

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

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



ЗВІТ

До лабораторної роботи №3

На тему: «Особливості програмування мовою Javascript та каскадні стилі»

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: особливості програмування мовою Javascript та каскадні стилі.

Мета роботи: оволодіти структурою мови Javascript для створення інтерактивних web-сайтів та різними способами розміщення коду Javascript.

Теоретичні відомості

Одним з інструментів підтримки динамічних сценаріїв при перегляді Web-сторінок у межах комп'ютера користувача є мова програмування JavaScript – спрощений варіант мови програмування Java. Нею забезпечується рух об'єктів на сторінці, введення та виведення параметрів, зміна зображень вікон тощо. Програми модифікації, створення гіпертекстових сторінок традиційно називають скриптами (scripts), які інтерпретуються програмою перегляду. Спосіб базується на ідеології об'єктно-орієнтованого програмування. Зупинимось на скриптах, написаних мовою JavaScript.

Мова програмування для реалізації інтерактивності web-сторінок: введення тексту, реакції на події, отримання інформації з сервера, здійснення обчислень, підлягає стандартизації, але не підтримується всіма браузерами. Javascript інтерпретується на браузері та інтегрується за змістом HTML/CSS. Наведемо приклад внесення динамічного тексту:

`document.write("message");` - друкує текст на сторінці
Ввімкнення коду Javascript в HTML можливе трьома способами:

1. У тіло сторінки (виконується під час завантаження):

```
<body>
...
<script type="text/javascript"> Javascript code
</script>
...
</body>
```

2. У заголовок сторінки (виконується як реакція на подію (викликом)):

```
<head>
...
<script type="text/javascript"> Javascript code
</script>
...
</head>
```

3. Посиланням на зовнішній .js файл (розміщений у заголовку чи тілі):

```
<script src="filename" type="text/javascript"></script>
```

```
<script src="example.js" type="text/javascript"></script>
```

Розглянемо приклади типів даних та команд мови програмування. Тип змінної визначається за присвоєним значенням. Перепризначення значення об'єктів робиться операціями присвоювання. Оголошення змінної здійснюється ключовим словом `var` (`var text = "text"`).

Індивідуальне завдання

На раніше розробленій сторінці із полем для гри у шахи:

- Стилїзувати сторінку різними способами формування стилів;
- Додати до валідації форми даних гравця валідацію за допомогою Javascript;
- Додати кнопку старт, після натиснення якої з'являтиметься таймер із активним відліком;
- За допомогою бібліотеки Chart.js, побудувати довільну обрану гістограму, яка відображатиме наявну кількість кожного із типів фігур.

Хід роботи

1. Додав валідації даних за допомогою Javascript.
2. Додав кнопку, після натискання якої з'являється таймер.
3. Встановив бібліотеку Chart.js за допомогою команди `npm install react-chartjs-2 chart.js`.



Рис. 1. Вигляд сторінки



Рис. 2. Форма, коли гравець ввів неправильні дані.



Рис. 3. Кнопка запустила таймер.



Рис. 4. Гістограма.

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 {useEffect, useState} from 'react';
import classes from './Header.module.css';

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);

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

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

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

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

  return (
    <header className={classes.header}>
      <h1>O!dweeb Chess</h1>
      {showTimer && <span>{numberToTime(timerCounter)}</span>}
      <nav>
        <ul>
          <li>
            <button
              onClick={timerBtnClickHandler}
              >{showTimer ? 'Stop Timer' : 'Show Timer'}</button>
          </li>
          <li>
            <button onClick={showChartHandler}>Show Chart</button>
          </li>
        </ul>
      </nav>
    )
  );

```

```

    </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
html For={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.onClick} />;
};

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

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

const Modal = (props) => {
  return (
    <>
      {ReactDOM.createPortal(
        <Backdrop onClick={props.onClick} />,
        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);
  }
}

```

Висновки

На лабораторній роботі я оволодів структурою мови Javascript для створення інтерактивних web-сайтів та різними способами розміщення коду Javascript. В ході виконання лабораторної роботи я додав валідацію даних за допомогою Javascript, додав кнопку старт, яка запускає таймер, і за допомогою бібліотеки Chart.js побудував гістограму, яка відображає наявну кількість кожного із типів фігур.