**Лабораторна робота №9**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

*ДУ «Житомирська політехніка».23.121.03.00 – Лр9*

Розроб.

Білотіл Я. О.

Перевір.

Дмитренко І. А.

*Т. контр.*

Н. контр.

Затверд.

Звіт

з лабораторної роботи

Літ.

Аркушів

*15*

*ФІКТ Гр. IПЗк-23-1*

**Тема: ООП в JavaScript**

**Мета роботи:** вивчити особливості використання ООП в JavaScript

**Завдання на лабораторну роботу**

**Завдання 0:** Ознайомитись з теоретичними відомостями за посиланням:

[Прототипи, наслідування](https://uk.javascript.info/prototypes)

[Класи](https://uk.javascript.info/classes)

**Завдання 1.**

Створіть клас "Модальне вікно" (Modal), який створює модальне вікно на сторінці. Клас повинен мати методи для відкриття, закриття та встановлення вмісту модального вікна. Зробіть можливість переміщення вікна по екрану за допомогою мишки (щоб можна було перетягувати за заголовок вікна).

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Модальне вікно</title>

<script src="script.js"></script>

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

</head>

<body>

<button id="openBtn">Відкрити модальне вікно</button>

<div id="myModal" class="modal">

<div class="modal-content">

<span id="closeBtn" style="float:right;cursor:pointer;">&times;</span>

<p>Це модальне вікно! Тут можна вставити будь-який контент.</p>

</div>

</div>

</body>

</html>

style.css

.modal {

display: none;

position: fixed;

z-index: 1;

left: 0;

top: 0;

width: 100%;

height: 100%;

overflow: auto;

background-color: rgba(0, 0, 0, 0.4);

}

.modal-content {

background-color: #fefefe;

margin: 15% auto;

padding: 20px;

border: 1px solid #888;

width: 80%;

}

script.js

const openButton = document.getElementById("openBtn");

const modal = document.getElementById("myModal");

const closeButton = document.getElementById("closeBtn");

openButton.addEventListener("click", openModal);

closeButton.addEventListener("click", closeModal);

modal.addEventListener("click", outsideClick);

function openModal() {

modal.style.display = "block";

}

function closeModal() {

modal.style.display = "none";

}

function outsideClick(e) {

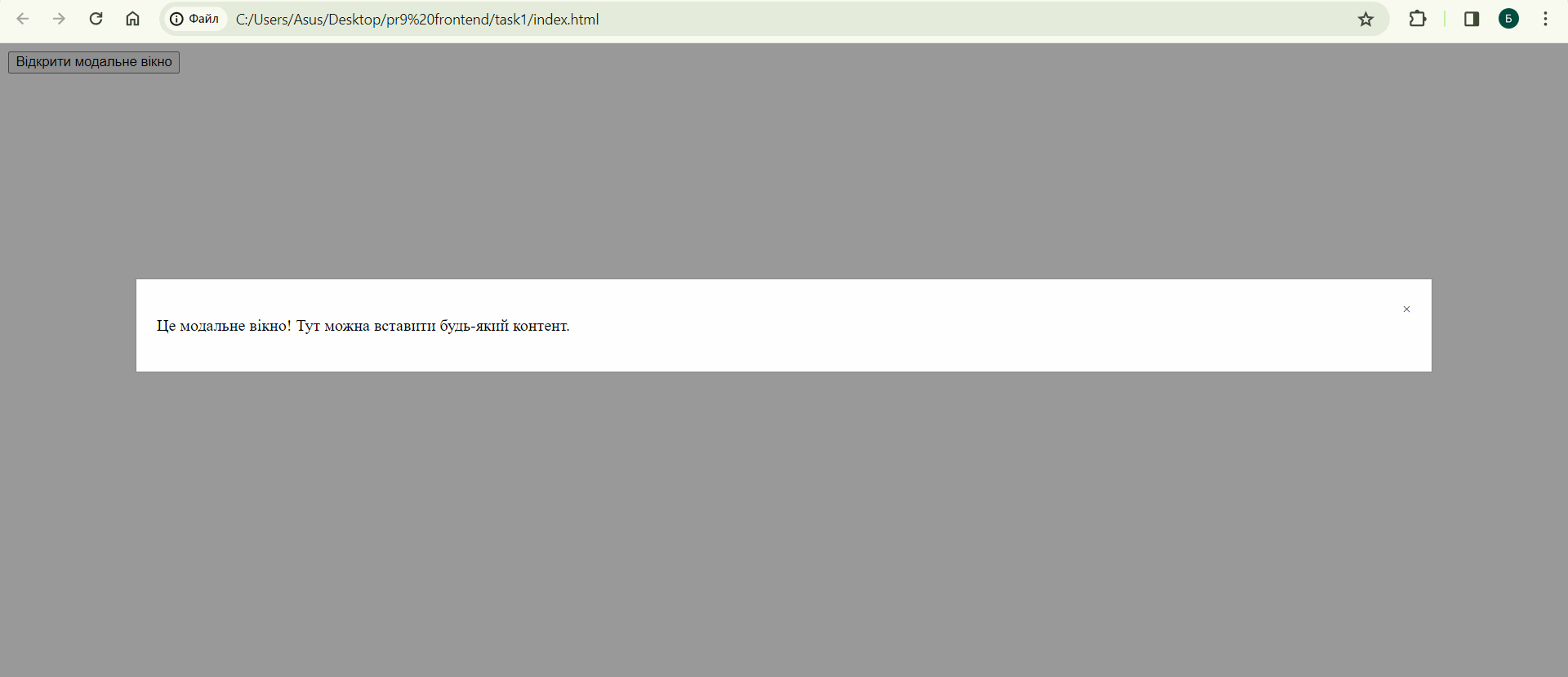
if (e.target === modal) {

modal.style.display = "none";

}

}

Результат роботи програми:



Мал. 1 – Завдання 1

**Завдання 2.**

Створіть клас "Слайдер" (Slider), який дозволяє створювати горизонтальний або вертикальний слайдер з можливістю перегляду зображень або вмісту. Клас повинен мати методи для додавання слайдів та переміщення між ними.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Горизонтальний слайдер з класами</title>

<script src="script.js"></script>

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

</head>

<body>

<div class="slider-container">

<div class="slider">

<div class="slide"><img src="image1.jpg" alt="Slide 1"></div>

<div class="slide"><img src="image2.jpg" alt="Slide 2"></div>

<div class="slide"><img src="image3.jpg" alt="Slide 3"></div>

</div>

</div>

<div class="slider-controls">

<button id="prevBtn">Назад</button>

<button id="nextBtn">Вперед</button>

<button id="addBtn">Додати слайд</button>

</div>

</body>

</html>

style.css

.slider-container {

width: 80%;

margin: 0 auto;

overflow: hidden;

position: relative;

}

.slider {

display: flex;

transition: transform 0.5s ease;

}

.slide {

flex: 0 0 100%;

}

.slider-controls {

text-align: center;

margin-top: 10px;

}

script.js

class Slider {

constructor() {

this.slider = document.querySelector('.slider');

this.slides = document.querySelectorAll('.slide');

this.prevButton = document.getElementById('prevBtn');

this.nextButton = document.getElementById('nextBtn');

this.addButton = document.getElementById('addBtn');

this.slideIndex = 0;

this.prevButton.addEventListener('click', () => this.prevSlide());

this.nextButton.addEventListener('click', () => this.nextSlide());

this.addButton.addEventListener('click', () => this.addSlide());

}

showSlides() {

this.slider.style.transform = `translateX(-${this.slideIndex \* 100}%)`;

}

nextSlide() {

if (this.slideIndex < this.slides.length - 1) {

this.slideIndex++;

} else {

this.slideIndex = 0;

}

this.showSlides();

}

prevSlide() {

if (this.slideIndex > 0) {

this.slideIndex--;

} else {

this.slideIndex = this.slides.length - 1;

}

this.showSlides();

}

addSlide() {

const newSlide = document.createElement('div');

newSlide.className = 'slide';

newSlide.innerHTML = `<img src="image4.jpg" alt="New Slide ${this.slides.length + 1}">`;

this.slider.appendChild(newSlide);

this.slides = document.querySelectorAll('.slide');

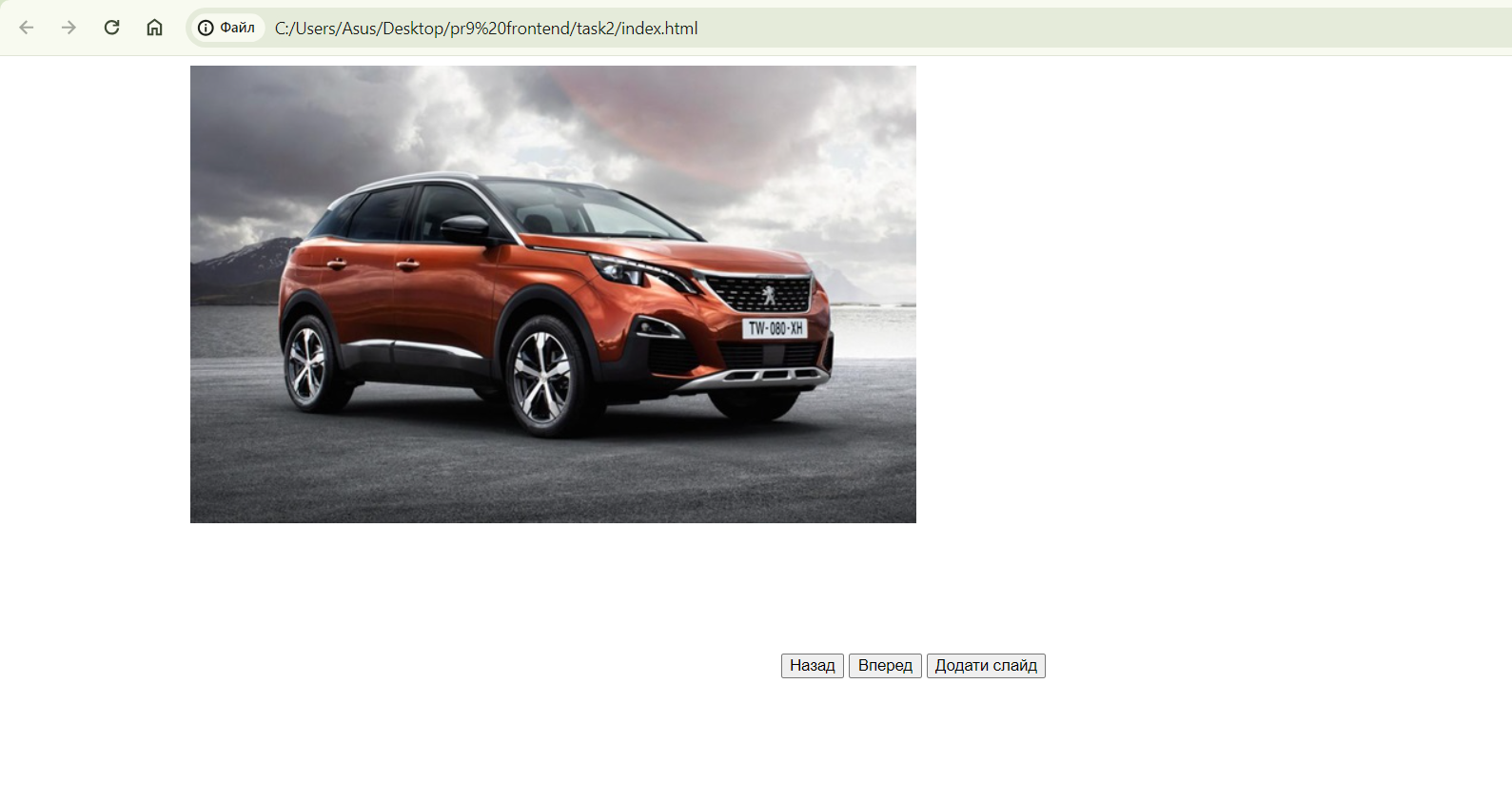
}

}

const mySlider = new Slider();

mySlider.showSlides();

Результат роботи програми:



Мал. 2 – Завдання 2

**Завдання 3.**

Створіть клас "Таблиця" (Table), який генерує HTML-таблицю на основі заданих даних. Клас повинен мати методи для додавання рядків та стовпців до таблиці.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Динамічна HTML-таблиця з класами</title>

<script src="script.js"></script>

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

</head>

<body>

<div>

<button onclick="myTable.addRow()">Додати рядок</button>

<button onclick="myTable.addColumn()">Додати стовпець</button>

</div>

<table id="myTable"></table>

</body>

</html>

style.css

table {

border-collapse: collapse;

margin: 20px auto;

}

th,

td {

border: 1px solid #ccc;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

script.js

class CustomTable {

constructor() {

this.table = document.getElementById('myTable');

this.rowCount = this.table.rows.length;

this.colCount = this.table.rows[0] ? this.table.rows[0].cells.length : 0;

}

addRow() {

const newRow = this.table.insertRow();

this.rowCount++;

for (let i = 0; i < this.colCount; i++) {

const newCell = newRow.insertCell();

newCell.innerHTML = `Row ${this.rowCount}, Col ${i + 1}`;

}

}

addColumn() {

for (let i = 0; i < this.rowCount; i++) {

const newCell = this.table.rows[i].insertCell();

this.colCount++;

newCell.innerHTML = `Row ${i + 1}, Col ${this.colCount}`;

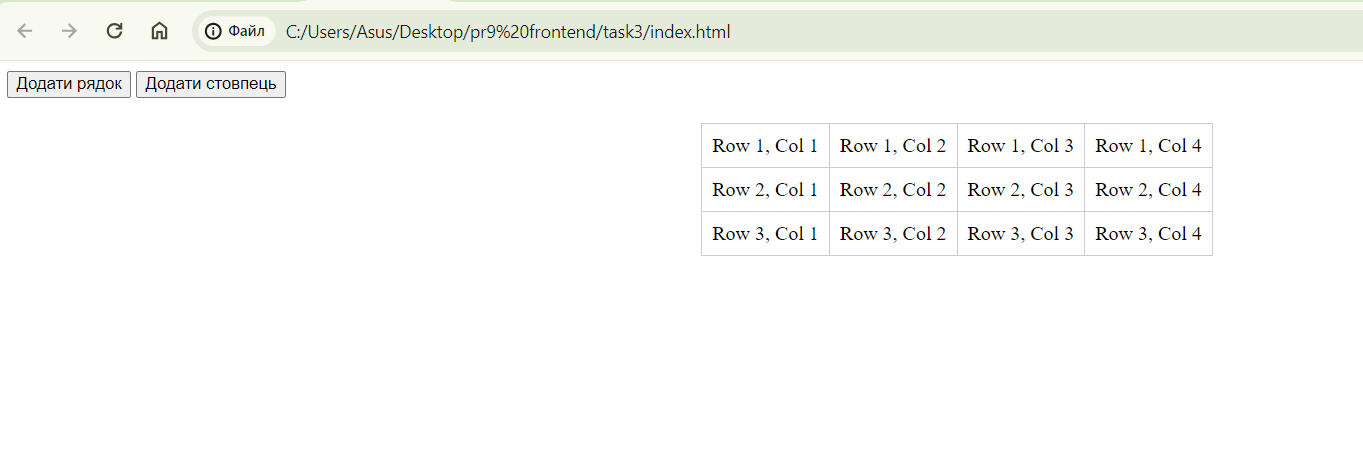
}

}

}

const myTable = new CustomTable();

Результат роботи програми:



Мал. 3 – Завдання 3

**Завдання 4.**

Створіть клас "Форма" (Form), який дозволяє створювати HTML-форми з заданими полями вводу, кнопками та обробниками подій для них. Клас повинен мати методи для обробки відправки форми та валідації введених даних.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Генератор HTML-форм з класами (з валідацією)</title>

<script src="script.js"></script>

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

</head>

<body>

<div id="formContainer"></div>

</body>

</html>

script.js

class FormGenerator {

constructor(containerId) {

this.container = document.getElementById(containerId);

this.fields = [];

this.submitHandler = null;

}

addInput(type, label) {

const input = document.createElement('input');

input.setAttribute('type', type);

const labelElem = document.createElement('label');

labelElem.textContent = label;

const br = document.createElement('br');

this.fields.push({ input, labelElem, br });

}

addButton(label, clickHandler) {

const button = document.createElement('button');

button.textContent = label;

button.addEventListener('click', clickHandler);

const br = document.createElement('br');

this.fields.push({ button, br });

}

setSubmitHandler(submitHandler) {

this.submitHandler = submitHandler;

}

validateForm() {

let isValid = true;

this.fields.forEach(field => {

if (field.input && field.input.hasAttribute('required') && field.input.value.trim() === '') {

isValid = false;

alert(`${field.labelElem.textContent} є обов'язковим полем`);

}

});

return isValid;

}

generateForm() {

this.fields.forEach(field => {

if (field.input) {

this.container.appendChild(field.labelElem);

this.container.appendChild(field.input);

this.container.appendChild(field.br);

} else if (field.button) {

this.container.appendChild(field.button);

this.container.appendChild(field.br);

}

});

if (this.submitHandler) {

const submitButton = document.createElement('button');

submitButton.textContent = 'Submit';

submitButton.type = 'submit';

submitButton.addEventListener('click', event => {

event.preventDefault();

if (this.validateForm()) {

this.submitHandler();

}

});

this.container.appendChild(submitButton);

}

}

}

const form = new FormGenerator('formContainer');

form.addInput('text', 'Ім\'я:');

form.addInput('email', 'Електронна пошта:');

form.addInput('password', 'Пароль:');

form.addButton('Надіслати', () => {

const name = form.fields[0].input.value;

const email = form.fields[1].input.value;

const password = form.fields[2].input.value;

console.log('Ім\'я:', name);

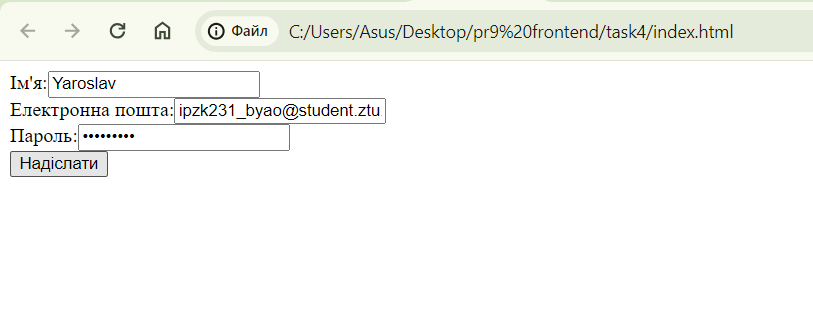
console.log('Електронна пошта:', email);

console.log('Пароль:', password);

});

form.generateForm();

Результат роботи програми:



Мал. 4 – Завдання 4

**Завдання 5.**

Створіть клас "Вкладка" (Tab), який створює вкладки для переключення між різними блоками вмісту. Клас повинен мати методи для додавання вкладок та вмісту для них, а також для переключення між вкладками.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Вкладки для переключення вмісту</title>

<script src="script.js"></script>

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

</head>

<body>

<div id="tabsContainer"></div>

<div id="tabContentContainer"></div>

<button id="addTabBtn">Додати вкладку</button>

<button id="changeContentBtn">Змінити вміст</button>

</body>

</html>

style.css

.tabContent {

display: none;

}

.activeTab {

display: block;

}

.tab {

cursor: pointer;

padding: 8px 16px;

border: 1px solid #ccc;

display: inline-block;

margin-bottom: -1px;

background-color: #f2f2f2;

}

.activeTab {

background-color: #fff;

border-bottom: 1px solid #fff;

}

script.js

class Tabs {

constructor(tabsContainerId, tabContentContainerId, addTabBtnId, changeContentBtnId) {

this.tabsContainer = document.getElementById(tabsContainerId);

this.tabContentContainer = document.getElementById(tabContentContainerId);

this.addTabBtn = document.getElementById(addTabBtnId);

this.changeContentBtn = document.getElementById(changeContentBtnId);

this.tabs = [];

this.tabContents = [];

this.addTabBtn.addEventListener('click', () => this.addTab('New Tab', 'New Content'));

this.changeContentBtn.addEventListener('click', () => this.changeContent());

}

addTab(tabName, tabContent) {

const tab = document.createElement('div');

tab.className = 'tab';

tab.textContent = tabName;

this.tabs.push(tab);

const tabContentElement = document.createElement('div');

tabContentElement.className = 'tabContent';

tabContentElement.textContent = tabContent;

this.tabContents.push(tabContentElement);

this.tabsContainer.appendChild(tab);

this.tabContentContainer.appendChild(tabContentElement);

tab.addEventListener('click', () => this.toggleTab(tab));

}

toggleTab(selectedTab) {

this.tabs.forEach((tab, index) => {

if (tab === selectedTab) {

tab.classList.add('activeTab');

this.tabContents[index].classList.add('activeTab');

} else {

tab.classList.remove('activeTab');

this.tabContents[index].classList.remove('activeTab');

}

});

}

changeContent() {

const activeTabContent = this.tabContentContainer.querySelector('.activeTab');

if (activeTabContent) {

activeTabContent.textContent = 'New Content for Active Tab';

}

}

}

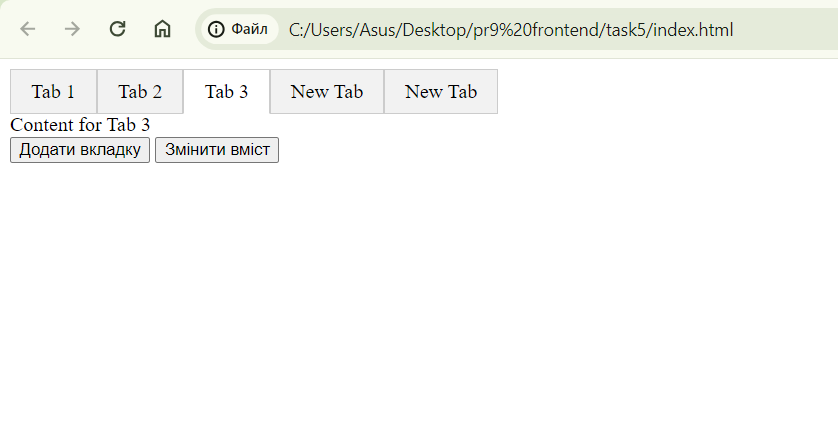
const tabs = new Tabs('tabsContainer', 'tabContentContainer', 'addTabBtn', 'changeContentBtn');

tabs.addTab('Tab 1', 'Content for Tab 1');

tabs.addTab('Tab 2', 'Content for Tab 2');

tabs.addTab('Tab 3', 'Content for Tab 3');

Результат роботи програми:



Мал. 5 – Завдання 5

**Завдання 6.**

Створіть клас "Сповіщення" (Notification), який дозволяє створювати спливаючі повідомлення на сторінці. Клас повинен мати методи для створення сповіщень з різними стилями та для їх закриття.

Код програми:

index.html

<!DOCTYPE html>

<html>

<head>

<title>Спливаючі повідомлення з кнопками закриття</title>

<script src="script.js"></script>

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

</head>

<body>

<button onclick="notifier.createNotification('success', 'Success message')">Повідомлення успіху</button>

<button onclick="notifier.createNotification('error', 'Error message')">Повідомлення про помилку</button>

<button onclick="notifier.createNotification('info', 'Info message')">Інформаційне повідомлення</button>

</body>

</html>

style.css

.notification {

position: fixed;

top: 20px;

left: 50%;

transform: translateX(-50%);

padding: 10px 20px;

border-radius: 5px;

color: #fff;

font-weight: bold;

display: none;

z-index: 999;

}

.success {

background-color: #4caf50;

}

.error {

background-color: #f44336;

}

.info {

background-color: #2196f3;

}

.closeBtn {

cursor: pointer;

float: right;

font-weight: normal;

margin-left: 10px;

}

script.js

class Notifications {

createNotification(type, message) {

const notification = document.createElement('div');

notification.classList.add('notification', type);

const closeBtn = document.createElement('span');

closeBtn.classList.add('closeBtn');

closeBtn.textContent = 'X';

closeBtn.addEventListener('click', () => {

document.body.removeChild(notification);

});

notification.appendChild(closeBtn);

const content = document.createElement('span');

content.textContent = message;

notification.appendChild(content);

document.body.appendChild(notification);

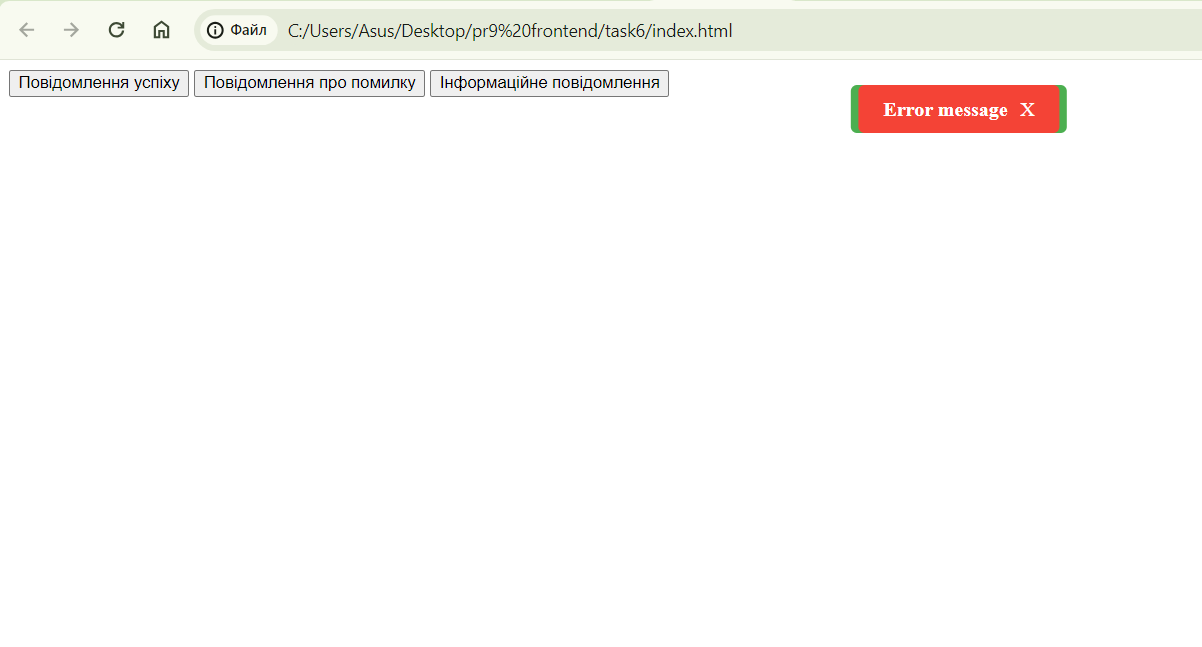
notification.style.display = 'block';

}

}

const notifier = new Notifications();

Результат роботи програми:



Мал. 6 – Завдання 6