Project Documentation — Student Grade Calculator

1. Project Overview

The Student Grade Calculator is a Python program that allows users to enter a student's name and marks, and then calculates the grade based on predefined conditions. The program also displays an encouraging message along with the grade.

2. Objectives

- Understand if-elif-else statements

- Handle user input

- Validate numeric input and ranges

- Use functions

- Use loops to retry invalid input

- Display formatted output with messages

3. Technical Requirements Completed

- input() used for user input

- Variables for storing inputs

- if-elif-else used for grading logic

- Function calculate_grade() added

- Input validation (0-100)

- while loop for invalid input retry

- Friendly output with emojis

4. Grading Logic

90-100: A (Excellent! Keep it up! ■)

80-89: B (Very Good! Keep it up! ■)

70-79: C (Good job! You can do even better! ■)

60-69: D (You passed! Keep learning! ■)

0-59: F (Don't give up! Try harder next time! ■)

5. Code Structure Explanation

(a) Function: calculate_grade() holds logic and returns grade + message

(b) Input Handling: while True loop ensures valid input

(c) Output Formatting: name.upper() used for emphasis

6. Tools Required

- Python 3.x

- Text editor (Notepad / VS Code / etc)

- Command prompt to run

7. Setup & Installation Instructions

1. Install Python 3

2. Verify using: python --version

3. Save file as: grade_calculator.py

4. Run using: python grade_calculator.py

8. Test Cases

Includes:

- Valid inputs

- Boundary tests

- Invalid inputs

- Multiple attempts

9. Challenges & Solutions

- Non-numeric input → Used try-except ValueError

- Out-of-range marks → Added 0 <= marks <= 100 check

- Prevent crash → Used while True loop

10. What I Learned

- Conditional statements

- Input validation

- Functions for modular code

- Loops for control flow

- Clean output formatting

- Testing methodology

11. Conclusion

The Student Grade Calculator meets all requirements and demonstrates practical use of Python fundamentals including decision logic, functions, loops, input handling, and formatted output.