

C-MONEY КРИПТОБИРЖА

Удобная платформа для хранения
счётов, перевода валюты и покупки
криптовалюты



Вступление

Пришло время объединить весь пройденный материал в едином большом задании, где вы сможете понять, как они применяются совместно. Этим заданием и является данная дипломная работа

Выполненная работа поможет вам как специалисту:

- Уверенность в своих силах
- Готовый проект в портфолио
- Опыт разработки приложения, максимально приближенного к реальным проектам





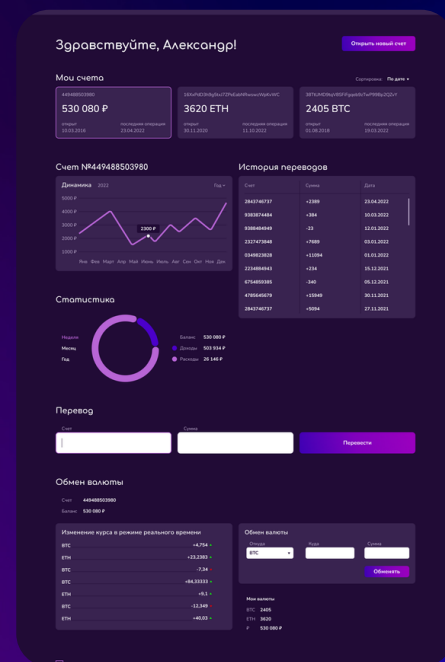
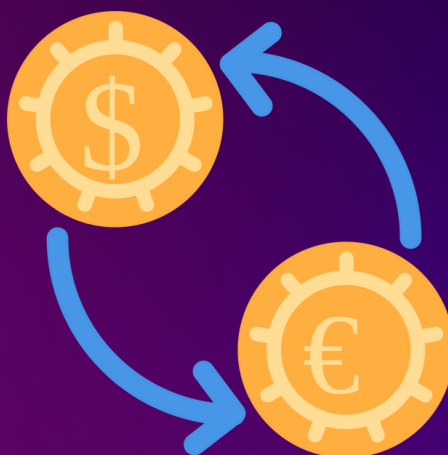
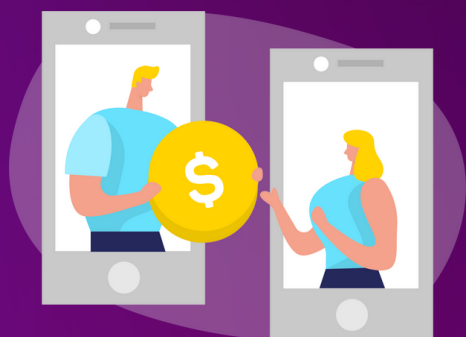
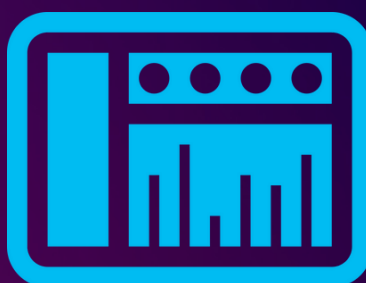
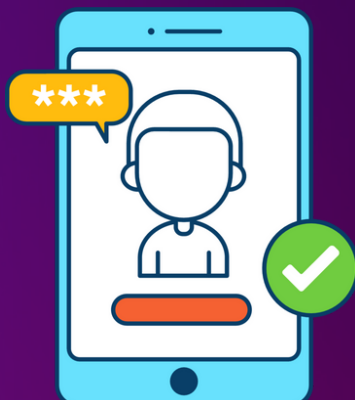
Задание

Необходимо разработать банковскую систему хранения и операций над криптовалютой: средствами и счетами.

Задачи:

1. Авторизация
2. Управление счетами
3. Переводы на другие счета
4. Обмен валюты

Макет



- **Авторизация**

Форма входа пользователя

- **Управление счетами**

Список счетов пользователя

Просмотр информации о существующей карте

Подробная история баланса по карте

- **Переводы на другие счета**

Форма для перевода средств

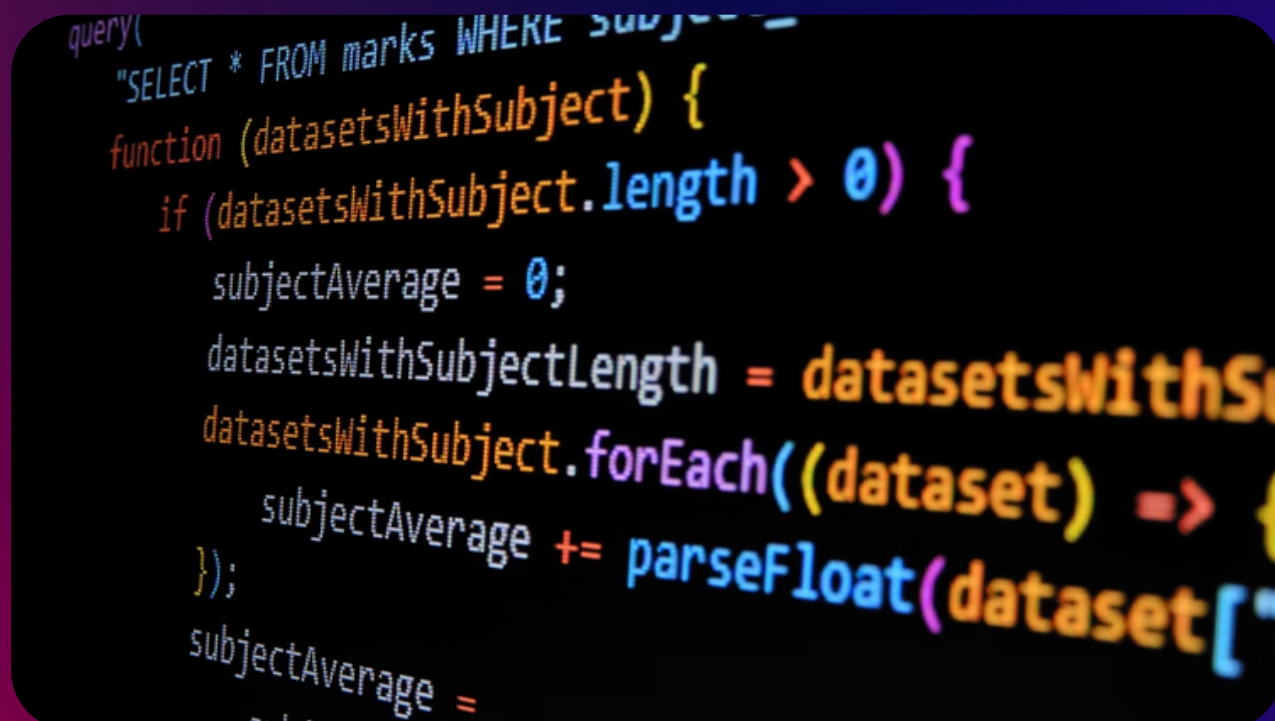
- **Обмен валюты**

Форма обмена валюты

Мониторинг курса валют



Инструкция по работе с API



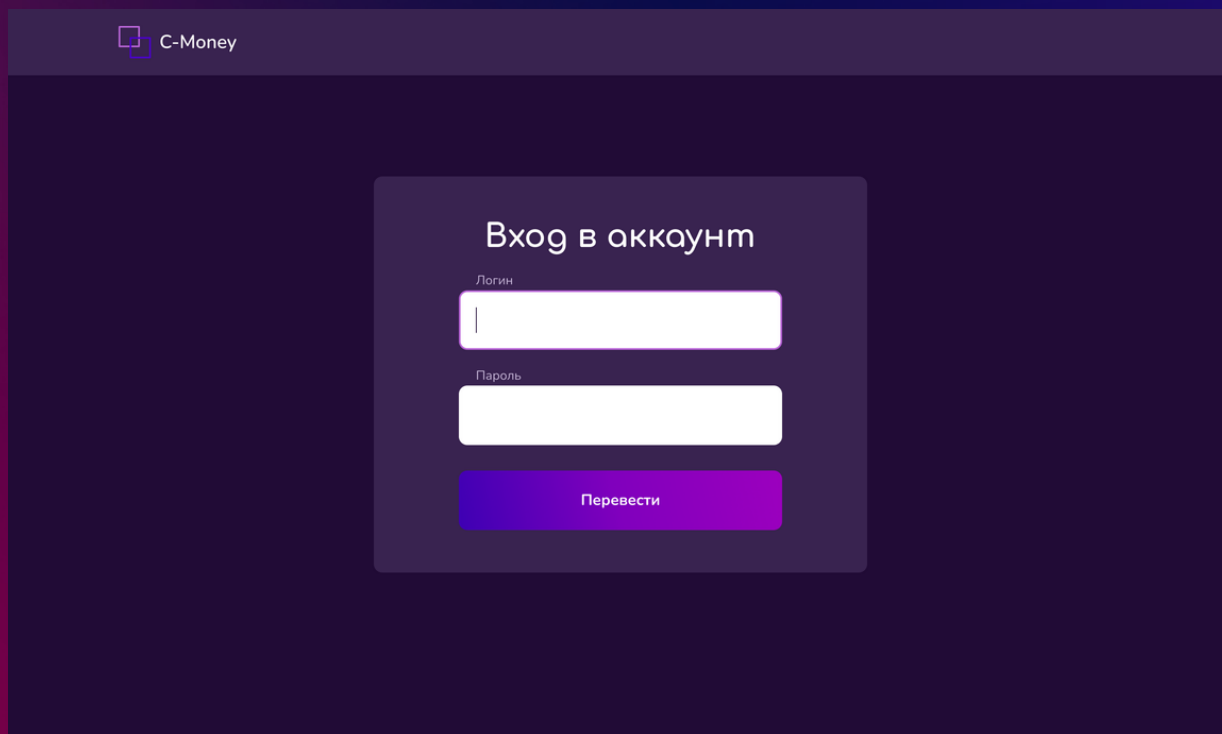
Инструкция по работе с API расположена в файле README.md в репозитории

Адрес репозитория

<https://github.com/maksim-leskin/c-money-api>

Проверяйте всегда, чтобы у вас была актуальная версия API

Форма входа

The image shows a screenshot of a web application's login page. At the top left, there is a logo consisting of two overlapping squares, one light blue and one light green, followed by the text "C-Money". The main content area has a dark blue background. In the center, there is a white rounded rectangle containing the login form. The form has the title "Вход в аккаунт" (Login to account) in bold. Below the title are two input fields: the first is labeled "Логин" (Login) and the second is labeled "Пароль" (Password). Both fields are empty. Below the password field is a red button with the text "Перевести" (Transfer) in white. The overall design is clean and modern.

На экране входа пользователь авторизуется в системе.
Регистрация пользователей через портал осуществляться не будет, их регистрируют в другом месте, по приглашениям.

Необходима валидация на заполненность полей.
Логин и пароль может иметь только латиницу, не менее 6 символов, без пробелов.
Логин начинаться может только с буквы, нельзя вводить числа и спецсимволы

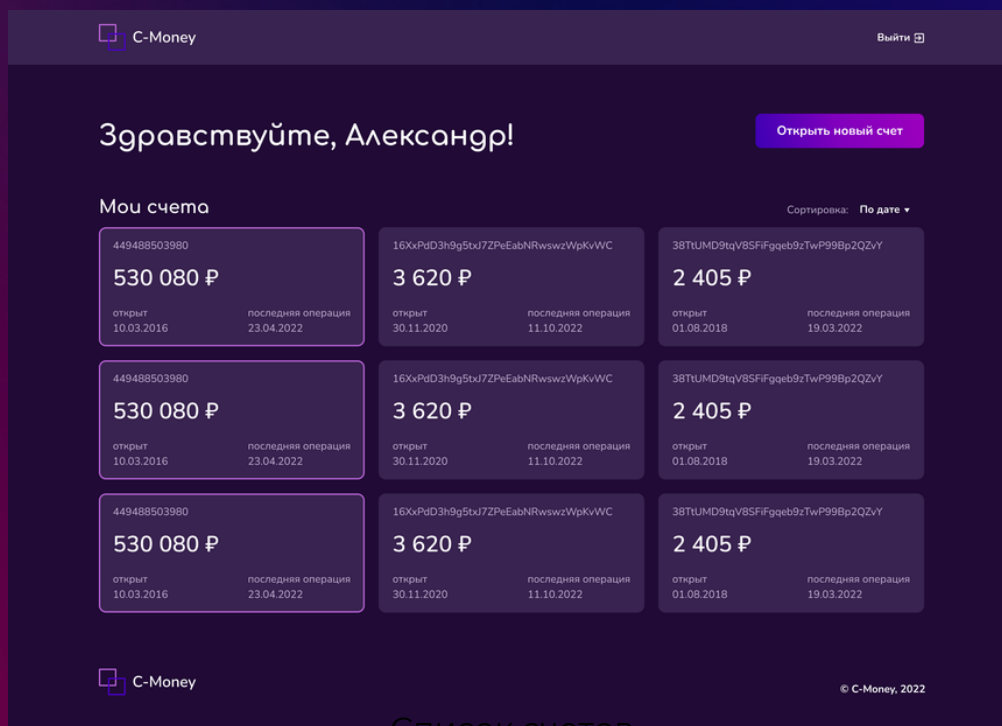
Это необходимо отразить в валидации.

После авторизации пользователь попадает в свой личный кабинет.

Если авторизация прошла неуспешно
Об этом необходимо
пользователя уведомить

Список счетов

GET /currencies



Список счетов

Это страница, на которую попадает пользователь после авторизации.

Здесь отображаем список счетов, которыми владеет пользователь.

Все счета на данной странице «рублевые», банк за кадром обслуживает их через виртуальную валюту с использованием блокчейн-технологии, но для пользователей эти счета просто представляются как обычные удобные рублёвые счета.

Выводим баланс данных счетов с постфиксом валюты рубля.

Список счетов

POST /create-account

GET /accounts

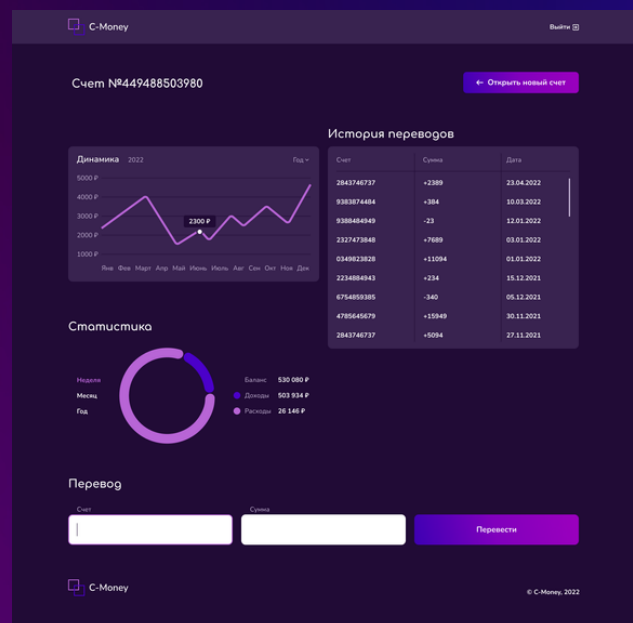
Открыть новый счет

Кнопка добавляет новый счёт данному пользователю, никаких дополнительных настроек не требуется, счёт создаётся автоматически, достаточно лишь дернуть ручку (сделать запрос на бэкенд) и запросить список счетов, где уже будет новый счёт с произвольно присвоенным ему номером.

Необходимо предусмотреть возможность сортировки счетов. Так как предполагается, что пользователю не потребуется большое количество счетов, то все сортировки выводим в порядке возрастания. Необходима сортировка по полям: «Номер счёта», «Баланс», «Дата открытия счета», «Последней транзакции».

При клике на счёт происходит переход на страницу подробного просмотра информации о счёте.

Кнопка «Вернуться» возвращает пользователя на страницу со списком счетов



Информация о счете

GET /account/{id}

На странице имеются следующие элементы:

- График истории баланса
- Список прошлых транзакций
- Статистика (необязательный блок)
- Форма переводов

График истории баланса

График выводит линейную диаграмму истории значений баланса данного счёта за последние 6 месяцев (или меньше, если данных недостаточно), самый актуальный месяц — справа.

Можно использовать любые библиотеки для реализации диаграммы, например:

- vanilla: <https://www.chartjs.org/>
- ReactJS: <https://react-chartjs-2.js.org/>

Список прошлых транзакций

История переводов выводит таблицу из 9 (или менее) записей последних транзакций с участием текущего просматриваемого счёта. Отрицательные суммы (исходящие переводы) выделяем цветом.

Дату форматируем в формате День.Месяц.Год.

Форма переводов

Форма имеет:

Поле для ввода счёта получателя.

Поле ввода суммы переводов.

Кнопка отправки данного перевода.

Предусмотреть валидацию на заполненность полей, и валидацию на ввод положительной суммы.

Обмен валюты

GET /all-currencies

GET /currencies

POST /currency-buy

Websocket /currency-feed

Здесь пользователи могут получить информацию о состоянии своих валютных счетов и быть в курсе последних колебаний курса и обменять одну валюту на другую

Изменение курса в режиме реального времени. Валютные пары выводятся через слэш в качестве разделителя. Около цифры значения изменения курса отображается стрелка направленная вверх или вниз в зависимости от текущего движения курса. Положительное и отрицательное движение подсвечивается зелёным и красным цветом стрелки соответственно.

Список всех известных валют получаем из API. У пользователя уже открыты счета во всех известных валютах, но отображаем валютные счета с положительным балансом. Предусмотреть валидацию на запрет перевода отрицательной суммы и обработать ошибки, например о недостатке средств

Обмен валюты

Откуда	Куда	Сумма
<input type="text" value="BTC"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Обменять"/>		



Технические требования

Выполненную работу необходимо предоставить в виде отдельного репозитория

Репозиторий должен включать в себя необходимые ресурсы для приложения и файл README с инструкциями по запуску сборки приложения. Результатом сборки будет папка `dist` (должна создаваться или очищаться автоматически в начале процесса выполнения сборки), в которой находится страница `index.html` в качестве точки входа в приложения, а также все ресурсы, необходимые приложению для работы.

При долгой загрузке каких-либо данных нужно отображать какую-либо визуальную индикацию загрузки данных.



Дополнительные задания

Чем больше дополнительных заданий вы выполните тем больше работа привлечет работодателя

Реализуйте end-to-end тесты на базовый функционал приложения.

Базовым функционалом приложения считаем: возможность авторизоваться, возможность просмотреть список счетов, возможность перевести сумму со счёта на счёт, возможность создать новый счёт и перевести с него сумму тоже.

Опишите в README инструкцию по их запуску и проверке результатов прохождения этих тестов, чтобы можно было в любое время убедиться в целостности и работоспособности проекта.

Кэширование данных

Данные, которые редко меняются в пределах сессии, можно запоминать и не перезапрашивать с сервера так часто.

Также это хорошо улучшает опыт пользователя, создавая впечатление того, что сайт загружает данные моментально.

Также развитием этой техники является техника фоновой подгрузки данных. Сначала вы отображаете данные из кэша, тем самым создавая у пользователя иллюзию моментальной загрузки страницы, в то время как фоном делаете запрос на обновление этих данных. Когда обновлённые данные вернулись, вы подменяете отображаемые данные на новые.



C-Money

Все ссылки

Макет

[https://www.figma.com/file/Rq9R6bJEqpOeIIsI9LIDM0/C-Money-app-\(Final-project\)](https://www.figma.com/file/Rq9R6bJEqpOeIIsI9LIDM0/C-Money-app-(Final-project))

Адрес API

<https://github.com/maksim-leskin/c-money-api>