

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут комп'ютерних наук та інформаційних технологій
Кафедра програмного забезпечення**



ЗВІТ

До лабораторної роботи №2

На тему: «Форма як головний елемент динаміки та зворотнього зв'язку з сервером»

З дисципліни: *«Програмування в Інтернет»*

Лектор:

ст. викладач кафедри ПЗ
Купльовський Б. Є.

Виконав:

ст. групи ПЗ-22
Павлів М. Я.

Прийняв:

доц. каф. ПЗ
Грицай О. Д.

«__» ____ 2022р.

Σ = ____

Львів – 2022

Тема роботи: форма як головний елемент динаміки та зворотнього зв'язку з сервером.

Мета роботи: оволодіти засобами керування формами, внесення інформація для сервера на динамічній web-сторінці.

Теоретичні відомості

Форма HTML містить групу елементів для приймання від користувача та пересилання даних у рядку запиту на web-сервер. Ними є `button`, `checkbox`, поля типу `text`, `input` тощо.

Важливий атрибут `action` вказує на URL сервера та програму на ньому для опрацювання даних запиту. Атрибут `method` вказує на вибір метода передачі даних: `get` чи `post`. Наведемо приклад тегу форми,

```
<form action="http://www.facerec.com/app.php" method="get">
  <fieldset>
    <label>Name: <input type="text" name="name" /></label>
    <label>sex: <input type="text" name="sex" /></label>
    <label>country?
      <input type="checkbox" name="country" />
    </label>
    <input id="sbutton" type="submit" value="Send
me your picture!" />
  </fieldset>
</form>
```

Для перевірки правильності внесених на форму даних використовуються скрипти JavaScript.

Індивідуальне завдання

1. Пройти наступні частини [курс по html](#) на codecademy: [HTML Forms](#) та [HTML Forms Validation](#)
2. Доповнити частину для внесення даних гравця web-сторінки із полем для шахів:
 - елементи внесення даних;
 - валідацію форми лише засобами HTML;
 - перемикачі;
 - багаторядкові поля для вводу тексту та приховані поля;
 - списки у формі;
 - кнопки у формі та реакції на них функцією JavaScript;
 - сформувати рядок даних, що буде передаватись з даної форми на сервер. Тематика сайту довільна, або з навчального матеріалу.

Хід роботи

1. Пройшов курс по html на codecademy.
2. Реалізував форму відповідно до вимог:



Рис. 1. Вигляд сторінки



Рис. 2. Вигляд форми, коли користувач ввів неправильні дані.



Рис. 3. Вигляд форми, коли користувач ввів правильні дані.

Код програми

index.js

```
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
```

```
ReactDOM.render(<App />, document.getElementById('root'));
```

index.css

```
@import
url('https://fonts.googleapis.com/css2?family=Noto+Sans+JP:wght@400;700
&display=swap');
```

```
* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}
```

```
html {
  font-family: 'Noto Sans JP', sans-serif;
}
```

```
:root {
  font-size: 4vmin;
}
```

```
body {
  margin: 0;
```

```
    background: #3f3f3f;
  }

```

App.js

```
import {useState, useReducer, useEffect} from 'react';
import Footer from './components/UI/Footer';
import ChessBoard from './components/Chess/ChessBoard';
import './App.css';
import layout from './layout';
import Form from './components/UI/Form';
import Input from './components/UI/Input';

const DEFAULT_STATE = {
  value: '',
  isValid: null
};

const emailReducer = (state, action) => {
  if (action.type === 'USER_INPUT') {
    return {value: action.val, isValid: action.val.includes('@')};
  }

  if (action.type === 'INPUT_BLUR') {
    return {value: state.value, isValid: state.value.includes('@')};
  }

  return {...DEFAULT_STATE};
};

const passwordReducer = (state, action) => {
  const passwordPattern = /^(?=.*[A-Za-z])(?=.*[0-9])(?=.*[!@#$%^&*()_\-+=|\\,./?])[A-Za-z0-9!@#$%^&*()_\-+=|\\,./?]{8,}$/i;

  if (action.type === 'USER_INPUT') {
    const value = action.val.trim();

    const isValid = value.length >= 8 && value.search(passwordPattern)
    >= 0;

    return {value: action.val, isValid: isValid};
  }

  if (action.type === 'INPUT_BLUR') {
    const value = state.value.trim();

    const isValid = value.length >= 8 && value.search(passwordPattern)
    >= 0;

    return {value: state.value, isValid: isValid};
  }
}
```

```

    return {...DEFAULT_STATE};
  };

  const App = () => {
    const [formIsValid, setFormIsValid] = useState(false);
    const [showForm, setShowForm] = useState(true);

    const [emailState, dispatchEmail] = useReducer(emailReducer, {
      ...DEFAULT_STATE
    });

    const [passwordState, dispatchPassword] = useReducer(passwordReducer, {
    {
      ...DEFAULT_STATE
    }
    });

    const {isValid: emailIsValid} = emailState;
    const {isValid: passwordIsValid} = passwordState;

    useEffect(() => {
      const identifier = setTimeout(() => {
        setFormIsValid(emailIsValid && passwordIsValid);
      }, 500);

      return () => {
        clearTimeout(identifier);
      };
    }, [emailIsValid, passwordIsValid]);

    const submitHandler = (event) => {
      event.preventDefault();
      setShowForm(false);
    };

    const emailChangeHandler = (event) => {
      dispatchEmail({type: 'USER_INPUT', val: event.target.value});
    };

    const passwordChangeHandler = (event) => {
      dispatchPassword({type: 'USER_INPUT', val: event.target.value});
    };

    const validateEmailHandler = () => {
      dispatchEmail({type: 'INPUT_BLUR'});
    };

    const validatePasswordHandler = () => {
      dispatchPassword({type: 'INPUT_BLUR'});
    };
  };

```

```

return (
  <>
    <main>
      <ChessBoard layout={layout} />
      {showForm &&
      <Form onSubmit={submitHandler} >
        <h2>Log In</h2>
        <Input
          input={{
            type: 'email',
            id: 'email',
            name: 'email',
            value: emailState.value,
            placeholder: 'test@domain.com',
            onChange: emailChangeHandler,
            onBlur: validateEmailHandler
          }}
          isValid={emailIsValid}
          hasLabel='true'
          labelContent='Email'
        />
        <Input
          input={{
            type: 'password',
            id: 'password',
            name: 'password',
            value: passwordState.value,
            placeholder: 'Password',
            onChange: passwordChangeHandler,
            onBlur: validatePasswordHandler
          }}
          isValid={passwordIsValid}
          hasLabel='true'
          labelContent='Password'
        />
        <button type='submit' disabled={!formIsValid}>
          Sign Up
        </button>
      </Form>}
    </main>
    <Footer>
      Read more about the <a
href='http://www.sakpalota.hu/index.php/en/chess/rules'>Game
      rules</a> and the <a
href='https://www.chess.com/puzzles/problem/57088'>puzzle</a>.
    </Footer>
  </>
);
};

```

```
export default App;
```

App.css

```
main {  
  display: flex;  
  flex-direction: row;  
  align-items: center;  
}
```

Form.js

```
import classes from './Form.module.css';
```

```
const Form = (props) => {  
  return (  
    <form  
      className={classes.form}  
      onSubmit={props.onSubmit}  
    >  
      {props.children}  
    </form>  
  );  
};
```

```
export default Form;
```

Form.module.css

```
.form {  
  align-self: center;  
  margin: auto;  
  color: #fafafa;  
  text-align: center;  
  border: #769655 5px solid;  
  border-radius: 20px;  
  padding: 0.2rem 1rem;  
}
```

```
.form h2 {  
  font-size: 0.8rem;  
  margin-bottom: 1rem;  
}
```

```
.form button {  
  margin-top: 1rem;  
  border: #769655 2px solid;  
  padding: 0.2rem 0.4rem;  
  background: none;  
  color: #fafafa;  
  border-radius: 20px;  
  font-weight: bold;
```



```

}

.form button:hover,
.form button:active {
  background: #769655;
  outline: none;
}

.form button:disabled,
.form button:focus:disabled,
.form button:hover:disabled,
.form button:active:disabled {
  background: #ccc;
  border-color: #ccc;
  color: #666666;
  cursor: not-allowed;
}
Input.js
import classes from './Input.module.css';

const Input = (props) => {
  const {isValid} = props;
  const className = `${classes.input} ${isValid === false ?
classes.invalid : ''}`;

  return (
    <div className={className}>
      {props.hasLabel && <label
htmlFor={props.input.id}>{props.labelContent}</label>}
      <input {...props.input} />
    </div>
  );
};

export default Input;
Input.module.css

.input {
  display: flex;
  align-items: center;
  text-align: left;
}

.input label, .input input {
  display: block;
}

.input label {
  flex: 1;
  font-size: 0.4rem;

```

```
}

.input input, input.invalid input {
  flex: 2;
  padding: 0.1rem;
  margin: 0.1rem;
}

.input input {
  border: #769655 2px solid;
}

.input.invalid input {
  border: #fa4343 2px solid;
}

.input input:focus {
  outline: none;
}
```

Висновки

На лабораторній роботі я оволодів засобами керування формами, реалізував власну форму, яка містить 2 поля для вводу: електронна пошта і пароль. Валідацію вводу реалізував за допомогою Javascript. Якщо користувач ввів 2 поля правильно, то тільки тоді активується кнопка для підтвердження форми.