Министерство науки и высшего образования Российской Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»

Образовательная программа «Корпоративные информационные системы»

Отчет по курсовому проекту

Тема: «Дневник питания»

**Выполнил:**

Студент группы 191-361

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Каменских С.С.

подпись, дата

Москва 2022

**ВВЕДЕНИЕ**

В рамках данного курсового проекта было разработано web-приложение «Дневник питания» для подсчета калорий и контроля питания.

Одной из самых актуальных проблем современного общества является проблема здорового питания. Полноценное питание предусматривает потребление необходимого количества белков, жиров, углеводов, витаминов, макро и микроэлементов для нормального функционирования организма. Физическое здоровье человека, состояние иммунной системы и многое другое – все это напрямую связано с проблемой здорового питания. С текущим темпом жизни все большее количество людей сталкиваются с проблемой питания.

Целью данной курсовой работы является разработка многостраничного сайта, который представляет из себя структурированную систему, предоставляющую пользователям полный функционал необходимый для контроля потребляемой пищи.

В соответствии с поставленной целью в курсовой работе решаются следующие задачи:

1. Разработка web-приложения с использованием системы контроля версий git.
2. Разработка Rest API.
3. Описать особенности функционирования и провести тестирование разработанного сайта.

**АРХИТЕКТУРА ПРОЕКТА**

**Выбор технологий и инструментов**

Для разработки веб-сайта были использованы следующие программные средства: фреймворк Vue.js, JavaScript, фреймворк Spring, Java, PostgresSQL, Postman.

В проекте реализована трехзвенная архитектура, состоящая из:

1. Frontend – отправление запросов.
2. Backend – основные вычисления, обработка данных, запросов.
3. БД – хранилище данных.

**Frontend**

Frontend реализован с помощью фреймворка Vue.js. Были подключены библиотеки Vue Router и axios для осуществления маршрутизации между страницами и получения данных с помощью GET, POST, DELETE запросов.

Проект состоит из 4 основных страниц. Вся маршрутизация между ними прописывается в файле router/index.js, представленном на рисунке 1.

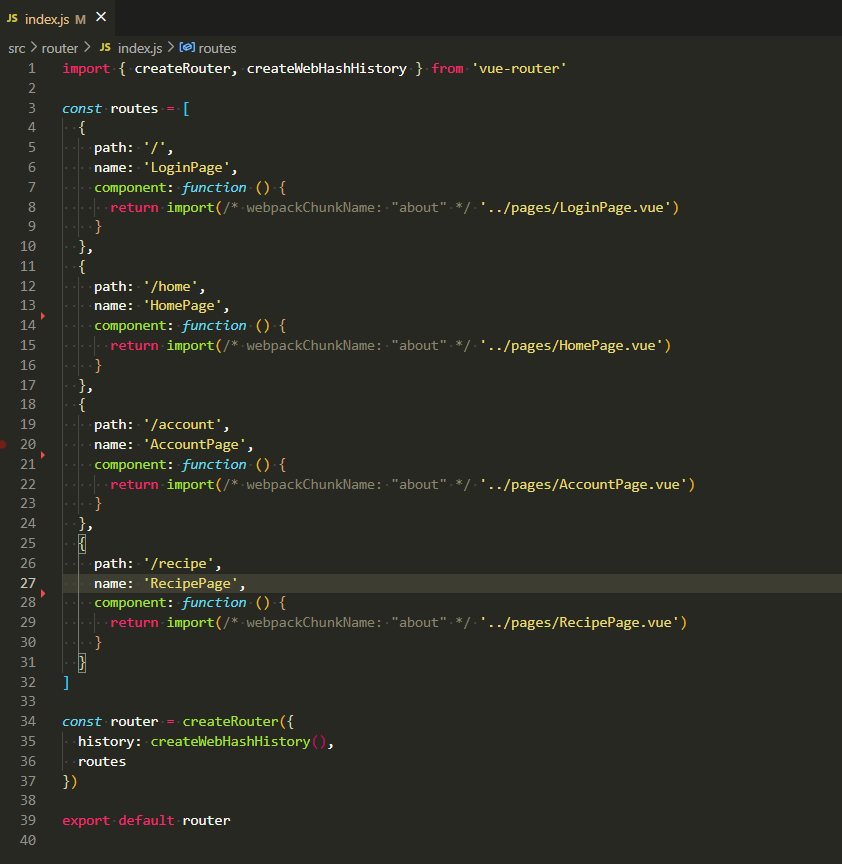


Рисунок 1 – Реализация маршрутизации

В файле store/index.js представлены глобальные переменные, массивы, основные методы для отправления запросов и получения ответов. В качестве примера представлен метод GETPARAMETERS, отвечающий за отправление запроса «http://localhost:8888/user/getParameters?userId=1» для получения данных о параметрах пользователя (Рисунок 2).

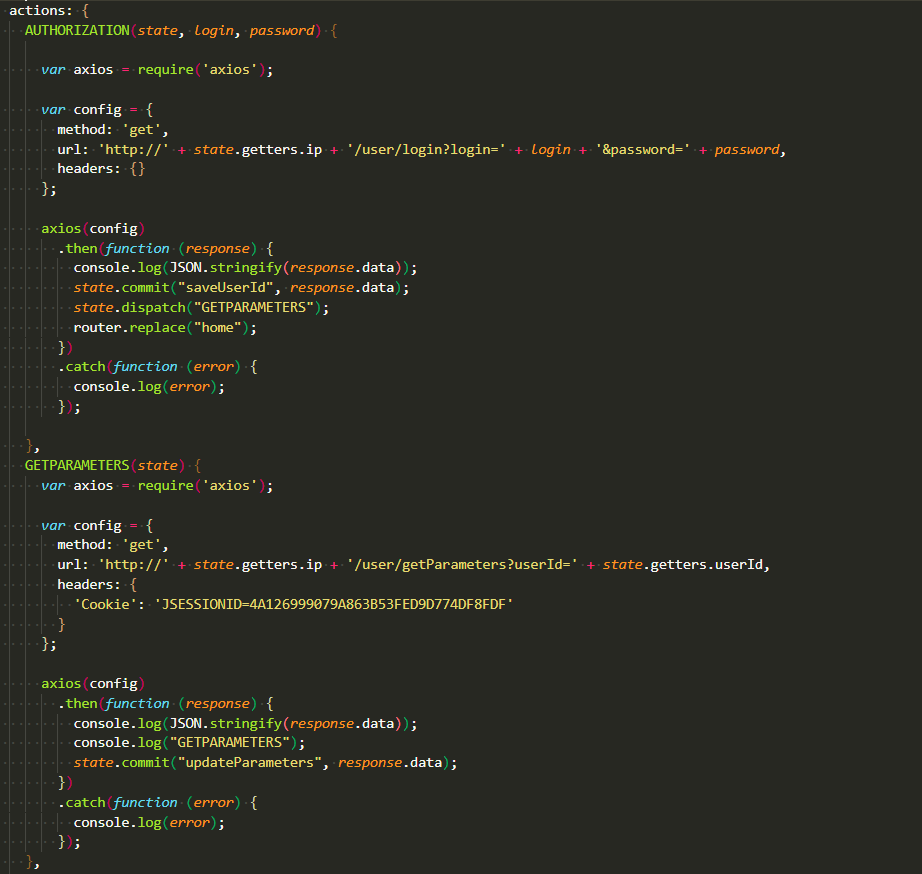


Рисунок 2 – Метод GETPARAMETERS

**Backend**

Backend реализован с помощью фреймворка Spring. Был использован Maven framework для автоматизации сборки проекта на основе POM (project object model), позволяющая подключать из интернета зависимости, не скачивая библиотеки в проект. Была также подключена библиотека Hibernate для взаимодействия с БД с помощью представления таблиц баз данных в виде классов Java. реляционного сопоставления объектов с открытым исходным кодом для Java.

Для подключения к БД был установлен драйвер PostgreSQL JDBC Driver и указаны необходимые настройки в конфигурационном файле (Рисунок 3).



Рисунок 3 – Конфигурационный файл с настройками для подключения к БД

Необходимо описать классы сущностей, в качестве примера представлен класс UserRecipes.java соответствующий таблице user\_recipes (Рисунок 4).

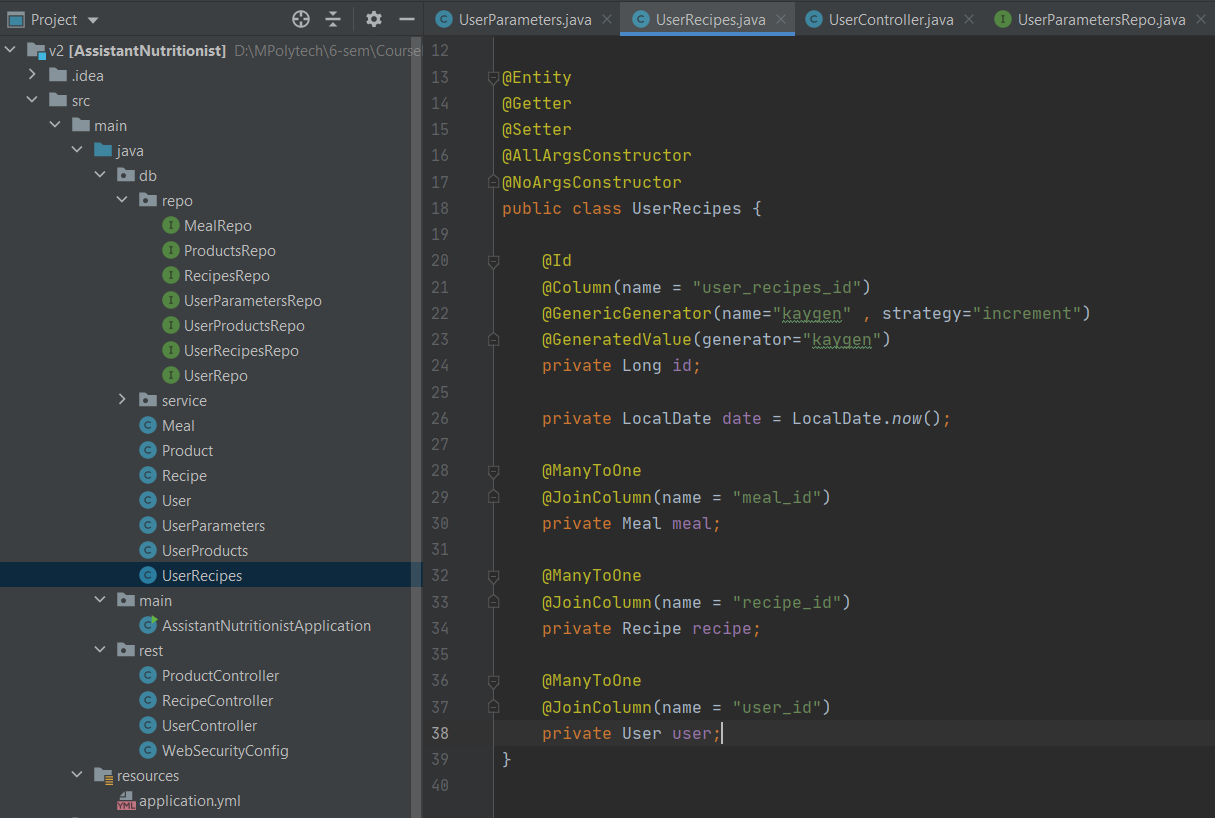


Рисунок 4 – Класс UserRecipes.java

Для взаимодействия приложения с базой данных необходимо для каждого класса-сущности определить интерфейс, содержащий набор необходимых методов для работы с данными (Рисунок 5).



Рисунок 5 – Интерфейс UserParametersRepo

Создание классов-контроллеров, в котором реализуется логика обработки клиентских запросов на эндпоинты. В качестве примера на рисунке 6 представлен контроллер – RecipeController.

@RestController — обозначает, что данный класс является REST контроллером, т.е. в данном классе будет реализована логика обработки клиентских запросов.

@RequestMapping("/recipe") — обозначает, что данный класс обрабатывает все запросы на адрес /recipe.

@PostMapping("/create") — обозначает, что данный метод обрабатывает POST запросы на адрес /recipe/clients.

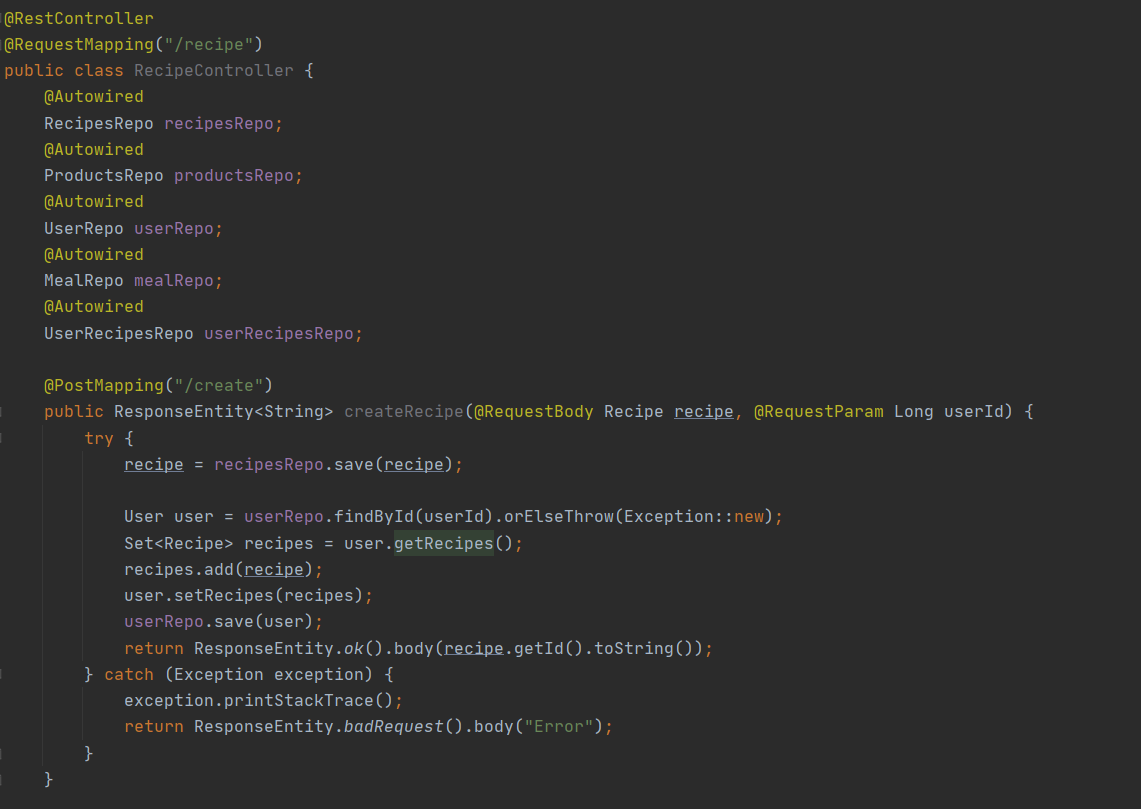


Рисунок 6 – Контроллер RecipeController

**СТРУКТУРА БАЗЫ ДАННЫХ**

В разработанном веб-сайте используется СУБД PostgresSQL. ER-диаграмма используемой базы данных приведена на рисунке 7. Для построения диаграмм был использован программный продукт Dbeaver.

В каждой таблице присутствуют ограничения, которые также показаны на ER-диаграмме (Рисунок 7).

Тестовые данные представлены на рисунках А.1–А.8 в приложения А.

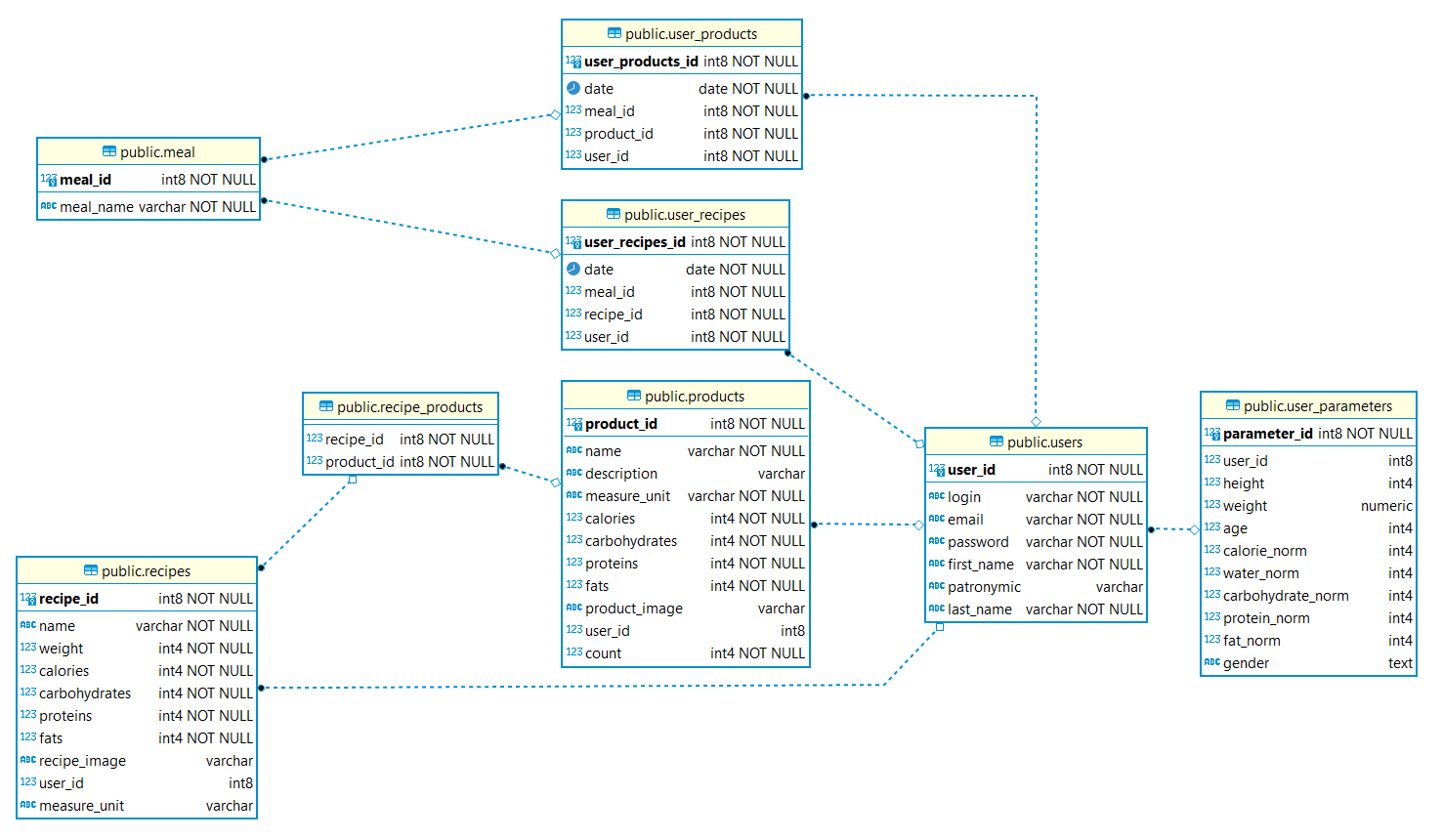


Рисунок 7 – ER-диаграмма базы данных AssistantNutritionist

**ТЕСТИРОВАНИЕ**

В процессе разработки веб-сайта неоднократно происходило ручное тестирование всего функционала для получения информации о качестве разработки, необходимой для отлавливания ошибок на стадии разработки, а также позволяющей убедиться, что весь функционал работоспособен.

Аналогично в процессе разработки веб-сайта производилось автоматизированное тестирование API c помощью программного обеспечения Postman. Были составлены тесты, покрывающие функционал различных модулей системы.

Тест «GetRecipes» проверяет работоспособность GET-запроса для получения всех рецептов пользователя. В качестве параметра запроса указывается id пользователя. Фрагмент ответа сервера в формате JSON и сам запрос представлены на рисунке 8.

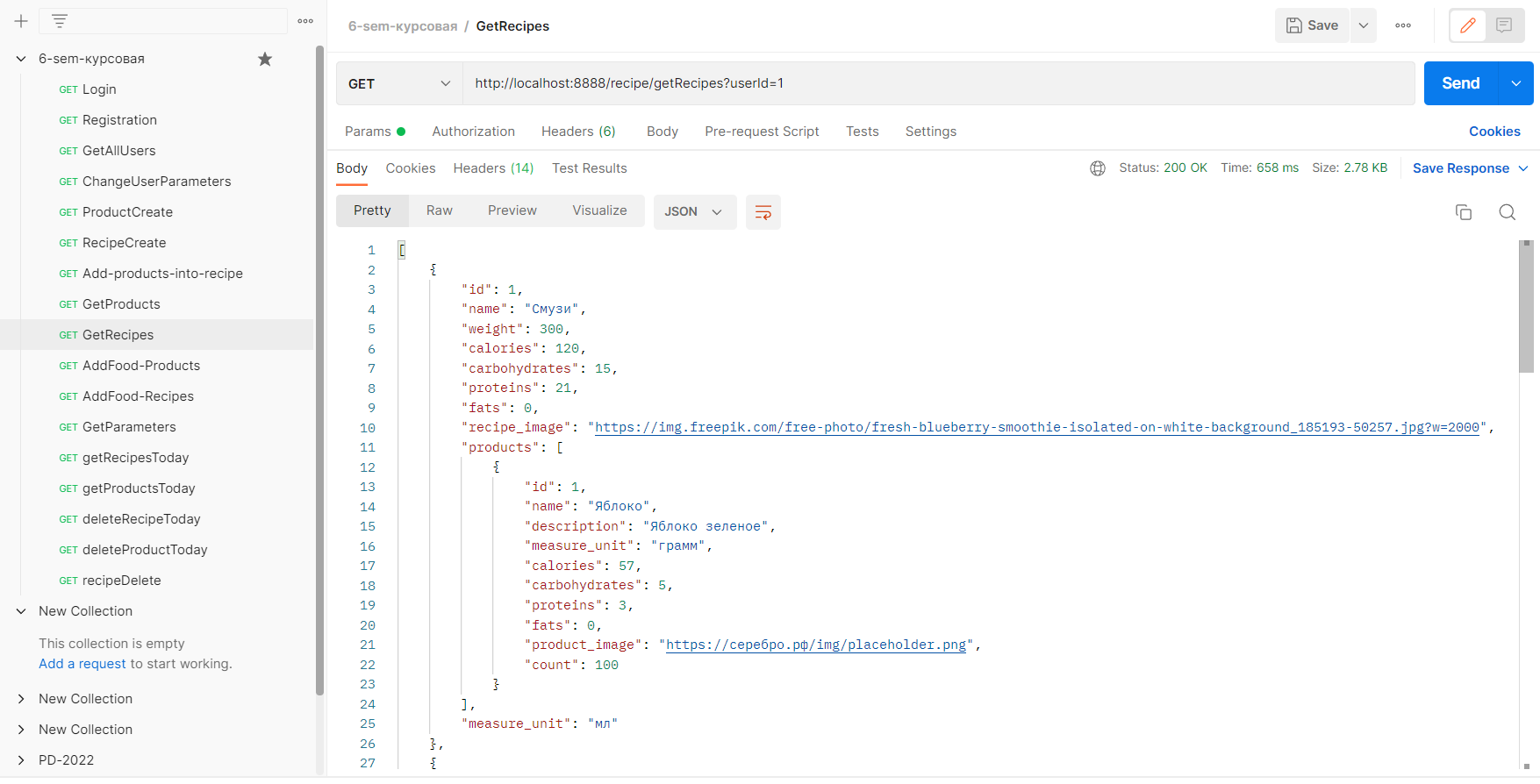


Рисунок 8 – Тест «GetRecipes»

**ЗАКЛЮЧЕНИЕ**

В рамках курсового проекта был разработан веб-сайт для контроля питания, включающий в себя: разработка пользовательского интерфейса, создание структуры базы данных, создание базы данных, разработка API, реализующего CRUD операции.

Ссылка на GitHub репозиторий со всеми исходными кодами: <https://github.com/Svetlana-sv/course_project_6_semester/tree/Frontend>.

**ПРИЛОЖЕНИЕ**

**Приложение А. Данные из БД**

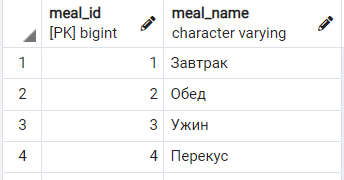


Рисунок А.1 – Таблица «meal»

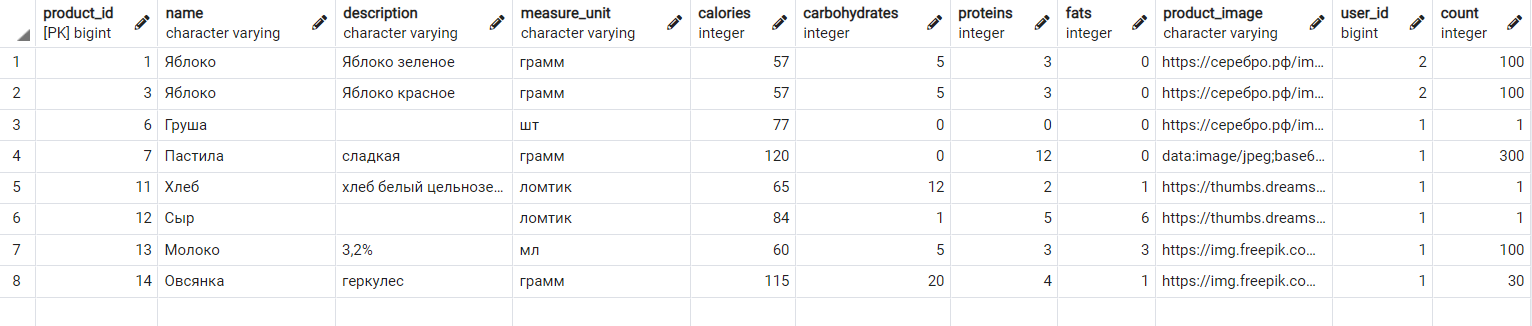


Рисунок А.2 – Таблица «products»

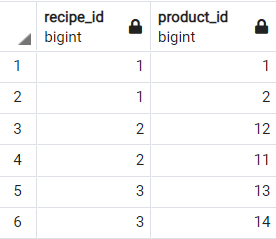


Рисунок А.3 – Таблица «recipe\_products»

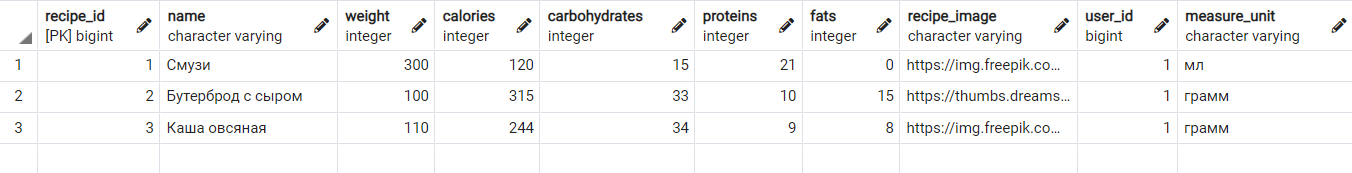


Рисунок А.4 – Таблица «recipes»

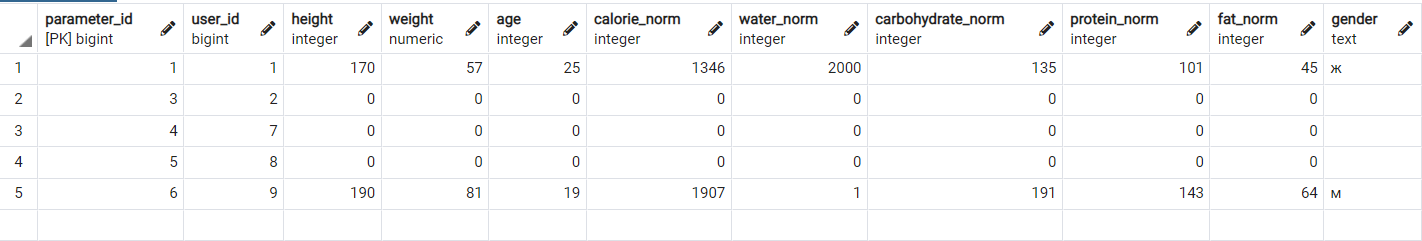


Рисунок А.5 – Таблица «user\_parameters»

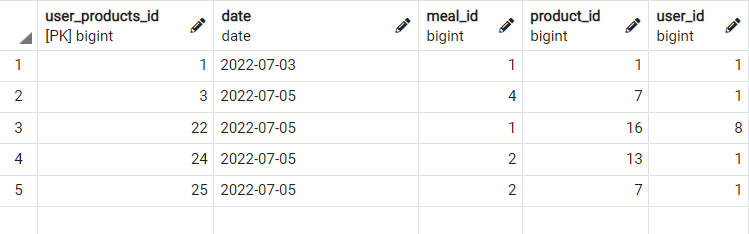


Рисунок А.6 – Таблица «user\_products»

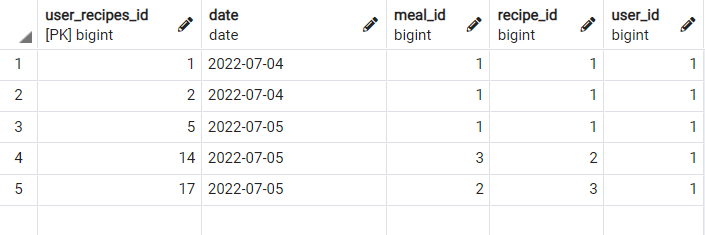


Рисунок А.7 – Таблица «user\_recipes»

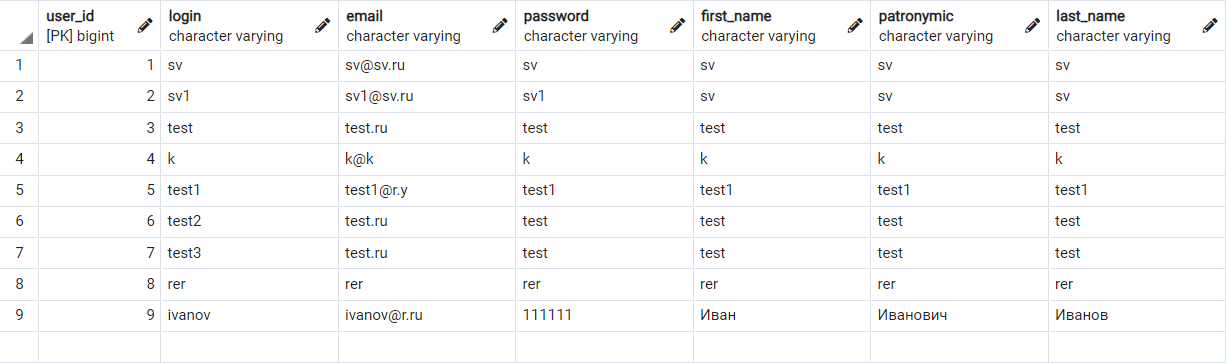


Рисунок А.8 – Таблица «users»