|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»**  **Московский приборостроительный техникум** |

ОТЧЕТ

по учебной практике

УП.04.01 Внедрение и поддержка программного обеспечения\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Профессионального модуля ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем .

Специальность 09.02.07 Информационные системы и программирование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Студент Шашков Владислав Сергеевич

*(фамилия, имя, отчество)*

Группа П50-1-21

Руководитель по практической подготовке от техникума

Серяк Даниил Владимирович.

*(фамилия, имя, отчество)*

«\_\_» \_\_\_\_\_\_ 202

Оглавление

[Практическая работа № 1 3](#_Toc179935472)

[Практическая работа № 2 9](#_Toc179935473)

[Практическая работа № 3 14](#_Toc179935474)

[Практическая работа № 4 18](#_Toc179935475)

[Практическая работа № 5 29](#_Toc179935476)

# Практическая работа № 1

Цель работы: Цель заключается в создании веб-приложения с использованием Spring Boot, которое включает в себя три страницы: главную, страницу калькулятора и конвертера валют. Главная страница содержит кнопки для перехода на другие страницы. Калькулятор выполняет базовые арифметические операции и отображает результат на отдельной странице с использованием @PostMapping для обработки данных. Конвертер валют предоставляет возможность выбора валют из двух выпадающих списков для перевода суммы с динамическим расчетом по заданным курсам, используя @GetMapping и @PostMapping для переходов и @RequestParam для передачи параметров

Ход работы:

1. Создаем Get запросы для перехода на страницы

Переход на страницы

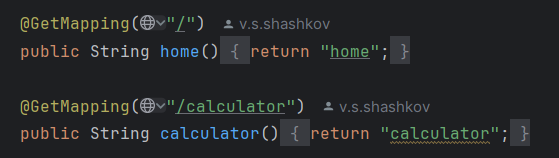


Рисунок 1 – Переход на страницы

1. Обработка POST-запроса

В этом коде обрабатываем POST-запросы на URl калькулятора, принимает три параметра из формы, а потом делает вычисление на основе выбранной операции, добавляет результат вычисления в модель



Рисунок 2 – Калькулятор

1. Ссылка на страницу конвертор валюта

Обработка GET-запросы на URL, возвращает имя шаблона

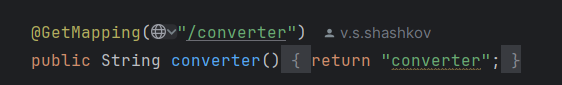


Рисунок 3 – Конвертор

1. Обработка POST-запроса на URL

Обработка POST-запроса, принимает 3 формы для исходной валюта, целевая валюта, сумма для конвертации. Это позволяет динамически отображать результаты

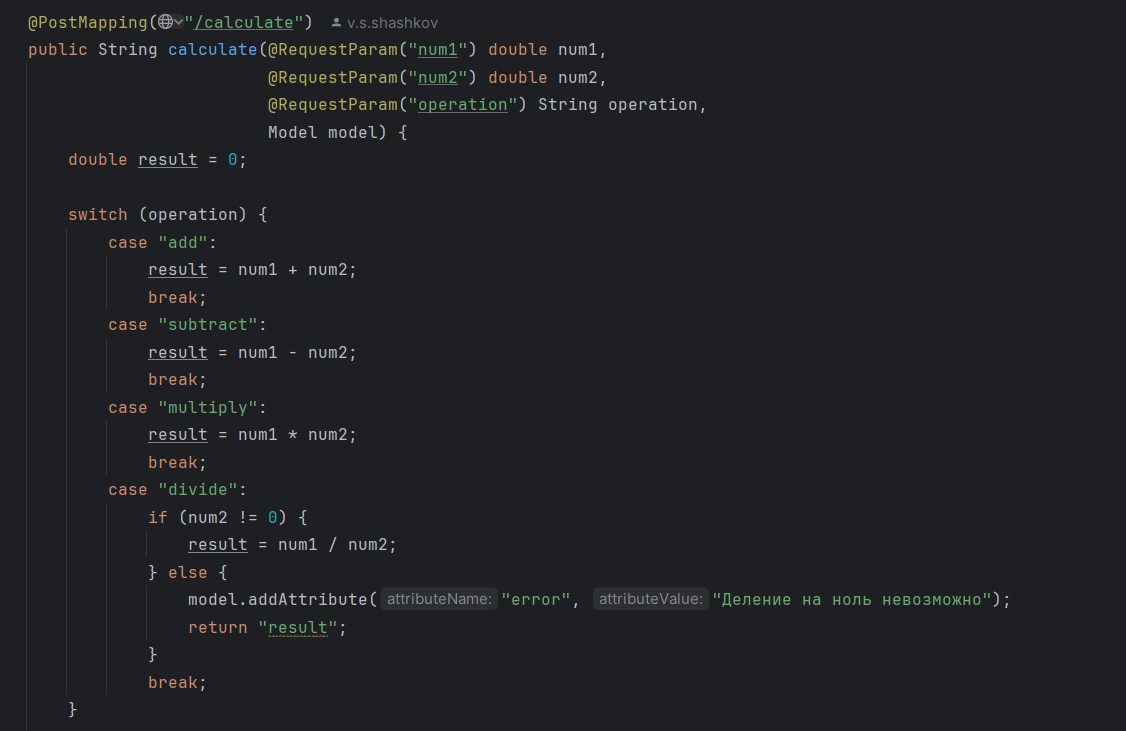


Рисунок 4 – Обработка Post-запроса

1. Работа конвертер

Этот код позволяет пользователю выбрать исходную и целевую валюту



Рисунок 5 - Конвертер

1. Калькулятор страница

Страница калькулятора для работы

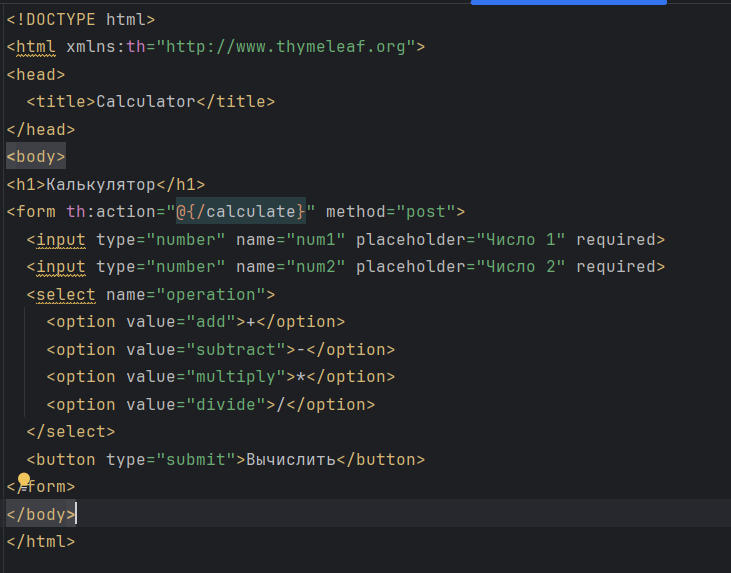


Рисунок 6 – Страница калькулятора

1. Результат конвертации

Страница работы конвертации



Рисунок 7 – Страница конвертации

1. Конвертер

Страница работы конвертера

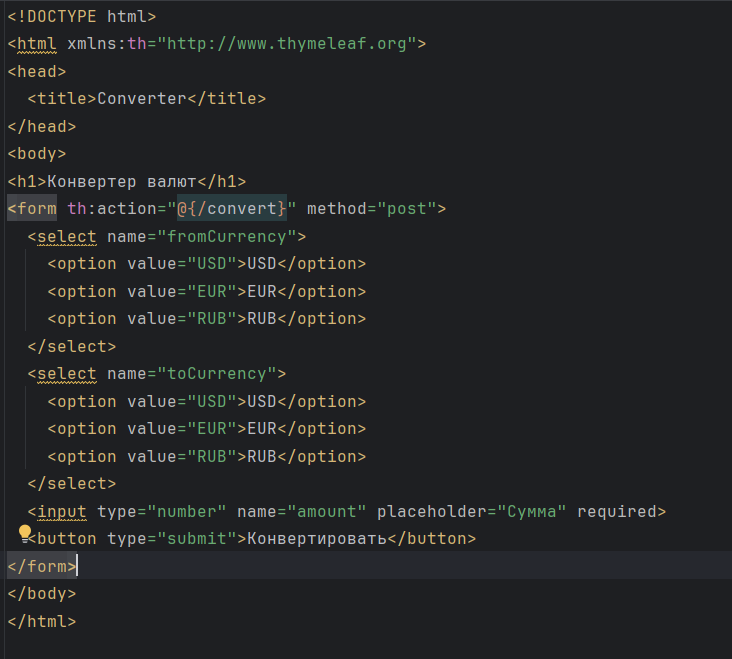


Рисунок 8 – Страница работы конвертера

1. Домашняя страница

Страница работы домашний страницы

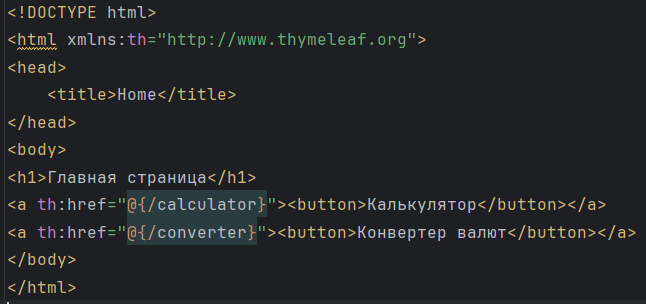


Рисунок 9 – Страница работы домашней страницы

1. Результат

Страница результата



Рисунок 10 – Страница результата

Вывод: В результате проделанной работы было создано веб-приложение на основе Spring Boot, состоящее из трёх страниц: главной, страницы калькулятора и конвертера валют. Главная страница обеспечивает удобную навигацию между разделами приложения с помощью кнопок. Калькулятор успешно выполняет базовые арифметические операции и отображает результаты на отдельной странице, используя аннотацию @PostMapping для обработки данных. Конвертер валют предоставляет пользователям возможность выбора валют из двух выпадающих списков и осуществляет перевод суммы с динамическим расчётом по заданным курсам, применяя аннотации @GetMapping и @PostMapping для переходов между страницами, а также @RequestParam для передачи параметров.

# Практическая работа № 2

Цель работы: Создать две сущности кода для обеспечения функциональности кода, создать поиск по параметру id, фильтрацию, реализовал разбиение на страницы для вывода данных.

Ход работы:

1. Сначала идет контроллер



Рисунок 11 – Контроллер

1. В нем хранятся методы для вывода всех студентов, добавление нового студента, обновление существующего, физическое удаление студента, множественное удаление студента, логическое удаление студента

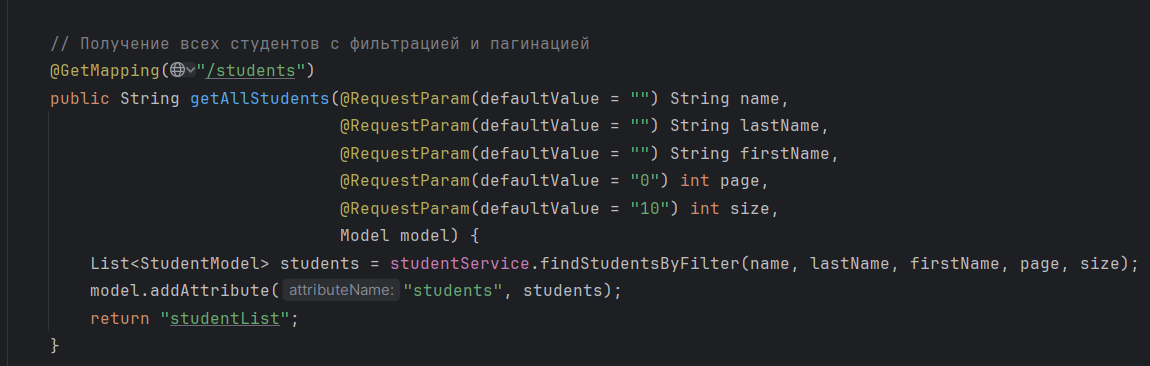


Рисунок 12 – Получение всех студентов

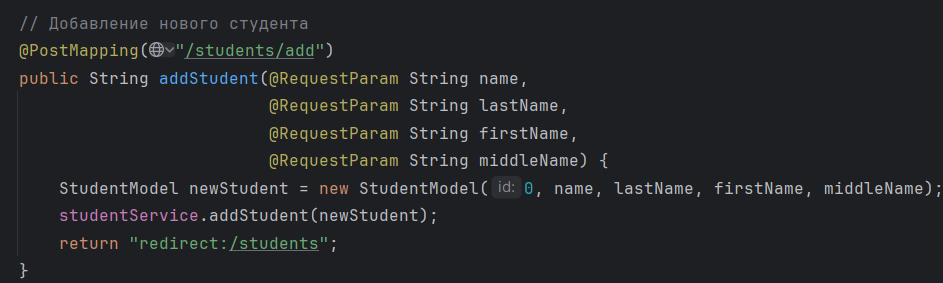


Рисунок 13 – Добавление нового студента

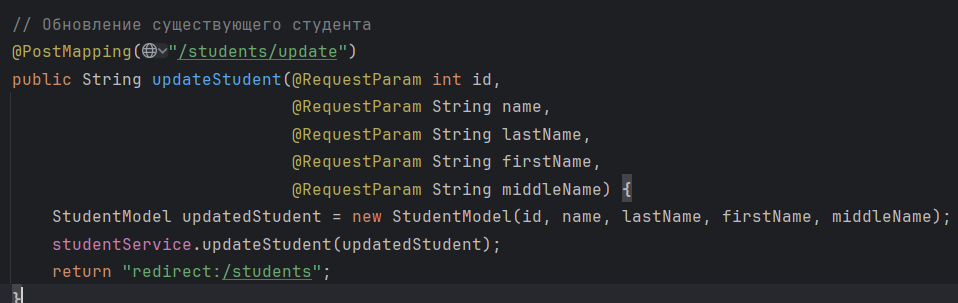


Рисунок 14 – Обновление существующего студента

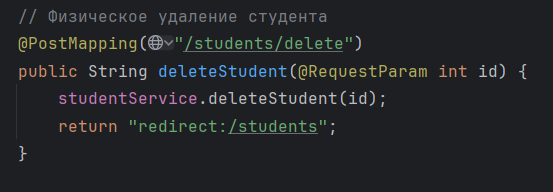


Рисунок 15 – Физическое удаление студента

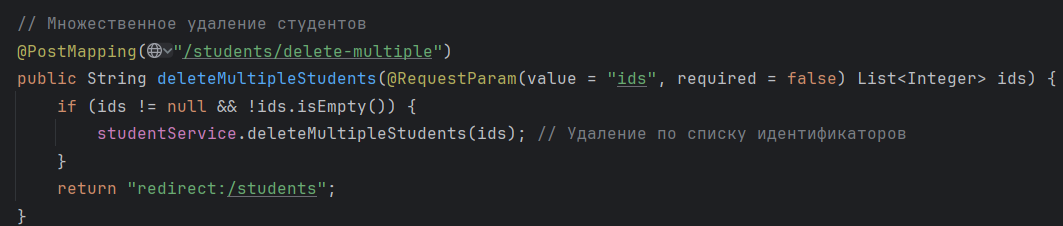


Рисунок 16 – Множественное удаление студента

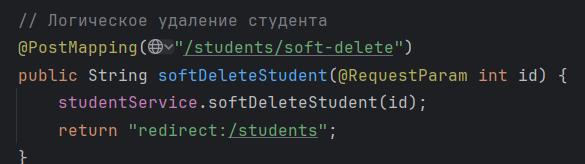


Рисунок 17 – Логическое удаление студента

1. Добавление функциональности сущности для работы с данными

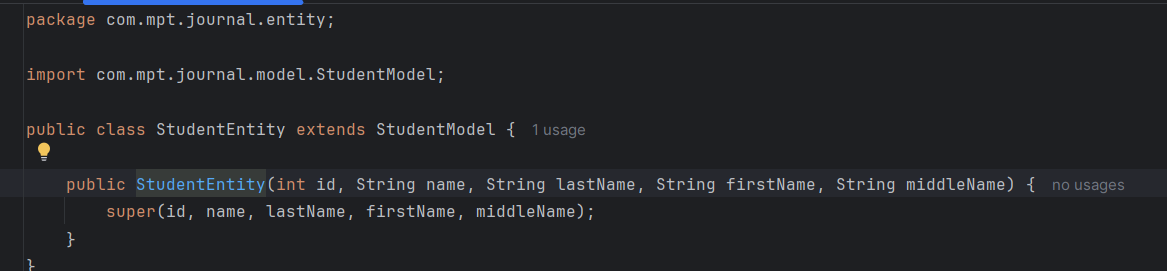


Рисунок 18 – Функциональность сущности

1. Создание моделей для заполнения их с помощью окон

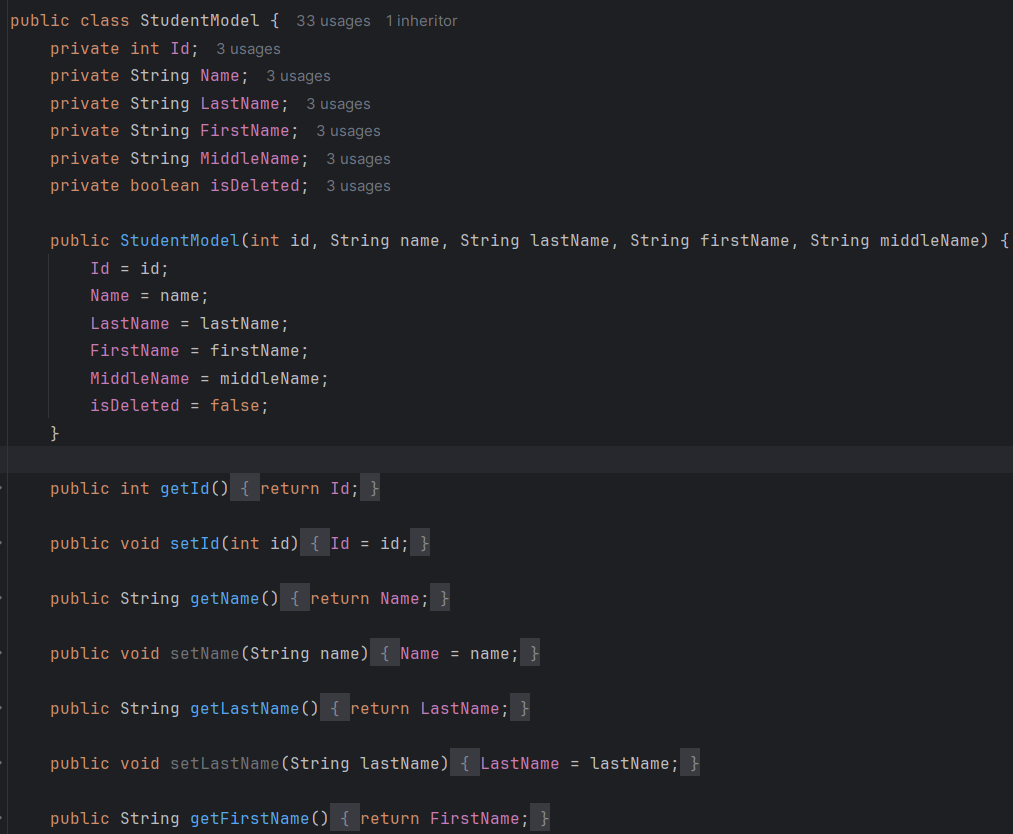


Рисунок 19 – Модель

1. Подключение моделек к функционалу

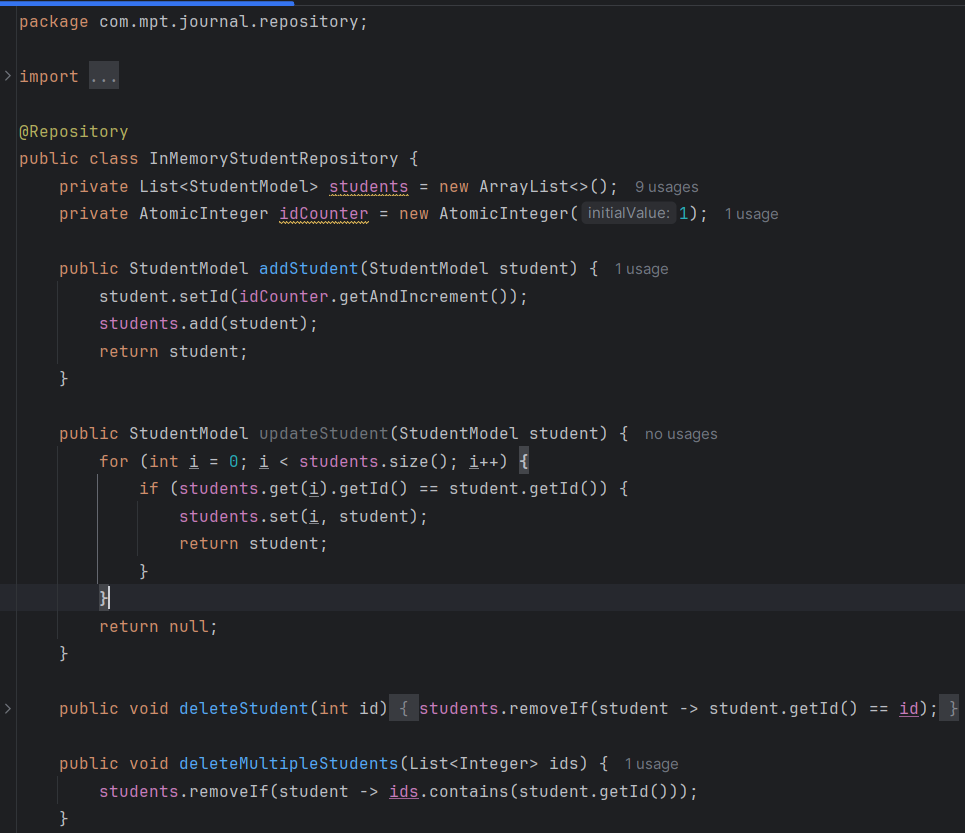


Рисунок 20 – Модель

1. Создание сервиса для работы функционала

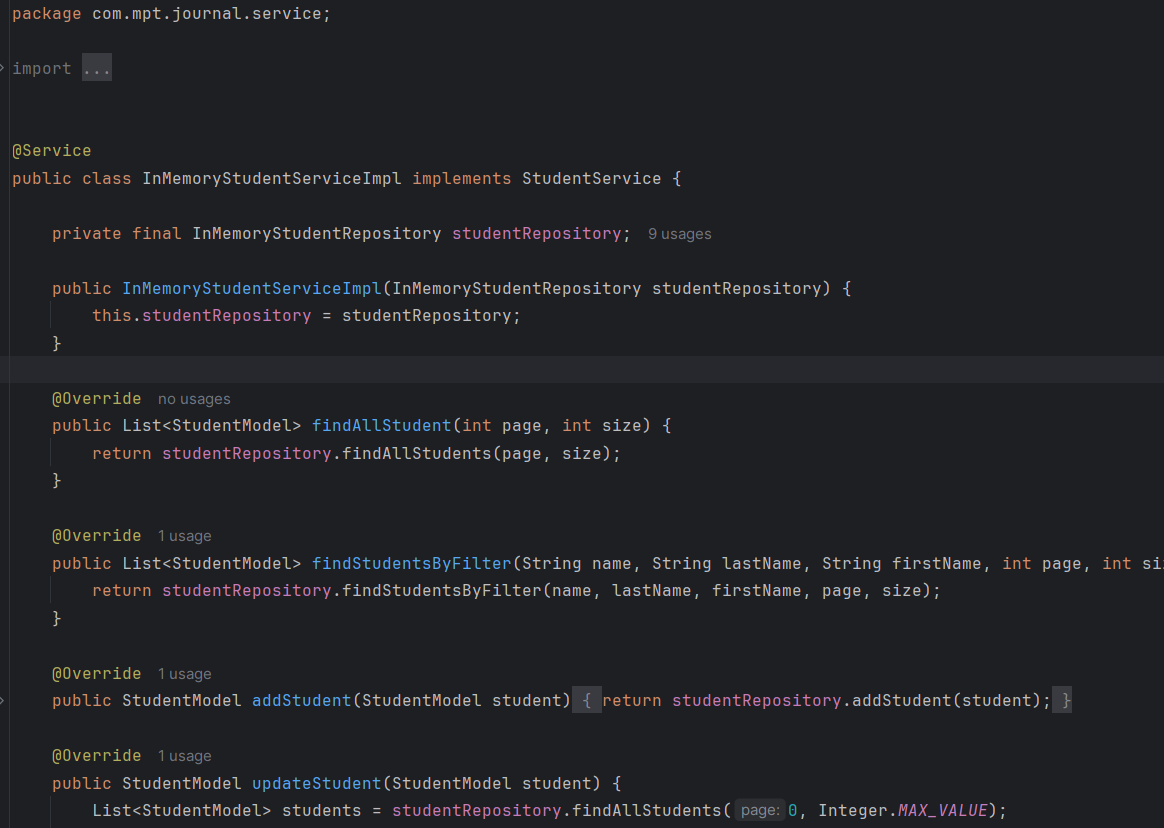


Рисунок 21 – Создание сервиса

1. Создание сервиса для записи моделей студента

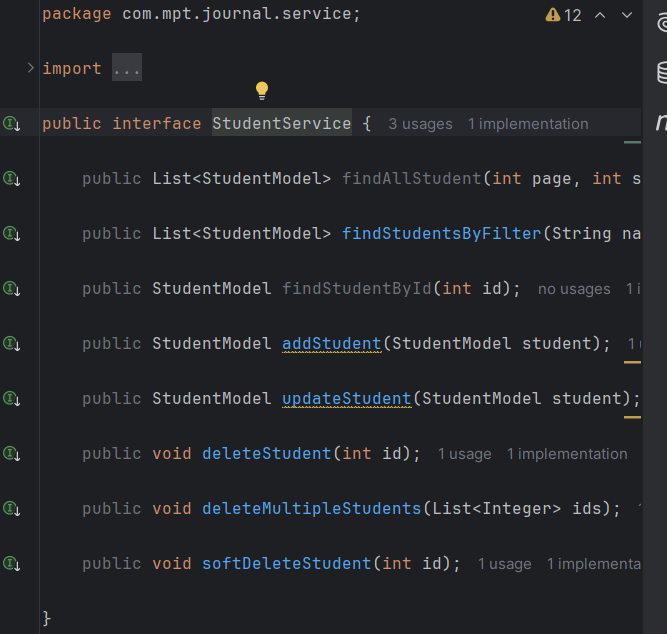


Рисунок 22 – Создание сервиса для студентов

1. Создание списка студентов в html для создания страницы

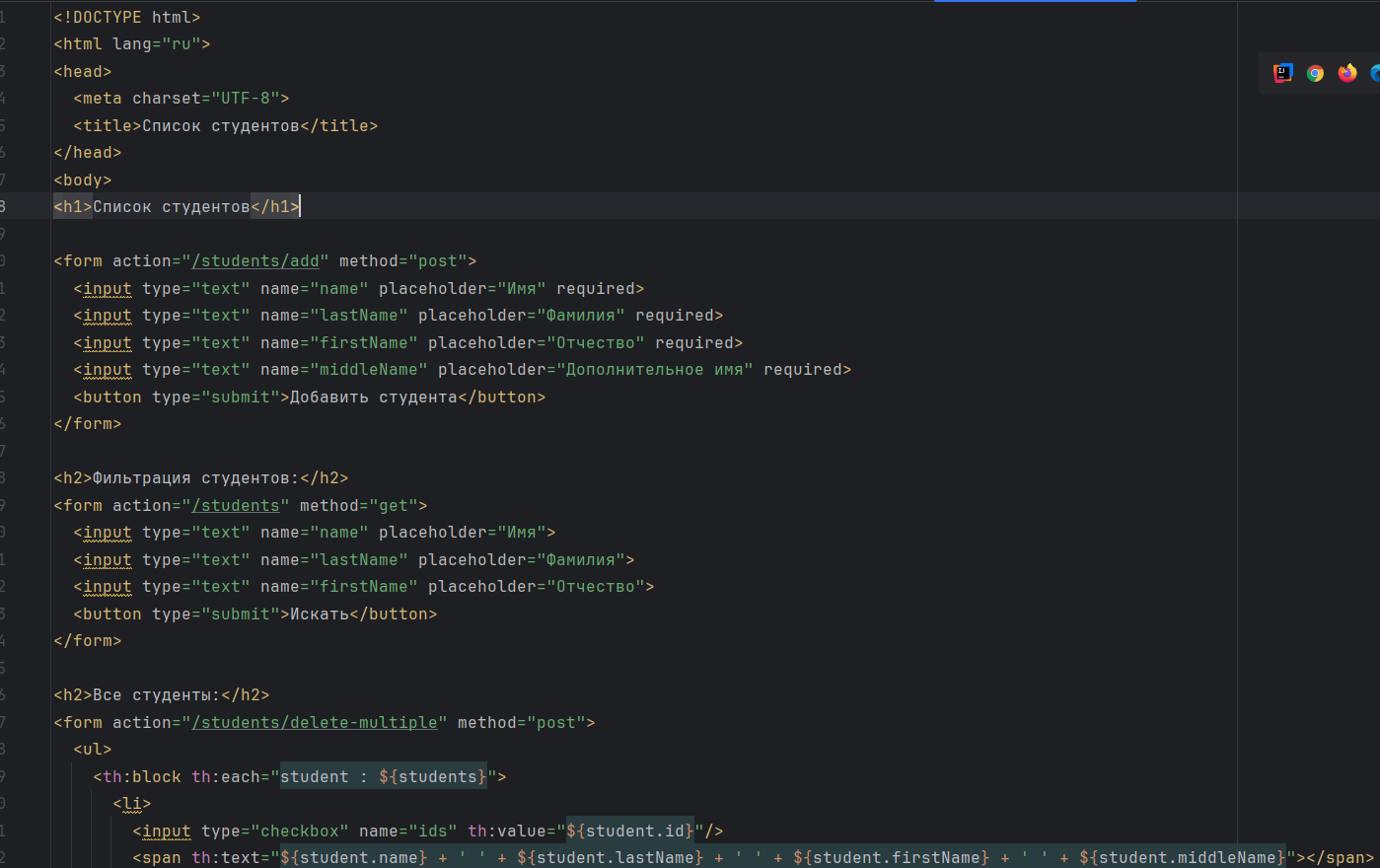


Рисунок 23 – Список студентов

Вывод: Создал две сущности, реализовал поиск по параметру, фильтрацию, реализовал разбиение на страницы для вывода данных.

# Практическая работа № 3

Цель работы: Необходимо подключить выбранную СУБД, сделать валидацию на каждое поле с помощью аннотаций.

Ход работы:

1. Сначала идет контроллер



Рисунок 24 – Контроллер

1. В контроллере написана форма, добавление студента, обновление и физическое удаление, а также множественной удаление студентов.

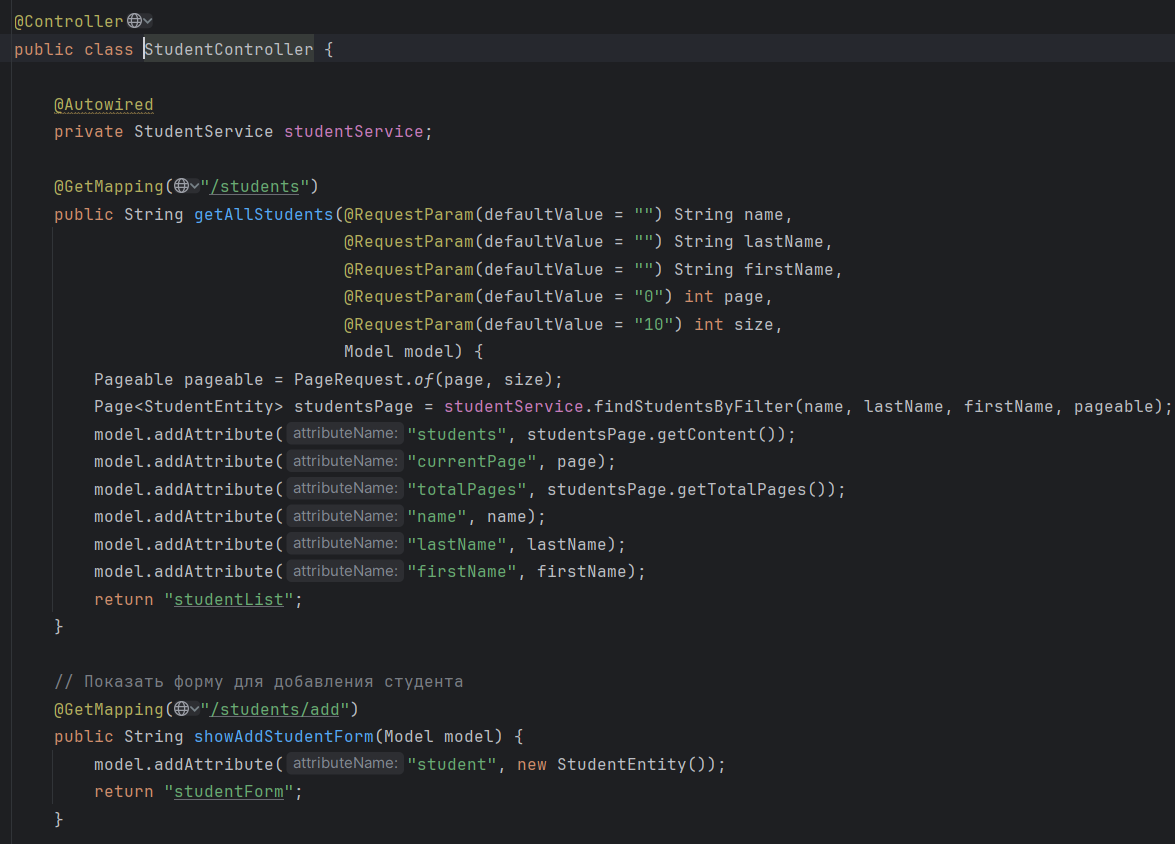


Рисунок 25 – Код контроллера

1. Создание StudentEntity для валидации

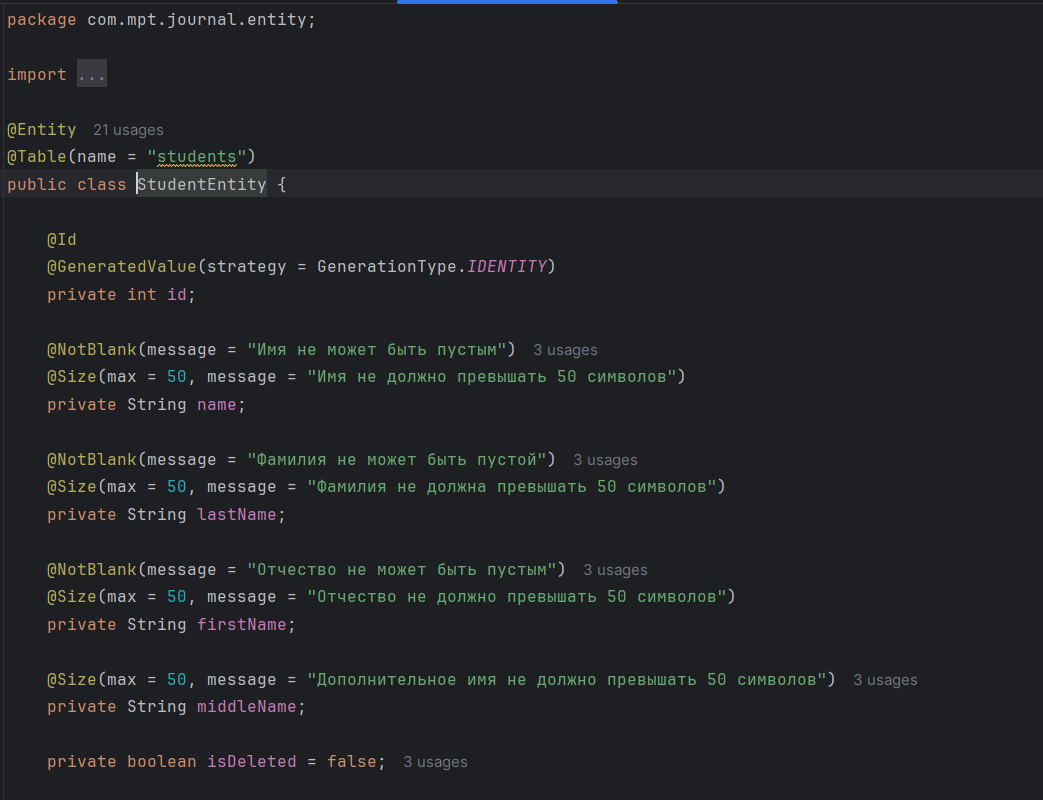


Рисунок 26 – Создание валидации

1. Создание интерфейса для репозитория

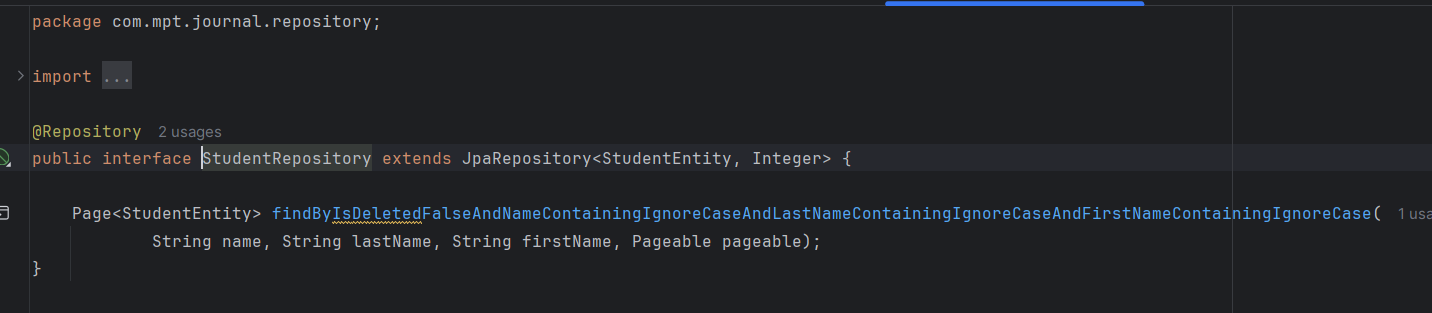


Рисунок 27 – Создание интерфейса

1. Создание интерфейса для нахождения студента по фильтру

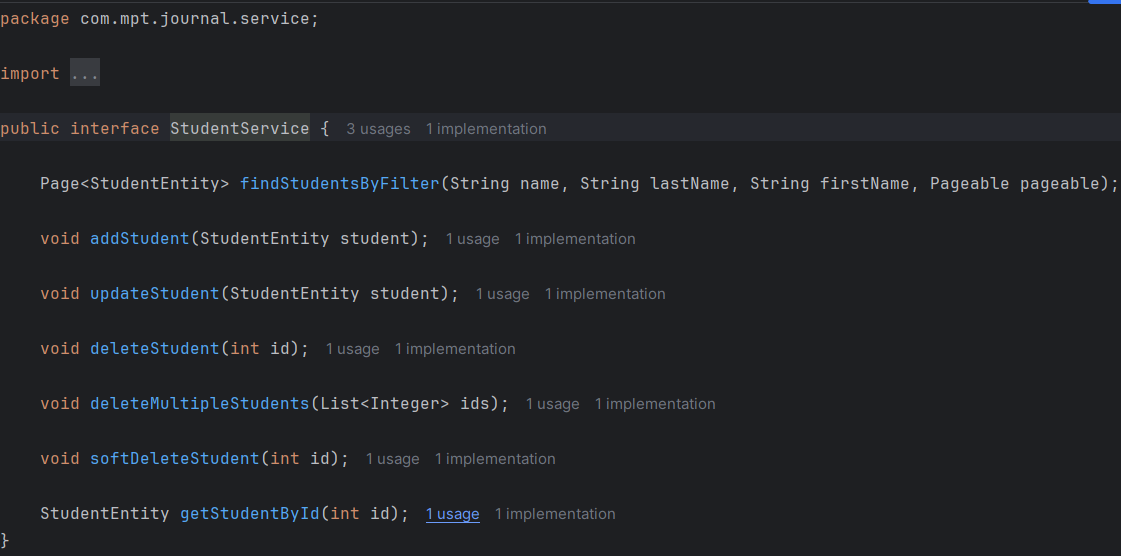


Рисунок 28 – Создание интерфейса

1. Создание сервиса для добавления, обновления, удаления, а также сортировки данных

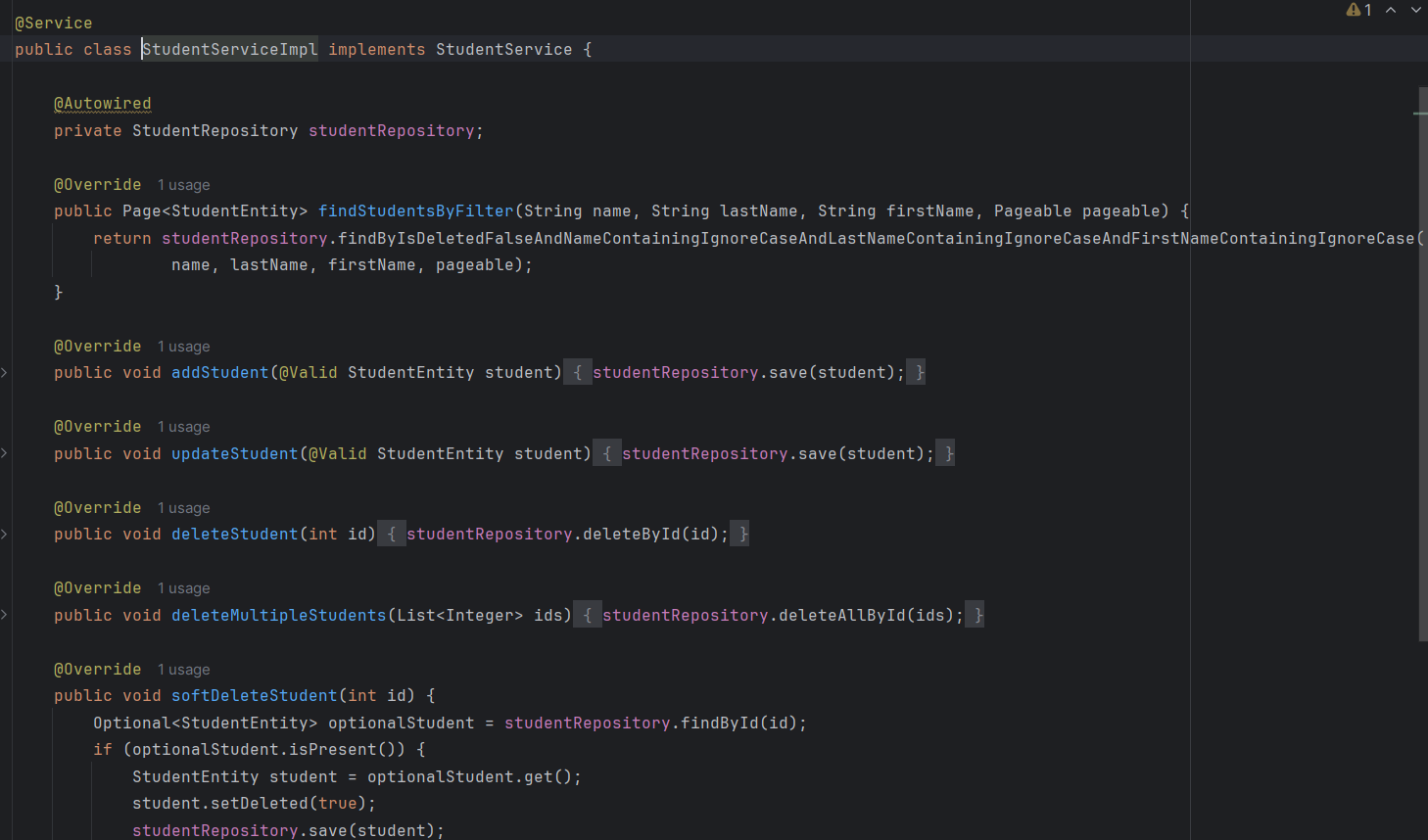


Рисунок 29 – Создание сервиса

1. Форма студента для того, чтобы удалять и редактировать студента

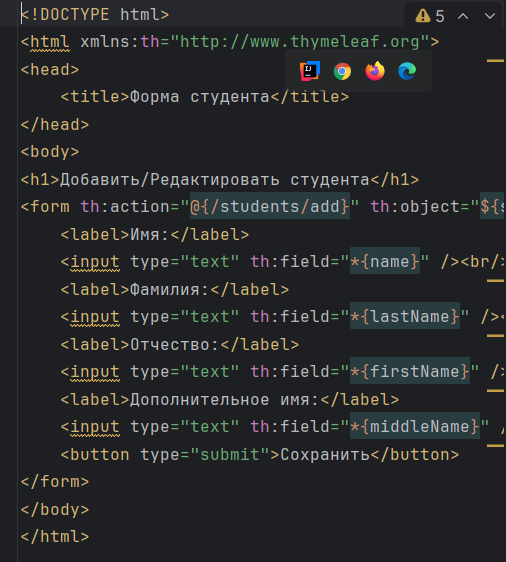


Рисунок 30 – Форма студента

1. Список студентов, в котором добавление, фильтрация и удаление студентов

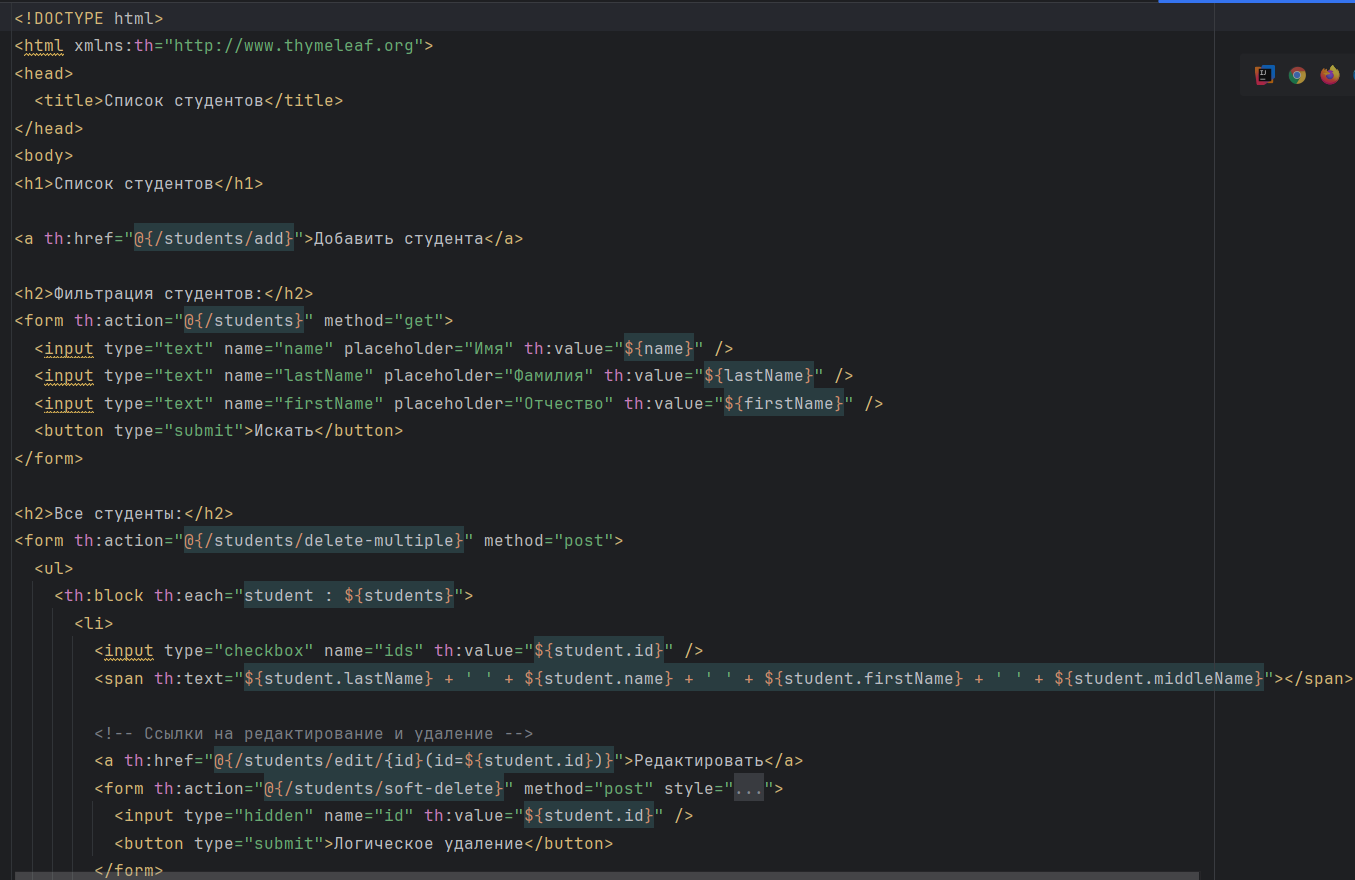


Рисунок 31 – Список студентов

1. Подключение к БД

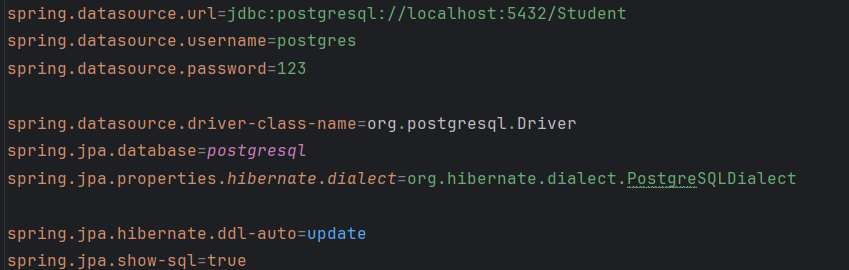


Рисунок 32 – Подключение к БД

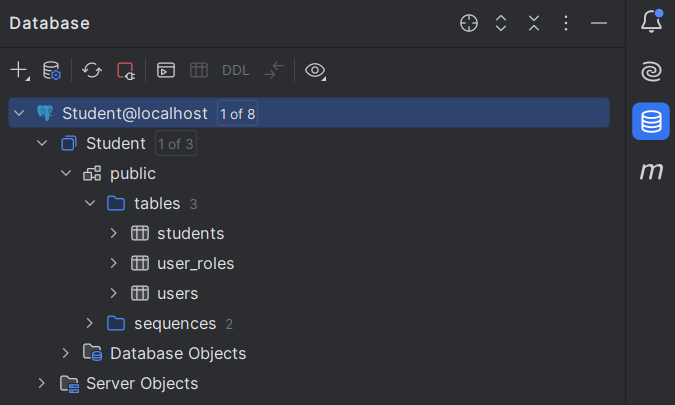


Рисунок 33 – Подключение к бд

Вывод: Подключил выбранную СУБД, сделал валидацию на каждое поле с помощью аннотаций

# Практическая работа № 4

Цель работы: Реализовать простую форму регистрации с проверкой обязательных полей, а также авторизация пользователя, еще нужно реализовать разные роли пользователей, сделать валидацию данных, безопасность данных

Ход работы:

1. Реализация шифрования пароля, а также защита страниц от запросов в виде crud-операций

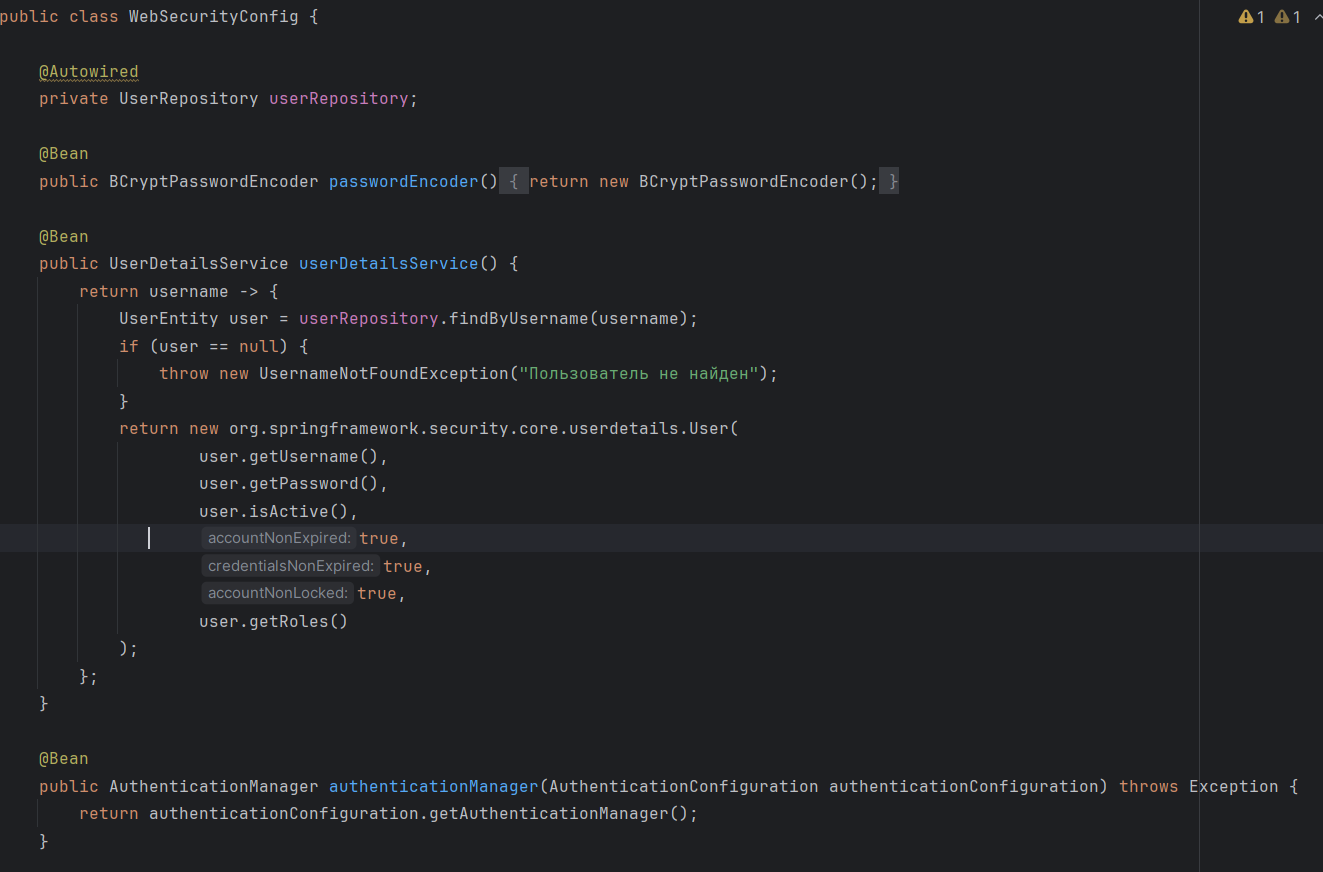


Рисунок 34 – Реализация шифрования пароля

1. Реализация контроллера для работы роли админа



Рисунок 35 – Роль админа

1. Создание контроллера для логина

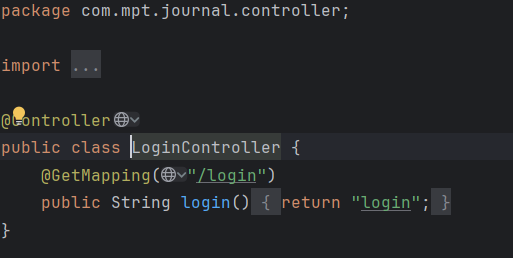


Рисунок 36 – Контроллер для логина

1. Реализация контроллера для страницы home

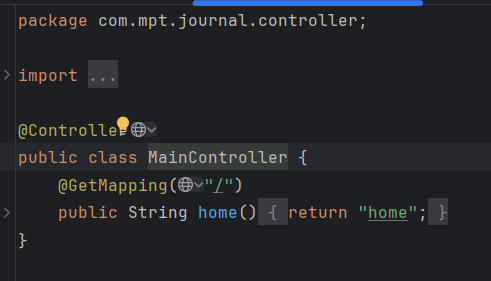


Рисунок 37 – Реализация home

1. Реализация функции для возможности регистрации

  
Рисунок 38 – Регистрация

1. Создание контроля для Студента

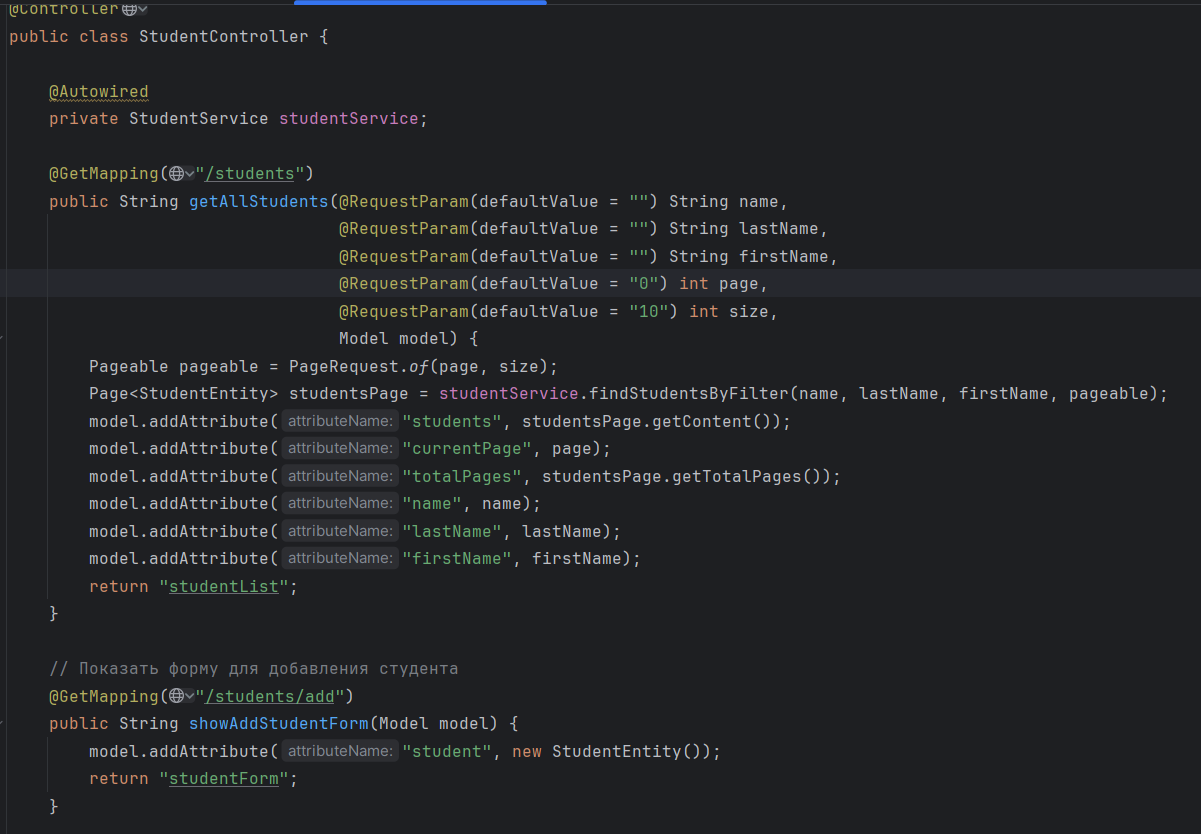


Рисунок 39 – Контроллер для студента

1. Создание файла для ролей

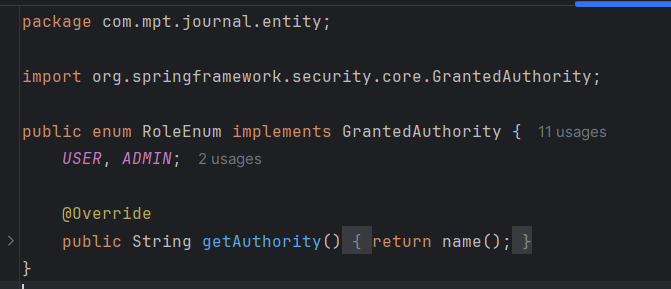


Рисунок 40 – Роли

1. Создание валидации для студента

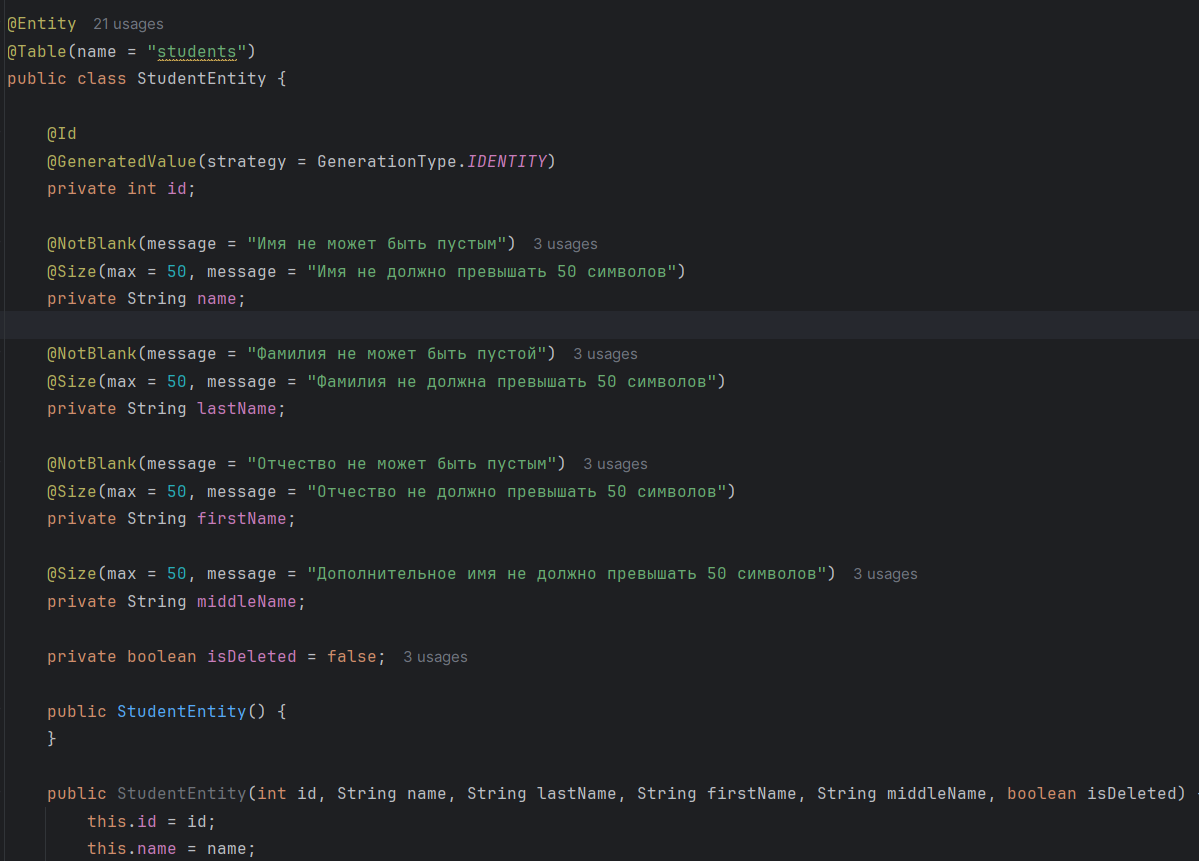


Рисунок 41 – Валидация для студента

1. Создание валидации для пользователя

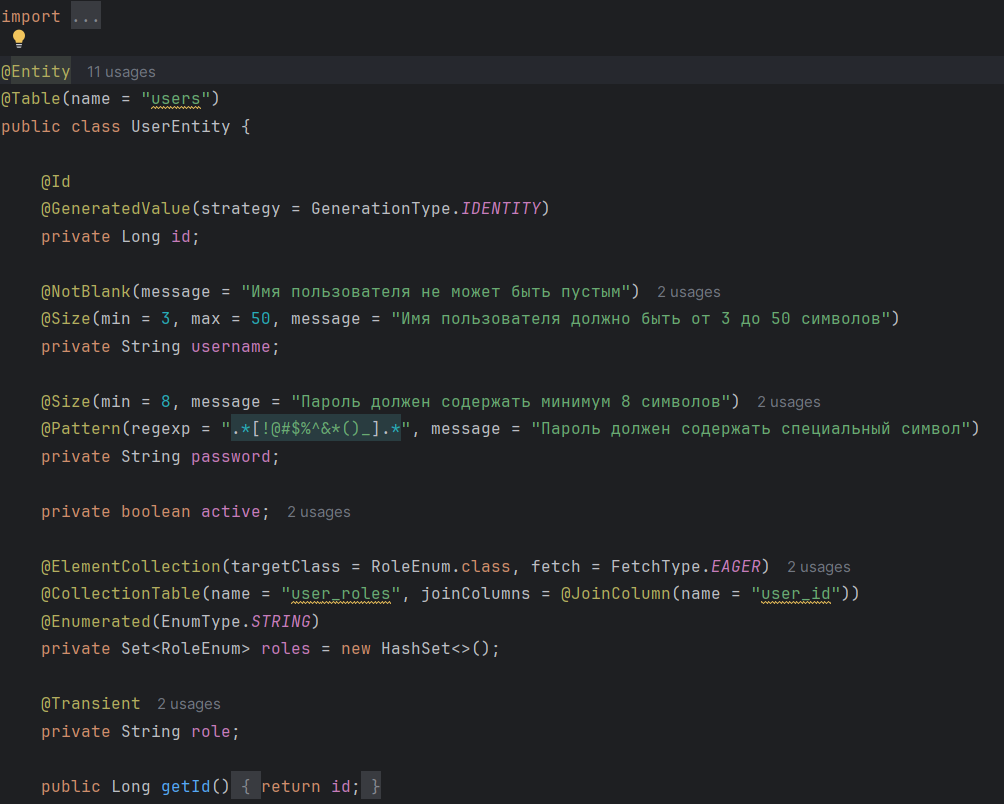


Рисунок 42 – Валидации

1. Создание репозитория для студента

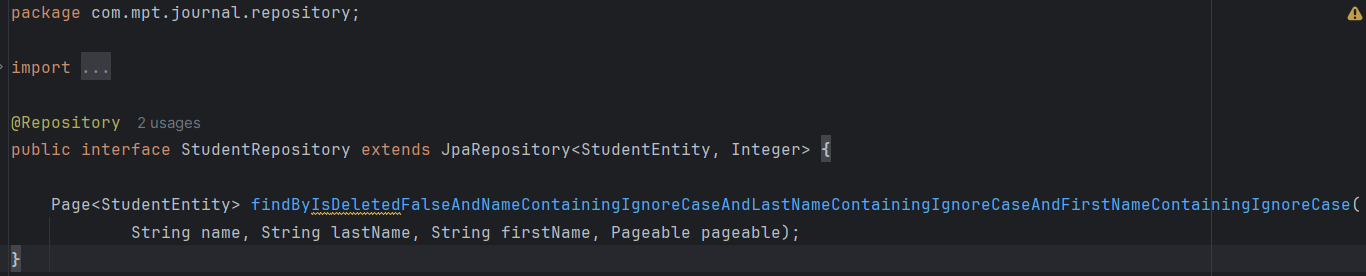


Рисунок 43 – Репозиторий

1. Создание репозитория для пользователя

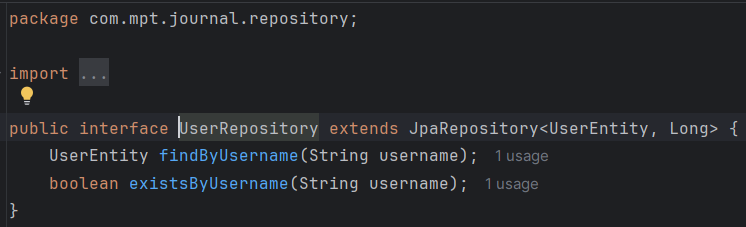


Рисунок 44 – Репозиторий для пользователя

1. Создание интерфейса для пользователя

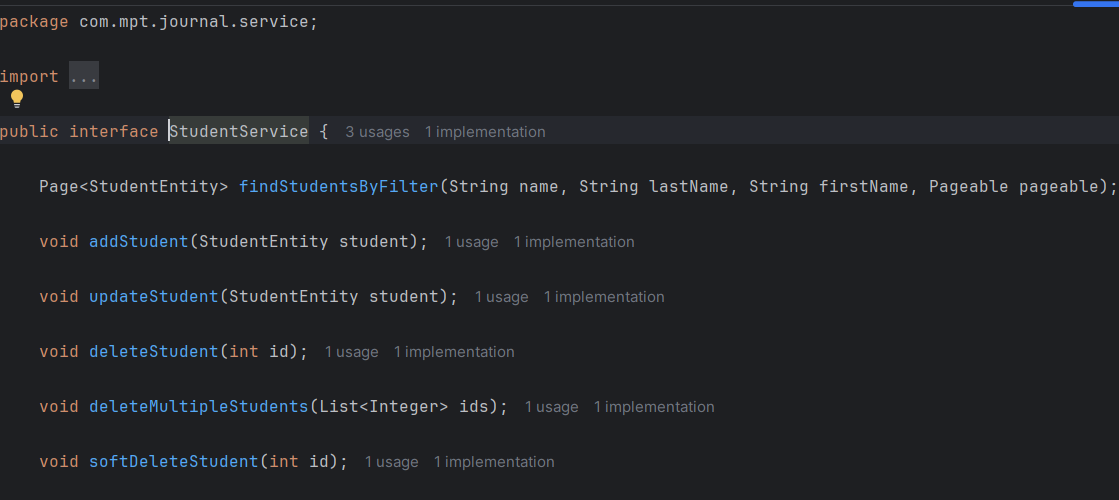


Рисунок 45 – Создание интерфейса

1. Создание сервиса для пользователя

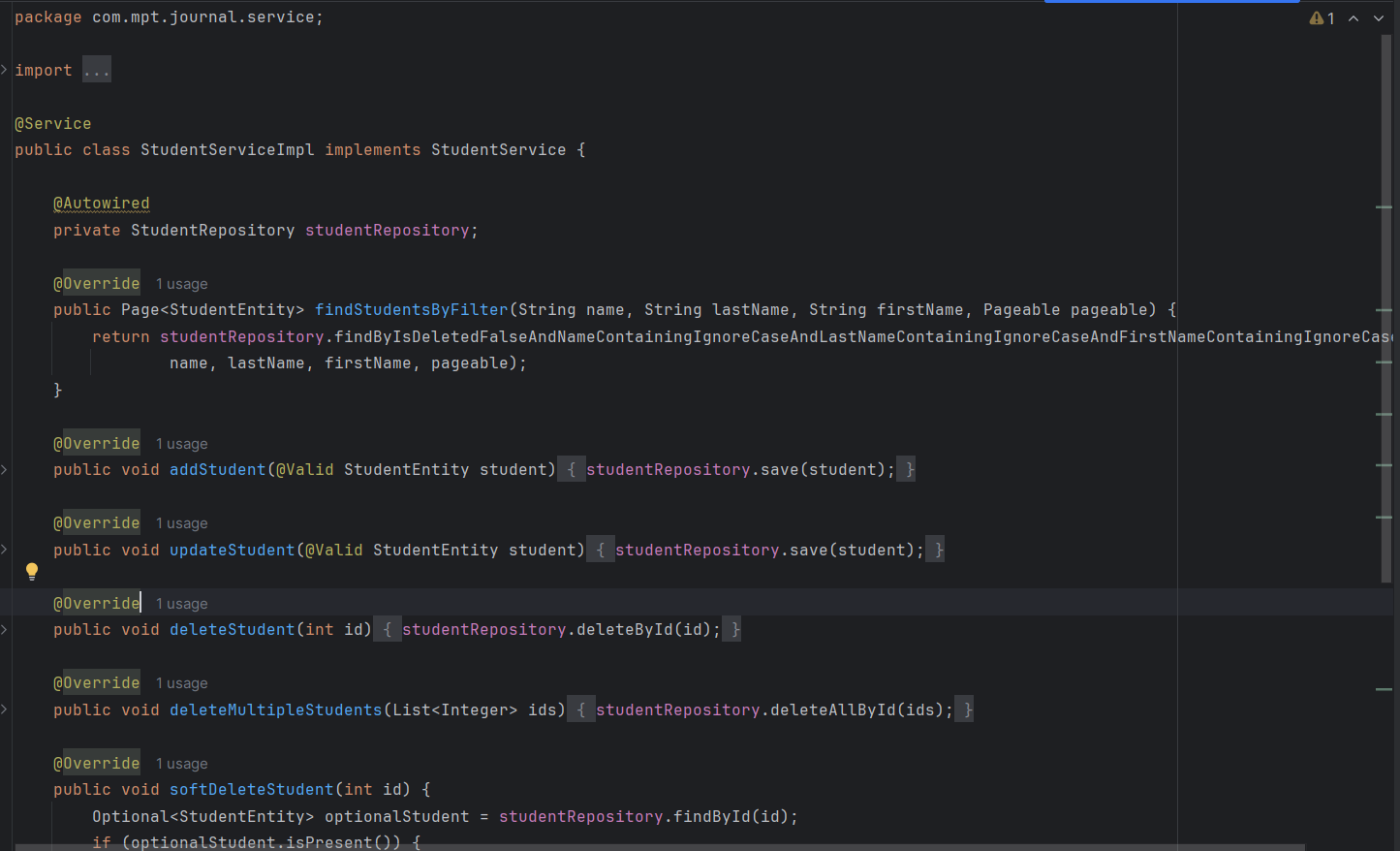


Рисунок 46 – Создание сервиса для работы студента

1. Создание домашней страницы

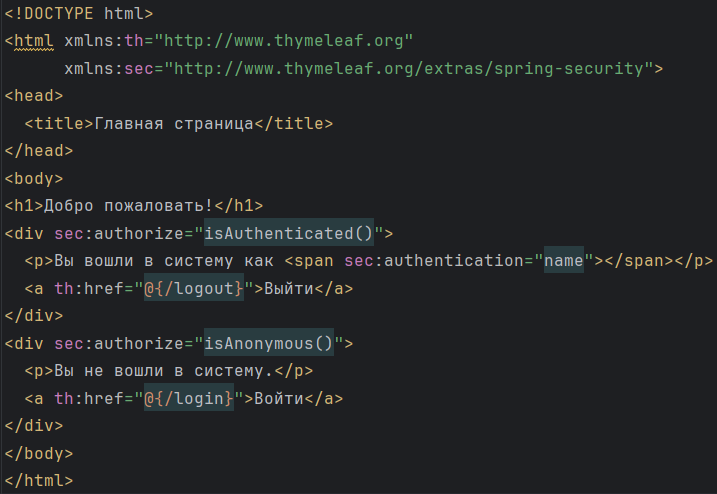


Рисунок 47 – Создание домашней страницы

1. Создание домашней страницы для логина



Рисунок 48 – Создание логина

1. Создание страницы регистрации

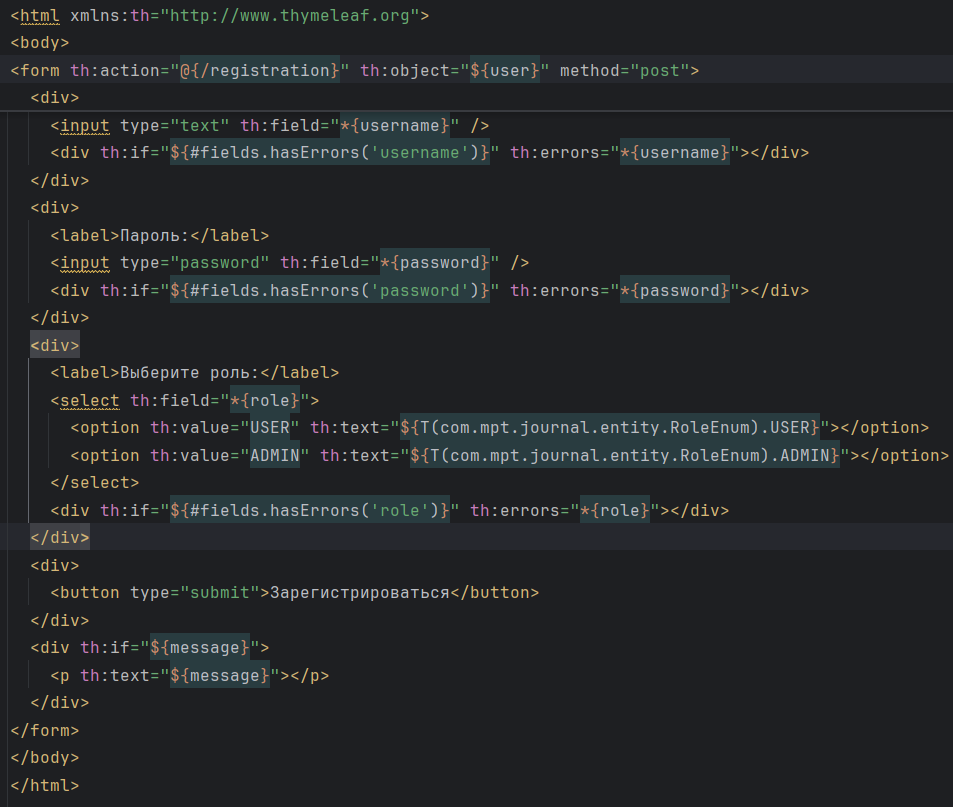


Рисунок 49 – Страница регистрации

1. Создание формы для студента

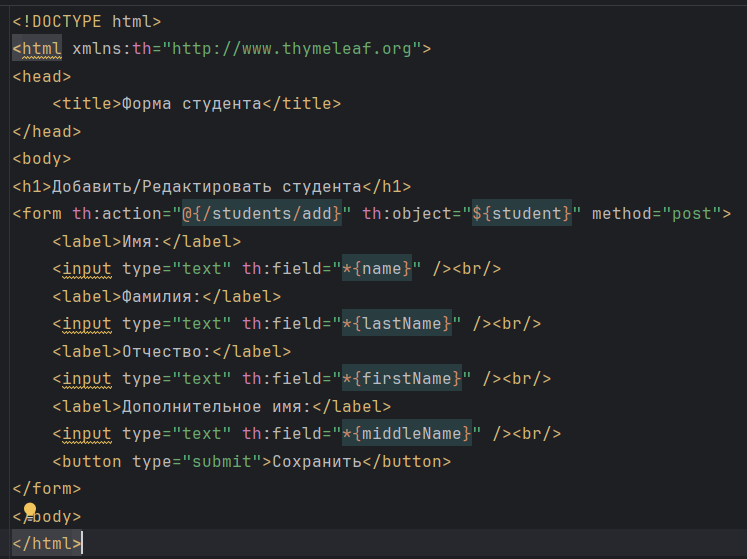


Рисунок 50 – Создание формы

1. Создание страницы для студента

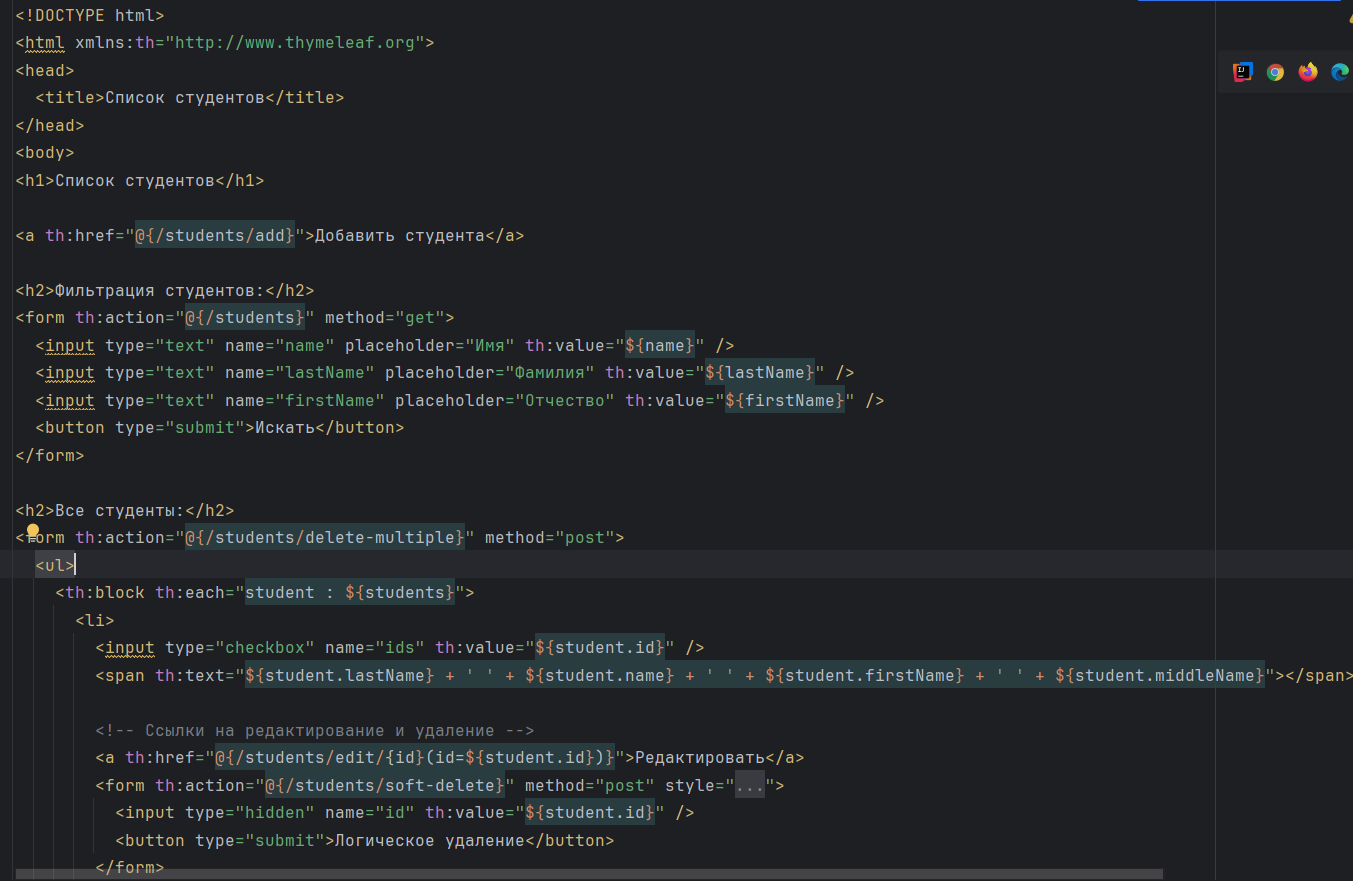


Рисунок 51 – Страница для студента

1. Редактирование пользователя

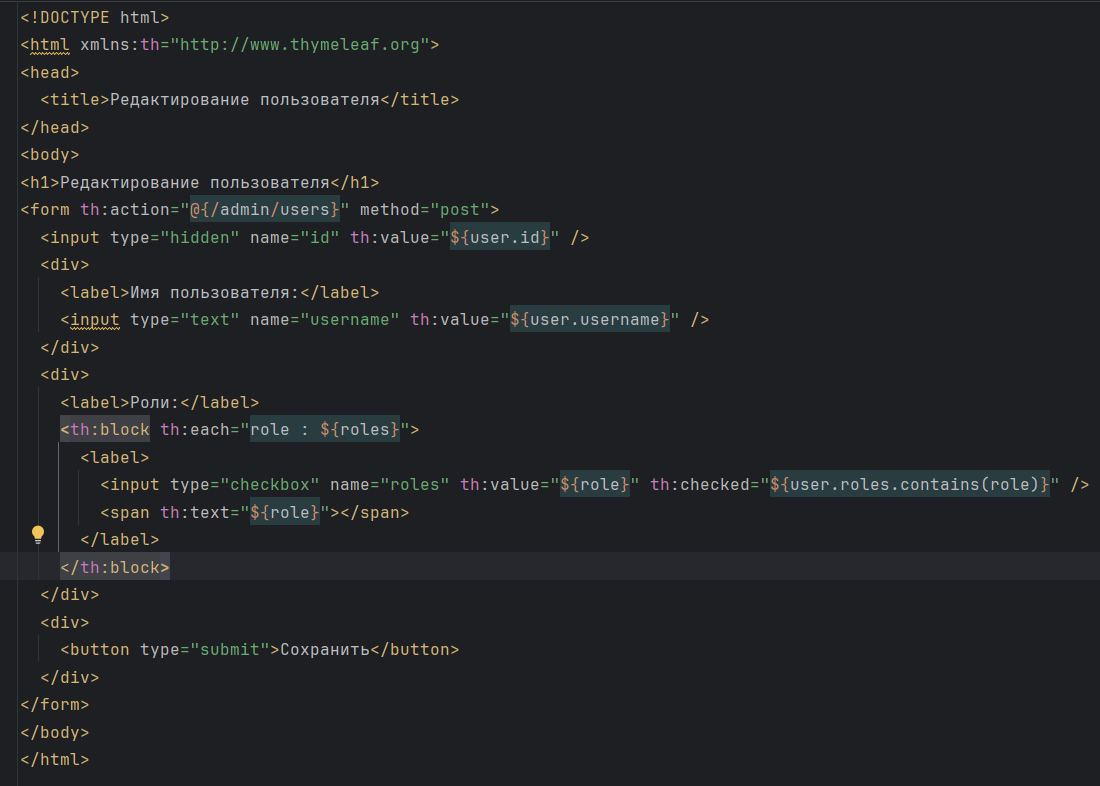


Рисунок 52 – Редактирование пользователя

1. Создание вывода списка пользователей

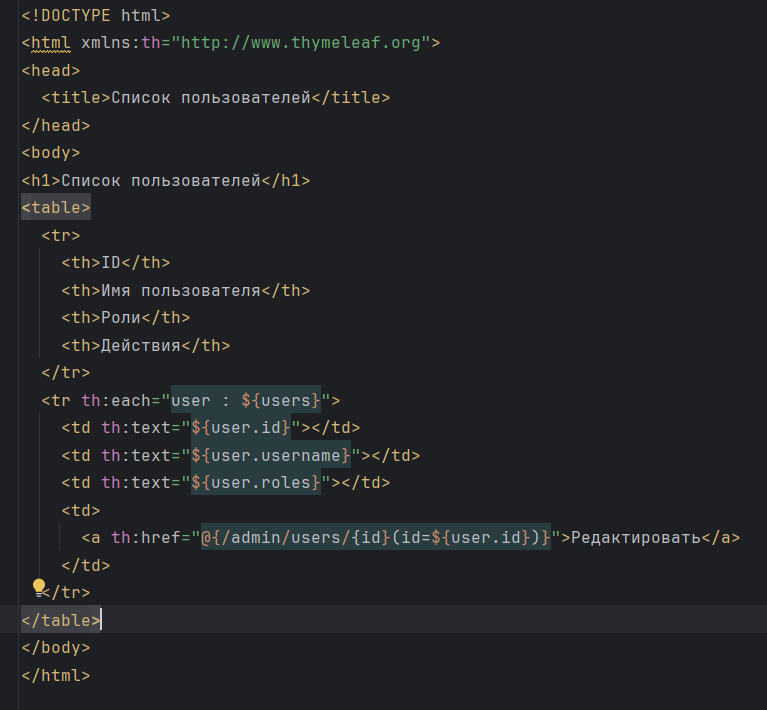


Рисунок 53 – Вывод списка пользователя

1. Добавление security защиты в настройке бд

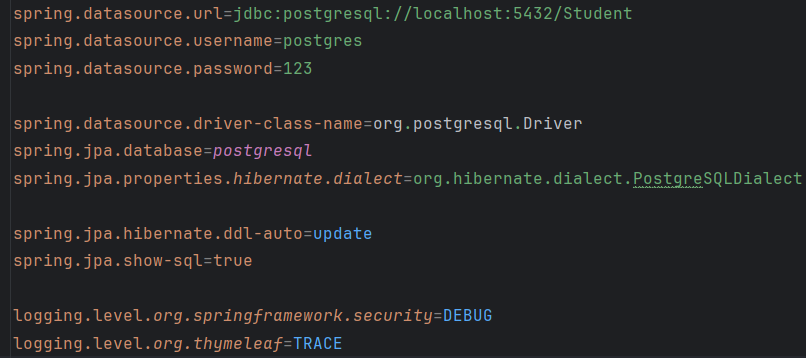


Рисунок 54 – Добавление бд

# Практическая работа № 5

Цель работы: Создание простого REST API, в котором расписаны все операции, а также использованы аннотации.

Ход работы:

1. Создание API

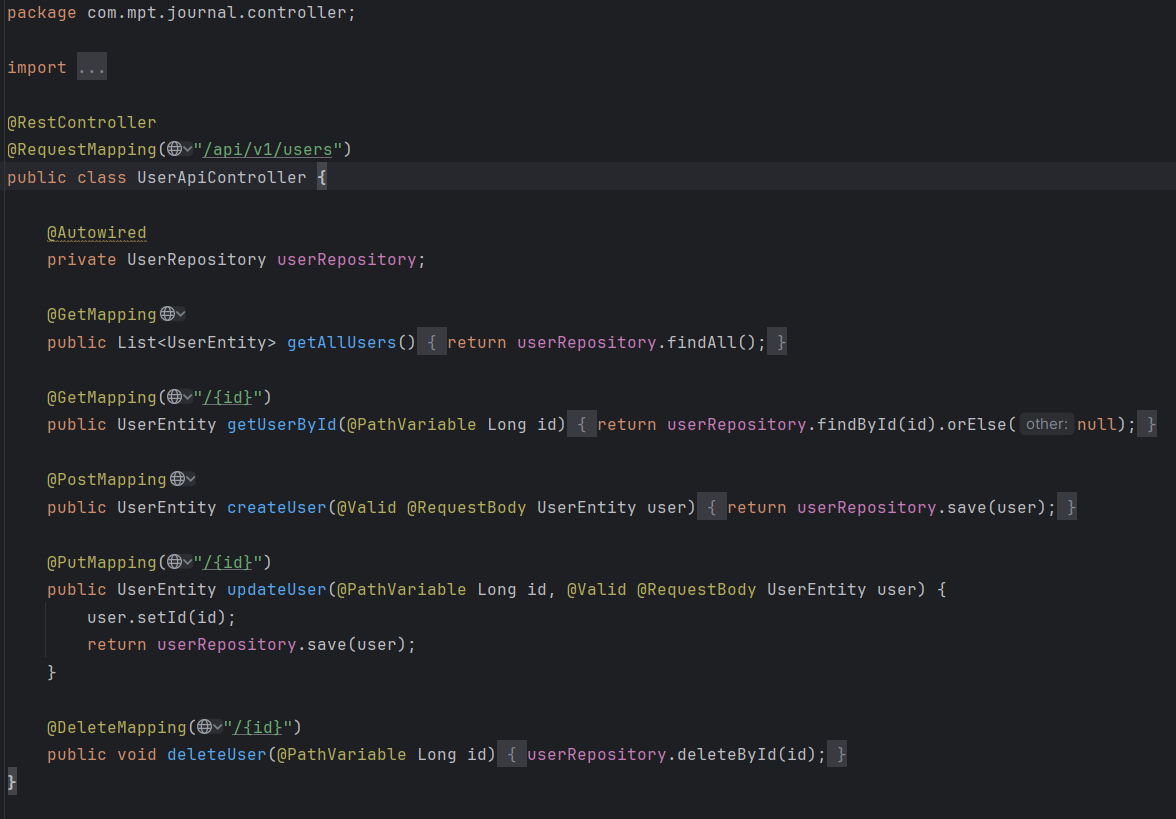


Рисунок 55 - API

1. Тестирование API

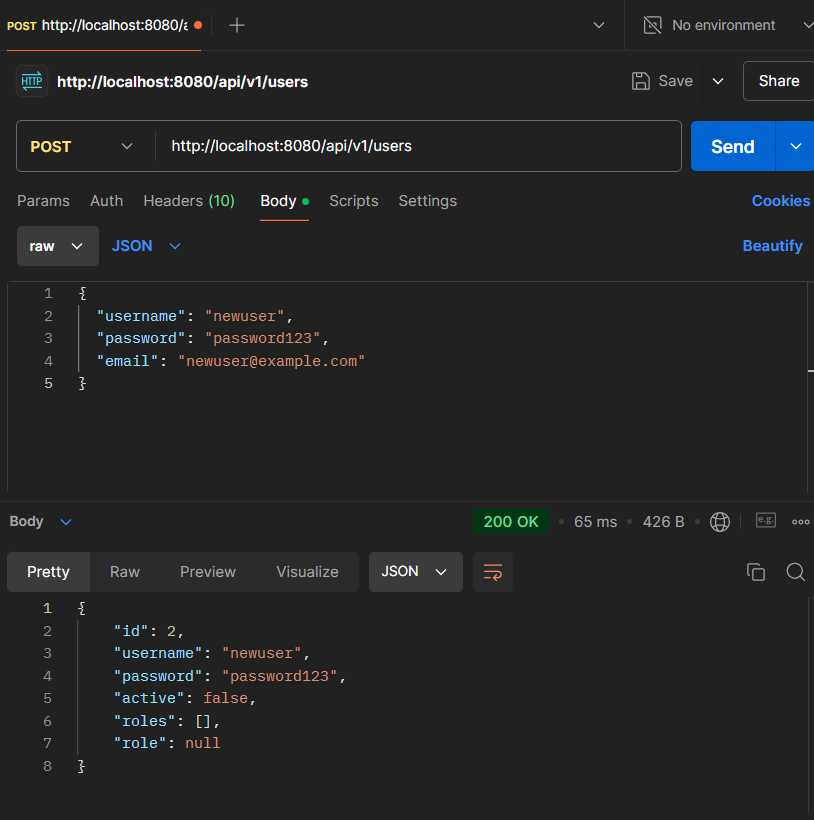


Рисунок 56 – Тестирование



Рисунок 57 – Тестирование

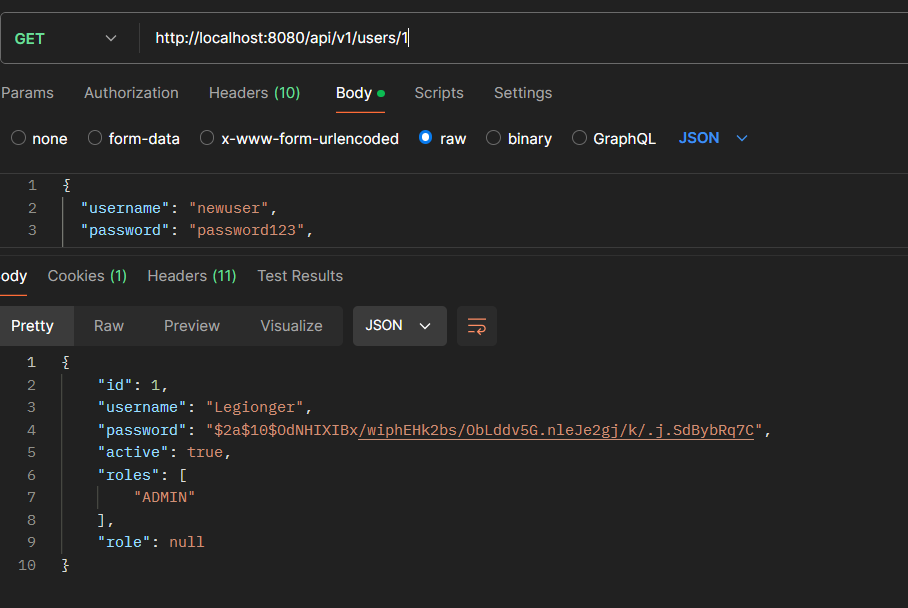


Рисунок 58 – Тестирование

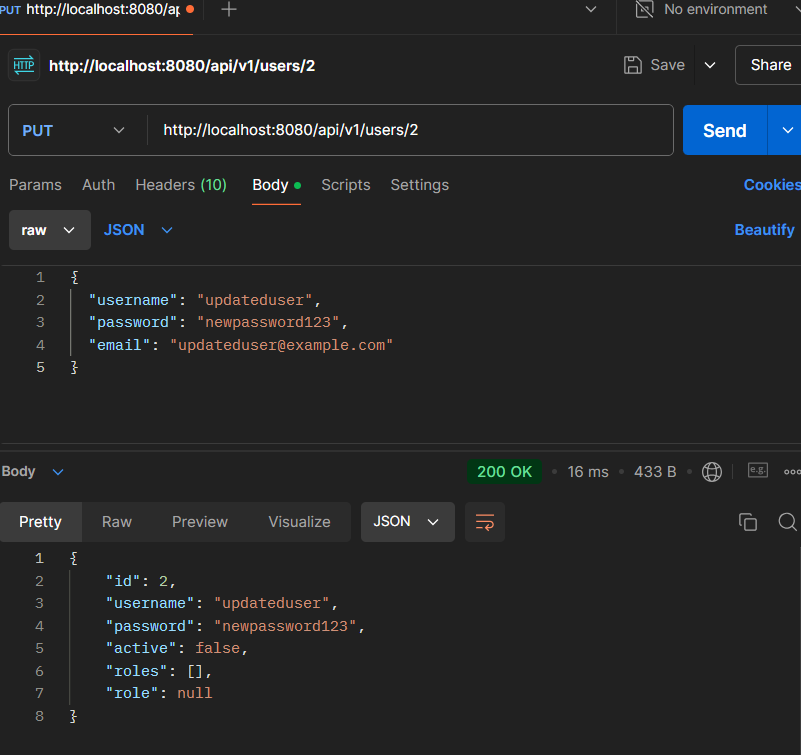


Рисунок 59 – Тестирование

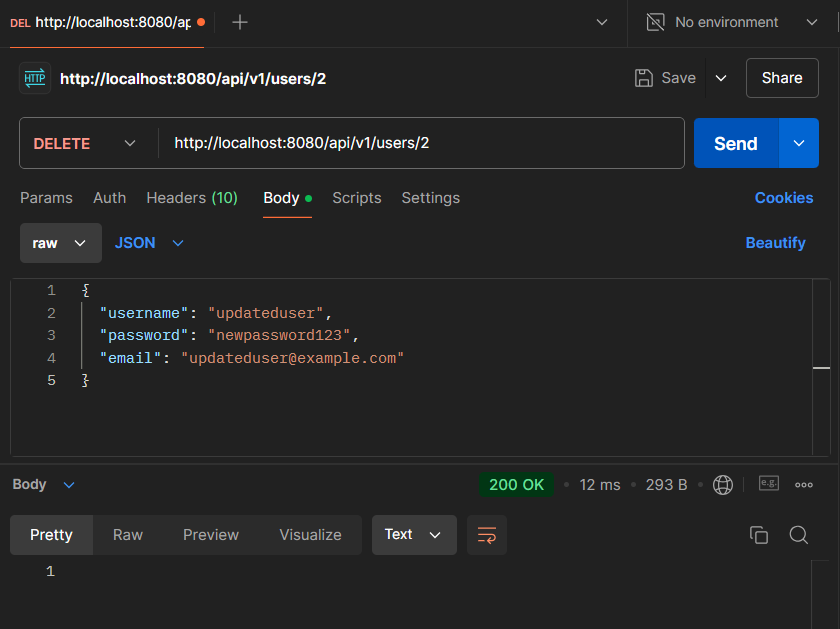


Рисунок 60 – Тестирование

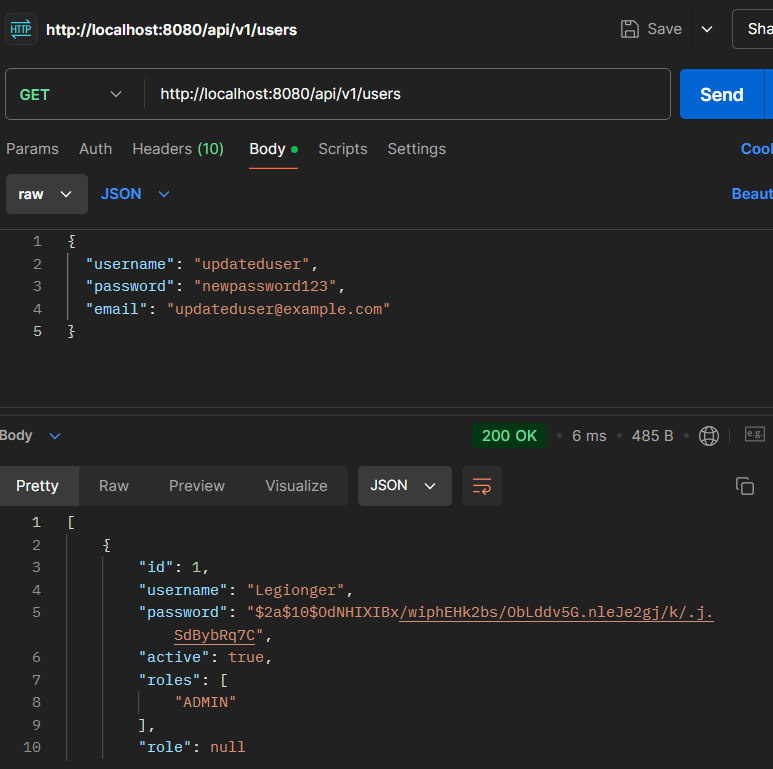


Рисунок 61 - Тестирование

Вывод: Создал API

# Итоговый проект