

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

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



**ЗВІТ**

До лабораторної роботи №4

**На тему:** «Використання JQuery (або React JS) та Bootstrap»

**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.

$\Sigma$ = \_\_\_\_ .....

Львів – 2022

**Тема роботи:** Використання JQuery (або React JS) та Bootstrap.

**Мета роботи:** Ознайомитись з функціями бібліотек для опрацювання змісту сторінки.

## Теоретичні відомості

React — це JavaScript-бібліотека для створення інтерфейсів користувача.

React DOM порівнює елемент і його дочірні елементи з попередніми та вносить в DOM тільки необхідні зміни для приведення DOM у бажаний стан.

```
const name = 'Josh Perez';  
const element = <h1>Hello, {name}</h1>;  
  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

Цей кумедний синтаксис тегів не є ні рядком, ні HTML.

Він має назву JSX, і це розширення синтаксису для JavaScript. Ми рекомендуємо використовувати його в React, щоб описати, як повинен виглядати інтерфейс користувача. JSX може нагадувати мову шаблонів, але з усіма перевагами JavaScript.

На відміну від DOM-елементів, елементи React — звичайні об'єкти, легкі для створення. React DOM бере на себе оновлення DOM для його відповідності React-елементам.

Компоненти дозволяють розділити інтерфейс користувача на незалежні частини, придатні до повторного використання, і сприймати їх як такі, що функціонують окремо один від одного.

Найпростішим способом визначення компонента є написання функції JavaScript:

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

Ця функція є валідним React-компонентом, оскільки вона приймає єдиний аргумент “пропс” (скорочено від properties - властивості), який є об'єктом з даними і повертає React-елемент. Такі компоненти ми називаємо “функціональними компонентами”, оскільки вони буквально є JavaScript функціями.

Ви також можете використовувати ES6 класи, щоб визначити компонент:

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

## Індивідуальне завдання

1. Раніше розроблену та доповнену сторінку із полем для гри у шахи перенести в React JS із використанням наступним мінімальним набором елементів React Bootstrap:

- елементи для занесення даних,
- таблиця,

- посилання,
  - зображенням,
  - кнопки.
2. Надіслати аякс запит у арі для отримання посилання на картинку та відобразити її на сторінці як аватар гравця. Лінк для отримання посилання із арі: <https://random.dog/woof.json?ref=apilist.fun>

### Хід роботи

1. Переніс раніше розроблену сторінку в React.JS.
2. При підтвердженні даних користувача, надсилається запит для отримання посилання на зображення і та відображається на сторінці.



Рис. 1. Вигляд сторінки.



Рис. 2. Користувач залогінився і відображається зображення у верхньому правому кутку.

## 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';
import Modal from './components/UI/Modal';

import {
  Chart,
  CategoryScale,
  LinearScale,
  BarElement,
  Title,
  Tooltip
} from 'chart.js';
import {Bar} from 'react-chartjs-2';
import Header from './components/UI/Header';

Chart.register(
  CategoryScale,
  LinearScale,
  BarElement,
  Title,
  Tooltip
);

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.value.trim();

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

      return {value: action.value, 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 [showChart, setShowChart] = useState(false);

    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'});
    };

    const options = {
        responsive: false,
        plugins: {
            title: {
                display: true,
                text: 'Chess Figures'
            }
        }
    };

    const labels = layout.reduce((labels, currentRow) => {
        const notNullCells = currentRow.filter(cell => cell !== null);
        notNullCells.forEach(label => {
            if (!labels.includes(label)) {
                labels.push(label);
            }
        });
    });

    return labels;
}, []);

```

```

const dataset = {
  label: 'Count',
  data: layout.reduce((cells, currentRow) => {
    const notNullCells = currentRow.filter(cell => cell !== null);
    notNullCells.forEach(cell => {
      if (cells.filter(cellInfo => cellInfo.label === cell).length === 0) {
        cells.push({
          label: cell,
          count: 1
        });
      } else {
        const existingCell = cells.find(cellInfo => cellInfo.label ===
cell);
        existingCell.count++;
      }
    });

    return cells;
  }, []).map(info => info.count)
};

const data = {
  labels,
  datasets: [
    {
      backgroundColor: 'rgba(53, 162, 235, 0.5)',
      ...dataset
    }
  ]
};

const toggleShowChartHandler = () => {
  setShowChart(prevShow => !prevShow);
};

return (
  <>
    <Header onShowChart={toggleShowChartHandler} />
    <main>
      {showChart && (
        <Modal onClose={toggleShowChartHandler}>
          <Bar options={options} data={data} type='bar' height={400}
width={600} />
        </Modal>
      )}
    <ChessBoard layout={layout} />
  </>
);

```



```

{showForm && (
  <Form onSubmit={handleSubmit}>
    <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.sakkpalota.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;  
}
```

**Footer.js**

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

```
const Footer = (props) => {  
  return (  
    <footer className={classes.footer}>  
      {props.children}  
    </footer>  
  );  
};
```

```
export default Footer;
```

**Footer.module.css**

```
.footer {  
  position: fixed;  
  bottom: 0;  
  text-align: center;  
  color: #fafafa;  
  width: 100%;  
  background: #769655;  
}
```

**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;
}

```

#### Header.js

```

import {useCallback, useEffect, useState} from 'react';
import classes from './Header.module.css';
import useHttp from '../../hooks/use-http';
import {Button} from 'react-bootstrap';

```

```

const numberToTime = (number) => {
  let minutes = Math.floor(number / 60);
  let seconds = number % 60;

  return `${('0' + minutes).slice(-2)}:${('0' + seconds).slice(-2)}`;
};

const Header = (props) => {
  const [showTimer, setShowTimer] = useState(false);
  const [timerCounter, setTimerCounter] = useState(0);
  const { isLoggedIn } = props;

  const {
    sendRequest,
    data: imgUrl,
    status
  } = useHttp(useCallback(async () => {
    const response = await
fetch('https://random.dog/woof.json?ref=apilist.fun');
    const data = await response.json();

    if (!response.ok) {
      throw new Error(data.message || 'Could not fetch image.');
    }

    return data.url;
  }, []));

  useEffect(() => {
    if (showTimer) {
      const timer = setTimeout(() => {
        setTimerCounter(prevCounter => prevCounter + 1);
      }, 1000);

      return () => clearTimeout(timer);
    } else {
      setTimerCounter(0);
    }
  }, [showTimer, timerCounter]);

  useEffect(() => {
    if (isLoggedIn && (!imgUrl || imgUrl.endsWith('.mp4'))) {
      sendRequest();
    }
  }, [isLoggedIn, sendRequest, imgUrl]);

  let img;

```

```

    if (status === 'completed' && !imgUrl.endsWith('.mp4')) {
      img = <img alt='user-avatar' src={imgUrl} />;
    }

    const timerBtnClickHandler = () => {
      setShowTimer(prevShowTimer => !prevShowTimer);
    };

    const showChartHandler = () => {
      props.onShowChart();
    };

    return (
      <header className={classes.header}>
        <h1>Oldweeb Chess</h1>
        {showTimer && <span>{numberToTime(timerCounter)}</span>}
        <nav>
          <ul>
            <li>
              <Button
                onClick={timerBtnClickHandler}
                variant='outline-primary'
              >
                {showTimer ? 'Stop Timer' : 'Show Timer'}
              </Button>
            </li>
            <li>
              <Button
                onClick={showChartHandler}
                variant='outline-primary'
              >
                Show Chart
              </Button>
            </li>
          </ul>
        </nav>
        {isLoggedIn && img}
      </header>
    );
  };

```

***export default Header;***

```

export default Header;
Header.module.css

```

```

.header {
  width: 100%;
  height: 2rem;
  display: flex;

```

```

    justify-content: space-between;
    align-items: center;
    background-color: #769655;
    color: white;
    padding: 0 10%;
    font-size: 0.8rem;
  }

```

```

.header ul {
  display: flex;
  list-style: none;
  margin: 0;
  padding: 0;
  align-items: center;
}

```

```

.header button {
  background-color: #fd8888;
  border: 1px solid #fd8888;
  padding: 0.2rem 0.5rem;
  margin: 0.5rem;
  color: #fafafa;
  font-weight: bold;
  border-radius: 10px;
}

```

```

.header button:hover,
.header button:active {
  background-color: #fa4343;
  cursor: pointer;
}

```

## 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;  
}
```

### **Modal.js**

```
import classes from './Modal.module.css';  
import ReactDOM from 'react-dom';
```

```
const Backdrop = (props) => {  
  return <div className={classes.backdrop} onClick={props.onClose} />;  
};
```

```

const ModalOverlay = (props) => {
  return (
    <div c/assName={classes.modal}>
      <div c/assName={classes.content}>{props.children}</div>
    </div>
  );
};

```

```

const portalElement = document.getElementById('overlay-root');

```

```

const Modal = (props) => {
  return (
    <>
      {ReactDOM.createPortal(
        <Backdrop onClose={props.onClose} />,
        portalElement
      )}
      {ReactDOM.createPortal(
        <ModalOverlay>{props.children}</ModalOverlay>,
        portalElement
      )}
    </>
  );
};

```

```

export default Modal;
Modal.module.css

```

```

.backdrop {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100vh;
  z-index: 20;
  background-color: rgba(0, 0, 0, 0.75);
}

```

```

.modal {
  position: fixed;
  background-color: white;
  padding: 1rem;
  border-radius: 14px;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);
  z-index: 30;
  animation: slide-down 300ms ease-out forwards;
  top: 20%;
}

```



```

@keyframes slide-down {
  from {
    opacity: 0;
    transform: translateY(-3rem);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

}

use-http.js

```

```

import { useReducer, useCallback } from 'react';

```

```

const httpReducer = (state, action) => {
  if (action.type === 'SEND') {
    return {
      data: null,
      error: null,
      status: 'pending',
    };
  }

  if (action.type === 'SUCCESS') {
    return {
      data: action.responseData,
      error: null,
      status: 'completed',
    };
  }

  if (action.type === 'ERROR') {
    return {
      data: null,
      error: action.errorMessage,
      status: 'completed',
    };
  }

  return state;
};

const useHttp = (requestFunction, startWithPending = false) => {
  const [httpState, dispatch] = useReducer(httpReducer, {
    status: startWithPending ? 'pending' : null,
    data: null,
    error: null,

```

```

});

const sendRequest = useCallback(
  async (requestData) => {
    dispatch({ type: 'SEND' });
    try {
      const responseData = await requestFunction(requestData);
      dispatch({ type: 'SUCCESS', responseData });
    } catch (error) {
      dispatch({
        type: 'ERROR',
        errorMessage: error.message || 'Something went wrong!',
      });
    }
  }, [requestFunction]
);

return {
  sendRequest,
  ...httpState,
};
};

export default useHttp;

```

## Висновки

На лабораторній роботі я переніс раніше розроблену веб-сторінку у React.JS. Вона повністю ідентична до тої, що була розроблена раніше. У цій роботі я використовував тільки функціональні компоненти. Коли користувач натискає кнопку Login, то відправляється аїах запит для отримання посилання на зображення і це зображення відображається на сторінці.