**First Vape**  
**C#**  
**Демидов Иван Григорьевич, ИСИПу-216**  
**11.03.2025**

### Цель проекта

Разработка API-сервиса для управления ассортиментом вейпов, включающего:

* Возможность создания, редактирования, удаления и просмотра информации о вейпах с помощью HTTP-запросов (POST, PUT, DELETE, GET).
* Использование DTO-классов для передачи данных, исключая утечку лишней информации.
* Генерация документации API через Swagger для удобного взаимодействия с клиентами и тестирования.

### Задачи

#### 1. Теоретическая часть

**1.1. Анализ архитектуры**

* Выбор платформы: ASP.NET Core.
* Использование минимального API для лаконичной структуры кода.
* Внедрение SQLite в качестве СУБД.
* Интеграция Swagger для автоматической генерации документации API.

**1.2. Создание архитектуры приложения**

* Настройка сервисов:
  + AddDbContext — подключение базы данных.
  + AddEndpointsApiExplorer, AddSwaggerGen — подготовка к Swagger-документации.
* Настройка middleware:
  + UseSwagger()
  + UseSwaggerUI()
* Определение модели Vape с полями:
  + Id (int) — уникальный идентификатор.
  + Name (string) — название.
  + Laste (string) — описание.
  + Price (double) — цена.
  + CreatedAt (DateTime) — дата создания.

**1.3. Работа с маршрутизацией и документацией**

* Создание следующих маршрутов:
  + POST /vape
  + PUT /vape/{id}
  + DELETE /vape/{id}
  + GET /vape
* Swagger позволяет наглядно тестировать каждый endpoint и видеть структуру запросов и ответов.

**1.4. Бизнес-анализ**

* Целевая аудитория: магазины и сервисы, работающие с электронными устройствами. Дети от 9 месяцев
* Проблема: необходимость в лёгкой системе учёта и управления ассортиментом.
* Решение: веб-API, интегрируемое в любую платформу.

**1.5. Системный анализ**

* Проверка ошибок:
  + Ответ 404 Not Found при несуществующем ID.
  + Игнорирование null значений в DTO при обновлении.
* Возможность подключения к другим сервисам или расширения функционала (например, логирование, аутентификация).

#### 2. Практическая часть

**2.1. Добавление вейпа**

* Метод: POST /vape
* Используемый DTO: CreateVapeDTO
* Пример:

csharp

КопироватьРедактировать

new Vape(dto.Name ?? "Неизвестный вейп", dto.Laste ?? "Описание не указано", dto.Price ?? 0)

* Валидация значений и установка значений по умолчанию.

**2.2. Обновление данных**

* Метод: PUT /vape/{id}
* Используемый DTO: UpdateVapeDTO
* Принцип: только те поля, что были указаны, будут обновлены.
* Обработка случая, если вейп не найден:

csharp

КопироватьРедактировать

return Results.NotFound("Вейп не найден");

**2.3. Удаление вейпа**

* Метод: DELETE /vape/{id}
* Проверка наличия объекта в БД, корректная обработка NotFound.
* Удаление не влияет на другие записи.

**2.4. Получение списка вейпов**

* Метод: GET /vape
* Возвращает список всех доступных вейпов.
* Возможное расширение: фильтрация по цене, названию или сортировка по дате.