

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**ФАКУЛЬТЕТ ИКТ**

**Отчет по лабораторной работе №2-3  
по дисциплине «Основы web-программирования»**

**Выполнил:  
Рейбандт А.А.  
Студент группы К3342**

**Проверил:  
Говоров А.И.**

**Санкт-Петербург  
2020 г.**

## Практическое задание:

Создать программную систему, предназначенную для администрации аэропорта некоторой компании-авиаперевозчика.

## Ход выполнения работы:

- была реализована главная вкладка сайта, содержащая общую информацию о рейсах, представленных в виде карточек:

### Список рейсов

Рейс № 15158 авиакомпании S7

Москва --> Краснодар

**Дата и время вылета:** 2020-05-07T06:00:00Z. **Дата и время прибытия:** 2020-05-07T07:10:00Z. **Длительность полета:** 1:10:00

**Количество проданных билетов/общее количество мест:** 200/200.

[Подробнее](#)

Рейс № 87592 авиакомпании Utair

Санкт-Петербург --> Ростов-на-Дону

**Дата и время вылета:** 2020-05-21T12:00:00Z. **Дата и время прибытия:** 2020-05-21T13:45:00Z. **Длительность полета:** 1:45:00

**Количество проданных билетов/общее количество мест:** 188/400.

[Подробнее](#)

- пример полного описания рейса:

# Информация о рейсе № 15158

- Пункт вылета: **Москва**
- Пункт назначения: **Краснодар**
- Дистанция: **1319 км**
- Авиакомпания: **S7**
- Дата и время вылета: **2020-05-07T06:00:00Z**
- Дата и время прибытия: **2020-05-07T07:10:00Z**
- Общее количество мест: **200**
- Количество проданных билетов: **200**

- 
- Прямой: **да**

- 
- Самолет: **Ty-154**
  - Экипаж:
    - Главный пилот: **FromDjango**
    - Второй пилот: **Ragnar**
    - Стюардессы: **July, Hope**
    - Стюард: **Jonny**

- также был реализован ряд фильтров, позволяющих упростить взаимодействие с сайтом. В их числе показ рейсов без пересадок, а также рейсов только со свободными местами. Было принято решение вынести их в *store* (инструмент *VueX*) для большей универсальности и принципа DRY. Фрагмент кода, воплощающий описанные фильтры в жизнь представлен ниже.

```

const store = new Vuex.Store( options: {
  state: {
    backendUrl: "http://127.0.0.1:8000/api/v1",
    filters: {},
    flights: [],
    profile: {},
  },
  mutations: {
    setFilters(state, filters) {
      state.filters = {
        ...state.filters,
        ...filters,
      };
    },
    setFlights(state, flights) {
      state.flights = flights;
    },
    setAirlineCompanies(state, airline_companies) {
      state.airline_companies = airline_companies;
    },
    setProfile(state, profile) {
      state.profile = profile;
    }
  },
  actions: {
    async getFlights({commit, state}, filters) {
      commit('setFilters', filters);

      const formData = new FormData();
      Object
        .keys(filters || {})
        .forEach(filter => formData.append(mapFilterToQueryParam(filter), filters[filter]));

      const queryString = new URLSearchParams(formData).toString();

      const flights = await fetch( input: `${state.backendUrl}/flight/?${queryString}`);
      commit('setFlights', await flights.json());
    },
  },
});

```

- была добавлена возможность кастомной регистрации и авторизации с последующей выдачей токена. В процессе реализации последней использовался инструмент *AuthToken*, предоставляемый Django Rest Framework'ом (DRF)

## Регистрация

Логин

Имя

Фамилия

Возраст

Паспортные данные

Пароль

Зарегистрироваться

## Авторизация

Логин

Пароль

Войти

- помимо этого, был разработан личный кабинет пользователя, содержащий актуальную информацию о нем, а также была предусмотрена возможность подать заявку на работу:

## Профиль пользователя

- Имя: **Иван Ургант**
- Возраст: **33**
- Паспортные данные: **456321789**
- Текущий статус: **Пользователь сайта**

Хочу работать у вас!

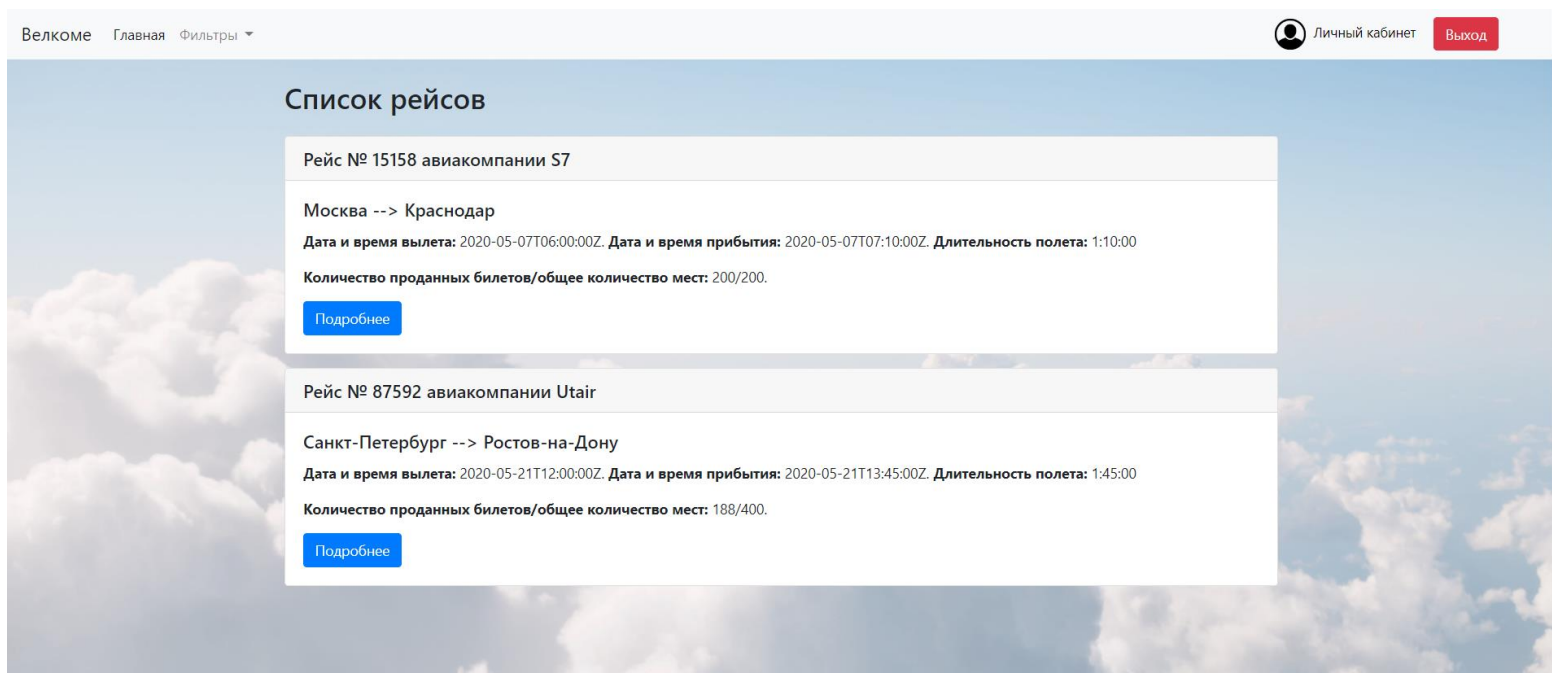
## Для подачи заявки заполните следующие поля

Образование

Желаемая должность

Подать заявку

- в итоге разработки получился функциональный сайт со следующим дизайном:



## Вывод:

В результате работы была реализована программную систему, предназначенную для администрации аэропорта. Во время разработки для написания серверной части (бэк-энда) проекта использовался Django + DRF, клиентская же часть (фронт-энд) был реализован с помощью Vue.js.