PROJECT: ERROR CALCULATOR

1. Identify a Real-World Problem or Need

In scientific experiments, engineering computations, and computer science applications, measurements often contain errors due to limitations of instruments, environmental conditions, and human factors. Understanding how much an observed value deviates from the true value is critical for:

- Accuracy analysis
- Performance evaluation
- Data reliability checks
- Experimental result validation

Students and professionals frequently need to compute Absolute Error, Relative Error, Percentage Error, etc., but doing this manually every time is slow and error-prone.

Thus, an Error Calculator is needed to automate error analysis accurately and efficiently.

2. Objectives and Expected Outcomes

Objectives

- To design a software tool that computes different types of measurement errors.
- To apply mathematical and computer science concepts to implement error formulas.

- To provide a fast and user-friendly interface for error calculation.
- To demonstrate structured programming, modularization, and algorithmic thinking.

Expected Outcomes

- A fully functional program that calculates:

✓Absolute Error

✓Relative Error

✓Percentage Error

✓True Error & Approximate Error

- A modular and well-tested system
- Increased understanding of numerical methods and accuracy analysis

□

3. Problem Definition

Measurements in real-world conditions deviate from true values. Engineers and scientists need to quantify errors to evaluate the reliability of measurements.

The goal is to develop a software tool that allows users to input:

- True value
- Measured/Approximate value

And automatically compute all major error metrics.

4. Requirement Analysis

4.1 Functional Requirements

- FR1: Accept input values (true value and measured value)
- FR2: Calculate Absolute Error
- FR3: Calculate Relative Error
- FR4: Calculate Percentage Error
- FR5: Generate a summary of error analysis
- FR6: Display results clearly
- FR7: Handle invalid input gracefully

4.2 Non-Functional Requirements

- Simple and easy-to-use interface
- High accuracy and precision
- Quick response time
- Code modularity and maintainability

4.3 System Requirements

- Python/Java/C/C++
- Any OS
- Standard math library support

6. Algorithm Development

Algorithm: Calculate Absolute Error

1. Start
2. Input true value (T)
3. Input measured value (M)

4. Absolute error = |T − M|
5. Display result
6. End

Algorithm: Calculate Relative Error

1. Start
2. Compute Absolute Error
3. Relative error = Absolute Error ÷ True value
4. Display result
5. End

Algorithm: Percentage Error

1. Start
2. Compute Relative Error
3. Percent Error = Relative Error × 100
4. Display result
5. End

Refinements

- Add GUI (Tkinter version available on request)
- Add graphing of errors
- Support batch error calculation
- Generate PDF of results
- Include scientific rounding