

|  |   |
|--|---|
|  | <p><b>Министерство науки и высшего образования Российской Федерации</b><br/> <b>Федеральное государственное бюджетное образовательное учреждение</b><br/> <b>высшего образования</b><br/> <b>«Московский государственный технический университет имени Н.Э.</b><br/> <b>Баумана</b><br/> <b>(национальный исследовательский университет)» (МГТУ им. Н.Э.</b><br/> <b>Баумана)</b></p> |
|--|---|

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

## Отчет по лабораторной работе №8

Студент Бондаренко Иван Геннадьевич

*фамилия, имя, отчество*

Группа ИУ5-51Б

Студент

24.12.2021 Бондаренко И.Г.  
*подпись, дата фамилия, и.о.*

Преподаватель

24.12.2021 Гапанюк Ю.Е.  
*подпись, дата*

*фамилия, и.о.*

2021г.

## **План:**

1. Введение и установка проекта
2. Роутинг в react приложении
3. Терминология
  - i. JSX
  - ii. Component
  - iii. Element
  - iv. Props
  - v. State
4. Жизненный цикл react приложения
  - i. Монтирование
  - ii. Обновление
  - iii. Размонтирование
5. Функциональные компоненты и хуки
  - i. Функциональные компоненты
  - ii. Хуки
    - a. Хук состояния
    - b. Хук Эффекта
    - c. Пользовательские хуки
6. Практика
  - i. Пример работы с хуками эффекта и состояния на простом приложении
  - ii. Разработка с компонентным подходом. Пример работы с API, библиотеками и пользовательскими хуками
7. Полезные ссылки

## Текст программы:

api/types/HotelDataType.ts:

```
export type HotelDataType = {  
  pk: number;  
  name: string;  
  url: string;  
  imageSrc: string;  
  stars: number;  
  country_code: number;  
};
```

api/addHotel.ts:

```
import { HotelDataType } from 'api/types';  
  
export const addHotel = async (values: Partial<HotelDataType>) => {  
  const body = new FormData();  
  for (const [key, value] of Object.entries(values)) {  
    body.append(key, value.toString());  
  }  
  const data = await fetch('http://127.0.0.1:8000/hotel/', {
```

```

    method: 'POST',
    body: body,
  });
  if (data.ok) {
    return true;
  } else {
    return false;
  }
};

```

api/getHotelById.ts:

```

import { HotelDataType } from 'api/types';

export const getHotelById = async (id: number) => {
  const data = await fetch(`http://127.0.0.1:8000/hotel/${id}`, {
    headers: {
      'Content-Type': 'json/application',
    },
  });
  if (data.ok) {
    const parsedData: HotelDataType = await data.json();
    return parsedData;
  } else {
    return undefined;
  }
};

```

api/getHotels.ts:

```

import { HotelDataType } from 'api/types';

export const getHotels = async () => {
  const data = await fetch('http://127.0.0.1:8000/hotel', {
    headers: {
      'Content-Type': 'json/application',
    },
  });

```

```

    },
  });
  if (data.ok) {
    const parsedData: HotelDataType[] = await data.json();
    return parsedData;
  } else {
    return undefined;
  }
};

```

components/Button/Button.style.tsx:

```

import { Colors } from 'constants/colors';

import styled from 'styled-components';

import { ButtonProps } from './Button.types';

export const Button = styled.button<ButtonProps>`
  background-color: ${({ backgroundColor }) =>
    backgroundColor || 'transparent'};

  padding: ${({ round }) => (round ? '5px 5px' : '10px 16px')};
  border: 1px solid ${({ backgroundColor }) => backgroundColor || 'transparent'};
  border-radius: 16px;
  cursor: pointer;

  color: ${Colors.TEXT_MAIN_COLOR};
  font-family: Roboto;
  font-size: 14px;
  font-weight: 400;

  &:hover {
    border-color: ${({ backgroundColor, backgroundHoverColor }) =>
      backgroundHoverColor
        ? backgroundHoverColor
        : backgroundColor || 'transparent'};
    background-color: ${({ backgroundColor, backgroundHoverColor }) =>

```

```

        backgroundHoverColor
        ? backgroundHoverColor
        : backgroundColor || 'transparent'});
    }
};

```

components/Button/Button.types.tsx:

```

export type ButtonProps = {
  backgroundColor?: string;
  backgroundHoverColor?: string;
  round?: boolean;
};

```

components/Card/Card.styles.tsx:

```

import { Colors } from 'constants/colors';

import styled from 'styled-components';

export const CardWrapper = styled.div`
  display: flex;
  flex: 1 0 21%;
  position: relative;
  height: 250px;
  border: 1px solid ${Colors.BORDER_MAIN};
  border-radius: 16px;
  margin: 5px;
  padding: 20px;
  flex-direction: column;
  justify-content: space-between;
  align-items: center;
  background-color: ${Colors.CARD_MAIN};
`;

```

components/Card/Card.tsx:

```
import { Colors } from 'constants/colors';

import React, { useCallback } from 'react';
import { removeHotelById } from 'api';
import { ReactComponent as IcClose } from 'assets/icons/close.svg';

import { Button } from 'components/Button';
import { Image } from 'components/Image';
import { StyledLink } from 'components/StyledLink/StyledLink.style';
import { Text } from 'components/Text';

import { CardWrapper } from './Card.styles';
import { CardProps } from './Card.types';
import { CardNavbar } from './CardNavbar';

export const Card = ({ data, onRemove }: CardProps): JSX.Element => {
  const handleRemoveCard = useCallback(async () => {
    await removeHotelById(data.pk);
    if (onRemove) {
      onRemove();
    }
  }, []);

  return (
    <CardWrapper>
      <CardNavbar>
        <Button round onClick={handleRemoveCard}>
          <IcClose width={16} height={16} />
        </Button>
      </CardNavbar>
      <a href={data.url}>
        <Image src={data.imageSrc}></Image>
      </a>
      <Text bold>{data.name}</Text>
      <Text>Количество звезд: {data.stars}</Text>
      <StyledLink to={` /hotel/${data.pk}`}>
        <Button
          backgroundColor={Colors.MAIN}

```



```

        backgroundHoverColor={Colors.MAIN_HOVERED}
      >
        Открыть
      </Button>
    </StyledLink>
  </CardWrapper>
);
};

```

components/Card/Card.types.tsx:

```

import { HotelDataType } from 'api/types';

export type CardProps = {
  data: HotelDataType;
  onRemove?: () => void;
};

```

components/Card/CardNavbar.styles.tsx:

```

import styled from 'styled-components';

export const CardNavbar = styled.div`
  position: absolute;
  top: 5px;
  right: 5px;
`;

```

components/Image/Image.styles.tsx:

```

import styled from 'styled-components';

export const Image = styled.img`
  width: 140px;
  height: 140px;
  border-radius: 16px;

```

```
`;  
`;
```

components/Navbar/Navbar.styles.tsx:

```
import { Colors } from 'constants/colors';  
  
import styled from 'styled-components';  
  
export const NavbarWrapper = styled.div`  
  display: flex;  
  flex-direction: row;  
  justify-content: space-around;  
  align-items: center;  
  width: 100%;  
  border: 1px solid ${Colors.BORDER_MAIN};  
  height: 50px;  
  background-color: ${Colors.BODY_MAIN};  
`;  
`;
```

components/Navbar/Navbar.tsx:

```
import React from 'react';  
  
import { Button } from 'components/Button';  
import { StyledLink } from 'components/StyledLink/StyledLink.style';  
  
import { NavbarWrapper } from './Navbar.styles';  
  
export const Navbar = () => {  
  return (  
    <NavbarWrapper>  
      <StyledLink to="/">  
        <Button>Home</Button>  
      </StyledLink>  
      <StyledLink to="/hotel/add">  
        <Button>Add hotel</Button>  
    </NavbarWrapper>  
  );  
};
```

```
    </StyledLink>
  </NavbarWrapper>
);
};
```

components/StyledForm/StyledForm.styles.tsx:

```
import { Colors } from 'constants/colors';

import styled from 'styled-components';

export const StyledForm = styled.form`
  border-radius: 16px;
  width: 50%;
  margin: auto;
  display: flex;
  align-items: center;
  flex-direction: column;
  padding: 20px;
  border: 1px solid ${Colors.BORDER_MAIN};
  background-color: ${Colors.BODY_MAIN};
  justify-content: space-between;
  height: 250px;
`;
```

components/StyledLink/StyledLink.styles.tsx:

```
import { Link } from 'react-router-dom';
import styled from 'styled-components';

export const StyledLink = styled(Link)`
  text-decoration: none;

  &:focus,
  &:hover,
  &:visited,
```

```
&:link,  
&:active {  
  text-decoration: none;  
}  
`;  
`;
```

components/Text/Text.styles.tsx:

```
import { Colors } from 'constants/colors';  
  
import styled from 'styled-components';  
  
import { TextProps } from './Text.types';  
  
export const Text = styled.p<TextProps>`  
  color: ${Colors.TEXT_MAIN_COLOR};  
  font-family: Roboto;  
  font-size: 14px;  
  font-weight: ${({ bold }) => (bold ? '700' : '400')};  
  
  margin: 0;  
`;  
`;
```

components/Text/Text.types.tsx:

```
export type TextProps = {  
  bold?: boolean;  
};
```

constants/colors.ts:

```
export enum Colors {  
  MAIN = '#24a0ed',  
  MAIN_HOVERED = '#1597DD',  
}
```

```
TEXT_MAIN_COLOR = 'white',

BODY_MAIN = '#181a1b',
CARD_MAIN = '#181a1b',
BORDER_MAIN = '#C8C8C8',

ALERT = '#cc0a24',
ALERT_HOVERED = '#ad0219',
}
```

pages/PageAddHotel/PageAddHotel.styles.tsx:

```
import { Colors } from 'constants/colors';

import styled from 'styled-components';

export const InputWrapper = styled.div`
  display: flex;
  width: 100%;
  color: ${Colors.TEXT_MAIN_COLOR};
  font-family: Roboto;
  font-size: 14px;
  font-weight: 400;
  justify-content: space-between;

  & > input,
  select {
    width: 50%;
  }
`;
```

pages/PageAddHotel/PageAddHotel.tsx:

```
import React, { useCallback, useState } from 'react';
import { Field, Form } from 'react-final-form';
import { addHotel } from 'api';
```

```

import { HotelDataType } from 'api/types';

import { Button } from 'components/Button';
import { Navbar } from 'components/Navbar/Navbar';
import { StyledForm } from 'components/StyledForm/StyledForm.styles';

import { InputWrapper } from './PageAddHotel.styles';

export const PageAddHotel = () => {
  const [loading, setLoading] = useState(false);

  const onSubmit = useCallback((values: Record<string, string>) => {
    setLoading(true);
    const newValues: Partial<HotelDataType> = {
      stars: parseInt(values['stars']),
      country_code: parseInt(values['country_code']),
      ...values,
    };
    addHotel(newValues).then(() => {
      setLoading(false);
    });
  }, []);

  return (
    <>
      <Navbar />
      <Form
        onSubmit={onSubmit}
        render={({ submitting, handleSubmit }) => (
          <StyledForm onSubmit={handleSubmit}>
            <InputWrapper>
              <label>Название</label>
              <Field
                name="name"
                type="text"
                component="input"
                placeholder="Name"
              />
            </InputWrapper>
          </StyledForm>
        )}
      />
    </>
  );
};

```

```
</InputWrapper>
<InputWrapper>
  <label>Ссылка на отель</label>
  <Field
    name="url"
    type="text"
    component="input"
    placeholder="Ссылка"
  />
</InputWrapper>
<InputWrapper>
  <label>Ссылка на изображение отеля</label>
  <Field
    name="imageSrc"
    type="text"
    component="input"
    placeholder="Ссылка"
  />
</InputWrapper>
<InputWrapper>
  <label>Количество звезд</label>
  <Field name="stars" component="select" defaultValue="default">
    <option value="default" disabled>
      Количество звезд
    </option>
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
  </Field>
</InputWrapper>
<InputWrapper>
  <label>Id страны</label>
  <Field
    name="country_code"
    type="text"
    component="input"
```

```

        placeholder="Home"
      />
    </InputWrapper>
    <Button type="submit" disabled={submitting || loading}>
      Добавить
    </Button>
  </StyledForm>
)}
/>
</>
);
};

```

pages/PageHotel/PageHotel.tsx:

```

import { Colors } from 'constants/colors';

import React, { useEffect, useState } from 'react';
import { useParams } from 'react-router';
import { getHotelById } from 'api';
import { HotelDataType } from 'api/types';

import { Button } from 'components/Button';
import { Card } from 'components/Card/Card';
import { Navbar } from 'components/Navbar/Navbar';
import { StyledLink } from 'components/StyledLink/StyledLink.style';

export const PageHotel = (): JSX.Element => {
  const [hotel, setHotel] = useState<HotelDataType>();
  const { id } = useParams();

  useEffect(() => {
    getHotelById(parseInt(id || '0')).then((data) => {
      setHotel(data);
    });
  }, [id]);

```



```

    if (hotel) {
      return (
        <>
          <Navbar />
          <Card data={hotel} />
        </>
      );
    }
    return <></>;
  };
};

```

pages/PageMain/PageMain.styles.tsx:

```

import styled from 'styled-components';

export const PageMainWrapper = styled.div`
  display: flex;
  flex-wrap: wrap;
`;

```

pages/PageMain/PageMain.tsx:

```

import React, { ChangeEvent, useCallback, useEffect, useState } from 'react';
import { getHotels } from 'api';
import { HotelDataType } from 'api/types';

import { Card } from 'components/Card/Card';
import { Navbar } from 'components/Navbar/Navbar';

import { PageMainWrapper } from './PageMain.styles';

export const PageMain = (): JSX.Element => {
  const [hotels, setHotels] = useState<HotelDataType[]>();

  useEffect(() => {
    getHotels().then((data) => {

```

```

        setHotels(data);
    });
}, []);

const onRemoveClick = useCallback((id: number) => {
    setHotels((prev) => prev?.filter((value) => value.pk !== id));
}, []);

const onFilter = useCallback((event: ChangeEvent<HTMLInputElement>) => {
    console.log(event.target.value);
    getHotels().then((data) => {
        setHotels(
            data?.filter((value) => value.name.includes(event.target.value)),
        );
    });
}, []);

return (
    <PageMainWrapper>
        <Navbar />
        <div
            style={{ width: '100%', display: 'flex', justifyContent: 'center ' }}
        >
            <input name="name" placeholder="Найти" onChange={onFilter}></input>
        </div>
        {hotels?.map((hotel) => (
            <Card
                key={hotel.pk}
                data={hotel}
                onRemove={() => onRemoveClick(hotel.pk)}
            />
        ))}
    </PageMainWrapper>
);
};

```

## App.tsx

```
import React from 'react';
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import { PageAddHotel } from 'pages/PageAddHotel';
import { PageHotel } from 'pages/PageHotel';
import { PageMain } from 'pages/PageMain';

import './App.css';

function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<PageMain />} />
        <Route path="/hotel/add" element={<PageAddHotel />} />
        <Route path="/hotel/:id" element={<PageHotel />} />
      </Routes>
    </BrowserRouter>
  );
}

export default App;
```

## index.tsx

```
import React from 'react';
import ReactDOM from 'react-dom';

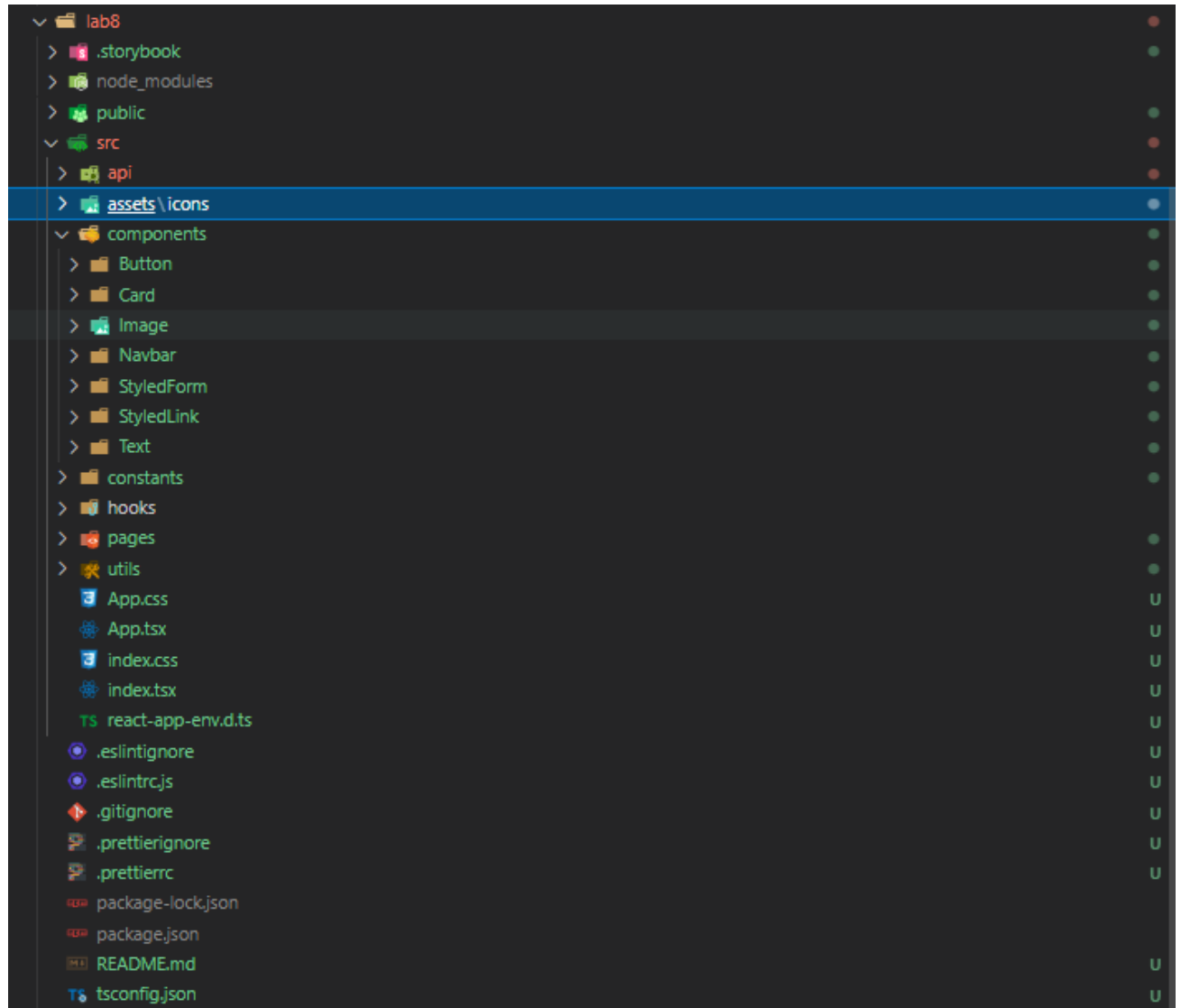
import App from './App';

import './index.css';

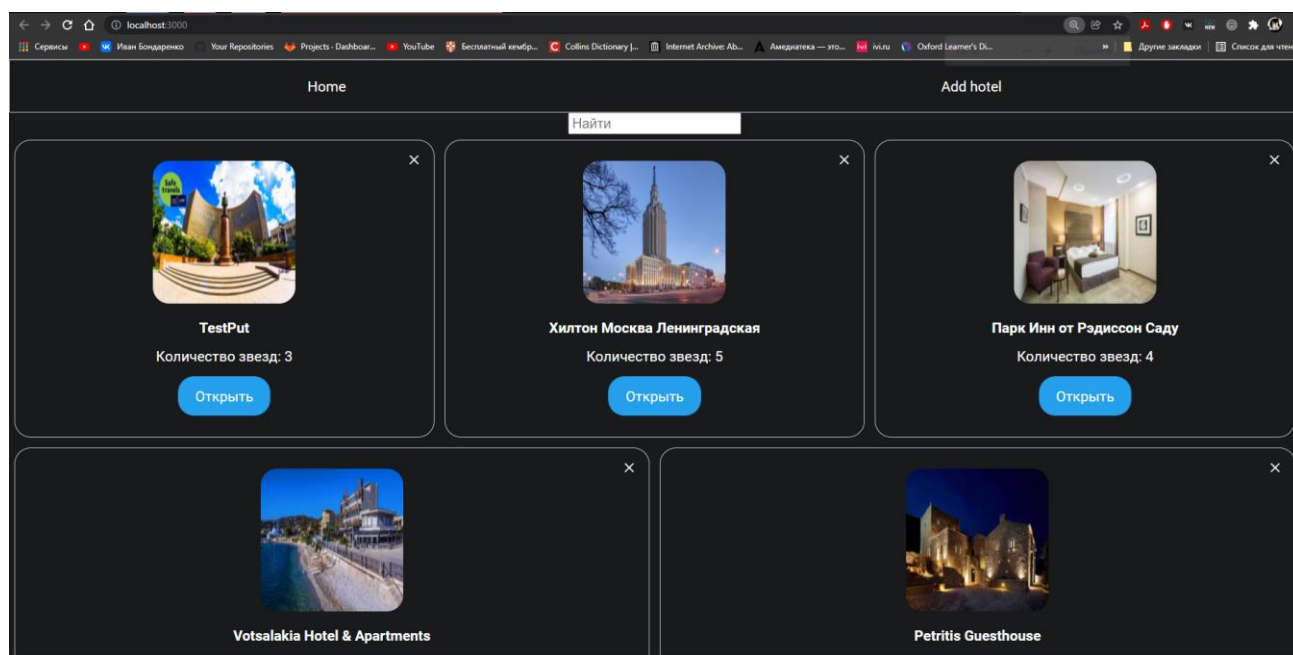
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
  , document.getElementById('root')
```

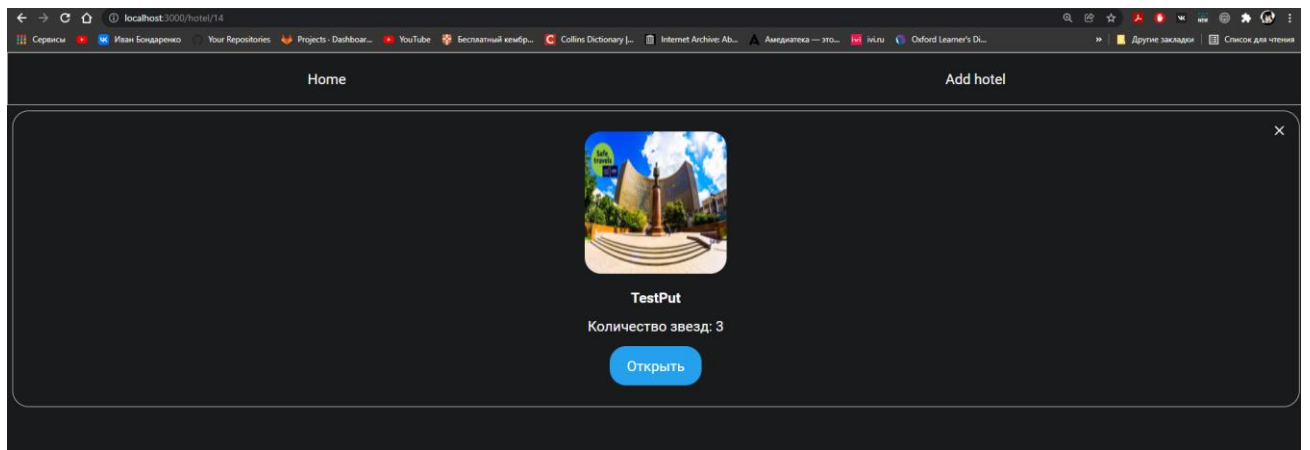
```
</React.StrictMode>,  
document.getElementById('root'),  
);
```

Структура проекта:

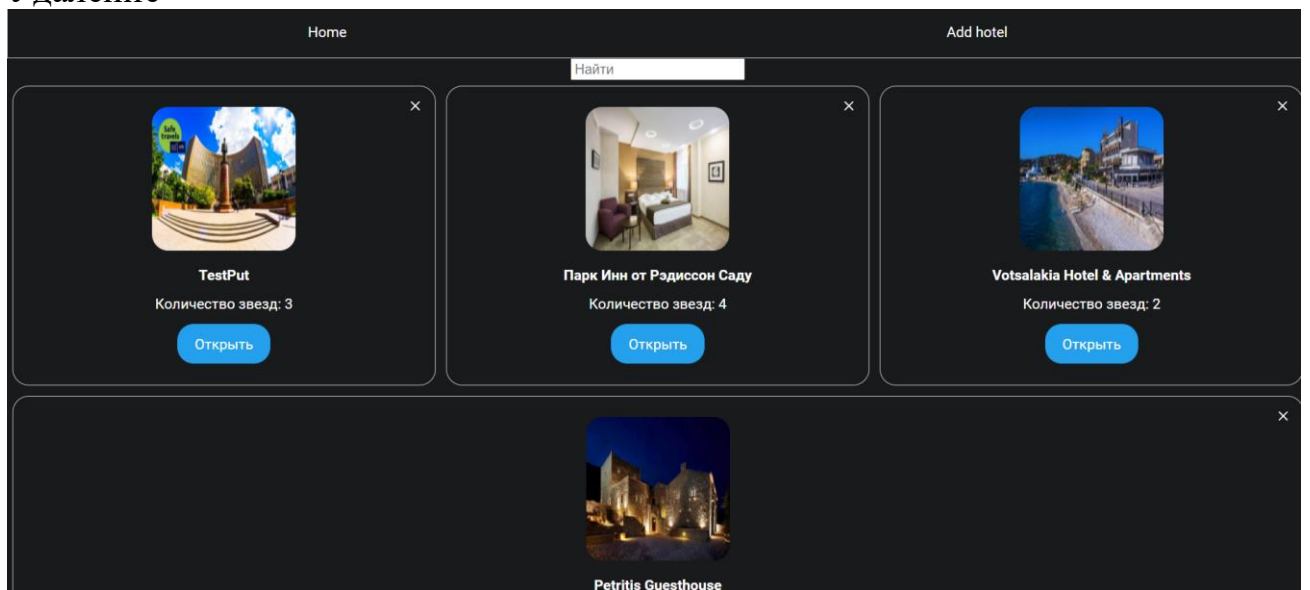


## Экранные формы с примерами выполнения программы:





## Удаление



localhost:3000/hotel/add

СервисыИван СонгаренкоYour RepositoriesProjects - Dashboar...YouTubeБесплатный кембр...Collins DictionaryInternet Archive: Ab...Амедиатека — это...huluOxford Learner's DI...Другие закладкиСписок для чтения

HomeAdd hotel

Название

Name

Ссылка на отель

Ссылка

Ссылка на изображение отеля

Ссылка

Количество звезд

Количество звезд

Id страны

Номер

Добавить

HomeAdd hotel

Название

Проверка

Ссылка на отель

Добавления

Ссылка на изображение отеля

Нового

Количество звезд

3

Id страны

4

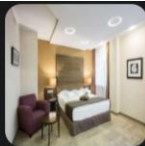
Добавить



TestPut

Количество звезд: 3

Открыть



Парк Инн от Рэдиссон Саду

Количество звезд: 4

Открыть



Votsalakia Hotel & Apartments

Количество звезд: 2

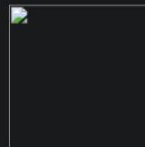
Открыть



Petritis Guesthouse

Количество звезд: 4

Открыть



Проверка

Количество звезд: 3

Открыть