

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет инфокоммуникационных технологий
Образовательная программа Мобильные и сетевые технологии
Направление подготовки 09.03.03 Прикладная информатика

О Т Ч Е Т

об учебной, ознакомительной практике

Тема задания: *Изучение и интеграция «Java's Spring boot framework» в качестве бэкэнда и Vue.js в качестве фронтэнда и создание веб-приложения*

Обучающийся *Шубхам Кунал, К34401*

Руководитель практики от университета: *Горлушкина Наталия Николаевна, доцент факультета инфокоммуникационных технологий*

Практика пройдена с оценкой _____

Дата _____

Санкт-Петербург
2021

Реферат

Изучение и интеграция «Java's Spring boot framework» в качестве бэкэнда и Vue.js в качестве фронтэнда и создание веб-приложения

Целью моей практики было изучение работы двух технологий: Spring boot и Vue.js, а также интеграция этих двух технологий вместе и использование Postgres в качестве базы данных.

Результатом практики стала разработка веб-приложения для отслеживания tutorиалов, данных лекторами.

Данный отчёт содержит 20 страниц, 12 рисунков и 6 литературных источников.

СОДЕРЖАНИЕ

Реферат.....	2
Введение	4
1. Освоение технологии	5
1.1 Понимание фреймворка Spring Boot	6
1.2 Понимание фреймворка Vue.js	8
1.3 Особенности Vue	9
1.4 Выводы по изучению технологии	10
2. Реализация	11
2.1 Реализация бэкенда: Spring-boot	11
2.2 Использование Postgres в качестве базы данных	14
2.3 Реализация фронтэнда: Vue.js	15
2.4 Результаты реализации	17
3. Заключение	19
4. Список использованных источников	20

Введение

Базой практики являлся национально-исследовательский университет ИТМО. Практика проходила на факультете информационных технологий. В ходе практики были изучены Spring boot и Vue.js, а также спроектирована серверная и фронтенд-архитектура, а также реализовано веб-приложение, которое помогает преподавателям отслеживать свои уроки.

Цель практики - изучение Vue.js и Spring boot. Понять, как работают их различные компоненты, как они интегрированы и как разработать веб-приложение

Для достижения цели были определены следующие задачи:

1. Изучение структуры фреймворков загрузки Vue и Spring.
2. Определение требований к серверной части и начальная разработка серверной части.
3. Определение требований к интерфейсу и начальная разработка интерфейса.
4. Проектирование интерфейса.
5. Реинтеграция и тестирование

При создании этого приложения были использованы следующие технологии:

1. IDE – IntelliJ
2. Бэкэнд – Spring Boot / Java
3. Фронтенд – Vue.js
4. Инспекция и Тестирование – Chrome

Глава 1

Освоение технологии

Задача этапа – Изучить Spring Boot. Изучение специальной литературы и другой научно-технической информации, достижений отечественной и зарубежной науки и техники в соответствующей области знаний.

Освоение всех необходимых технологий проходили в несколько этапов:

- Изучение рекомендованной литературы.
- Изучение документации к выбранному для работы программному обеспечению и платформам.
- Просмотр официальных лекций на Youtube

Все необходимые знания и компетенции были получены в срок, до начала практической деятельности. К началу работы над проектом были освоены все необходимые технологии

1.1 Понимание фреймворка Spring Boot

Чтобы понять фреймворк Spring boot, мы сначала должны понять фреймворк Spring. Как это связано с Spring boot.

Spring Framework - это фреймворк корпоративного уровня с открытым исходным кодом для создания автономных приложений производственного уровня, работающих на виртуальной машине Java (JVM).

Java Spring Boot (Spring Boot) - это фреймворк, который ускоряет и упрощает разработку веб-приложений и микросервисов с помощью Spring Framework благодаря трем основным возможностям:

1. Автоконфигурация
2. «Opinionated» подход к настройке
3. Возможность создавать автономные приложения

Эти функции работают вместе, чтобы предоставить вам инструмент, который позволяет настроить приложение на основе Spring с минимальной конфигурацией и настройкой.

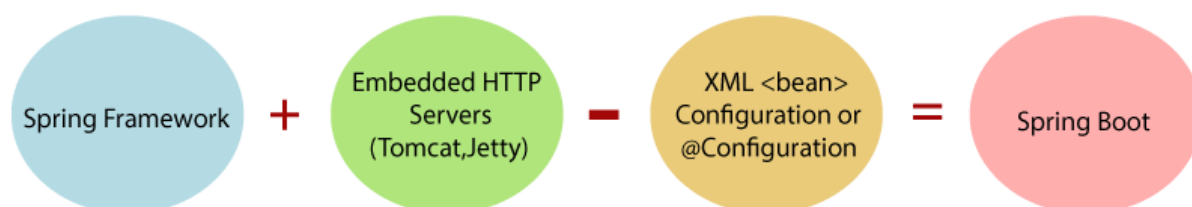


Рис 1: Соотношение Spring и Spring boot

1. Автоконфигурация - Автоконфигурация означает, что приложения инициализируются с заранее установленными зависимостями, которые нам не нужно настраивать вручную. Поскольку Java Spring Boot поставляется со встроенными возможностями автоконфигурации, он автоматически настраивает как базовую платформу Spring Framework, так и сторонние пакеты на основе настроек.
2. «Opinionated» подход к настройке - Spring Boot использует упорный подход к добавлению и настройке стартовых зависимостей, исходя из потребностей проекта. Следуя собственному суждению, Spring Boot выбирает, какие пакеты устанавливать и какие значения по умолчанию использовать, вместо того, чтобы требовать от нас принимать все эти решения самостоятельно и настраивать все вручную.
3. Возможность создавать автономные приложения - Spring Boot помогает разработчикам создавать приложения, которые просто запускаются. В частности, он позволяет нам создавать автономные приложения, которые работают сами по себе, не полагаясь на внешний веб-сервер, путем встраивания веб-сервера, такого как Tomcat или Netty, в наше приложение во время процесса инициализации.

1.2 Понимание фреймворка Vue.js

VueJS - это прогрессивная среда JavaScript, используемая для разработки интерактивных веб-интерфейсов. Основное внимание уделяется части просмотра, и ее очень легко интегрировать с другими проектами и библиотеками.

Одна важная особенность Vue - возможность использовать компоненты. Компоненты - это многократно используемые экземпляры Vue с настраиваемыми элементами HTML. Компоненты можно повторно использовать сколько угодно раз или использовать в другом компоненте, что делает его дочерним компонентом. В компоненте Vue можно использовать данные, вычисление, наблюдение и методы.

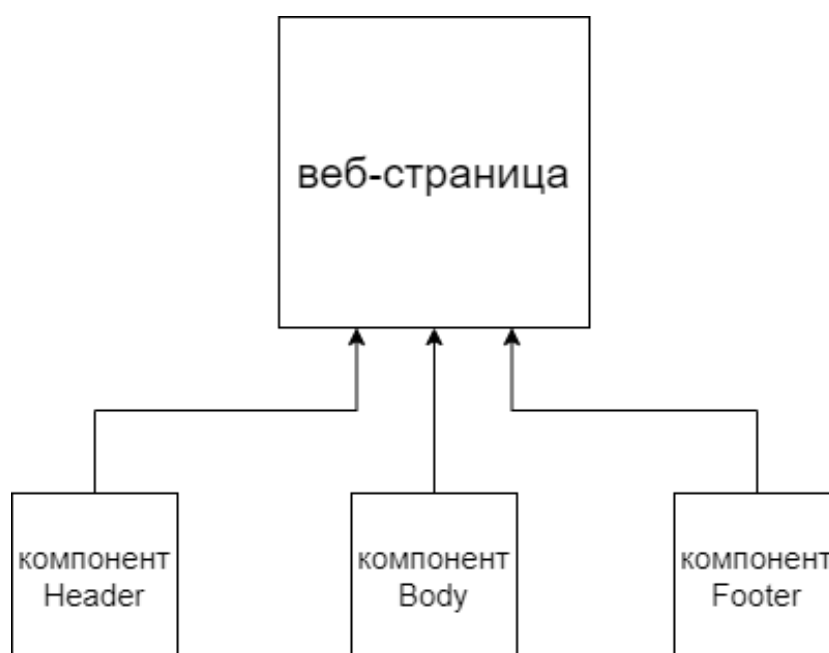


Рис 2: Состав страниц Vue.js

1.3 Особенности Vue

1. Возможность работы с существующим DOM-деревом - Это одна из ключевых особенностей фреймворка
2. Наличие директив для расширения возможностей работы с DOM-деревом - Пользовательские директивы обеспечивают удобный интерфейс для низкоуровневого взаимодействия с элементами дерева DOM.
3. Относительно небольшой размер - Vue.js, несмотря на все свои возможности, имеет наименьший размер JavaScript-кода среди «большой тройки» (Angular, React и Vue.js).
4. Наличие большого количества готовых дополнительных библиотек - У фреймворка их много.

1.4 Выводы по изучению технологии

В результате изучения большого количества материалов по Spring Boot и Vue.js был достигнут определенный уровень знаний. Я хорошо разобрался в этих технологиях и узнал, как проектировать сервер в Spring Boot и реализовывать проекты, интегрируя Spring Boot и Vue.

Благодаря этой практике я познакомился с новым многоуровневым стилем проектирования серверов с использованием Spring Boot.

Глава 2

2.1 Реализация бэкенда: Spring-boot

В проекте было сделано несколько пакетов, разделяющих работу бэкенда: controller, model, repository.

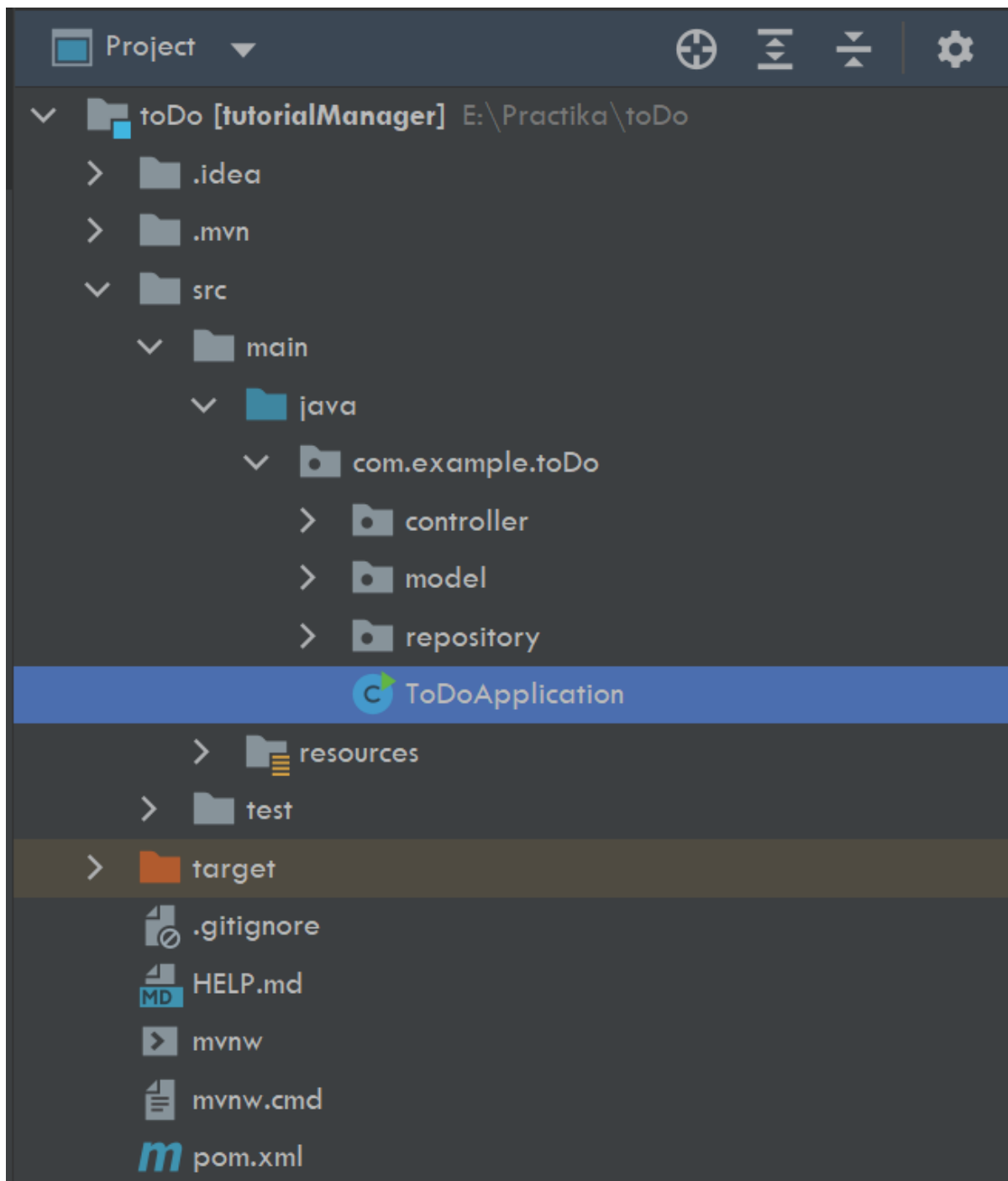


Рис 3: Организация бэкэнд-задач.

Способ организации различных задач отражает философию Spring-boot. В каждом пакете есть классы, выполняющие определенные задачи, и они интегрируются в структуру с помощью «внедрения зависимостей» и аннотаций.

Внедрение зависимостей — это фундаментальный аспект среды Spring, с помощью которого контейнер Spring «внедряет» объекты в другие объекты или «зависимости».

Он допускает «loose coupling» компонентов и переносит ответственность за управление компонентами на контейнер.

Loose coupling: это означает, что классы в основном независимы. Если единственное знание, которое класс А имеет о классе В, это то, что класс В показал через свой интерфейс, то класс А и класс В называются «loosely coupled».

```
public class TutorialController {  
    @Autowired  
    TutorialRepository tutorialRepository;  
}
```

Рис 4: класс TutorialRepository был loosely coupled с классом TutorialController, тем самым достигая внедрения зависимостей

@Autowired используется для автоматического связывания bean-компонента Spring с методами установки, переменной экземпляра и конструктором. Когда мы используем аннотацию @Autowired, контейнер Spring автоматически подключает компонент, сопоставляя тип данных

В бэкенде были созданы следующие пакеты:

1. Controller - определяет конечные точки нашего сервера
2. Model - для определения базовых сущностей серверной части.
3. Repository - единица хранения нашего бэкенда
4. `ToDoApplication.java` - точка входа в наше бэкэнд-приложение

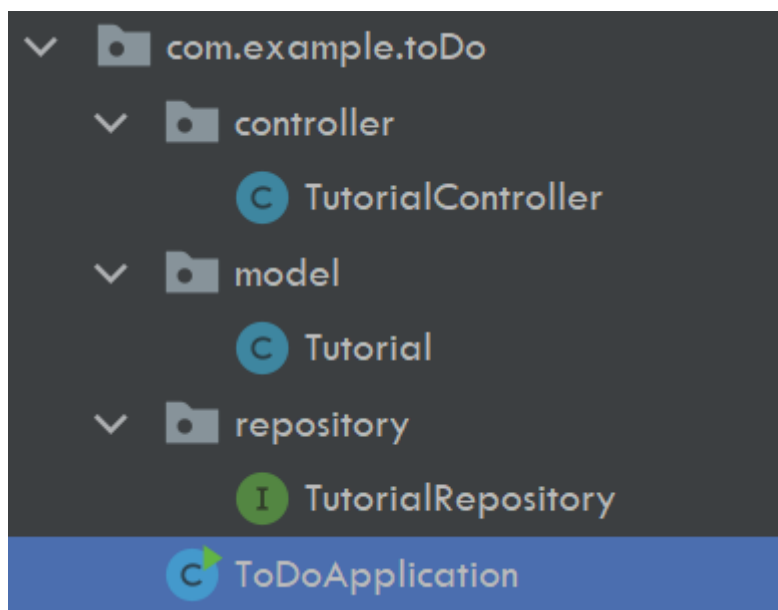
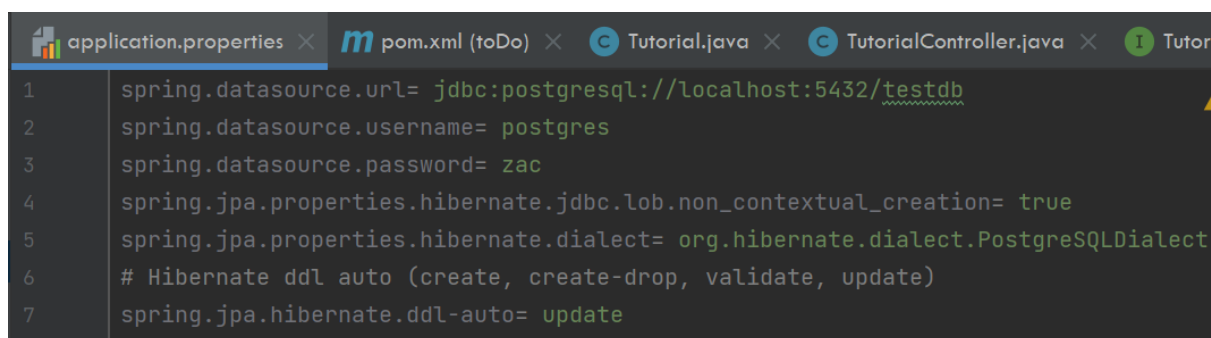


Рис 5: бэкэнд-проект расширен

В `Tutorial.java` была разработана модель `Tutorial`. Ему было присвоено 4 свойства: `id`, `title`, `description`, `published`. Интерфейс `TutorialRepository.java` использовался для создания объектов доступа к данным и хранения экземпляров `Tutorials`. `TutorialController` в пакете контроллера используется для создания API. Этот API будет использоваться интерфейсом в качестве точки доступа.

2.2 Использование Postgres в качестве базы данных

Spring boot предоставляет простой способ интеграции серверной части с базой данных. В папке «ресурсы» в главном каталоге есть файл с именем `application.properties`. Все, что нам нужно сделать, это предоставить подробную информацию о нашей учетной записи postgres, и она готова к использованию.



```
1 spring.datasource.url= jdbc:postgresql://localhost:5432/testdb
2 spring.datasource.username= postgres
3 spring.datasource.password= zac
4 spring.jpa.properties.hibernate.jdbc.lob.non_contextual_creation= true
5 spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.PostgreSQLDialect
6 # Hibernate ddl auto (create, create-drop, validate, update)
7 spring.jpa.hibernate.ddl-auto= update
```

Рис 6: Файла `application.properties` для интеграции базы данных

Этот файл автоматически создает базу данных, если она не существует, и заполняет ее данными, которые мы предоставляем с помощью нашего бэкэнда.

2.3 Реализация фронтэнда: Vue.js

Были разработаны три основных компонента: Tutorial.vue, AddTutorial.vue, TutorialsList.vue

Как следует из названия, AddTutorial.vue используется для добавления учебника в наш список путем отправки модели учебника, которую мы разработали в нашем бэкенде, в наш бэкенд, а затем отправляет данные в базу данных. Tutorial.vue выполняет действия CRUD над учебными пособиями. TutorialsList.vue заполняет все учебные пособия на нашем экране.

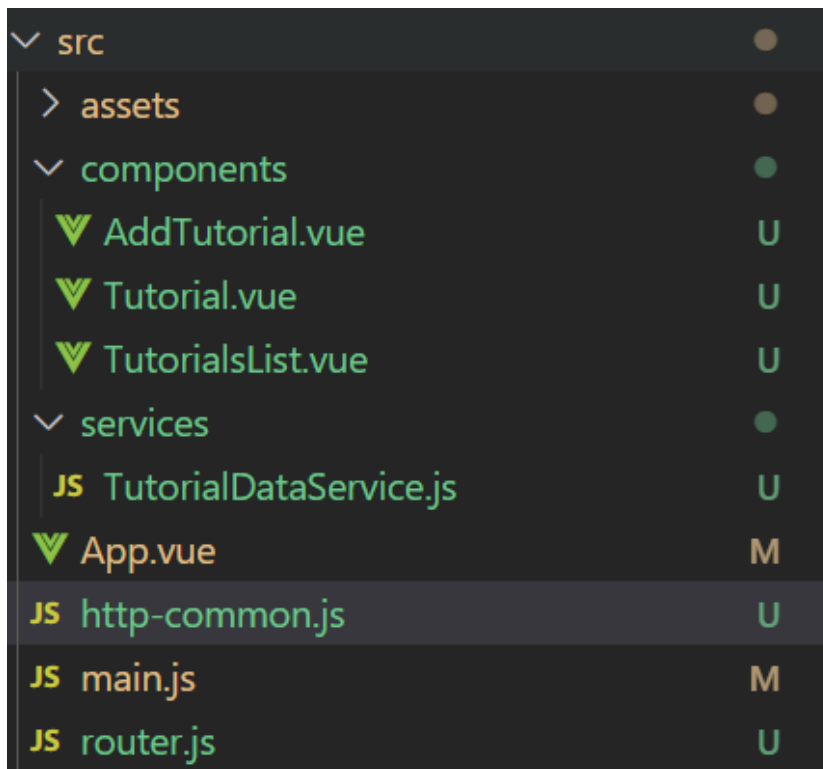


Рис 7: Каталог нашего фронтенда

Каталог assets используется для хранения многократно используемых ресурсов, таких как логотип и изображения и services каталогов были созданы для обеспечения интерфейса для различных служб CRUD.

В нашем проекте использовалась библиотека Axios для работы с API во фронтенде.

```
import axios from "axios";

export default axios.create({
  baseURL: "http://localhost:8080/api",
  headers: {
    "Content-type": "application/json"
  }
});
```

Компонент App представляет собой контейнер с Router-View. У него есть навигационная панель, которая ссылается на пути маршрутов

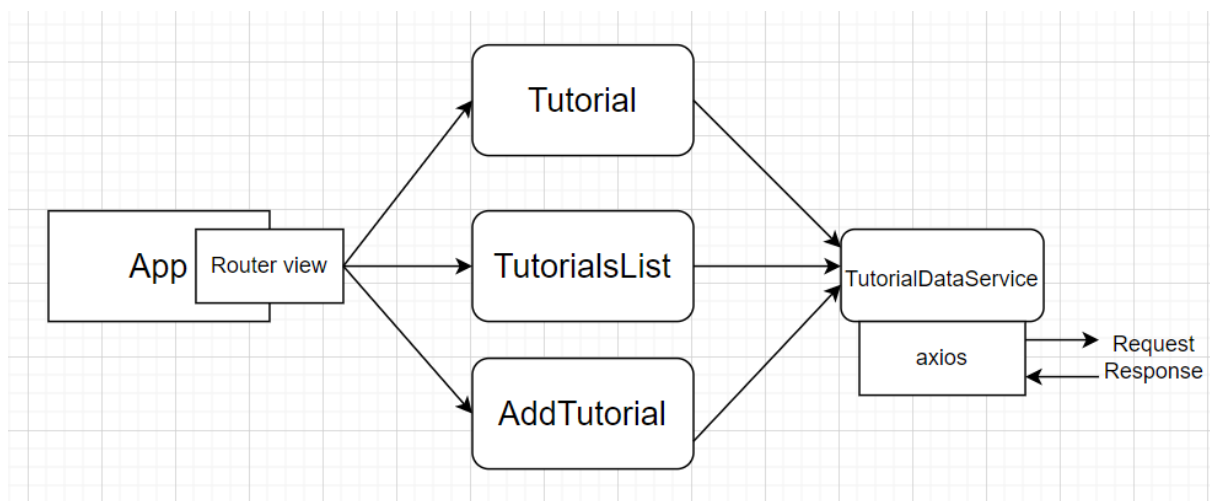


Рис 8: Взаимодействие фронтенда с бэкендом

2.3 Результаты реализации

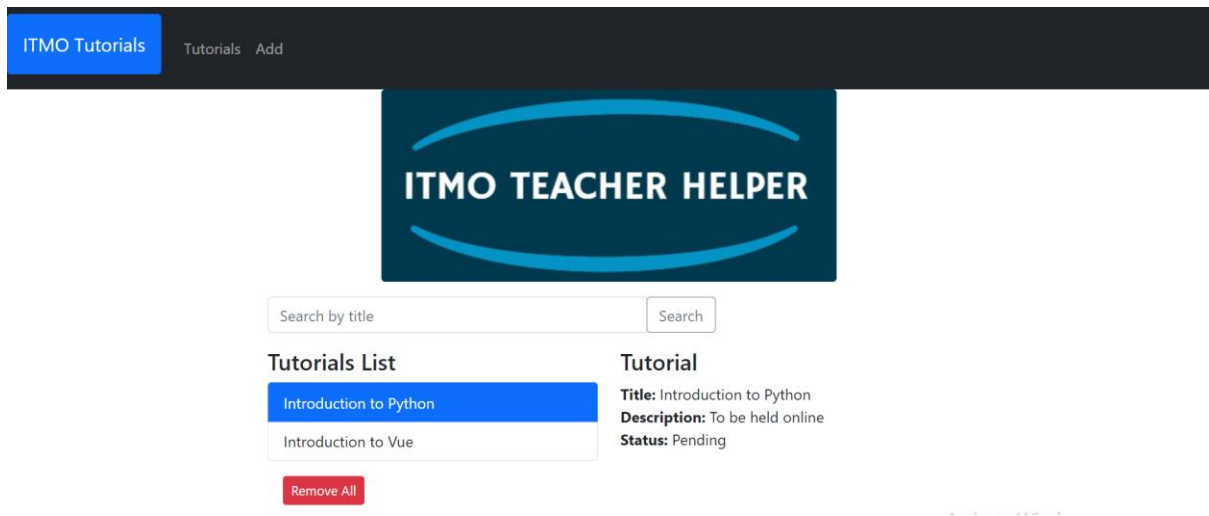


Рис 9: Домашняя страница фронтенда

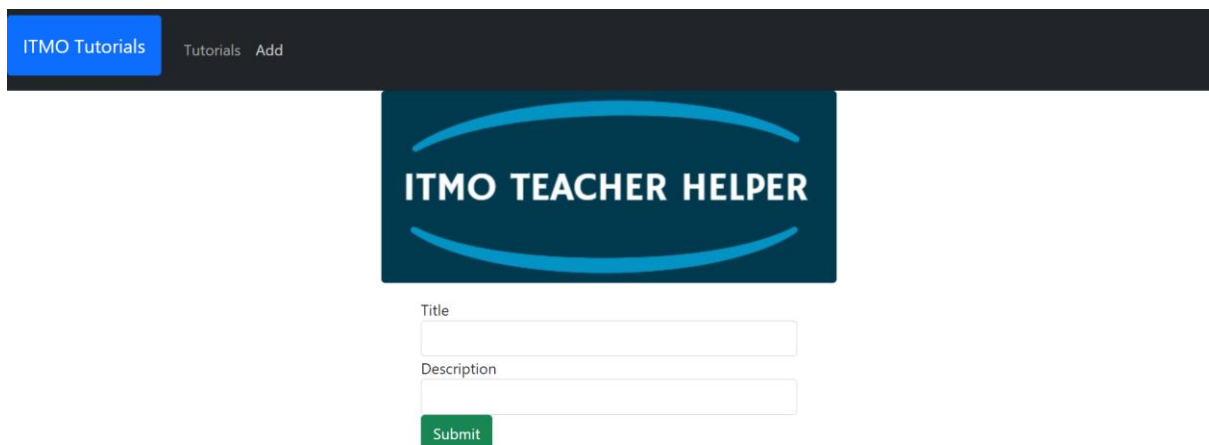


Рис 10: Раздел для добавления учебников.

ITMO Tutorials

Tutorials Add

ITMO TEACHER HELPER

Tutorial

Title

Introduction to Python

Description

To be held online

Status: Pending

Publish Delete Update

Рис 11: Раздел для редактирования tutorиалов.

```
testdb=# \l
               List of databases
  Name | Owner  | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | English_India.1252 | English_India.1252 |
student | postgres | UTF8 | English_India.1252 | English_India.1252 |
template0 | postgres | UTF8 | English_India.1252 | English_India.1252 | =c/postgres +
template1 | postgres | UTF8 | English_India.1252 | English_India.1252 | =c/postgres +
test | postgres | UTF8 | English_India.1252 | English_India.1252 |
testdb | postgres | UTF8 | English_India.1252 | English_India.1252 |
(6 rows)

testdb=# \c testdb
You are now connected to database "testdb" as user "postgres".
testdb=# SELECT * FROM tutorials;
 id | description | published | title
-----+-----+-----+-----
 10 | To be held online | f | Introduction to Python
 11 | to be held offline! | f | Introduction to Uue
(2 rows)

testdb=#
```

Рис 12: Данные успешно передаются базе данных.

Глава 3

Заключение

Во время практики подробно изучил различные аспекты разработки бэкэнд и фронтэнд. Я успешно спроектировал и разработал веб-приложение. Для дальнейшего более детального изучения этих двух технологий я буду создавать все более и более сложные веб-приложения, использующие эти технологии в будущем.

Благодаря этой практике я решил в будущем быть бэкэнд-программистом.

Приложения были протестированы и работает хорошо

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. JavaTPoint <https://www.javatpoint.com/spring-boot-tutorial2>
2. Tutorialspoint https://www.tutorialspoint.com/vuejs/vuejs_routing.htm
3. Official View Documentation <https://vuejs.org/v2/guide/>
4. Spring Boot Official reference <https://docs.spring.io/spring-boot/docs/current/reference/pdf/spring-boot-reference.pdf>
5. Web Server Programming by Neil Gray
<http://docshare03.docshare.tips/files/4994/49944158.pdf>
6. Postgres tutorials - <https://www.tutorialspoint.com/postgresql/index.htm>