o 4. ПРОЕКТ LME – «Локальный мессенджер с открытой архитектурой»

Репозиторий: https://gitflic.ru/project/ivanovga/lme

---

§ 1. Общее описание

LME (Local Messaging Engine) — это полнофункциональный веб-мессенджер, написанный на **FastAPI** и **SQLAlchemy**. Он реализует все основные функции современного обмена сообщениями:

- регистрация и аутентификация пользователей (JWT)
- личные и групповые чаты
- обмен текстом, файлами (изображениями, видео, документами)
- WebSocket-каналы с пингами и офлайн-статусом
- административный интерфейс (список пользователей, удаление, имитация сообщений)
- полностью открытая архитектура: REST-API + WebSocket + клиентское приложение (Tk-inter)

Это демо-ветка закрытого проекта с расширенной архитектурой, предназначена для образовательных, ознакомительных и исследовательских целей.

---

§ 2. Ключевые возможности

Категория	**Функция**	**Кратко**
Аутентификация	Регистрация, вход в систему, сброс пароля	JWT-токены, проверка сложности пароля, автоматическая отправка почты
Сообщения	Текст, файл, системные	Публичные и приватные чаты, история сообщений, статус доставки (отправлено/доставлено/прочитано)
WebSocket	Подключение, пинг/ответ, онлайн-статус	Redis-хранилище для статуса, цикл проверки связи, корректное завершение работы
Панель администратора	Список пользователей, удаление, имитация, просмотр всех сообщений	Веб-интерфейс на Tkinter, REST- эндпоинты
Безопасность	Защита от CSRF, политика CORS, ограничение скорости	

Категория	Функция	Кратко
	(в памяти, с поддержкой Redis), только HTTPS, строгая политика заголовков	
Масштабируемость	Асинхронная БД (PostgreSQL/SQLite), пул соединений, Redis-пул, логирование в файлы + консоль	
Документация	Swagger / ReDoc, auto-генерация OpenAPI, примеры запросов	
Тесты	Unit-/интеграционные/нагрузочные (pytest-asyncio, locust, k6), покрытие > 90 %	

---

§ 3. Технологический стек

Уровень	Библиотека	Функционал
Бэкенд	FastAPI, Pydantic v2, SQLAlchemy 2.x, Asyncpg / SQLite	Быстрый асинхронный REST-API + WebSocket
База данных	PostgreSQL (для продакшена) / SQLite (для разработки)	Хранение пользователей, сообщений, контактов
Кэш/мессенджер	Redis (asyncio)	Онлайн-статусы, ограничение скорости, публикация/подписка для WebSocket
Электронная почта	aiosmtplib, email.mime	Регистрация, сброс пароля
Логирование	loguru	Консоль + файлы, уровень `DEBUG`/`INFO`
Клиент	Tkinter + requests + websockets	Клиент с графическим интерфейсом для демонстрации
CI/CD	GitHub Actions (или GitLab CI), Docker, Docker-Compose	Автоматический тест, линтинг, сборка образа
Документация	OpenAPI, Markdown	Swagger UI, ReDoc

---

§ 4. Установка и запуск

```
# 1. Клонируем репозиторий
git clone https://gitflic.ru/project/ivanovga/lme.git
cd lme
```

```
# 2. Создаем виртуальное окружение
python -m venv .venv
source .venv/bin/activate # Windows: .venv\Scripts\activate

# 3. Устанавливаем зависимости
pip install -r requirements.txt

# 4. Заполняем .env (пример в .env.example)
cp .env.example .env
# EDIT: укажите DATABASE_URL, REDIS_URL, SMTP_* и т.д.

# 5. Запускаем миграции (если используете Alembic)
усовершенствованная головка перегонного куба

# 6. Запускаем сервер
uvicorn app.main: приложение - перезагрузка
```

Важно:

- Для тестовой почты можно воспользоваться сервисом MailHog (порт 1025).
- Должен быть запущен Redis (localhost:6379) — его можно быстро запустить с помощью Docker: `docker run -d -p 6379:6379 redis:7.2`.

§ 5. Контакты

Автор	Должность	Контакт
Лаврова Елена Владимировна	Профессор, доктор технических наук, директор Учебно-научного института информационных технологий	UNIIT-PSTU@yandex.ru
Иванов Григорий Александрович	Ассистент кафедры информационной безопасности	ivanov.g.a@ro.ru

---

§ 6. Планы развития

1. **Микросервисы** — разделение API, WebSocket и клиентского приложения.
2. **Масштабируемость** — Redis-кластер, PostgreSQL-реплика, Kubernetes.
3. **Модульная аутентификация** — OAuth2, SSO, LDAP.
4. **Расширенный поиск** — ElasticSearch для сообщений.
5. **Мобильный клиент** — Flutter / React-Native.

Примечание: Перенесите проект в среду Docker, настройте CI/CD, и вы получите полностью готовый к развертыванию мессенджер, который можно использовать как учебный проект или в качестве прототипа для реальных сервисов.