

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчёт

Лабораторная работа №1

Выполнила:

Злотникова Карина

K33392

Проверил:

Добряков Д. И.

Задание

Калькулятор калорий: Разработка инструмента для расчета суточной нормы калорий с учетом введенных пользователем данных о поле, возрасте, росте, весе и уровне физической активности.

Решение

1. Вход и регистрация.

Была выполнена верстка страницы регистрации и входа.

Скриншот:

Добро пожаловать в
Калорийк

Электронная почта

Пароль

Регистрация

Вход

Файл SingUp.tsx

```
import { useState } from "react";
//import axios from "axios";
import { useNavigate } from "react-router-dom";
import "../index.scss";
export const SignUp = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const nav = useNavigate();

  const signInHandle = () => {
    //Пока нет бд регистрацию не обрабатываем
    alert("Регистрация прошла успешно!");
    nav("main");
  };
};
```

```

const logInHandle = () => {
  //Пока нет бд вход по почте не обрабатываем
  nav("main");

return (
  <div className="wrapper">
    <h1>
      Добро пожаловать в <h1 className="headerTitle">Калорийк</h1>
    </h1>
    <div className="options">
      <div className="option">
        <h2>Электронная почта</h2>
        <input
          type="email"
          value={email}
          onChange={ (e) => setEmail(e.target.value) }
          className="input"
        />
      </div>
      <div className="option">
        <h2>Пароль</h2>
        <input
          type="password"
          value={password}
          onChange={ (e) => setPassword(e.target.value) }
          className="input"
        />
      </div>
    </div>
    <div className="buttons">
      <button className="button" onClick={signInHandle}>
        Регистрация
      </button>
      <button className="button" onClick={logInHandle}>
        Вход
      </button>
    </div>
  </div>
);
};

```

2. Калькулятор калорий.

Была выполнена верстка главной страницы, где расположен калькулятор калорий, суточной нормы калорий, поиск продуктов и карточки продуктов с функциональными кнопками потребления.

Скриншот:

Калорийк

Найти продукт...

Какая твоя норма калорий в день? Давай посчитаем!


Пол: Мужской

Рост (в см):

Вес (в кг):

Возраст:


Рассчитать



Яблоко

Пищевая ценность: 47кКал


Съел(a)



Банан

Пищевая ценность: 96кКал


Съел(a)



Киви

Пищевая ценность: 48кКал

Съел(a)



Яблоко

Пищевая ценность: 47кКал

Съел(a)

3. Калькулятор потребления.

При расчете суточной нормы калорий выводится также счетчик потребленных за день калорий. Нажав на кнопку в карточке продукта, пользователь может отметить его потребление и автоматически выполнится расчет оставшихся на день калорий.

Калорийк

Найти продукт...

Какая твоя норма калорий в день? Давай посчитаем!

Пол: Мужской

Рост (в см): 180


Вес (в кг): 70


Возраст: 30


Рассчитать


Ваша суточная норма калорий: 1721.00 кал

Остаток суточной нормы калорий: 1721 кал









Файл Main.tsx

```
import "../../index.scss";
import "../Main.scss";
import { useState, useEffect } from "react";
import { FoodItem } from "../../components/FoodItem/FoodItem";
import axios from "axios";
import { Header } from "../../components/Header/Header";
import FoodData from "../../db.json";
type Food = {
  title: string;
  foodImg: string;
  calories: number;
};

export const Main = () => {
  const [gender, setGender] = useState("male");
  const [height, setHeight] = useState("");
  const [weight, setWeight] = useState("");
  const [age, setAge] = useState("");
  const [bmr, setBmr] = useState(0);
  const [foods, setFoods] = useState<Food[]>(FoodData.foods);
  const [remainingCalories, setRemainingCalories] = useState(0);
  const [searchQuery, setSearchQuery] = useState("");

  const handleEatClick = (calories: number) => {
    const updatedRemainingCalories = remainingCalories - calories;
    setRemainingCalories(updatedRemainingCalories);
  };

  const handleSearch = (query: string) => {
    setSearchQuery(query);
  };

  const filteredFoods = foods.filter((food) =>
    food.title.toLowerCase().includes(searchQuery.toLowerCase())
  );

  useEffect(() => {
    axios
      .get<Food[]>("http://localhost:3000/foods")
      .then((response) => {
        setFoods(response.data);
      })
      .catch((error) => {
        console.error("Ошибка получения данных о продуктах:", error);
      });
  });
}
```

```

    });
  }, []);

const handleCalculateCalories = (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();

  if (!gender) {
    alert("Выберите пол");
    return;
  }

  const weightFloat = parseFloat(weight);
  const heightFloat = parseFloat(height);
  const ageFloat = parseFloat(age);

  if (isNaN(weightFloat) || isNaN(heightFloat) || isNaN(ageFloat)) {
    alert("Пожалуйста, введите корректные данные");
    return;
  }

  let bmr = 0;
  if (gender === "female") {
    bmr = 655 + 9.6 * weightFloat + 1.8 * heightFloat - 4.7 * ageFloat;
  } else if (gender === "male") {
    bmr = 66 + 13.7 * weightFloat + 5 * heightFloat - 6.8 * ageFloat;
  }

  setBmr(bmr);
  setRemainingCalories(bmr);

  // alert(`Ваша норма калорий в день: ${bmr.toFixed(2)}`);
};

return (
  <div className="mainWrapper">
    <Header onSearch={handleSearch} />
    <h1 className="text">
      Какая твоя норма калорий в день? Давай посчитаем!
    </h1>
    <form onSubmit={handleCalculateCalories} className="mainForm">
      <div className="formGroup">
        <label>Пол:</label>
        <select
          className="formOption"
          value={gender}

```

```

        onChange={ (e) => setGender(e.target.value) }
      >
      <option value="male">Мужской</option>
      <option value="female">Женский</option>
    </select>
  </div>
  <div className="formGroup">
    <label>Рост (в см):</label>
    <input
      className="formOption"
      type="number"
      value={height}
      onChange={ (e) => setHeight(e.target.value) }
    />
  </div>
  <div className="formGroup">
    <label>Вес (в кг):</label>
    <input
      className="formOption"
      type="number"
      value={weight}
      onChange={ (e) => setWeight(e.target.value) }
    />
  </div>
  <div className="formGroup">
    <label>Возраст:</label>
    <input
      className="formOption"
      type="number"
      value={age}
      onChange={ (e) => setAge(e.target.value) }
    />
  </div>
  <div className="buttons">
    <button className="button" type="submit">
      Рассчитать
    </button>
  </div>
</form>
{bmr > 0 && (
  <h2 className="text">
    Ваша суточная норма калорий: {bmr.toFixed(2)} кал
  </h2>
)}
{remainingCalories > 0 && (

```

```

        <div className="text">
            Остаток суточной нормы калорий: {remainingCalories} кал
        </div>
    )}
    <div className="foodSelection">
        {filteredFoods.map((food, index) => (
            <div key={index}>
                <FoodItem
                    title={food.title}
                    foodImg={food.foodImg}
                    calories={food.calories}
                    onEatClick={() => handleEatClick(food.calories)}
                />
            </div>
        ))}
    </div>
</div>
);
};

```

Файл FoodItem.tsx

```

import "../FoodItem.scss";
import "../../index.scss";
type FoodItemProps = {
    title: string;
    foodImg: string;
    calories: number;
    onEatClick: () => void;
};

export const FoodItem = ({
    title,
    foodImg,
    calories,
    onEatClick,
}: FoodItemProps) => {
    return (
        <div className="foodWrapper">
            <img className="foodImg" src={foodImg} />
            <h3>{title}</h3>
            <p>Пищевая ценность: {calories}кКал</p>
            <div className="buttonsWrapper">
                <div className="buttons">
                    <button className="button" onClick={onEatClick}>

```



```

        Съел(a)
      </button>
    </div>
  </div>
</div>
);
};

```

4. Поиск.

При введении пользователем названия продукта в строку поиска, происходит автоматическая фильтрация и отображение подходящих по названию продуктов, на главной странице.

Файл Header.tsx

```

type HeaderProps = {
  onSearch: (query: string) => void;
};

export const Header = ({ onSearch }: HeaderProps) => {
  const handleSearch = (e: React.ChangeEvent<HTMLInputElement>) => {
    onSearch(e.target.value);
  };

  return (
    <div className="headerWrapper">
      <h1>Калорийк</h1>
      <input
        className="input"
        placeholder="Найти продукт..."
        onChange={handleSearch}
      />
    </div>
  );
};

```

Часть кода файла Main.tsx

```

const handleSearch = (query: string) => {
  setSearchQuery(query);
};

```

```
const filteredFoods = foods.filter((food) =>
  food.title.toLowerCase().includes(searchQuery.toLowerCase())
);
```

Вывод

Верстка и функционал были реализованы.