

Задача: Расширение функционала приложения для управления постами в React

Описание:

У вас уже есть React-приложение, которое:

- Выводит список постов.
- Позволяет добавлять новые посты.

Необходимо добавить функционал:

1. **Удаление постов** — возможность удалить пост из списка.
2. **Редактирование постов** — возможность изменить текст существующего поста.

Используйте хуки `useState` и `useEffect` для управления состоянием и жизненным циклом компонента. Сохраняйте посты в `json-server`, чтобы изменения сохранялись между перезагрузками страницы.

Требования:

Посты хранятся в виде массива объектов с полями:

```
{  
  id: number, // Уникальный идентификатор  
  title: string // Название поста  
  views: number // Количество просмотров  
}
```

- Добавьте кнопку "Удалить" для каждого поста, которая удаляет его из списка.
- Добавьте кнопку "Редактировать" для каждого поста, которая позволяет изменить текст поста.
- Сохраняйте изменения в `json-server` при добавлении, удалении или редактировании постов.
- Восстанавливайте список постов из `json-server` с помощью GET запроса при загрузке страницы.
- Обеспечьте базовую валидацию: текст поста не должен быть пустым при добавлении или редактировании.

Что уже сделано:

- Компонент отображает список постов.
- Реализована форма для добавления новых постов.
- Используются хуки `useState` для управления списком постов и полем ввода.
- Посты сохраняются в `json-server` с помощью POST запроса при добавлении.

Задачи для реализации:

1. **Удаление постов:**

- Добавить кнопку "Удалить" рядом с каждым постом.
- При клике на кнопку удалять пост из состояния и обновлять json-server с помощью DELETE запроса.

2. Редактирование постов:

- Добавить кнопку "Редактировать" рядом с каждым постом.
- При клике на кнопку показывать форму (или поле ввода) с текущим текстом поста.
- После редактирования сохранять обновленный текст в состоянии и в json-server с помощью PUT запроса.
- Добавить кнопку "Сохранить" для подтверждения изменений и "Отмена" для возврата к исходному состоянию..

3. UI/UX:

- Обеспечить интуитивный интерфейс: например, выделять пост, который редактируется.
- Показывать сообщение об ошибке, если пользователь пытается сохранить пустой пост.