Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное   
образовательное учреждение высшего образования   
«Самарский национальный исследовательский университет   
имени академика С. П. Королева»

Институт информатики и кибернетики

Кафедра технической кибернетики

**Технологии сетевого программирования**

**Отчет по лабораторной работе 0**

**«Проектирование приложения»**

Выполнили:

Ломов В.Б.,

Ксенофонтов В.А.

группа 6304-010302D

Самара, 2025

**Онлайн-бармен**

**1. Концепция приложения.**

Приложение будет помогать пользователям создавать коктейли как на основе только имеющихся ингредиентов, так и ингредиентов, которые необходимо приобрести.

**2. Схема БД.**

 **ADMINS** (администраторы)

* id (SERIAL, PK) – уникальный идентификатор администратора
* login (VARCHAR(50), UNIQUE, NOT NULL) – имя администратора
* password (TEXT, NOT NULL) – пароль

 **USERS** (пользователи)

* id (SERIAL, PK) – уникальный идентификатор пользователя
* login (VARCHAR(50), UNIQUE, NOT NULL) – имя пользователя
* password (TEXT, NOT NULL) – пароль

 **USER\_INGREDIENTS** (ингредиенты пользователя)

* id (SERIAL, PK) – уникальный идентификатор
* user\_id (INTEGER, FK → USERS.id, ON DELETE CASCADE) – связь с пользователем
* ingredient\_id (INTEGER, FK → INGREDIENTS.id, ON DELETE CASCADE) – связь с ингредиентом

 **INGREDIENTS** (ингредиенты)

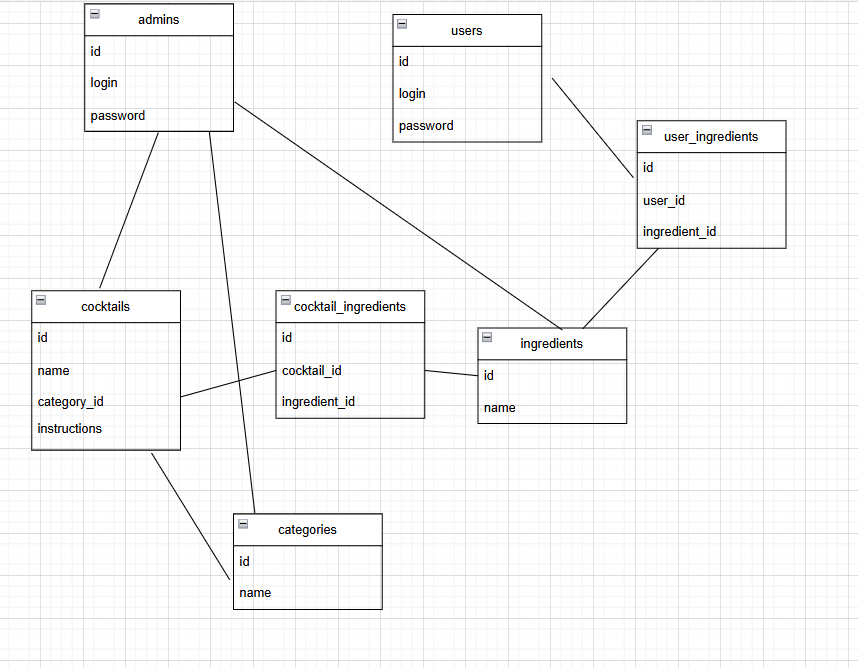
* id (SERIAL, PK) – уникальный идентификатор ингредиента
* name (VARCHAR(100), UNIQUE, NOT NULL) – название ингредиента

 **COCKTAILS** (коктейли)

* id (SERIAL, PK) – уникальный идентификатор коктейля
* name (VARCHAR(100), UNIQUE, NOT NULL) – название коктейля
* category\_id (INTEGER, FK → CATEGORIES.id, NOT NULL) – категория коктейля
* instructions (TEXT) – рецепт приготовления
* ingredient\_id (INTEGER, FK → INGREDIENTS.id, ON DELETE CASCADE) – ингредиент

 **CATEGORIES** (категории коктейлей)

* id (SERIAL, PK) – уникальный идентификатор
* name (VARCHAR(50), UNIQUE, NOT NULL) – название категории



**Схема отношений:**

* ADMINS (1) → (∞) INGREDIENTS
* ADMINS (1) → (∞) CATEGORIES
* ADMINS (1) → (∞) COCKTAILS
* USERS (1) → (∞) USER\_INGREDIENTS (∞) ← (1) INGREDIENTS
* COCKTAILS (∞) → (1) CATEGORIES
* COCKTAILS (∞) ↔ (∞) INGREDIENTS (через COCKTAIL\_INGREDIENTS)

**3. API схема.**

**1. Аутентификация**

* **POST** /api/auth/register/ – Регистрация нового пользователя
* **POST** /api/auth/login/ – Вход пользователя
* **POST** /api/auth/logout/ – Выход пользователя

**2. Ингредиенты**

* **GET** /api/ingredients/ – Получить список всех ингредиентов

**3. Ингредиенты пользователя**

* **GET** /api/user-ingredients/ – Получить список ингредиентов текущего пользователя
* **POST** /api/user-ingredients/ – Добавить новый ингредиент
* **DELETE** /api/user-ingredients/ – Удалить игредиент

**4. Коктейли**

* **GET** /api/cocktails/ – Получить список всех коктейлей
* **GET** /api/cocktails/{id}/ – Получить информацию о конкретном коктейле
* **POST** /api/cocktails/ – Добавить новый коктейль (для администраторов)
* **DELETE** /api/cocktails/{id}/ – Удалить коктейль (для администраторов)

**5. Поиск коктейлей по ингредиентам**

* **GET** /api/cocktails/by-ingredients/ – Получить коктейли, которые можно приготовить из имеющихся ингредиентов пользователя
* **GET** /api/cocktails/missing-ingredients/ – Получить коктейли, для которых не хватает ингредиентов, с указанием недостающих продуктов

**6. Категории коктейлей**

* **GET** /api/categories/ – Получить список всех категорий
* **GET** /api/categories/{id}/ – Получить информацию о конкретной категории
* **POST** /api/categories/ – Добавить новую категорию (для администраторов)
* **DELETE** /api/categories/{id}/ – Удалить категорию (для администраторов)

**4**. **Используемые технологии**

Front: HTML + CSS, Django AlpineJS

Back: python, django

БД: PostgresSQL