**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**ФАКУЛЬТЕТ ИКТ**

Отчет

по лабораторной работе №2-3

по курсу «Основы Web-программирования»

Тема:

РЕАЛИЗАЦИЯ WEB-СЕРВИСОВ СРЕДСТВАМИ Django REST framework, Vue.js, Muse-UI

Вариант 1

Выполнила:

Андреева Е.А.

Студентка группы К3343

Проверил:

Говоров А. И.

Санкт-Петербург

2020

**Цель работы:** овладеть практическими навыками и умениями реализации web-сервисов средствами Django REST framework, Vue.js, Muse-UI.

**Оборудование**: компьютерный класс.

**Программное обеспечение**: Python 3.6, Django REST framework, Vue.js, Muse-UI (или аналогичная библиотека), PostgreSQL \*.

**Практическое задание:**

Реализовать сайт используя вышеуказанные технологии, в соответствии с практическим заданием.

Задание 1.

Создать программную систему, предназначенную для администратора гостиницы. Такая система должна обеспечивать хранение сведений об имеющихся в гостинице номерах, о проживающих в гостинице клиентах и о служащих, убирающихся в номерах. Количество номеров в гостинице известно, и имеются номера трех типов: одноместный, двухместный и трехместный, отличающиеся стоимостью проживания в сутки. В каждом номере есть телефон. О каждом проживающем должна храниться следующая информация: номер паспорта, фамилия, имя, отчество, город, из которого он прибыл, дата поселения в гостинице, выделенный гостиничный номер. О служащих гостиницы должна храниться информация следующего содержания: фамилия, имя, отчество, где (этаж) и когда (день недели) он убирает. Служащий гостиницы убирает все номера на одном этаже в определенные дни недели, при этом в разные дни он может убирать разные этажи.

Работа с системой предполагает получение следующей информации:

• о клиентах, проживавших в заданном номере, в заданный период времени;

• о количестве клиентов, прибывших из заданного города,

• о том, кто из служащих убирал номер указанного клиента в заданный день недели,

• сколько в гостинице свободных номеров;

• список клиентов с указанием места жительства, которые проживали в те же дни, что и заданный клиент, в определенный период времени.

Администратор должен иметь возможность выполнить следующие операции:

• принять на работу или уволить служащего гостиницы;

• изменить расписание работы служащего;

• поселить или выселить клиента.

Необходимо предусмотреть также возможность автоматической выдачи отчета о работе гостиницы за указанный квартал текущего года. Такой отчет должен содержать следующие сведения:

• число клиентов за указанный период в каждом номере с указанием ФИО клиента, города, откуда он прибыл, количества дней проживания,

• общая сумма дохода за каждый номер;

• суммарный доход по всей гостинице.

**Ход работы:**

1. Разработка модели базы данных

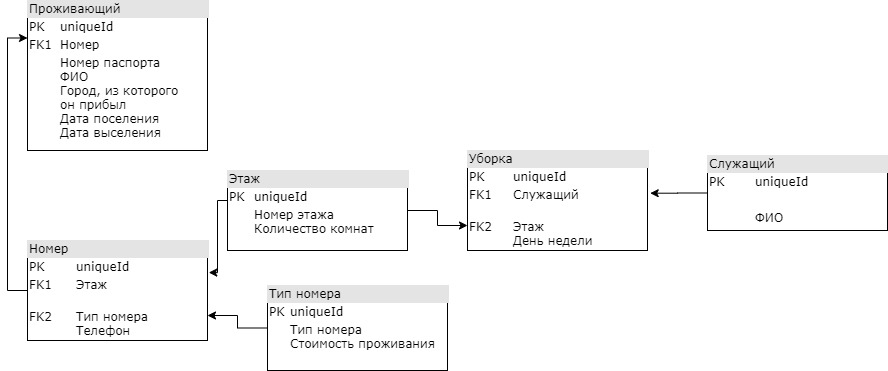
В соответствии с вариантом была разработана модель базы данных, представленная на рисунке 1.

Рисунок 1 – модель базы данных

1. Создание моделей

Согласно разработанной базе данных были созданы следующие модели:

* Floor – Этаж. Данная модель содержит информацию о номере этажа и количестве комнат на этом этаже
* RoomType – Тип комнаты. Данная модель содержит информацию о стоимости комнаты определенного типа (одноместная, двухместная, трехместная)
* Room – Комната. Данная модель описывает номер комнаты, на каком этаже она находится и номер телефона.
* Resident – Проживающий. Данная модель содержит информацию о проживающем: имя, фамилия, отчество, номер паспорта, откуда он прибыл, в какую комнату поселен и на какой срок.
* Servant – Служащий. Данная модель хранит данные о служащем: фамилию, имя, отчество.
* Cleaning – Уборка. Данная модель содержит информацию о том, какой служащий убирает комнаты на каком этаже и в какой день недели.

1. Создание отображений

Для создания отображений использовался класс ViewSet, который обладает встроенными атрибутами для последующего создания функций CRUD для модели. Согласно варианту, были созданы следующие отображения:

* class FloorViewSet – отображение для модели Этаж
* class RoomTypeViewSet – отображение для модели Комната
* class ResidentViewSet – отображение для модели Проживающий
* class ServantViewSet – отображение для модели Служащий
* class CleaningViewSet – отображение для модели Уборка

1. Полученные интерфейсы в Django REST
   1. Этажи

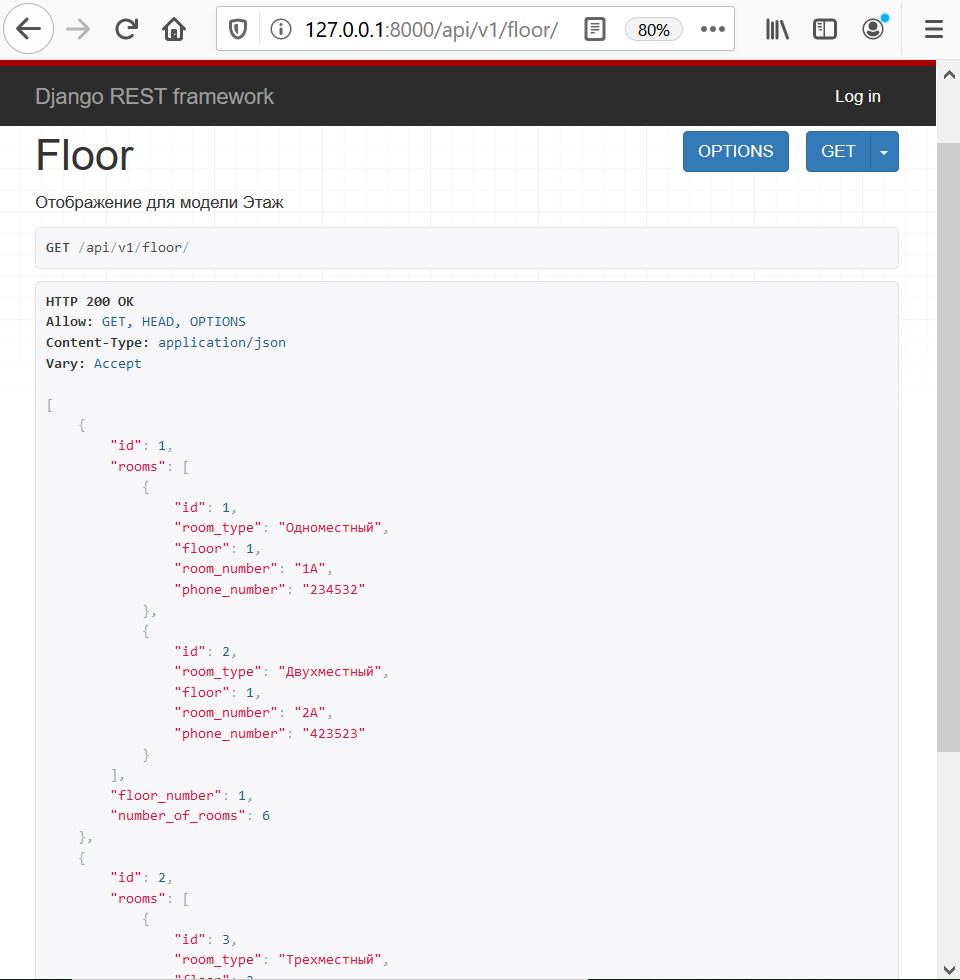
Выводятся данные обо всех этажах, а также о конатах, которые на них находятся.

Рисунок 2 – Этажи в Django REST

* 1. Комната

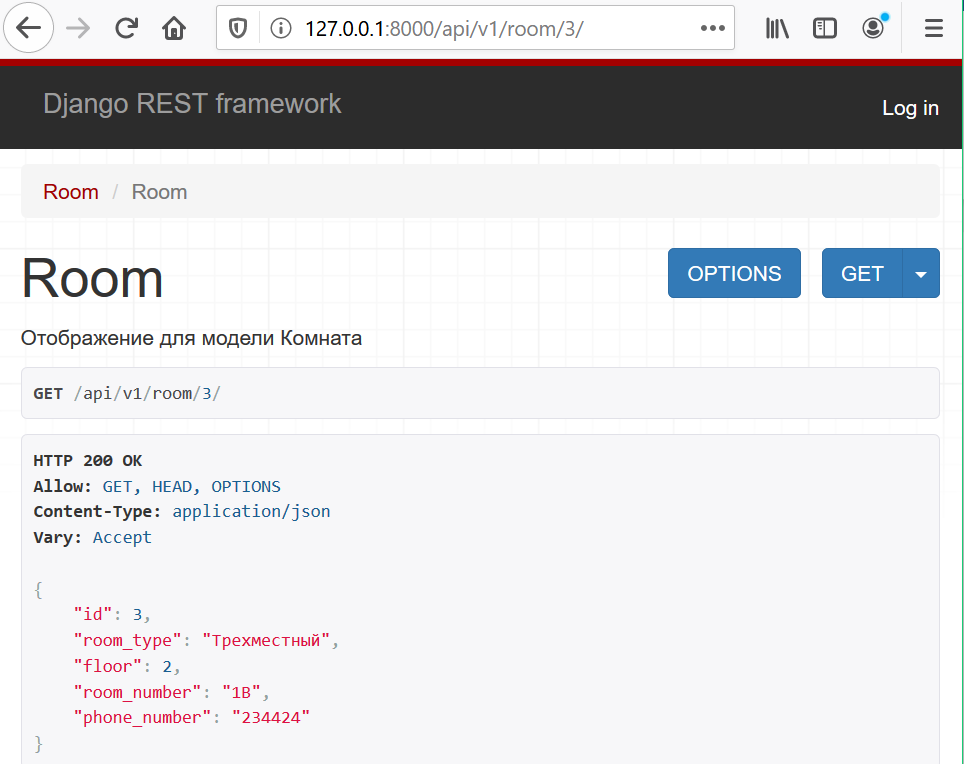
Вывод заданной комнаты с подробной информацией о ней.

Рисунок 3 – заданная Комната в Django REST

* 1. Типы комнат

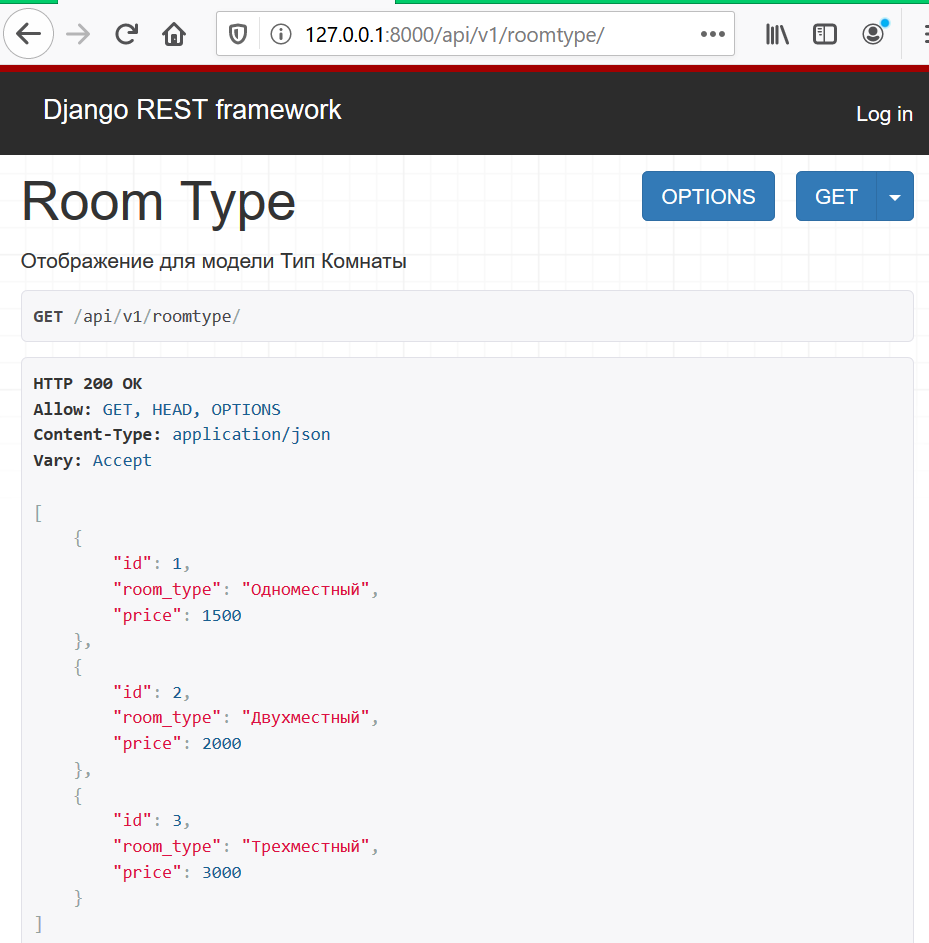
Выводится информация о типе комнат в гостинице.

Рисунок 4 – Типы комнат в Django REST

* 1. Служащие

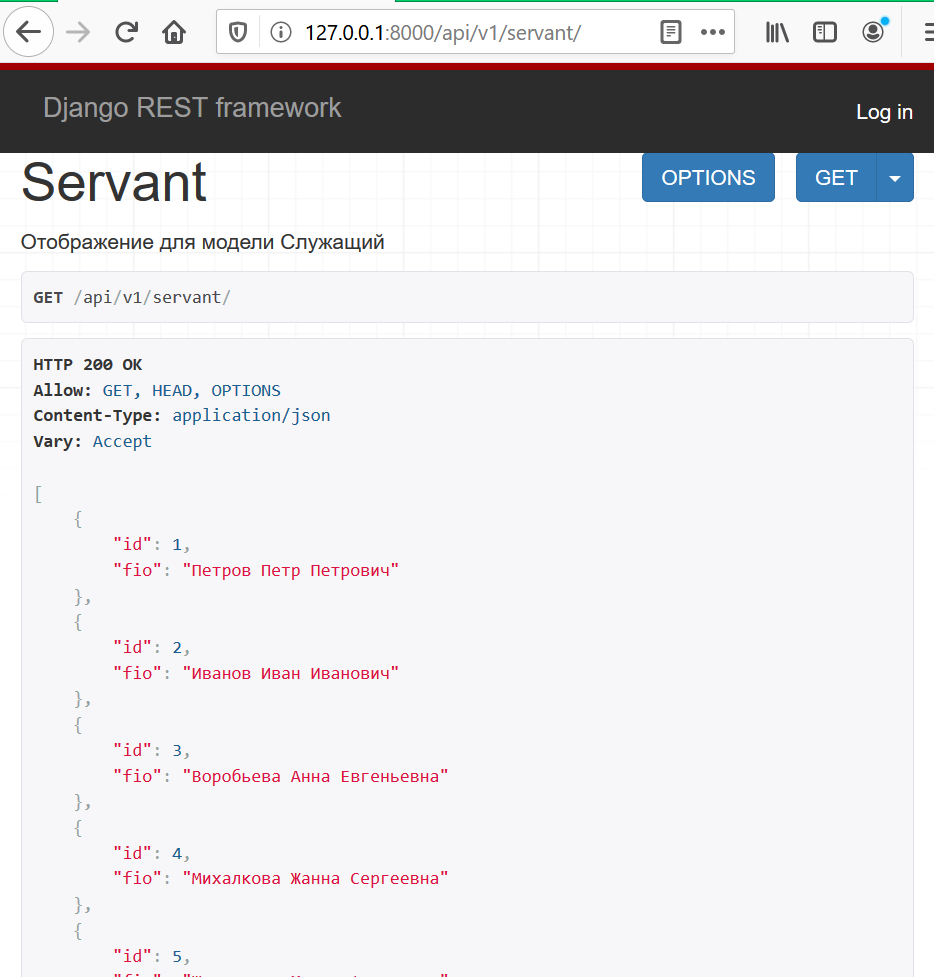
Вывод информации обо всех служащих.

Рисунок 5 – Служащие в Django REST

* 1. Уборка

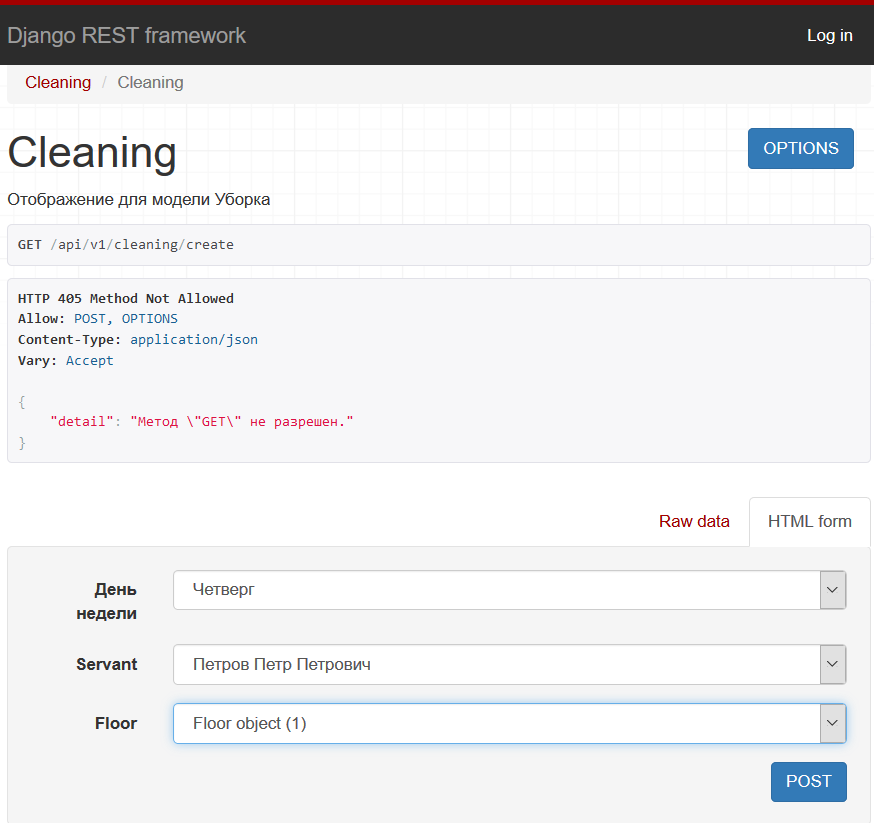
Добавление новой записи в модель Уборка.

Рисунок 6 – добавление новой записи в Django REST

* 1. Проживающий

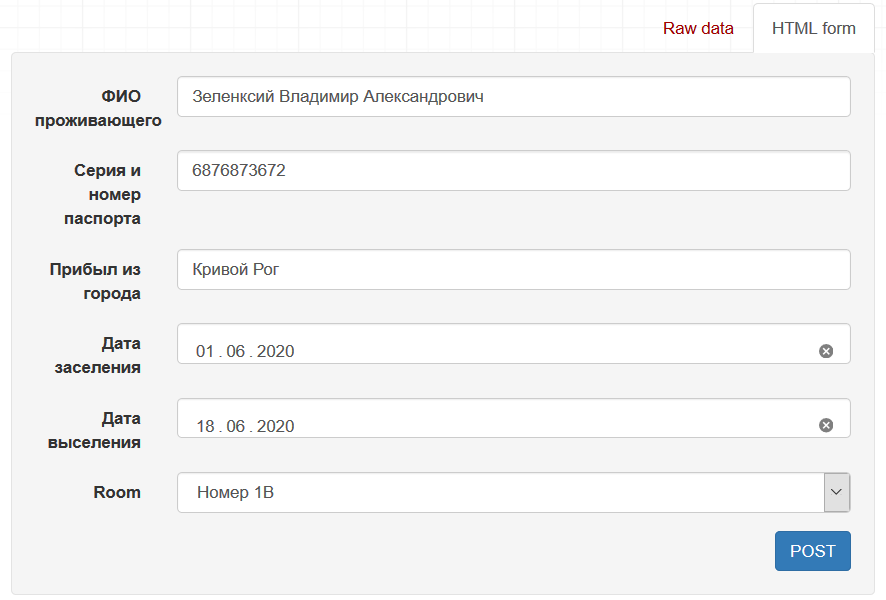
Добавление нового проживающего.

Рисунок 7 – добавление Проживающего в Django REST

1. Полученные интерфейсы с помощью Vue.js
   1. Стартовая страница

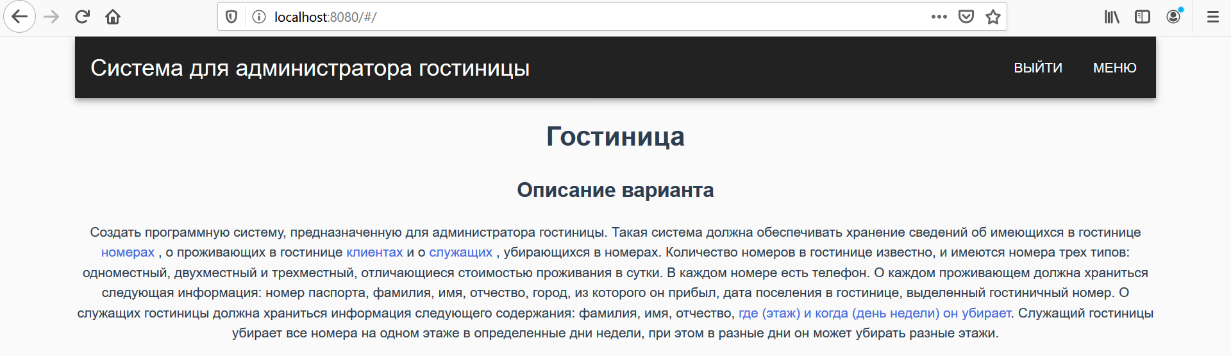
Стартовая страница web-сервиса с верхним меню навигации и описанием варианта лабораторной работы.

Рисунок 8 – Стартовая страница web-сервиса

* 1. Вход

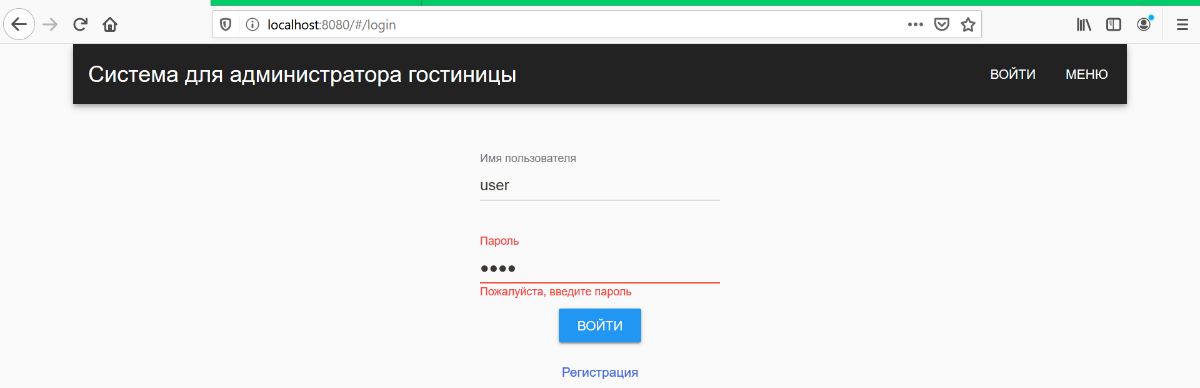
Страница авторизации пользователя имеет форму входа, а также ссылку на страницу регистрации.

Рисунок 9 – Страница входа

* 1. Регистрация новых пользователей

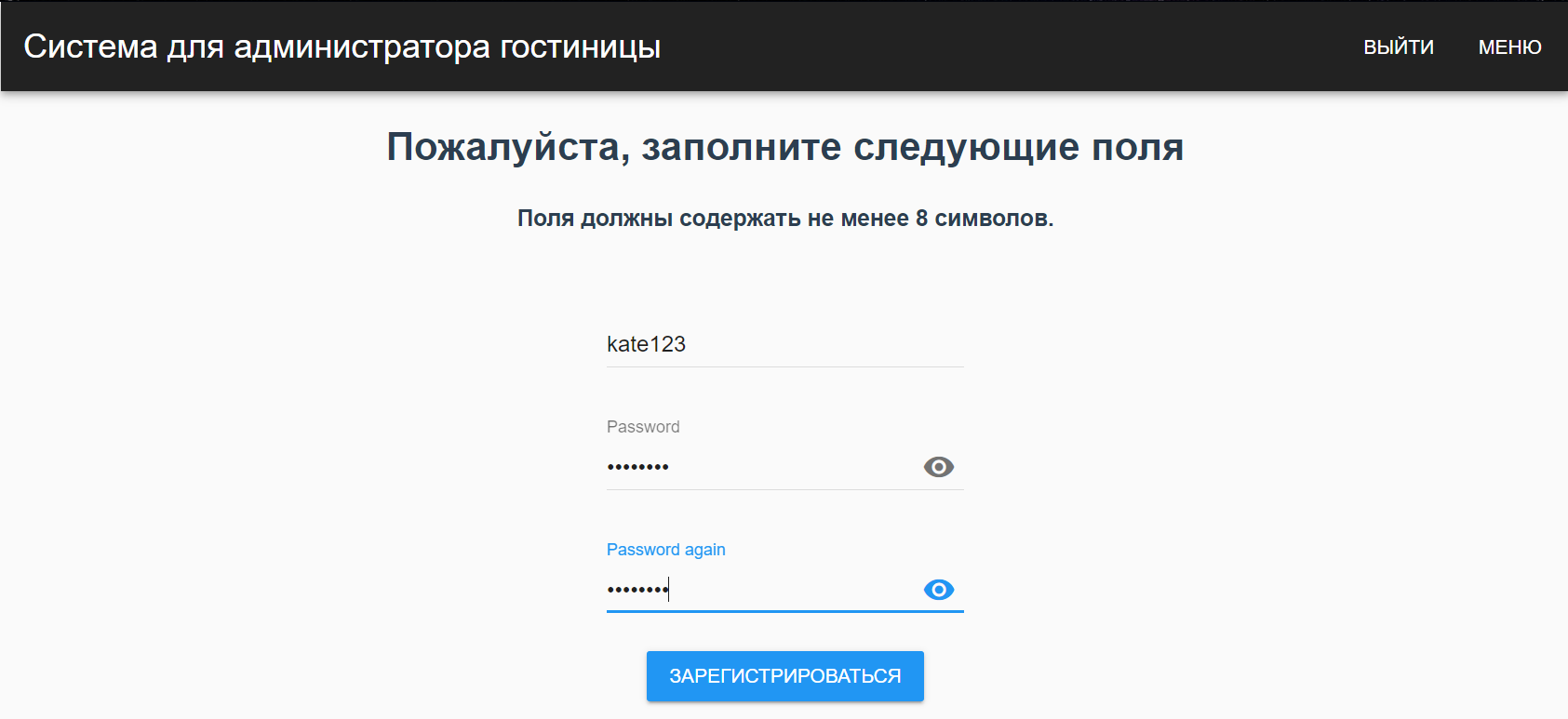
Страница регистрации нового пользователя представляет собой форму, которую необходимо заполнить для создания нового пользователя. В случае правильного заполнения всех полей и отсутствия пользователя с таким же username, пользователь будет зарегистрирован и перенаправлен на главную страницу.

Рисунок 10 – Страница регистрации нового пользователя

* 1. Просмотр номеров гостиницы

На данной странице показывается информация о номерах. При нажатии на кнопку «Показать все номера» появляется список номеров, которые находятся в данной гостинице, при нажатии на кнопку «Показать информацию о номерах» появляется подробная информация о номерах.

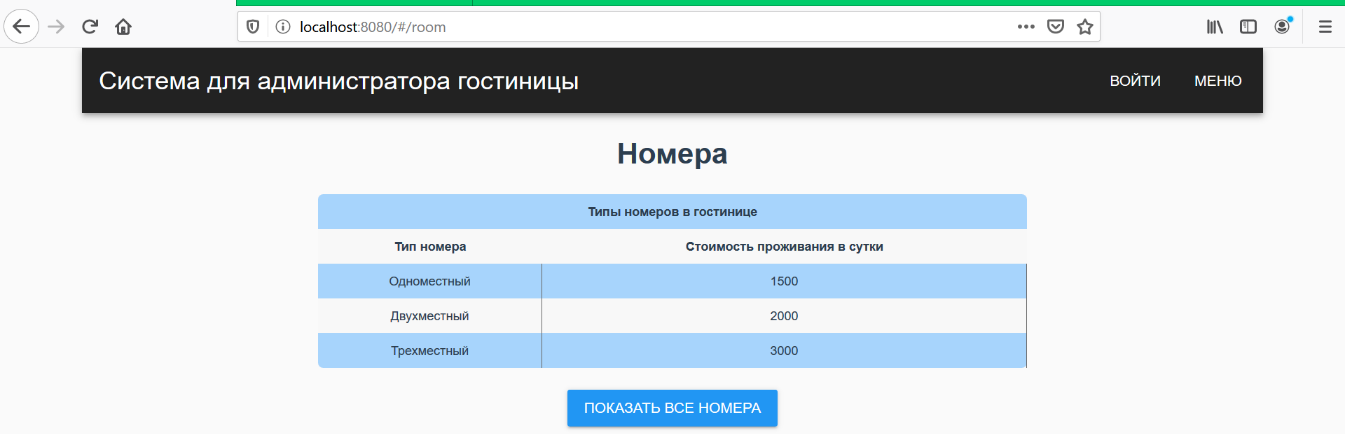
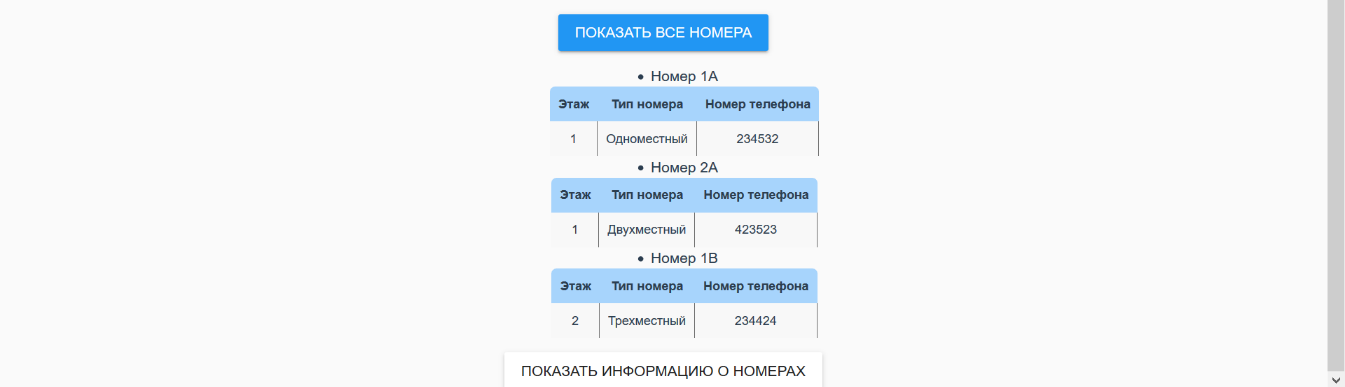


Рисунок 11 – Страница «Номера»

* 1. Просмотр всех проживающих

Страница со списком проживающих. При нажатии на кнопку «Добавить проживающего» появляется форма для создания нового проживающего, при нажатии на кнопку «Удалить проживающего» - форма удаления проживающего.

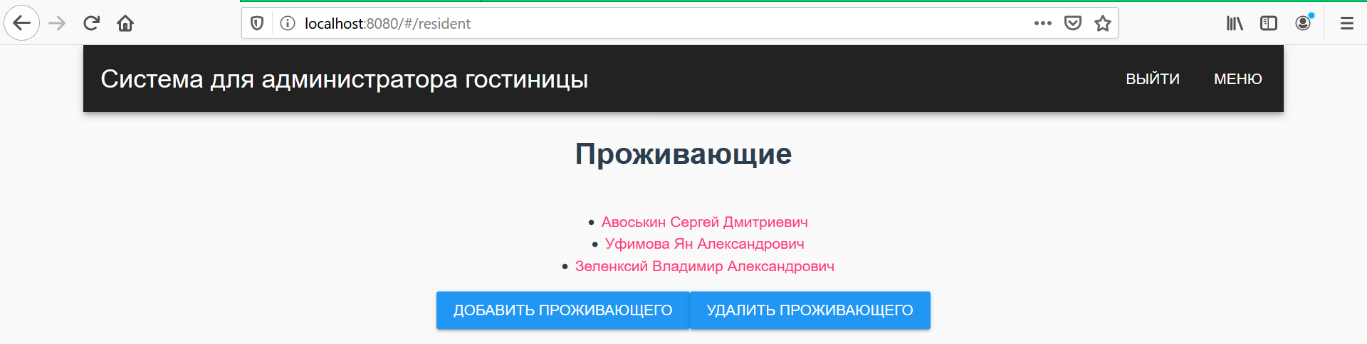
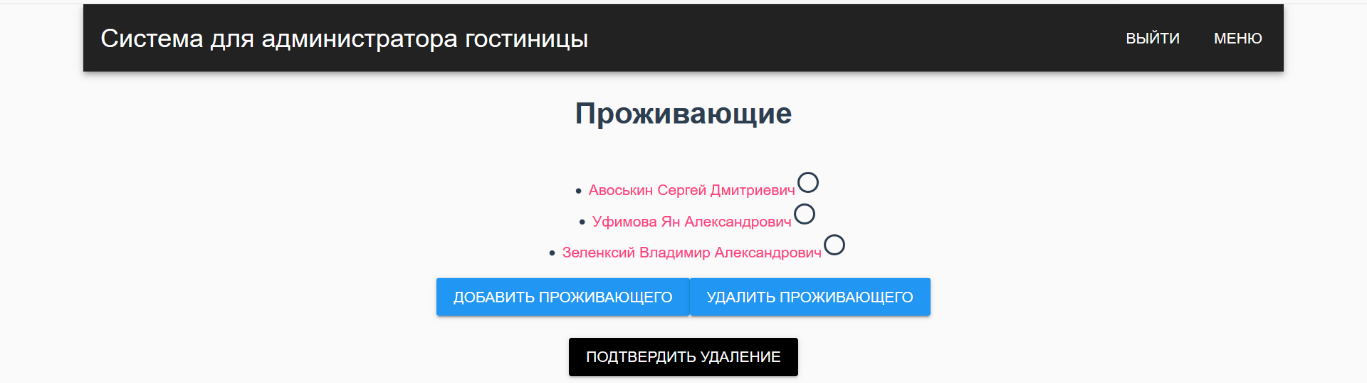


Рисунок 12 – Страница «Проживающие»

* 1. Просмотр выбранного проживающего

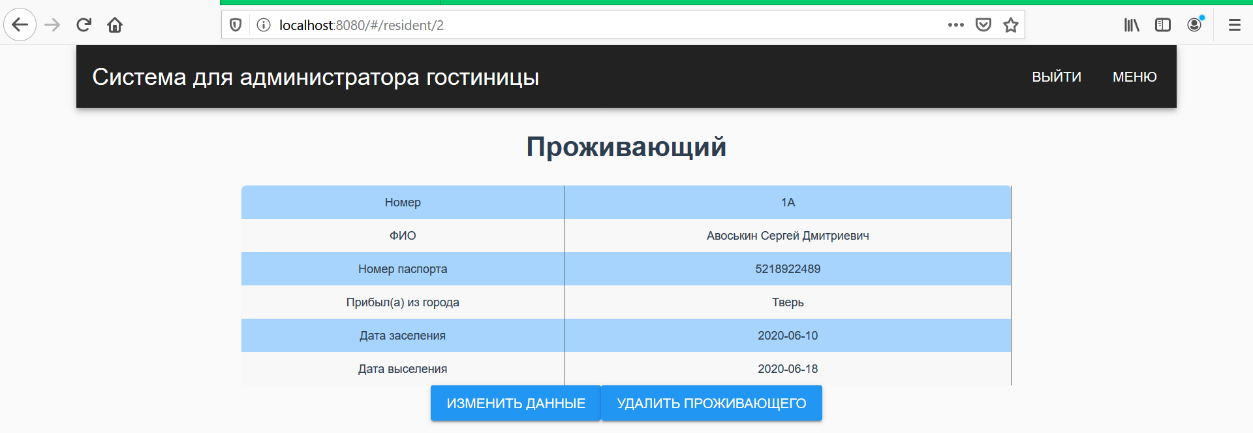
Страница с подробной информацией о проживающем. При нажатии на кнопку «Изменить данные» - появляется форма для изменения информации о проживающем, при нажатии на «Удалить проживающего» - данный проживающий удаляется и происходит перенаправление на страницу «Проживающие»

Рисунок 13 – Страница «Проживающий»

* 1. Просмотр всех служащих гостиницы

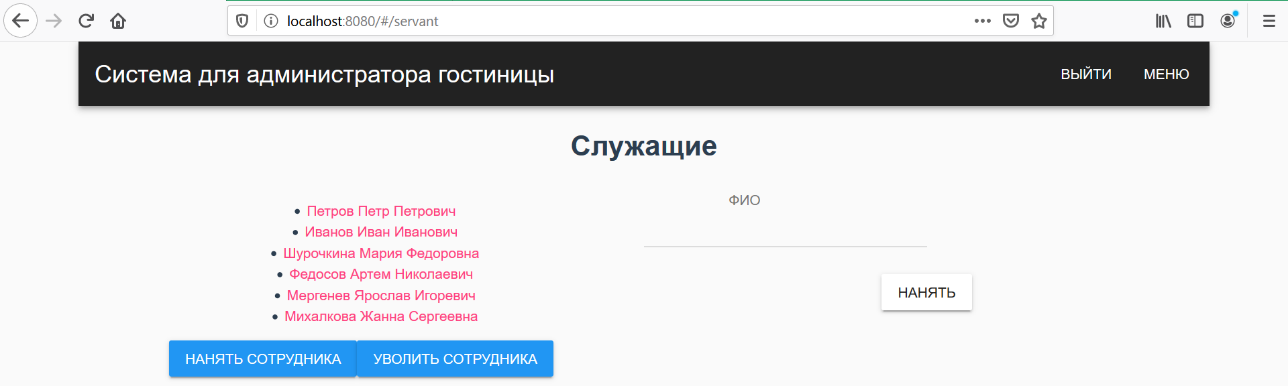
Страница со списком служащих, при нажатии на каждую запись происходит перенаправление на личную страницу служащего. Имеются также кнопки добавления и удаления служащего.

Рисунок 14 – Страница со всеми служащими гостиницы

* 1. Просмотр графика уборки

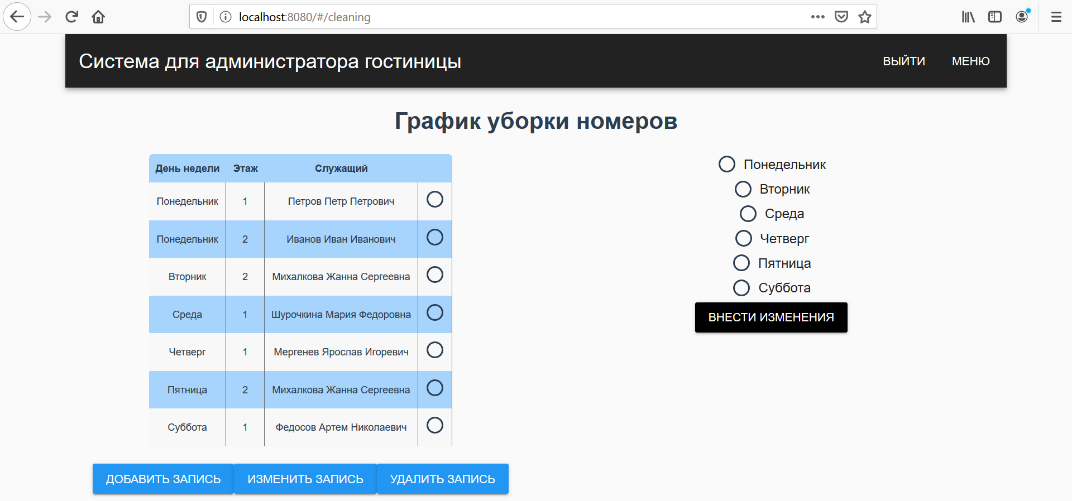
Страница с графиком уборки номеров. Кнопки «Добавить запись», «Изменить запись», «Удалить запись» реализуют соответствующие функции.

Рисунок 15 – Страница с графиком уборки номеров

**Вывод**:

В ходе выполнения лабораторной работы были получены навыки создания web-приложения с помощью web-фреймворка Django REST языка программирования Python, web-фреймворка Vue языка программирования JavaScript и JS Muse-UI.