МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

**ОТЧЕТ**

по лабораторной работе №0

**Проектирование приложения**

Выполнили студенты

группы 6304-010302D

Семенов Илья

Бочкарев Данил

Самара 2025

**Предметная область** - Сайт для подбора персонализированных рационов питания

**Цель проекта** – создать платформу, которая помогает пользователям составлять оптимальные рационы питания на основе их индивидуальных характеристик, предпочтений и бюджета.

**Бизнес цель:**

* Помочь пользователям легко находить подходящие блюда и рационы
* Учитывать особые диетические потребности и ограничения
* Предоставлять подробную информацию о питательной ценности продуктов
* Создать удобный интерфейс для планирования питания

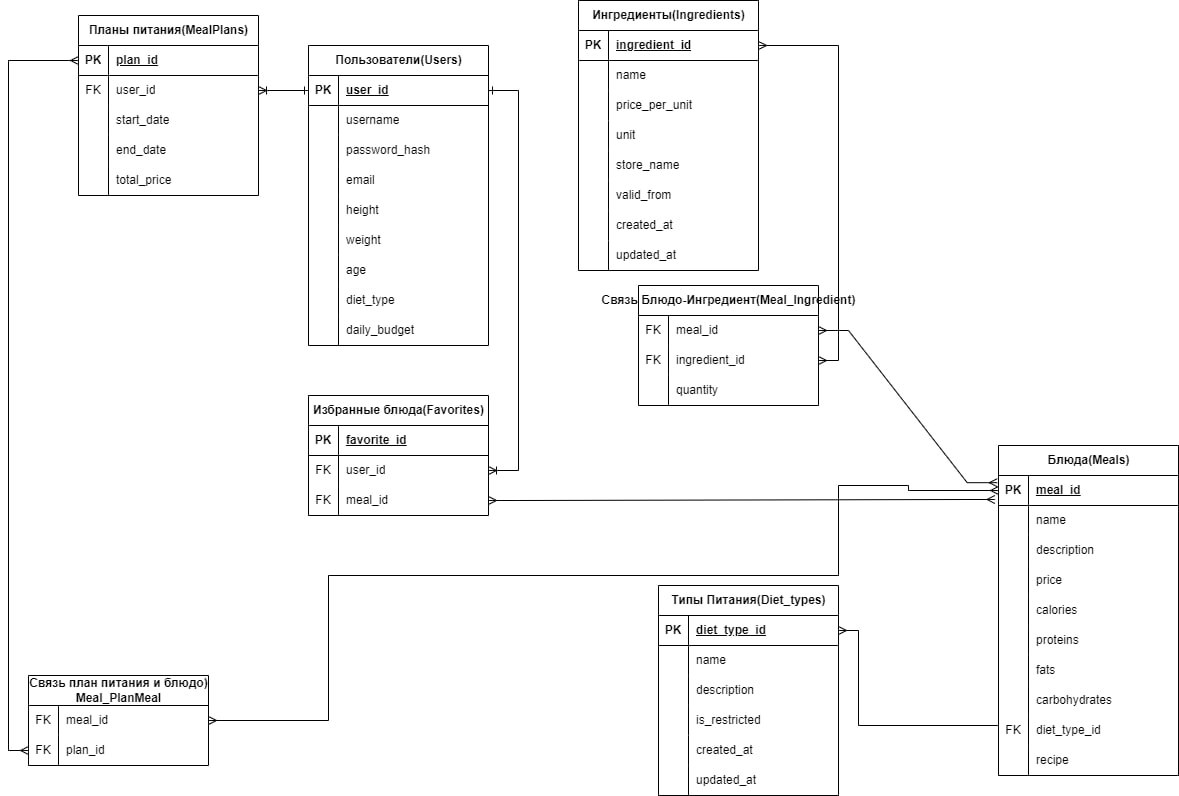
**Функционал:**

* Регистрация/Авторизация пользователей
* Ввод персональных данных (рост, вес, возраст, тип питания)
* Установка бюджета
* Поиск и фильтрация блюд по параметрам
* Просмотр детальной информации о блюдах
* Создание персональных планов питания
* Сохранение избранных рецептов

**Примеры сценариев использования:**

1. **Подбор рациона:**
   1. Пользователь заходит на сайт и заполняет анкету с персональными данными
   2. Выбирает тип питания и устанавливает дневной бюджет
   3. Получает список подходящих блюд
   4. Может просматривать детальную информацию о каждом блюде
2. **Управление учетной записью:**
   1. Пользователь может обновлять свои данные
   2. Сохранять избранные рецепты
   3. Просматривать историю своих планов питания

**Логическая схема базы данных**



**Выбор сервера приложений и СУБД**

- Операционная система: Windows

- СУБД: PostgreSQL

**Список используемых технологий**

**Backend:**

- Python - основной язык программирования

- FastAPI - асинхронный фреймворк для создания высокоэффективных API

- OAuth2 - для реализации системы аутентификации и авторизации

- Docker - для контейнеризации приложения

- Django (частично) - может использоваться для административной части или как дополнение к FastAPI

**Frontend:**

**-** HTML + CSS - базовые технологии для создания пользовательского интерфейса

- Jinja2 – инструмент для html-шаблонизации

**Дополнительные инструменты:**

**-** PostgreSQL - реляционная база данных для хранения информации о пользователях, блюдах и планах питания

- GitHub - для версионирования кода и совместной разработки

**Схема взаимодействия компонентов**

1**. Клиентская часть (Frontend)**

**Описание:**

* Пользователь взаимодействует с веб-интерфейсом через браузер.
* Для динамического обновления данных используются HTTP-запросы к API.

2. **Backend (FastAPI)**

**Описание:**

* FastAPI выступает в роли серверного приложения, которое обрабатывает запросы пользователей.
* OAuth2 реализует аутентификацию и авторизацию пользователей.

3. **База данных (PostgreSQL)**

**Технологии:** PostgreSQL

**Описание:**

* PostgreSQL хранит все данные о пользователях, блюдах, планах питания и других сущностях.

4. **Контейнеризация (Docker)**

**Технологии:** Docker

**Описание:**

* Приложение разворачивается в контейнерах Docker для обеспечения одинаковой среды разработки и деплоя.
* Docker Compose описывает взаимосвязь между сервисами (приложение, база данных, кеш и т.д.).

5. **Версионирование кода (GitHub)**

**Технологии:** GitHub

**Описание:**

* GitHub используется для хранения исходного кода и управления версиями.

**API (OpenAPI/Swagger)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| /auth/register | POST | Регистрация пользователя | {“username”: “user1”, “password”: “password123”, “email”: “user@example.com”} | {“message”: “Регистрация успешна”} |
| /auth/login | POST | Авторизация пользователя | {“username”: “user1”, “password”: “password123”} | {“access\_token”: “jwt\_token\_here”, “token\_type”: “bearer”} |
| /user/profile | GET | Получить профиль пользователя |  | {  “user\_id”: 1,  “username”: “user1”,  “height”: 180,  “weight”: 75,  “age”: 30,  “diet\_type”: “sports”,  “daily\_budget”: 500  } |
| /user/profile | PUT | Обновить профиль пользователя | {  “height”: 185,  “weight”: 80,  “age”: 31,  “diet\_type”: “vegetarian”,  “daily\_budget”: 600  } | {“message”: “Профиль успешно обновлен”} |
| /meals | GET | Получить список блюд |  | [  {  “meal\_id”: 1,  “name”: “Куриная грудка с овощами”,  “description”: “Запеченная куриная грудка с овощами”,  “price”: 150,  “calories”: 300,  “proteins”: 30,  “fats”: 5,  “carbohydrates”: 10  },  ...  ] |
| /meals/{meal\_id} | GET | Получить детальную информацию о блюде |  | {  “meal\_id”: 1,  “name”: “Куриная грудка с овощами”,  “description”: “Запеченная куриная грудка с овощами”,  “ingredients”: [  {“name”: “куриная грудка”, “quantity”: “200г”, “price”: 80},  {“name”: “брокколи”, “quantity”: “100г”, “price”: 30},  {“name”: “морковь”, “quantity”: “50г”, “price”: 10}  ],  “recipe”: “1. Разогрейте духовку до 200°C\n2. Приготовьте ингредиенты...\n”,  “calories”: 300,  “proteins”: 30,  “fats”: 5,  “carbohydrates”: 10  } |
| /plans | POST | Создать план питания | {  “meals”: [1, 2, 3],  “duration”: “1 week”,  “total\_price”: 1000  } | {  “plan\_id”: 1,  “message”: “План создан успешно”  } |
| /favorites | POST | Добавить блюдо в избранное | {  “meal\_id”: 1  } | {  “message”: “Блюдо добавлено в избранное”  } |
| /favorites/{meal\_id} | DELETE | Удалить блюдо из избранного |  | {  “message”: “Блюдо удалено из избранного”  } |
| /search | GET | Поиск блюд по параметрам | {  “diet\_type”: “vegetarian”,  “min\_price”: 100,  “max\_price”: 300,  “calories”: 500  } | [  {  “meal\_id”: 2,  “name”: “Овощной салат”,  “description”: “Свежий салат из сезонных овощей”,  “price”: 120,  “calories”: 200,  “proteins”: 5,  “fats”: 10,  “carbohydrates”: 15  },  ...  ] |
| /history | GET | Получить историю планов питания |  | [  {  “plan\_id”: 1,  “created\_at”: “2023-10-01T12:00:00Z”,  “meals”: [1, 2, 3],  “total\_price”: 1000  },  ...  ] |

[Ссылка на GitHub](https://github.com/Obabab/Aboltus)