

| | | |
|---------|-------------------------------------|----|
| 1 | Komponenty | 3 |
| 1.1 | Pojęcie..... | 3 |
| 1.2 | Rodzaje | 3 |
| 1.2.1 | Komponenty funkcyjne | 3 |
| 1.2.2 | Komponenty klasowe | 3 |
| 1.3 | Tworzenie komponentów | 3 |
| 1.4 | Użycie komponentów | 3 |
| 1.5 | Kolejność przetwarzania | 4 |
| 2 | Komponenty funkcyjne wersja 1 | 4 |
| 2.1 | Wygląd strony | 4 |
| 2.2 | Pliki do zmiany | 4 |
| 2.2.1 | ./src/Book.js | 4 |
| 2.2.2 | ./src/index.js | 5 |
| 3 | Komponenty klasowe | 5 |
| 3.1 | Wersja 02 | 5 |
| 3.1.1 | Wygląd strony | 5 |
| 3.1.2 | Plik ./src/Book.js | 5 |
| 3.1.3 | funkcja super() | 6 |
| 3.1.3.1 | słowo kluczowe super() w js | 6 |
| 3.1.3.2 | funkcja super() w React | 6 |
| 3.1.4 | prpos w React | 6 |
| 3.2 | wersja 3..... | 6 |
| 3.2.1 | Wygląd strony | 6 |
| 3.2.2 | Plik ./Book.js | 6 |
| 3.3 | Wersja 4 | 7 |
| 3.3.1 | Wygląd strony | 7 |
| 3.3.2 | Pliki do zmiany/utworzenia | 7 |
| 3.3.2.1 | Plik ./Header | 7 |
| 3.3.2.2 | ./Footer.js | 7 |
| 3.3.2.3 | ./Book.js | 8 |
| 3.4 | Wersja 5 | 8 |
| 3.4.1 | Wygląd strony – bez zmian | 8 |
| 3.4.2 | Pliki do zmiany / utworzenia | 8 |
| 3.4.2.1 | ./src/Footer.js | 8 |
| 3.4.2.2 | ./src/Data.js | 9 |
| 3.4.2.3 | ./src/BookOpis.js | 9 |
| 3.4.2.4 | Plik ./src/Book.js | 9 |
| 3.5 | Wersja 5 | 10 |
| 3.5.1 | Wygląd strony | 10 |

| | | |
|---------|---------------------------------------|----|
| 3.5.2 | Pliki do zmiany | 10 |
| 3.5.2.1 | ./src/BookOpis.js..... | 10 |
| 3.5.2.2 | Plik ./src/App.css..... | 10 |
| 3.5.2.3 | ./src/Book.js..... | 11 |
| 3.5.2.4 | Plik ./src/index.js..... | 11 |
| 3.5.2.5 | Plik ./src/App.js | 11 |
| 3.6 | Wersja 7 – organizacja katalogów..... | 12 |
| 3.6.1 | Nowa struktura katalogów..... | 12 |
| 3.6.2 | Pliki do zmiany | 12 |
| 3.6.2.1 | ./src/App.js..... | 12 |
| 3.7 | ./src/biblioteka/Booj.js | 12 |

1 Komponenty

1.1 Pojęcie

Koncepcyjnie, komponenty są jak javascriptowe funkcje:

- Przyjmują wartości na wejściu (nazywane "właściwościami" (ang. *props*))
- zwracają reactowe elementy opisujące, co powinno się pojawić na ekranie.

Komponenty pozwalają podzielić interfejs użytkownika na niezależne, pozwalające na ponowne użycie części i myśleć o każdej z nich osobno.

1.2 Rodzaje

1.2.1 Komponenty funkcyjne

- Funkcje napisane w js np.:

```
function Welcome(props) {  
  return <h1>Cześć, {props.name}</h1>;  
}
```

- Ta funkcja jest poprawnym reactowym komponentem funkcyjnym przyjmuje pojedynczy argument "props" (który oznacza "właściwości", z ang. *properties*), będący obiektem z danymi, zwraca reactowy element.

1.2.2 Komponenty klasowe

- Klasy dziedziczące po klasie React.Component

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Cześć, {this.props.name}</h1>;  
  }  
}
```

- Ta klasa jest poprawnym reactowym komponentem klasowym rozszerza klasę React.Component ma dostęp do właściwości props swojej klasy zwraca reactowy element

1.3 Tworzenie komponentów

- Komponenty zdefiniowane przez użytkownika muszą być pisane wielką literą
- Jeżeli nazwa elementu zaczyna się od małej litery, oznacza to odniesienie do wbudowanego komponentu, takiego jak na przykład: <div> lub , i skutkuje przekazaniem ciągu znaków 'div' lub 'span' do funkcji React.createElement.
- Nazwy typów, które zaczynają się od wielkiej litery, na przykład: <Foo />, są kompilowane do wywołania React.createElement(Foo) i odnoszą się do komponentów zdefiniowanych lub zaimportowanych do plików zawierających kod JavaScript tworzonych przez programistę

1.4 Użycie komponentów

- Komponenty funkcyjne: powinny być najczęściej stosowane by przy budowaniu aplikacji korzystać z prostych komponentów funkcyjnych, odnoszących się do pojedynczych elementów.
- Komponenty klasowe powinny być używane do tych części aplikacji, które służą do przechowywania danych lub zawierają logikę biznesową
- Komponenty przy zwracaniu wyniku mogą odwoływać się do innych komponentów. Pozwala to używać tej samej abstrakcji komponentu na dowolnym poziomie szczegółowości. Przycisk, formularz, okno dialogowe, ekran - w aplikacjach reactowych tego typu składniki są zwykle reprezentowane przez dedykowane komponenty.

1.5 Kolejność przetwarzania

```
function Welcome(props) {
  return <h1>Cześć, {props.name}</h1>;
}

const root = ReactDOM.createRoot(document.getElementById('root'));
const element = <Welcome name="Sara" />;
root.render(element);
```

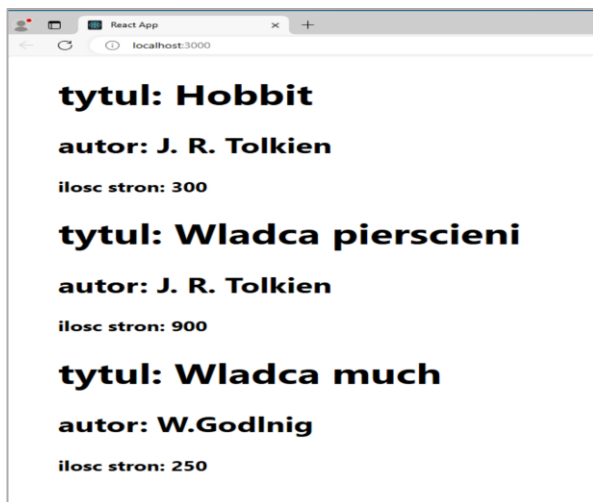
Renderowanie napisu „Cześć, Sara”

1. wywołujemy metodę `root.render()` z elementem `<Welcome name="Sara" />`.
2. React wywołuje komponent `Welcome` z właściwością / propsem `{name: 'Sara'}`.
3. Nasz komponent `Welcome` jako wynik zwraca element `<h1>Cześć, Sara</h1>`.
4. React DOM w optymalny sposób aktualizuje drzewo DOM, aby odpowiadało elementowi `<h1>Cześć, Sara</h1>`.

2 Komponenty funkcyjne wersja 1

Komponenty funkcyjne to funkcje

2.1 Wygląd strony



2.2 Pliki do zmiany

2.2.1 ./src/Book.js

```
1 const book = [
2   {id: 0, title: "Hobbit",author: "J. R. Tolkien",pages: "300"},
3   {id: 1, title: "Władca pierścieni", author: "J. R. Tolkien",pages: "900"},
4   {id: 2, title: "Władca much", author: "W.Godlnig",pages: "250"}
5 ];
6
7 function Book(){
8   return (
9     <div className="App">
10      {book.map((item,index)=>(
11        <ol>
12          <h1>tytul: {item.title}</h1>
13          <h2>autor: {item.author}</h2>
14          <h4>ilosc stron: {item.pages} </h4>
15        </ol>
16      ))}
17    </div>
18  )
19 };
20
21 export default Book;
```

Podobnie jak wcześniej definicja funkcji i eksport żeby można było skorzystać, równie dobrze można by to wpisać w App.js i efekt byłby ten sam

- Linie 1 – 5: definicja zmiennej book, która jest 3 elementową tablicą rekordów o 3 polach
- Linie 7 – 19: funkcja Book(), która zwraca przygotowany do wyświetlenia komponent opisujący książkę
- Linie 10 – 16: wypisanie zawartości tablicy
- Linie 11 – 15: musi być jakiś wyliczający element html-a (lista, tabela) bo inaczej map nie chce działać, przy kompilacji pojawiają się błędy
- Linia 21: eksport funkcji Book(), tak żeby można ją było gdzieś zaimportować

2.2.2 ./src/index.js

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.css';
4 import Book from './Book';
5 import reportWebVitals from './reportWebVitals';
6
7 const root = ReactDOM.createRoot(document.getElementById('root'));
8 root.render(
9   <React.StrictMode>
10     <Book />
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
```

- Linie 1 – 5: importy wszystko poza linią 4 to to co było w pliku
- Linia 4 - import funkcji Book z pliku Book, standardowo jest tu App
- Linia 10 – uruchomienie funkcji Book

3 Komponenty klasowe

3.1 Wersja 02

3.1.1 Wygląd strony



3.1.2 Plik ./src/Book.js

```
1 import React, { Component } from "react";
2 class Book extends Component {
3   constructor(props) {
4     super(props);
5     this.state = {
6       book: {id: 0, title: "Hobbit",author: "J. R. Tolkien",pages: "300"}
7     };
8   }
9   render(){
10     return(
11       <div className="App">
12         <h1>tytul: {this.state.book.title}</h1>
13         <h2>autor: {this.state.book.author}</h2>
14         <h4>ilosc stron {this.state.book.pages}</h4>
15       </div>
16     );
17   }
18 }
19 export default Book;
```

- Linia 1: import modułów Reacta w tym klasy Component
- Linie 2 – 8: definicja klasy komponentowej Book, dziedziczącej po klasie Component
- Linia 3: definicja konstruktora, przyjmujący jako parametr zmienną props,
- linia 4: konstruktor za pomocą polecenia super, wywołuje konstruktor klasy bazowej
- linie 5 – 6: definicja stanu komponentu: dane opisujące aktualny stan obiektu (np. dane interfejsu użytkownika: otwarte / zamknięte okna, przyciski, dane pobrane z bazy danych)
- linie 9 – 17: funkcja render() generująca kod JSX (JavaScript XML – format zapisu kodu HTML i XML stosowany w JavaScript i bibliotece React), który zawsze powinien posiadać jeden znacznik nadrzędny; tu tekst składa się z nagłówków h1, h2 i h4, dlatego został dodany dodatkowy nadrzędny znacznik div
- linie 12, 13, 14 – odwołania do wartości klasy zdefiniowanej w liniach 3 - 7

3.1.3 funkcja super()

3.1.3.1 słowo kluczowe super() w js

Słowo kluczowe super w JavaScript jest używane do uzyskiwania dostępu do właściwości literału obiektu lub klasy lub do wywoływania konstruktora bazowej

3.1.3.2 funkcja super() w React

- wywołuje konstruktor klasy nadrzędnej, używany go, gdy potrzebny jest dostęp do zmiennych w klasie bazowej.
- Zwraca obiekt reprezentujący klasę nadrzędną.
- Właściwym sposobem użycia jest sytuacja, gdy odnosi się do zmiennej instancji bezpośredniej klasy nadrzędnej

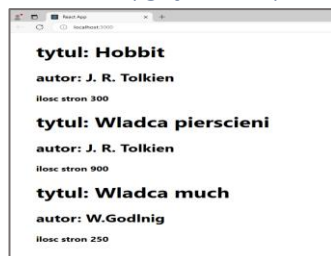
3.1.4 prpos w React

props (nazywane “właściwościami” (ang. *props od properties*)), to argument komponentu:

- pomagający w sterowaniu
- umożliwiającą komunikację między różnymi częściami aplikacji
- umożliwiającą przekazanie parametrów do komponentów

3.2 wersja 3

3.2.1 Wygląd strony



3.2.2 Plik ./Book.js

```

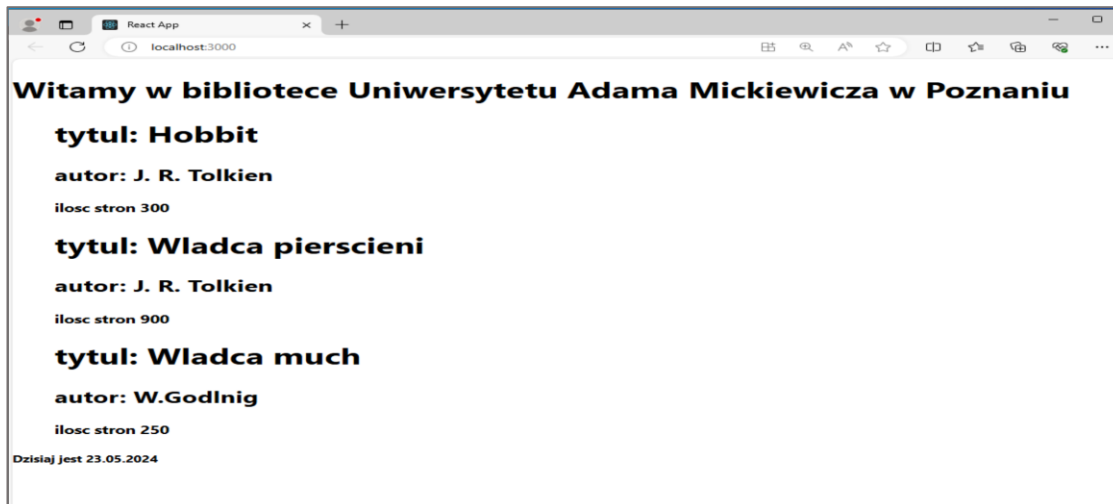
1 import React, { Component } from "react";
2 class Book extends Component {
3   constructor(props) {
4     super(props);
5     this.state = {
6       book: [
7         {id: 0, title: "Hobbit",author: "J. R. Tolkien",pages: "300"},
8         {id: 1, title: "Władca pierścieni", author: "J. R. Tolkien",pages: "900"},
9         {id: 2, title: "Władca much", author: "W.Godlnig",pages: "250"}
10      ]
11    };
12  }
13  render(){
14    return(
15      <div className="App">
16        {this.state.book.map((item, index)=>(
17          <ol>
18            <h1>tytuł: {item.title}</h1>
19            <h2>autor: {item.author}</h2>
20            <h4>ilosc stron {item.pages}</h4>
21          </ol>
22        ))}
23      </div>
24    );
25  }
26 }
27 export default Book;

```

- Linie 6 – 10: stan początkowy obiektu jest ustawiony jako 3 elementowa tablica składająca się z rekordów
- Linie 16 – 22: uruchomienie metody map na obiekcie zdefiniowanym w liniach 6 – 10
- Linie 17 i 23: deklaracja listy, bez deklaracji jakiegoś elementu strony internetowej umożliwiającego wyliczanie map nie działa
- Linie 18 – 20: odwołania do wartości pól rekordów, będących elementami kolejnych wierszy zwracanych przez metodę map

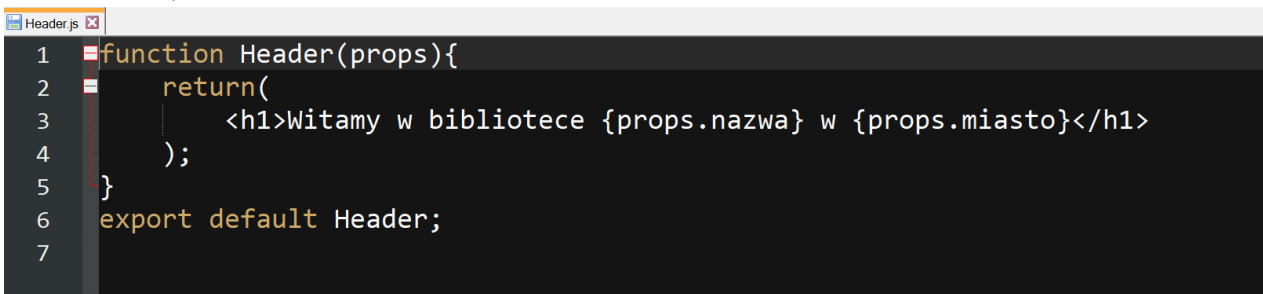
3.3 Wersja 4

3.3.1 Wygląd strony



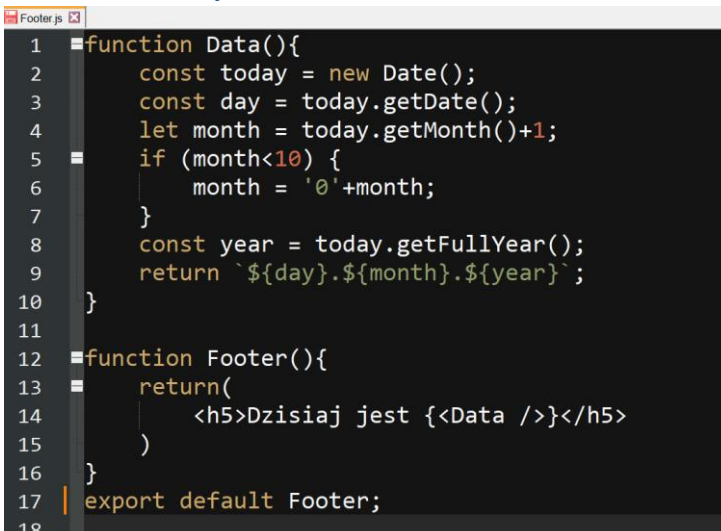
3.3.2 Pliki do zmiany/utworzenia

3.3.2.1 Plik ./Header



- Definicja nagłówka strony internetowej
- Linia 3 – wygenerowanie napisu z wartością parametrów przekazanych przez props

3.3.2.2 ./Footer.js



- Definicja stopki strony internetowej

- Linie 1 – 10: funkcja zwracająca aktualną datę
- Linie 1, 2 i 8: const oznacza że wartość zmiennej nie będzie się zmieniać
- Linia 4: month musi mieć deklarację let bo dla miesięcy od stycznia do września przed cyfrą będzie dopisywane 0 linie 5 – 7
- Linia 9: zwrócenie łańcucha z datą, muszą być \$ bo inaczej nie traktuje napisów w {} jako nazw zmiennych oraz całość jest w ciapach pod tyldą, inaczej jest to zwykły napis
- Linie 12 – 16: definicja funkcji Footer
- Linia 14 – uruchomienie funkcji Data

3.3.2.3 ./Book.js

```

1 import Header from "../Header";
2 import Footer from "../Footer";
3 import React, { Component } from "react";
4 class Book extends Component {
5   constructor(props) {
6     super(props);
7     this.state = {
8       book: [
9         {id: 0, title: "Hobbit",author: "J. R. Tolkien",pages: "300"},
10        {id: 1, title: "Władca pierścieni", author: "J. R. Tolkien",pages: "900"},
11        {id: 2, title: "Władca much", author: "W.Godlnig",pages: "250"}
12      ]
13    };
14  }
15  render(){
16    return(
17      <div className="App">
18        <Header nazwa='Uniwersytetu Adama Mickiewicza' miasto="Poznaniu" />
19        {this.state.book.map((item, index)=>(
20          <ol>
21            <h1>tytuł: {item.title}</h1>
22            <h2>autor: {item.author}</h2>
23            <h4>ilosc stron {item.pages}</h4>
24          </ol>
25        ))}
26        <Footer />
27      </div>
28    );
29  }
30 }
31 export default Book;

```

- Linia 1: import funkcji Header z pliku Header w katalogu bieżącym
- Linia 2: import funkcji Footer z pliku Footer w katalogu bieżącym
- Linia 18: wywołanie funkcji Header z parametrami / props-ami nazwa i miasto
- Linia 26: wywołanie funkcji Footer

3.4 Wersja 5

3.4.1 Wygląd strony – bez zmian

3.4.2 Pliki do zmiany / utworzenia

3.4.2.1 ./src/Footer.js

```

1 import Data from "../Data";
2
3 function Footer(){
4   return(
5     <h5>Dzisiaj jest {Data} </h5>
6   )
7 }
8 export default Footer;

```

- Linia 1: usunąć funkcję generującą datę, zamiast tego zaimportować komponent z pliku Data
- Linia 5: wywołanie funkcji nie ulega zmianie

3.4.2.2 ./src/Data.js

```
1 function Data(){
2   const today = new Date();
3   const day = today.getDate();
4   let month = today.getMonth()+1;
5   if (month<10) {
6     month = '0'+month;
7   }
8   const year = today.getFullYear();
9   return `${day}.${month}.${year}`;
10 }
11
12 export default Data;
```

- Przenieść zawartość funkcji Data z pliku Footer.js do pliku Data.js
- Linia 12 – wyeksportować funkcję

3.4.2.3 ./src/BookOpis.js

```
1 import React, { Component } from "react";
2 class BookOpis extends Component {
3   constructor(props) {
4     super(props);
5     this.state = {
6       book: [
7         {id: 0, title: "Hobbit",author: "J. R. Tolkien",pages: "300"},
8         {id: 1, title: "Władca pierścieni", author: "J. R. Tolkien",pages: "900"},
9         {id: 2, title: "Władca much", author: "W.Godlnig",pages: "250"}
10      ]
11    };
12  }
13  render(){
14    return(
15      <div className="App">
16        {this.state.book.map((item,indx)=>(
17          <ol>
18            <h1>tytul: {item.title}</h1>
19            <h2>autor: {item.author}</h2>
20            <h4>ilosc stron {item.pages}</h4>
21          </ol>
22        ))}
23      </div>
24    );
25  }
26 }
27 export default BookOpis;
```

- Linie 1 – 12: skopiować z pliku ./src/Book.js import Reacta i definicję klasy Book zmienić jej nazwę na BookOpis
- Linie 12 – 25: w funkcji return usunąć odwołania do funkcji Footer (było po linii 23) i Header (było po linii 15)
- Linia 27 – eksport klasy BookOpis

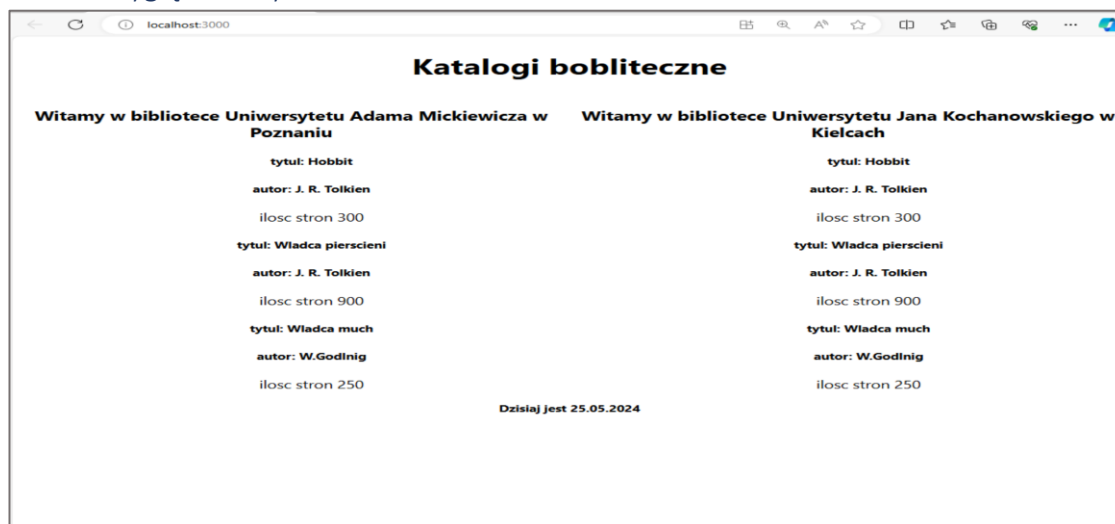
3.4.2.4 Plik ./src/Book.js

```
1 import React, { Component } from "react";
2 import Header from "./Header";
3 import Footer from "./Footer";
4 import BookOpis from "./BookOpis";
5
6 class Book extends Component{
7   render(){
8     return(
9       <div>
10         <Header nazwa='Uniwersytetu Adama Mickiewicza' miasto="Poznaniu" />
11         <BookOpis/>
12         <Footer/>
13       </div>
14     )
15   }
16 }
17
18 export default Book;
```

- Linie 2 – 4: zaimportować komponenty do opisu książki
- Linie 10 – 12: uruchomić komponenty opisujące książkę

3.5 Wersja 5

3.5.1 Wygląd strony



3.5.2 Pliki do zmiany

3.5.2.1 ./src/BookOpis.js

```
1 import React, { Component } from "react";
2 class BookOpis extends Component {
3   constructor(props) {
4     super(props);
5     this.state = {
6       book: [
7         {id: 0, title: "Hobbit",author: "J. R. Tolkien",pages: "300"},
8         {id: 1, title: "Władca pierścieni", author: "J. R. Tolkien",pages: "900"},
9         {id: 2, title: "Władca much", author: "W.Godlnig",pages: "250"}
10      ]
11    };
12  }
13  render(){
14    return(
15      <div className="App">
16        {this.state.book.map((item,inedx)=>(
17          <ol>
18            <h5>tytuł: {item.title}</h5>
19            <h5>autor: {item.author}</h5>
20            <p>ilosc stron {item.pages}</p>
21          </ol>
22        ))}
23      </div>
24    );
25  }
26 }
27 export default BookOpis;
```

- Linie 18 – 20: zmieniony poziom nagłówków

3.5.2.2 Plik ./src/App.css

```
1 .App {
2   text-align: center;
3 }
4
5 .Left{
6   float:left;
7   width: 50%;
8 }
9
10 .Right{
11   float: right;
12   width: 50%;
13 }
14
15 .Stopka{
16   clear: both;
17 }
18
```

- Usunięcie zbędnych deklaracji, .App gdzie jest wywołane to zostało
- Linie 5 – 17 – deklaracje klas do rozłożenia elementów na stronie

3.5.2.3 ./src/Book.js

```
1 import React, { Component } from "react";
2 import Header from "../Header";
3 import Footer from "../Footer";
4 import BookOpis from "../BookOpis";
5
6 class Book extends Component {
7   render() {
8     return (
9       <div>
10        <h1>Katalogi biblieczne</h1>
11        <div className="Left">
12          <Header nazwa='Uniwersytetu Adama Mickiewicza' miasto="Poznanu" />
13          <BookOpis />
14        </div>
15        <div className="Right">
16          <Header nazwa='Uniwersytetu Jana Kochanowskiego' miasto="Kielcach" />
17          <BookOpis />
18        </div>
19        <div className="Stopka">
20          <Footer />
21        </div>
22      </div>
23    );
24  }
25 }
26
27 export default Book;
```

- Linia 10 – dodany nagłówek h1
- Linie 11 – 14: dodany div z deklaracją klasy Left i usunięte wywołanie funkcji Footer
- Linie 15 – 18: dodany div z deklaracją klasy Right, zmiana wartości propsów w wywołaniu komponentu Header, usunięcie komponentu Footer
- Linie 19 – 21: dodanie div-a z deklaracją klasy Stopka i wstawienie komponentu Footer
- Linie 9 – 29: div, który pozwala na podpięcie wszystkiego do DOM-u, wszystko co jest renderowane musi zawierać się w znaczniku, który pozwala na wpięcie do DOM-u (div, head, foot)

3.5.2.4 Plik ./src/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
```

- Linia 4: zastąpienie importu komponentu Book importem komponentu App

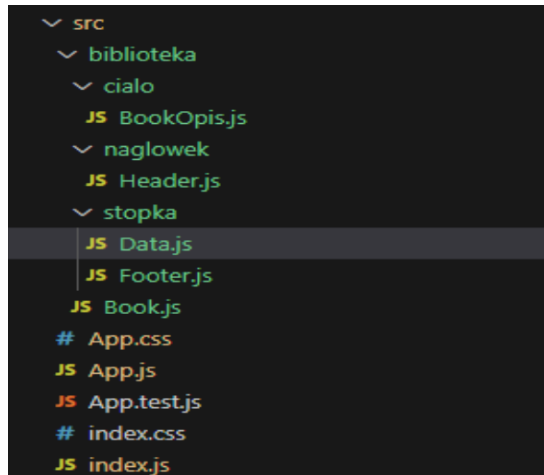
3.5.2.5 Plik ./src/App.js

```
1 import Book from './Book'
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <Book />
8     </div>
9   );
10 }
11
12 export default App;
```

- Usunięcie niepotrzebnej zawartości
- Linia 1 – import komponentu Book
- Linia 7 – uruchomienie funkcji Book

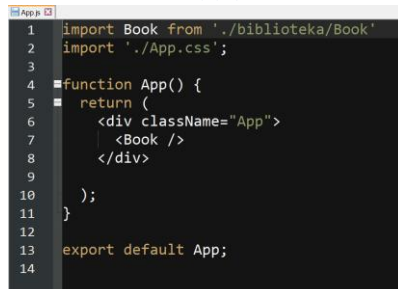
3.6 Wersja 7 – organizacja katalogów

3.6.1 Nowa struktura katalogów



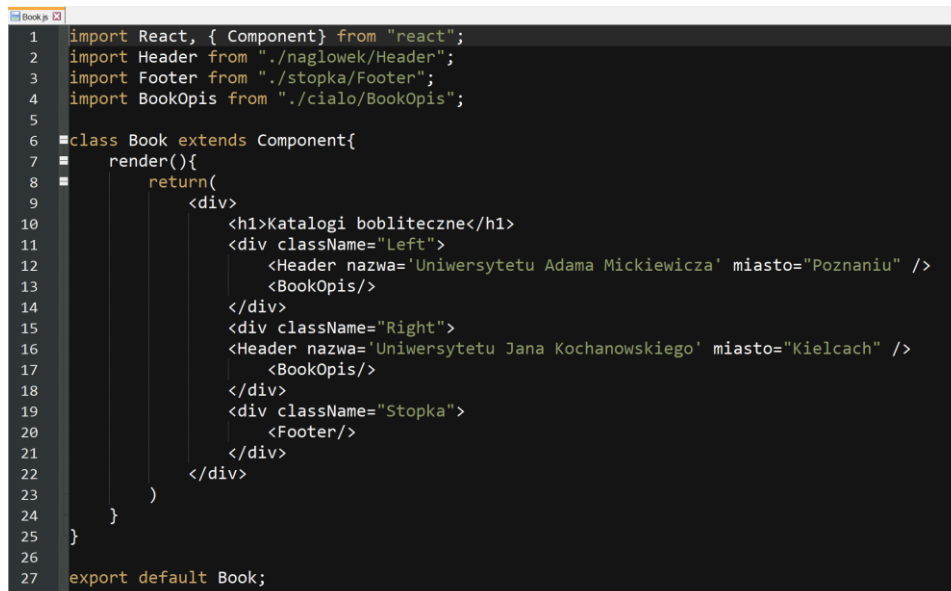
3.6.2 Pliki do zmiany

3.6.2.1 ./src/App.js



- Linia 1 – nowa ścieżka dostępu do pliku gromadzącego wszystkie komponenty tworzące bibliotekę
- Linia 2 – odwołanie do pliku z kaskadowymi arkuszami stylów jest tylko tutaj, więc format strony zostanie zachowany bez dalszych modyfikacji

3.7 ./src/biblioteka/Book.js



- Linie 2 – 4: odwołanie do komponentów Header, Footer, BookOpis