The background is a dark blue gradient. On the left, there is a large, semi-transparent circular image of a circuit board. Overlaid on the top left are two overlapping triangles, one blue and one light green. In the top right corner, there is a 3D perspective view of a circuit board's traces.

Дипломная работа

По дисциплине:
WEB - разработка на Python /
Django

Студент:
Блощенко Вячеслав



СОДЕРЖАНИЕ

- Введение
- Цель и Задачи
- Технологический стек
- Структура базы данных
- Функционал
- Сложности
- Заключение



Введение

Сегодня я представлю вам результаты своей дипломной работы, посвященной созданию веб-приложения для банка с безопасной авторизацией через JWT токен и множеством функциональных возможностей веб-приложения банка с акцентом на безопасные финансовые операции и современные инструменты управления счетами.



Цель и Задачи

Цель моей дипломной работы заключалась в создании современного веб-приложения банка, обеспечивающего пользователям безопасность финансовых операций предоставляющего возможности управления счетами. С возможностью масштабирования для подключения различных модулей приложения.



1

Безопасная авторизация: Реализовать механизм безопасной авторизации через JWT токен для защиты учетных записей пользователей

2

Виртуальная карта: Разработать функционал выпуска виртуальной карты, предоставляя пользователям удобный инструмент для проведения онлайн-транзакций.

3

Переводы между картам: Внедрить возможность перевода средств между виртуальными картами внутри и вне банка обеспечивая легкость и быстроту операций.



4

Актуальный курс валют: Подключить внешние источники данных для получения актуальных курсов валют и регулярно обновлять эту информацию чтобы пользователи могли проводить конвертацию средств по текущему обменному курсу.

5

Атомарные транзакции: Применить атомарные транзакции в процессе выполнения транзакций гарантируя, что весь процесс выполняется целиком и успешно, либо не выполняется вовсе, предотвращая возможные проблемы с целостностью данных.

6

Отслеживание транзакций: Реализовать механизмы отслеживания транзакций для просмотра их пользователем.



7

Разработка модульной структуры: Выделение ключевых функциональных блоков приложения в отдельные модули для повышения читаемости и облегчения сопровождения кода.

8

Масштабируемость: Обеспечение гибкости в добавлении новых функциональных возможностей. Реализация интерфейсов и API, которые легко расширяются для внедрения новых сервисов и функций.

9

Управление доступом: Обеспечение возможности восстановления учетных записей пользователей в случае утери доступа. Выдачу конфиденциальных данных только при авторизации пользователя



Технологический стек

1

Django: веб-фреймворк, который позволяет создавать безопасные и поддерживаемые веб-сайты.

2

Django REST Framework: набор инструментов для создания REST API с помощью Django.

3

JSON Web Token (JWT): это открытый стандарт для создания токенов доступа, основанный на формате JSON.



4

PostgreSQL: предоставляет надежное и масштабируемое решение для хранения данных, обеспечивая высокую степень целостности и надежности данных

5

Requests: Интеграция библиотеки Requests для обеспечения удобства выполнения HTTP-запросов с целью взаимодействия с внешними сервисами или API.

6

Celery: для создания асинхронных задач, что позволяет выполнять операции в фоновом режиме, обеспечивая более эффективную обработку длительных задач.



7

Redis: в качестве брокера сообщений для Celery, что обеспечивает эффективную передачу и управление задачами в распределенной системе.

8

CORS: для обеспечения безопасного обмена ресурсами между веб-приложением и внешними источниками данных, разрешая или ограничивая доступ к ресурсам.

9

Middleware: связующее программное обеспечение, которое помогает приложению и серверу обмениваться друг с другом запросами. Для сохранения данных в cookie и их последующего удаления

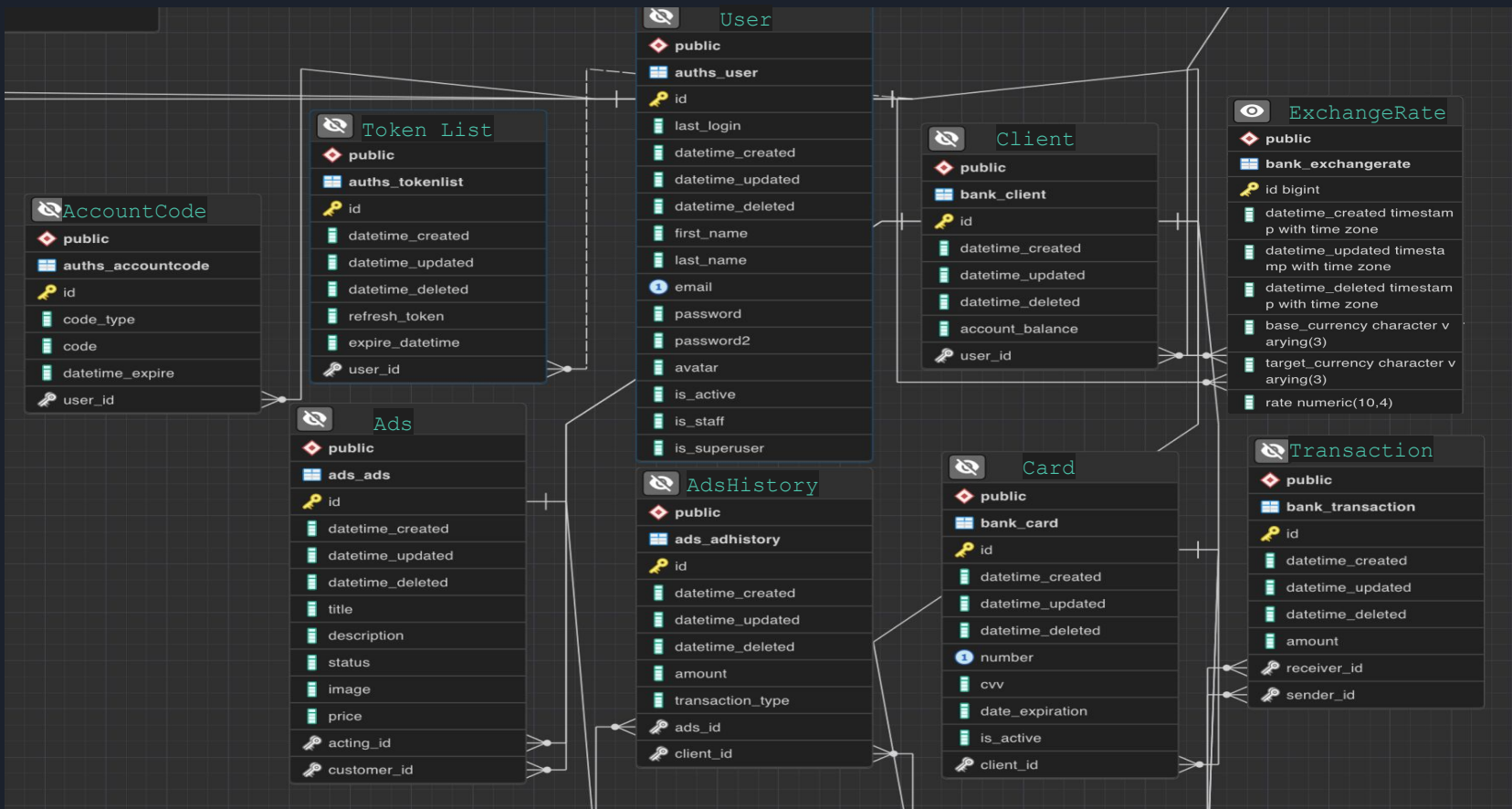


10

Axios: Интеграция библиотеки Axios во фронтенд для удобного выполнения асинхронных HTTP-запросов к серверу, обеспечивая плавную и отзывчивую работу интерфейса.

11

Animejs: Использование Anime.js для добавления визуальных эффектов и анимаций, улучшая пользовательский интерфейс и создавая более привлекательный внешний вид.





Функционал

- 1 **Регистрация:** Пользователи могут создавать свои учетные записи, предоставляя необходимую информацию для полноценного использования всех функций приложения.
- 2 **Активация аккаунта:** После завершения процесса регистрации, каждому пользователю отправляется уникальный код подтверждения на электронную почту. Для активации аккаунта пользователь должен ввести этот код, обеспечивая дополнительный уровень безопасности.
- 3 **Авторизация:** Зарегистрированные пользователи могут входить в систему, используя свои учетные данные, обеспечивая персонализированный и безопасный доступ к функциональности приложения.



4

Восстановление пароля: Пользователь получает на свою электронную почту уникальный код для восстановления пароля, который необходимо ввести в специальное поле приложения для подтверждения своей личности.

5

Создание карт: Пользователи имеют возможность создавать виртуальные карты, которые предоставляют удобный инструмент для проведения различных операций в системе.

6

Перевод денег: Пользователи могут осуществлять переводы средств между своими виртуальными картами, обеспечивая быструю и удобную передачу средств.



7

Пополнение баланса: Пользователи могут пополнять свой баланс, выбирая различные способы для пополнения, обеспечивая гибкость и удобство использования.

8

Просмотр истории транзакций: Пользователи имеют доступ к полному списку своих финансовых транзакций.

9

Курс и конвертация валют: Получение актуального курса валют через внешний источник и встроенная функциональность конвертации валют по актуальному курсу.

10

Доска объявлений: Функционал масштабируемости приложения, предоставляющую возможность для взаимодействия между заказчиком и исполнителем, процесс предоставления и выполнения услуг.



Сложности

В процессе создания моей дипломной работы, я столкнулся с несколькими сложностями, которые требовали дополнительных усилий и поиска их решения:

Внедрение middleware в проект связанный с безопасностью, такой как обработка JWT токенов, связанными с аутентификацией, авторизацией, сохранением и удалением токена обновления в cookie

Работа с финансовыми транзакциями требовала внимания к деталям и строгого соблюдения норм безопасности. Это включало в себя защиту от мошенничества и обеспечение конфиденциальности финансовых данных пользователей.

Внедрение библиотеки Axios обеспечение надежной обработки ошибок и различных статусов HTTP, управление безопасностью передачи токенов, интеграцию с асинхронным кодом и управление таймингами, обеспечения безопасности и эффективности запросов к API.



Заключение

В результате разработки и реализации моей дипломной работы, были успешно достигнуты поставленные цели и успешно преодолены различные технические сложности, для выполнения поставленных функциональных возможностей моего проекта.

Также, были учтены аспекты масштабируемости, производительности, и модульности приложения для обеспечения его готовности к дальнейшему развитию и внедрению новых функциональных возможностей.



Спасибо за внимание!

На этом презентация моей дипломной работы окончена.

Большая благодарность моим преподавателям:

Табашнюк Евгений

Лехман Андрей

Маратов Мад

Контакты:

Номер телефона: +7(705)-251-01-91

Телеграмм: @Sterben2360

