A
Project Report
Blog Web Application

Introduction:

The purpose of this project is to develop a React web application that fetches posts from an external API and allows users to perform various actions such as delete, real-time search, upload posts, and sort posts based on their titles. The application is built using React, Redux, Redux-Thunk, and Tailwind CSS for styling.

Technologies used:

1. React: React is a JavaScript library used for building user interfaces. It is widely used for creating single-page applications and mobile applications.
2. Redux: Redux is a predictable state container for JavaScript apps. It is used for managing the state of the application.
3. Redux-Thunk: Redux-Thunk is middleware used for handling asynchronous actions in Redux.
4. Tailwind CSS: Tailwind CSS is a utility-first CSS framework used for styling web applications.

Features:

1. Fetching Posts: The application fetches posts from an external API (Add a little bit of body text) using Axios and displays them on the screen.
2. Delete Posts: Users can delete posts by clicking on the delete button associated with each post.
3. Real-time Search: Users can search for posts in real-time by entering keywords in the search box.
4. Upload Posts: Users can upload posts by clicking on the upload button and entering the post title and body.
5. Sort Posts: Users can sort posts based on their titles in ascending or descending order.

Design and Implementation:

The application is designed using React components and Tailwind CSS for styling. The application is divided into several components, including the App component, the Header component, the Feed component, and the Blog component. The App component is the root component that renders the Header component and the Feed component. The Header component contains the search and upload features, while the Feed component contains the list of Blog components that display the posts fetched from the API. The application uses Axios to fetch the posts from the external API. Redux is used for managing the state of the application, and Redux-Thunk is used for handling asynchronous actions. The state of the application is divided into two slices: the post slice and the search slice. The post slice contains the list of posts fetched from the API, while the search slice contains the search keyword entered by the user.

The Blog component is used for displaying each post fetched from the API. The component contains the title, body, and delete button associated with each post. Users can delete posts by clicking on the delete button, and the application updates the state of the post slice accordingly. The search feature is implemented by updating the state of the search slice with the search keyword entered by the user. The application then filters the posts in real-time based on the search keyword entered by the user. The upload feature is implemented by creating a form for users to enter the post title and body. The form data is then sent to the external API using Axios.