

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

Факультет ИКТ

Образовательная программа Интеллектуальные системы в гуманитарной сфере

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

О Т Ч Е Т

по курсовой работе

Тема задания: РЕАЛИЗАЦИЯ WEB-СЕРВИСОВ СРЕДСТВАМИ Django REST framework.

Обучающийся: Ревенко Анна Павловна, К3343

Руководитель: Говоров А. И.

Подписи членов комиссии:

(подпись)

(подпись)

(подпись)

Дата ____

Санкт-Петербург
2020

1. СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ПРОЕКТИРОВАНИЕ	5
Проектирование программной системы	5
Проектирование базы данных	6
РЕАЛИЗАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ	7
Общая информация	7
Реализация интерфейсов "Входа" и "Регистрации" пользователей	8
Реализация интерфейса "Подача заявления"	10
Реализация интерфейса "Все поданные заявления "	14
Реализация интерфейса "Поиск заявления"	15
ЗАКЛЮЧЕНИЕ	18
СПИСОК ЛИТЕРАТУРЫ	19

2. ВВЕДЕНИЕ

Каждый год на протяжении очень долгих лет огромное количество абитуриентов проступают в различные учебные заведения. Приемная комиссия вынуждена обрабатывать тысячи заявлений, но если раньше все это делалось вручную, то с развитием технологий web-сервисы облегчили эту задачу. Данный проект описывает программную систему, предназначенную для работников приемной комиссии колледжа. Она обеспечивает хранение, просмотр и изменение сведений об абитуриентах. Результатом работы приемной комиссии является список абитуриентов, зачисленных в колледж.

Секретарь приемной комиссии регистрирует абитуриентов. Для каждого абитуриента в базу данных заносятся следующие сведения: фамилия, имя, отчество, паспортные данные, какое учебное заведение, где и когда окончил, наличие золотой или серебряной медали, название специальности, на которые поступает абитуриент. При подаче заявления абитуриент указывает форму обучения (очная, очно-заочная (вечерняя), заочная), поступление на бюджет или контракт. Абитуриент может поступать вне конкурса (инвалиды, сироты). Также существуют абитуриенты-целевики, которые поступают по договорам с направляющими организациями, и обучаются на коммерческой основе. Абитуриенты, поступающие на базе 9 классов, участвуют в конкурсе аттестатов. Для них указывается информация по 4-м профильным дисциплинам и средний балл по всем остальным дисциплинам аттестата.

На основе этих данных строится рейтинг абитуриентов. Абитуриенты, поступающие на базе 11 классов, предоставляют сертификаты ЕГЭ по 2 дисциплинам, на основе чего строится рейтинг абитуриентов. Конкурс для абитуриентов на базе 9 и 11 классов отдельный, т.к. они поступают на разные курсы. Абитуриент может не только подать, но и забрать документы, а также перевести их на другую специальность. Известно количество мест на каждый факультет. Приемная комиссия по результатам экзаменов должна сформировать списки абитуриентов, зачисленных в колледж.

Секретарю приемной комиссии могут потребоваться следующие сведения:

- Список абитуриентов, подавших заявление на заданную специальность;
- Количество абитуриентов, подавших заявления на каждую специальность по каждой форме обучения на бюджет (или контракт);

- Количество абитуриентов на базе 9 и 11 классов, поступающих на бюджет (или контракт);
- Общее количество поданных заявлений ежедневно;
- Конкурс на каждую специальность по каждой форме обучения на бюджет. Необходимо предусмотреть возможность получения документа, представляющего собой сгруппированный по заданной специальности список абитуриентов по заданной форме обучения, зачисленных в колледж, с указанием набранных ими баллов по аттестату. Отчет должен содержать проходной балл по специальности в целом, а также количество абитуриентов, поступающих на специальность.

Список интерфейсов, реализованных для программной системы приемной комиссии колледжа:

1. Список абитуриентов, подавших заявление на заданную специальность.
2. Количество абитуриентов, подавших заявления на каждую специальность по каждой форме обучения на бюджет (или контракт).
3. Количество абитуриентов на базе 9 и 11 классов, поступающих на бюджет (или контракт).
4. Общее количество поданных заявлений ежедневно.
5. Конкурс на каждую специальность по каждой форме обучения на бюджет (или контракт).
6. Возможность подать заявление.
7. Возможность просмотра своего места в конкурсе по каждому поданному заявлению.

1. ПРОЕКТИРОВАНИЕ

1.1.Проектирование программной системы

Для реализации программной системы для приемной комиссии колледжа за основу был взят Django Rest Framework. С помощью него реализовано сервисное отображение данных, выборка необходимой информации, добавление новой информации и исполнение запросов в соответствии с поставленными задачами функционала.

Бэкенд программной системы был реализован средствами Django Rest Framework, для реализации фронтенда был выбран Vue.js, а также использована база данных - PostgreSQL.

Для осуществления фронтенда системы использовались Vue.js с плагином Vuetify. Vuetify был выбран по причине быстрой скорости сборки клиентской части, а также удобства и простоты дизайна. Представления системы отображены в папке Components и полностью удовлетворяют запрашиваемому функционалу.

PostgreSQL была использована в качестве базы данных, а с помощью редактора PyCharm производилась настройка базы данных, последующая работа и сборка системы. Для удобства запуска, а также быстроты дальнейшей сборки на желаемом сервисе были добавлены Docker контейнеры для каждой части архитектуры приложения.

1.2.Проектирование базы данных

В качестве сущностей для разрабатываемой программной системы были выбраны Абитуриент, Документы, Баллы_аттестат, Экз_баллы, Секретарь, Факультет, Кафедра, Специальность, Реализация_спец-ти. Связывающая сущность – Заявление.

Реализация вышеописанной базы данных представлена на Рис.1.

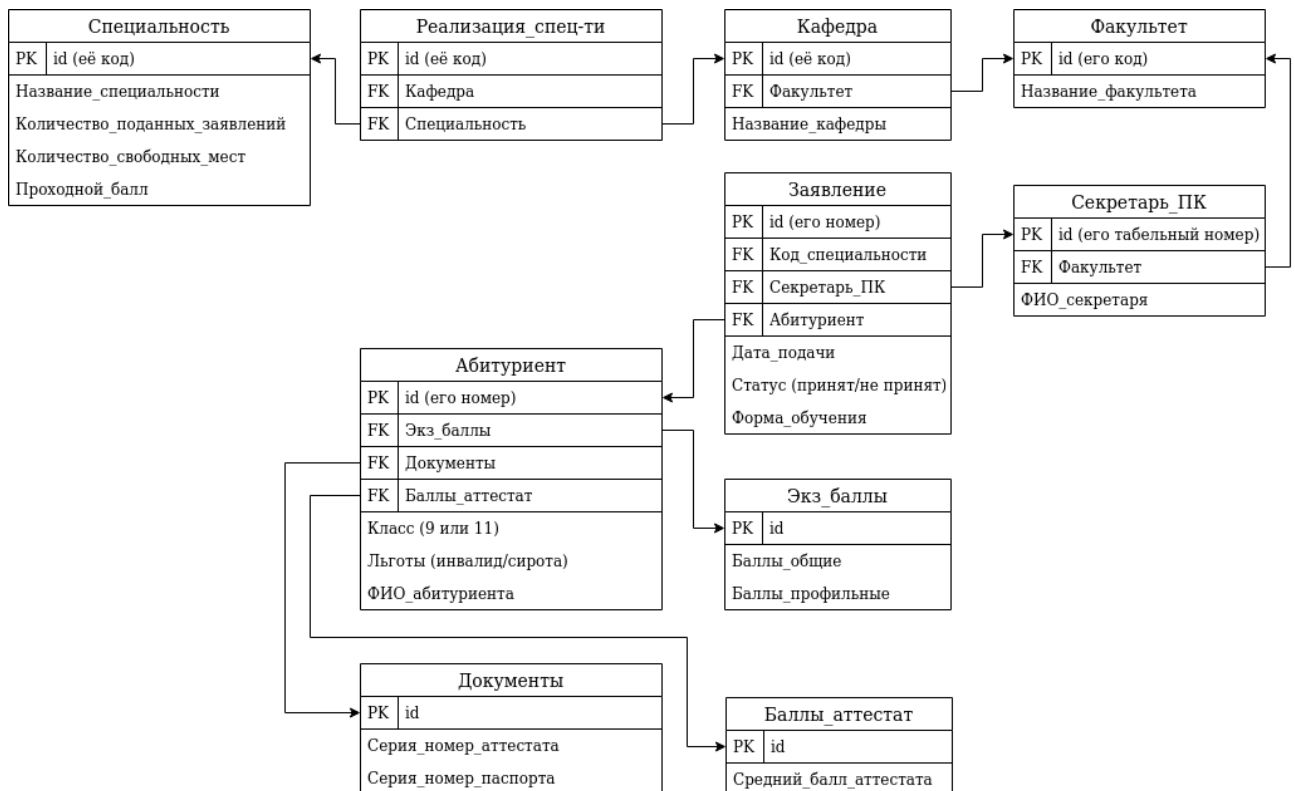


Рис.1 - Диаграмма базы данных программной системы

2. РЕАЛИЗАЦИЯ ПРОГРАММНОЙ СИСТЕМЫ

2.1.Общая информация

Весь проект разделён на следующие интерфейсы:

- Регистрация и вход для абитуриентов;
- Подача заявления на поступление в колледж (необходима регистрация пользователя);
- Личный кабинет абитуриента (содержит всю информацию об абитуриенте, необходимую для поступления – личные данные, документы, баллы за экзамены, средний балл аттестата, а также отображает все поданные им заявления);
- Страница конкурсного рейтинга всех поданных заявлений (с возможностью фильтровать их в зависимости от выбора специальности, класса абитуриента, формы обучения, даты подачи);

Вся навигация в проекте и дизайн реализованы с помощью плагина Vue.js Vuetify. Для реализации программной системы были использованы следующие модули: django, django rest framework, psycopg2, django cors headers.

Добавленные модели представлены на Рис.2.

COLLEGE		
Applications	+ Add	Change
Certificates	+ Add	Change
Departments	+ Add	Change
Documents	+ Add	Change
Enrollees	+ Add	Change
Exams	+ Add	Change
Facultys	+ Add	Change
Secretarys	+ Add	Change
Specialization realises	+ Add	Change
Specializations	+ Add	Change

Рис.2 - Модели

2.2.Реализация интерфейсов «Входа» и «Регистрации» пользователей

Логика событий регистрации: вначале пользователь придумывает логин и пароль, затем заполняет информацию о себе (сущность “Абитуриент”), и в конце авторизуется, как пользователь и может приступить к нужным действиям. Регистрация абитуриента и вход в личный кабинет представлены на Рис.3, 4 и 5.

The screenshot shows the 'Вход абитуриента' (Abiturient Login) page. At the top is a teal navigation bar with links: 'Приёмная комиссия колледжа', 'Конкурс', 'Подать заявление', 'Личный кабинет', 'Регистрация', 'Вход', and 'Выход'. The main heading is 'Вход абитуриента'. Below it, there is a section 'Имя пользователя' with a text input field containing 'Имя пользователя'. Below that is a section 'Пароль' with a text input field containing 'Пароль'. A blue button labeled 'Вход' is positioned below the password field. At the bottom, there is a link: 'Если вы ещё не зарегистрированы, то сделайте это тут.'.

Рис.3 - Реализация Входа в личный кабинет на сайте приемной комиссии

The screenshot shows the 'Регистрация абитуриента' (Abiturient Registration) page. At the top is a teal navigation bar with links: 'Приёмная комиссия колледжа', 'Конкурс', 'Подать заявление', 'Личный кабинет', 'Регистрация', 'Вход', and 'Выход'. The main heading is 'Регистрация абитуриента'. Below it, there is a link: 'Если вы уже зарегистрированы, то можете войти тут.'.

The form is divided into several sections:

- Данные для входа**: Includes fields for 'Имя пользователя' (containing 'Имя пользователя') and 'Пароль' (containing 'Пароль').
- Информация о Вас**: Includes fields for 'Фамилия', 'Имя', 'Отчество', 'Отчество', 'Класс', 'Выберите класс' (dropdown menu), 'Льготы', and 'Выберите льготы' (dropdown menu).
- Ваши документы**: Includes fields for 'Аттестат', 'Паспорт', and 'Паспорт'.
- Балл по экзаменам**: Includes fields for 'Профильные' (containing 'Профильные') and 'Общие' (containing 'Общие').
- Средний балл аттестата**: Includes a field for 'Средний балл аттестата' (containing 'Средний балл аттестата').

A blue button labeled 'Регистрация' is located at the bottom of the form.

Рис.4, 5 - Реализация Регистрации абитуриентов и добавления личной информации

Данный интерфейс отображает общую информацию об абитуриенте, зарегистрировавшемся на сайте приемной комиссии. Информация об абитуриенте включает в себя его логин, ФИО, документы, баллы за экзамены и средний балл аттестата и другие данные, необходимые для поступления.

Реализация интерфейса Enrollee представлена на Рис.6.

Django REST framework

Create Enrollee

OPTIONS

GET /api/add/enrollee/

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

*Detail: "Method 'GET' not allowed."

Raw data HTML form

School class 9

Privelege No

First name

Last name

Patronymic

Documents Certificate 1234 Passport 1234

Exams Prole points 200 Common points 100

Certificate avg 4

User Anna

POST

Рис.6 – Create Enrollee

Реализация интерфейса Doc представлена на Рис.7.

Django REST framework

Create Doc

OPTIONS

GET /api/add/doc/

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

*Detail: "Method 'GET' not allowed."

Raw data HTML form

Certificate num

Passport num

POST

Рис.7 – Create Doc

Реализация интерфейса Exam представлена на Рис.8.

Django REST framework

Create Exam

OPTIONS

GET /api/add-exam/

HTTP 405 Method Not Allowed

Allow: POST, OPTIONS

Content-Type: application/json

Vary: Accept

{
 "detail": "Method 'GET' not allowed."
}

Raw data HTML form

Profile points

Common points

POST

Рис.8 – Create Exam

Реализация интерфейса Cert представлена на Рис.9.

Django REST framework

Create Cert

OPTIONS

GET /api/add-cert/

HTTP 405 Method Not Allowed

Allow: POST, OPTIONS

Content-Type: application/json

Vary: Accept

{
 "detail": "Method 'GET' not allowed."
}

Raw data HTML form

Avg mark

POST

Рис.9 – Create Cert

2.3.Реализация интерфейса «Подача заявления»

Данный интерфейс предоставляет возможность абитуриенту подать заявление на поступление в колледж - создать новую запись в базе данных в таблице “Заявление”. Подать заявление на сайте может только авторизованный пользователь. Пример реализации интерфейса представлен на Рис.10.

Рис.10 - Реализация интерфейса «Поддача заявления»

При отсутствии поданных заявлений в личном кабинете высвечивается информация об этом (Рис.11).

Рис.11 – Отсутствие поданных заявлений абитуриентом

При успешной подаче заявления появляется всплывающее окно, уведомляющее об этом (Рис.12).

The screenshot shows a web interface for submitting an application. At the top, a teal navigation bar contains links: "Приёмная комиссия колледжа", "Конкурс", "Подать заявление", "Личный кабинет", "Регистрация", "Вход", and "Выход". Below this, a green banner displays the message: "Ваше заявление успешно подано, вы сможете отследить его в личном кабинете". The main section is titled "Подача заявления" and contains two dropdown menus: "Специальность" (with "Информ" selected) and "Форма" (with "Бюджет" selected). A blue "Подать" button is positioned below the second dropdown.

Рис.12 – Всплывающее окно при успешной подаче заявления

Успешно поданное заявление отображается в личном кабинете абитуриента (Рис.13).

The screenshot displays the "Кабинет абитуриента" (Applicant's Cabinet) page. The teal navigation bar at the top includes the same links as in Figure 12. The page title is "Кабинет абитуриента" with a subtitle "Здесь вы сможете увидеть все свои заявления". Below the title, it states "Найдено 1 заявлений". A card displays the following information: "Курский Владимир Георгиевич" (with a blue "Изменить" link), "Дата подачи: 2020-07-01", "Форма обучения: Contract", "Состояние: In process", and "Ответственный: Василий Фёдоров Васькович" (with a blue "Подробнее" link).

Рис.13 – Заявление абитуриента в его личном кабинете

Реализация интерфейса Application представлена на Рис.14.

Django REST framework

Create Application

OPTIONS

GET /api/add-application/

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

Raw data HTML form

Date: dd.mm.yyyy

State: Accepted

Education form: Contract

Specialization: ISGS

Secretary: _____

Enrollee: Ivanov Ivan Ivanovich

POST

Рис.14 – Create Application

Конкретное поданное заявление абитуриента можно увидеть с помощью Get Application (на Рис.15 представлено одно из поданных заявлений).

Django REST framework

Get All Application Get Application

OPTIONS GET

GET /api/applications/1

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

```
{
  "id": 1,
  "secretary": {
    "id": 1,
    "faculty": {
      "id": 1,
      "name": "ЮФМ"
    },
    "first_name": "Евгений",
    "last_name": "Иванов",
    "patronymic": "Ильич"
  },
  "specialization": {
    "id": 1,
    "name": "ИТОС",
    "flow_positions": 30,
    "applications_num": 200,
    "national_quota": 100
  },
  "enrollee": {
    "id": 1,
    "enrollee_name": "ИМ",
    "previous_name": "ИМ",
    "birthdate": {
      "id": 4,
      "certificate_num": 1234,
      "passport_num": 1234
    }
  },
  "dates": {
    "id": 1,
    "previous_position": 200,
    "current_position": 100
  },
  "certificate_exam": {
    "id": 2,
    "avg_rank": {
    }
  },
  "first_name": "Иван",
  "last_name": "Иванов",
  "patronymic": "Иванович"
},
{
  "id": 1,
  "enrollee_name": "Иванов",
  "birthdate": {
    "id": 1,
    "certificate_num": "123456789",
    "passport_num": "123456789"
  }
}
```

Рис.15 – Get Application

2.4.Реализация интерфейса «Все поданные заявления»

Данный интерфейс предоставляет информацию обо всех поданных заявлениях. На странице они расположены в рейтинговом порядке, абитуриенты с самыми высокими баллами высвечиваются в верху таблицы. Пример реализации интерфейса представлен на Рис.16.

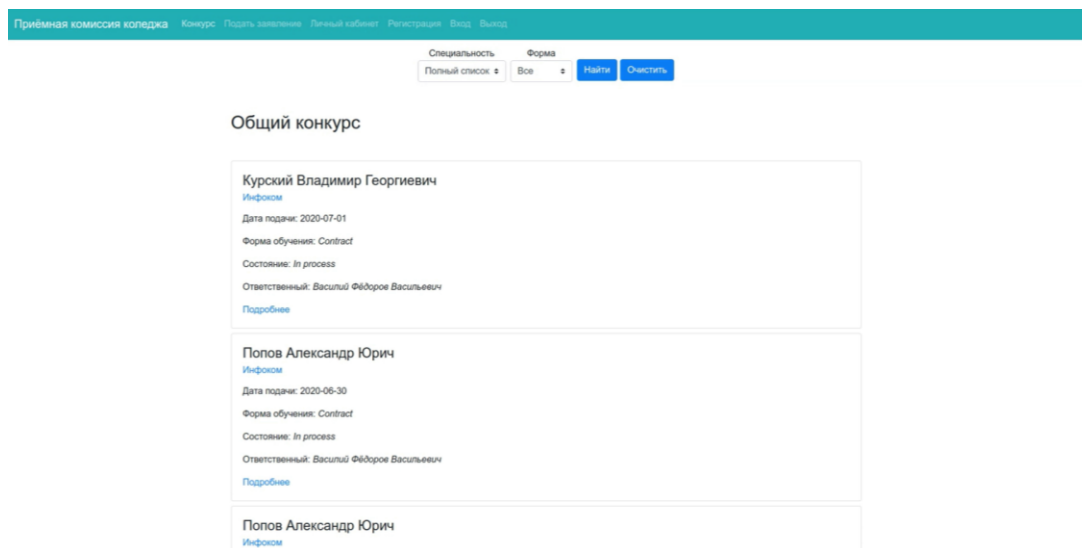


Рис.16 – Реализация интерфейса «Все поданные заявления»

Все поданные заявления можно увидеть с помощью Get All Application (на Рис.17 представлены несколько уже поданных заявлений).

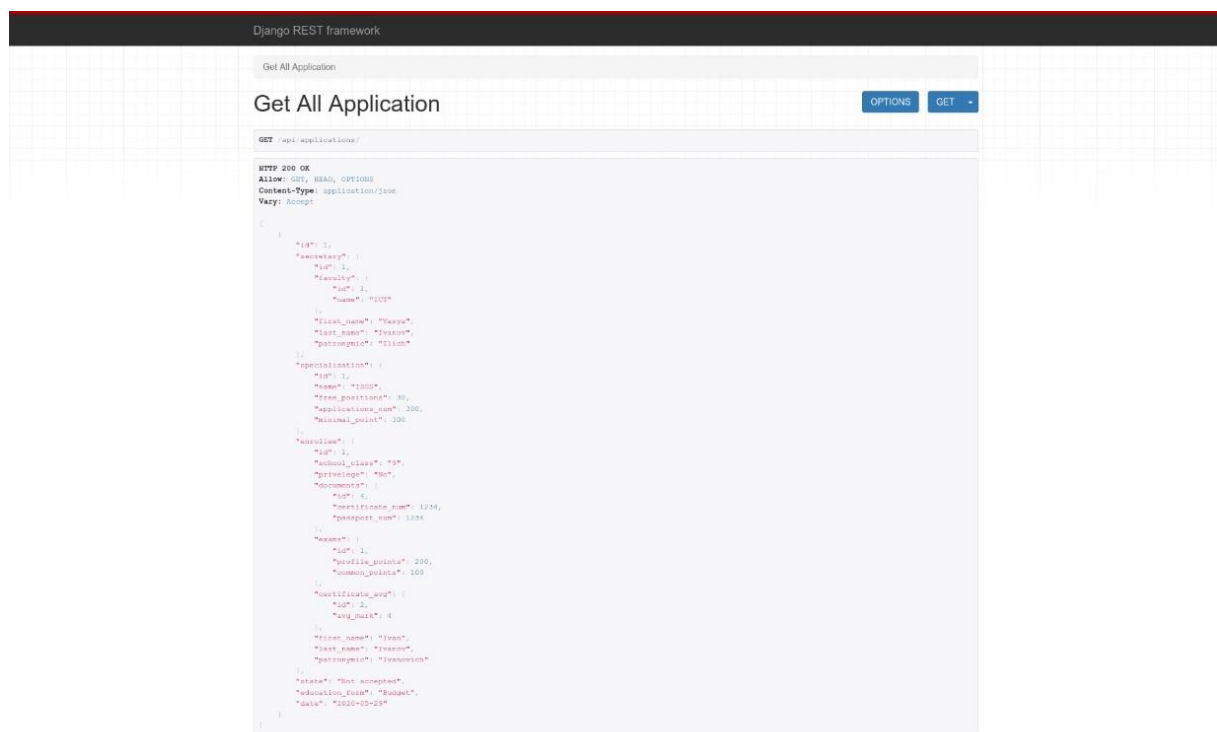


Рис.17 – Get All Application

2.5.Реализация интерфейса «Поиск заявления»

Данный интерфейс позволяет сортировать поданные заявления, благодаря имеющимся фильтрам. Можно произвести поиск заявления по специальности, классу абитуриента, дате подачи заявления, форме обучения и статусу самого заявления.

Пример реализации интерфейса представлен на Рис.18 (выбрана специальность Инфоком, класс абитуриента 9, дата подачи 30 июня 2020г., контрактная форма обучения).

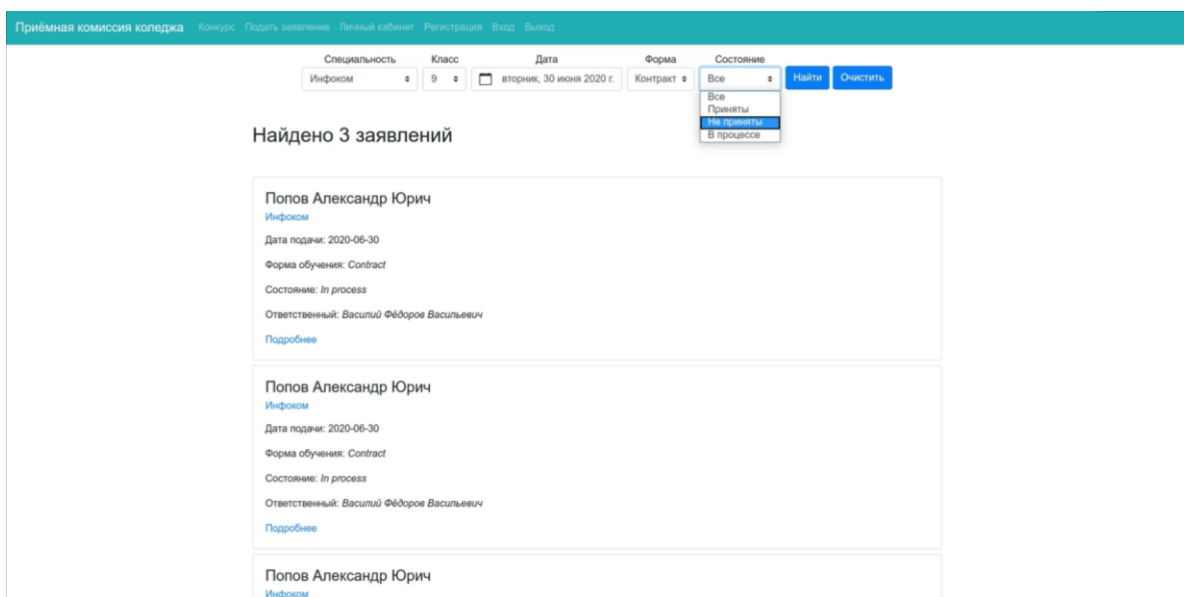


Рис.18 – Реализация интерфейса «Поиск заявления»

Получение информации по всем специализациям можно получить с помощью Get All Specialization (Рис.19).



Рис.19 – Get All Specialization

Получение информации по конкретной специализации можно получить с помощью Get Specialization (На Рис.20 представлена информация по одной из специализаций).

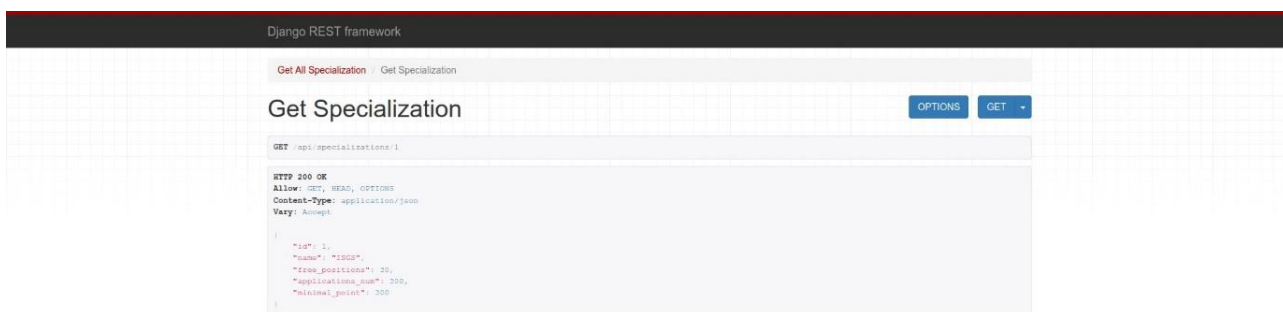


Рис.20 – Get Specialization

3. ЗАКЛЮЧЕНИЕ

Цель курсовой работы заключалась в реализации программной системы для приемной комиссии колледжа, которая могла бы обеспечивать хранение, просмотр и изменение сведений об абитуриентах.

Для решения данной задачи был выбран стек технологий: Django Rest Framework, Vue.js, PostgreSQL, что позволило вполне быстро и удобно реализовать задуманный проект. В результате выполнения курсовой работы удалось выполнить все ранее поставленные задачи, которые соответствуют запросам технического задания. Цель курсовой работы достигнута в полном размере.

В качестве дальнейшего развития проекта можно усовершенствовать дизайн web-сайта, например, добавить изображения отсканированных копий документов абитуриентов колледжа. Также может быть добавлена функция связи с приемной комиссией через чат в личном кабинете абитуриента.

4. СПИСОК ЛИТЕРАТУРЫ

1. *Django Rest Framework*. Документация Django Rest Framework [Электронный ресурс]. URL: <https://www.django-rest-framework.org> (дата обращения: 02.07.2020).
2. *Vue.js*. Документация Vue.js [Электронный ресурс]. URL: <https://vuejs.org> (дата обращения: 02.07.2020).
3. *Vuetify*. Документация Vuetify [Электронный ресурс]. URL: <https://vuetifyjs.com/ru/> (дата обращения: 02.07.2020).
4. *WebDevBlog*. Создание Django API используя Django Rest Framework [Электронный ресурс]. URL: <https://webdevblog.ru/sozдание-django-api-ispolzuya-django-rest-framework-apiview/> (дата обращения: 02.07.2020).
5. *Evantotuts+*. JWT Аутентификация в Django [Электронный ресурс] URL: <https://code.tutsplus.com/ru/tutorials/how-to-authenticate-with-jwt-in-django--cms-30460> (дата обращения: 02.07.2020).
6. *Asyncee*. Продвинутое запросы в Django: сортировка по дате [Электронный ресурс]. URL: <https://asyncee.github.io/posts/advanced-django-querying-sorting-events-by-date/> (дата обращения: 02.07.2020).
7. *Stack Overflow*. Set initial vuetify vue-select value [Электронный ресурс]. URL: <https://stackoverflow.com/questions/51392719/set-initial-vuetify-v-select-value> (дата обращения: 02.07.2020).