Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение

высшего образования

Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

**Лабораторная работа №2-3**

**по дисциплине:**

**«Основы Web-программирования»**

**Выполнил**:

студент 3 курса ИКТ

группы\_К3343

Димова Алина Евгеньевна

**Проверил**:

преподаватель

Говоров Антон Игоревич

Санкт-Петербург

2020

Тема работы: реализация web-сервисов средствами Django REST framework, Vue.js, Muse-UI.

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

Практическое задание

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

Вариант 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. Реализованная модель данных представлена на рисунке 1.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Модель данных

На основе представленной схемы в файле models.py были реализованы модели в виде таблиц. Файл содержит таблицы:

|  |  |
| --- | --- |
| Название модели | Описание |
| Client | Информация о клиентах |
| Worker | Информация о работниках |
| Floor | Информация об этаже |
| Room | Информация о номере |
| Cleaning | Информация об уборке (какой работник какой этаж убирает) |
| Checkin | Информация о заселении (какой клиент в какой номер) |

1. Следующим этапом разработки системы было создание сериализаторов в файле serializers.py и представлений в файле views.py. Для создания сериализаторов был использован класс ModelSerializer, а для представлений использованы APIView и generics.ListAPIView.

|  |  |
| --- | --- |
| Название представления | Назначение |
| Rooms | Вывод списка номеров |
| Clients | Вывод списка клиентов и добавление нового клиента |
| ClientsList | Фильтр клиентов |
| FloorsView | Вывод всех имеющихся этажей |
| RoomDetailsView | Информация об определенном номере |
| ClientDetailsView | Информация об определенном клиенте |
| WorkersView | Вывод всех работников и добавление работника |
| DelWorker | Удаление работника |
| CleaningTable | Вывод списка уборок одного работника и изменение расписания уборок |
| AddCheckin | Вывод списка заселений и добавление заселения |
| CheckinList | Вывод списка заселений по клиентам |
| CleaningsView | Вывод списка уборок по дням |

1. Итоговый файл urls.py содержит такие пути как:

|  |  |
| --- | --- |
| URL | Куда направляет |
| Clients/ | На страницу со всеми клиентами |
| client\_detail/ | На страницу с информацией об одном клиенте |
| clients\_filter/ | Фильтрует клиентов по городу |
| Rooms/ | На страницу со всеми номерами |
| Workers/ | На страницу со всеми работниками |
| del\_worker/ | На страницу удаления работника |
| room\_detail/ | На страницу с информацией об одном номере |
| Cleanings/ | На страницу с уборками одного работника |
| cleanings\_view/ | На страницу с уборками за один день |
| Checkin/ | На страницу со всеми заселениями |
| checkin\_filter/ | Фильтрует заселения по клиентам |
| Floors/ | На страницу со всеми этажами |

Теперь рассмотрим фронтэнд реализованной системы.

В начале работы необходимо войти в систему. Пример стартовой страницы представлен на рисунке 2.

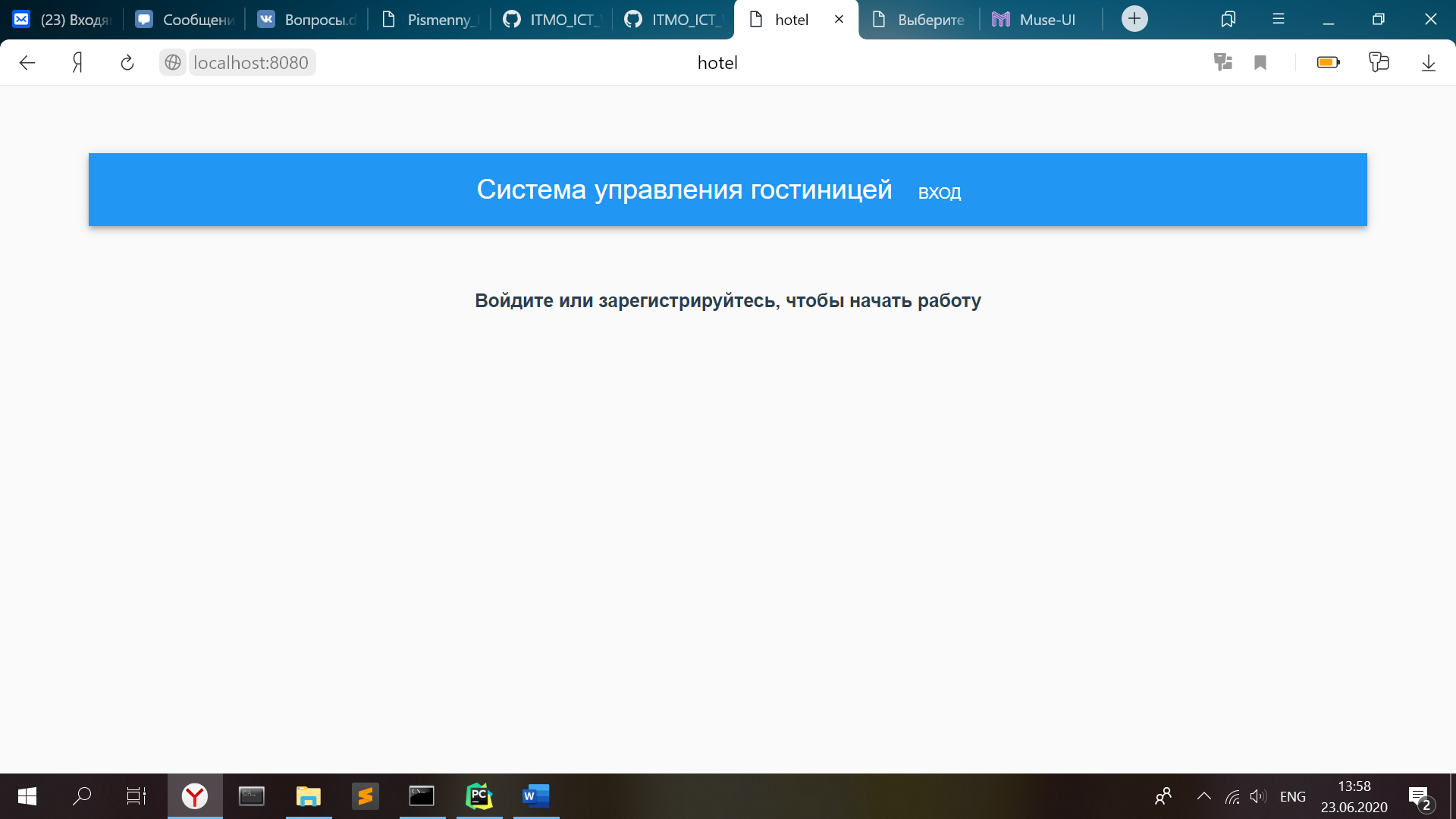


Рисунок 2 – Стартовая страница

После нажатия на кнопку Вход открывается страница входа и регистрации (рис. 3).

В начале страницы есть форма для входа. Если же пользователь не зарегистрирован, то на этой же странице имеется форма регистрации. После регистрации пользователю предлагается войти в свой аккаунт.

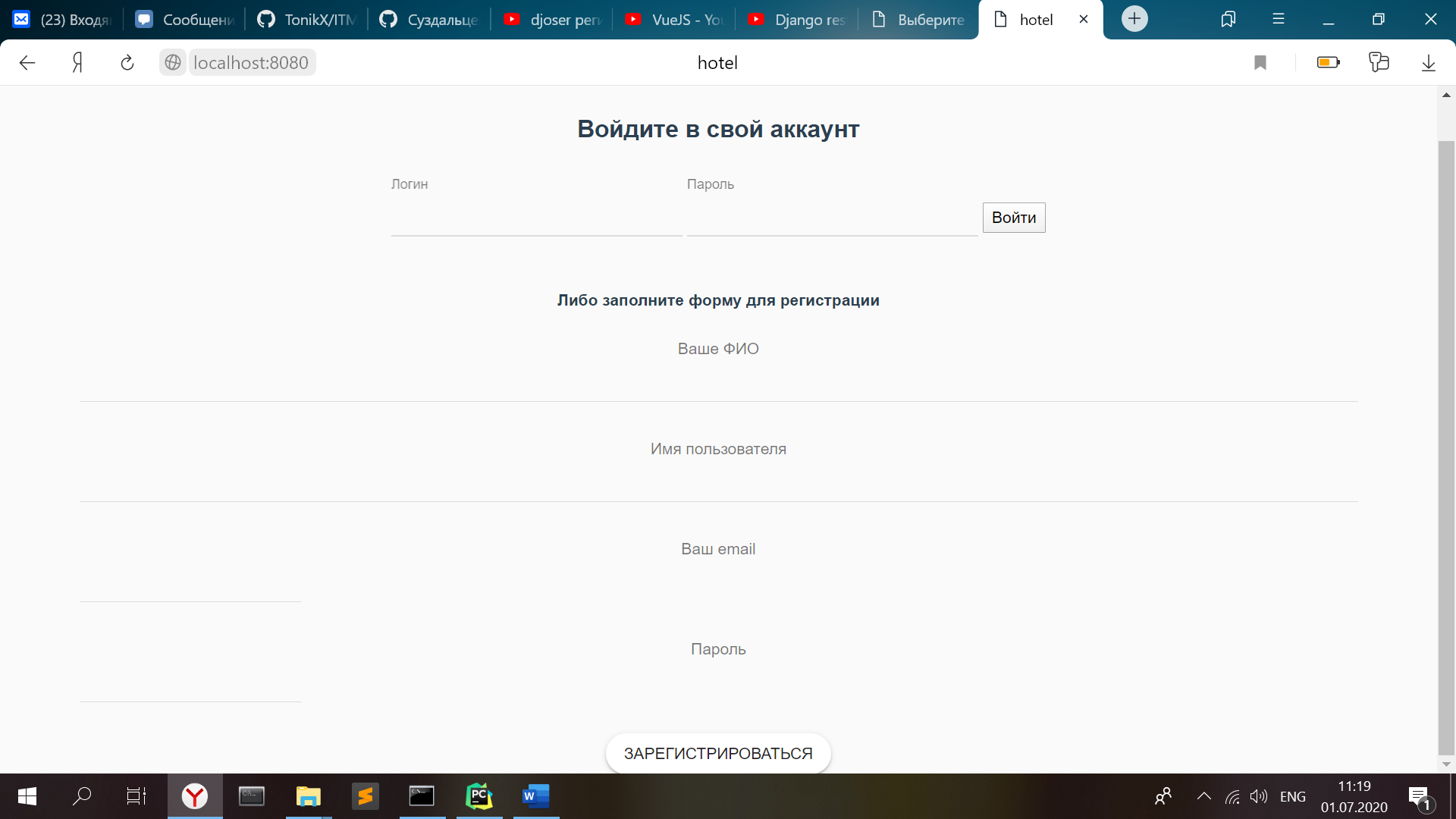


Рисунок 3 – Страница входа и регистрации

После входа на главной странице появляется список комнат с их описанием и историей заселений. На рисунке 4 представлен пример страницы со списком комнат.

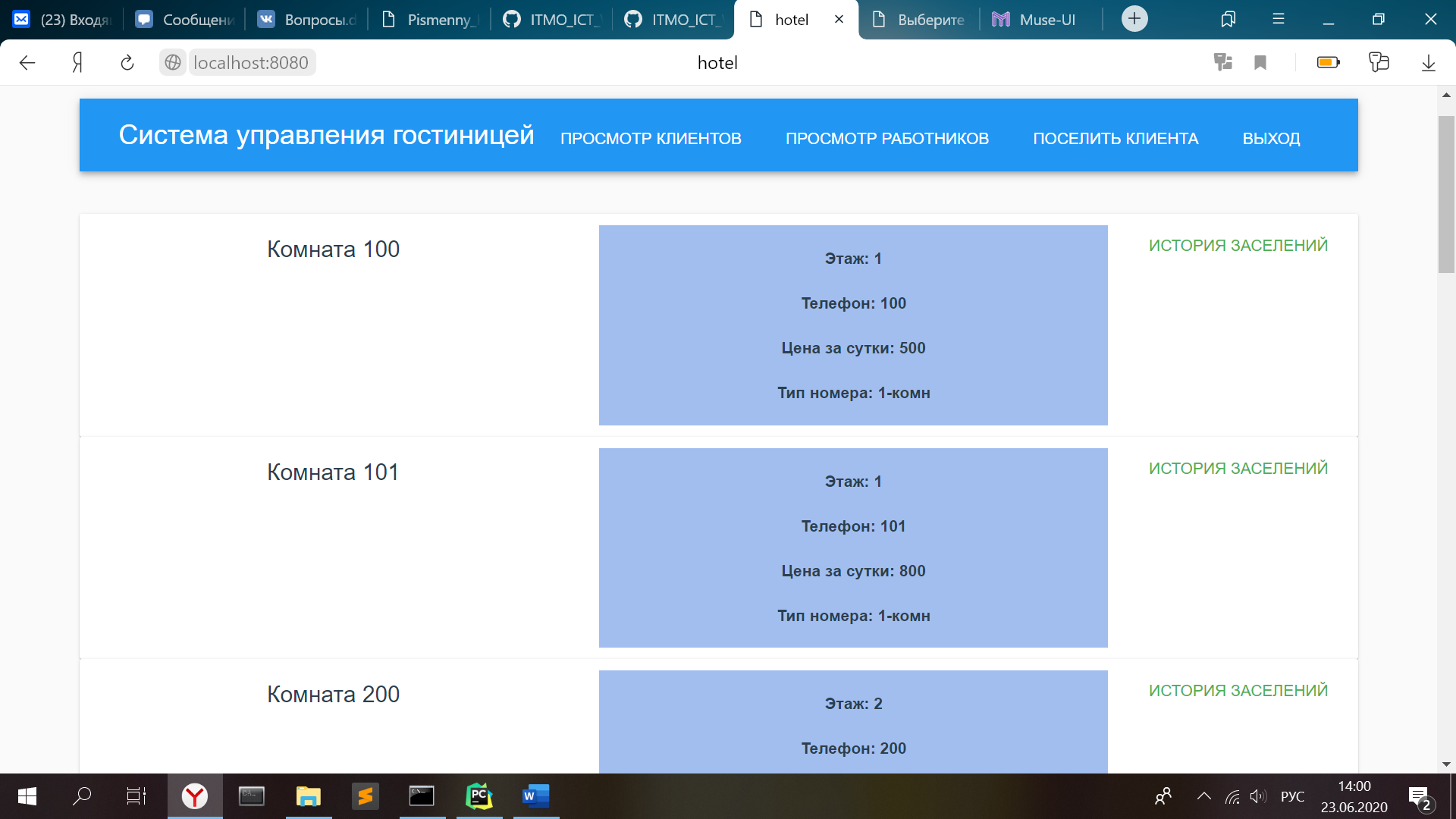


Рисунок 4 – Главная страница

Кнопка История заселений открывает список клиентов и даты их проживания в данном номере. На рисунке 5 представлен пример открытой истории заселений.

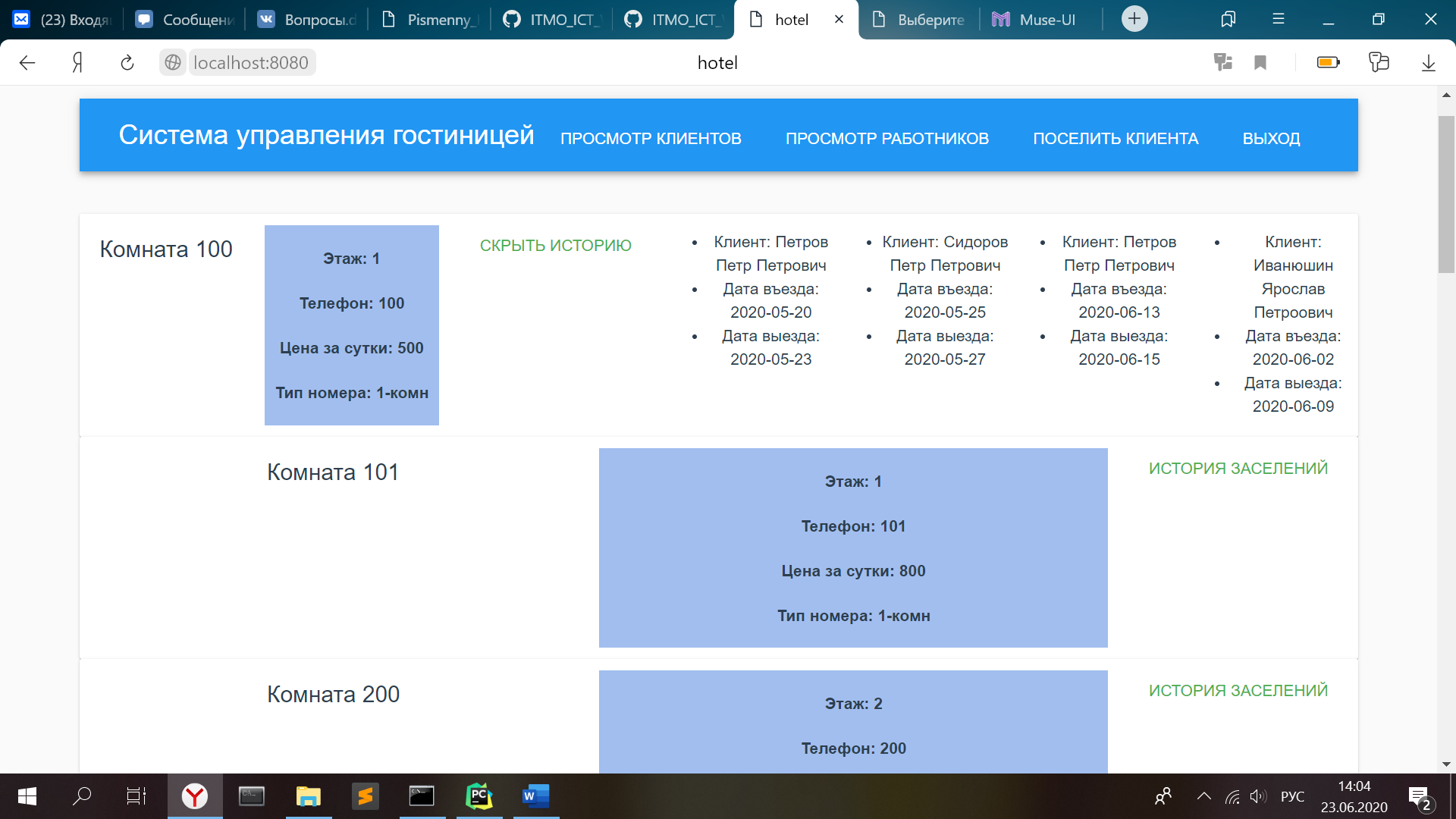


Рисунок 5 – Открытая история заселений

При нажатии на пункт меню Просмотр клиентов открывается страница со списком имеющихся клиентов. На рисунке 6 представлен пример страницы с клиентами.

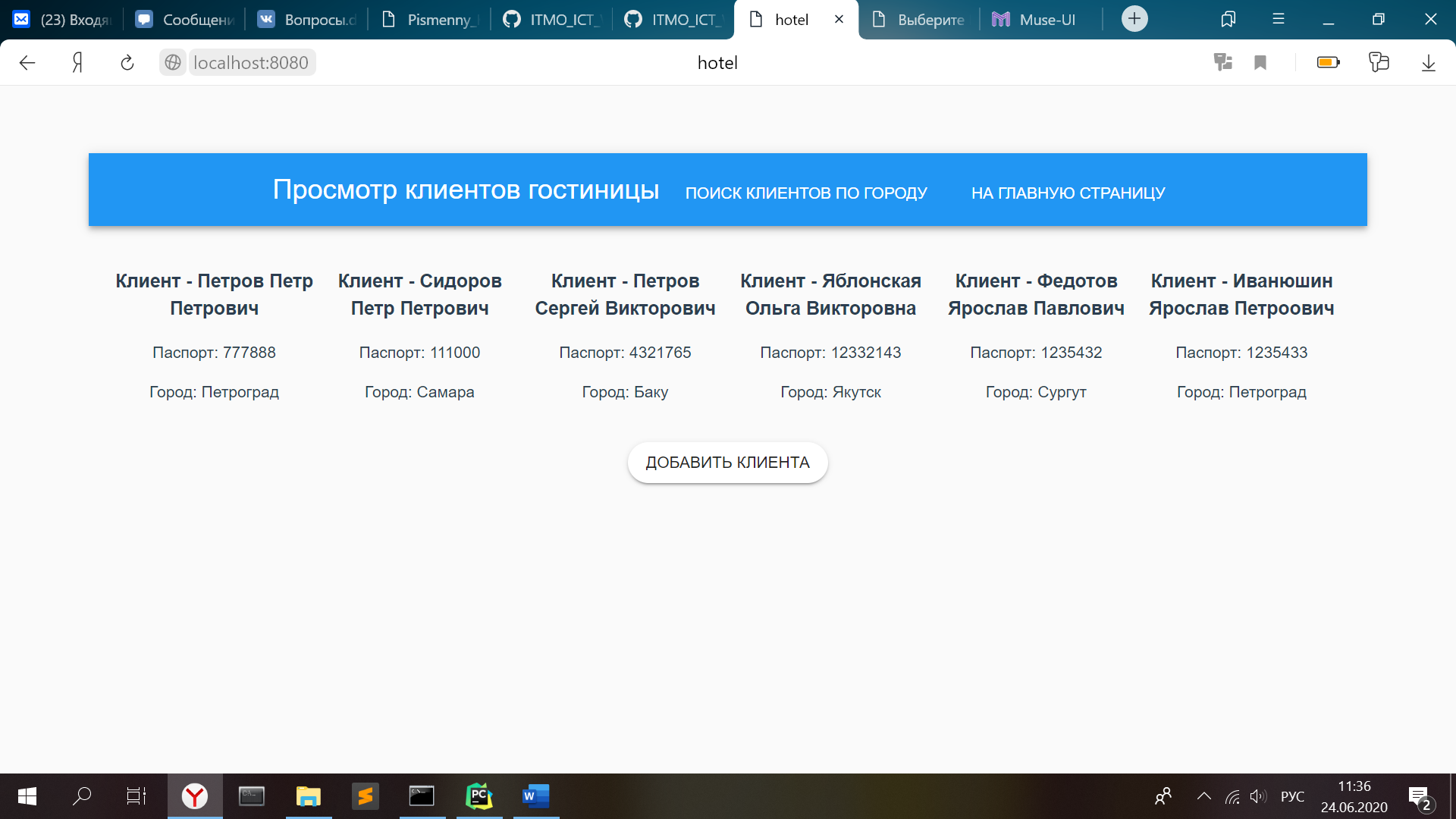


Рисунок 6 – Страница со списком клиентов

Реализовано добавление клиентов в систему. Форма добавления клиента представлена на рисунке 7.

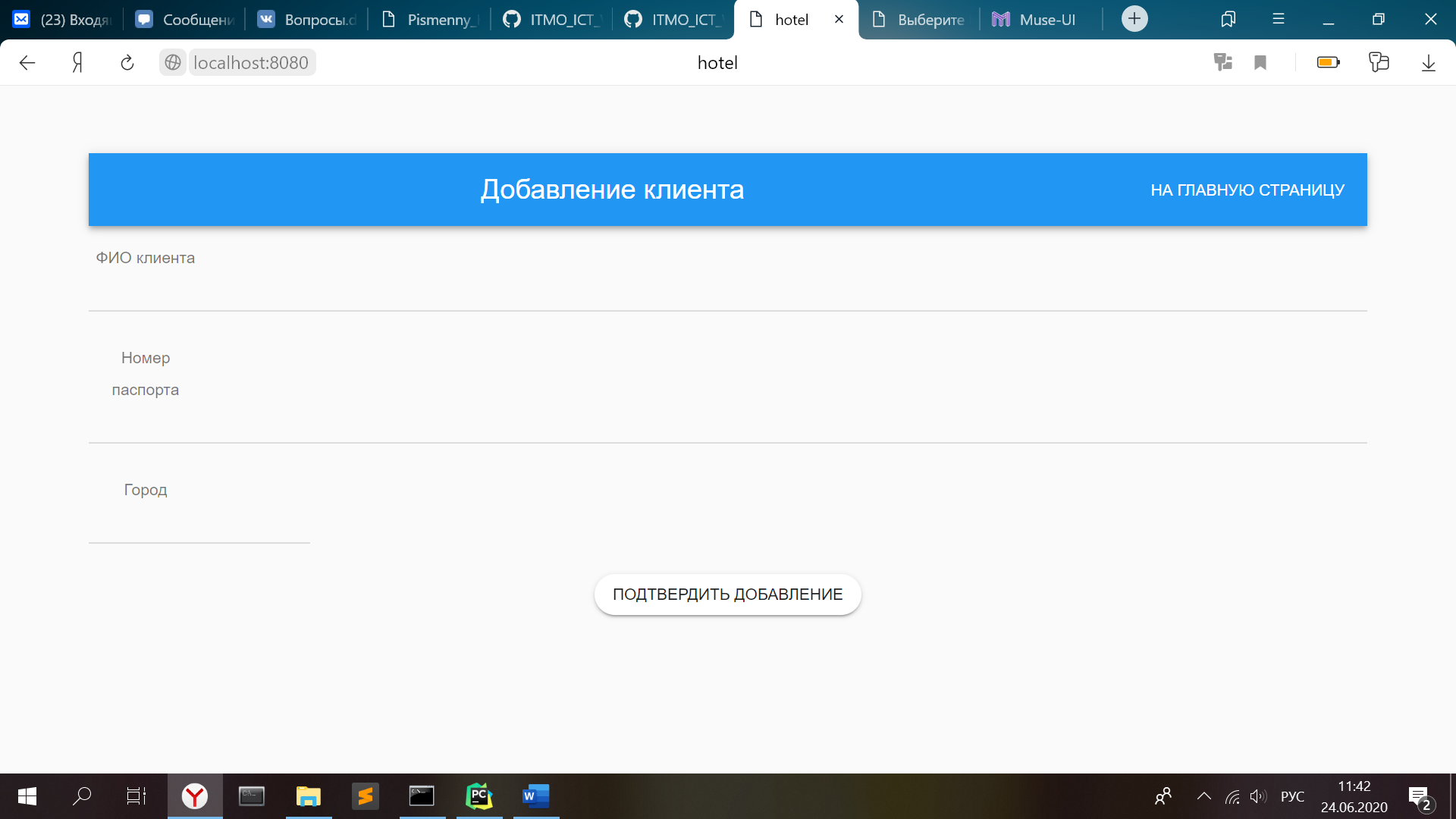


Рисунок 7 – Страница добавления клиента

В системе реализован поиск клиентов по городу. Чтобы найти нужного клиента, необходимо написать полное название города, из которого он прибыл. Пример поиска представлен на рисунке 8.

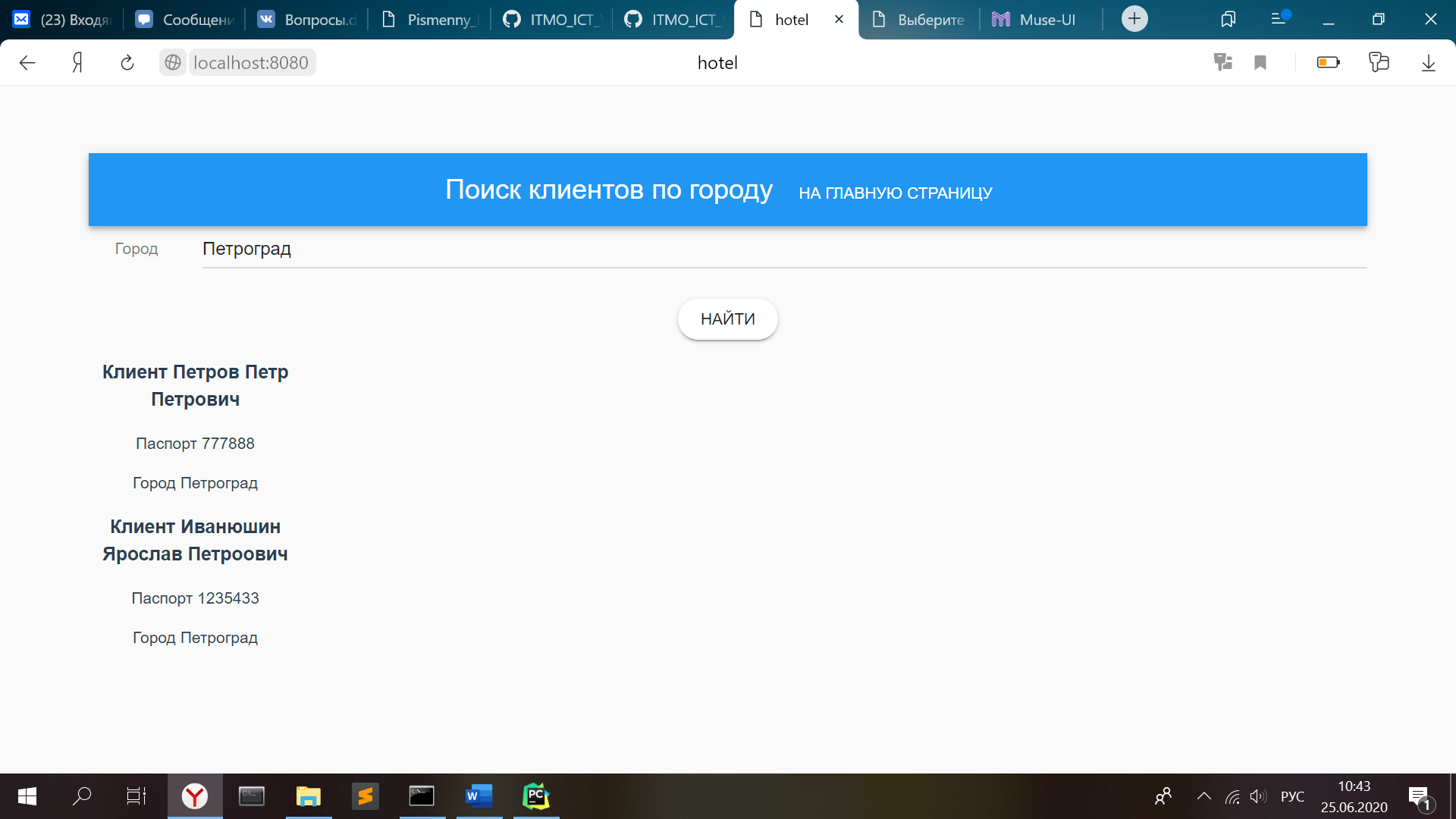


Рисунок 8 – Поиск клиентов

В системе предусмотрена возможность заселения клиента в номер. Для этого нужно нажать на кнопку Поселить клиента на основной странице и заполнить форму (рис. 9). В форме предлагается выбрать номер комнаты, клиента и даты заезда и выезда. Чтобы заселить клиента в комнату, в системе должна храниться информация о клиенте. То есть при поселении нового клиента, его сначала нужно добавить.

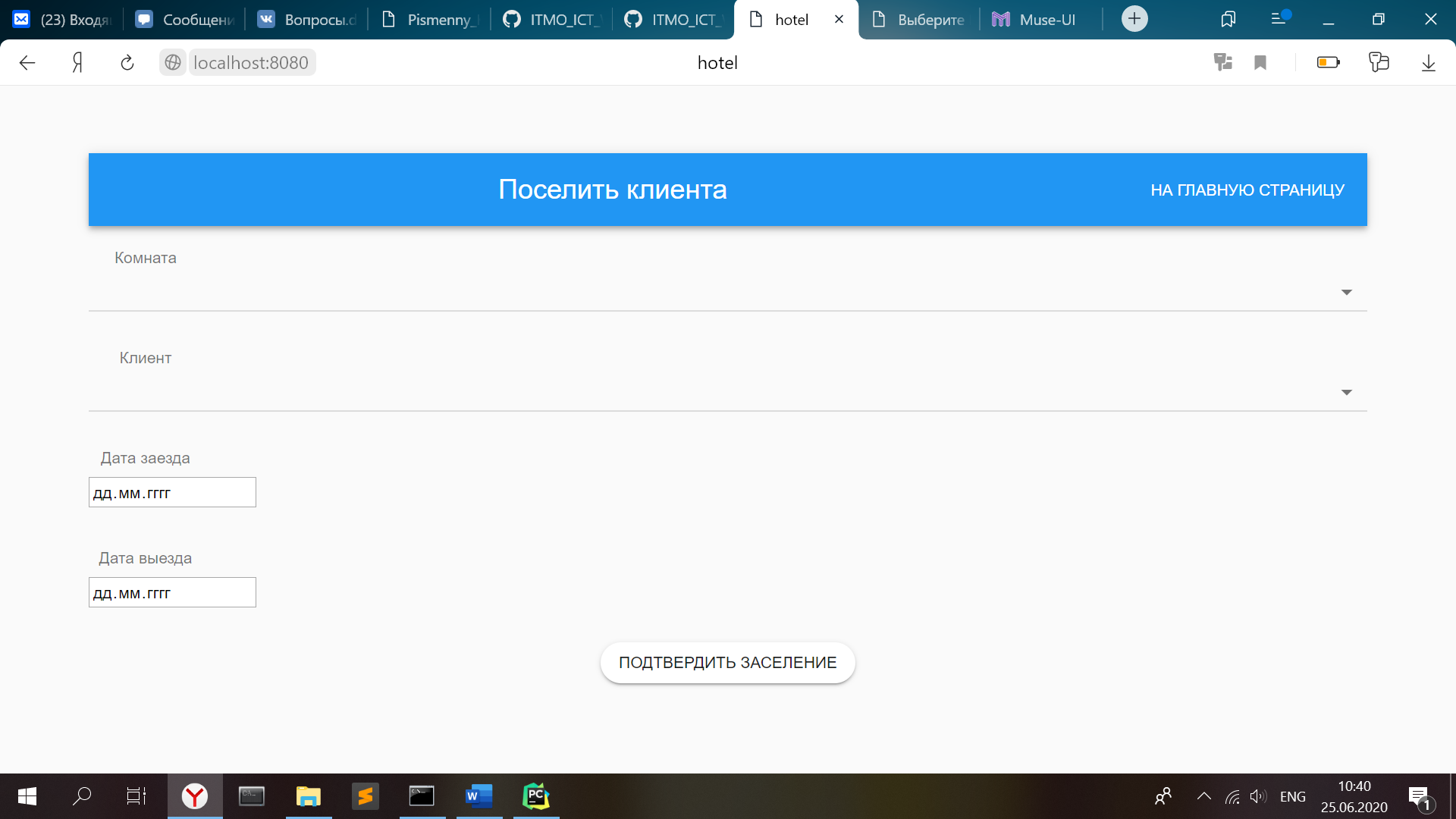


Рисунок 9 – Форма заселения клиента

Кроме работы с номерами и клиентами в системе возможно просмотреть список имеющихся работников и расписание их уборок. Список уборок отсортирован по дням недели. В один день один работник убирает только один этаж.

После расписания каждого работника есть кнопка Удалить работника. Если на нее нажать, то определенный работник удаляется из системы и его расписание стирается.

Пример страницы представлен на рисунке 10.

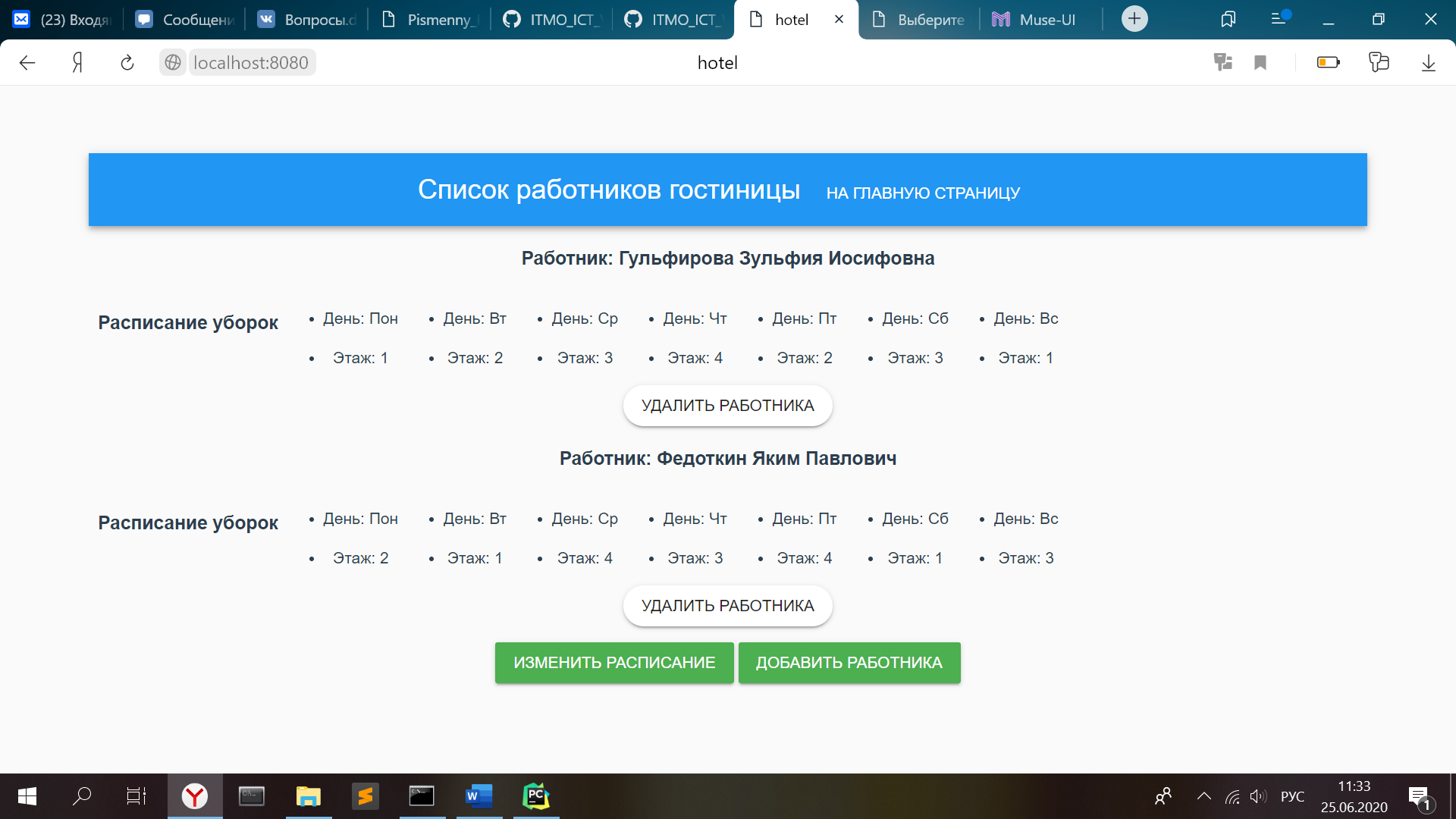


Рисунок 10 – Просмотр работников

Реализовано добавление нового работника. На рисунке 11 представлена форма добавления работника.

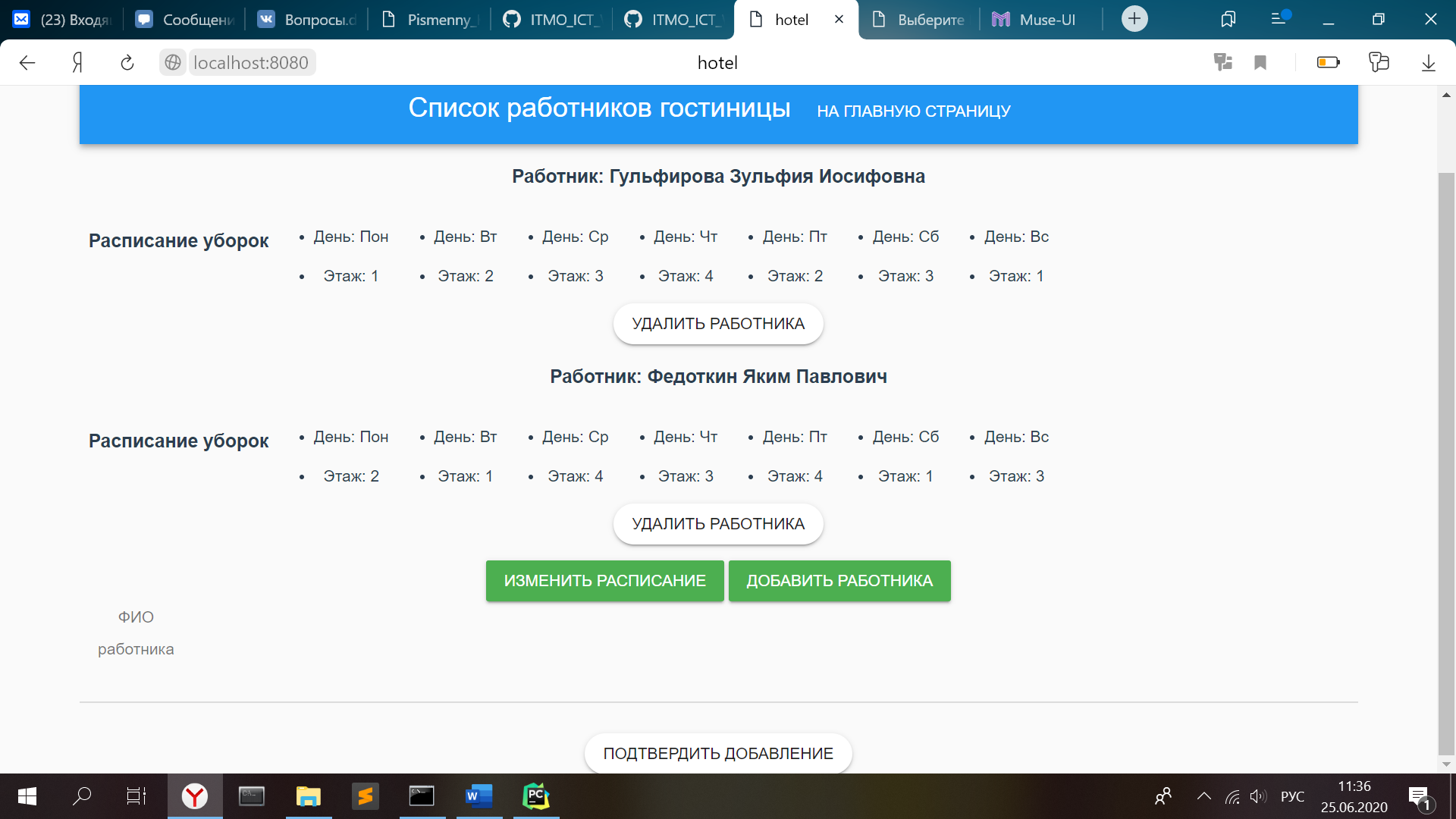


Рисунок 11 – Форма добавления работника

Также возможно изменить расписание уборок определенного работника. Так как в один день работник убирает только один этаж, при изменении расписания старое расписание не сохраняется.

Форма изменения расписания работника представлена на рисунке 12.

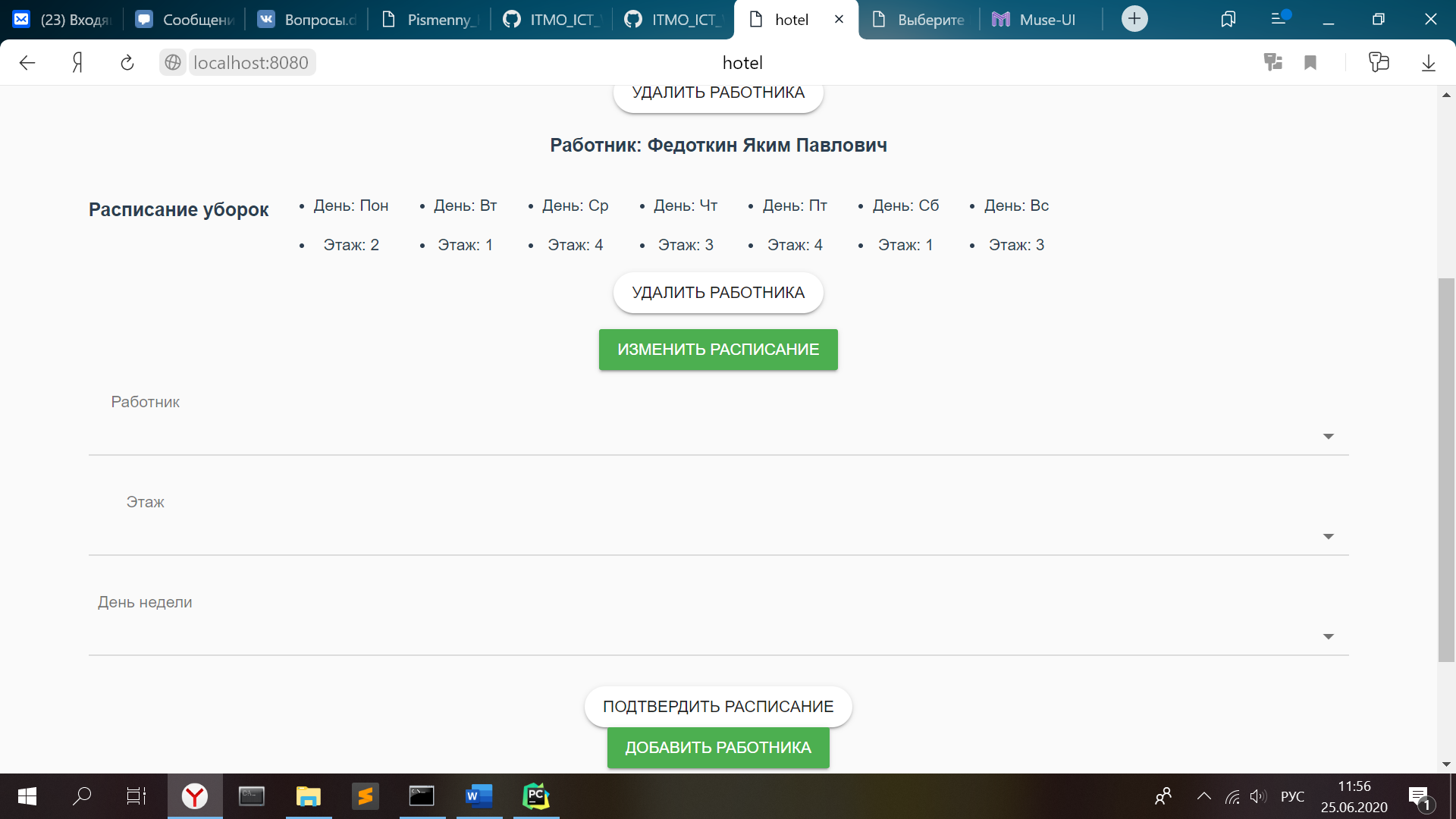


Рисунок 12 – Форма изменения расписания работника

**Вывод**

В ходе выполнения лабораторной работы получены практические навыки реализации веб-сервисов с помощью Django REST, Vue.js и Muse UI.