

Тестовое задание на позицию **Junior Fullstack JS Developer**

Приложение: одностраничное Web приложение для учета и создания задач (To Do List), основной функционал: добавление задач (максимум - 255 символов), удаление задач, изменение существующих задач: изменение текста задачи и статуса её готовности, фильтрация существующих задач и отображение небольшой статистики (количества всех задач к выполнению и количество выполненных задач). Дизайн на Ваше усмотрение, но как пример рекомендуем это: <https://dev-to-uploads.s3.amazonaws.com/i/o96lsrld21tk232kidu4.png> (для изменения текста - дабл клик по задаче, для изменения статуса один клик), язык интерфейса - английский

Front-end часть:

Должна быть реализована с использованием React 17 версии (или vue.js) с использованием функциональных компонентов и хуков, сами задачи должны храниться в Redux, а для асинхронных задач рекомендуется использовать Redux-Saga или же Redux-Thunk, сами же запросы должны отправляться с использованием axios. Для компонентов можно использовать material-ui (mui) библиотеку, при создании собственных компонентов большим плюсом будет использование атомарного дизайна. Для CSS рекомендуется использовать styled-components или @emotion/css. Для инициализации проекта можно использовать create-react-app, но ручная настройка через webpack 5 версии и выше будет плюсом. Так же полная реализация на TS и использование тайпингов тоже будет плюсом

Back-end часть:

Должна быть реализована на node.js с использованием одной из библиотек на выбор: express.js/koa.js/fastify/nest.js. При выборе Nest.js - TS для back-end части становится обязательным, рекомендуется выбрать fastify. СУБД - MongoDB или PostgreSQL, рекомендуемые драйвера к Node.js - mongodb для MongoDB и pg для PostgreSQL соответственно. Разделение логики на контроллеры для обработки HTTP части и сервисы для обработки непосредственно доменной логики будет плюсом. Реализация на TS и использование тайпингов будет плюсом, можно использовать ts-node, но опять же рекомендуется использовать webpack 5+

Оформить проект можно в виде моно репозитория или двух репозиториев, использование docker/docker-compose для запуска проекта будет большим плюсом, но описание по запуску всего проекта для тестирования обязательна в README файлах репозиториев

