K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

NAME – DRISHTI

ROLL NO. – 2501730232

SUBJECT – PROGRAMMING FOR PROBLEM SOLVING USING PYTHON

SUBMITTED TO – MR. SAMEER FAROOQ

PROJECT 2

1. Project Objective

The goal of this project was to develop a **command-line-based GradeBook Analyzer** in Python that allows instructors to:

- Input student marks either manually or via a CSV file
- Perform statistical analysis (average, median, highest, lowest)
- Assign letter grades (A–F)
- Generate pass/fail reports and grade distribution
- Display results in a clean, formatted table

This tool demonstrates core Python concepts including file handling, data structures, functions, error handling, and user interaction.

2. Key Features Implemented

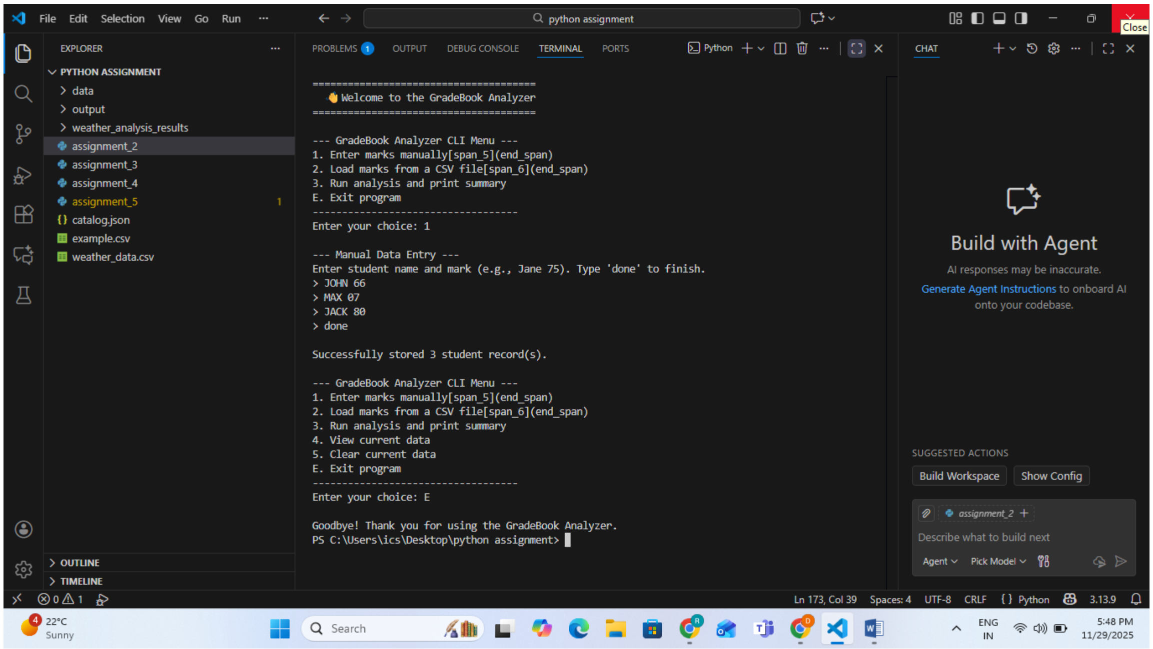
Feature	Description
Manual Data Entry	Interactive input supporting multi-word names and validation (0–100 range)
CSV File Import	Robust loading of student data from .csv files with error handling
Statistical Analysis	Calculates mean, median, max, and min scores using built-in and statistics module
Grade Assignment	Converts numerical scores to letter grades (A: ≥90, B: 80–89, ..., F: <60)
Grade Distribution	Counts number of students in each grade category
Pass/Fail Classification	Customizable threshold (set to ≥40); lists passed and failed students
Detailed Results Table	Neatly formatted output with aligned columns (Name, Marks, Grade)
Interactive CLI Menu	User-friendly loop with options to load, analyze, view, clear, or exit
Command-Line Argument Support	Auto-load CSV if filename provided at startup (python gradebook.py data.csv)

3. Technical Implementation Highlights

- **Data Structure:** Used a global dictionary marks with student name (key) → score (value) for efficient storage and lookup.
- **Modular Design:** Separated concerns into functions:
  - Data input (manual_data_entry, load_csv_data)
  - Analysis (calculate_average, calculate_median, etc.)

  - Grading (assign_grade, calculate_grades_and_distribution)
  - Display (run_analysis)
- **Robust Error Handling:**
  - Validates score range (0–100)
  - Handles invalid CSV formats, missing files, non-integer marks
  - Graceful exit on EOF (Ctrl+D/Z)
- **Code Quality:**
  - Comprehensive docstrings
  - Use of f-strings for clean formatting
  - List comprehensions for pass/fail filtering
  - Sorted output for consistent reporting

SCREEN SHORT



```
======================================
  👋 Welcome to the GradeBook Analyzer
======================================

--- GradeBook Analyzer CLI Menu ---
1. Enter marks manually[span_5](end_span)
2. Load marks from a CSV file[span_6](end_span)
3. Run analysis and print summary
E. Exit program
-----------------------------------
Enter your choice: 1

--- Manual Data Entry ---
Enter student name and mark (e.g., Jane 75). Type 'done' to finish.
> JOHN 66
> MAX 07
> JACK 80
> done

Successfully stored 3 student record(s).

--- GradeBook Analyzer CLI Menu ---
1. Enter marks manually[span_5](end_span)
2. Load marks from a CSV file[span_6](end_span)
3. Run analysis and print summary
4. View current data
5. Clear current data
E. Exit program
-----------------------------------
Enter your choice: E

Goodbye! Thank you for using the GradeBook Analyzer.
PS C:\Users\ics\Desktop\python assignment>
```

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.