**Date of Experiment and Submission: 07-03-25**

**SVKM’S NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

Department of Mechatronics Engineering

**RPA Lab**

Subject- Robotic Process Automation

**EXPERIMENT NO. 6**

**Objective:**

The objective of this experiment is to understand the capabilities of Microsoft Power Apps. This experiment involves creating a Calculator application.

**Prerequisites:**

1. Computers or laptops with access to Internet.
2. Power Apps account

**Theory:**

**1. Overview of PowerApps**

PowerApps is a low-code development platform by Microsoft that allows users to create custom applications without requiring extensive programming knowledge. It provides a drag-and-drop interface for designing apps and supports integration with various Microsoft and third-party services.

**2. Introduction to Basic Arithmetic Operations**

Arithmetic operations form the foundation of mathematical calculations in computing. The four fundamental operations are:

* **Addition (+)**: Combining two numbers to get their sum.
* **Subtraction (-)**: Finding the difference between two numbers.
* **Multiplication (×)**: Calculating the product of two numbers.
* **Division (÷)**: Splitting a number into equal parts.

In this experiment, we use PowerApps to perform these operations dynamically based on user inputs.

**3. Using PowerApps for Calculator Development**

PowerApps provides built-in formulas and functions to perform calculations on user inputs. We utilize:

* **Text Input Controls** to enter numbers.
* **Button Controls** to trigger operations.
* **Labels** to display results.
* **Formulas** (e.g., Value(Input1.Text) + Value(Input2.Text)) to process calculations.

**Task List:**

**1. Planning & Setup**

* Define requirements (addition, subtraction, multiplication, division)
* Open PowerApps and create a new canvas app

**2. Designing the UI**

* Add a **Text Input** control for user input (two number fields)
* Add **Buttons** for operations (+, -, ×, ÷, Clear)
* Add a **Label** to display the result

**3. Implementing Functionality**

Assign formulas to each operation button:

* Addition (Text(Result\_Label) = Value(Input1.Text) + Value(Input2.Text))
* Subtraction (Text(Result\_Label) = Value(Input1.Text) - Value(Input2.Text))
* Multiplication (Text(Result\_Label) = Value(Input1.Text) \* Value(Input2.Text))
* Division (Text(Result\_Label) = If(Value(Input2.Text) <> 0, Value(Input1.Text) / Value(Input2.Text), "Error"))
* Assign a formula to the **Clear** button (Reset(Input1); Reset(Input2); Reset(Result\_Label))

**4. Enhancements & Validations**

* Validate inputs (ensure only numbers are entered)
* Handle division by zero (show an error message)
* Format the output for better readability

**5. Testing & Debugging**

* Test each arithmetic operation
* Test edge cases (e.g., empty input, zero division)
* Fix any issues found

**6. Publishing & Deployment**

* Save and publish the app
* Share with users if needed

**Procedure Overview:**

**1. Creating a New PowerApps Canvas App**

1. Open PowerApps and create a new Canvas app.
2. Choose Blank App and select Tablet or Phone Layout based on preference.

**2. Designing the User Interface (UI)**

1. Add two Text Input controls for user number inputs.
2. Add four Button controls for operations (+, -, ×, ÷).
3. Add a Label to display the result.
4. Add a Clear button to reset inputs and results.
5. Customize the layout, button colors, and text sizes for better usability.

**3. Implementing Functionality with Formulas**

1. Assign PowerApps formulas to each operation button:
   * Addition: Set(Result, Value(Input1.Text) + Value(Input2.Text))
   * Subtraction: Set(Result, Value(Input1.Text) - Value(Input2.Text))
   * Multiplication: Set(Result, Value(Input1.Text) \* Value(Input2.Text))
   * Division: Set(Result, If(Value(Input2.Text) <> 0, Value(Input1.Text) / Value(Input2.Text), "Error"))
2. Set the Result Label to display Result.
3. Configure the Clear Button with Reset(Input1); Reset(Input2); Set(Result, "").

**4. Adding Input Validation & Enhancements**

1. Ensure only numbers are entered in input fields.
2. Prevent division by zero by displaying an error message.
3. Format the result label for better readability (e.g., decimal precision).

**5. Testing & Debugging**

1. Test each arithmetic operation.
2. Check how the app handles blank inputs and zero division.
3. Make UI/UX improvements based on test results.

**6. Publishing & Deployment**

1. Save the app in PowerApps.
2. Publish and share with users if needed.
3. Provide documentation or a short guide for users.

**Flow Screenshots:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Input Screenshots:**

Addition:

A screenshot of a computer

AI-generated content may be incorrect.

Subtraction:

A screenshot of a computer

AI-generated content may be incorrect.

Multiplication:

A screenshot of a computer

AI-generated content may be incorrect.

Division:

A screenshot of a computer

AI-generated content may be incorrect.

**Output Screenshot:**

Addition:

A screenshot of a computer

AI-generated content may be incorrect.

Subtraction:

A screenshot of a computer

AI-generated content may be incorrect.

Multiplication:

A screenshot of a computer

AI-generated content may be incorrect.

Division:

A screenshot of a computer

AI-generated content may be incorrect.

**Conclusion:**

Created and tested the working of a calculator app using Microsoft PowerApps successfully.