

**Министерство образования и науки Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО**  
**ОБРАЗОВАНИЯ**  
**УНИВЕРСИТЕТ ИТМО**

Факультет инфокоммуникационных технологий

Образовательная программа 09.03.02

Направление подготовки (специальность) Мобильные сетевые технологии

**О Т Ч Е Т**

о курсовой работе

Тема задания: реализация одностраничного веб-приложения (SPA) с использованием фреймворка React

Обучающийся Коровин Александр Игоревич, К33402

Руководитель: Добряков Д. И., преподаватель

Оценка за курсовую работу \_\_\_\_

Дата \_\_\_\_

Санкт-Петербург  
2022

# ГЛАВА 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

## 1 Средства разработки

ITMO\_ICT\_WebDevelopment\_2021-20221 курс, в рамках которого мы разрабатывали сервис для выбранного нами приложения, позволил мне реализовать серверную часть приложения. Клиентская часть написана на React.

## 2 Функциональные требования

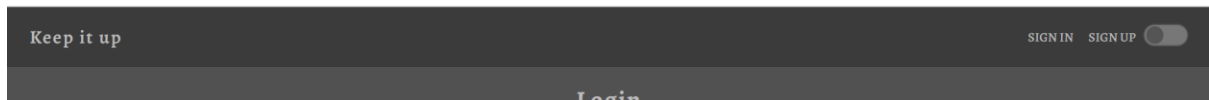
Функциональные требования по этому проекту заключались в наличие 9 компонентов в

1. Разработка одностраничного веб-приложения (SPA) с использованием фреймворка Vue.JS
2. Использование миксинов (необязательно, но желательно)
3. Использование Vuex (необязательно, но желательно)
4. В проекте должно быть, как минимум, 10 страниц (обязательно)

## 3 Проектирование и реализация серверной части

Используемый стек технологий: Django, Django-Rest, React.js, Axios, React router dom

## Основная часть



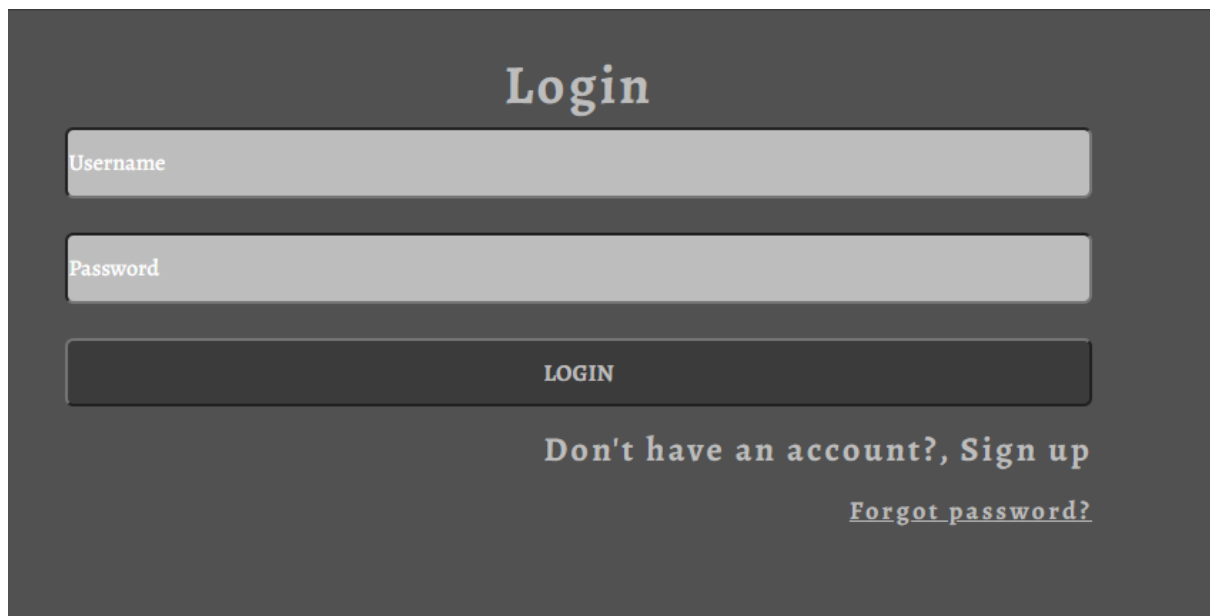
Хедер до логина



Хедер после

```
<>
<nav className="navbar" sx={{backgroundColor: 'nav'}}>
  <div className="logo"><Link sx={{color: 'text'}} to="/" title="Home">Keep it up</Link></div>
  {user ? (<ul>
    <li><Link sx={{color: 'text'}} to="/bookings" title="Bookings">BOOKINGS</Link></li>
    <li><Link sx={{color: 'text'}} to="/profile" title="Profile">PROFILE</Link></li>
    <li><Link sx={{color: 'text'}} to="/logout" onClick={logout} title="Logout">LOGOUT</Link></li>
    <li><ToggleThemeSwitchBox/></li>
  </ul>) :
  (
    <ul>
      <li><Link sx={{color: 'text'}} to="/login" title="Sign in">SIGN IN</Link></li>
      <li><Link sx={{color: 'text'}} to="/register" title="Sign in">SIGN UP</Link></li>
      <li><ToggleThemeSwitchBox/></li>
    </ul>
  )
}
</nav>
</>
```

Код хедера



A login form on a dark gray background. At the top, the word "Login" is centered in a large, bold, light gray serif font. Below it are two light gray rectangular input fields. The first field is labeled "Username" in a small, light gray sans-serif font. The second field is labeled "Password" in a small, light gray sans-serif font. Below these fields is a dark gray rectangular button with the word "LOGIN" in a light gray, all-caps, sans-serif font. At the bottom, the text "Don't have an account?, Sign up" is centered in a light gray sans-serif font, followed by the text "[Forgot password?](\"#\")" in a light gray sans-serif font.

Login

Username

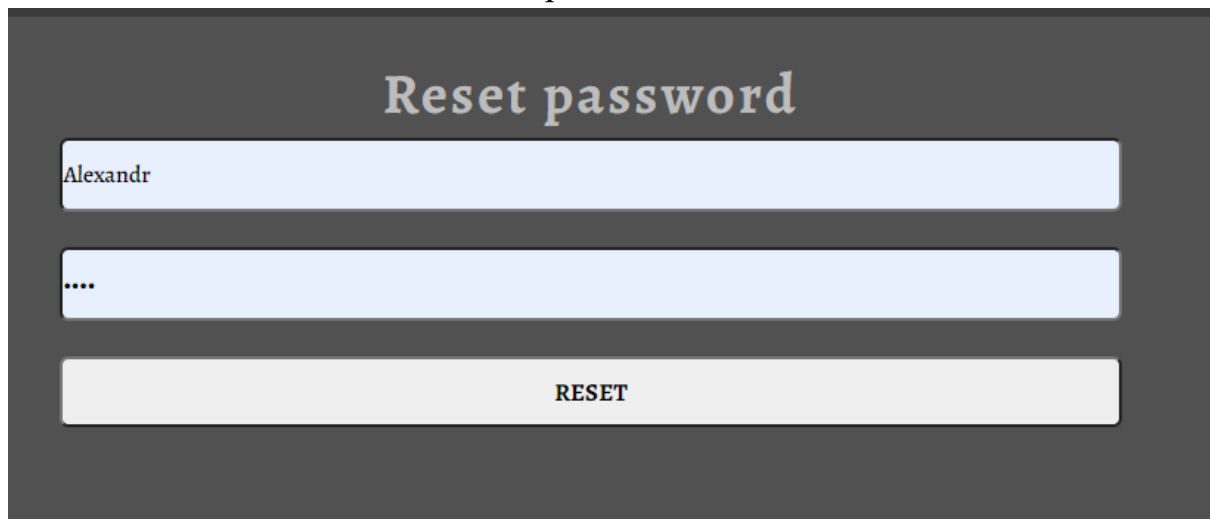
Password

LOGIN

Don't have an account?, Sign up

[Forgot password?](#)

Форма логина



A "Reset password" form on a dark gray background. At the top, the text "Reset password" is centered in a large, bold, light gray serif font. Below it are two light blue rectangular input fields. The first field contains the text "Alexandr" in a small, light gray sans-serif font. The second field contains four dots "...." in a small, light gray sans-serif font. Below these fields is a light gray rectangular button with the word "RESET" in a dark gray, all-caps, sans-serif font.

Reset password

Alexandr

....

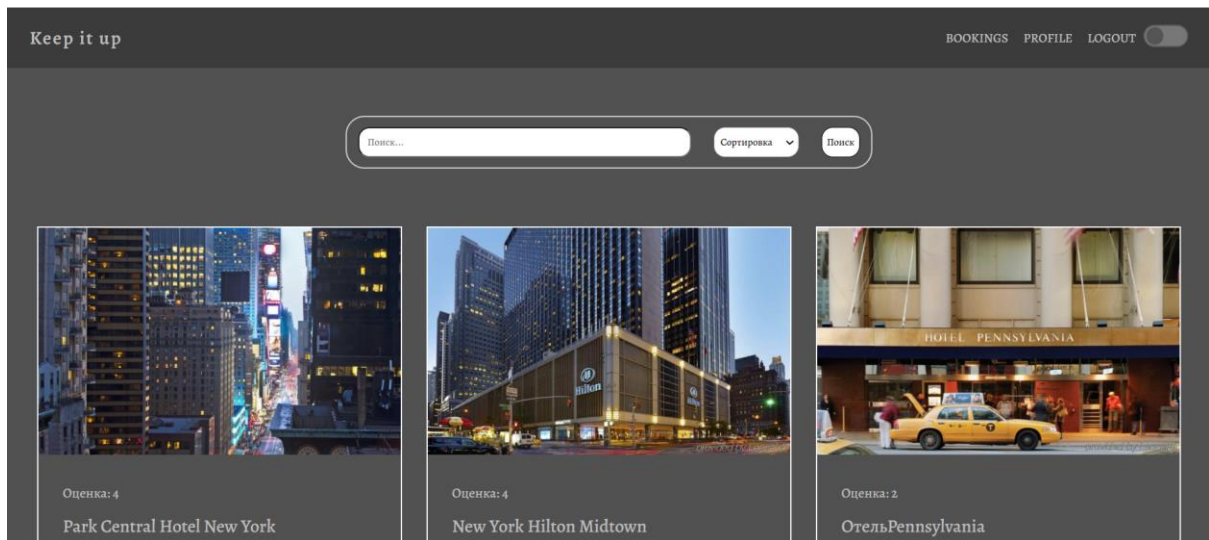
RESET

Форма восстановления пароля

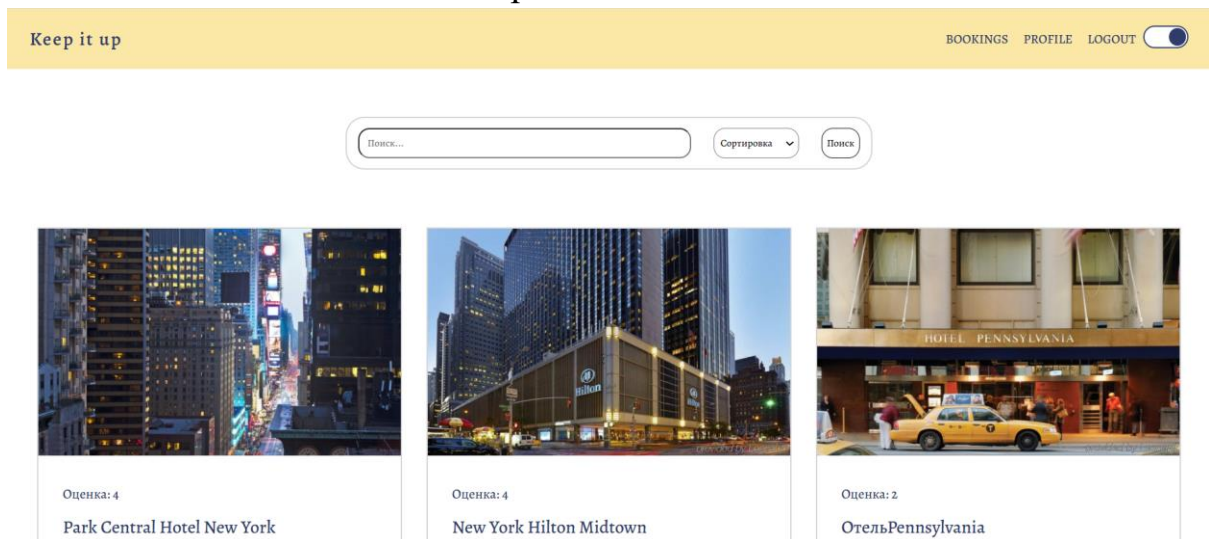
Форма регистрации

```
<div className="main-head">
  <form onSubmit={handleSubmit} className="form" sx={{color: 'text'}}>
    <h1>Login</h1>
    <label htmlFor="login"></label>
    <input sx={{
      backgroundColor: 'text',
      color: 'white',
      '&::placeholder': {color: 'white'},
      '&:focus': {backgroundColor: 'text'}
    }} onChange={handleChange} type="text" title="Username"
      placeholder="Username" name="username"/>
    <label htmlFor="password"></label>
    <input sx={{
      backgroundColor: 'text',
      color: 'white',
      '&::placeholder': {color: 'white'},
      '&:focus': {backgroundColor: 'text'}
    }} onChange={handleChange} type="password" title="Password" placeholder="Password" name="password"/>
    <button sx={{
      color: 'text',
      backgroundColor: 'nav',
      border: '1px solid border',
      '&:hover': {backgroundColor: 'text', color: 'nav'}
    }} type="submit">Login
    </button>
    <h2>Don't have an account?, <Link sx={{color: 'text'}} to="/register">Sign up</Link></h2>
    <h3><Link sx={{color: 'text'}} to="/forgot-password">Forgot password?</Link></h3>
  </form>
</div>
```

Код формы логина для примера



Главная страница с темной темой



Главная страница со светлой темой

```
<>
<div className="filter-area">
  <input className="search-input" type="text" placeholder="Поиск..." value={searchQuery}
    onChange={e => setSearchQuery(e.target.value)}/>
  <Sort value={selectedSort} onChange={sortHotels} defaultValue="Сортировка"
    options={[
      {value: 'title', name: 'По названию'},
      {value: 'address', name: 'По адресу'},
      {value: 'rate', name: 'По рейтингу'},
      {value: 'price', name: 'По цене'},
    ]}/>
  <button title="search button" onClick={fetchSort} className="search-button" type="button">Поиск
</button>
</div>
<HotelsList hotels={hotels}/>
</>
```

Код главной страницы

```
const theme = {
  initialColorModeName: 'light',
  colors: {
    text: '#303c6c',
    background: 'white',
    nav: '#fbe8a6',
    border: 'lightgray',
    modes: {
      dark: {
        text: '#bdbdbd',
        background: '#515151',
        nav: '#3b3b3b',
        border: 'white'
      }
    }
  }
}

export default theme
```

## Код тем

```
const ToggleThemeSwitchBox = () => {
  const [colorMode, setColorMode] = useColorMode()
  const [isChecked, setIsChecked] = useState()
  const theme = localStorage.getItem({ key: 'theme-ui-color-mode' })
  useEffect(() => {
    if (theme === 'light') {
      setIsChecked({ value: true })
    } else {
      setIsChecked({ value: false })
    }
  }, [theme])
  const toggleTheme = () => {
    if (colorMode === 'light') {
      setIsChecked({ value: false })
      setColorMode('dark')
    } else {
      setIsChecked({ value: true })
      setColorMode('light')
    }
  }
  return (
    <div className="toggle">
      <input checked={isChecked}
        onChange={toggleTheme}
        type="checkbox" id="toggle"/>
      <label htmlFor="toggle"></label>
    </div>
  )
};

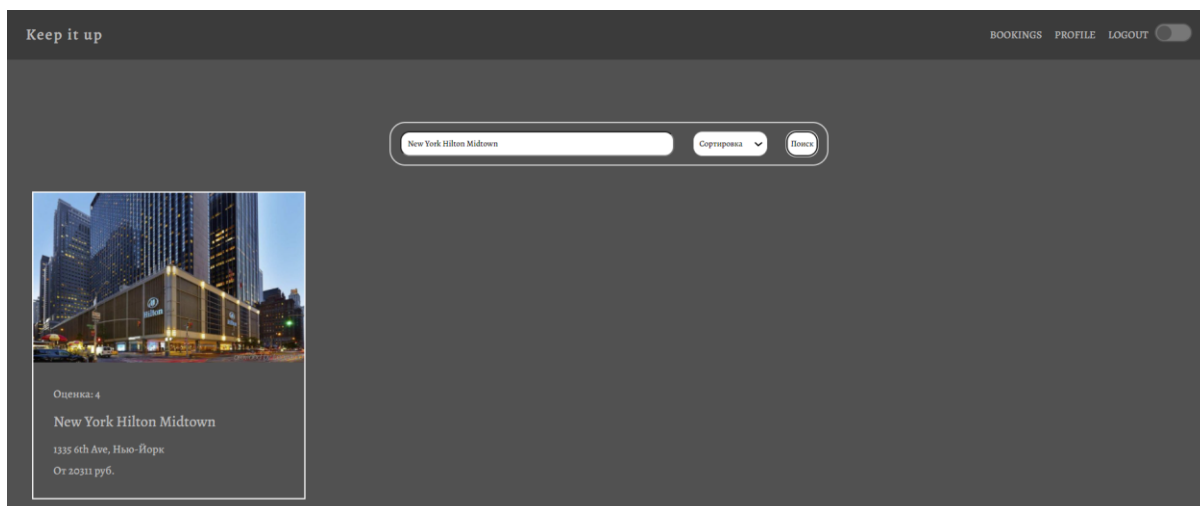
.toggle input[type='checkbox'] {
  display: none;
}

.toggle label {
  background-color: #777;
  border: 2px solid #555;
  border-radius: 50px;
  cursor: pointer;
  display: inline-block;
  position: relative;
  transition: all ease-in-out 0.3s;
  width: 60px;
  height: 30px;
}

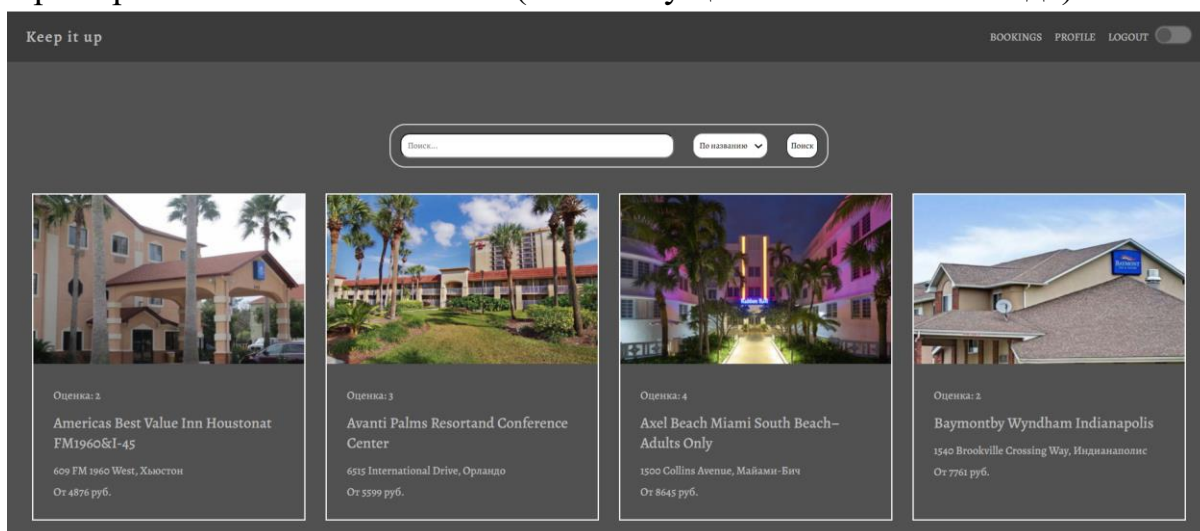
.toggle label::after {
  background-color: #555;
  border-radius: 50%;
  content: ' ';
  cursor: pointer;
  display: inline-block;
  position: absolute;
  left: 2px;
  top: 2px;
  transition: all ease-in-out 0.3s;
  width: 23px;
  height: 23px;
}

.toggle input[type='checkbox']:checked ~ label {
  background-color: white;
  border-color: #303c6c;
}
```

## Код переключателя тем + его css



## Пример поиска по имени отеля (поиск осуществляется на бэкенде)



## Пример сортировки по названию (сортировка осуществляется на бэкенде)




```
const Sort = ({options, defaultValue, value, onChange}) => {
  return (
    <select
      className="sort"
      value={value}
      onChange={event => onChange(event.target.value)}
    >
      <option disabled value="">{defaultValue}</option>
      {options.map((option) =>
        <option key={option.value} value={option.value}>{option.name}</option>
      )}
    </select>
  );
};

export default Sort;

<Sort value={selectedSort} onChange={sortHotels} defaultValue="Сортировка"
  options={[
    {value: 'title', name: 'По названию'},
    {value: 'address', name: 'По адресу'},
    {value: 'rate', name: 'По рейтингу'},
    {value: 'price', name: 'По цене'},
  ]}/>
```

Код компонента сортировки, как было показано в коде главной страница, он принимает значения, по которым сортируется массив объектов, а в качестве “дефолтной” опции принимает параметр “Сортировка”

Keep it up
BOOKINGS
PROFILE
LOGOUT



Оценка: 4  
New York Hilton Midtown  
1335 6th Ave, Нью-Йорк  
От 20 311 руб. руб.

Check in
mm/dd/yyyy

Check out
mm/dd/yyyy

Guests

Reserve

Joe  
dsadsad  
Joe  
dsadsad  
Joe  
Amazing hotel  
Joe

Страница бронирования отеля, на ней необходимо задать дату приезда и

отъезда, а также количество гостей + на ней же выводятся комментарии оставленные пользователями

```
async function FetchingHotel() {
  let response = await axiosInstance.get( url: `hotels/${params.id}`)
  setHotel(response.data)
}


async function FetchingReviews() {
  let response = await axiosInstance.get( url: `reviews/?hotel=${params.id}`)
  console.log(response.data)
  setReviews(response.data)
}

const handleSubmit = (e) => {
  e.preventDefault()
  axiosInstance.post( url: 'reserves/create', data: {
    check_in_date: checkInDate,
    check_out_date: checkOutDate,
    guests_count: parseInt(guestsCount),
    hotel: parseInt(params.id),
    guest: jwtDecode(localStorage.getItem( key: 'access_token')).user_id,
  })
  .then((res : AxiosResponse<any> ) => {
    history.push( path: "/profile")
  })
}
```

Пример axios запроса для получения информации об отеле и комментариях, а также создания брони

Keep it up

BOOKINGS PROFILE LOGOUT




Оценка: 4

New York Hilton Midtown

1335 6th Ave, Нью-Йорк

Check in Date : 2022-01-19

Check out Date : 2022-01-27




Оценка: 5

Waldorf Astoria

301 Park Avenue, Нью-Йорк

Check in Date : 2022-01-21

Check out Date : 2022-01-21




Оценка: 3

Skyline Hotel

725 10th Ave, Нью-Йорк

Check in Date : 2022-01-25

Check out Date : 2022-01-31



Оценка: 3

The Belvedere Hotel

319 W 48th St, Нью-Йорк

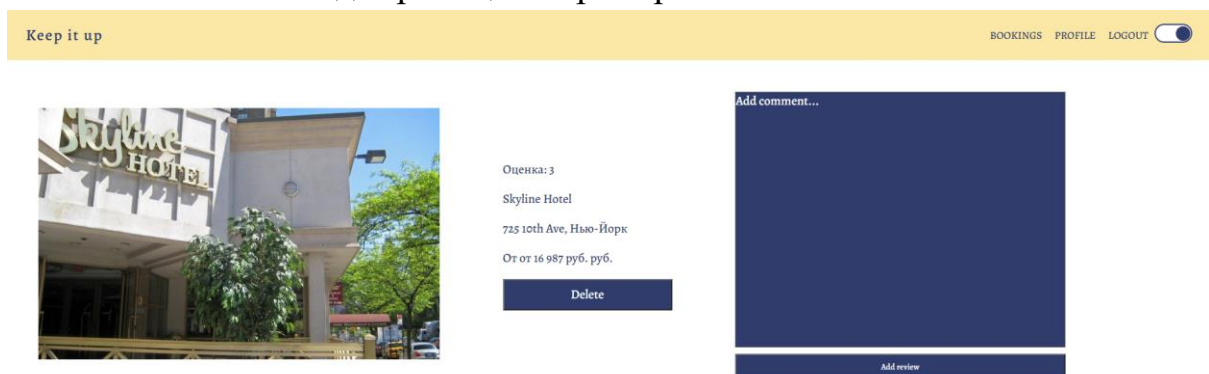
Check in Date : 2022-01-17

Check out Date : 2022-01-27

Страница забронированных отелей пользователя, доступная по клику на ссылку Bookings в навбаре

```
return (  
  <div className={classes.hotelsContainer}>  
    {reserves.map((reserve) =>  
      <div className={classes.hotelContainer} onClick={() => history.push(`/profile/reserve/${reserve.id}`)}>  
        <article className="hotel-cell" sx={{border: '2px solid', borderColor: 'border'}}>  
          <figure className={classes.hotelImage}><img src={reserve.hotel.img} alt="Hotel image"/></figure>  
          <div className={classes.hotelContent}>  
            <div className={classes.hotelRating} sx={{color: 'text'}}>Оценка: {reserve.hotel.rate}</div>  
            <div className={classes.hotelTitle} sx={{color: 'text'}}>{reserve.hotel.title}</div>  
            <div className={classes.hotelAddress} sx={{color: 'text'}}>{reserve.hotel.address}</div>  
            <div className={classes.hotelCheckDate} sx={{color: 'text'}}>Check in Date : {reserve.check_in_date} </div>  
            <div className={classes.hotelCheckDate} sx={{color: 'text'}}>Check out Date : {reserve.check_out_date}</div>  
          </div>  
        </article>  
      </div>  
    )}  
  </div>  
)
```

### Код страницы забронированных отелей



Также пользователь может удалить бронь или добавить комментарий к отелю

```

<div className="hotel-reserve">
  <article className="hotel">
    <figure className="-image"><img src={hotel.img} alt="Hotel image"/></figure>

    <div className="content">
      <div className="rating" sx={{color: 'text'}}>Оценка: {hotel.rate}</div>
      <div className="title" sx={{color: 'text'}}>{hotel.title}</div>
      <div className="address" sx={{color: 'text'}}>{hotel.address}</div>
      <div className="price" sx={{color: 'text'}}>От {hotel.price} py6.</div>
      <button sx={{
        backgroundColor: 'text',
        color: 'white',
        '&::placeholder': {color: 'white'},
        '&:focus': {backgroundColor: 'text'}
      }} className="reserve-button" onClick={handleDeleteSubmit}>Delete
    </button>
    </div>
    <form className="reserve-params" onSubmit={handleDeleteSubmit}>
      <textarea sx={{
        backgroundColor: 'text',
        color: 'white',
        '&::placeholder': {color: 'white'},
        '&:focus': {backgroundColor: 'text'}
      }} onChange={(e) => setComment(e.target.value)} className="review-add" name="comment" id="comment"
        cols="50" rows="15"
        placeholder="Add comment..."></textarea>
      <button sx={{
        backgroundColor: 'text',
        color: 'white',
        '&::placeholder': {color: 'white'},
        '&:focus': {backgroundColor: 'text'}
      }} onClick={handleAddReview} className="review-add-button">Add review
    </button>
    </form>
  </article>
</div>

```

## Код страницы по удалению и добавлению комментария



Страница профиля пользователя, где можно загрузить фото и посмотреть информацию по аккаунту



Choose File pwa\_vert...15 (1).jpg

Name: Alexandr

Email: a@a.ru

Date joined: 2021-12-22

```
<form className="profile_form">
  <div className="profile_image_container">
    <figure className="profile_image">
      <img src={image} alt="Profile image"/>
    </figure>
    <label htmlFor="file-upload" className="upload_profile_image">
    </label>
    <input id="file-upload" onChange={imageHandler} type="file"/>
  </div>
  <div className="user_info_container">
    <p className="user_info_tab">Name: {user.username}</p>
    <p className="user_info_tab">Email: {user.email}</p>
    <p className="user_info_tab">Date joined: {user.date_joined}</p>
  </div>
</form>
```

Код формы пользователя

```

const ProfileForm = () => {
  const [image, setImage] = useState({ initialState: "https://static.vecteezy.com/system/resources/praviews/002/318/271/non-2x/user-profile-icon-free-vector.jpg" })
  const [user, setUser] = useState({ initialState: {} })

  const imageHandler = (e) => {
    const reader = new FileReader()
    reader.onload = () => {
      if (reader.readyState === 2) {
        setImage(reader.result)
      }
    }
    reader.readAsDataURL(e.target.files[0])
  }

  useEffect(() => {
    FetchingUser()
  }, [dept, []])

  async function FetchingUser() {
    let response = await axiosInstance.get(`http://127.0.0.1:8000/users/?id=${jwtDecode(localStorage.getItem('access_token')).user_id}`)
    console.log(response.data)
    response.data[0].date_joined = response.data[0].date_joined.slice(0,10)
    setUser(response.data[0])
  }
}

```

Код обработки фото профиля

## Заключение

### Вывод по работе

Благодаря данной работе я научился нужным методам работы с клиентской частью, изучил react.js, а также узнал много нового о axios. Так же впервые пришлось самостоятельно реализовывать всю архитектуру приложения. С полным осознанием того, что нужно делать для работоспособности.

## СПИСОК ЛИТЕРАТУРЫ

1. Документация Django <https://docs.djangoproject.com/en/3.0/>.
2. Документация Django REST Framework <https://www.django-rest-framework.org>.
3. Документация React <https://ru.reactjs.org/docs/getting-started.html>.
4. Документация React Router Dom <https://v5.reactrouter.com/web/guides/quick-start>