

# Measurement Website

BCP104 – Programming with Python

Department of Computer Science

Accra Technical University

Date: September 17, 2025

## Group Members

Name	Index Number	Contribution
Lincoln Karikari Boateng Bekoe Adjei Richmond	01245314B 01245877B	Designed frontend, backend and Testing

# 1 Introduction

This project is a Flask-based web application designed to serve as a measurement toolkit. The app provides two main features: area calculation of geometric shapes and unit conversion between metric and imperial systems.

**Problem Statement:** Often, users require a quick and simple tool to compute areas and perform unit conversions without resorting to manual calculations or external services.

**Objectives:**

- To provide an interactive platform for calculating areas of common shapes.
- To implement a unit conversion system for length, mass, area, volume, and temperature.
- To design a responsive and user-friendly web interface.

## 2 Requirements

### 2.1 Functional Requirements

- Calculate area of square, rectangle, circle, triangle, and trapezoid.
- Convert units for length, mass, area, volume, and temperature.
- Prevent invalid input such as negative values (except for temperature).
- Provide a help modal displaying supported units.

### 2.2 Non-Functional Requirements

- Usability: Dark-themed and responsive interface.
- Performance: Real-time calculations without page reloads.
- Compatibility: Works across modern browsers.

## 3 System Overview

The web application consists of:

- A **frontend interface** built with HTML, CSS, and JavaScript that provides calculators and interactive input validation.
- A **backend API** implemented in Flask that handles area formulas and unit conversion logic.

Target users include students, engineers, and professionals who need quick access to measurement tools.

## 4 Technical Design

The system follows a client-server architecture:

- The **frontend** collects inputs, validates them, and sends requests using JavaScript's `fetch` API.
- The **backend** exposes two endpoints: `/api/area` for area calculations and `/api/unit` for conversions.
- Conversion logic uses reference dictionaries that normalize units into base SI units (meters, kilograms, etc.).

## 5 Technologies Used

- **Python 3.8+** with **Flask** for backend APIs.
- **HTML5, CSS3, JavaScript** for the frontend.
- **Math module** in Python for circle and trapezoid calculations.
- No external database is required since calculations are in-memory.

## 6 Implementation Details

### 6.1 Core Logic

- Area functions defined in Python implement standard geometric formulas.
- Conversion dictionaries (e.g., `LENGTH_TO_M`) provide base values for unit transformation.
- JavaScript dynamically enables/disables input fields depending on the selected shape.

### 6.2 Challenges

- Preventing negative values required input event listeners.
- Handling temperature conversions needed special logic since it is not a linear scale like other units.

## 7 Testing & Evaluation

- Manual tests were conducted by inputting different values into the calculators.
- Unit tests of formulas confirmed correctness of calculations.
- Edge cases (e.g., zero values, invalid units) were tested to ensure proper error handling.

Limitations include the absence of persistent data storage and no support for very complex units.

## 8 Conclusion & Future Work

### Lessons Learned:

- Integration of frontend and backend components using Flask.
- Use of JavaScript for real-time interactivity and validation.

### Future Improvements:

- Add support for more shapes (e.g., polygons, ellipses).
- Provide user history of past calculations.
- Extend to include additional measurement systems.

## 9 References

- Flask Documentation: <https://flask.palletsprojects.com/>
- MDN Web Docs for JavaScript: <https://developer.mozilla.org/>
- Python Math Library: <https://docs.python.org/3/library/math.html>