

**Московский государственный технический университет
им. Н. Э. Баумана**

Факультет «Информатика и системы управления»

**Отчёт по лабораторной работе №8
по курсу «Разработка интернет-приложений»**

Выполнил:

**студент группы ИУ5-51Б
Абдрашитова А.Н.**

Проверил:

**преподаватель каф. ИУ5
Гапанюк Ю. Е.**

Подпись и дата: 28.12.2021

**Подпись и дата:
28.12.2021**

Москва, 2021 г.

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/ProductType.ts:

```
export type ProductType = {  
  pk: number;   name: string;  
  price: number; quantity:  
  number;   manufacturer:  
  string;   weight: number;  
  expiration_date: Date;  
  description: string;  
  imgSrc: string; };
```

api/getProductById.ts:

```
import { ProductType } from 'api/types';  
export const getProductById = async (id: number) => {  const data  
= await fetch(`http://127.0.0.1:8000/products/${id}/`, {  
  headers: {  
    'Content-Type': 'json/application',  
  },  });  if (data.ok) {    const parsedData:  
ProductType = await data.json();    return  
parsedData;  
  } else {  
    return undefined;  
  }  
};
```

api/getProducts.ts:

```
import { ProductType } from 'api/types';  
  
export const getProducts = async () => {  const data = await  
fetch('http://127.0.0.1:8000/products/', {    headers: {  
      'Content-Type': 'json/application',  
    },  });  if (data.ok) {    const parsedData:  
ProductType[] = 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 27%; position: relative; height: 500px; 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};
  box-shadow: 0px -1px 12px 14px rgba(34, 60, 80, 0.2);
`;

```

components/Card/Card.tsx:

```
import { Colors } from 'constants/colors';
import React, { useCallback } from 'react'; import {
removeProductById } 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, hasButton }: CardProps): JSX.Element =>
{
  const handleRemoveCard = useCallback(async () => {
    await
    removeProductById(data.pk);
    if (onRemove) {
      onRemove();
    }
  }, []);

  return (
    <CardWrapper>
      <CardNavbar>
        <Button round onClick={handleRemoveCard}>
          <IcClose width={16} height={16} />
        </Button>
      </CardNavbar>
      <Image src={data.imgSrc}></Image>
      <Text bold>{data.name}</Text>
      <Text>Цена: {data.price} руб.</Text>
      <Text>Количество на складе: {data.quantity}</Text>

      <Text>Производитель: {data.manufacturer}</Text>
      <Text>Вес: {data.weight} г.</Text>
      <Text>Срок годности: {data.expiration_date}</Text>
      <Text>Описание: {data.description}</Text>
      {hasButton && (
        <StyledLink to={` /products/${data.pk}`}>
          <Button
            backgroundColor={Colors.MAIN}
            backgroundHoverColor={Colors.MAIN_HOVERED}>
            Открыть
          </Button>
        </StyledLink>
      )}
    </CardWrapper>
  );
};
```

```
};
```

components/Card/Card.types.tsx:

```
import { ProductType } from 'api/types';
export type CardProps =
{
  data: ProductType;
  onRemove?: () => void;
  hasButton?: boolean;
};
```

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: 250px;
height: 250px; 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-between; align-
items: center; width: 100%; margin-
bottom: 20px; border: 1px solid
${Colors.BORDER_MAIN}; height: 50px;
background-color: ${Colors.BODY_MAIN};
  & > a > button { font-
size: 20px;
  }
  & > img {
width: 46px;
height: 46px;
  }
`;

```

components/Navbar/Navbar.tsx:

```

import React from 'react'; import React, { ChangeEvent,
useCallback } from 'react'; import { getProducts } from
'api'; import { ProductType } from 'api/types';

import { Button } from 'components/Button'; import
{ Image } from 'components/Image';
import { StyledLink } from 'components/StyledLink/StyledLink.style';
import { NavbarWrapper } from './Navbar.styles';

type NavbarProps = {
  productCb?: React.Dispatch<React.SetStateAction<ProductType[] | undefined>>;
};
export const Navbar = ({ productCb }: NavbarProps) => {
  const onFilter = useCallback((event: ChangeEvent<HTMLInputElement>) => {
if (productCb) { getProducts().then((data) => {
productCb(data?.filter((value) =>
value.name.includes(event.target.value)));
  });
  });
}

```

```

    }  },
  []);
return (
  <NavbarWrapper>
    <StyledLink to="/">
      <Button>Главная</Button>
    </StyledLink>
    <Image
src="http://abali.ru/wpcontent/uploads/2013/03/Gerb_MGTU_imeni_Baumana.png"></Image>
    {productCb && (
      <div style={{ width: '100%', display: 'flex', justifyContent: 'flex-end',
marginRight: '10px' }}>
        <input name="name" placeholder="Найти" onChange={onFilter}></input>
      </div>
    )}
  </NavbarWrapper>
);
};

```

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: ${({ bold }) => (bold ? '24px' : '16px')};
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 = '#c2d44e',  
  MAIN_HOVERED = 'grey',  
  
  TEXT_MAIN_COLOR = 'white',  
  
  BODY_MAIN = '#393e45',  
  CARD_MAIN = '#393e45',  
  BORDER_MAIN = '#393e45',  
  
  ALERT = '#cc0a24',   ALERT_HOVERED =  
  '#ad0219',  
}
```

pages/PageProduct/PageProduct.tsx:

```
import React, { useEffect, useState } from  
'react'; import { useParams } from 'react-router';  
import { getProductById } from 'api'; import {  
  ProductType } from 'api/types';  
  import { Card } from 'components/Card/Card';  
import { Navbar } from  
'components/Navbar/Navbar';  
  export const PageProduct = (): JSX.Element => {  
const [product, setproduct] = useState<ProductType>();  
const { id } = useParams();  
  useEffect(() => {   getProductById(parseInt(id ||  
'0')).then((data) => {   setproduct(data);  
    });  
  }, [id]);  
  if (product)  
{
```

```

    return (
      <>
        <Navbar />
        <Card data={product} />
      </>
    );
  }
  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, { useCallback, useEffect, useState } from 'react';
import { getProducts } from 'api'; import { ProductType } 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 [products, setProducts] = useState<ProductType[]>();
  useEffect(() => {
    getProducts().then((data) => {
      setProducts(data);
    });
  }, []);

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

  return (
    <PageMainWrapper>
      <Navbar productCb={setProducts} />

      {products?.map((product) => (
        <Card key={product.pk} data={product} hasButton onRemove={() =>
onRemoveClick(product.pk)} />
      ))}
    </PageMainWrapper>
  );
}

```

```
    </PageMainWrapper>
  );
};
```

App.tsx

```
import React from 'react'; import { BrowserRouter, Route, Routes
} from 'react-router-dom'; import { PageMain } from
'pages/PageMain'; import { PageProduct } from
'pages/PageProduct';
import
'./App.css';
function App()
{  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<PageMain />} />
        <Route path="/products/:id" element={<PageProduct />} />
      </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'),
);
```

Экранные формы с примерами выполнения программы:

Главная


Найти



Ромашка аптечная

Цена: 100 руб.
Количество на складе: 6
Производитель: ИП Бондаренкович И.Г.
Вес: 50 г.
Срок годности: 2025-05-25T00:00:00Z
Описание: Ромашка - спасение от всего


Открыть



Капсулы Sustal

Цена: 500 руб.
Количество на складе: 8
Производитель: ИП Путинович А.Б.
Вес: 250 г.
Срок годности: 2025-05-25T00:00:00Z
Описание: Сустанль – это растительный комплекс в виде капсул, находящихся в специальной среде-активаторе (также на основе природного сырья).


Открыть



Календула

Цена: 100 руб.
Количество на складе: 18
Производитель: ООО Алтай-сила
Вес: 100 г.
Срок годности: 2125-05-25T00:00:00Z
Описание: Календула — лекарственное растение с ярко-оранжевыми душистыми цветами.

Открыть



Капсулы Sustal

Цена: 500 руб.
Количество на складе: 8
Производитель: ИП Путинович А.Б.
Вес: 250 г.
Срок годности: 2025-05-25T00:00:00Z
Описание: Сустанль – это растительный комплекс в виде капсул, находящихся в специальной среде-активаторе (также на основе природного сырья).


Открыть



Календула

Цена: 100 руб.
Количество на складе: 18
Производитель: ООО Алтай-сила
Вес: 100 г.
Срок годности: 2125-05-25T00:00:00Z
Описание: Календула — лекарственное растение с ярко-оранжевыми душистыми цветами.

Открыть



Капсулы Sustal

Цена: 500 руб.
Количество на складе: 8
Производитель: ИП Путинович А.Б.
Вес: 250 г.
Срок годности: 2025-05-25T00:00:00Z
Описание: Сустанль – это растительный комплекс в виде капсул, находящихся в специальной среде-активаторе (также на основе природного сырья).

По нажатию на крестик в верхнем правом углу товар удаляется.