.
- Deploy your application frontend and backend on a hosting platform like Vercel, Netlify, or any other deployment platform
- Ensure the application is fully functional, thoroughly tested, and push all final changes to the repository.
- Make sure that every function of backend and frontend should be implemented in your assignment

---

This assignment description now includes the requirement for CRUD operations to manage blog posts, ensuring comprehensive functionality for creating, editing, viewing, and deleting blogs within the application.