

**Тестовое задание: Учетная книга финансов****Стек технологий**

- React Hooks
- TypeScript
- MobX
- React Router (или аналог)
- NodeJS
- Express или Fastify (или NestJS, если есть опыт)
- PostgreSQL
- Sequelize или TypeORM

**Описание**

Создать одностраничное приложение с маршрутизацией, в котором будут следующие маршруты:

- Страница авторизации
- Страница регистрации
- Страница со списком кошельков
- Страница кошелька
- Страница доходов
- Страница расходов
- Описание функционала страниц

**Страница авторизации**

На этой странице отображается форма аутентификации, состоящая из следующих полей:

- Логин
- Пароль

Длина логина должна быть не меньше 8 символов и не больше 255 символов. Длина пароля не должна быть больше 72 символов. Для валидации полей можно использовать библиотеку *joi*.

Также на этой странице должна быть ссылка на страницу регистрации. Если пользователь не авторизован – открытие любой другой страницы, кроме страницы регистрации, должно перенаправлять на эту страницу.

**Страница регистрации**

На этой странице отображается форма регистрации, состоящая из следующих полей:

- Логин
- Пароль
- Повторите пароль

Длина логина должна быть не меньше 8 символов и не больше 255 символов. Длина пароля не должна быть больше 72 символов. Если значение поля «Пароль» не совпадает со значением поля «Повторите пароль», тогда форма считается не валидной. Для валидации полей можно использовать библиотеку *joi*.

Также на этой странице должна быть ссылка на страницу аутентификации.

## Страница со списком кошельков

На этой странице отображаются все кошельки, которые были добавлены пользователем. Также на этой странице должна быть кнопка «Добавить кошелёк». Кошелёк можно редактировать и удалять. Удаление должно происходить с подтверждением. Реализация добавления, редактирования и удаления должна быть с использованием модального окна.

Информация, которая должна отображаться для каждого кошелька: название, текущий баланс. По нажатию на кошелёк должна открываться страница кошелька.

## Страница кошелька

На этой странице отображаются списки расходов и доходов кошелька. Отображаться должны только последние 5 значений в каждом списке. Под каждым списком должна быть кнопка «Показать все», которая откроет либо страницу расходов, либо страницу доходов, зависит от списка. На этой странице значения расходов\доходов редактировать или удалять нельзя.

## Страница доходов

На этой странице отображаются все доходы, сверху должна быть кнопка «Добавить доход». Доходы можно редактировать и удалять, удаление с подтверждением. Информация, которая должна быть у каждого дохода: название, значение. Последний добавленный доход должен отображаться в самом верху списка.

## Страница расходов

На этой странице отображаются все расходы, сверху должна быть кнопка «Добавить расход». Расходы можно редактировать и удалять, удаление с подтверждением. Информация, которая должна быть у каждого расхода: название, значение. Последний расход должен отображаться в самом верху списка.

## Меню

На каждой странице, кроме страниц «Аутентификация» и «Регистрация», сверху должно находиться меню, в котором должны быть следующие пункты:

Мои кошельки — открывает страницу кошельков

Выход — выход из текущей сессии пользователя и перенаправление на страницу аутентификации

## Описание серверной части

Пока пользователь не авторизован, доступ до REST API маршрутов, связанных с кошельком, доходами и расходами, должен быть запрещен. Один пользователь не должен иметь возможность взаимодействовать с кошельком, расходами и доходами другого пользователя. Пароли не должны храниться в чистом виде, необходимо хешировать пароль с помощью алгоритма *bcrypt*. Создание и изменение таблиц в базе данных необходимо выполнять через миграции. Все данные, которые приходят от клиента, необходимо валидировать.