

# Aplikacja Web

Wykonaj aplikację internetową typu front-end obsługującą zapisy na kursy z zastosowaniem dostępnej na stanowisku egzaminacyjnym biblioteki React. Zastosuj bibliotekę Bootstrap do zdefiniowania stylu formularza.

Obraz 1.

Dodaj kontakt	Lista kontaktów
Imię: <input type="text"/>	
Numer telefonu: <input type="text"/>	
<input type="button" value="Dodaj"/>	<input type="button" value="Wyczyść listę"/>

Obraz 2.

Dodaj kontakt	Lista kontaktów
Imię: <input type="text"/>	• Jan (567465778)
Numer telefonu: <input type="text"/>	
<input type="button" value="Dodaj"/>	<input type="button" value="Wyczyść listę"/>

Na obrazach 1 i 2 przedstawiono działanie aplikacji przygotowanej w środowisku React.js. Na obrazie 1 przedstawiono stan początkowy aplikacji, na obrazie 2 przedstawiono stan po wybraniu przycisku "Dodaj", gdy w polach edycyjnych wpisane były odpowiednio wartości "Jan" oraz "567465778". W panelu prawym widoczne jest wyświetlenie imienia i numeru telefonu.

## Założenia aplikacji:

- Aplikacja składa się z komponentu głównego, komponentu formularza, komponentu listy kontaktów i komponentu elementu listy kontaktów
- Danymi aplikacji jest tablica z imionami i numerami telefonów
- Aplikacja wyświetla:
  - Panel prawy, zawierający:
    - Nagłówek drugiego stopnia o treści: "Lista kontaktów"
    - Listę nienumerowaną, generowaną automatycznie dla wszystkich elementów tablicy, niezależnie od jej wymiaru
    - Każdy element listy powinien mieć przypisany unikalny klucz
  - Panel lewy, zawierający:
    - Nagłówek drugiego stopnia o treści: "Dodaj kontakt"
    - Formularz składający się z:
      - pola edycyjnego i jego etykiety o treści "Imię:"
      - pola edycyjnego pola numerycznego i jego etykiety o treści "Numer telefonu:"

- przycisku "Dodaj"
  - Aplikacja w stanie początkowym wyświetla puste pola formularza
- Po wybraniu przycisku formularza jest generowane zdarzenie zatwierdzenia formularza, które:
  - Dodaje element do listy w panelu prawym, wartość wpisaną w pierwszym polu formularza, połączoną z wartością w drugim polu formularza
  - Czyści obydwa pola formularza
- Elementy aplikacji są formatowane zgodnie z obrazem 1 i obrazem 2 za pomocą stylów biblioteki Bootstrap. Do budowy szablonu HTML należy wykorzystać pomoc zamieszczoną w Przykładzie 1. Należy zastosować znaczące nazwy dla identyfikatorów pól formularza
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze web powinno znaleźć się archiwum całego projektu o nazwie web.zip oraz pliki z kodem źródłowym, które były modyfikowane.

## Przykład 1. Wybrane elementy biblioteki React.js i biblioteki Bootstrap - przykłady

### React.js

To use Bootstrap add: `import 'bootstrap/dist/css/bootstrap.css';`

### Bootstrap Forms (Źródło <https://getbootstrap.com/docs/4.0/components/forms/>)

Be sure to use an appropriate type attribute on all inputs (e.g., email for email address or number for numerical information) to take advantage of newer input controls like email verification, number selection, and more. Textual form controls—like `<input>`s, `<select>`s, and `<textarea>`s - are styled with the `.form-control` class. Included are styles for general appearance, focus state, sizing, and more.

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" />
  </div>
  <div class="form-group">
    ...
  </div>
</form>
```

Important! In React render method use `className` instead of `class`; `htmlFor` instead of `for`.

### Bootstrap flex

```
<div className="d-flex justify-content-between w-full">
  ....
</div>
```

```
</div>
```

Use `w-full` to set the width value to 100%.

Important! In React render method use `className` instead of `class`; `htmlFor` instead of `for`.

### **Bootstrap buttons (Źródło <https://getbootstrap.com/docs/4.0/components/buttons/>)**

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control. The `btn` classes are designed to be used with the `<button>` element. Add modifier classes as: `btn-primary`, `btn-secondary`, `btn-success` and more to `btn` class in order to add background colors. e.g. `<button type="button" class="btn btn-success">Success</button>`

### **Bootstrap grid**

```
<div class="container text-center">
  <div class="row">
    <div class="col">
      ...
    </div>
    <div class="col">
      ...
    </div>
  </div>
```

Important! In React render method use `className` instead of `class`; `htmlFor` instead of `for`.

## **Przykład 2. Wybrane wbudowane funkcje języka JavaScript i hooki biblioteki React.js**

### **Array methods in JavaScript**

Method	Description
<code>map(el =&gt; el.id)</code>	Creates a new array from calling a function for every array element.
<code>filter(str =&gt; str.length &lt;= 3)</code>	Creates a new array filled with elements that pass a test provided by a function.
<code>reduce((total, num) =&gt; total + num, 0)</code>	Executes a reducer function for array element and returns a single value: the function's accumulated result.

### **React.js hooks**

Method	Description
<pre>const countRef = useRef();</pre>	Creates a mutable reference that persists across renders, without causing re-renders
<pre>const [count, setCount] = useState(0);</pre>	Creates an accesible value, that persists across renders and method to change it.
<pre>useEffect(() =&gt; {...}, [])</pre>	Allows to perform side effects in components, after change in the items specified in dependency table.