Project Report: React Blog Platform

---

1. Introduction

The React Blog Platform project is a web application designed to provide users with the ability to create, view, edit, and delete blog posts. This application demonstrates core CRUD (Create, Read, Update, Delete) functionalities, essential front-end development principles, and simple data persistence using local storage.

---

2. Project Objectives

The primary objectives of this project are to:

- Build a responsive, user-friendly blog application.
- Implement CRUD functionalities with React.
- Explore and apply state management in React.
- Enable simple data persistence to retain data after a page refresh.
- Use Tailwind css to create an aesthetically pleasing and intuitive user interface.

---

3. Technologies Used

- **Frontend**: React
- **Routing**: React Router
- **Styling**: Tailwind css, CSS
- **Data Persistence**: Local Storage

---

4. Features and Functionalities

4.1 Create

- Users can add new blog posts by filling out a form with fields for the post title and content.
- The new post is saved and immediately displayed in the list of blog posts.

4.2 Read

- All blog posts are displayed on the home page in a concise format, showing titles and brief content previews.
- Users can click on a blog post to view the full content on a separate page.

4.3 Update

- Users can edit existing blog posts by clicking the "Edit" button on each post.
- The edit functionality uses the same form as the create function, allowing users to update the post’s title or content.

4.4 Delete

- Users can delete any blog post by clicking the "Delete" button next to the post.
- Deleting a post removes it from both the display list and local storage.

---

5. Application Structure and Components

5.1 Components

1. **BlogList Component**
   - Displays a list of all blog posts.
   - Includes options to read, edit, or delete each post.
2. **BlogForm Component**
   - Provides a form to add or edit a blog post.
   - Contains input fields for title and content, with appropriate validation.
3. **BlogDetail Component**
   - Displays the full content of a selected blog post.
   - Provides a link back to the main list of blog posts.

---

6. Data Persistence

To retain data after refreshing, this project uses the **Local Storage API**. When a post is created, edited, or deleted, the state is updated in local storage, ensuring data remains persistent across sessions.

---

7. Conclusion

The React Blog Platform project successfully achieves the goal of building a basic blog application with essential CRUD functionalities. Through this project, foundational concepts of React development, routing, state management, and Tailwind css styling were applied. Additionally, data persistence using local storage demonstrates a simple yet effective way to retain user data across sessions.