

Краткое описание реализованного функционала

В рамках мини-MVP был разработан веб-сервис для редактирования текста с использованием локальной LLM-модели (Ollama). Основные возможности:

1. ИИ-редактор текста

- Пользователь вводит текст и выбирает операцию: перефразирование, исправление или сокращение.
- Обработка выполняется локальной моделью (например, `qwen2.5:3b`) через API Ollama.
- За каждый запрос списывается 1 токен, при ошибке модели токен возвращается пользователю.

2. Регистрация, авторизация и личный кабинет

- Регистрация по email и паролю с безопасным хранением хеша (Argon2).
- Авторизация выполнена через JWT (Bearer Token).
- В личном кабинете отображаются email пользователя, баланс токенов, форма пополнения и интерфейс редактора.

3. Пополнение баланса

- Реализовано тестовое пополнение без реальных платежей.
- Пользователь вводит сумму, конвертация выполняется на основе текущего курса.
- Создаётся запись транзакции со статусом `success`.
- Баланс пользователя увеличивается пропорционально сумме.

4. Админ-панель

- Доступна только пользователям с флагом `is_admin = True`.
- Админ может просматривать:
 - список пользователей,
 - список транзакций,
 - текущие настройки курса токена.
- Реализована возможность изменения валюты и курса: “1 токен = X единиц валюты”.

5. Использованный стек технологий

- **Backend:** FastAPI, SQLAlchemy, SQLite, OAuth2, JWT, Passlib (Argon2)
- **LLM:** локальный запуск модели через Ollama
- **Frontend:** HTML, CSS, JavaScript (подключается через FastAPI)
- **Инструменты:** Makefile, структура модулей, разделение на роутеры

Потенциальные улучшения при наличии дополнительного времени

1. Улучшение пользовательского интерфейса

- Использование UI-фреймворков (Tailwind, Bootstrap).
- Добавление уведомлений, модальных окон, адаптивной верстки.

2. Усиление системы авторизации

- Добавление refresh-токенов.
- Гибкое разграничение ролей (admin / user / moderator).

3. Расширение функционала редактора

- История запросов пользователя.
- Сохранение результатов и экспорт текстов.

4. Улучшение архитектуры проекта

- Перенос конфигураций в `.env`.
- Переход с SQLite на PostgreSQL.

5. Улучшение системы биллинга

- Подключение реальных платежных систем.
- Расширение статусов транзакций.

6. Оптимизация работы LLM

- Настройки модели: температура, длина ответа, system prompt.
- Возможность выбора моделей прямо в интерфейсе.
- Кеширование частых запросов.