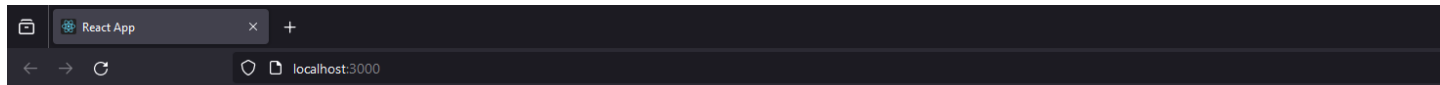


React - Kalkulator BMI



Obliczanie BMI

- 1Waga: 55 kg, Wzrost: 180 cm, BMI: 16.98, Data: 2024-12-04
- 2Waga: 60 kg, Wzrost: 184 cm, BMI: 17.72, Data: 2024-05-11

```
4
5
6  /*****
7   * klasa: App
8   * opis:
9   *   Główny komponent aplikacji do obliczania i zarządzania
10  *   listą wskaźników BMI. Umożliwia dodawanie, edycję oraz
11  *   usuwanie wpisów.
12  * pola:
13  *   items - lista przechowująca wpisy BMI
14  * autor: Bartek Pyka
15  *****/
16
17 export default function App() {
18   const [items, setItems] = useState([]);
19   const [weight, setWeight] = useState('');
20   const [height, setHeight] = useState('');
21   const [date, setDate] = useState('');
22
23   // Zatwierdzenie formularza
24   const handleSubmit = (e) => {
25     e.preventDefault();
26     const bmi = (weight / Math.pow(height / 100, 2)).toFixed(2);
27     setItems([...items, { weight, height, bmi, date }]);
28     setWeight('');
29     setHeight('');
30     setDate('');
31   };
32
33   // Usuwanie z tablicy danego elementu
34   const handleDeleteItem = (index) => {
35     setItems(items.filter((item, idx) => idx !== index));
36   };
37
38   return (
39     <div className='App'>
40       <h1>Obliczanie BMI</h1>
41       {/* Formularz */}
42       <form onSubmit={handleSubmit}>
43         <input type="number" value={weight} onChange={e => setWeight(e.target.value)} placeholder="Waga w kg" required/>
44         <input type="number" value={height} onChange={e => setHeight(e.target.value)} placeholder="Wzrost w cm" required/>
45         <input type="date" value={date} onChange={e => setDate(e.target.value)} required/>
46         <button type="submit">Dodaj</button>
47       </form>
48
49       {/* Lista */}
50       <ul style={{textAlign:"left"}}>
51         {items.map((item, index) => (
52           <li key={index}>
53             {index + 1}Waga: {item.weight} kg, Wzrost: {item.height} cm, BMI: {item.bmi}, Data: {item.date}
54             <button onClick={() => handleDeleteItem(index)}>Usuń</button>
55           </li>
56         ))}
57       </ul>
58     </div>
59   );
60 }
```

```

import React, { useState } from 'react';

import './App.css';

/*****
 * klasa: App
 * opis:
 *       Główny komponent aplikacji do obliczania i zarządzania
 *       listą wskaźników BMI. Umożliwia dodawanie, edycję oraz
 *       usuwanie wpisów.
 * pola:
 *       items - lista przechowująca wpisy BMI
 * autor: Bartek Pyka
 *****/

export default function App() {
  const [items, setItems] = useState([]);
  const [weight, setWeight] = useState('');
  const [height, setHeight] = useState('');
  const [date, setDate] = useState('');

  // Zatwierdzenie formularza
  const handleSubmit = (e) => {
    e.preventDefault();
    const bmi = (weight / Math.pow(height / 100, 2)).toFixed(2);
    setItems([...items, { weight, height, bmi, date }]);
    setWeight('');
    setHeight('');
    setDate('');
  };

  // Usuwanie z tablicy danego elementu
  const handleDeleteItem = (index) => {
    setItems(items.filter((item, idx) => idx !== index));
  };

  return (
    <div className='App'>
      <h1>Obliczanie BMI</h1>
      {/* Formularz */}
      <form onSubmit={handleSubmit}>
        <input type="number" value={weight} onChange={e => setWeight(e.target.value)}
placeholder="Waga w kg" required/>
        <input type="number" value={height} onChange={e => setHeight(e.target.value)}
placeholder="Wzrost w cm" required/>
        <input type="date" value={date} onChange={e => setDate(e.target.value)}
required/>
        <button type="submit">Dodaj</button>
      </form>
    </div>
  );
}

```

```
    {/* Lista */}
    <ul style={{textAlign:"left"}}>
      {items.map((item, index) => (
        <li key={index}>
          {index + 1}Waga: {item.weight} kg, Wzrost: {item.height} cm, BMI:
{item.bmi}, Data: {item.date}
          <button onClick={() => handleDeleteItem(index)}>Usuń</button>
        </li>
      ))}
    </ul>
  </div>
);
}
```