**Лабораторная работа №11. Разработка приложения "Post Manager" с использованием внешнего API**

**Задание:** создать одностраничное веб-приложение для управления списком постов. Пользователь должен иметь возможность:

* Просматривать список постов
* Добавлять новые посты
* Редактировать существующие посты
* Удалять посты

Реализовать модуль postsAPI.ts, содержащий функции для выполнения CRUD-операций (создание, чтение, обновление, удаление) с внешним API ресурса posts на основе https://jsonplaceholder.typicode.com/posts.

**Требования:**

* Использование Redux Toolkit (createSlice, createAsyncThunk)
* Типизированные компоненты и состояние
* Реализованы все CRUD-операции

**Технологии:**

* **React**
* **Redux Toolkit**
* **TypeScript**
* **Axios** (для запросов к API)
* **JSONPlaceholder API** (https://jsonplaceholder.typicode.com/posts)

**Требования к функционалу:**

**1. Просмотр постов:**

* Получить и отобразить список постов из API.
* Отображать title и body каждого поста.
* Загрузка постов происходит при монтировании компонента.

**2. Добавление поста:**

* Форма с полями title и body.
* После отправки формы пост добавляется в общий список.

**3. Редактирование поста:**

* Возможность редактировать title и body существующего поста.
* После редактирования пост обновляется в списке.

**4. Удаление поста:**

* Возможность удалить пост из списка.

**Общие требования к модулю** postsAPI.ts

* Использовать библиотеку **axios** для выполнения HTTP-запросов.
* Все функции должны быть **асинхронными** (async/await).
* Экспортировать каждую функцию по отдельности.
* Все функции должны возвращать типизированные значения.
* Предусмотреть типизацию данных (Post, NewPost и т.д.).

**Требуемый функционал**

**1. fetchPosts**

* Назначение: получить список всех постов.
* Метод: GET
* URL: https://jsonplaceholder.typicode.com/posts
* Возвращаемое значение: массив объектов Post.

**2. createPost**

* Назначение: создать новый пост.
* Метод: POST
* URL: https://jsonplaceholder.typicode.com/posts
* Аргумент: объект NewPost (без id)
* Возвращаемое значение: объект Post с присвоенным id.

**3. updatePost**

* Назначение: обновить существующий пост.
* Метод: PUT
* URL: https://jsonplaceholder.typicode.com/posts/:id
* Аргумент: объект Post (включая id)
* Возвращаемое значение: обновлённый объект Post.

**4. deletePost**

* Назначение: удалить пост по ID.
* Метод: DELETE
* URL: https://jsonplaceholder.typicode.com/posts/:id
* Аргумент: id поста (число)
* Возвращаемое значение: void (ничего не возвращает при успешном удалении).

**Структура проекта:**

src/

├── app/

│ └── store.ts

├── features/posts/

│ ├── postsSlice.ts

│ ├── postsAPI.ts

│ └── Posts.tsx

├── components/

│ └── PostForm.tsx

│ └── PostItem.tsx

├── App.tsx

└── main.tsx