

Alan Chachoł 3F

Person

1. Plik PersonForm.jsx. Wykorzystując informacje zdobyte z poprzednich lekcji wykonuje następujący kod:

PearsonForm renderuje formularz edycji danych osobowych. Zawiera on pola do wprowadzenia peselu, imienia, nazwiska, adresu, numeru telefonu, adresu e-mail oraz płci. Wartości które zostały wprowadzone są przekazywane do `onInputChange` za pomocą funkcji `onChange`.

```
import React from "react";
import "../PearsonForm.css";
const PearsonForm = (props) => {
  return (
    <div className='EdycjaLekcji'>
      <div className='EdycjaLekcji_input-grupa'>
        <br />
        <label htmlFor='pesel'>Podaj pesel: </label>
        <input
          placeholder='Wpisz pesel'
          type='number'
          id='pesel'
          name='pesel'
          onChange={(e) =>
            props.onInputChange({ [e.target.name]: e.target.value })
          }
        />
      </div>
      <div className='EdycjaLekcji_input-grupa'>
        <label htmlFor='name'>Podaj imie: </label>
        <input
          placeholder='Wpisz imie'
          type='name'
          id='name'
          name='name'
          onChange={(e) =>
            props.onInputChange({ [e.target.name]: e.target.value })
          }
        />
      </div>
      <div className='EdycjaLekcji_input-grupa'>
        <label htmlFor='lastname'>Podaj nazwisko: </label>
        <input
          placeholder='Wpisz nazwisko'
          type='text'
          id='lastname'
          name='lastname'
          onChange={(e) =>
            props.onInputChange({ [e.target.name]: e.target.value })
          }
        />
      </div>
      <div className='EdycjaLekcji_input-grupa'>
        <label htmlFor='lastname'>Podaj pełny adres: </label>
        <input
          placeholder='Wpisz adres'

```

```

<label htmlFor='lastname'>Podaj pełny adres: </label>
<input
  placeholder='Wpisz adres'
  type='text'
  id='adres'
  name='adres'
  onChange={(e) =>
    props.onInputChange({ [e.target.name]: e.target.value })
  }
/>
</div>

<div className='EdycjaLekcji_input-grupa'>
<label htmlFor='lastname'>Podaj tel.: </label>
<input
  placeholder='Wpisz tel'
  type='tel'
  id='tel'
  name='tel'
  onChange={(e) =>
    props.onInputChange({ [e.target.name]: e.target.value })
  }
/>
</div>

<div className='EdycjaLekcji_input-grupa'>
<label htmlFor='lastname'>Podaj email.: </label>
<input
  placeholder='Wpisz email'
  type='email'
  id='email'
  name='email'
  onChange={(e) =>
    props.onInputChange({ [e.target.name]: e.target.value })
  }
/>
</div>

<div className='EdycjaLekcji_input-grupa'>
<label htmlFor='lastname'>Podaj płeć.: </label>
<input
  placeholder='Wpisz płeć'
  type='text'
  id='plec'
  name='plec'
  onChange={(e) =>
    props.onInputChange({ [e.target.name]: e.target.value })
  }
/>
</div>

```

```

<div className='EdycjaLekcji_input-grupa'>
  <label htmlFor='lastname'>Podaj email.: </label>
  <input
    placeholder='Wpisz email'
    type='email'
    id='email'
    name='email'
    onChange={(e) =>
      props.onInputChange({ [e.target.name]: e.target.value })
    }
  />
</div>

<div className='EdycjaLekcji_input-grupa'>
  <label htmlFor='lastname'>Podaj płeć.: </label>
  <input
    placeholder='Wpisz płeć'
    type='text'
    id='plec'
    name='plec'
    onChange={(e) =>
      props.onInputChange({ [e.target.name]: e.target.value })
    }
  />
</div>

<button onClick={() => props.onSave()}>OK</button>
<button id='btncancel' type='reset'>
  Cancel
</button>
</div>
];
};

export default PearsonForm;

```

Gdy formularz zostanie wypełniony użytkownik może użyć przycisku OK, który spowoduje wywołanie funkcji 'onSave'.

2. Plik Person

zarządzamy tutaj danymi osobowymi. Używamy 'Output' do renderowania danych osobowych oraz 'PearsonForm' do interakcji z użytkownikiem.

```
import React, { Component } from "react";
import "../Pearson.css";
import Output from "../Output";
import PearsonForm from "../PearsonForm";

class Pearson extends Component {
  constructor() {
    super();
    this.state = {
      Pearson: [
        {
          pesel: "Przykład", name: "Przykład", lastname: "Przykład", adres: "Przykład", tel: "Przykład", email: "Przykład", plec: "Przykład" },
      ],
      editPearson: {
        pesel: "Brak",
        name: "Brak",
        lastname: "Brak",
        adres: "Brak",
        tel: "Brak",
        email: "Brak",
        plec: "Brak",
      },
    };
  }

  this.addPearson = this.addPearson.bind(this);
  this.savePearson = this.savePearson.bind(this);
}

addPearson(val) {
  this.setState((prevState) => {
    return {
      editPearson: Object.assign(prevState.editPearson, val),
    };
  });
}

savePearson() {
  this.setState((prevState) => {
    const { Pearson, editPearson } = prevState;
    const lastPearson = Pearson[Pearson.length - 1];

    if(lastPearson.pesel == editPearson.pesel){
      editPearson.pesel = "Błąd powtarzający się pesel!";
    }
  });
}
```

```

    if(lastPearson.pesel == editPearson.pesel){
      editPearson.pesel = "Błąd powtarzający się pesel!";
    }

    const newLesson = {
      pesel:editPearson.pesel || lastPearson.pesel,
      name: editPearson.name || lastPearson.name,
      lastname: editPearson.lastname || lastPearson.lastname,
      adres: editPearson.adres || lastPearson.adres,
      tel: editPearson.tel || lastPearson.tel,
      email: editPearson.email || lastPearson.email,
      plec: editPearson.plec || lastPearson.plec,
    };

    return {
      Pearson: [...Pearson, newLesson],
      editPearson: {
        ...newLesson,
      },
    };
  });
}

```

```

render() {
  const Pearson = this.state.Pearson.map((elm) => {
    return (
      <OutPut
        key={elm.pesel}
        name={elm.name}
        pesel={elm.pesel}
        lastname={elm.lastname}
        adres = {elm.adres}
        tel = {elm.tel}
        email={elm.email}
        plec = {elm.plec}
      />
    );
  });
}

```

```

    );
  });

  return (
    <div className='EdycjaLekcji'>
      <br></br>
      <h1 id='h1'>Dane Osobowe</h1>

      <PearsonForm
        onChange={(val) => this.addPearson(val)}
        onSave={() => this.savePearson()}
      ></PearsonForm>

      <table>
        <tr>
          <th>Pesel</th>
          <th>Name</th>
          <th>Lastname</th>
          <th>Adres</th>
          <th>Tel</th>
          <th>Email</th>
          <th>Płeć</th>
        </tr>
        <tbody>
          {Pearson}
        </tbody>
      </table>
    </div>
  );
}

export default Pearson;

```

3. Plik Output.jsx

Na początku definiujemy 'BoxOutPut' i przyjmuje on właściwości 'props'. W dalszej części kodu są definiowane oczekiwane typy właściwości.

```

import PropTypes from "prop-types";

function BoxOutPut(props) {

  return (
    <tr>
      <td>{props.pesel}</td>
      <td>{props.name}</td>
      <td>{props.lastname}</td>
      <td>{props.adres}</td>
      <td>{props.tel}</td>
      <td>{props.email}</td>
      <td>{props.plec}</td>
    </tr>
  );
}

BoxOutPut.propTypes = {
  pesel: PropTypes.string,
  name: PropTypes.string,
  lastname: PropTypes.string,
  adres: PropTypes.string,
  tel: PropTypes.string,
  email: PropTypes.string,
  plec: PropTypes.string,
};

export default BoxOutPut;

```

Ostatnia linia pozwala na wykorzystanie 'BoxOutPust' w innych
częściach aplikacji

4. Wygląd strony

