AI ASSISTED CODING

Assignment-3.2

N. Madhuvani

2303A52117

Batch-40

Task Description-1

• Progressive Prompting for Calculator Design: Ask the AI to design a simple calculator

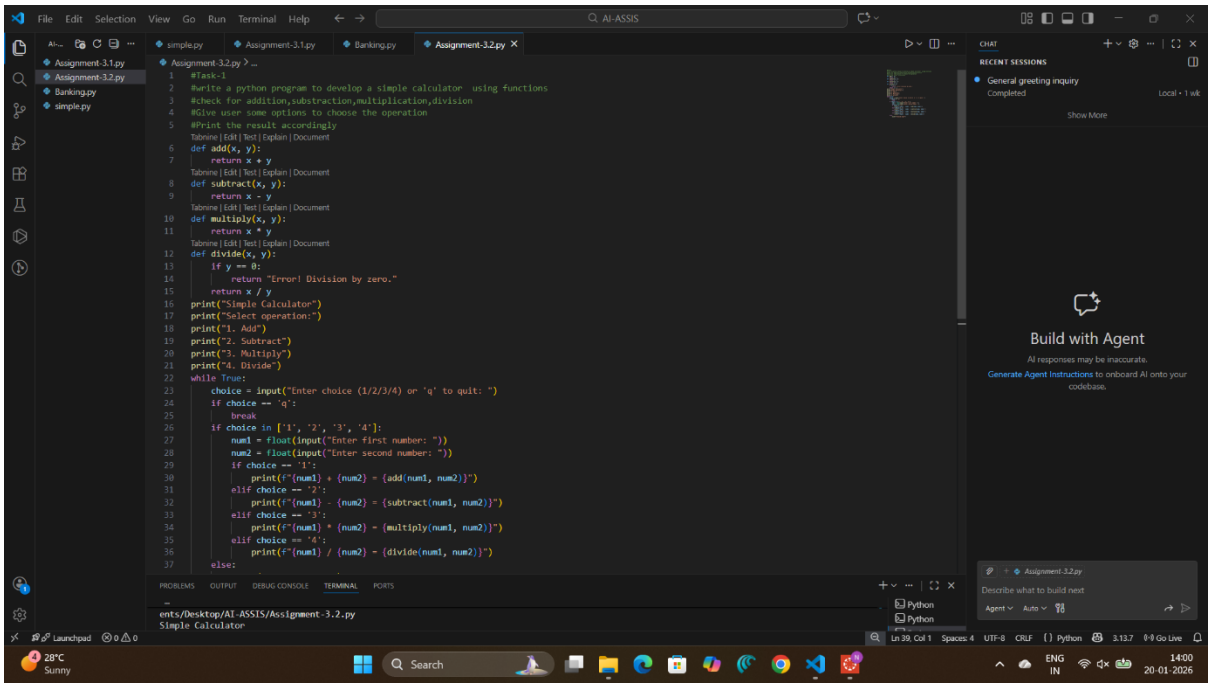
program by initially providing only the function name. Gradually enhance the prompt by

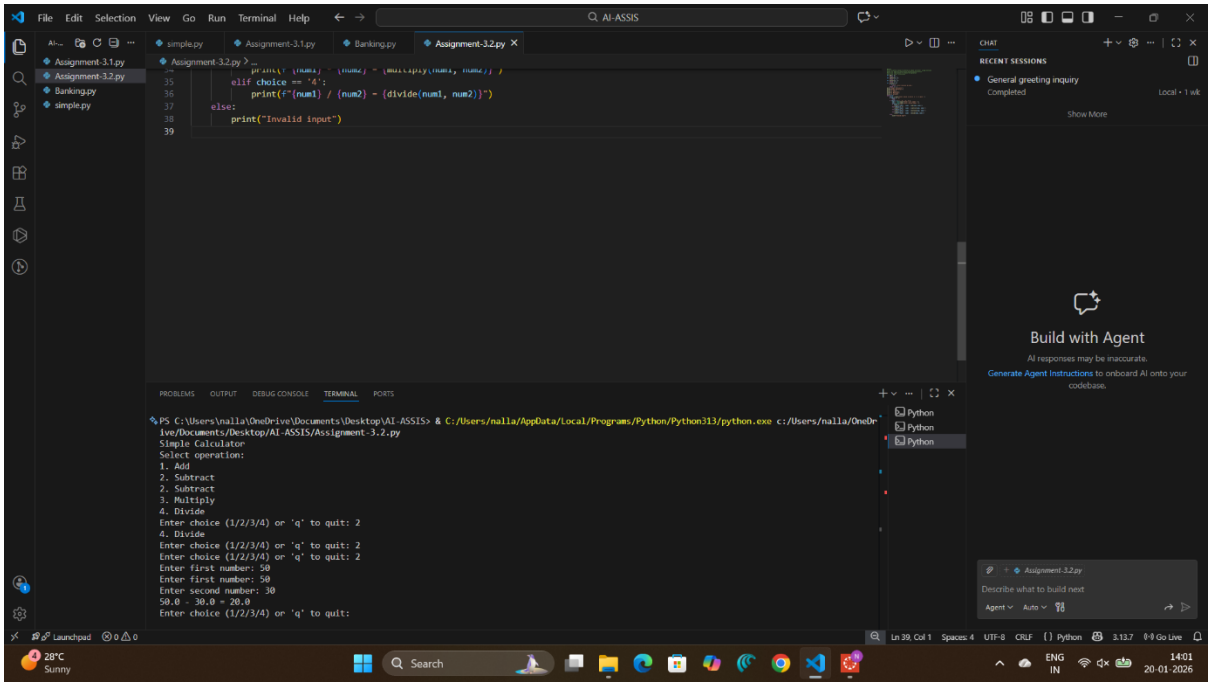
adding comments and usage examples.

Expected Output-1

• Comparison showing improvement in AI-generated calculator logic and structure.

Code and Output:





Analysis:

Analysis: Progressive Prompting for Calculator Design

Progressive prompting means giving instructions step by step to the AI.

First, only the function name is given, so the output is basic.

Next, comments are added to explain what the calculator should do.

Then, usage examples are included to show how the program works.

Each step improves clarity and accuracy of the AI response.

This method helps in getting a complete and correct calculator program.

Task Description-2

• Refining Prompts for Sorting Logic: Start with a vague prompt for sorting student marks,

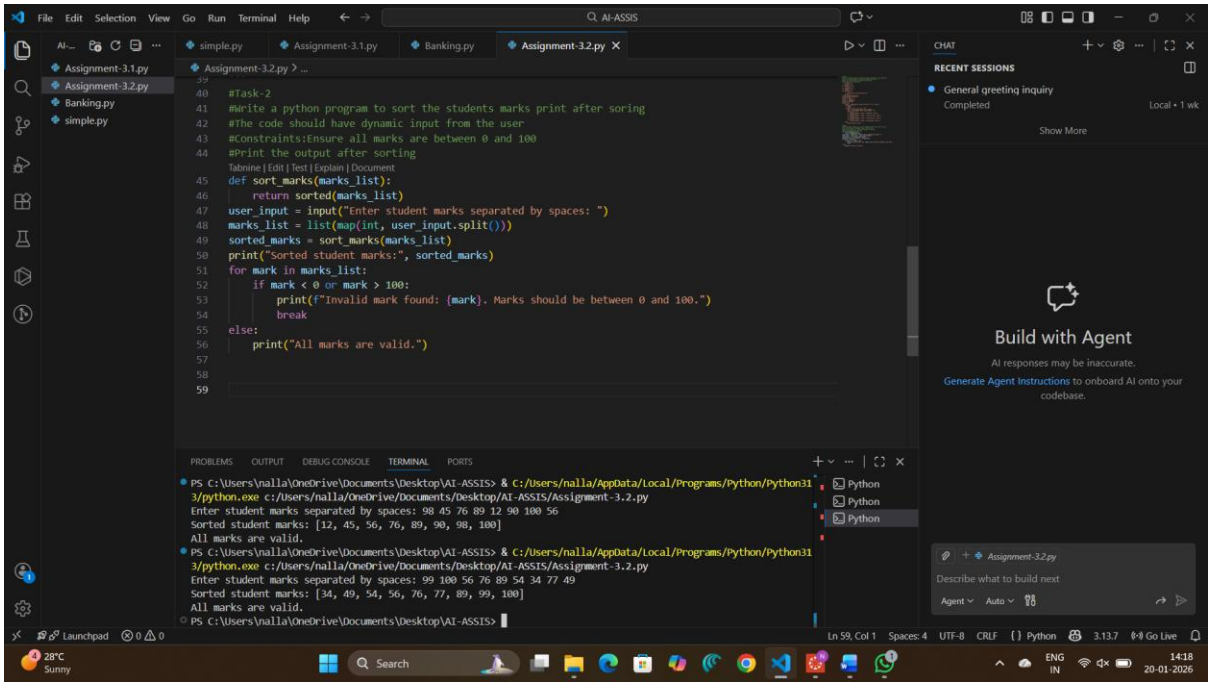
then refine it to clearly specify sorting order and constraints.

Expected Output-2

• AI-generated sorting function evolves from ambiguous logic to an accurate and efficient

implementation.

Code and Output:



Analysis:

Refining Prompts for Sorting Logic

Initially, a vague prompt is given to sort student marks.

Due to less clarity, the AI may give an incomplete or unclear solution.

The prompt is then refined to specify sorting order (ascending or descending).

Additional constraints (such as valid marks or number of students) are added.

Clear instructions help the AI produce accurate and expected output.

Refining prompts reduces confusion and errors in the solution.

Task Description-3

• Few-Shot Prompting for Prime Number Validation: Provide multiple input-output

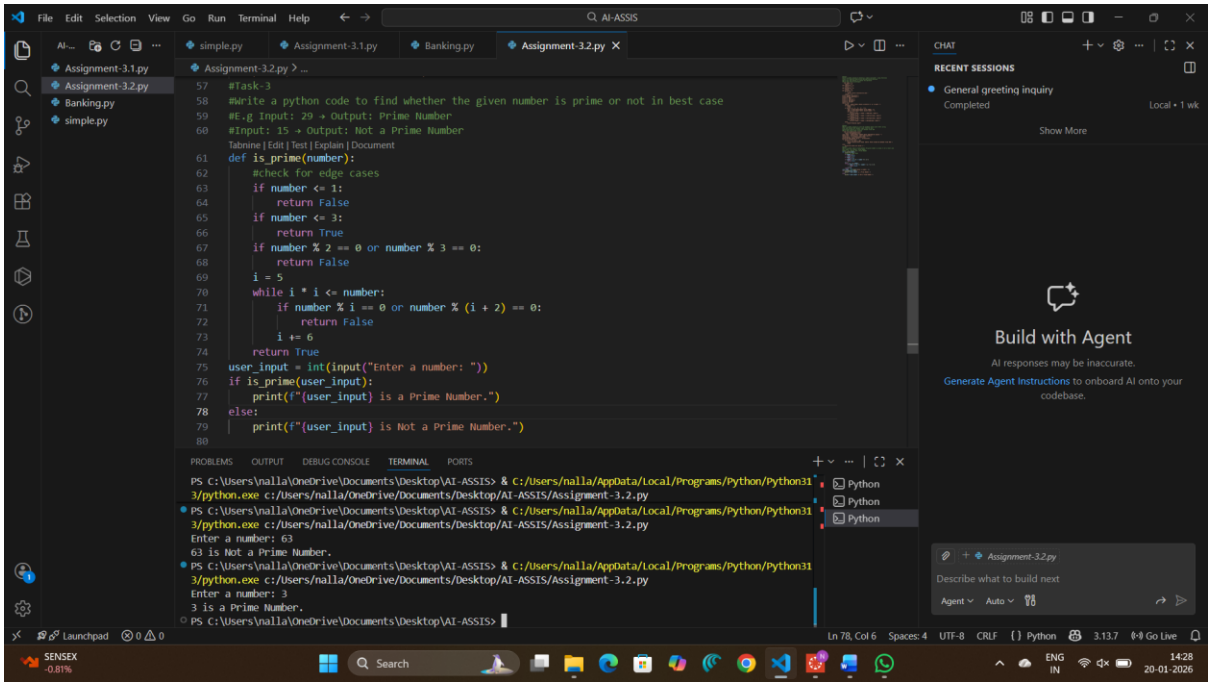
examples for a function that checks whether a number is prime. Observe how few-shot

prompting improves correctness.

Expected Output-3

• Improved prime-checking function with better edge-case handling.

Code and Output:



Analysis:

Few-Shot Prompting for Prime Number Validation

Few-shot prompting means giving multiple input-output examples to the AI.

The examples show how prime and non-prime numbers behave.

This helps the AI understand the exact logic of prime number checking.

The AI produces more correct and reliable results.

Errors are reduced compared to a prompt with no examples.

Few-shot prompting improves accuracy and consistency of the function.

Task Description-4

• Prompt-Guided UI Design for Student Grading System: Create a user interface for a

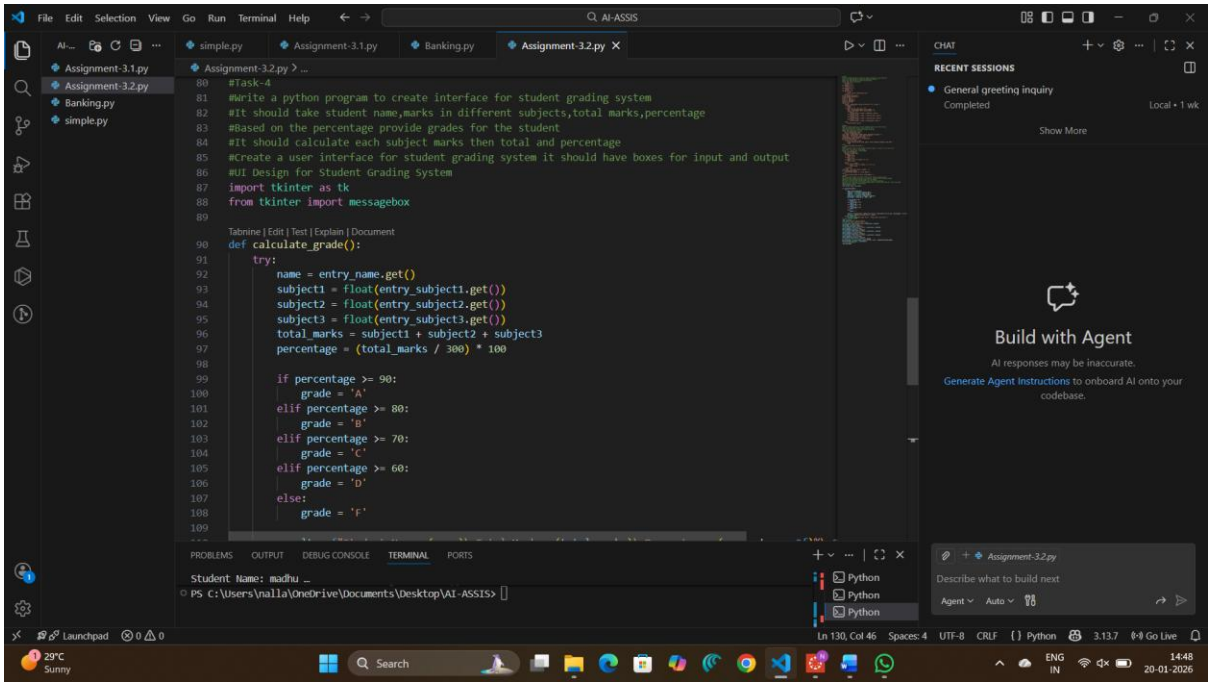
student grading system that calculates total marks, percentage, and grade based on user

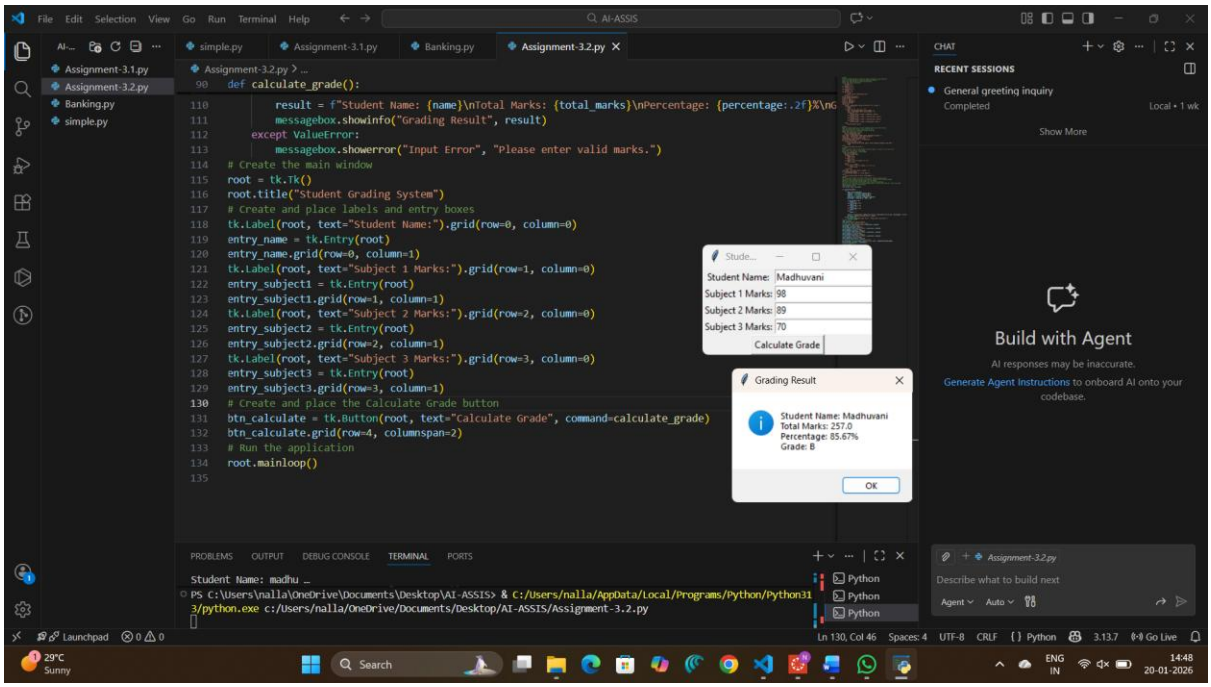
input.

Expected Output-4

• Well-structured UI code with accurate calculations and clear output display.

Code and Output:





Analysis:

Prompt-Guided UI Design for Student Grading System

Prompt-guided design gives clear instructions for UI creation.

The prompt specifies inputs like student marks.

The system calculates total marks and percentage automatically.

It assigns grades based on rules given in the prompt.

Clear prompts help design a user-friendly interface.

Prompt guidance reduces errors and confusion in UI behavior.

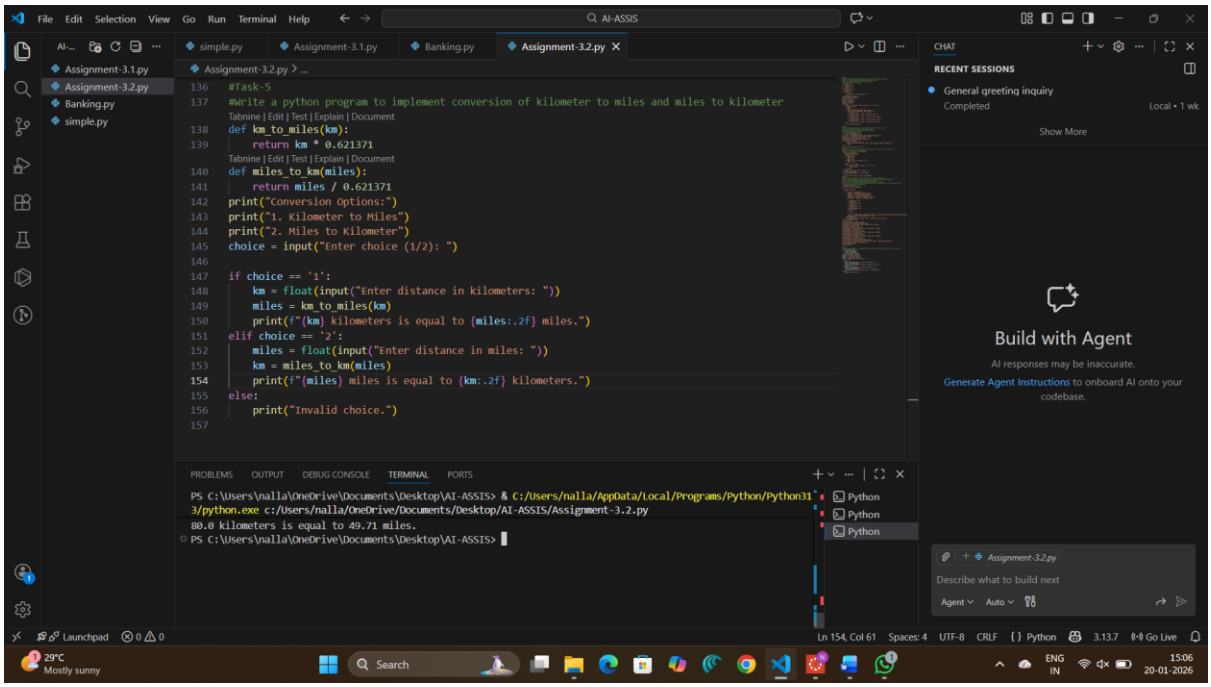
Task Description-5

• Analyzing Prompt Specificity in Unit Conversion Functions: Improving a Unit Conversion Function (Kilometers to Miles and Miles to Kilometers) Using Clear Instructions.

Expected Output-5

• Analysis of code quality and accuracy differences across multiple prompt variations.

Code and Output:



Analysis:

Prompt Specificity in Unit Conversion Functions

Prompt specificity means giving clear and detailed instructions.

Initially, a vague prompt may cause incorrect or incomplete conversion.

Clear instructions specify conversion type (km to miles, miles to km).

The prompt defines input and output format clearly.

Specific prompts improve accuracy and correctness of the function.

This reduces errors and misunderstanding in unit conversion.