

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина: Фронт-энд разработка**

**Отчет**

**Лабораторная работа 3**

**Выполнил: Пластун Елизавета**

**Группа: К33402**

**Проверил: Добряков Д. И.**

**Санкт-Петербург**

**2023 г.**

# Задача Разработка одностраничного веб-приложения

## Ход работы

Для бекенд части был выбран вариант из дисциплины web-development. Бек реализован на Django. Для фронтенд части был выбран фреймворк react. Для роуминга была использована библиотека react route.

Реализован следующий функционал:

- Авторизация пользователя
- Регистрация пользователя
- Деавторизация пользователя
- Просмотр рейсов
- Просмотр оставшихся мест на рейс
- Создание рейсов
- Создание маршрутов
- Удаление рейсов
- Редактирование рейсов
- Удаление маршрутов

Продемонстрирую работу в браузере:

Авиасейлс

liza

Выйти

Рейсы

Маршруты

Места на рейс

Создать полет

16:18	19:19	россия → турция	Редактировать	Удалить
04.12.2023	26.03.2024			
14:20	20:20	англия → сша	Редактировать	Удалить
15.03.2024	17.03.2024			

## Роутер:

```
function App() {
  return (
    <div>
      <Routes>
        <Route path="/">
          <Route path="/login" element={<Login />} />
          <Route path="/registration" element={<Registration />} />

          <Route element={<PrivateRoute />>
            <Route index element={<Main />} />
            <Route path="/flight/new" element={<FlightForm />} />
            <Route path="/flight/edit/:flightId" element={<FlightEditForm />} />

            <Route path="/route/new" element={<RouteForm />} />
          </Route>
        </Route>
      </Routes>
    </div>
  );
}
```

Все маршруты пользователя

К примеру при переходе по адресу `/flight/new` пользователь будет переходить на форму создания рейса.

## Компоненты:

```
return (
  <div className="cell-container">
    <div className="cell-time">
      <div className="cell-datetime">
        <div className="cell-hours">
          {formatDateTime(flight.departure_datetime)[0]}
        </div>
        <div className="cell-date">
          {formatDateTime(flight.departure_datetime)[1]}
        </div>
      </div>
      <div className="cell-datetime">
        <div className="cell-hours">
          {formatDateTime(flight.arrival_datetime)[0]}
        </div>
        <div className="cell-date">
          {formatDateTime(flight.arrival_datetime)[1]}
        </div>
      </div>
    </div>
  </div>
)
```

```

    </div>
  </div>
  <div className="cell-route">
    <div>{routeData.departure_point}</div>
    <img src={Arrow} alt="" className="arrow" />
    <div>{routeData.destination_point}</div>
  </div>
  <div className="cell-button-group">
    <span
      className="cell-button"
      onClick={() => navigate('flight/edit/${flight.id}')}
    >
      Редактировать
    </span>
    <span
      className="cell-button"
      onClick={() => onDelete(flight)}
    >
      Удалить
    </span>
  </div>
</div>
);
}

export default FlightCell;

```

Это код компонента ячейки рейса:

16:18	19:19	россия → турция	Редактировать	Удалить
04.12.2023	26.03.2024			

Для взаимодействия с api используем встроенную библиотеку fetch.

```

const getRoutes = useCallback(() => {
  fetch("http://localhost:8000/api/routes/all/", {
    method: "GET",
    headers: {
      Authorization: `Token ${token}`,
    },
  })
  .then((response) => response.json())
  .then((result) => {
    setRoutes(result);
  })
  .catch((error) => console.error("Error fetching data:", error));
}, [token]);

```

Вот пример запроса, который возвращает все маршруты.

**вывод:** была реализована фронтенд часть для сервиса авиасейлс.