

FRONTEND

1. Pobieranie danych z bazy danych i wyświetlanie w formie tabel (do przeglądania):

a) Tabela z autorami

```
7 export default class AuthorsTable extends Component {
8   constructor(props) {
9     super(props);
10    this.state = {
11      authors: [],
12    };
13  }
14
15  componentWillMount() {
16    this.getDataFromApi();
17  }
18
19  getDataFromApi() {
20    return axios.get('http://ddanowskids.ddns.net:8080/person/list')
21      .then((response) => {
22        this.setState({
23          authors: response.data,
24        });
25      })
26      .catch(error => console.log(error.message));
27  }
28
29  render() {
30    return (
31      <BootstrapTable data={this.state.authors} search version='4' pagination>
32        <TableHeaderColumn isKey dataField='id' dataSort>ID</TableHeaderColumn>
33        <TableHeaderColumn dataField='surname' dataSort>Nazwisko</TableHeaderColumn>
34        <TableHeaderColumn dataField='name' dataSort>Imię</TableHeaderColumn>
35        <TableHeaderColumn dataField='name2' dataSort>Drugie imię</TableHeaderColumn>
36        <TableHeaderColumn dataField='orcid' dataSort>Numer ORCID</TableHeaderColumn>
37      </BootstrapTable>
38    );
39  }
40 }
41
```

b) Tabela z cytowaniami

```
7 export default class CitationsTable extends Component {
8   constructor(props) {
9     super(props);
10    this.state = {
11      citations: [],
12    };
13  }
14
15  componentWillMount() {
16    this.getDataFromApi();
17  }
18
19  getDataFromApi() {
20    return axios.get('http://ddanowskids.ddns.net:8080/citation/list')
21      .then((response) => {
22        let data = response.data;
23        for (let i = 0; i < data.length; i++) {
24          data[i].unitCitingTitle = data[i].unitCiting.title;
25          data[i].unitCitedTitle = data[i].unitCited.title;
26        }
27
28        this.setState({
29          citations: response.data,
30        });
31        console.log(response.data);
32      })
33      .catch(error => console.log(error.message));
34  }
35
36  render() {
37    return (
38      <BootstrapTable data={this.state.citations} search version='4' pagination>
39        <TableHeaderColumn isKey dataField='id' dataSort>ID</TableHeaderColumn>
40        <TableHeaderColumn dataField='originalText' dataSort>Cytowanie</TableHeaderColumn>
41        <TableHeaderColumn dataField='unitCitingTitle' dataSort>Jednostka cytująca</TableHeaderColumn>
42        <TableHeaderColumn dataField='unitCitedTitle' dataSort>Jednostka cytowana</TableHeaderColumn>
43        <TableHeaderColumn dataField='type' dataSort>Typ</TableHeaderColumn>
44      </BootstrapTable>
45    );
46  }
47 }
48
```

c) Tabela z wydawnictwami

```
7   export default class HousesTable extends Component {
8     constructor(props) {
9       super(props);
10      this.state = {
11        houses: [],
12      };
13    }
14
15    componentWillMount() {
16      this.getDataFromApi();
17    }
18
19    getDataFromApi() {
20      return axios.get('http://ddanowskids.ddns.net:8080/institution/list')
21        .then((response) => {
22          this.setState({
23            houses: response.data,
24          });
25        })
26        .catch(error => console.log(error.message));
27    }
28
29    render() {
30      return (
31        <BootstrapTable data={this.state.houses} search version='4' pagination>
32          <TableHeaderColumn isKey dataField='id' dataSort>ID</TableHeaderColumn>
33          <TableHeaderColumn dataField='name' dataSort>Nazwa wydawnictwa</TableHeaderColumn>
34        </BootstrapTable>
35      );
36    }
37  }
```

d) Tabela z publikacjami

```
7 export default class PublicationsTable extends Component {
8   constructor(props) {
9     super(props);
10    this.state = {
11      publications: [],
12    };
13  }
14
15  componentWillMount() {
16    this.getDataFromApi();
17  }
18
19  getDataFromApi() {
20    return axios.get('http://ddanowskids.ddns.net:8080/bib-unit/list')
21      .then((response) => {
22        console.log(response);
23        let data = response.data;
24        for (let i = 0; i < data.length; i++) {
25          data[i].publisherIdName = data[i].publisherId.name;
26          if (data[i].journalId != null) data[i].journalIdName = data[i].journalId.name; else data[i].journalIdName = '-';
27          data[i].institutionIdName = data[i].publisherId.institutionId.name;
28          if (data[i].vol == null) data[i].vol = '-';
29        }
30
31        this.setState({
32          publications: response.data,
33        });
34        console.log("działa", response.data);
35      })
36      .catch(error => console.log("nie działa", error.message));
37  }
38
39  render() {
40    return (
41      <BootstrapTable data={this.state.publications} search version="4" pagination>
42        <TableHeaderColumn isKey dataField="id" dataSort>ID</TableHeaderColumn>
43        <TableHeaderColumn dataField="title" dataSort>Tytuł</TableHeaderColumn>
44        <TableHeaderColumn dataField="bibType" dataSort>Typ</TableHeaderColumn>
45        <TableHeaderColumn dataField="publisherIdName" dataSort>Autor</TableHeaderColumn>
46        <TableHeaderColumn dataField="institutionIdName" dataSort>Wydawnictwo</TableHeaderColumn>
47        <TableHeaderColumn dataField="journalIdName" dataSort>Czasopismo</TableHeaderColumn>
48        <TableHeaderColumn dataField="year" dataSort>Rok wydania</TableHeaderColumn>
49        <TableHeaderColumn dataField="vol" dataSort>Wolumin</TableHeaderColumn>
50        <TableHeaderColumn dataField="issue" dataSort>Wydanie</TableHeaderColumn>
51        <TableHeaderColumn dataField="isbn" dataSort>ISBN</TableHeaderColumn>
52        <TableHeaderColumn dataField="doi" dataSort>DOI</TableHeaderColumn>
53      </BootstrapTable>
54    );
55  }
56 }
```

2. Dodawanie nowych rekordów:

a) Dodawanie autora

```
22 export default class AddAuthor extends React.Component {
23   constructor(props) {
24     super(props);
25     this.state = {
26       id: null,
27       surname: '',
28       name: '',
29       name2: '',
30       orcid: '',
31     };
32
33     this.updateId = this.updateId.bind(this);
34     this.updateSurname = this.updateSurname.bind(this);
35     this.updateName = this.updateName.bind(this);
36     this.updateName2 = this.updateName2.bind(this);
37     this.updateOrcid = this.updateOrcid.bind(this);
38     this.addAuthor = this.addAuthor.bind(this);
39   }
40
41   updateId(e) {
42     this.setState({
43       id: e.target.value,
44     });
45   }
46
47   updateSurname(e) {
48     this.setState({
49       surname: e.target.value,
50     });
51   }
52
53   updateName(e) {
54     this.setState({
55       name: e.target.value,
56     });
57   }
58
59   updateName2(e) {
60     this.setState({
61       name2: e.target.value,
62     });
63   }
64
65   updateOrcid(e) {
66     this.setState({
67       orcid: e.target.value,
68     });
69   }
```



```

71   addAuthor(e) {
72     e.preventDefault();
73
74     return axios.post('http://ddanowskids.ddns.net:8080/person/new', {
75       id: this.state.id,
76       surname: this.state.surname,
77       name: this.state.name,
78       name2: this.state.name2,
79       orcid: this.state.orcid,
80     })
81     .then((response) => {
82       window.location.replace('/#/authors');
83     })
84     .catch(error => console.log(error.message));
85   }
86
87
88   render() {
89     return (
90       <div className="animated fadeIn">
91         <br />
92         <h1>Autor <Badge color="secondary" style={styles.badgeStyle}>Nowy</Badge></h1>
93         <br />
94         <InputGroup style={styles.inputWidth}>
95           <Input placeholder="ID (id)" onChange={this.updateId} />
96         </InputGroup>
97         <br />
98         <InputGroup style={styles.inputWidth}>
99           <Input placeholder="Nazwisko (surname)" onChange={this.updateSurname} />
100        </InputGroup>
101        <br />
102        <InputGroup style={styles.inputWidth}>
103          <Input placeholder="Imię (name)" onChange={this.updateName} />
104        </InputGroup>
105        <br />
106        <InputGroup style={styles.inputWidth}>
107          <Input placeholder="Drugie imię (name2)" onChange={this.updateName2} />
108        </InputGroup>
109        <br />
110        <InputGroup style={styles.inputWidth}>
111          <Input placeholder="ORCID (orcid)" onChange={this.updateOrcid} />
112        </InputGroup>
113        <br />
114        <Button color="primary" style={styles.addButton} onClick={this.addAuthor}>Dodaj</Button>
115      </div>
116    );
117  }
118 }
119 }

```

b) Dodawanie cytowania

```
22 export default class AddCitation extends React.Component {
23   constructor(props) {
24     super(props);
25     this.state = {
26       id: null,
27       original: '',
28       citing: '',
29       cited: '',
30       type: '',
31     };
32
33     this.updateId = this.updateId.bind(this);
34     this.updateOriginal = this.updateOriginal.bind(this);
35     this.updateCiting = this.updateCiting.bind(this);
36     this.updateCiting = this.updateCited.bind(this);
37     this.updateType = this.updateType.bind(this);
38     this.addCitation = this.addCitation.bind(this);
39   }
40
41   updateId(e) {
42     this.setState({
43       id: e.target.value,
44     });
45   }
46
47   updateOriginal(e) {
48     this.setState({
49       original: e.target.value,
50     });
51   }
52
53   updateCiting(e) {
54     this.setState({
55       citing: e.target.value,
56     });
57   }
58
59   updateCited(e) {
60     this.setState({
61       cited: e.target.value,
62     });
63   }
64
65   updateType(e) {
66     this.setState({
67       type: e.target.value,
68     });
69   }
}
```

```

71 addCitation(e) {
72   e.preventDefault();
73   return axios.post('http://ddanowskids.ddns.net:8080/citation/new', {
74     id: this.state.id,
75     originalText: this.state.original,
76     type: this.state.type,
77     unitCited: {
78       id: this.state.id,
79       bibType: 'article',
80       publisherId: {
81         id: this.state.id+1,
82         name: "Ten tego",
83         institutionId: {
84           id: this.state.id,
85           name: "Wydawnictwo ABW"
86         }
87       },
88       journalId: {
89         id: this.state.id,
90         publisherId: {
91           id: this.state.id+1,
92           name: "Hugh Hefner",
93           institutionId: {
94             id: this.state.id,
95             name: "Wydawnictwo ABW"
96           }
97         },
98         name: "Playboy 2",
99         issn: "2451-1986"
100       },
101       isbn: "ISBN3",
102       doi: "DOI3",
103       title: "Stan Lee's view on databases",
104       ppFrom: 78,
105       ppTo: 82,
106       year: 2010,
107       vol: null,
108       issue: 1
109     },
110     unitCitedTitle: this.state.cited,
111     unitCiting: {
112       id: this.state.id,
113       bibType: 'book',
114       publisherId: {
115         id: 4,
116         name: 'name',
117         institutionId: {
118           id: 2,
119           name: 'Wydawnictwo CBS',

```



```

120     },
121   },
122   journalId: null,
123   isbn: 'ISBN4',
124   doi: 'DOI4',
125   title: 'Core Java. Volume I. Fundamentals',
126   ppFrom: null,
127   ppTo: null,
128   year: 2016,
129   vol: 1,
130 },
131 unitCitingTitle: this.state.citing,
132 ))
133 .then((response) => {
134   window.location.replace('/#/publications');
135 })
136 .catch(error => console.log(error.message));
137 }
138
139 render() {
140   return (
141     <div className="animated fadeIn">
142       <br />
143       <h1>Cytowanie <Badge color="secondary" style={styles.badgeStyle}>Nowe</Badge></h1>
144       <br />
145       <InputGroup style={styles.inputWidth}>
146         <Input placeholder="ID (id)" onChange={this.updateId} />
147       </InputGroup>
148       <br />
149       <InputGroup style={styles.inputWidth}>
150         <Input placeholder="Oryginalny tekst (originalText)" onChange={this.updateOriginal} />
151       </InputGroup>
152       <br />
153       <InputGroup style={styles.inputWidth}>
154         <Input placeholder="Jednostka cytująca (unitCiting.title)" onChange={this.updateCiting} />
155       </InputGroup>
156       <br />
157       <InputGroup style={styles.inputWidth}>
158         <Input placeholder="Jednostka cytowana (unitCited.title)" onChange={this.updateCited} />
159       </InputGroup>
160       <br />
161       <InputGroup style={styles.inputWidth}>
162         <Input placeholder="Typ (type)" onChange={this.updateType} />
163       </InputGroup>
164       <br />
165       <Button color="primary" style={styles.addButton} onClick={this.addCitation}>Dodaj</Button>
166     </div>
167   );
168 }
169 }
170 };

```

c) Dodawanie wydawnictwa

```
22 export default class AddHouse extends React.Component {
23   constructor(props) {
24     super(props);
25     this.state = {
26       id: null,
27       name: '',
28     };
29
30     this.updateId = this.updateId.bind(this);
31     this.updateName = this.updateName.bind(this);
32     this.addHouse = this.addHouse.bind(this);
33   }
34
35   updateId(e) {
36     this.setState({
37       id: e.target.value,
38     });
39   }
40
41   updateName(e) {
42     this.setState({
43       name: e.target.value,
44     });
45   }
46
47   addHouse(e) {
48     e.preventDefault();
49
50     return axios.post('http://ddanowskids.ddns.net:8080/institution/new', {
51       id: this.state.id,
52       name: this.state.name,
53     })
54       .then((response) => {
55         window.location.replace('/#/houses');
56       })
57       .catch(error => console.log(error.message));
58   }
59
60   render() {
61     return (
62       <div className="animated fadeIn">
63         <br />
64         <h1>Wydawnictwo <Badge color="secondary" style={styles.badgeStyle}>Nowe</Badge></h1>
65         <br />
66         <InputGroup style={styles.inputWidth}>
67           <Input placeholder="ID (id)" onChange={this.updateId} />
68         </InputGroup>
69         <br />
70         <InputGroup style={styles.inputWidth}>
71           <Input placeholder="Nazwa wydawnictwa (name)" onChange={this.updateName} />
72         </InputGroup>
73         <br />
74         <Button color="primary" style={styles.addButton} onClick={this.addHouse}>Dodaj</Button>{' '}
75       </div>
76     );
77   }
78 };
```

d) Dodawanie publikacji

```
25 export default class KindChoice extends React.Component {
26   constructor(props) {
27     super(props);
28     this.state = {
29       dropdownOpen: false,
30       checked: 0,
31       bibType: '',
32       doi: '',
33       institutionIdName: '',
34       isbn: '',
35       journalIdName: '',
36       publisherIdName: '',
37       title: '',
38       author: '',
39       id: undefined,
40       year: undefined,
41       vol: undefined,
42       issue: undefined,
43     };
44
45     this.toggle = this.toggle.bind(this);
46     this.checkKind = this.checkKind.bind(this);
47     this.updateBibType = this.updateBibType.bind(this);
48     this.updateDoi = this.updateDoi.bind(this);
49     this.updateInstitutionIdName = this.updateInstitutionIdName.bind(this);
50     this.updateIsbn = this.updateIsbn.bind(this);
51     this.updateJournalIdName = this.updateJournalIdName.bind(this);
52     this.updatePublisherIdName = this.updatePublisherIdName.bind(this);
53     this.updateTitle = this.updateTitle.bind(this);
54     this.updateAuthor = this.updateAuthor.bind(this);
55     this.updateId = this.updateId.bind(this);
56     this.updateYear = this.updateYear.bind(this);
57     this.updateVol = this.updateVol.bind(this);
58     this.updateIssue = this.updateIssue.bind(this);
59     this.addNewPublications = this.addNewPublications.bind(this);
60   }
61
62   addNewPublications(e) {
63     e.preventDefault();
64
65     return axios.post('http://ddanowskids.ddns.net:8080/bib-unit/new', {
66       id: this.state.id,
67       bibType: this.state.bibType,
68       publisherId: {
69         id: this.state.id,
70         name: this.state.publisherIdName,
71         institutionId: {
72           id: this.state.id + 2,
73           name: this.state.institutionIdName,
```

```

74     },
75     },
76     journalId: null,
77     isbn: this.state.isbn,
78     doi: this.state.doi,
79     title: this.state.title,
80     ppFrom: null,
81     ppTo: null,
82     year: this.state.year,
83     vol: this.state.vol,
84     issue: this.state.issue,
85   })
86   .then((response) => {
87     window.location.replace('/#/publications');
88   })
89   .catch(error => console.log(error.message));
90 }
91
92 updateBibType(e) {
93   this.setState({
94     bibType: e.target.value,
95   });
96 }
97 updateDoi(e) {
98   this.setState({
99     doi: e.target.value,
100   });
101 }
102 updateInstitutionIdName(e) {
103   this.setState({
104     institutionIdName: e.target.value,
105   });
106 }
107 updateIsbn(e) {
108   this.setState({
109     isbn: e.target.value,
110   });
111 }
112 updateJournalIdName(e) {
113   this.setState({
114     journalIdName: e.target.value,
115   });
116 }
117 updatePublisherIdName(e) {
118   this.setState({
119     publisherIdName: e.target.value,
120   });
121 }

```



```
122     updateTitle(e) {
123       this.setState({
124         title: e.target.value,
125       });
126     }
127     updateAuthor(e) {
128       this.setState({
129         author: e.target.value,
130       });
131     }
132     updateId(e) {
133       this.setState({
134         id: e.target.value,
135       });
136     }
137     updateYear(e) {
138       this.setState({
139         year: e.target.value,
140       });
141     }
142     updateVol(e) {
143       this.setState({
144         vol: e.target.value,
145       });
146     }
147     updateIssue(e) {
148       this.setState({
149         issue: e.target.value,
150       });
151     }
152
153     toggle() {
154       this.setState({
155         dropdownOpen: !this.state.dropdownOpen,
156       });
157     }
158
159     checkKind(e) {
160       const data = parseInt(e.target.value, 10);
161       this.setState({
162         checked: data,
163       });
164       if(data == 1) {
165         this.setState({
166           bibType: 'book',
167         });
168       }
169       if(data == 2) {
170         this.setState({
171           bibType: 'journal',
172         });
173       }
174     }
175   }
176 }
177
178 export default BookForm;
```



```

169     if(data == 2) {
170       this.setState({
171         bibType: 'journal',
172       });
173     }
174     if(data == 3) {
175       this.setState({
176         bibType: 'article',
177       });
178     }
179
180     console.log(this.state.bibType);
181   }
182
183   render() {
184     return (
185       <div className="animated fadeIn">
186         <br />
187         <h1>Publikacja <Badge color="secondary" style={styles.badgeStyle}>Nowa</Badge></h1>
188         <br />
189         <InputGroup style={styles.inputWidth}>
190           <Input placeholder="Id (id)" onChange={this.updateId} />
191         </InputGroup>
192         <br />
193         <InputGroup style={styles.inputWidth}>
194           <Input placeholder="Tytuł (title)" onChange={this.updateTitle} />
195         </InputGroup>
196         <br />
197         <Dropdown isOpen={this.state.dropdownOpen} toggle={this.toggle}>
198           <DropdownToggle caret style={styles.buttonColors}>
199             Jakiego typu jest publikacja (bibType)?
200           </DropdownToggle>
201           <DropdownMenu>
202             <DropdownItem header>Wybierz typ</DropdownItem>
203             <DropdownItem value={1} onClick={e => this.checkKind(e)}>Książka</DropdownItem>
204             <DropdownItem value={2} onClick={e => this.checkKind(e)}>Czasopismo</DropdownItem>
205             <DropdownItem value={3} onClick={e => this.checkKind(e)}>Artykuł</DropdownItem>
206           </DropdownMenu>
207         </Dropdown>
208         <br />
209         {this.state.checked === 1 ? <div><h4><Badge color="secondary" style={styles.badgeStyle}>Książka</Badge></h4></div> : ''}
210         {this.state.checked === 2 ? <div><h4><Badge color="secondary" style={styles.badgeStyle}>Czasopismo</Badge></h4></div> : ''}
211         {this.state.checked === 3 ? <div><h4><Badge color="secondary" style={styles.badgeStyle}>Artykuł</Badge></h4></div> : ''}
212         <br />
213         <InputGroup style={styles.inputWidth}>
214           <Input placeholder="Autor (publisherId.name)" onChange={this.updatePublisherIdName} />
215         </InputGroup>
216         <br />
217         <InputGroup style={styles.inputWidth}>
218           <Input placeholder="Wydawnictwo (institutionId.name)" onChange={this.updateInstitutionIdName} />
219         </InputGroup>

```

```

217     <InputGroup style={styles.inputWidth}>
218       <Input placeholder="Wydawnictwo (institutionId.name)" onChange={this.updateInstitutionIdName} />
219     </InputGroup>
220     <br />
221     <InputGroup style={styles.inputWidth}>
222       <Input placeholder="Czasopismo (journalId.name)" onChange={this.updateJournalIdName} />
223     </InputGroup>
224     <br />
225     <InputGroup style={styles.inputWidth}>
226       <Input placeholder="Rok wydania (year)" onChange={this.updateYear} />
227     </InputGroup>
228     <br />
229     <InputGroup style={styles.inputWidth}>
230       <Input placeholder="Wolumin (vol)" onChange={this.updateVol} />
231     </InputGroup>
232     <br />
233     <InputGroup style={styles.inputWidth}>
234       <Input placeholder="Wydanie (issue)" onChange={this.updateIssue} />
235     </InputGroup>
236     <br />
237     <InputGroup style={styles.inputWidth}>
238       <Input placeholder="ISBN (isbn)" onChange={this.updateIsbn} />
239     </InputGroup>
240     <br />
241     <InputGroup style={styles.inputWidth}>
242       <Input placeholder="DOI (doi)" onChange={this.updateDoi} />
243     </InputGroup>
244     <br />
245     <Button color="primary" style={styles.addButton} onClick={this.addNewPublications}>Dodaj</Button>{' '}
246   </div>
247   );
248 }
249 }

```

e) Wybór dodawania

```
15 export default class TypeChoice extends React.Component {
16   constructor(props) {
17     super(props);
18
19     this.toggle = this.toggle.bind(this);
20     this.checkType = this.checkType.bind(this);
21     this.state = {
22       dropdownOpen: false,
23       checked: 0,
24     };
25   }
26
27   toggle() {
28     this.setState({
29       dropdownOpen: !this.state.dropdownOpen,
30     });
31   }
32
33   checkType(e) {
34     const data = parseInt(e.target.value, 10);
35     this.setState({
36       checked: data,
37     });
38   }
39
40   render() {
41     return (
42       <div>
43         <Dropdown isOpen={this.state.dropdownOpen} toggle={this.toggle}>
44           <DropdownToggle caret style={styles.buttonColors}>
45             Co chcesz dodać?
46           </DropdownToggle>
47           <DropdownMenu>
48             <DropdownItem header>Wybierz typ</DropdownItem>
49             <DropdownItem value={1} onClick={e => this.checkType(e)}>Publikacja</DropdownItem>
50             <DropdownItem value={2} onClick={e => this.checkType(e)}>Cytowanie</DropdownItem>
51             <DropdownItem value={3} onClick={e => this.checkType(e)}>Autor</DropdownItem>
52             <DropdownItem value={4} onClick={e => this.checkType(e)}>Wydawnictwo</DropdownItem>
53           </DropdownMenu>
44         </Dropdown>
55         {this.state.checked === 1 ? <div><AddPublication /></div> : ''}
56         {this.state.checked === 2 ? <div><AddCitation /></div> : ''}
57         {this.state.checked === 3 ? <div><AddAuthor /></div> : ''}
58         {this.state.checked === 4 ? <div><AddHouse /></div> : ''}
59       </div>
60     );
61   }
62 }
```

3. System zakładek

a) Ścieżki

```
17 const Full = () => (  
18   <div className="app">  
19     <Header />  
20     <div className="app-body">  
21       <Sidebar />  
22       <main className="main">  
23         <Breadcrumb />  
24         <Container fluid>  
25           <Switch>  
26             <Route path="/home" name="Strona główna" component={Home} />  
27             <Route path="/publications" name="Publikacje" component={Publications} />  
28             <Route path="/citations" name="Cytowania" component={Citations} />  
29             <Route path="/authors" name="Autorzy" component={Authors} />  
30             <Route path="/houses" name="Wydawnictwa" component={Houses} />  
31             <Route path="/add" name="Dodaj..." component={Add} />  
32             <Redirect from="/" to="/home" />  
33           </Switch>  
34         </Container>  
35       </main>  
36       <Aside />  
37     </div>  
38     <Footer />  
39   </div>  
40 );  
41  
42 export default Full;
```

b) Route

```
1  const routes = {  
2    '/': 'Strona główna',  
3    '/login': 'Login',  
4    '/home': 'Strona główna',  
5    '/publications': 'Publikacje',  
6    '/citations': 'Cytowania',  
7    '/authors': 'Autorzy',  
8    '/houses': 'Wydawnictwa',  
9    '/add': 'Dodaj...',  
10  };  
11  export default routes;
```

c) Nawigacja w panelu bocznym (menu)

```
1  export default {
2    items: [
3      {
4        name: 'Strona główna',
5        url: '/home',
6        icon: 'icon-home',
7        badge: {
8          variant: 'info',
9          text: '',
10        },
11      },
12      {
13        name: 'Publikacje',
14        url: '/publications',
15        icon: 'icon-book-open',
16        badge: {
17          variant: 'info',
18          text: '',
19        },
20      },
21      {
22        name: 'Cytowania',
23        url: '/citations',
24        icon: 'icon-link',
25        badge: {
26          variant: 'info',
27          text: '',
28        },
29      },
30      {
31        name: 'Autorzy',
32        url: '/authors',
33        icon: 'icon-people',
34        badge: {
35          variant: 'info',
36          text: '',
37        },
38      },
39      {
40        name: 'Wydawnictwa',
41        url: '/houses',
42        icon: 'icon-graduation',
43        badge: {
44          variant: 'info',
45          text: '',
46        },
47      },
48      {
49        name: 'Dodaj...',
50        url: '/add',
51        icon: 'icon-plus',
52        badge: {
53          variant: 'info',
54          text: '',
55        },
56      },
57    ],
58  },
```