### Internship Project Report

#### Project Title: Development of a Web-Based Calculator

#### Prepared By:

Ms.Aziza Azim Khalfe

#### Internship Position:

Web Development Intern

#### Company/Organization:

Unified Mentor

### Abstract

This report details the development of a web-based calculator as part of my internship project. The primary objective of this project was to design and implement a functional calculator using HTML, CSS, and JavaScript. This involved creating a user-friendly interface, ensuring responsive design, and incorporating basic arithmetic operations. The following sections will outline the project objectives, technologies used, detailed implementation, and conclusions drawn from the project.

### Table of Contents

1. Introduction
2. Project Objectives
3. Technologies Used
4. Implementation
   1. HTML Structure
   2. CSS Styling
   3. JavaScript Functionality
5. Results and Discussion
6. Conclusion

#### Introduction

This project report outlines the development of a simple web-based calculator using HTML, CSS, and JavaScript. The goal of this project was to create a functional and aesthetically pleasing calculator application that can perform basic arithmetic operations, clear the input, backspace for corrections, and evaluate expressions.

#### Project Objectives

1. Develop a user-friendly calculator interface.
2. Implement basic arithmetic operations: addition, subtraction, multiplication, and division.
3. Provide functionalities for clearing the display and backspacing.
4. Allow keyboard input to interact with the calculator.
5. Ensure responsiveness and cross-browser compatibility.

#### Technologies Used

* **HTML**: For creating the structure of the calculator.
* **CSS**: For styling the calculator and ensuring a responsive design.
* **JavaScript**: For implementing the calculator's functionality and interactivity.

**Implementation:**

#### HTML Structure

The HTML file defines the structure of the calculator, including the display area and the buttons for digits, operations, and functionalities like clear and backspace.

#### CSS Styling

The CSS file styles the calculator to ensure a modern and clean look. It includes styles for the overall layout, buttons, display area, and hover effects for better user experience.

#### JavaScript Functionality

The JavaScript file handles the calculator's logic, including appending numbers and operators to the display, evaluating expressions, and handling keyboard input.

**HTML File:**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Calculator</title>

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

</head>

<body>

<div id="calc">

<div id="content">

<form>

<div id="output">

<input type="text" id='res'>

</div>

<div class="btn">

<input type="button" value='C' onclick="Clear()" id="clear">

<input type="button" value='←' onclick="Back()" id="backspace">

<input type="button" value='%' onclick="Solve('%')">

<input type="button" value='/' onclick="Solve('/')">

<br>

<input type="button" value='7' onclick="Solve('7')">

<input type="button" value='8' onclick="Solve('8')">

<input type="button" value='9' onclick="Solve('9')">

<input type="button" value='x' onclick="Solve('\*')">

<br>

<input type="button" value='4' onclick="Solve('4')">

<input type="button" value='5' onclick="Solve('5')">

<input type="button" value='6' onclick="Solve('6')">

<input type="button" value='-' onclick="Solve('-')">

<br>

<input type="button" value='1' onclick="Solve('1')">

<input type="button" value='2' onclick="Solve('2')">

<input type="button" value='3' onclick="Solve('3')">

<input type="button" value='+' onclick="Solve('+')">

<br>

<input type="button" value='00' onclick="Solve('00')">

<input type="button" value='0' onclick="Solve('0')">

<input type="button" value='.' onclick="Solve('.')">

<input type="button" value='=' onclick="Result()">

</div>

</form>

</div>

</div>

<script src='index.js'></script>

</body>

</html>

**CSS File:**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

#calc {

width: 100%;

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

}

#content {

background: #2c302c;

padding: 20px;

border-radius: 10px;

}

#content form input {

border: 0;

outline: 0;

width: 50px;

height: 50px;

border-radius: 8px;

font-size: 15px;

margin: 10px;

cursor: pointer;

}

#backspace {

background-color: rgb(237, 89, 30);

color: white;

}

#res {

padding: 10px;

}

#clear {

background-color: rgb(237, 89, 30);

color: white;

}

form #output {

display: flex;

justify-content: flex-end;

margin: 15px 0;

}

form #output input {

text-align: right;

flex: 1;

font-size: 25px;

}

**JS File:**

function Solve(val) {

var v = document.getElementById('res');

v.value += val;

}

function Result() {

var num1 = document.getElementById('res').value;

try {

var num2 = eval(num1.replace('x', '\*'));

document.getElementById('res').value = num2;

} catch {

document.getElementById('res').value = 'Error';

}

}

function Clear() {

var inp = document.getElementById('res');

inp.value = '';

}

function Back() {

var ev = document.getElementById('res');

ev.value = ev.value.slice(0, -1);

}

document.addEventListener('keydown', function (event) {

const key = event.key;

const validKeys = '0123456789+-\*/.%';

if (validKeys.includes(key)) {

Solve(key === '\*' ? 'x' : key);

} else if (key === 'Enter') {

Result();

} else if (key === 'Backspace') {

Back();

} else if (key.toLowerCase() === 'c') {

Clear();

}

});

### Conclusion

This project successfully demonstrates the integration of HTML, CSS, and JavaScript to create a functional and interactive web-based calculator. The calculator is user-friendly and supports both button clicks and keyboard inputs, providing a comprehensive and responsive user experience.