 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Лабораторний практикум №9**

з дисципліни «*Веб-технології та веб-дизайн - 1*»

# на тему: *«*JS frameworks*»*

**Виконав:**

студент гр. БС-81

Збаровський Д. Д.

**Перевірив:**

ас. каф. БМК Матвійчук О. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

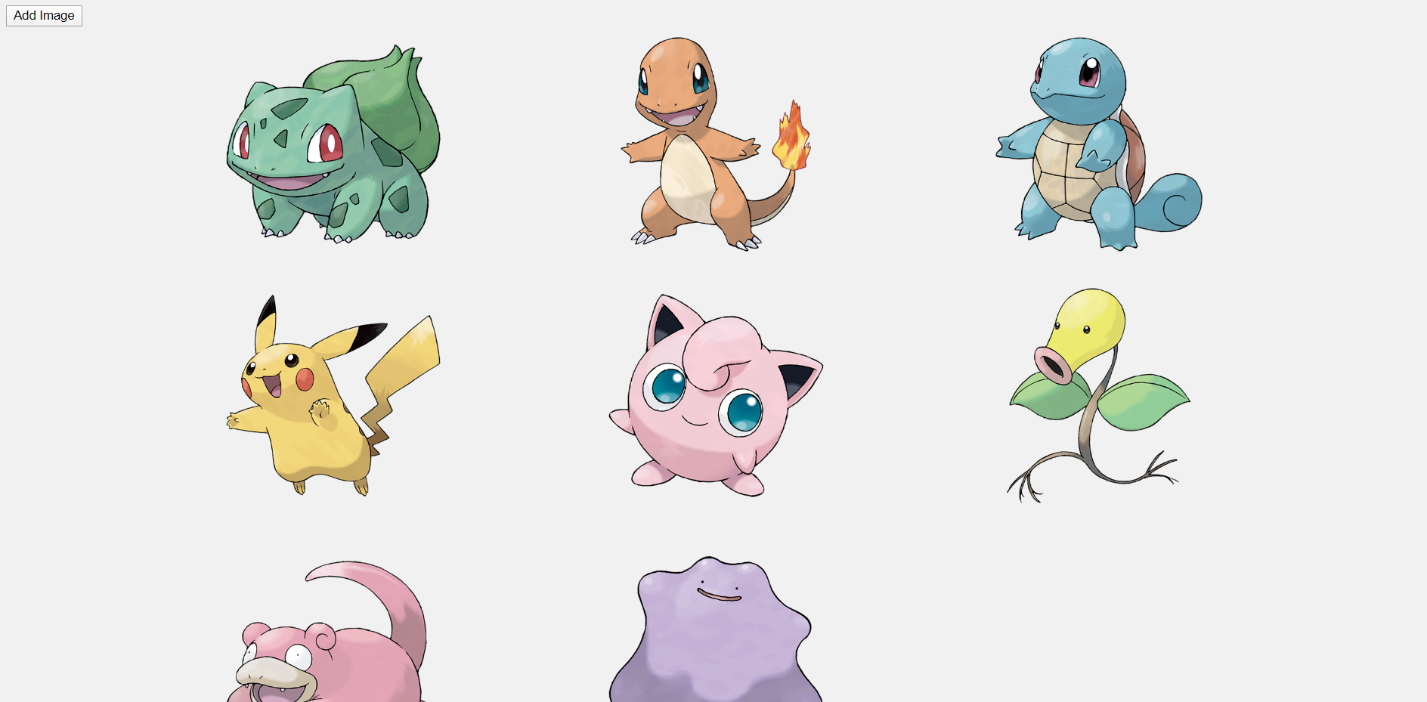
(підпис викладача)

Київ 2020

**Завдання:**

**Створити динамічну галерею зображень (index.html).**

* Вибрати фреймворк для реалізації галереї (Vue, React, Angular)
* Реалізувати галерею зображень як у лабораторній роботі #6 (включаючи режим preview)
* в Preview режимі відображати картинки сусіди (по 1 з кожної сторони, на кільцевому списку. В галереї в кожен момент часу має бути не менше 3х зображень).
* Додати функціонал з видалення зображень
* Додати функціонал з розширення галереї (форму для додавання нових зображень)
* Галерею реалізувати за паттерном Model-View-ViewModel





**Index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>My React Gallery</title>

<link rel="stylesheet" href="style.css">

<script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

</head>

<body>

<div id="app"></div>

<script type="text/babel">

const { createElement, Component } = React;

const { render } = ReactDOM;

function mod(a, b) {

return ((a % b) + b) % b;

}

class MyAddImageModal extends Component {

constructor(props) {

super(props);

}

render() {

return (

<div className="modal-mask">

<div className="modal-wrapper">

<div className="modal-container">

<div className="modal-header">

<h3>Add Image</h3>

</div>

<div className="modal-body">

<div>

<input type="file" onChange={this.props.onFileChange}/>

</div>

</div>

<div className="modal-footer">

<div style={{height: '18px'}}>

<button className="modal-default-button" onClick={this.props.onCancel}>

Cancel

</button>

<button className="modal-default-button" onClick={this.props.onAdd} disabled={this.props.addDisabled}>

Add

</button>

</div>

</div>

</div>

</div>

</div>

);

}

}

class MyPreview extends Component {

constructor(props) {

super(props);

}

render() {

return (

<div className="modal-mask">

<div className="modal-wrapper" onClick={this.props.onExit}>

<div className="preview-container">

<img src={this.props.url1} onClick={(e) => { e.stopPropagation(); this.props.onPrev(); }}/>

<img src={this.props.url2} onClick={(e) => { e.stopPropagation(); this.props.onExit(); }} style={{width: '30%'}}/>

<img src={this.props.url3} onClick={(e) => { e.stopPropagation(); this.props.onNext(); }}/>

<p><a href="#" onClick={(e) => { e.preventDefault(); e.stopPropagation(); this.props.onRemove(); }}>Remove</a></p>

</div>

</div>

</div>

);

}

}

class MyImage extends Component {

constructor(props) {

super(props);

}

render() {

return (

<div className="image-container">

<img src={this.props.url} onClick={this.props.onClick}/>

</div>

);

}

}

class MyApp extends Component {

constructor(props) {

super(props);

this.state = {

showModal: false,

modalFile: null,

urls: [

'resources/001.png',

'resources/004.png',

'resources/007.png',

'resources/025.png',

'resources/039.png',

'resources/069.png',

'resources/079.png',

'resources/132.png',

],

showPreview: false,

idx1: -1,

idx2: -1,

idx3: -1,

};

}

render() {

return (

<div>

<button

onClick={() => { this.setState({ modalFile: null, showModal: true }); }}

>

Add Image

</button>

<div className="gallery-container">

{

this.state.urls.map(url =>

<MyImage key={url} url={url} onClick={() => this.preview(url)}/>

)

}

</div>

{this.state.showPreview &&

<MyPreview

url1={this.state.urls[this.state.idx1]}

url2={this.state.urls[this.state.idx2]}

url3={this.state.urls[this.state.idx3]}

onPrev={() => this.previewPrev()}

onNext={() => this.previewNext()}

onRemove={() => this.previewRemove()}

onExit={() => this.previewExit()}

/>

}

{this.state.showModal &&

<MyAddImageModal

onFileChange={(e) => this.onFileChange(e)}

onCancel={() => this.setState({showModal: false})}

onAdd={() => this.modalAdd()}

addDisabled={this.state.modalFile == null}

/>

}

</div>

);

}

onFileChange(event) {

const modalFile = event.target.files[0];

this.setState((state) => ({

modalFile

}));

}

modalAdd() {

this.setState((state) => ({

urls: [...state.urls, URL.createObjectURL(state.modalFile)],

showModal: false

}));

}

preview(url) {

this.setState((state) => {

const idx2 = state.urls.indexOf(url);

return {

idx1: mod(idx2 - 1, state.urls.length),

idx2,

idx3: mod(idx2 + 1, state.urls.length),

showPreview: true

};

});

}

previewPrev() {

this.setState((state) => ({

idx1: mod(state.idx1 - 1, state.urls.length),

idx2: mod(state.idx2 - 1, state.urls.length),

idx3: mod(state.idx3 - 1, state.urls.length)

}));

}

previewNext() {

this.setState((state) => ({

idx1: mod(state.idx1 + 1, state.urls.length),

idx2: mod(state.idx2 + 1, state.urls.length),

idx3: mod(state.idx3 + 1, state.urls.length)

}));

}

previewRemove() {

this.setState((state) => ({

urls: state.urls.filter((\_, i) => i != state.idx2),

showPreview: false

}));

}

previewExit() {

this.setState((state) => ({

showPreview: false

}));

}

}

render(createElement(MyApp), document.querySelector('#app'));

</script>

</body>

</html>

**Style.css**

body {

background-color: #f1f1f1;

}

img {

object-fit: contain;

}

.gallery-container {

display: grid;

gap: 1rem;

max-width: 1200px;

margin: 0 auto;

}

@media (min-width: 600px) {

.gallery-container {

grid-template-columns: repeat(2, 1fr);

}

}

@media (min-width: 900px) {

.gallery-container {

grid-template-columns: repeat(3, 1fr);

}

}

.image-container {

width : 250px;

height: 250px;

margin: 0 auto;

}

.image-container img {

position: absolute;

width : inherit;

height: inherit;

cursor: pointer;

}

.preloader, .placeholder {

position: absolute;

width : 400px;

height: auto;

top : 50%;

left : 50%;

transform: translate(-50%, -50%);

}

.modal-mask {

position: fixed;

z-index: 9998;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-color: rgba(0, 0, 0, 0.5);

display: table;

transition: opacity 0.15s ease;

}

.modal-wrapper {

display: table-cell;

vertical-align: middle;

}

.modal-container {

width: 300px;

margin: 0px auto;

padding: 20px 30px;

background-color: #fff;

border-radius: 2px;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.33);

transition: all 0.15s ease;

font-family: Helvetica, Arial, sans-serif;

}

.modal-header h3 {

margin-top: 0;

color: blue;

}

.modal-body {

margin: 20px 0;

}

.modal-default-button {

float: right;

margin-left: 5px;

}

.modal-enter {

opacity: 0;

}

.modal-leave-active {

opacity: 0;

}

.modal-enter .modal-container,

.modal-leave-active .modal-container {

-webkit-transform: scale(1.1);

transform: scale(1.1);

}

.preview-container {

text-align: center;

transition: all 0.15s ease;

}

.preview-container img {

vertical-align: middle;

width: 20%;

cursor: pointer;

}

.preview-container a {

color: white;

font-family: Helvetica, Arial, sans-serif;

}

p:empty:before {

content: ' ';

white-space: pre;

}