федеральное государственное автономное образовательное учреждение высшего образования

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
(ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ)
(Факультет информационных технологий)

(Институт Принтмедиа и информационных технологий)
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 4

Дисциплина: WebDev____
____

Тема: "Итоговый проект"____
____
____

Выполнил(а): студент(ка) группы **221-3711**

Малыгин Александр Сергеевич
(Фамилия И.О.)

Дата, подпись __________________ ____________
(Дата) (Подпись)

Проверил: _______________________________ ____________
(Фамилия И.О., степень, звание) **(Оценка)**

Дата, подпись __________________ ____________
(Дата) (Подпись)

Замечания:____
____

Москва 2024

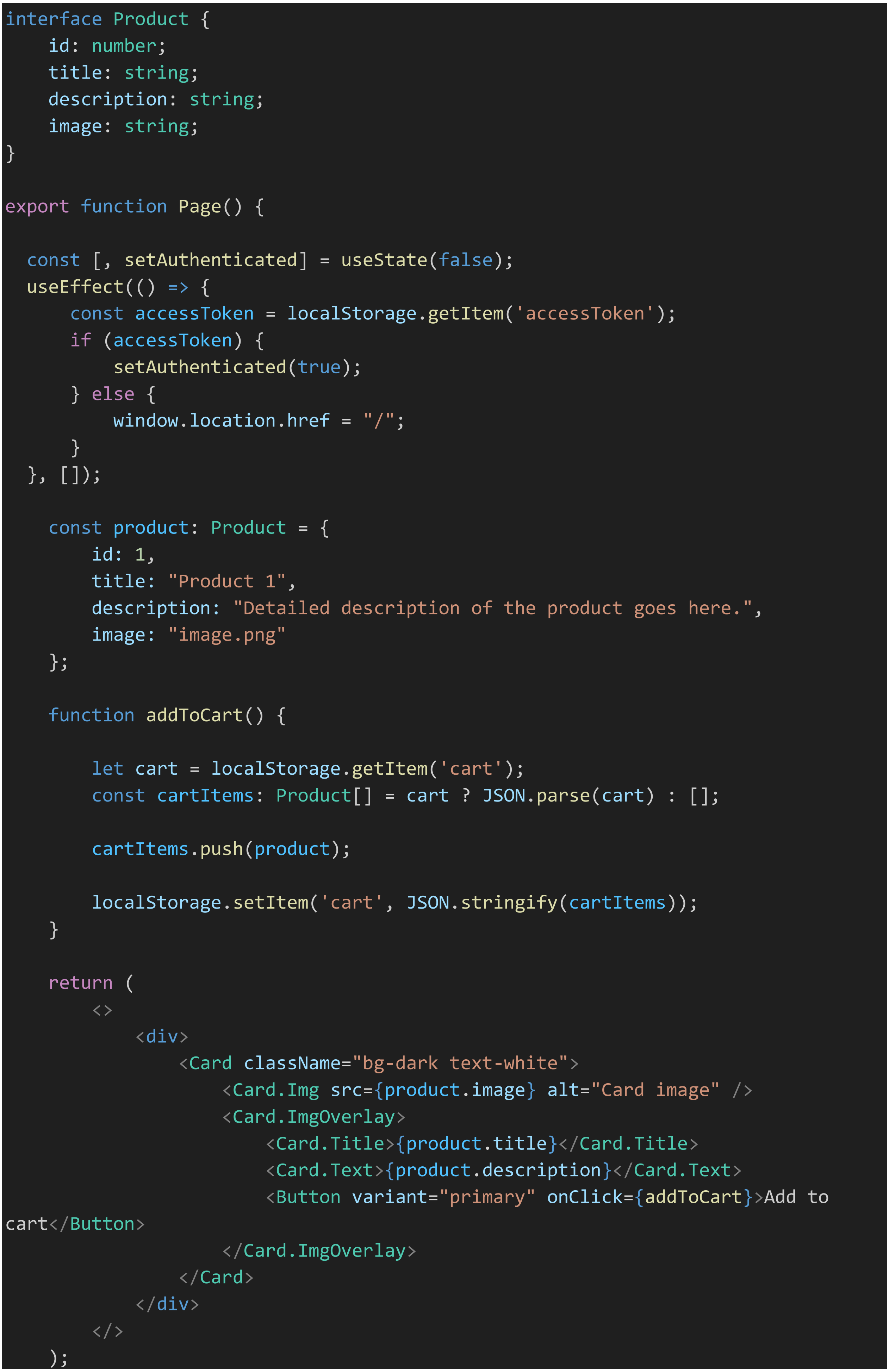
Код

App.tsx

```
import 'bootstrap/dist/css/bootstrap.min.css';
import NavbarComponent from './Navbar';
import Footer from './Footer';
import MouseTracker from './MouseTracker';
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import { MainPage } from './MainpageOBJ/MainPage';
import { LoginPage } from './AuthOBJ/LoginPage';
import { Page } from './Page';
import { Task } from './DailyPage/DailyTaskPage';
import { Cart } from './Cart';
import { PageTwo } from './Page2';


const App: React.FC = () => {
  return (
    <BrowserRouter>
    <>
      <NavbarComponent setSateButton={function (): void {
        throw new Error('Function not implemented.');
      } } />
      <Footer />
      <MouseTracker />
      <Routes>
        <Route path="" element={<LoginPage/>}/>
        <Route path="/home" element={<MainPage/>}/>
        <Route path="/page" element={<Page/>}/>
        <Route path="/page2" element={<PageTwo/>}/>
        <Route path="/Task" element={<Task/>}/>
        <Route path="/Cart" element={<Cart/>}/>
      </Routes>
    </>
    </BrowserRouter>
  );
};

export default App;
```

Page.tsx

```
import { useEffect, useState } from 'react';
import { Button, Card } from 'react-bootstrap';
```



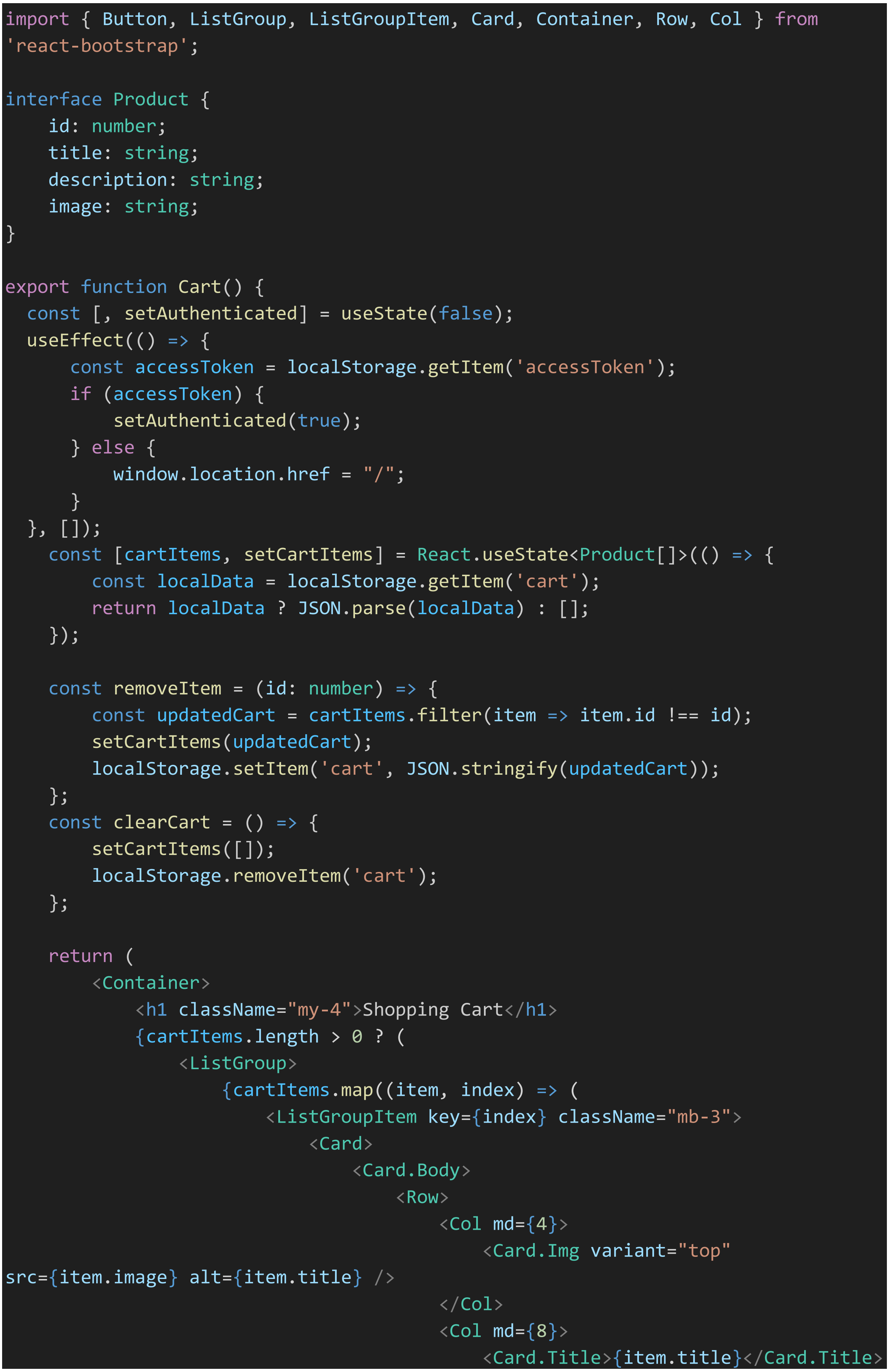
```
interface Product {
    id: number;
    title: string;
    description: string;
    image: string;
}

export function Page() {

  const [, setAuthenticated] = useState(false);
  useEffect(() => {
      const accessToken = localStorage.getItem('accessToken');
      if (accessToken) {
          setAuthenticated(true);
      } else {
          window.location.href = "/";
      }
  }, []);

    const product: Product = {
        id: 1,
        title: "Product 1",
        description: "Detailed description of the product goes here.",
        image: "image.png"
    };

    function addToCart() {

        let cart = localStorage.getItem('cart');
        const cartItems: Product[] = cart ? JSON.parse(cart) : [];

        cartItems.push(product);

        localStorage.setItem('cart', JSON.stringify(cartItems));
    }

    return (
        <>
            <div>
                <Card className="bg-dark text-white">
                    <Card.Img src={product.image} alt="Card image" />
                    <Card.ImgOverlay>
                        <Card.Title>{product.title}</Card.Title>
                        <Card.Text>{product.description}</Card.Text>
                        <Button variant="primary" onClick={addToCart}>Add to
cart</Button>
                    </Card.ImgOverlay>
                </Card>
            </div>
        </>
    );
}
```

LoginPage.tsx с токеном авторизации и axios вместо fetch

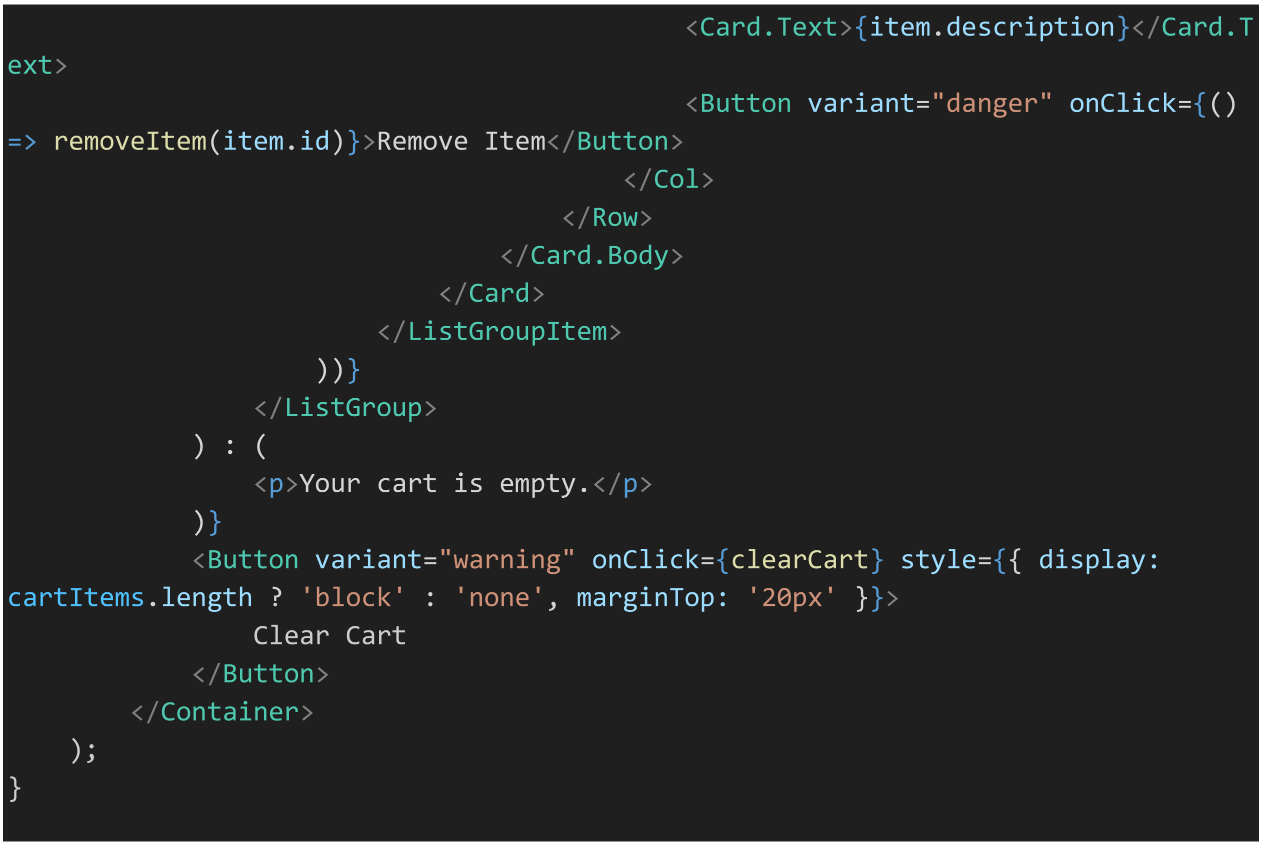
```
import { Button, Col, Container, Form, Row } from "react-bootstrap";
import axios from "axios";
import { useState } from "react";

export function LoginPage(){
    const [, setLoggedIn] = useState(false);
    const [username, setUsername] = useState('');
    const [password, setPassword] = useState('');
    const [, setError] = useState('');
    const [, setAccessToken] = useState('');

    const handleUsernameChange = (event: React.ChangeEvent<HTMLInputElement>) =>
{
        setUsername(event.target.value);
    };

    const handlePasswordChange = (event: React.ChangeEvent<HTMLInputElement>) =>
{
        setPassword(event.target.value);
    };

    const handleLogin = () => {
        axios.post('https://localhost:7039/api/clients', {
            name: username,
            age: password
        })
        .then(() => {
            setLoggedIn(true);
            setUsername('');
            setPassword('');
            setError('');

            const token = generateToken();

            localStorage.setItem('accessToken', token);
            setAccessToken(token);
            window.location.href = "/home";
        })
        .catch(() => {
            setError('incorect username or password');
            setUsername('');
            setPassword('');
        });
    };

    const handleLogout = () => {
        setLoggedIn(false);
        setUsername('');
        setPassword('');
        setError('');
        setAccessToken('');
        localStorage.removeItem('accessToken');
    };

    const generateToken = () => {
        return 'asdadfcsacoasp[dko[aclpawkl1223dasdawe23';
    };

    return(
        <div>
            <Container className="mt-5">
                <Row className="justify-content-center">
                    <Col md={6}>
                        <h2 className="mb-4">Login</h2>
                        <Form>
                            <Form.Group controlId="formBasicUsername">
                                <Form.Label>Username</Form.Label>
                                <Form.Control type="text" placeholder="write
username" value={username} onChange={handleUsernameChange} />
                            </Form.Group>

                            <Form.Group controlId="formBasicPassword">
                                <Form.Label>Password</Form.Label>
                                <Form.Control type="password" placeholder="write
password" value={password} onChange={handlePasswordChange} />
                            </Form.Group>

                            <Button variant="primary" type="button"
onClick={handleLogin}>
                                Login
                            </Button>
                        </Form>
                    </Col>
                </Row>
                <div style={{ position: 'fixed', left: '10px', bottom: '70px' }}>
                    <Button variant="primary" type="button"
onClick={handleLogout}>Logout</Button>
                </div>
            </Container>
        </div>
    );
}
```

Cart.tsx

```
import React, { useEffect, useState } from 'react';
```

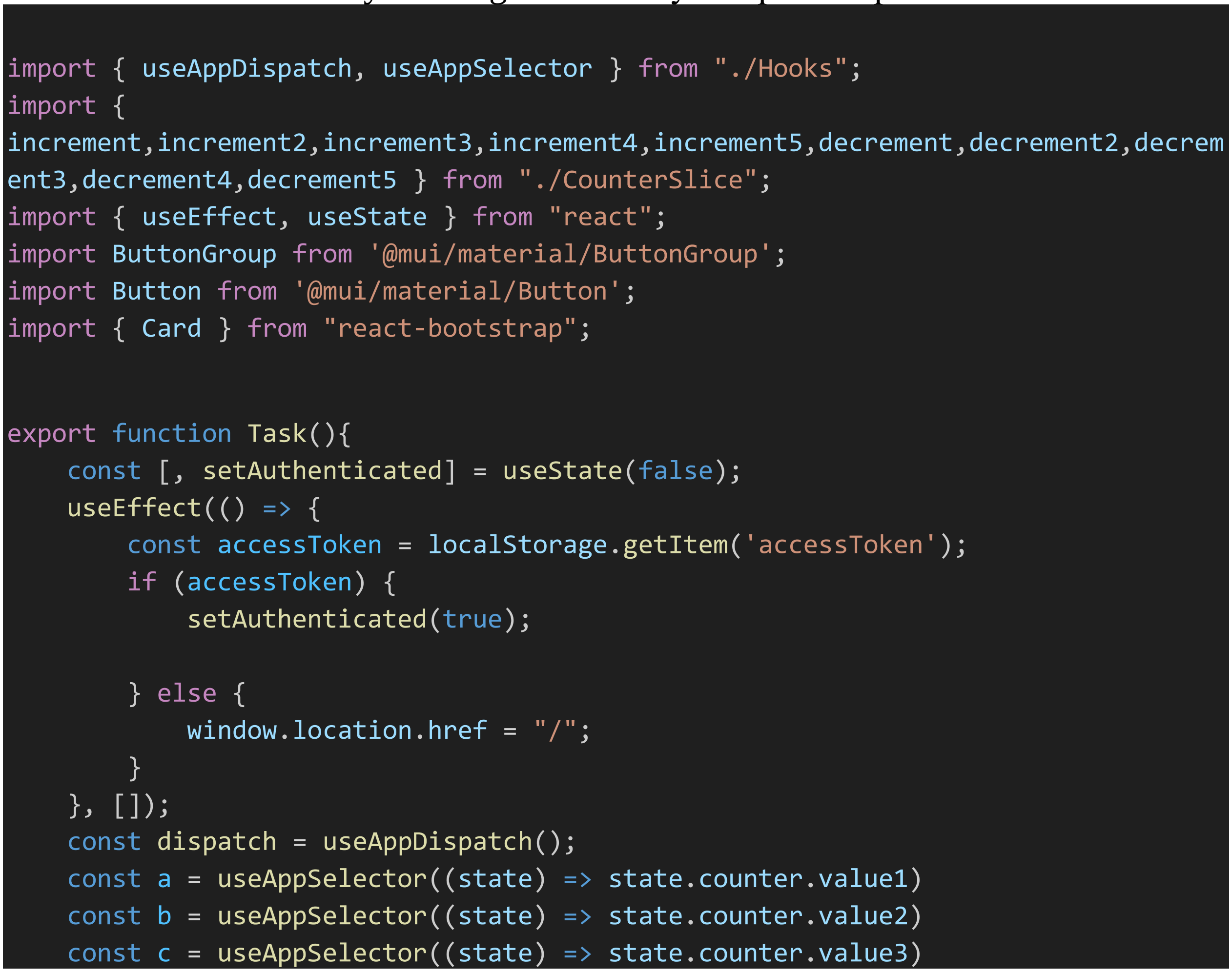


```
import { Button, ListGroup, ListGroupItem, Card, Container, Row, Col } from
'react-bootstrap';

interface Product {
    id: number;
    title: string;
    description: string;
    image: string;
}

export function Cart() {
  const [, setAuthenticated] = useState(false);
  useEffect(() => {
      const accessToken = localStorage.getItem('accessToken');
      if (accessToken) {
          setAuthenticated(true);
      } else {
          window.location.href = "/";
      }
  }, []);
    const [cartItems, setCartItems] = React.useState<Product[]>(() => {
        const localData = localStorage.getItem('cart');
        return localData ? JSON.parse(localData) : [];
    });

    const removeItem = (id: number) => {
        const updatedCart = cartItems.filter(item => item.id !== id);
        setCartItems(updatedCart);
        localStorage.setItem('cart', JSON.stringify(updatedCart));
    };
    const clearCart = () => {
        setCartItems([]);
        localStorage.removeItem('cart');
    };

    return (
        <Container>
            <h1 className="my-4">Shopping Cart</h1>
            {cartItems.length > 0 ? (
                <ListGroup>
                    {cartItems.map((item, index) => (
                        <ListGroupItem key={index} className="mb-3">
                            <Card>
                                <Card.Body>
                                    <Row>
                                        <Col md={4}>
                                            <Card.Img variant="top"
src={item.image} alt={item.title} />
                                        </Col>
                                        <Col md={8}>
                                            <Card.Title>{item.title}</Card.Title>
```

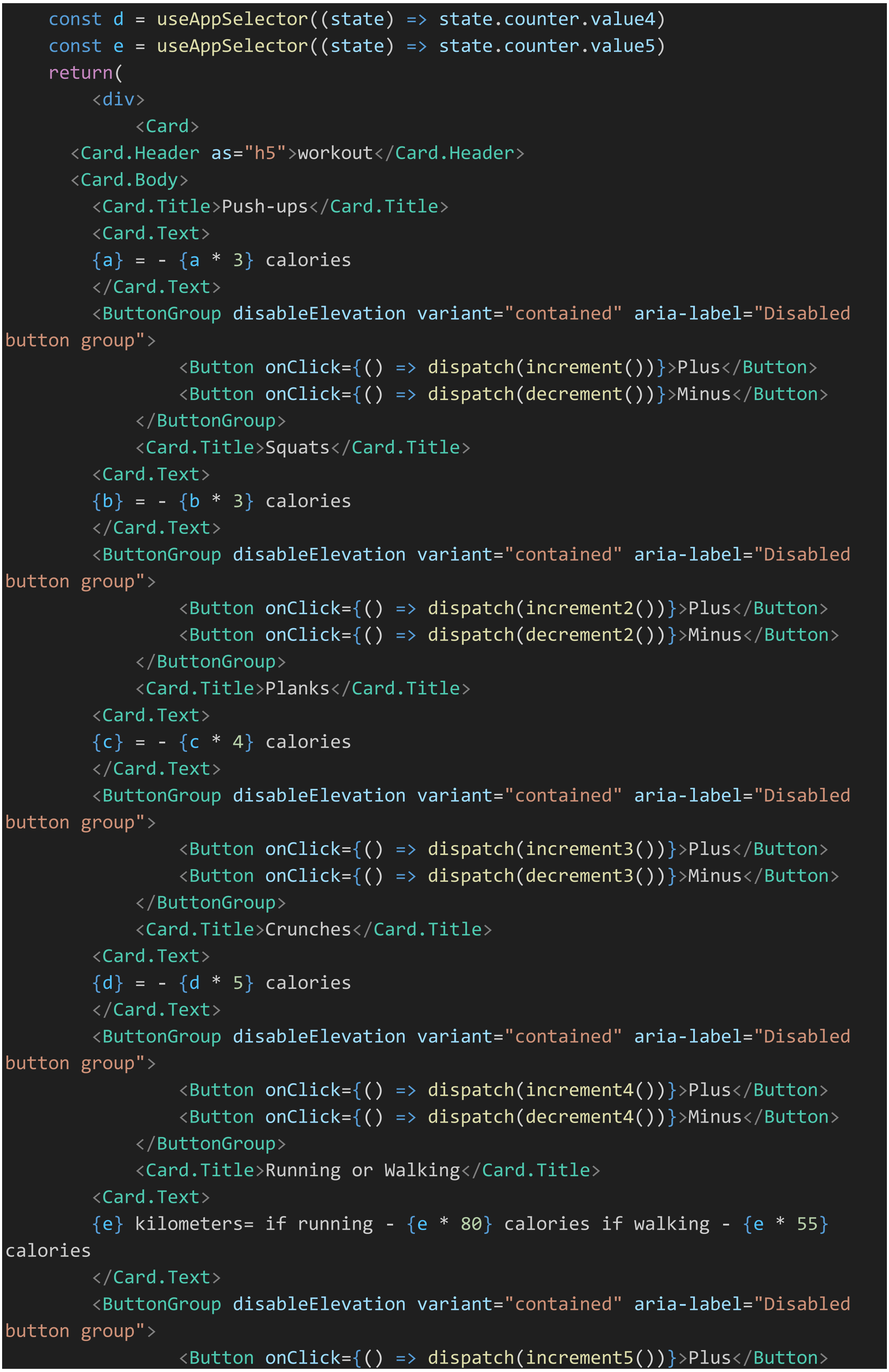


```
                                            <Card.Text>{item.description}</Card.T
ext>
                                            <Button variant="danger" onClick={()
=> removeItem(item.id)}>Remove Item</Button>
                                        </Col>
                                    </Row>
                                </Card.Body>
                            </Card>
                        </ListGroupItem>
                    ))}
                </ListGroup>
            ) : (
                <p>Your cart is empty.</p>
            )}
            <Button variant="warning" onClick={clearCart} style={{ display:
cartItems.length ? 'block' : 'none', marginTop: '20px' }}>
                Clear Cart
            </Button>
        </Container>
    );
}
```

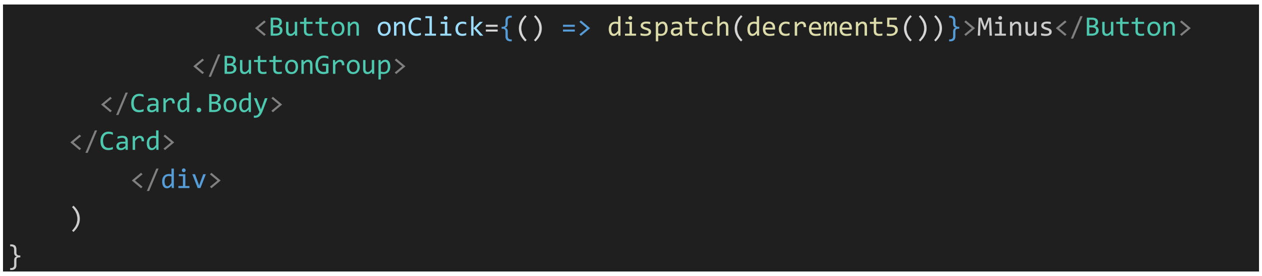
DailyTaskPage.tsx калькулятор каллорий



```
import { useAppDispatch, useAppSelector } from "./Hooks";
import {
increment,increment2,increment3,increment4,increment5,decrement,decrement2,decrem
ent3,decrement4,decrement5 } from "./CounterSlice";
import { useEffect, useState } from "react";
import ButtonGroup from '@mui/material/ButtonGroup';
import Button from '@mui/material/Button';
import { Card } from "react-bootstrap";


export function Task(){
    const [, setAuthenticated] = useState(false);
    useEffect(() => {
        const accessToken = localStorage.getItem('accessToken');
        if (accessToken) {
            setAuthenticated(true);

        } else {
            window.location.href = "/";
        }
    }, []);
    const dispatch = useAppDispatch();
    const a = useAppSelector((state) => state.counter.value1)
    const b = useAppSelector((state) => state.counter.value2)
    const c = useAppSelector((state) => state.counter.value3)
```



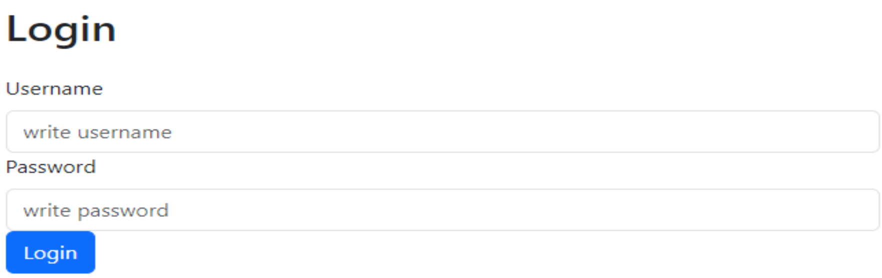
```
    const d = useAppSelector((state) => state.counter.value4)
    const e = useAppSelector((state) => state.counter.value5)
    return(
        <div>
            <Card>
      <Card.Header as="h5">workout</Card.Header>
      <Card.Body>
        <Card.Title>Push-ups</Card.Title>
        <Card.Text>
        {a} = - {a * 3} calories
        </Card.Text>
        <ButtonGroup disableElevation variant="contained" aria-label="Disabled button group">
                <Button onClick={() => dispatch(increment())}>Plus</Button>
                <Button onClick={() => dispatch(decrement())}>Minus</Button>
            </ButtonGroup>
            <Card.Title>Squats</Card.Title>
        <Card.Text>
        {b} = - {b * 3} calories
        </Card.Text>
        <ButtonGroup disableElevation variant="contained" aria-label="Disabled button group">
                <Button onClick={() => dispatch(increment2())}>Plus</Button>
                <Button onClick={() => dispatch(decrement2())}>Minus</Button>
            </ButtonGroup>
            <Card.Title>Planks</Card.Title>
        <Card.Text>
        {c} = - {c * 4} calories
        </Card.Text>
        <ButtonGroup disableElevation variant="contained" aria-label="Disabled button group">
                <Button onClick={() => dispatch(increment3())}>Plus</Button>
                <Button onClick={() => dispatch(decrement3())}>Minus</Button>
            </ButtonGroup>
            <Card.Title>Crunches</Card.Title>
        <Card.Text>
        {d} = - {d * 5} calories
        </Card.Text>
        <ButtonGroup disableElevation variant="contained" aria-label="Disabled button group">
                <Button onClick={() => dispatch(increment4())}>Plus</Button>
                <Button onClick={() => dispatch(decrement4())}>Minus</Button>
            </ButtonGroup>
            <Card.Title>Running or Walking</Card.Title>
        <Card.Text>
        {e} kilometers= if running - {e * 80} calories if walking - {e * 55} calories
        </Card.Text>
        <ButtonGroup disableElevation variant="contained" aria-label="Disabled button group">
                <Button onClick={() => dispatch(increment5())}>Plus</Button>
```



```
                <Button onClick={() => dispatch(decrement5())}>Minus</Button>
            </ButtonGroup>
      </Card.Body>
    </Card>
        </div>
    )
}
```

Пример работы

Авторизация



Login

Username

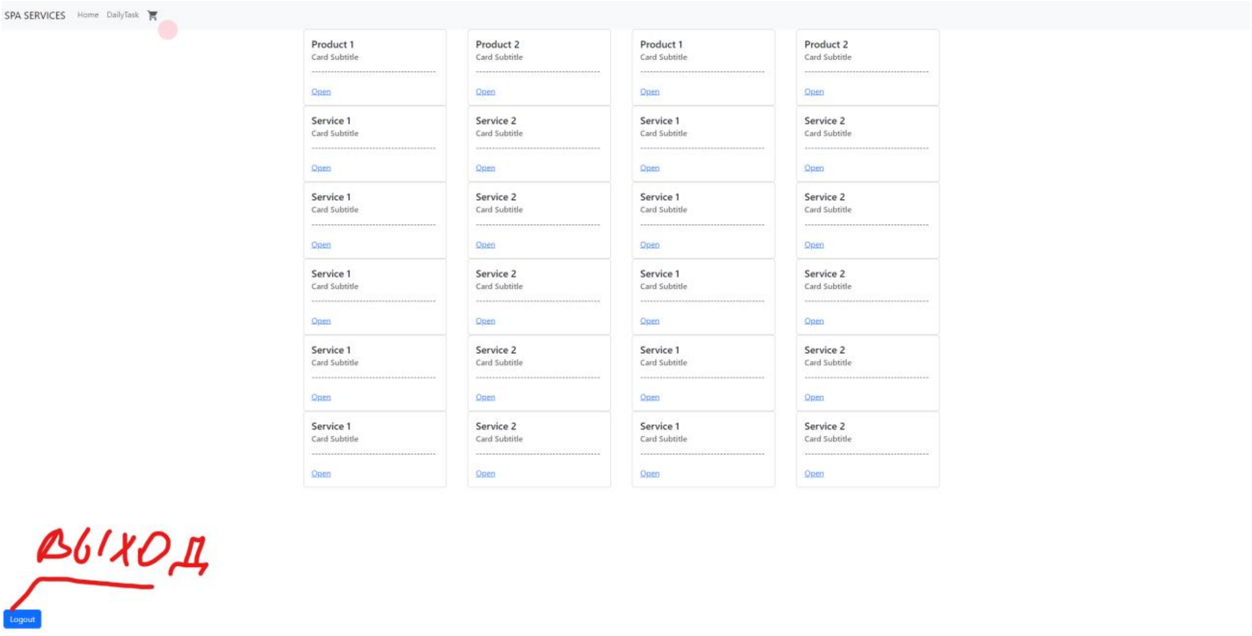
write username

Password

write password

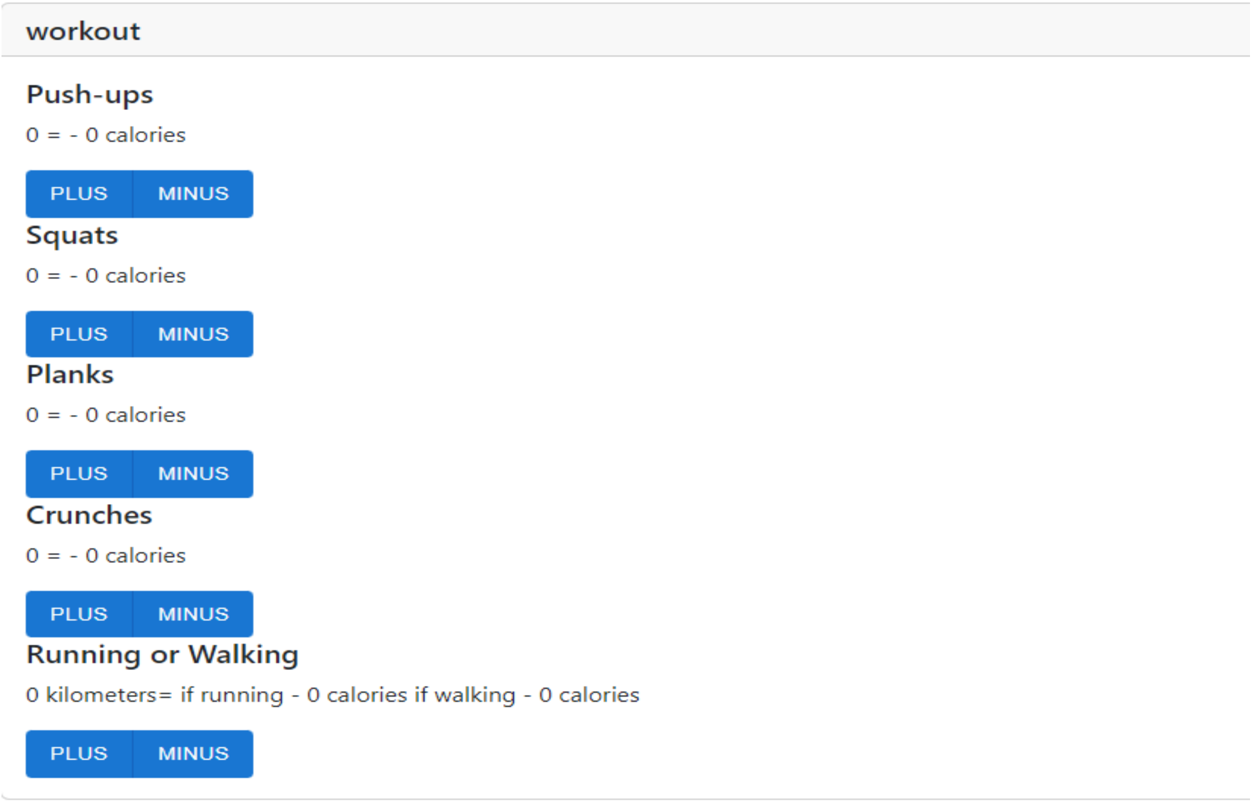
При переходе на страницы home и т.д идёт возврат на страницу авторизации т.к токен отсутствует

Главная страница с товарами



Product 1 Card Subtitle Open	Product 2 Card Subtitle Open	Product 1 Card Subtitle Open	Product 2 Card Subtitle Open
Service 1 Card Subtitle Open	Service 2 Card Subtitle Open	Service 1 Card Subtitle Open	Service 2 Card Subtitle Open
Service 1 Card Subtitle Open	Service 2 Card Subtitle Open	Service 1 Card Subtitle Open	Service 2 Card Subtitle Open
Service 1 Card Subtitle Open	Service 2 Card Subtitle Open	Service 1 Card Subtitle Open	Service 2 Card Subtitle Open
Service 1 Card Subtitle Open	Service 2 Card Subtitle Open	Service 1 Card Subtitle Open	Service 2 Card Subtitle Open
Service 1 Card Subtitle Open	Service 2 Card Subtitle Open	Service 1 Card Subtitle Open	Service 2 Card Subtitle Open

Dailytask



workout

Push-ups

0 = - 0 calories

Squats

0 = - 0 calories

Planks

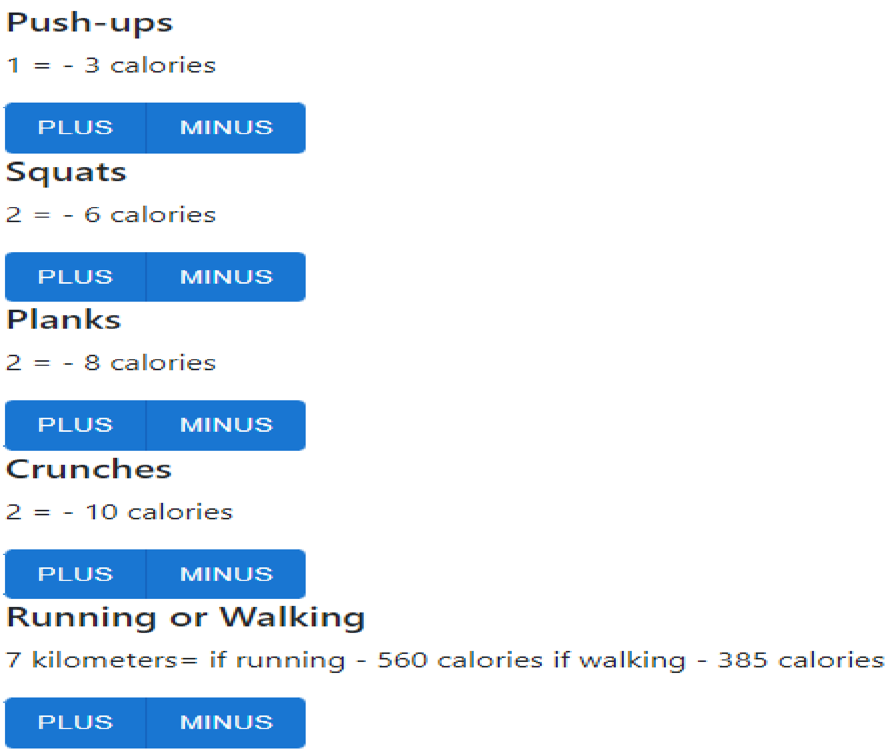
0 = - 0 calories

Crunches

0 = - 0 calories

Running or Walking

0 kilometers= if running - 0 calories if walking - 0 calories



Push-ups

1 = - 3 calories

Squats

2 = - 6 calories

Planks

2 = - 8 calories

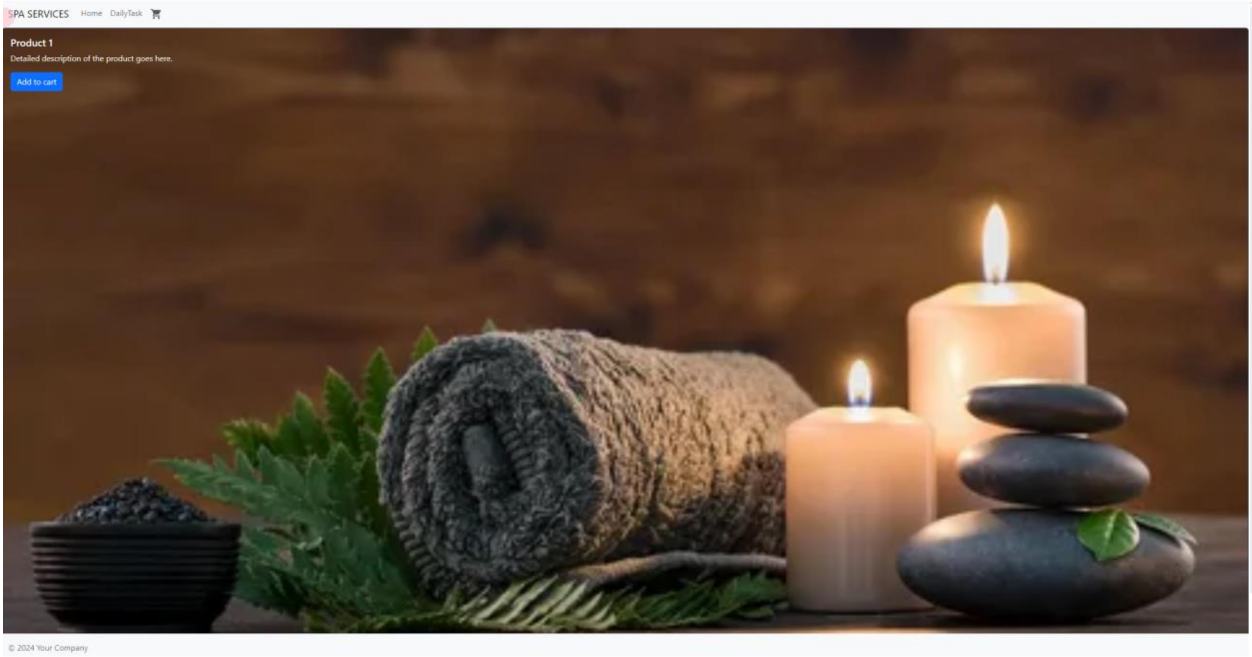
Crunches

2 = - 10 calories

Running or Walking

7 kilometers= if running - 560 calories if walking - 385 calories

Page.tsx



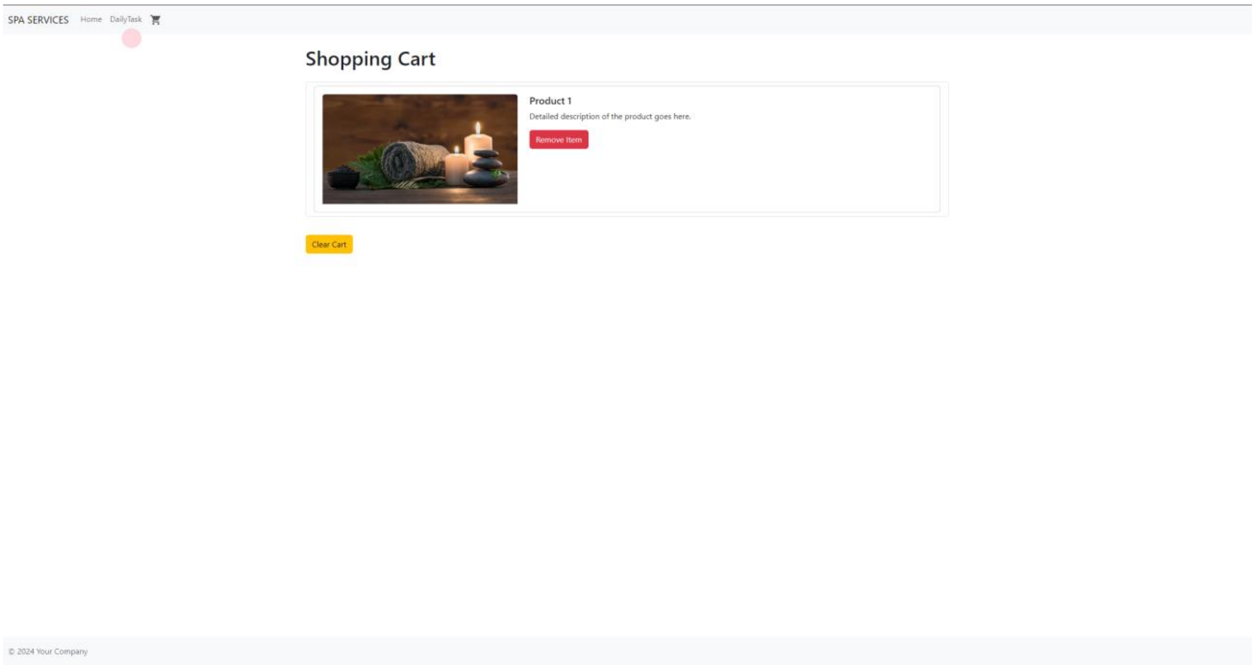
Product 1

Detailed description of the product goes here.

© 2024 Your Company

Cart.tsx

С товаром с страницы page



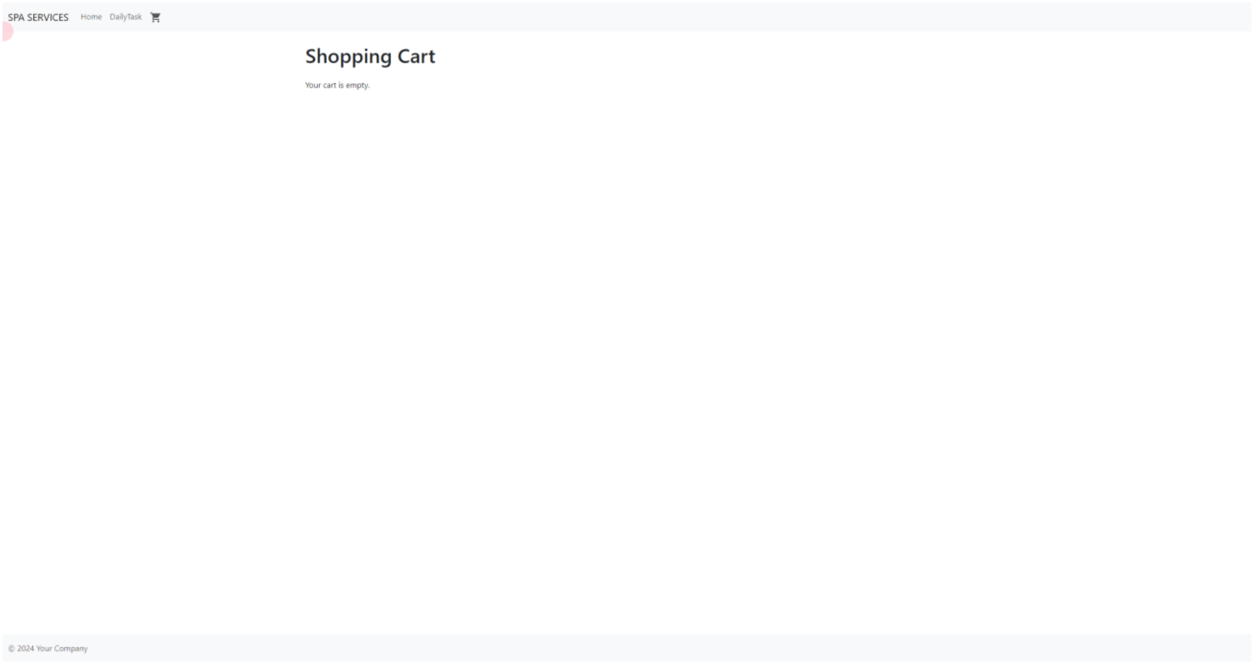
Shopping Cart

Product 1

Detailed description of the product goes here.

© 2024 Your Company

И без после удалиние через кнопку



Shopping Cart

Your cart is empty.

© 2024 Your Company