

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”

Факультет _____

Образовательная программа _____

Направление подготовки (специальность) 45.03.04 - Интеллектуальные системы в гуманитарной сфере

О Т Ч Е Т

по курсовой работе

Тема задания: Реализация web-сервисов средствами Django REST framework, Vue.js, Muse-UI

Обучающийся Фроловс А.. К3342

Руководитель: Говоров А.И., Чунаев А.В, Антонов М.Б.

Оценка _____

Дата _____

Санкт-Петербург
20 20

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ	4
1.1 Описание варианта	4
1.2 Описание предметной области и функциональные требования	5
ГЛАВА 2. ОПИСАНИЕ СЕРВЕРНОЙ ЧАСТИ.....	7
2.1 Модель данных.....	7
2.2 Сериализация и создание отображений	8
2.3 Разработанные интерфейсы в Django REST	9
ГЛАВА 3. ОПИСАНИЕ КЛИЕНТСКОЙ ЧАСТИ.....	14
3.1 Описание средств разработки клиентской части.....	14
3.2 Разработанные клиентские интерфейсы Vue.....	14
ЗАКЛЮЧЕНИЕ.....	29
СПИСОК ЛИТЕРАТУРЫ	30

ВВЕДЕНИЕ

Создание web-приложений сегодня считается одним из перспективных направлений деятельности для многих компаний, занятых в сфере высокотехнологичных цифровых и компьютерных технологий. Отказ от традиционного программного обеспечения и перенос бизнес-инструментов на веб — это тренд, который нельзя упускать.

Разработка web-приложений стало актуальной темой для многих специализированных компаний, и одновременно доступной для простых пользователей. Одной из главных служб Интернет является WWW-служба (World • Wide Web - Всемирная паутина), которая используется для доставки гипертекстовых документов. Одним из основных технологических направлений Интернет является особый класс программ, получивший название "вебприложения", работа с которыми происходит посредством WWW. Вебприложения относятся к классу систем "клиент-сервер", в которых в качестве клиентского программного обеспечения используется веб-браузер, а в роли сервера выступает веб-сервер. [1]

Целью данной курсовой работы является создание web-приложения согласно выбранному варианту с использованием современных средств web-разработки, таких как Django REST, Vue.js и Muse-ui.

Задачи, которые должны быть выполнены в рамках курсовой работы:

- Изучить предметную область
- Выявить функциональные требования
- Разработать web-приложение
- Провести контейнеризацию и оркестрацию web-приложения

ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И ФУНКЦИОНАЛЬНЫХ ТРЕБОВАНИЙ

1.1 Описание варианта

Создать программную систему, предназначенную для завуча школы. Она должна обеспечивать хранение сведений о каждом учителе, классном руководстве, о предметах, которые он преподает в заданный период, номере закрепленного за ним кабинета, о расписании занятий. Существуют учителя, которые не имеют собственного кабинета. Об учениках должны храниться следующие сведения: фамилия и имя, в каком классе учится, какую оценку имеет в текущей четверти по каждому предмету.

Завуч должен иметь возможность добавить сведения о новом учителе или ученике, внести в базу данных четвертные оценки учеников каждого класса по каждому предмету, удалить данные об уволившемся учителе и отчисленном из школы ученике, внести изменения в данные об учителях и учениках, в том числе поменять оценку ученика по тому или иному предмету. В задачу завуча входит также составление расписания.

Завучу могут потребоваться следующие сведения:

- Какой предмет будет в заданном классе, в заданный день недели на заданном уроке?

- Сколько учителей преподает каждую из дисциплин в школе?
- Список учителей, преподающих те же предметы, что и учитель, ведущий информатику в заданном классе.
- Сколько мальчиков и девочек в каждом классе?
- Сколько кабинетов в школе для базовых и профильных дисциплин?

Необходимо предусмотреть возможность получения документа, представляющего собой отчет об успеваемости заданного класса. Отчет включает сведения об успеваемости за четверть по каждому предмету. Необходимо подсчитать средний балл по каждому предмету, по классу в целом, указать общее количество учеников в классе. Для класса указать классного руководителя.

1.2 Описание предметной области и функциональные требования

Согласно варианту, в качестве предметной области выбрана абстрактная школа. Информационная система для данной сферы занимается управлением стандартными для школы процессами: принятием и исключением учеников, наймом и увольнением учителей, составлением расписания и выставлением четвертных оценок. Данные об учениках и учителях содержат личную информацию, которая не должна быть доступна любому пользователю вебприложения, следовательно, необходимо реализовать авторизацию и регистрацию завуча в системе. Таким образом, нужно было разработать web-приложение, которое должно обладать следующим функционалом:

Для пользователя:

1. Авторизация и регистрация в web-приложении.
2. Просмотр информации об учителях, учениках, классах, расписании и оценках.
3. Возможность добавления/изменения/удаления учителя, ученика, записи в расписании и оценки.
4. Получение результатов запросов. 5. Получение отчета по выбранному классу.

Распределение всего функционала показано на use-case диаграмме, представленной на рисунке 1.

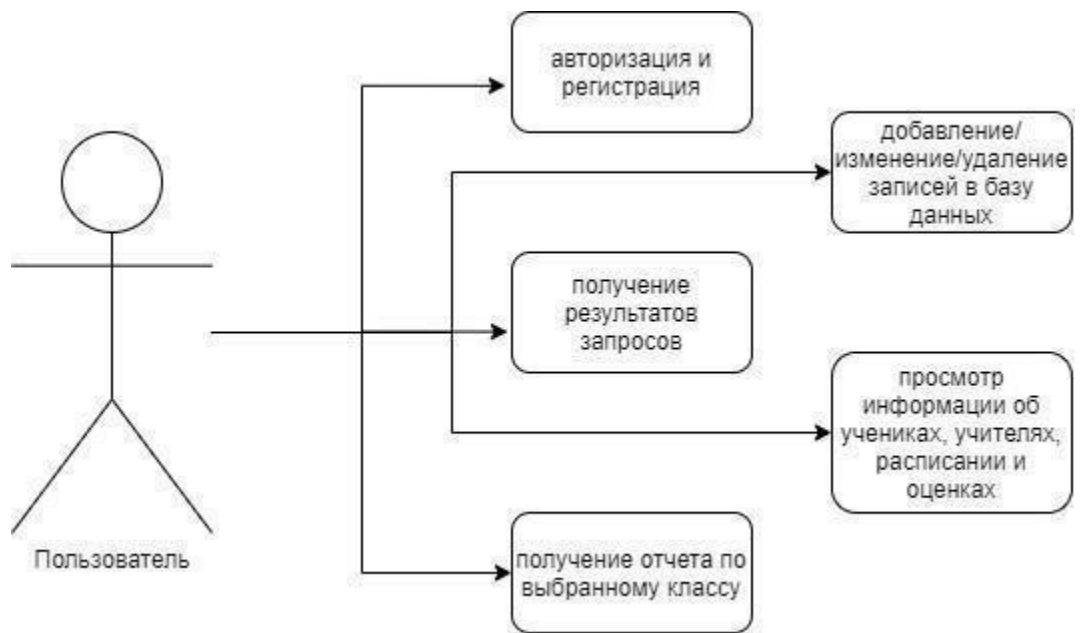


Рисунок 1 – Use-case диаграмма

ГЛАВА 2. ОПИСАНИЕ СЕРВЕРНОЙ ЧАСТИ

2.1 Модель данных

Согласно варианту и выявленным функциональным требованиям была создана модель данных, представленная на рисунке 2.

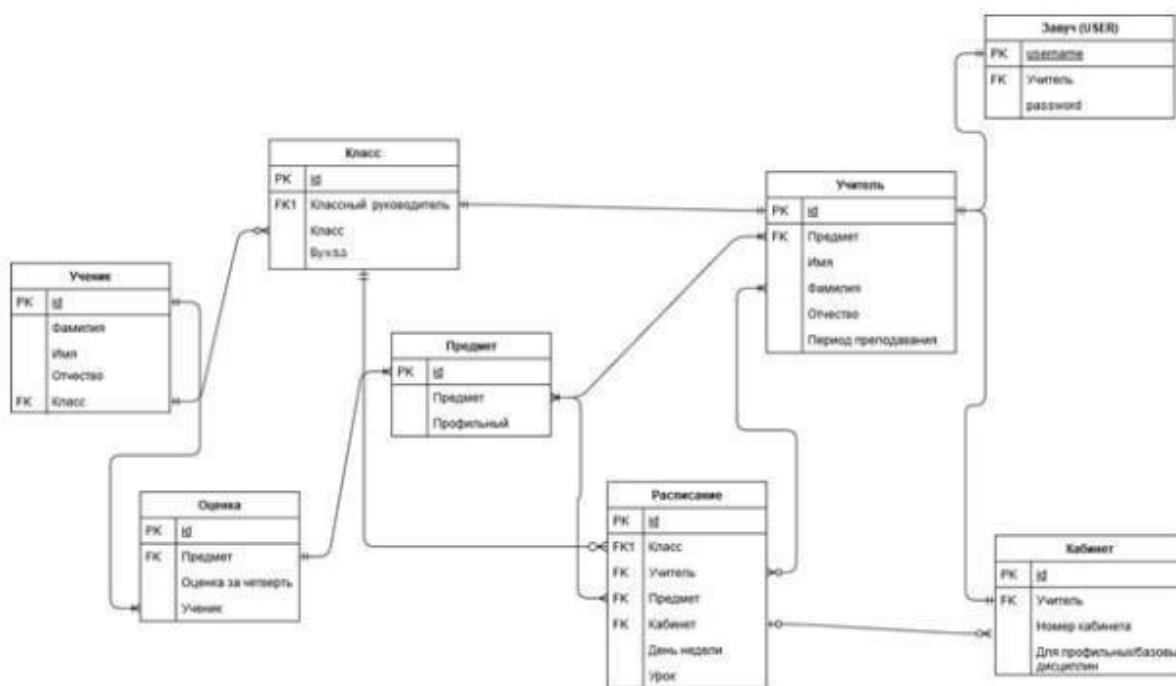


Рисунок 2 – модель данных

Данная модель данных содержит 8 сущностей, между которыми установлены отношения один-ко-многим, многие-ко-многим и один-к-одному.

Для управления базой данных была выбрана PostgreSQL – объектно-реляционная система управления базами данных (ОРСУБД, ORDBMS), основанная на POSTGRES, Version 12.3.

После подключения созданной на сервере Postgre базы данных к бэкенду Django были созданы следующие модели в файле models.py:

- Class Subject – Предмет – данная модель содержит информацию о названии предмета и является он профильным, или базовым.

- Class Teacher – Учитель – данная модель содержит информацию об учителе: его имя, фамилию, отчество, предмет, который он преподает, и в течение какого периода он работает в школе.
- Class Room – Кабинет – данная модель содержит информацию о кабинете, ответственном за него учителе и является он кабинетом для профильных, или базовых, дисциплин.
- Class Nclass – Класс – данная модель содержит информацию о классе: его номер и литера; учителя, который является классным руководителем.
- Class Children – Ученик – данная модель содержит информацию об ученике: фамилию, имя, отчество, пол и в каком классе он учится.
- Class Grade – Оценка – данная модель содержит информацию о семестровых оценках каждого ученика по определенному предмету.
- Class Timetable – Расписание – данная модель содержит информацию о записи в расписании: какой учитель где и когда ведет какой предмет у какого класса.

2.2 Сериализация и создание отображений

Среда сериализации Django предоставляет механизм для «перевода» моделей Django в другие форматы. Обычно эти другие форматы основаны на тексте и используются для отправки данных Django, но сериализатор может обрабатывать любой формат (текстовый или нет) [2]. Использование сериализаторов в данной работе необходимо для того, чтобы серверная часть, написанная на Django REST, могла без проблем обмениваться данными с клиентской частью, написанной на Vue.js. Сериализация также помогает сделать вывод данных из моделей более понятным для пользователей, например, с помощью функции `related_name` можно заменить значение внешнего ключа, которое по умолчанию является `id`, на другое поле модели, которая является внешним ключом.

Отображения для запросов были созданы с помощью встроенной в платформу REST абстракции для работы с ViewSets, которая позволяет разработчику сконцентрироваться на моделировании состояния и взаимодействий API и оставить обработку URL-адреса автоматически, основываясь на общих соглашениях.

Класс `ViewSet` - это просто тип представления на основе классов, который не предоставляет никаких обработчиков методов, таких как `.get ()` или `.post ()`, и вместо этого предоставляет такие действия, как `.list ()` и `.create ()`. Обработчики метода для `ViewSet` связаны только с соответствующими действиями в момент завершения представления, используя метод `.as_view ()`. [3]. Использование классов `ViewSet` позволяет практически сразу получить CRUD(Create-Update-Delete) для любой модели с использованием одного, или нескольких, если это необходимо, сериализаторов.

Для создания отображений, отвечающих за выполнение запросов и предоставления отчета, был выбран класс `APIView`. Использование класса `APIView` во многом аналогично использованию обычного класса `View`, как обычно, входящий запрос отправляется соответствующему методу-обработчику, например `.get ()` или `.post ()`. Кроме того, в классе может быть установлено несколько атрибутов, которые управляют различными аспектами политики API [4].

2.3 Разработанные интерфейсы в Django REST

Далее представлены некоторые из разработанных интерфейсов (endpoint) в Django REST.

1. Вывод всех учителей

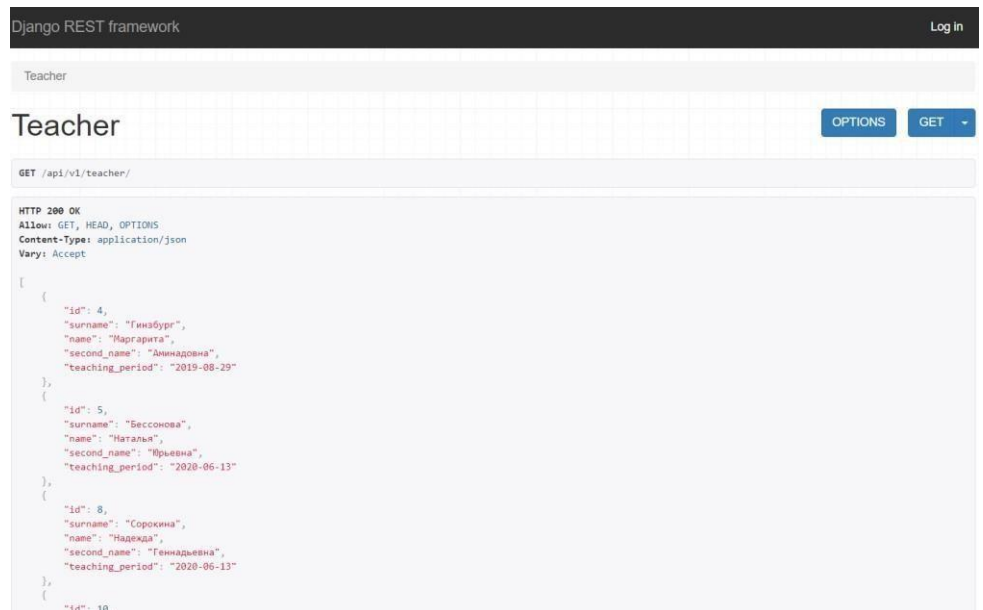


Рисунок 3 – вывод всех учителей

2. Добавление нового учителя

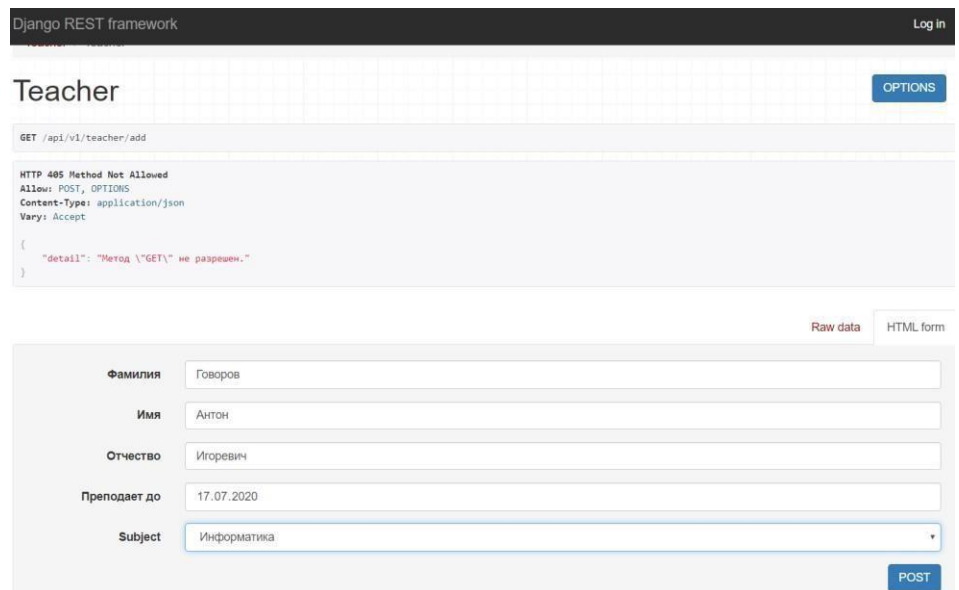


Рисунок 4 –добавление нового учителя

3. Изменение данных в модели Учитель

Teacher

OPTIONS

GET /api/v1/teacher/5/update

HTTP 405 Method Not Allowed
 Allow: POST, OPTIONS
 Content-Type: application/json
 Vary: Accept

```
{
  "detail": "Метод \"GET\" не разрешен."
}
```

Raw data HTML form

Фамилия

Имя

Отчество

Преподает до

POST

Рисунок 5 – изменение данных об определенном учителе

4. Удаление учителя

Teacher

DELETE OPTIONS

GET /api/v1/teacher/5/delete

HTTP 405 Method Not Allowed
 Allow: DELETE, OPTIONS
 Content-Type: application/json
 Vary: Accept

```
{
  "detail": "Метод \"GET\" не разрешен."
}
```

Рисунок 6 – удаление выбранного учителя 5. Вывод всех учеников

Children

OPTIONS GET

GET /api/v1/children/

HTTP 200 OK
 Allow: GET, HEAD, OPTIONS
 Content-Type: application/json
 Vary: Accept

```
{
  [
    {
      "id": 8,
      "surname": "Иждова",
      "name": "Анастасия",
      "second_name": "Анатолевна"
    },
    {
      "id": 10,
      "surname": "Иванов",
      "name": "Иван",
      "second_name": "Александрович"
    },
    {
      "id": 11,
      "surname": "Клементьева",
      "name": "Екатерина",
      "second_name": "Николаевна"
    },
    {
      "id": 12,
      "surname": "Николаева",
      "name": "Ангелика",
      "second_name": "Леонидовна"
    },
    {
      "id": 13,

```

Рисунок 7 – вывод всех учеников 6.

Добавление нового ученика

Children

OPTIONS

GET /api/v1/children/add

HTTP 405 Method Not Allowed

Allow: POST, OPTIONS

Content-Type: application/json

Vary: Accept

{

"detail": "Метод \"GET\" не разрешен."

}

Raw data

HTML form

Фамилия

Имя

Отчество

Пол

Nclass

1 Б

POST

Рисунок 8 – добавление нового ученика

7. Информация об определенном классе

GET /api/v1/nclass/5/

HTTP 200 OK

Allow: GET, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

{

"id": 5,

"teacher": "Луцкий",

"children": [

{

" "id": 16,

" "surname": "Семёнова",

" "name": "Анна",

" "second_name": "Александровна"

},

{

" "id": 17,

" "surname": "Терентьев",

" "name": "Юрий",

" "second_name": "Александрович"

},

{

" "id": 18,

" "surname": "Фёдоров",

" "name": "Владимир",

" "second_name": "Николаевич"

},

{

" "id": 19,

" "surname": "Александрова",

" "name": "Ассоль",

" "second_name": "Львовна"

},

]

}

Рисунок 9 – вывод информации об определенном классе

8. Добавление записи в расписание

Timetable

OPTIONS

GET /api/v1/timetable/add

```
HTTP 405 Method Not Allowed
Allow: POST, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "detail": "Метод \"GET\" не разрешен."
}
```

Raw data

HTML form

Урок	1-8:30-9:10
День недели	Четверг
Nclass name	1 Б
Предмет	Русский язык
Учитель	Гинзбург Маргарита Аминовна
Кабинет	123 кабинет

POST

Рисунок 10 – добавление новой записи в расписание

ГЛАВА 3. ОПИСАНИЕ КЛИЕНТСКОЙ ЧАСТИ

3.1 Описание средств разработки клиентской части

Для создания клиентской части был использован JavaScript-фреймворк Vue.js и его библиотека Muse-ui. Vue.js — это JavaScript библиотека для создания веб-интерфейсов с использованием шаблона архитектуры MVVM (Model-View-ViewModel).

Поскольку Vue работает только на «уровне представления» и не используется для промежуточного программного обеспечения и бэкэнда, он может легко интегрироваться с другими проектами и библиотеками. Vue.js содержит широкую функциональность для уровня представлений и может использоваться для создания мощных одностраничных вебприложений.

3.2 Разработанные клиентские интерфейсы Vue

Согласно заданию и созданной архитектуре web-приложения, были созданы следующие интерфейсы Vue.

1. Главная страница

На главной странице присутствует изображение школы.

Скриншот представлен на рисунке 11.



Рисунок 11 –Главная страница

2. Список всех учителей школы

Данный интерфейс представляет собой список всех учителей школы. Каждая строка списка – это активная ссылка на досье выбранного учителя. При нажатии на кнопку «Добавить учителя» появляется модальное окно Скриншот представлен на рисунке 12.

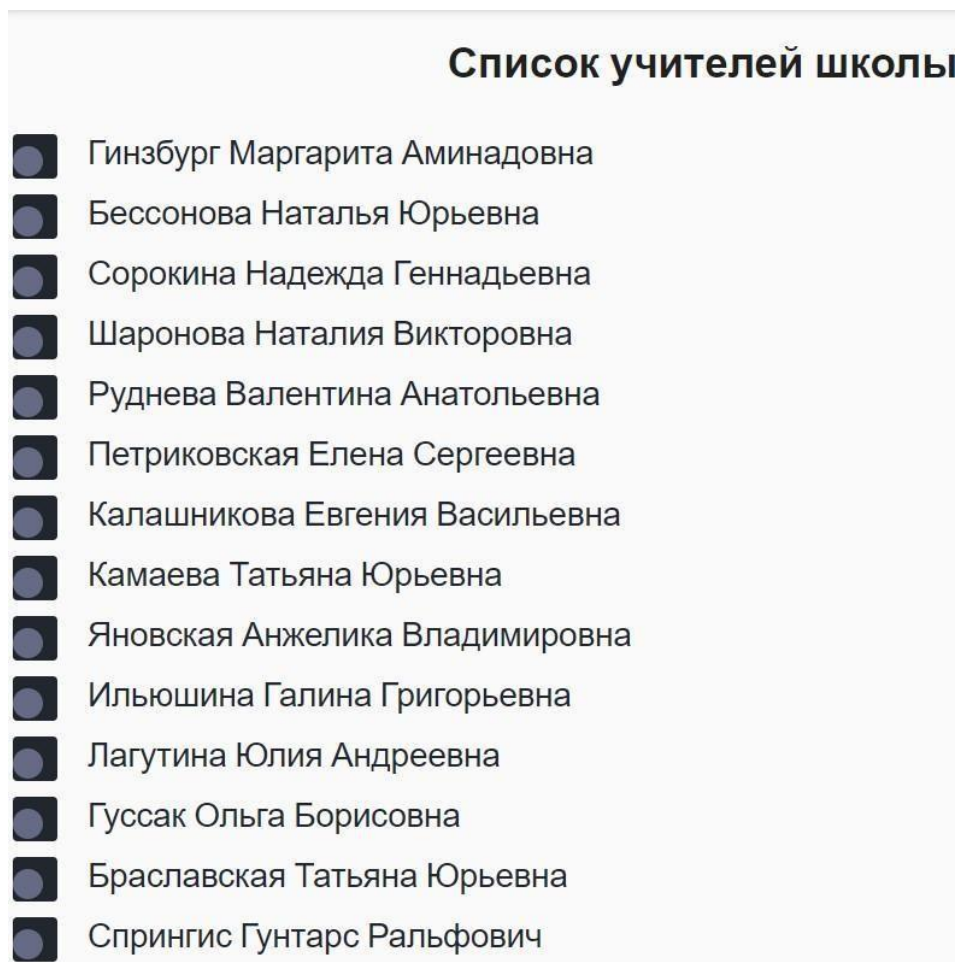


Рисунок 12 –Список всех учителей школы

3. Добавление нового учителя

Данное модельное окно имеет форму создания новой записи в модели «Учитель».

Скриншот представлен на рисунке 13.

Добавление учителя

X

Фамилия

Имя

Отчество

Предмет

Преподает

до Год-
Месяц-День

Добавить

Рисунок 13 –модальное окно «Добавление нового учителя»

4. Досье учителя

Данный интерфейс представляет собой таблицу с данными о выбранном учителе. При нажатии на кнопку «Удалить учителя» данный учитель удаляется из базы данных и происходит перенаправление на страницу «Учителя». Скриншот представлен на рисунке 14.

Досье учителя	
Фамилия	Спрингис
Имя	Гунтарс
Отчество	Ральфович
Предмет	Труд
Имеет классное руководство	6 "Г"
Личный кабинет	Кабинет №45
Преподает до	2020-06-13
<div>ИЗМЕНИТЬ ДАННЫЕ ОБ УЧИТЕЛЕ</div> <div>УДАЛИТЬ УЧИТЕЛЯ</div>	

Рисунок 14 –Досье учителя 5.

Изменение данных об учителе

При нажатии на кнопку «Изменить данные об учителе» в таблице с данными об учителе появляются поля ввода. При нажатии на кнопку «Внести изменения» данные о выбранном учителе изменяются и страница обновляется. Скриншот представлен на рисунке 15.

Фамилия	Спрингис Спрингис
Имя	Гунтарс Гундарс
Отчество	Ральфович Карлович
Предмет	Труд
Имеет классное руководство	6 "Г"
Личный кабинет	Кабинет №45
Преподает до	2020-06-13 2020-07-13

ВНЕСТИ ИЗМЕНЕНИЯ

УДАЛИТЬ УЧИТЕЛЯ

Рисунок 15 –изменение данных об учителе

6. Список классов

Данный интерфейс представляет собой список всех классов школы. При нажатии на кнопку «Показать учеников» происходит перенаправление на страницу «Ученики», при нажатии на «Показать расписание» - на страницу «Расписание».

Скриншот представлен на рисунке 16.



Рисунок 16 –Список классов

7. Список учеников выбранного класса

Данный интерфейс представляет собой список учеников выбранного класса. Каждая строка списка – это активная ссылка на досье выбранного ученика. При нажатии на кнопку «Добавить ученика» появляется модальное окно Скриншот представлен на рисунке 17.



Рисунок 17 –Список учеников выбранного класса

8. Добавление нового ученика

Данное модельное окно имеет форму создания новой записи в модели «Ученик».

Скриншот представлен на рисунке 18.

Добавление ученика

Фамилия

Имя

Отчество

Пол ☐ Мужской ☐ Женский

Класс

Добавить

Рисунок 18 –модальное окно «Добавление нового ученика»

9. Досье ученика

Данный интерфейс представляет собой таблицу с данными о выбранном ученике. При нажатии на кнопку «Удалить ученика» данный ученик удаляется из базы данных и происходит перенаправление на страницу «Ученики школы».

Скриншот представлен на рисунке 19.

Фамилия	Слётов
Имя	Гандельман
Отчество	Андреевич
Пол	Мужской
Класс	1

Четвертные оценки

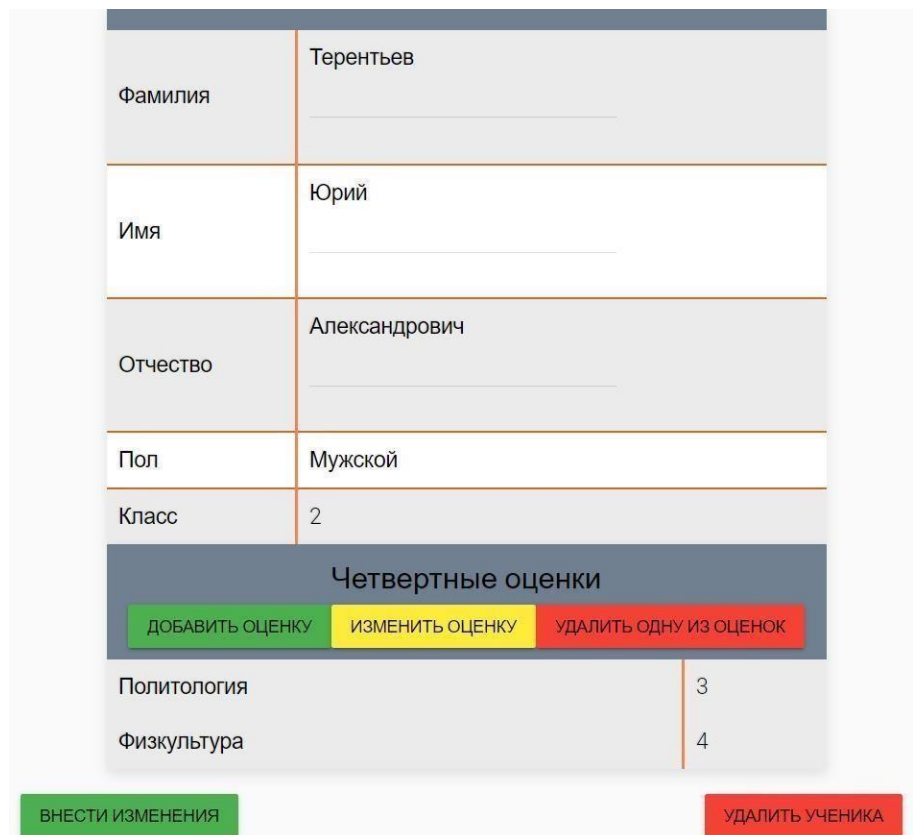
ДОБАВИТЬ ОЦЕНКУ ИЗМЕНИТЬ ОЦЕНКУ УДАЛИТЬ ОДНУ ИЗ ОЦЕНОК

ИЗМЕНИТЬ ДАННЫЕ ОБ УЧЕНИКЕ УДАЛИТЬ УЧЕНИКА

Рисунок 19 –Досье ученика

10. Изменение данных об ученике

При нажатии на кнопку «Изменить данные об ученике» в таблице с данными об ученике появляются поля ввода. При нажатии на кнопку «Внести изменения» данные о выбранном ученике изменяются и страница обновляется. Скриншот представлен на рисунке 20.



Фамилия	Терентьев
Имя	Юрий
Отчество	Александрович
Пол	Мужской
Класс	2

Четвертные оценки

ДОБАВИТЬ ОЦЕНКУ ИЗМЕНИТЬ ОЦЕНКУ УДАЛИТЬ ОДНУ ИЗ ОЦЕНОК

Политология	3
Физкультура	4

ВНЕСТИ ИЗМЕНЕНИЯ УДАЛИТЬ УЧЕНИКА

Рисунок 20 –Изменение данных об ученике

11. Добавление оценки ученику

При нажатии на кнопку «Добавить оценку» в интерфейсе «Ученик» появляется модальное окно с формой добавления записи в модель «Оценка». Скриншот представлен на рисунке 21.

Добавление четвертной оценки x

Ученик Николаева

Оценка ☐ 5 ☐ 4 ☒ 3 ☐ 2

Предмет Математика

Добавить

Рисунок 21 –добавление оценки ученику

12. Изменение оценки

При нажатии на кнопку «Изменить оценку» в интерфейсе «Ученик» в таблице с оценками появляются поля ввода и опция выбора оценки. После нажатия на кнопку «Внести изменения» изменения вносятся в базу данных и страница обновляется. Скриншот представлен на рисунке 22.

Четвертные оценки

ДОБАВИТЬ ОЦЕНКУ
ВНЕСТИ ИЗМЕНЕНИЯ
УДАЛИТЬ ОДНУ ИЗ ОЦЕНОК

Политология	3	<input checked="" type="radio"/> Изменить <input type="text" value="5"/>
Физкультура	4	<input type="radio"/> Изменить <input type="text" value="5"/>

Рисунок 22 –изменение оценки

13. Удаление оценки

При нажатии на кнопку «Удалить одну из оценок» в интерфейсе «Ученик» рядом

с каждой оценкой появляется опция выбора, при нажатии на кнопку «Подтвердить» выбранная оценка удаляется и страница обновляется. Скриншот представлен на рисунке 23.



Рисунок 23 –удаление оценки

14. Список всех учеников школы

Данный интерфейс представляет собой список всех учеников школы. Каждая строка списка – это активная ссылка на досье выбранного ученика. При нажатии на кнопку «Добавить ученика» появляется модальное окно, скриншот которого представлен на рисунке 18. Скриншот самого интерфейса представлен на рисунке 24.

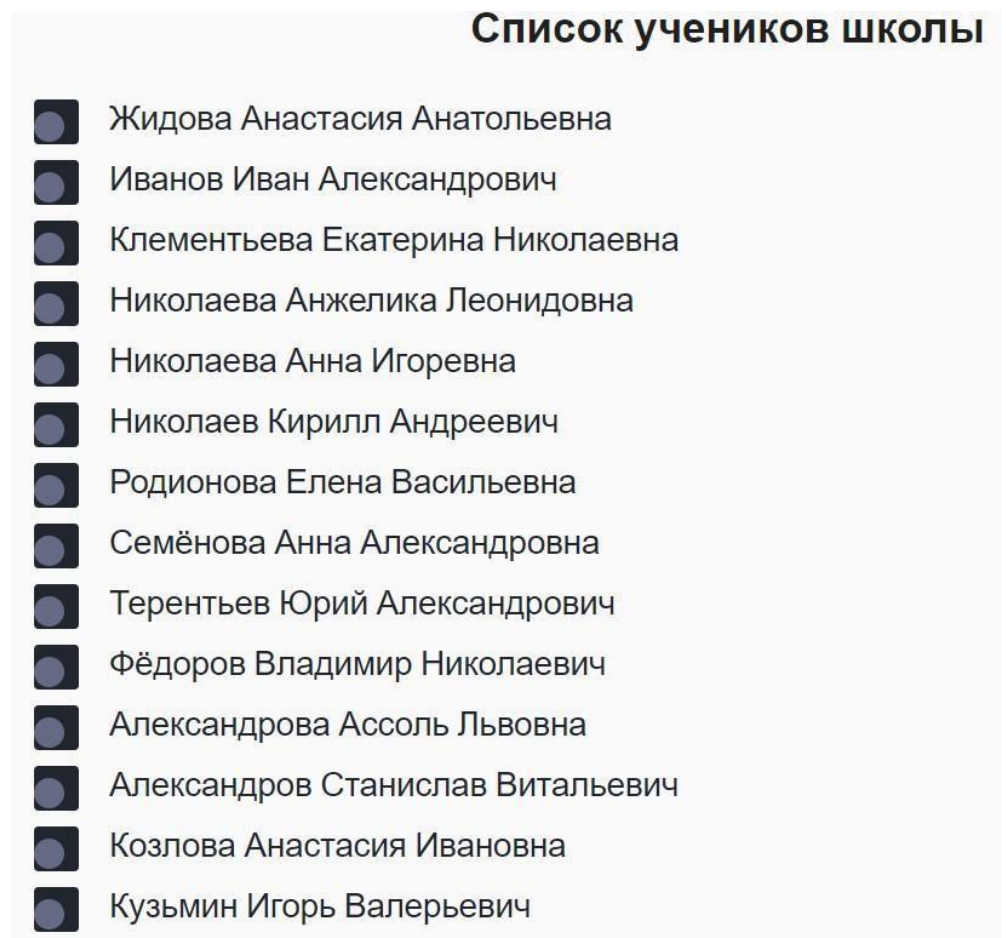


Рисунок 24 –Список всех учеников школы

15. Добавление записи в расписание

При нажатии на кнопку «Добавить запись в расписание» в интерфейсе «Расписание класса» появляется модальное окно с формой добавления записи в модель «Расписание». Скриншот представлен на рисунке 25.

Добавление записи в расписание

X

День недели ☐ Понедельник ☐ Вторник ☐ Среда
☐ Четверг ☐ Пятница ☐ Суббота

Урок

Класс

Предмет

Учитель

Кабинет

Добавить

Рисунок 25 –модальное окно «Добавление новой записи в расписание» 16.

Изменение записи в расписании

При нажатии на кнопку «Изменить одну из записей» в интерфейсе «Расписание класса» в таблице появляются поля для ввода и опция выбора записи для изменения. После нажатия на кнопку «Внести изменения» изменения вносятся в базу данных и страница обновляется.

Скриншот представлен на рисунке 26.

Пятница <input type="radio"/> Понедельник <input type="radio"/> Вторник <input type="radio"/> Среда <input type="radio"/> Четверг <input type="radio"/> Пятница <input type="radio"/> Суббота			
2-9:20-10:00	Кабинет №101	Литература	Петриковская
<input checked="" type="radio"/> Изменить			
4-11:00-11:40			

Рисунок 26 –изменение одной из записей в расписании

17. Удаление записи из расписания

При нажатии на кнопку «Удалить одну из записей» в интерфейсе «Расписание класса» рядом с каждой записью появляется опция выбора, при нажатии на кнопку «Подтвердить» выбранная запись удаляется из базы данных и страница обновляется. Скриншот представлен на рисунке 27.



Рисунок 27 –удаление одной из записей в расписании

18. Список дисциплин, преподаваемых в школе

Данный интерфейс представляет собой список предметов, преподаваемых в школе. Скриншот представлен на рисунке 28.

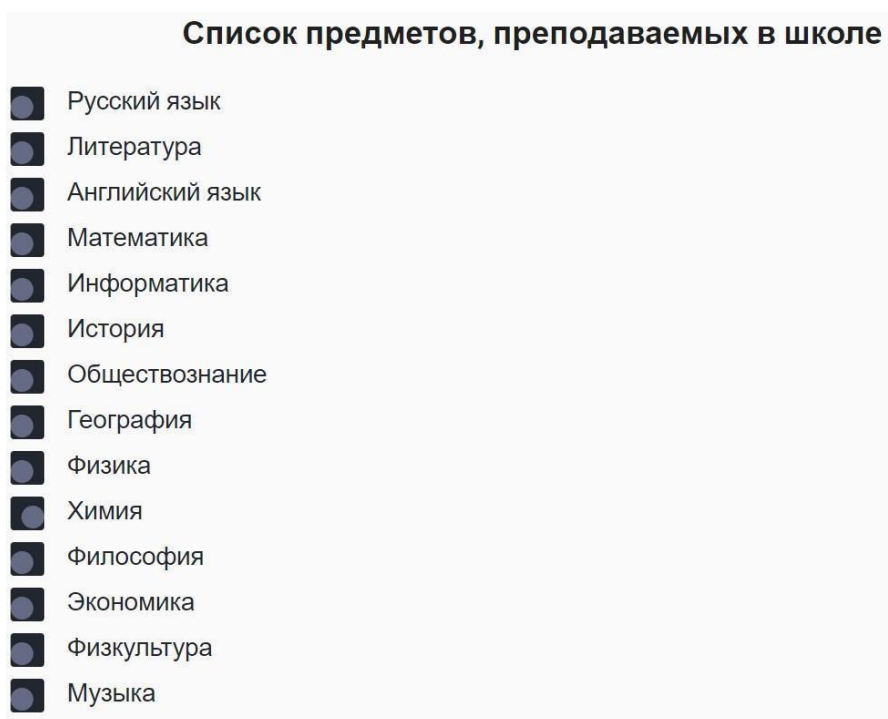


Рисунок 28 –Список предметов, преподаваемых в школе

19. Список кабинетов школы

Данный интерфейс представляет собой список кабинетов школы. Скриншот

представлен на рисунке 29.

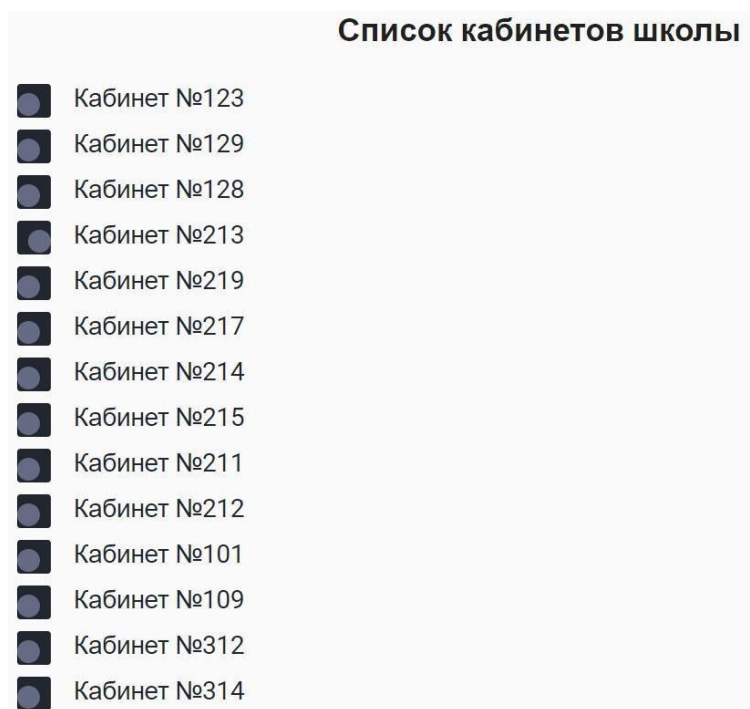


Рисунок 29 –Список кабинетов школы

20. Вход

Данный интерфейс представляет собой форму входа. В случае некорректного ввода логина и/или пароля появится соответствующее предупреждение. При нажатии на активную ссылку «зарегистрироваться» пользователь будет перенаправлен на страницу регистрации. Скриншот представлен на рисунке 30.

The image shows a login form with two input fields. The first field is labeled "Username" and contains the text "user2". Below it is a hint: "A character with a username of 6-12 length". The second field is labeled "Password" and contains masked characters ".....". To the right of the password field is a small icon of a key with a checkmark. Below the password field is a red error message: "Please enter password". At the bottom of the form is a grey button labeled "ВОЙТИ". Below the button is a line of text: "Нет логина и пароля? Тогда Вам необходимо [зарегистрироваться](#)".

Рисунок 30 –Вход

21. Регистрация

Интерфейс представляет собой форму регистрации нового пользователя. В случае успешного прохождения регистрации пользователь будет перенаправлен на страницу входа. Скриншот представлен на рисунке 31.

Пожалуйста, заполните следующие поля

Поля должны содержать не менее 8 символов.

user2

Password
..... visibility

Password again
..... visibility

ЗАРЕГИСТРИРОВАТЬСЯ

Рисунок 31 –Регистрация

22. Реализация фильтров

Интерфейс «Фильтры» представляет собой два фильтр, реализованных на бэкреализованных на бэкенде с помощью django-filters, в котором при выборе дня недели и урока и нажатии на кнопку отфильтровать отображается результат фильтрации. Скриншот представлен на рисунке 32.

Расписание классов

5					
Среда	4-11:00-11:40	Кабинет №45	Изобразительное искусство	Борисова	
6					
Среда	4-11:00-11:40	Кабинет №314	Психология	Лущекин	

ФИЛЬТР

Фильтр по дню

Среда

Фильтр по уроку

4-11:00-11:40

ОТФИЛЬТРОВАТЬ

Рисунок 32 –Фильтры к курсовой работе

ЗАКЛЮЧЕНИЕ

Данная работа показывает полученные в течение семестра навыки создания webприложений с помощью web-фреймворка Django языка программирования Python, webфреймворка Vue.js языка программирования JavaScript и библиотека Muse-UI.

В рамках выбранного варианта была создана программная система для завуча школы, обладающая всем необходимым заявленным функционалом:

- авторизация и регистрация в web-приложении.
- просмотр информации об учителях, учениках, классах, расписании и оценках.
- возможность добавления/изменения/удаления учителя, ученика, записи в расписании и оценки.
- получение результатов запросов.
- получение отчета по выбранному классу.

Заключительном этапом создания курсовой была работа с Docker – контейнеризация и оркестрация созданного web-приложения. Таким образом, конечный вариант web-приложения представляет собой образ, который можно развернуть на любой локальной машине, то есть теперь приложение находится уже на стадии разработки, а на стадии продакшена.

СПИСОК ЛИТЕРАТУРЫ

1. Модели и методы разработки крупномасштабных веб-приложений [Электронный ресурс]. – URL:
<https://www.dissercat.com/content/modeliimetodyrazrabotki-krupnomasshtabnykh-veb-prilozheniia> (дата обращения: 08.07.2020)
2. Сериализовать класс Django [Электронный ресурс]. – URL:
<https://fooobar.com/questions/7704637/serialize-class-django> (дата обращения: 08.07.2020)
3. ViewSets [Электронный ресурс]. – URL: <https://www.django-rest-framework.org/apiguide/viewsets/> (дата обращения: 08.07 2020)
4. Class-based Views [Электронный ресурс]. – URL:
<https://www.django-restframework.org/api-guide/views/> (дата посещения 08.07.2020)