

# Finance tracker

Проектный практикум

Команда № 8

Приложение для отслеживания расходов

# Команда

**Степанов Егор**  
Тимлид/Аналитик

**Урасинов Даниил**  
Ведущий разработчик

**Исаков Константин**  
Разработчик

**Соколов Аким**  
Разработчик

**Молодцов Павел**  
Тестировщик

Репозиторий команды: <https://github.com/MIPT-PD-PP-1/finance-tracker>

# Описание проекта

Приложение для управления личными финансами, позволяющее пользователям фиксировать доходы и расходы и получать аналитику по ним.

Функции:

- Добавление транзакций;
- Сортировка транзакций по дате и категориям;
- Аналитика для пользователя и группы пользователей;
- Напоминания о регулярных платежах.

# Использованные технологии и инструменты

FastAPI – реализация API.

Pydantic – валидация данных.

SQLAlchemy, PostgreSQL – модели данных и работа с БД.

Alembic – управление миграциями базы данных.

pytest, pytest-asyncio – тестирование.

Docker / Docker Compose, Make – инфраструктура и запуск окружения.

Git / GitHub – контроль версий и фиксация изменений.

# API. Основные эндпоинты

## 🔒 Аутентификации

Эндпоинт	Метод	Описание	Путь
Регистрация	POST	Создать нового пользователя	/api/auth/register
Авторизация	POST	Вход и получение токена	/api/auth/login
Просмотр пользователя	GET	Просмотр текущего пользователя	/api/auth/me
Смена пароля	PUT	Изменить пароль пользователя	/api/auth/change-password
Обновление токена	POST	Обновить access token	/api/auth/refresh-token
Статистика пользователя	GET	Общая статистика и аналитика пользователя с группировкой по категориям и периодам	/api/auth/me/statistics

## 👤 Группы

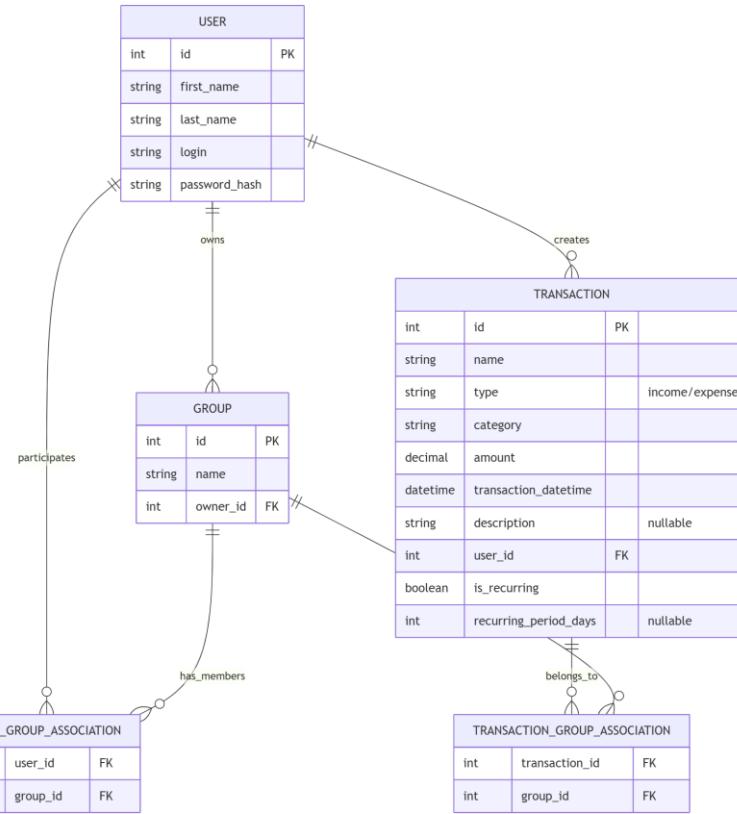
Эндпоинт	Метод	Описание	Путь
Получить список	GET	Список групп пользователя	/api/groups
Создать группу	POST	Создать новую группу	/api/groups
Просмотреть группу	GET	Посмотреть данные группы	/api/groups/{group_id}
Редактировать группу	PUT	Обновить данные группы	/api/groups/{group_id}
Удалить группу	DELETE	Удалить группу	/api/groups/{group_id}
Список пользователей	GET	Список пользователей группы	/api/groups/{group_id}/users
Добавить пользователя	POST	Добавить пользователя в группу	/api/groups/{group_id}/users/{user_id}
Удалить пользователя	DELETE	Удалить пользователя из группы	/api/groups/{group_id}/users/{user_id}
Статистика группы	GET	Общая статистика и аналитика группы с группировкой по категориям и периодам	/api/groups/{group_id}/statistics

# API. Основные эндпоинты

## Транзакции

Эндпоинт	Метод	Описание	Путь
Получить список	GET	Список транзакций пользователя с пагинацией и фильтрами	/api/transactions
Создать транзакцию	POST	Добавить доход или расход	/api/transactions
Предстоящие платежи	GET	Список предстоящих регулярных платежей	/api/transactions/upcoming
Регулярные транзакции	GET	Список всех регулярных транзакций	/api/transactions/recurring
Транзакции группы	GET	Список транзакций группы с пагинацией и фильтрами	/api/transactions/group/{group_id}
Просмотреть транзакцию	GET	Получить транзакцию по ID	/api/transactions/{transaction_id}
Редактировать транзакцию	PUT	Обновить данные транзакции	/api/transactions/{transaction_id}
Удалить транзакцию	DELETE	Удалить транзакцию	/api/transactions/{transaction_id}

# База данных



# Тестирование

Проект покрыт автотестами на pytest.

Тесты проверяют все эндпоинты API: аутентификацию, группы и транзакции.

Что тестируется:

Блок	Покрытие
Аутентификация	Регистрация, логин, /me, смена пароля, refresh токена
Группы	CRUD групп, добавление/удаление участников, аналитика
Транзакции	CRUD транзакций, пагинация, фильтрация, привязка к группам

Каждый эндпоинт проверяется на:

- Успешные сценарии (happy path)
- ✗ Ошибки валидации (422)
- 🔒 Отсутствие авторизации (401/403)
- 🔎 Несуществующие ресурсы (404)
- ⚡ Доступ к чужим данным (403)

# Результаты

- ★ Создан backend приложения с использованием FastAPI, PostgreSQL и Docker.
- ★ Реализованы основные функции приложения.
- ★ Реализованы API для пользователей, групп пользователей и транзакций.
- ★ Настроена система миграций базы данных с помощью Alembic.
- ★ Код покрыт автотестами на pytest.

Спасибо  
за внимание!

