

**Лабораторна робота №2**

З дисципліни

**“Web-дизайн”**

**Тема: “Розробка функціональності Web-додатка мовою JavaScript”**

Виконав: студент 3 курсу

ФПМ групи КВ-91

Зиков А.С.

Перевірив: Петрашенко А.В.

Київ - 2022р.

**Мета:** ознайомитись із засобами мови Javascript та навчитись їх застосовувати побудови Web-інтерфейсу користувача.

**Загальне завдання:** розробити функціональність для статичних сторінок Web-додатку першої лабораторної роботи із використанням шаблону MVC.

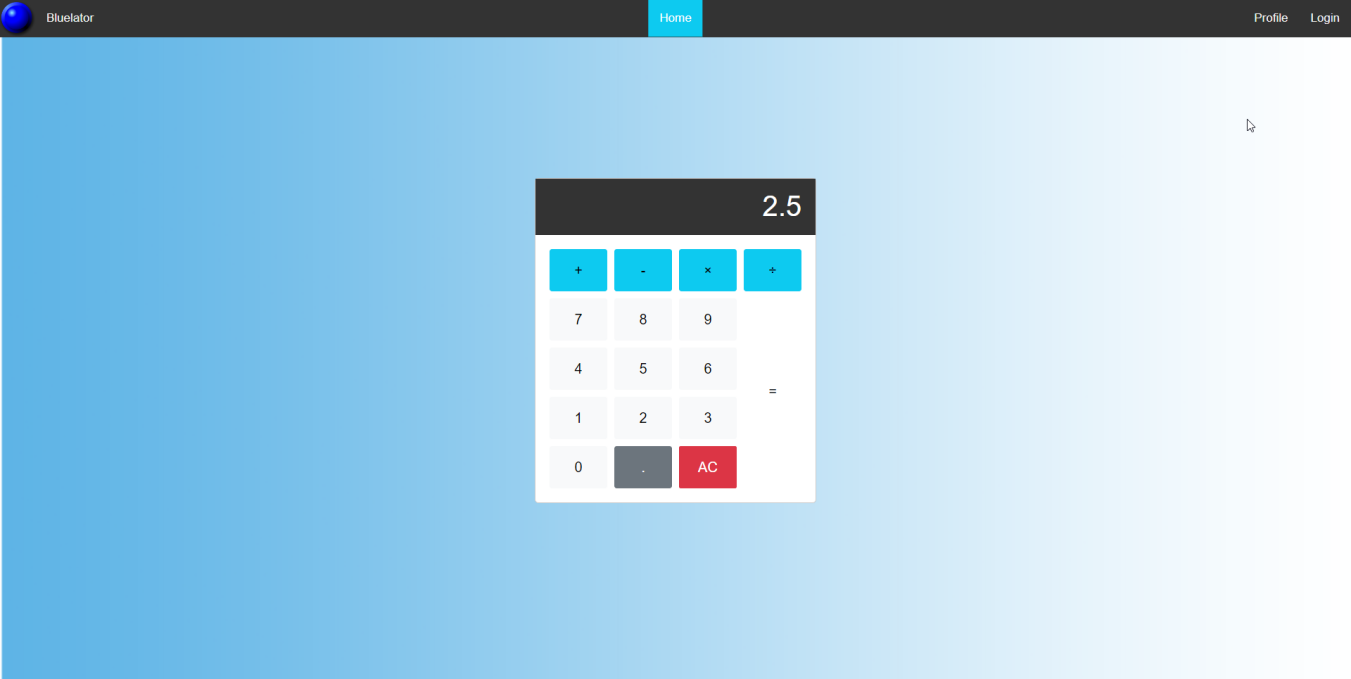
**Інструменти розробки:** мови HTML5, CSS3, Javascript.

**Розташування робіт: <https://github.com/Luritron/Web-design/tree/main/Lab2>**

**Виконання**

**Інформація:** за допомогою JavaScript було реалізовано calculator.js та login.js. Перший описує функціональність калькулятора: роботу з числами та виконання над ними базових операцій. Login.js описує вхід до аккаунту користувача за зазделегіть зазначенними файлами, що прописані у 10 строці: username == “admin” && password == “1234”. У противному випадку виникне помилка входу.

**Демонстрація**

Рис. 1 - демонстрація роботоспроможності калькулятора

Оскільки історії вводу попердніх даних та поточних в даному калькуляторі не реалізовано, то задля того щоб не надсилати декілька скріншотів демонстрації його обчислень, на даному рисунку проведена операція 2.5\*4=10. (ознайомитись власноруч із роботоспроможністю калькулятора можна в репозиторії другої лабораторної роботи на гітхабі за посиланням вище)

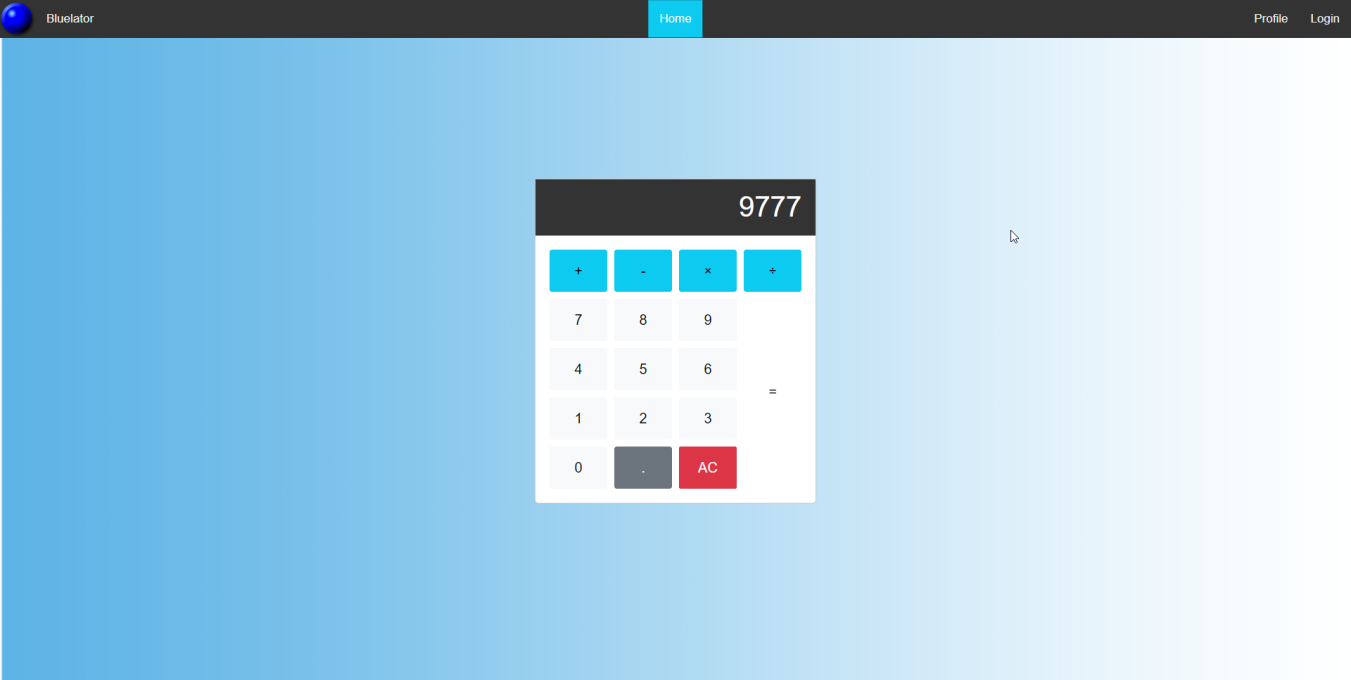


Рис. 2 - демонстрація роботоспроможності калькулятора

Схожа ситуація, в даному випадку була проведена операція 8989+788=9777.

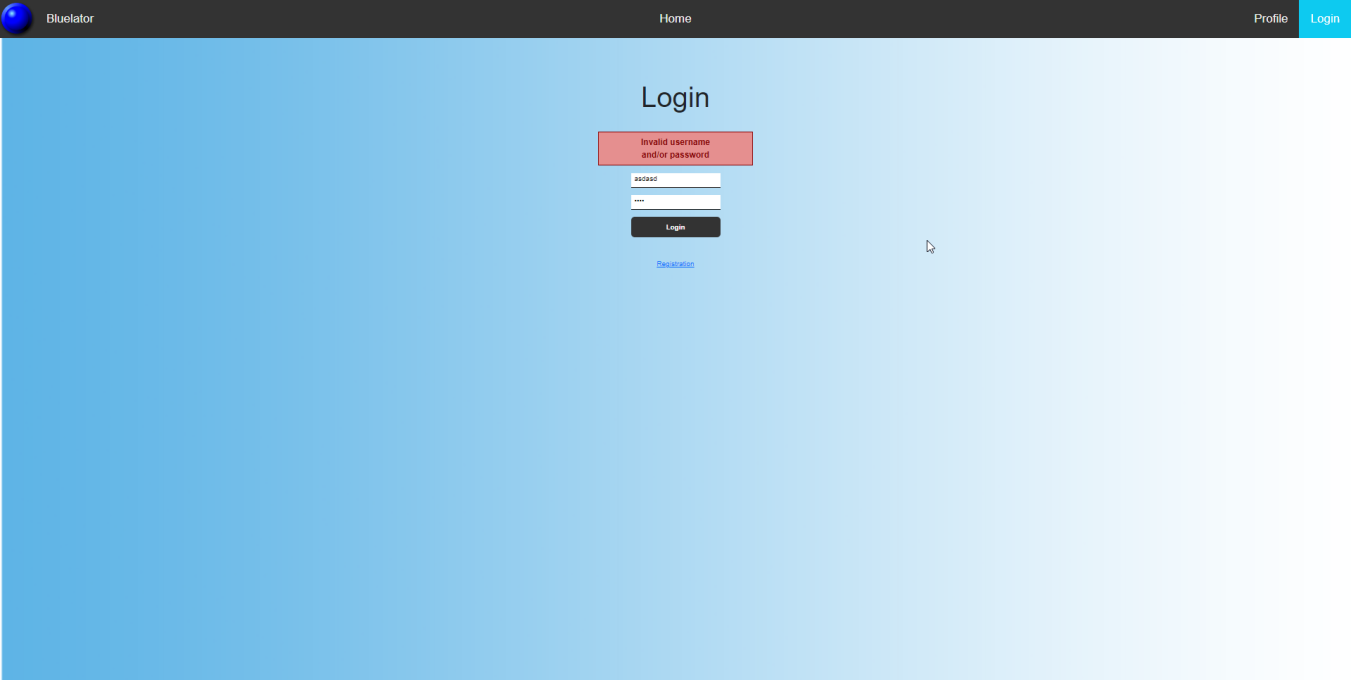


Рис. 3 - демонстрація некоретного введення даних у поле Login

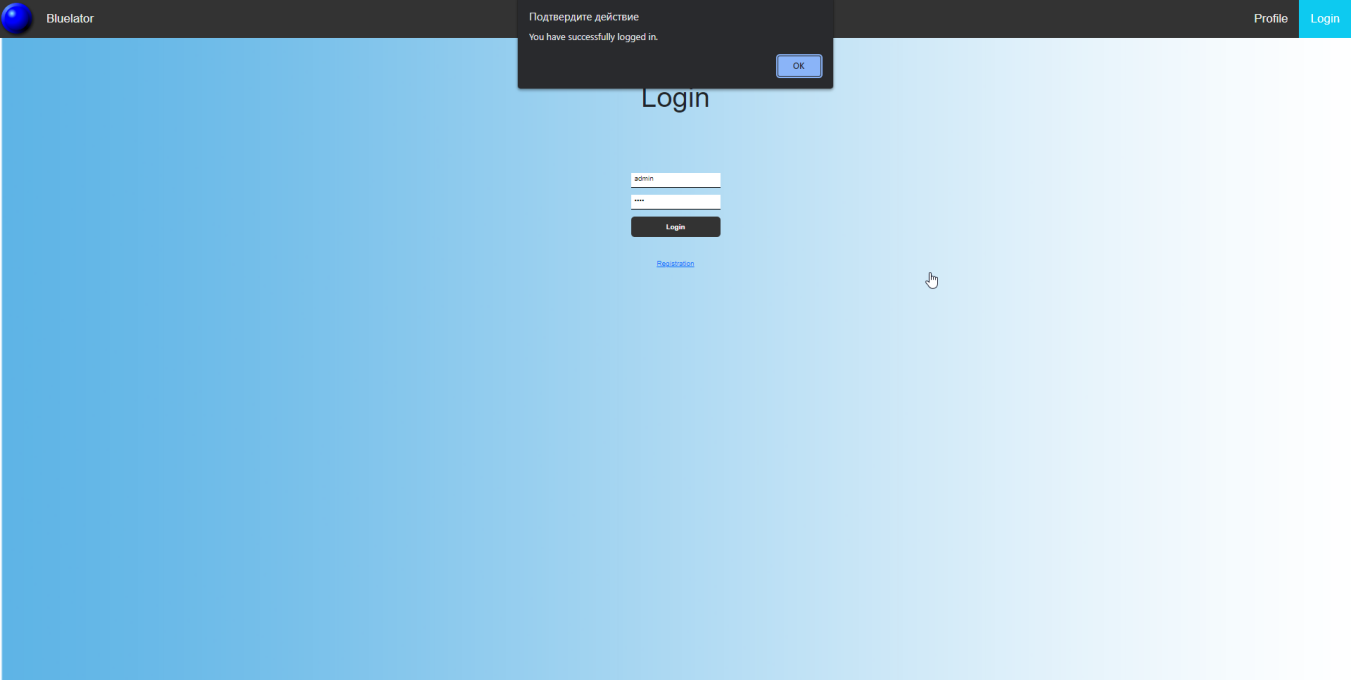


Рис. 4 - демонстрація коректного введення даних у поле Login

Приклад реалізації calculator.js:

const calculator = {

displayValue: '0',

firstOperand: null,

waitingForSecondOperand: false,

operator: null,

};

function inputDigit(digit) {

const { displayValue, waitingForSecondOperand } = calculator;

if (waitingForSecondOperand === true) {

calculator.displayValue = digit;

calculator.waitingForSecondOperand = false;

} else {

calculator.displayValue = displayValue === '0' ? digit : displayValue + digit;

}

}

function inputDecimal(dot) {

// If the `displayValue` does not contain a decimal point

if (!calculator.displayValue.includes(dot)) {

// Append the decimal point

calculator.displayValue += dot;

}

}

function handleOperator(nextOperator) {

const { firstOperand, displayValue, operator } = calculator

const inputValue = parseFloat(displayValue);

if (operator && calculator.waitingForSecondOperand) {

calculator.operator = nextOperator;

return;

}

if (firstOperand == null) {

calculator.firstOperand = inputValue;

} else if (operator) {

const currentValue = firstOperand || 0;

const result = performCalculation[operator](currentValue, inputValue);

calculator.displayValue = String(result);

calculator.firstOperand = result;

}

calculator.waitingForSecondOperand = true;

calculator.operator = nextOperator;

}

const performCalculation = {

'/': (firstOperand, secondOperand) => firstOperand / secondOperand,

'\*': (firstOperand, secondOperand) => firstOperand \* secondOperand,

'+': (firstOperand, secondOperand) => firstOperand + secondOperand,

'-': (firstOperand, secondOperand) => firstOperand - secondOperand,

'=': (firstOperand, secondOperand) => secondOperand

};

function resetCalculator() {

calculator.displayValue = '0';

calculator.firstOperand = null;

calculator.waitingForSecondOperand = false;

calculator.operator = null;

}

function updateDisplay() {

const display = document.querySelector('.calculator-screen');

display.value = calculator.displayValue;

}

updateDisplay();

const keys = document.querySelector('.calculator-keys');

keys.addEventListener('click', (event) => {

const { target } = event;

if (!target.matches('button')) {

return;

}

if (target.classList.contains('operator')) {

handleOperator(target.value);

updateDisplay();

return;

}

if (target.classList.contains('decimal')) {

inputDecimal(target.value);

updateDisplay();

return;

}

if (target.classList.contains('all-clear')) {

resetCalculator();

updateDisplay();

return;

}

inputDigit(target.value);

updateDisplay();

});

Приклад реалізації login.js:

const loginForm = document.getElementById("login-form");

const loginButton = document.getElementById("login-form-submit");

const loginErrorMsg = document.getElementById("login-error-msg");

loginButton.addEventListener("click", (e) => {

e.preventDefault();

const username = loginForm.username.value;

const password = loginForm.password.value;

if (username === "admin" && password === "1234") {

alert("You have successfully logged in.");

location.reload();

} else {

loginErrorMsg.style.opacity = 1;

}

})