

## Laboratorium 09 - Programowanie Aplikacji Webowych - React

### Wprowadzenie:



#### 1. Sposób tworzenia aplikacji React'owych:

Aplikacje React'owe są tworzone na podstawie komponentów. Komponent jest to element interfejsu użytkownika, który ma własną logikę oraz wygląd (kolory, ułożenie). Komponentem może być mały element typu przycisk, a nawet całą stronę. Takie podejście pozwala izolować od siebie elementy, a następnie razem łączyć podczas końcowej fazy. Na przykład formularz może składać się z wielu elementów interfejsu jak etykiety, przyciski albo pola wejściowe. Każdy ten element może być komponentem. React korzysta z wirtualnego DOM'u (*Document Object Model*) podczas tworzenia aplikacji.



#### 2. Format JSX (*JavaScript eXtension*):

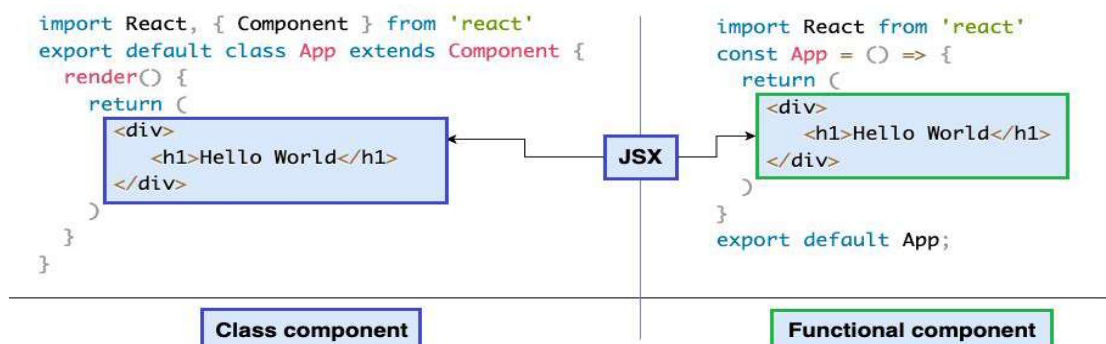
Format JSX pozwala na pisanie kodu JavaScript, który wygląda jak kod HTML. React nie wymaga używania JSX, ale większość deweloperów uważa to za przydatne podczas tworzenia aplikacji.



#### 3. Klasowe komponenty, a funkcyjne komponenty:

React pozwala na tworzenie komponentów na dwa sposoby: **klasowe** oraz **funkcyjne**. Pierwszy sposób pokazuje standardowe podejście, gdzie tworzy się klasę, która dziedziczy po klasie „Component”. Zaś drugie podejście jest zdefiniowaną funkcją, która nie może dziedziczyć po klasie, ze względów semantycznych/składniowych. Druga opcja jest łatwiejsza i szybsza w implementacji.

```
1 import React, { Component } from "react"; import React from "react";
2
3 class ClassComponent extends Component { function FunctionalComponent() {
4   render() { return <h1>Hello, world</h1>;
5     return <h1>Hello, world</h1>; }
6   }
7 }
```





## 0. Pobranie Node.js (npm - node package manager):

The screenshot shows the Node.js download page. It features a dark theme with a green header. The main content area is divided into two sections: 'LTS' (Recommended For Most Users) and 'Current' (Latest Features). Under 'LTS', there are three download options: Windows Installer (node-v20.10.0-x64.msi), macOS Installer (node-v20.10.0.pkg), and Source Code (node-v20.10.0.tar.gz). Under 'Current', there are three download options: Windows Installer (node-v20.10.0-x64.msi), macOS Installer (node-v20.10.0.pkg), and Source Code (node-v20.10.0.tar.gz). Below these options is a table with columns for the download type and the architecture (32-bit, 64-bit, ARM64, ARMv7, ARMv8).

Download Type	32-bit	64-bit	ARM64
Windows Installer (.msi)			
Windows Binary (.zip)			
macOS Installer (.pkg)			
macOS Binary (.tar.gz)			
Linux Binaries (x64)			
Linux Binaries (ARM)			
Source Code			

Linki:

<https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>

<https://nodejs.org/en/download>

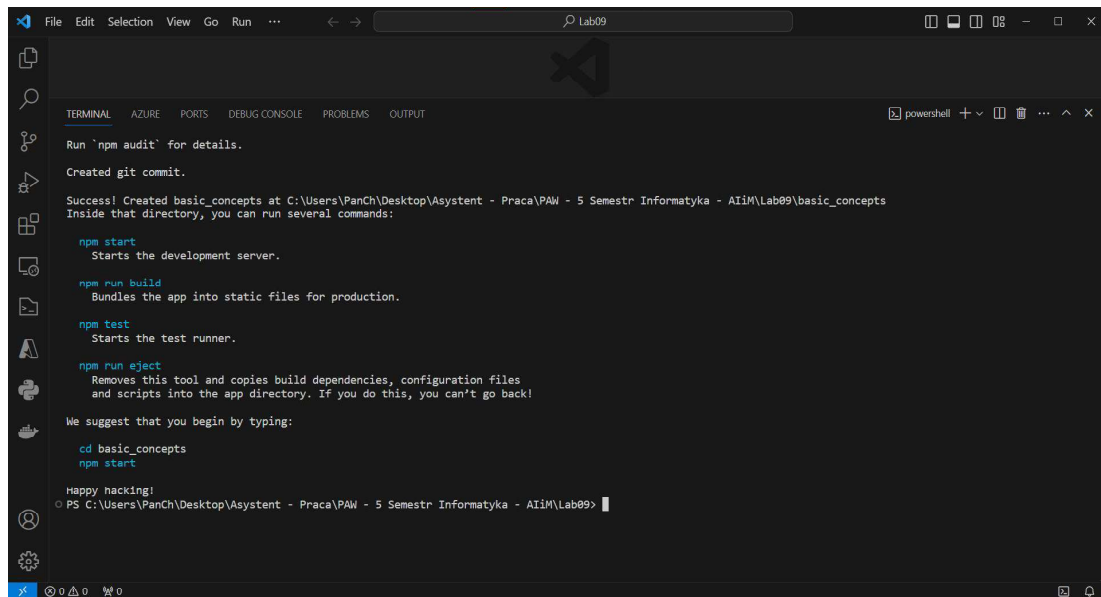


## 1. Utworzenie projektu/ folderu o nazwie „Lab09”.

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a folder named 'LAB09'. The main editor area is empty. The Terminal panel at the bottom shows the command prompt with the command 'npm create-react-app "basic\_concepts"' and the output 'webpack compiled with 1 warning'.

## 2. Wywołanie programu do tworzenia React projektu o nazwie „basic\_concepts”.

The screenshot shows a terminal window with the command prompt. The command 'npm create-react-app "basic\_concepts"' has been entered and executed. The output shows the command prompt with the command 'npm create-react-app "basic\_concepts"' and the output 'webpack compiled with 1 warning'.



```
File Edit Selection View Go Run ... Lab09
TERMINAL
Run 'npm audit' for details.
Created git commit.
Success! Created basic_concepts at C:\Users\PanCh\Desktop\Asystem - Praca\PAW - 5 Semestr Informatyka - AIIM\Lab09\basic_concepts
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

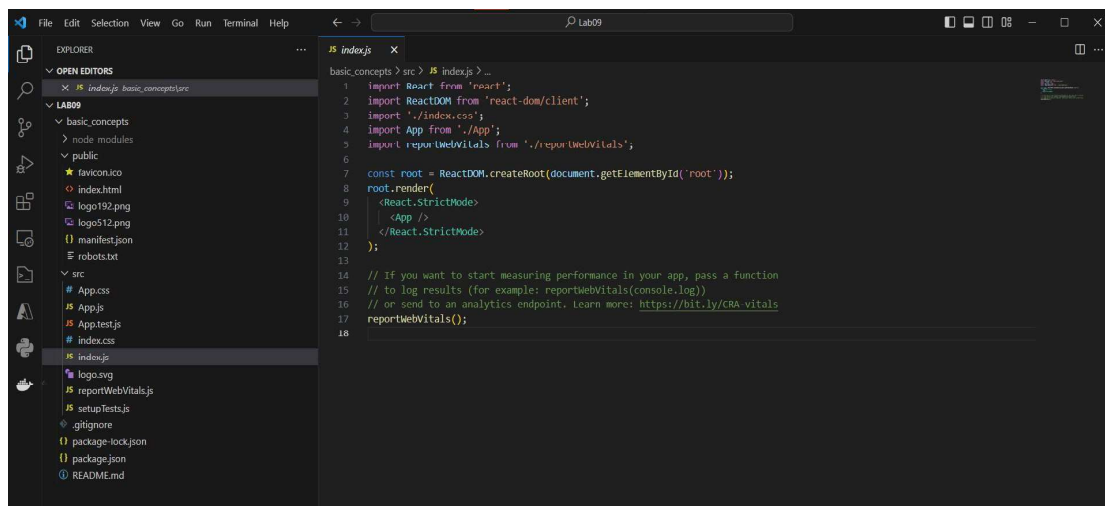
We suggest that you begin by typing:

  cd basic_concepts
  npm start

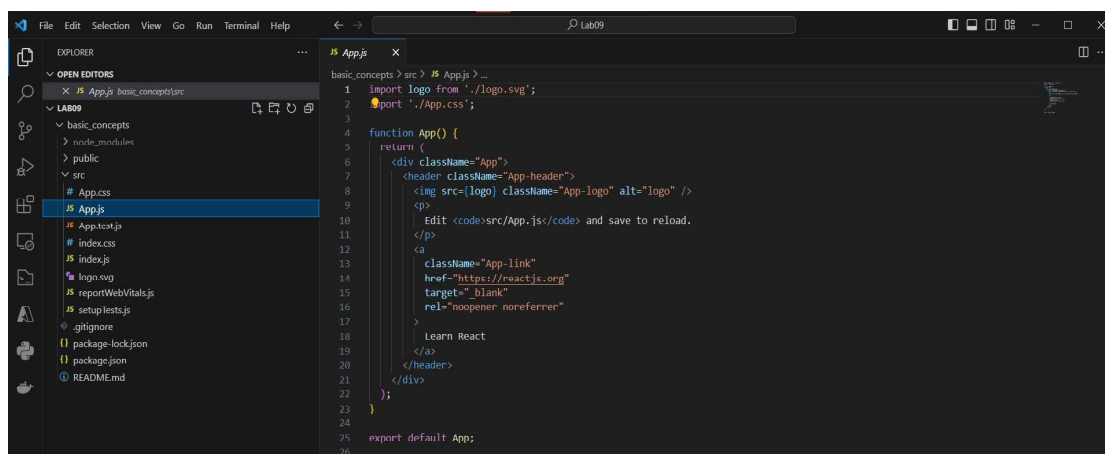
Happy hacking!
PS C:\Users\PanCh\Desktop\Asystem - Praca\PAW - 5 Semestr Informatyka - AIIM\Lab09>
```

3. Aktualna struktura projektu składająca się folderów:

- a) **node\_modules** - zawiera pakiety (.js) do projektu,
- b) **public** - publiczne zasoby,
- c) **src** - kody źródłowe aplikacji,
- d) **package.json** - lista pakietów oraz konfiguracji,



```
File Edit Selection View Go Run Terminal Help Lab09
EXPLORER
  OPEN EDITORS
  X JS index.js basic_concepts/src
  LAB09
    basic_concepts
      node_modules
      public
      favicon.ico
      index.html
      logo192.png
      logo512.png
      manifest.json
      robots.txt
      src
        App.css
        App.js
        App.test.js
        index.css
        index.js
        logo.svg
        reportWebVitals.js
        setupTests.js
        .gitignore
        package-lock.json
        package.json
        README.md
  JS index.js
    basic_concepts > src > JS index.js ...
    1 import React from 'react';
    2 import ReactDOM from 'react-dom/client';
    3 import './index.css';
    4 import App from './App';
    5 import {reportWebVitals} from './reportWebVitals';
    6
    7 const root = ReactDOM.createRoot(document.getElementById('root'));
    8 root.render(
    9   <React.StrictMode>
    10     <App />
    11   </React.StrictMode>
    12 );
    13
    14 // If you want to start measuring performance in your app, pass a function
    15 // to log results (for example: reportWebVitals(console.log))
    16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
    17 reportWebVitals();
    18
```



```
File Edit Selection View Go Run Terminal Help Lab09
EXPLORER
  OPEN EDITORS
  X JS App.js basic_concepts/src
  LAB09
    basic_concepts
      node_modules
      public
      src
        App.css
        App.js
        App.test.js
        index.css
        index.js
        logo.svg
        reportWebVitals.js
        setupTests.js
        .gitignore
        package-lock.json
        package.json
        README.md
  JS App.js
    basic_concepts > src > JS App.js ...
    1 import logo from './logo.svg';
    2 import './App.css';
    3
    4 function App() {
    5   return (
    6     <div className="App">
    7       <header className="App-header">
    8         <img src={logo} className="App-logo" alt="logo" />
    9         <p>
    10           Edit <code>src/App.js</code> and save to reload.
    11         </p>
    12         <a
    13           className="App-link"
    14           href="https://reactjs.org"
    15           target="_blank"
    16           rel="noopener noreferrer"
    17         >
    18           Learn React
    19         </a>
    20       </header>
    21     </div>
    22   );
    23 }
    24
    25 export default App;
    26
```

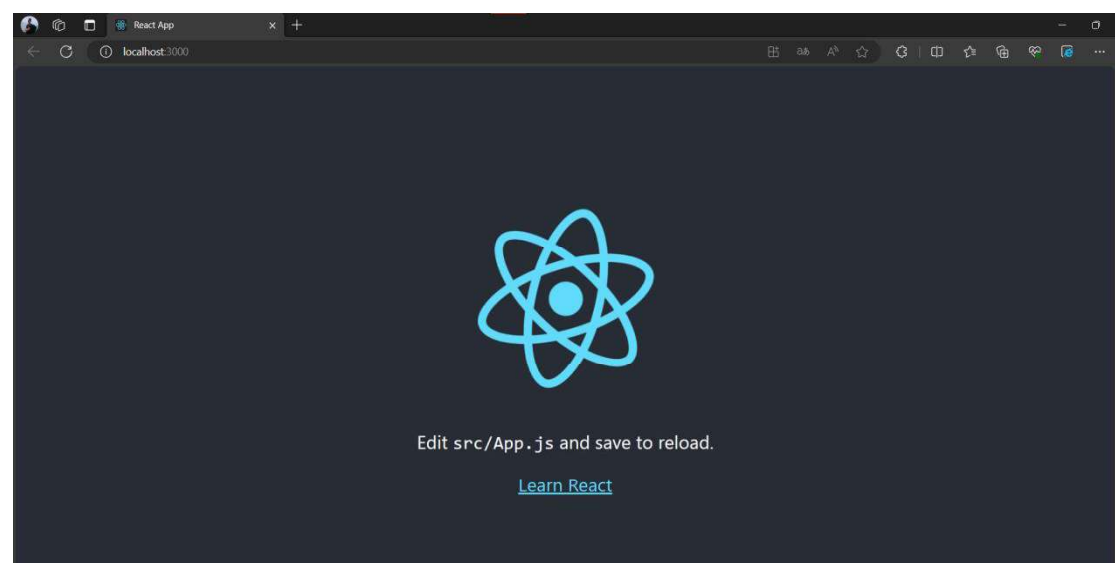
#### 4. Uruchomienie aplikacji: **npm start**

```
PS C:\Users\PanCh\Desktop\Asystent - Praca\PAW - 5 Semestr Informatyka - AIiM\Lab09> cd .\basic_concepts\
PS C:\Users\PanCh\Desktop\Asystent - Praca\PAW - 5 Semestr Informatyka - AIiM\Lab09\basic_concepts> dir

Directory: C:\Users\PanCh\Desktop\Asystent - Praca\PAW - 5 Semestr Informatyka - AIiM\Lab09\basic_concepts

Mode                LastWriteTime         Length Name
----                -
d-----         02.12.2023        13:42         node_modules
d-----         02.12.2023        13:42         public
d-----         02.12.2023        13:42         src
-a-----         02.12.2023        13:42          310 .gitignore
-a-----         02.12.2023        13:42     1263906 package-lock.json
-a-----         02.12.2023        13:42          817 package.json
-a-----         02.12.2023        13:42          3359 README.md

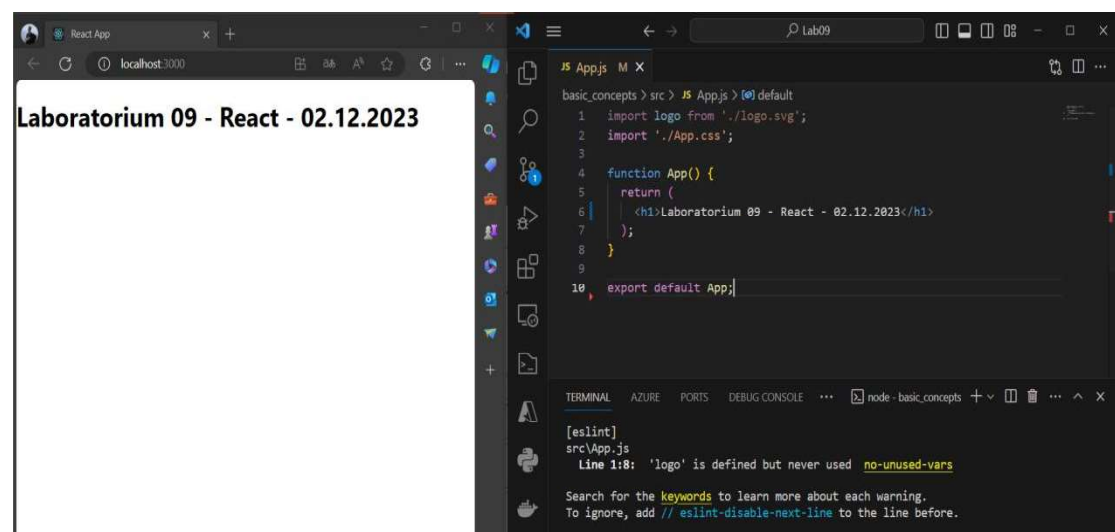
PS C:\Users\PanCh\Desktop\Asystent - Praca\PAW - 5 Semestr Informatyka - AIiM\Lab09\basic_concepts> npm start
```



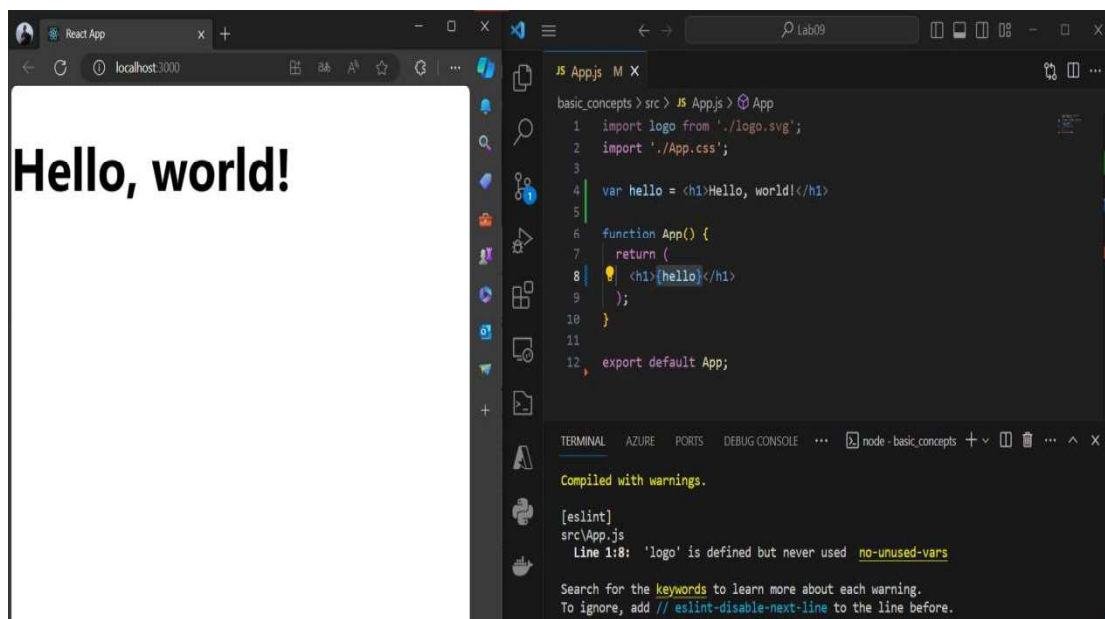
#### 5. Modyfikacja **App.js**, uproszczenie:



##### A) Ustawienie tylko jednego znacznika:



B) Przypisanie znacznika do zmiennej, następnie jej umieszczenie:

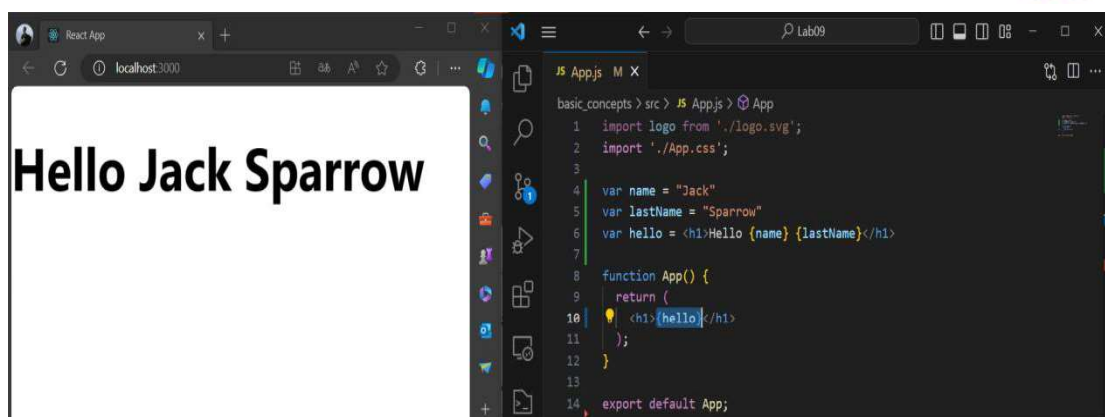


The screenshot shows a web browser at localhost:3000 displaying "Hello, world!". The code editor on the right shows the following code:

```
1 import logo from './logo.svg';
2 import './App.css';
3
4 var hello = <h1>Hello, world!</h1>
5
6 function App() {
7   return (
8     <h1>{hello}</h1>
9   );
10 }
11
12 export default App;
```

The terminal at the bottom shows a warning from eslint: "Line 1:8: 'logo' is defined but never used no-unused-vars".

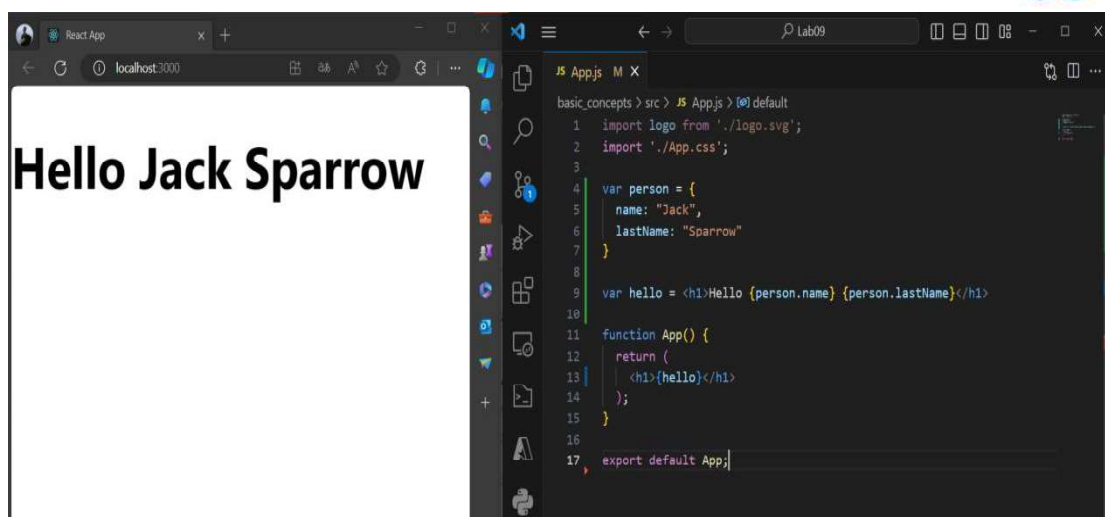
C) Połączenie dwóch zmiennych w JSX:



The screenshot shows a web browser at localhost:3000 displaying "Hello Jack Sparrow". The code editor on the right shows the following code:

```
1 import logo from './logo.svg';
2 import './App.css';
3
4 var name = "Jack"
5 var lastName = "Sparrow"
6 var hello = <h1>Hello {name} {lastName}</h1>
7
8 function App() {
9   return (
10    <h1>{hello}</h1>
11  );
12 }
13
14 export default App;
```

D) Wykorzystanie klasy/ obiektu w JSX:



The screenshot shows a web browser at localhost:3000 displaying "Hello Jack Sparrow". The code editor on the right shows the following code:

```
1 import logo from './logo.svg';
2 import './App.css';
3
4 var person = {
5   name: "Jack",
6   lastName: "Sparrow"
7 }
8
9 var hello = <h1>Hello {person.name} {person.lastName}</h1>
10
11 function App() {
12   return (
13     <h1>{hello}</h1>
14   );
15 }
16
17 export default App;
```





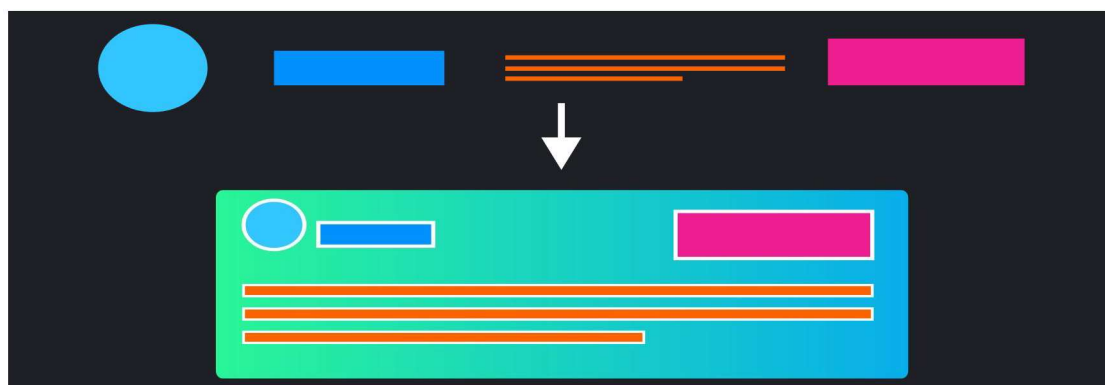
E) Stworzenie wielu obiektów, a następnie ich wyświetlenie:

```
1 var people =
2   [
3     {name: "Jack", lastName: "Sparrow"},
4     {name: "Geralt", lastName: "Rivia"},
5     {name: "Bruce", lastName: "Wayne"},
6     {name: "Peter", lastName: "Parker"},
7   ]
8
9 function SayHello(person)
10 {
11   return <h1>Hello: {person.name} {person.lastName}</h1>
12 }
13
14 var greetings01 = SayHello(people[0])
15 var greetings02 = SayHello(people[1])
16 var greetings03 = SayHello(people[2])
17 var greetings04 = SayHello(people[3])
18
19 function App() {
20   return (
21     <div className="greetings">
22       <h1>{greetings01}</h1>
23       <h2>{greetings02}</h2>
24       <h3>{greetings03}</h3>
25       <h4>{greetings04}</h4>
26     </div>
27   );
28 }
29
30 export default App;
```



F) Zamiana obiektów na komponenty, zgodnie z praktyką:

```
1 function Person(props)
2 {
3   return <div className="person">
4     <h2>
5       Name: {props.name},
6       Last name: {props.lastName},
7       Age: {props.age}
8     </h2>
9   </div>
10 }
11
12 function App() {
13   return (
14     <div className="greetings">
15       <Person name="Jack" lastName="Sparrow" age={27}/>
16       <Person name="Bruce" lastName="Wayne" age={30}/>
17       <Person name="Peter" lastName="Parker" age={25}/>
18     </div>
19   );
20 }
21
22 export default App;
```





## 8. Dodanie funkcji warunkowych w zależności od wieku:

The screenshot shows a web browser at localhost:3000 displaying three lines of text: "Name: Jack, Last name: Sparrow, Age: 27 - Young.", "Name: Bruce, Last name: Wayne, Age: 30 - Old.", and "Name: Peter, Last name: Parker, Age: 25 - Young.". The VS Code editor on the right shows the following code:

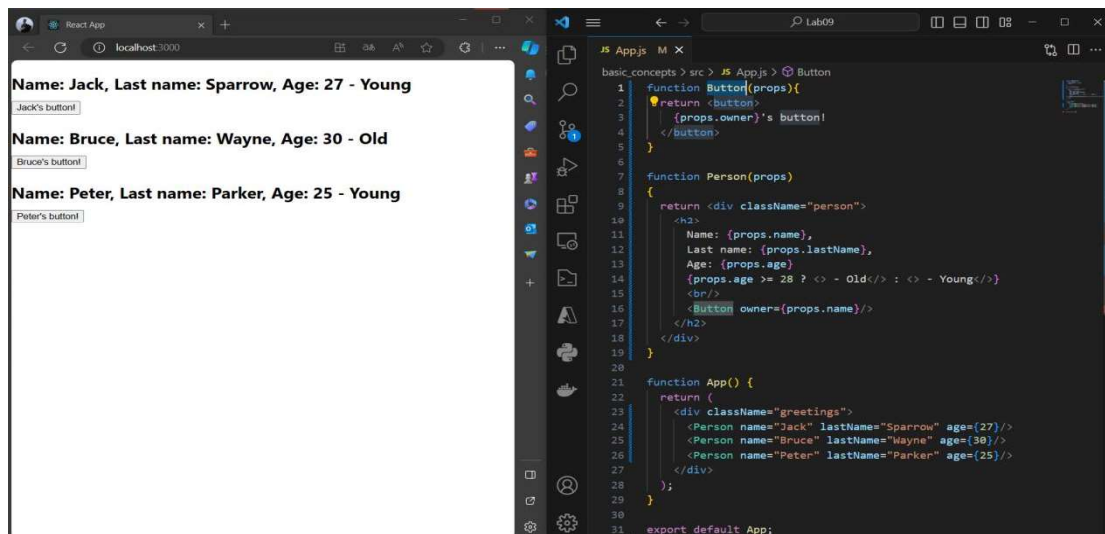
```
1 function Person(props)
2 {
3   return <div className="person">
4     <h2>
5       Name: {props.name},
6       Last name: {props.lastName},
7       Age: {props.age}
8       {props.age >= 28 ? <> - Old.</> : <> - Young.</>}
9     </h2>
10  </div>
11 }
12
13 function App() {
14   return (
15     <div className="greetings">
16       <Person name="Jack" lastName="Sparrow" age={27}/>
17       <Person name="Bruce" lastName="Wayne" age={30}/>
18       <Person name="Peter" lastName="Parker" age={25}/>
19     </div>
20   );
21 }
22
23 export default App;
```

## 9. Umieszczenie listy osób w strukturze danych typu tablica oraz wypisanie:

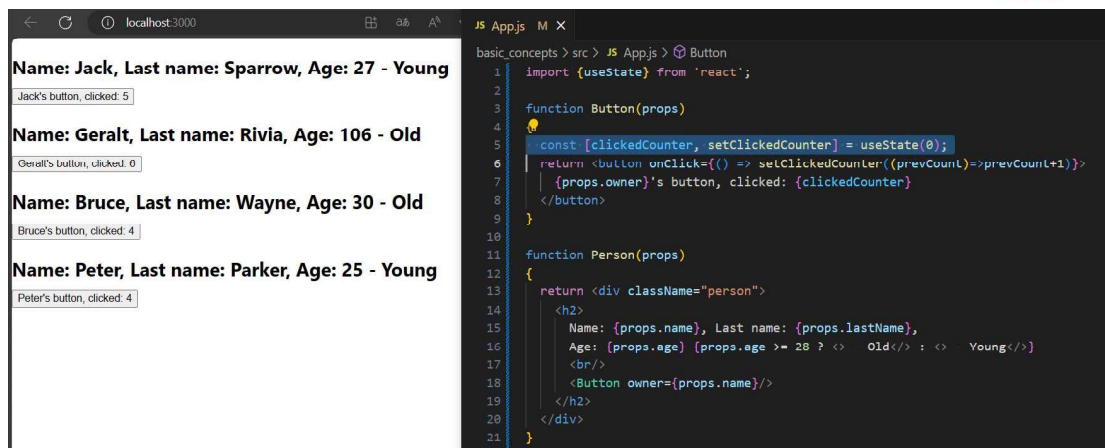
The screenshot shows a web browser at localhost:3000 displaying four lines of text: "Name: Jack, Last name: Sparrow, Age: 27 - Young", "Name: Geralt, Last name: Rivia, Age: 106 - Old", "Name: Bruce, Last name: Wayne, Age: 30 - Old", and "Name: Peter, Last name: Parker, Age: 25 - Young". The VS Code editor on the right shows the following code:

```
1 function Person(props)
2 {
3   return <div className="person">
4     <h2>
5       Name: {props.name}, Last name: {props.lastName},
6       Age: {props.age} {props.age >= 28 ? <> - Old.</> : <> - Young.</>}
7     </h2>
8   </div>
9 }
10
11 var people = [
12   {name: "Jack", lastName: "Sparrow", age: 27},
13   {name: "Geralt", lastName: "Rivia", age: 106},
14   {name: "Bruce", lastName: "Wayne", age: 30},
15   {name: "Peter", lastName: "Parker", age: 25},
16 ]
17
18 var listedPeople = people.map(person=>
19   <Person name={person.name} lastName={person.lastName} age={person.age}/>
20 )
21
22 function App() {
23   return (
24     <div className="greetings">
25       {listedPeople}
26     </div>
27   );
28 }
29 export default App;
```

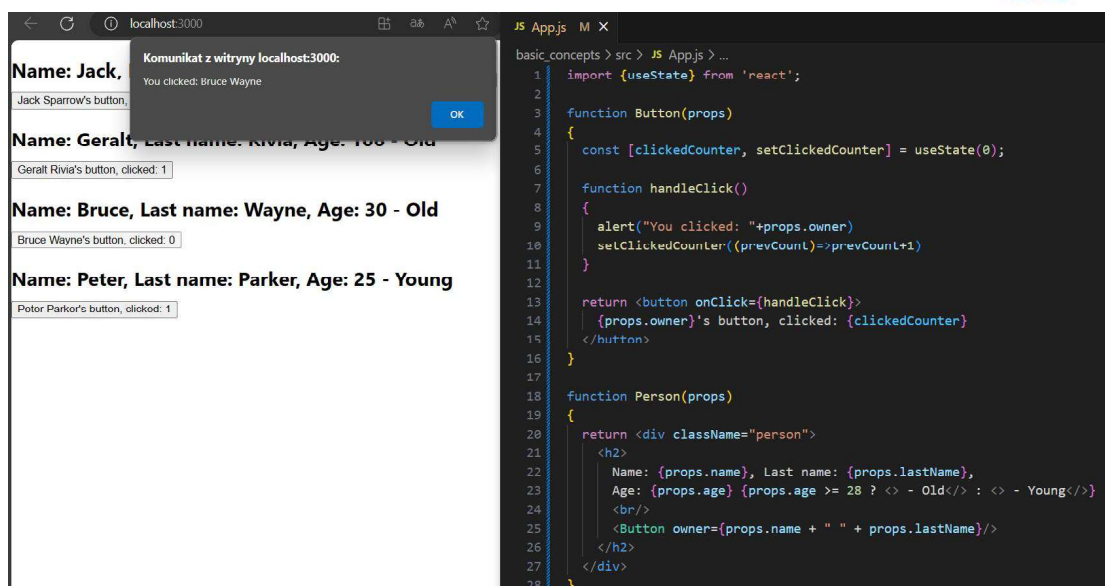
10. Dodanie komponentu „Button”, który ma jednego właściciela „Person”:



11. Dodanie ilości kliknięć przy użyciu „React Hooks” ~ **useState**:

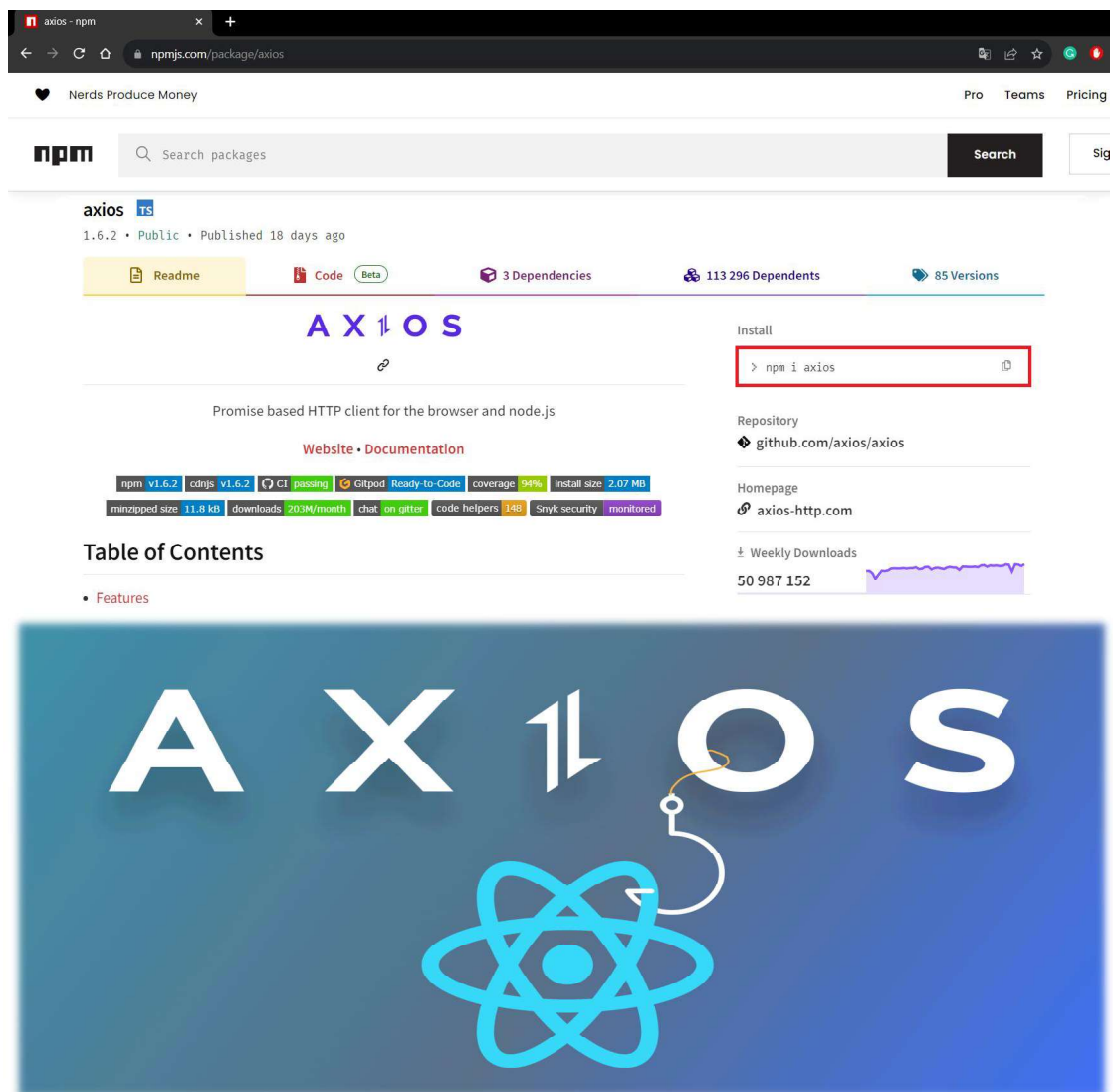


12. Dodanie dodatkowego alertu kiedy użytkownik wcisnie przycisk:





### 13. Pobranie pakietu axios do łączenia z REST API:



The screenshot shows the npm package page for 'axios'. The page includes the following information:

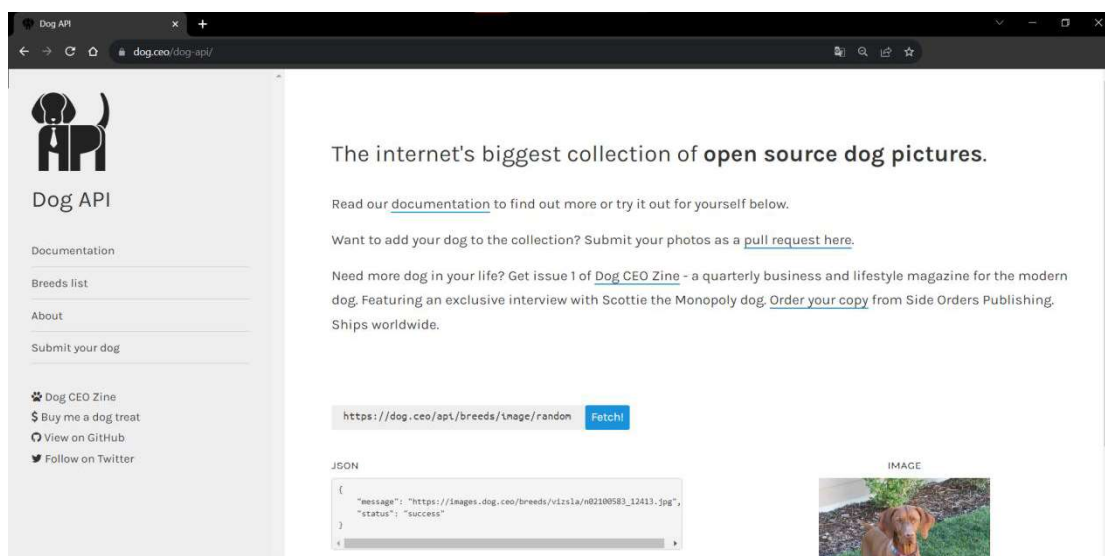
- Package Name:** axios
- Version:** 1.6.2
- Published:** 18 days ago
- Dependencies:** 3
- Dependents:** 113,296
- Versions:** 85

The main content area features the Axios logo and the text 'Promise based HTTP client for the browser and node.js'. Below this, there are links to the 'Website' and 'Documentation'. A 'Table of Contents' section is also visible, with 'Features' listed.

On the right side, there is an 'Install' section with a red box highlighting the command: `> npm i axios`. Below this, the 'Repository' is listed as 'github.com/axios/axios' and the 'Homepage' as 'axios-http.com'. A 'Weekly Downloads' graph shows 50,987,152 downloads.

At the bottom, there is a large blue banner with the Axios logo and the text 'AXIOS'.

### 14. Używanie REST API do obsługi obrazków psów (nie wymaga API KEY):



The screenshot shows the Dog API website. The page includes the following information:

- Logo:** Dog API
- Navigation:** Documentation, Breeds list, About, Submit your dog, Dog CEO Zine, Buy me a dog treat, View on GitHub, Follow on Twitter
- Text:** The internet's biggest collection of open source dog pictures.
- Links:** Read our [documentation](#) to find out more or try it out for yourself below. Want to add your dog to the collection? Submit your photos as a [pull request here](#).
- Text:** Need more dog in your life? Get issue 1 of [Dog CEO Zine](#) - a quarterly business and lifestyle magazine for the modern dog. Featuring an exclusive interview with Scottie the Monopoly dog. [Order your copy](#) from Side Orders Publishing. Ships worldwide.
- API Call:** `https://dog.ceo/api/breeds/image/random` with a 'Fetch!' button.
- JSON Response:**

```
{  "message": "https://images.dog.ceo/breeds/vizsla/n02100583_12413.jpg",  "status": "success"}
```
- Image:** A photo of a dog, labeled 'IMAGE'.



## 15. Testowanie REST API przy użyciu skryptu w Python'ie:

```
1 import requests
2
3 url = "https://dog.ceo/api/breeds/image/random"
4
5 dogs_json = []
6
7 for i in range(10):
8     dogs_json.append(requests.get(url).json())
9
10 print(dogs_json)
11
12 url = "https://dog.ceo/api/breeds/image/random/32"
13
14 dogs_json = requests.get(url).json()
15
16 print(dogs_json)
```

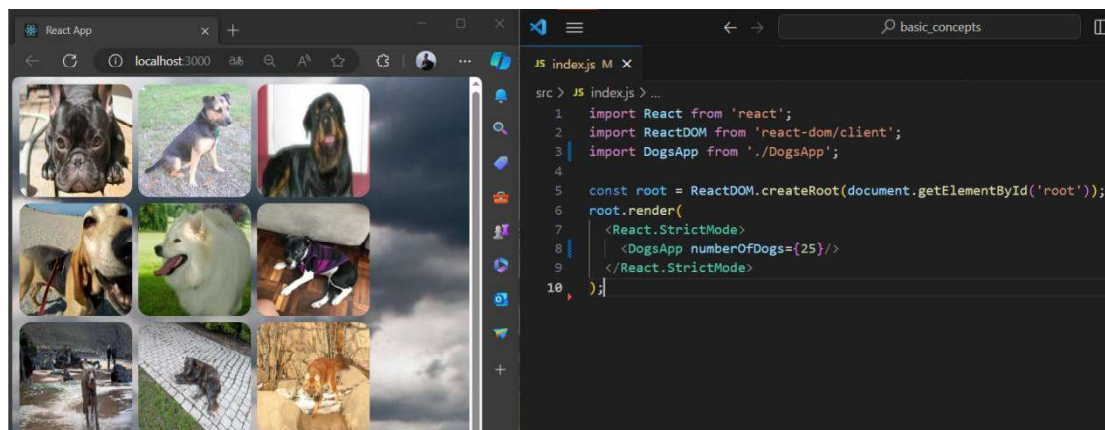
Terminal output:

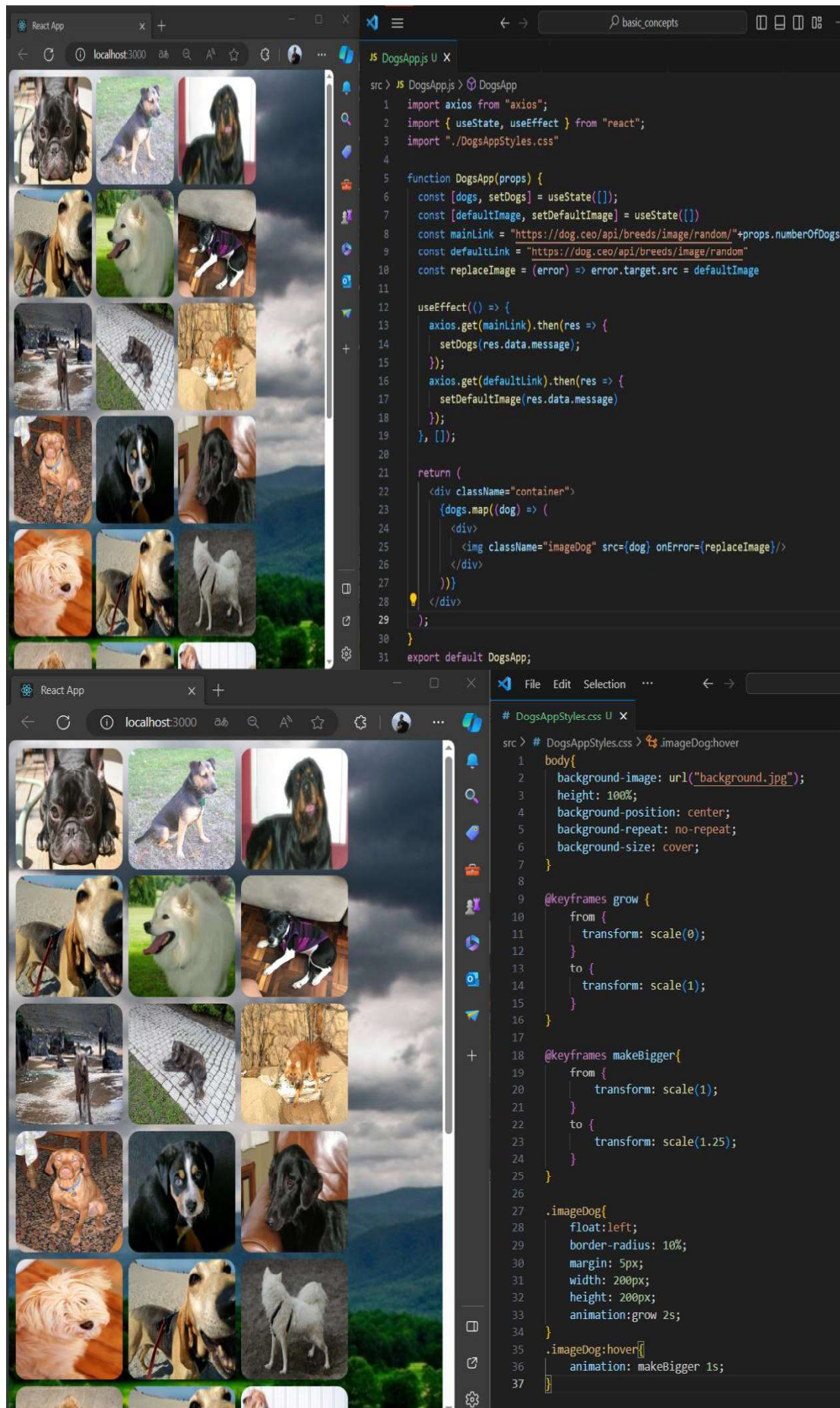
```
PS C:\Users\PanCh\Desktop\Aysystem - Pracownik - 5 Semestr Informatyka - AIIM\Lab09\basic_concepts> py .\test_ap
[{"message": "https://images.dog.ceo/breeds/leonberg/n02111129_844.jpg", "status": "success"}, {"message": "http
06382_154.jpg", "status": "success"}, {"message": "https://images.dog.ceo/breeds/poodle-toy/n02113624_8951.jpg",
tps://images.dog.ceo/breeds/germanshepherd/n02106662_22764.jpg", "status": "success"}, {"message": "https://imag
095889_2138.jpg", "status": "success"}, {"message": "https://images.dog.ceo/breeds/terrier-norfolk/n02094114_215
e": "https://images.dog.ceo/breeds/gaddi-indian/Gaddi.jpg", "status": "success"}, {"message": "https://images.do
00100_BURST20191103202017556_COVER.jpg", "status": "success"}, {"message": "https://images.dog.ceo/breeds/terrie
": "success"}, {"message": "https://images.dog.ceo/breeds/setter-irish/n02100877_5229.jpg", "status": "success"}]
{"message": ["https://images.dog.ceo/breeds/mix/Milka1.jpg", "https://images.dog.ceo/breeds/mudhol-indian/Indian
udhol.jpg", "https://images.dog.ceo/breeds/chow/n02112137_16815.jpg", "https://images.dog.ceo/breeds/cotondetule
/IMG_20160830_202631573.jpg", "https://images.dog.ceo/breeds/hound-blood/n02088466_7015.jpg", "https://images.do
ceo/breeds/airedale/n02096051_8892.jpg", "https://images.dog.ceo/breeds/appenzeller/n02107908_5002.jpg", "https
images.dog.ceo/breeds/terrier-sealyham/n02095889_2187.jpg", "https://images.dog.ceo/breeds/bluetick/n02088632_14
.jpg", "https://images.dog.ceo/breeds/hound-walker/n02089867_785.jpg", "https://images.dog.ceo/breeds/spitz-japa
```



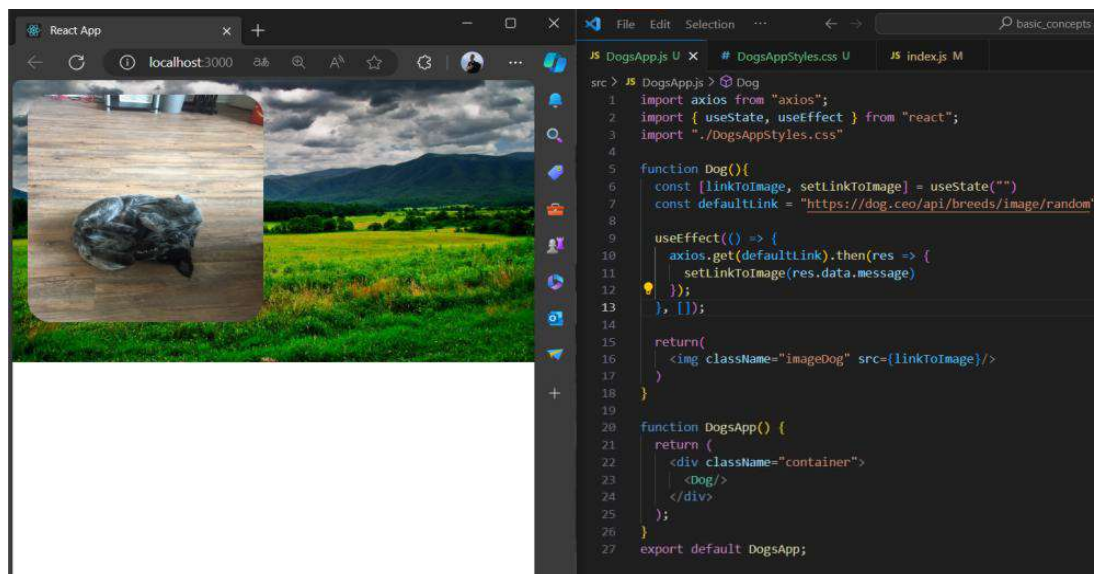
## 16. Utworzenie komponentu „DogApps”:

```
1 import axios from "axios";
2 import { useState, useEffect } from "react";
3 import "./DogsAppStyles.css"
4
5 function DogsApp(props) {
6     const [dogs, setDogs] = useState([]);
7     const [defaultImage, setDefaultImage] = useState('')
8     const mainLink = "https://dog.ceo/api/breeds/image/random/" + props.numberOfDogs
9     const defaultLink = "https://dog.ceo/api/breeds/image/random"
10    const replaceImage = (error) => error.target.src = defaultImage
11
12    useEffect(() => {
13        axios.get(mainLink).then(res => {
14            setDogs(res.data.message);
15        });
16        axios.get(defaultLink).then(res => {
17            setDefaultImage(res.data.message)
18        });
19    }, []);
20
21    return (
22        <div className="container">
23            {dogs.map((dog) => (
24                <div>
25                    <img className="imageDog" src={dog} onError={replaceImage}/>
26                </div>
27            ))}
28        </div>
29    );
30
31    export default DogsApp;
```

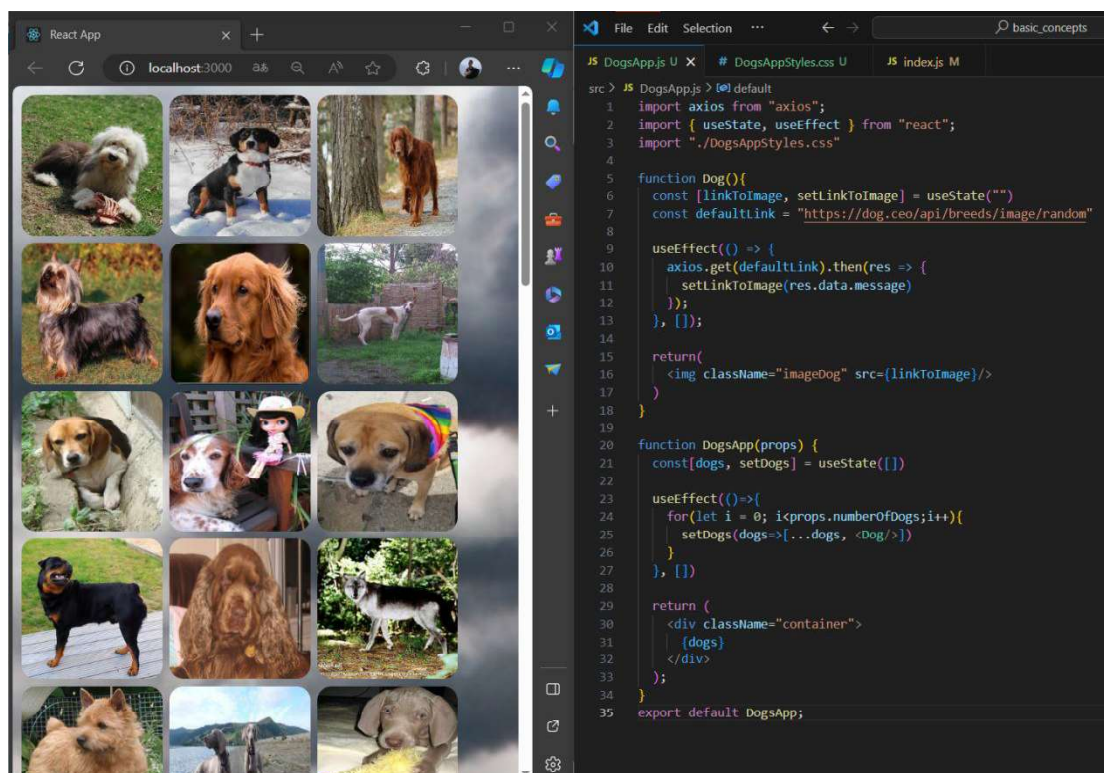








17. Utworzenie komponentu „DogApps” w oparciu o komponent „DogApp”:



### Do zrobienia:

Utworzenie aplikacji, która korzysta z dowolnego REST API (darmowe, bez API KEY), a następnie wyświetla je na ekranie zgodnie z praktyką. Podział na komponenty (izolacja) oraz czytelny kod.



### Sprawozdanie

W sprawozdaniu w systemie Sprawer wyślij link do aplikacji React z wybranym REST API opisanej w punkcie Do zrobienia oraz utworzone pliki z własnymi kodami źródłowymi (uwaga: wyłącznie własne kody).