**Практическое занятие 1: Выбор и проработка идеи проекта**

**Цель:**  
Определить тему проекта, сформулировать требования и спроектировать базовую архитектуру.

**Задание:**

1. Разделиться на команды (или работать индивидуально).

Команда:

- Никита

- Иван

- Егор

- Димас

1. Выбрать идею проекта (можно свою или из предложенных):
   * API для трекинга личных финансов и т.д
2. Описать концепцию проекта (1-2 страницы):
   * Проблема, которую решает проект

Большинство людей не отслеживают свои расходы, потому что существующие приложения либо слишком перегружены функциями, либо не позволяют гибко настраивать под себя.

Для разработчиков и энтузиастов также не хватает простой, понятной API, которую можно встроить в свои инструменты, чат-боты, телеграм-каналы, таблицы и т.п.

Проект решает задачу простого учета личных финансов через открытый и легко интегрируемый интерфейс.

* + Ключевые функции
* Добавление доходов и расходов
* Список всех трак
* Просмотр подробной информации по трате
* Редактирование траты
* Удаление траны
  + Основные пользователи

Люди, которым нужно легко отслеживать свои траты - Хотят просто записывать расходы и доходы без сложных интерфейсов

Разработчики - Хотят интегрировать трекинг финансов в свои проекты: боты, Telegram-каналы, дашборды

1. Разработать high-level архитектуру:
   * Какие сервисы и компоненты будут использоваться?

**API-сервер**

* Обрабатывает входящие HTTP-запросы.
* Реализует CRUD по транзакциям.
* Валидация и маршрутизация запросов.
* REST-интерфейс (или gRPC в будущем, но REST на MVP).

**База данных**

* Хранит пользователей, транзакции, категории.
* Простая реляционная структура.

**Админ-интерфейс / Swagger UI**

* Для тестирования и демонстрации API.
* Swagger/OpenAPI документация.
  + Будет ли это монолит или микросервисная архитектура?

Был выбран монолит, так как:

* быстрее разрабатывать;
* проще развертывать;
* меньше DevOps-накладных расходов;
* нет необходимости в сложной оркестрации (как в микросервисах).

При этом архитектура будет модульной, что упростит миграцию в микросервисы при росте.

* + Какие технологии планируется использовать (БД, кеш, брокеры сообщений и т. д.)?

|  |  |
| --- | --- |
| **Компонент** | **Технология** |
| Язык | Go |
| Фреймворк | chi |
| ORM | - |
| API-документация | Swagger |
| Аутентификация | JWT |
| CI/CD | GitHub Actions / Docker Hub |
| Хостинг | Yandex Cloud |

1. Защитить проект (презентация на 5-10 минут).