

Задание: Разработка приложения "Менеджер задач" на React + TypeScript

💡 **Цель:** Разработать веб-приложение "Менеджер задач", позволяющее просматривать список задач и редактировать их.

🔗 Технические требования

1. Стек технологий:

- React (с использованием TypeScript)
- React Router v6 для маршрутизации
- Сборщик проекта : на ваш выбор (н-р, Vite, Webpack, Rspack и др.)
- UI-фреймворк : на ваш выбор (н-р, Ant Design, Material UI, AdmiralDS и др.)
- Инструменты стилизации: на ваш выбор (н-р, css modules, styled-components и др.)

2. Особенности реализации:

- Поддержка SPA: переход между страницами без перезагрузки
- При работе со списком задач можно использовать React Hooks и React Context API.
- Поддержка 2х последних версий всех браузеров

3. Маршруты:

- Главная страница с задачами: /
- Страница детальной информации и редактирования задачи: /task/:id

4. Компоненты:

✓ TaskList

- Отображает список задач в виде адаптивной сетки карточек (Grid или Flexbox).
- Каждая задача отображается через компонент TaskItem

✓ TaskItem

- Карточка задачи, содержащая:
- Обязательный текстовый заголовок
- Опциональное текстовое описание
- Цветовые плашки (например, Chip/Tag) с информацией:
 - Категория: Bug / Feature / Documentation / Refactor / Test
 - Статус: To Do / In Progress / Done
 - Приоритет: Low / Medium / High
- Кнопка "Редактировать" открывает форму редактирования. Можно реализовать как модальное окно, так и отдельную страницу по маршруту /task/:id.

✓ TaskDetails

- Форма редактирования задачи, открывается по маршруту /task/:id.
- Отображает полную информацию о задаче.
- Форма позволяет изменить:

- Заголовок
- Описание
- Категорию
- Статус
- Приоритет
- Кнопки: "Сохранить", "Отмена". Нажатие на кнопку приводит к сохранению внесенных изменений и возврату к списку задач.

Дополнительные рекомендации

1. Примените стилизацию компонентов согласно выбранному UI-фреймворку.
2. Убедитесь в том, что интерфейс адаптирован под разные устройства (desktop / mobile).
3. Добавьте возможность фильтрации задач по статусу, категории или приоритету (по желанию).

Что нужно сдать?

1. Ссылка на GitHub-репозиторий с решением.
2. README.md с кратким описанием проекта, инструкцией по запуску и используемыми технологиями.
3. Рабочее приложение с корректно реализованной навигацией и функционалом редактирования задач.

Примеры внешнего вида (на усмотрение студента):

- Можно ориентироваться на популярные менеджеры задач (Trello, Notion, Jira и т.д.)
- Важно соблюдать читаемость и удобство взаимодействия.