

# Тестовое задание на вакансию Frontend-разработчик

Необходимо написать одностраничный сайт с заметками, хранящимися в браузере.

Каждая заметка содержит следующие данные:

- Название (текст);
- Содержание (текст);
- Дата создания (дата-время);
- Дата последнего редактирования (дата-время).

Все заметки должны выводиться в виде списка карточек в центре страницы.

На карточке должны находиться все данные заметки, а также интерактивные элементы для реализации возможности:

- Редактирования заметки (редактируется только название и содержание);
- Удаления заметки.

Для хранения данных использовать хранилище браузера.

Для удобства пользователя необходимо реализовать сортировку карточек по:

- Названию;
- Дате создание;
- Дате последнего изменения.

Интерактивные элементы вне карточек должны находиться в шапке страницы.

Исходный код необходимо опубликовать в репозитории на удобной git-платформе (рекомендуется Github).

В изменения в репозитории должны быть разделены на коммиты по смысловым блокам.

## Чек-лист исходного кода

- ☐ Весь код написан с использованием TypeScript;
- ☐ Код не содержит необоснованных типов `any`;
- ☐ Описаны типы для всех структур данных;
- ☐ Используются исключительно функциональные компоненты;
- ☐ Применена мемоизация (React.memo, useState, useMemo, useCallback, useEffect и т.д.);
- ☐ Применён LESS;
- ☐ Применён React Bootstrap;
- ☐ Код в репозитории разделён на коммиты;
- ☐ Коммиты имеют названия, кратко описывающие внесённые изменения.

## Чек-лист готового продукта

- ☐ При открытии страницы загружаются все существующие заметки в список;
- ☐ На карточках отображается вся информация о конкретной заметке;
- ☐ На сайте есть кнопка для добавления новой карточки и она выполняет свои функции;
- ☐ На карточке получается воспользоваться интерактивными элементами и отредактировать/удалить заметку;
- ☐ После закрытия и нового открытия страницы и браузера заметки не изменяются;
- ☐ На странице имеются интерактивные элементы для сортировки заметок по названию/дате создания/дате редактирования и они выполняют свои функции.

## Полезные примечание

Для упрощения разработки предлагается к использованию [Create React App](#). Он содержит готовый шаблон приложения для разработки на ReactTS.

Для улучшения визуальной составляющей и упрощения стилизации рекомендуется [React Bootstrap](#), он позволяет минимизировать написание собственных компонентов на HTML и CSS.

Для собственных стилей рекомендуется использовать [LESS](#).

Для проверки навыков работы требуется не использовать тип `any` в [TypeScript](#) без обоснованной необходимости.

Готовый код можно вложить в репозиторий для портфолио.

# Время выполнения

Для личной статистики рекомендуется фиксировать время поиска информации и выполнения работы.

Уровень навыка	Приблизительное время выполнения
Стажёр	4 дня (32 часа)
Junior	1.5 дня (12 часов)
Middle	1.5 часа
Senior	1 час
Lead	1 час