ABSTRACT

This mini project presents the design and development of a desktop-based calculator application using Python and the Tkinter library. The application performs basic arithmetic operations through a clean and user-friendly graphical interface. The project focuses on event-driven programming, GUI layout management, and error handling mechanisms.

METHODOLOGY

The project follows a structured software development approach including design, implementation, and testing phases.

Development Approach

Design Phase: UI layout and button placement were planned.
Implementation Phase: Core calculator logic was implemented using Python functions.
Testing Phase: The application was tested using valid and invalid inputs.

Technical Architecture

The system consists of a Tkinter-based GUI, event handlers, and a Python evaluation engine.

Algorithm Flow

User inputs expression → Expression displayed → Evaluate using eval() → Display result or error.

CODE EXPLANATION

The code is modular and easy to understand.

Library Import

Tkinter library is imported to create the graphical user interface.

Core Functions

Press Function: Inserts input values.
Clear Function: Clears the display.

Calculate Function: Evaluates the expression.

Main Window Configuration

The main application window is created using Tk().

Entry Widget Creation

Entry widget is used as the calculator display.

Button Grid Generation

Buttons are dynamically generated using grid layout.

Clear Button Creation

Clear and Backspace buttons are added for better usability.

Event Loop Execution

mainloop() keeps the application running.

FEATURES AND FUNCTIONALITY

The calculator provides efficient arithmetic computation.

Key Features

Simple UI, Backspace support, Error handling, Dark theme.

User Interface

The interface uses a dark theme with clear visual separation.

LIMITATIONS AND FUTURE ENHANCEMENTS

The current version has limited functionality.

Current Limitations

Supports only basic arithmetic operations.

Potential Enhancements

Scientific functions, keyboard input, history panel.

CONCLUSION

The project demonstrates effective use of Python Tkinter for GUI-based applications.