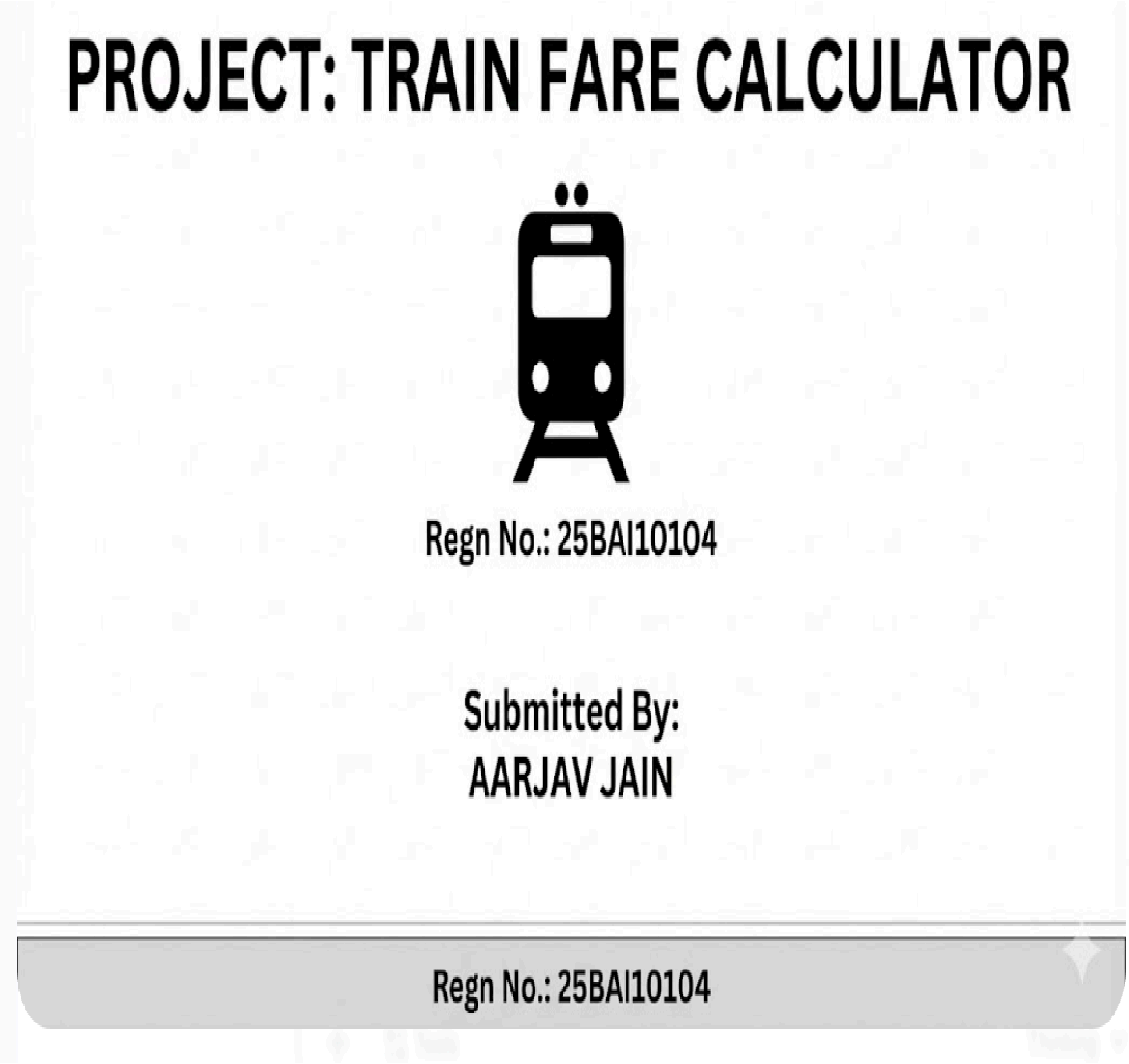


PROJECT: TRAIN FARE CALCULATOR

Regn No.: 25BAI10104

Submitted By:
AARJAV JAIN

Regn No.: 25BAI10104

1. Introduction

This project is a simple Train Fare Calculator made using Python.
The program takes the distance to travel and the coach type from the user, then calculates the fare.
It also applies discount options like Student, Senior Citizen, or Military and adds 5% GST to the final amount.
The project is made using basic Python concepts like input, output, functions, and simple calculation.

2. Problem Statement

People often want to estimate train travel cost quickly.
Manual calculation becomes confusing because different coach classes have different rates and different type of people has different discounts,
So, this project solves the problem by calculating the train fare in a simple python program.

3. Objectives

To design a simple and easy-to-understand calculator for train fare.

To use Three major functional modules.

To implement discounts and GST calculation.

To learn how to take user input and display output clearly.

To add a logical workflow of how the user interacts with the system

4. Functional Requirements

The system must perform the following functions:

1. Take Input:

Distance (in km)

Coach selection (1–5) between the options

2. Calculate Base Fare:

General = ₹0.5 per km

Sleeper = ₹1.5 per km

3AC = ₹3.0 per km

2AC = ₹4.0 per km

1AC = ₹5.0 per km

3. Apply Discounts:

Student → 10%

Senior Citizen → 40%

Military → 20%

Otherwise no discount

4. Add GST:

5% GST on discounted fare

5. Show Output:

Final total fare

5. Non-Functional Requirements

Usability: Very easy to operate .

Reliability: Correct fare, discount, and GST should be applied.

Maintainability: Code is modular and clear.

Performance: Should respond quickly for any valid in

6. System Architecture

Modules used in the program:

get_details() → Input

basic_fare() → Base fare calculation

apply_offer() → For discount calculation

add_tax() → For GST calculation

show_total() → Output display

7. Design Diagrams

7.1 Use Case Diagram

Use Cases:

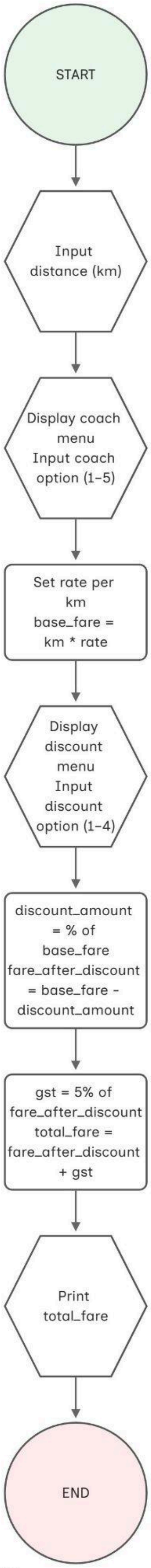
Enter distance

Select coach type

Select discount

View total fare

7.2 workflow diagram


START
Input distance (km)
Display coach menu
Input coach option (1–5)
Set rate per km
base_fare = km * rate
Display discount menu
Input discount option (1–4)
discount_amount = % of base_fare
fare_after_discount = base_fare - discount_amount
gst = 5% of fare_after_discount
total_fare = fare_after_discount + gst
Print total_fare
END

8. Implementation Details

Programming Language: Python

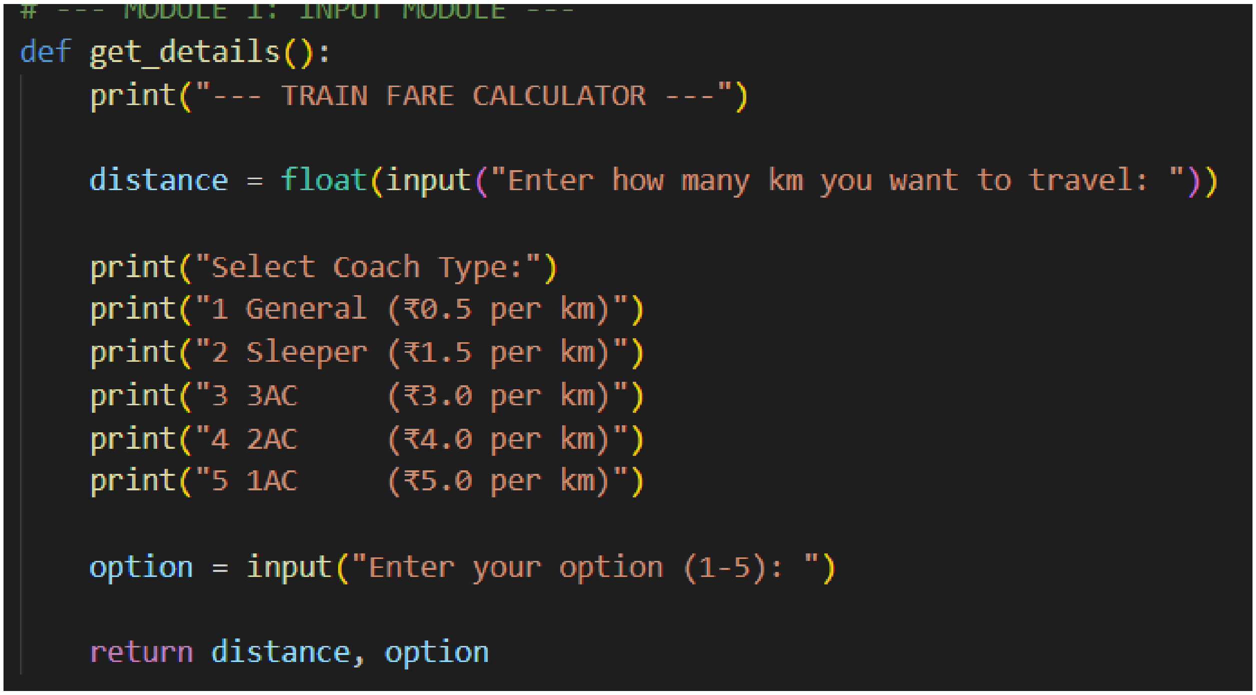
File Name: train_fare.py

Input: distance, coach, discount

Output: final calculated fare

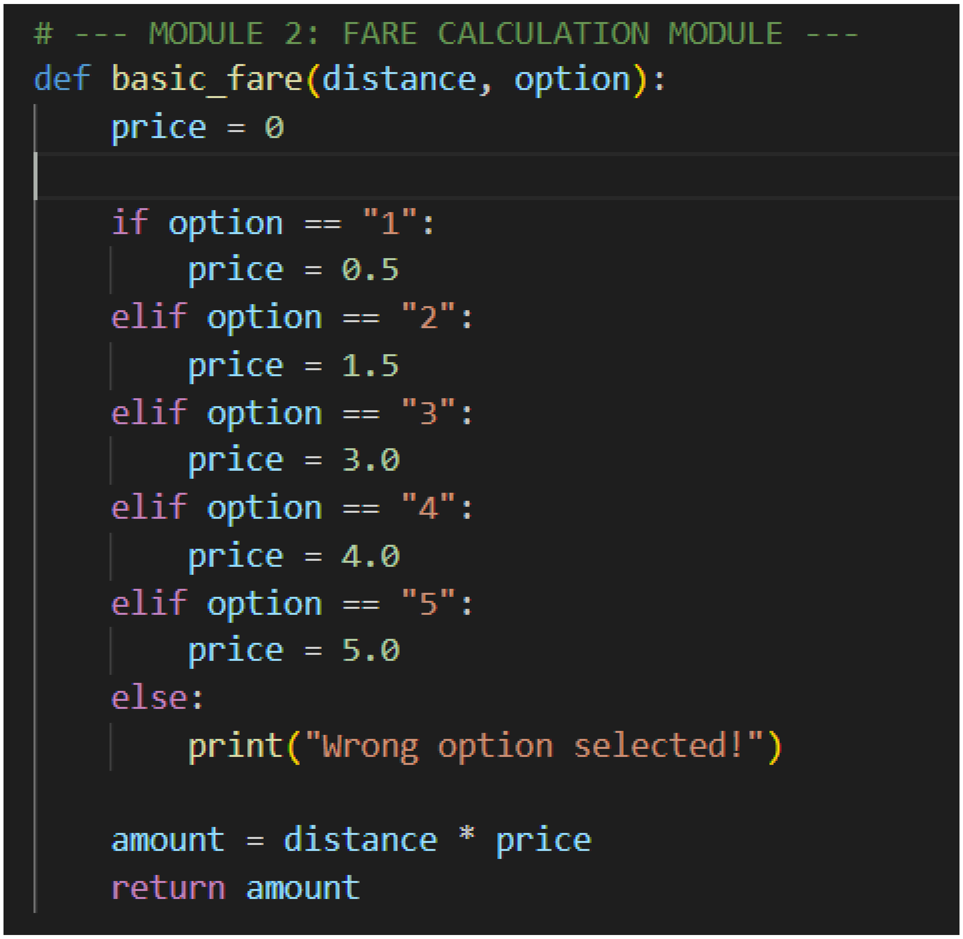
10.SCREENSHOT/RESULT

MODULE1 : INPUT MODULE



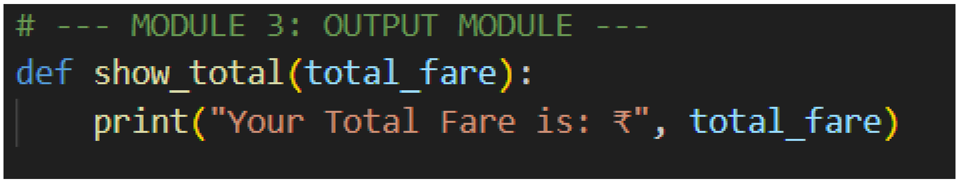
```python
# --- MODULE 1: INPUT MODULE ---
def get_details():
    print("--- TRAIN FARE CALCULATOR ---")

    distance = float(input("Enter how many km you want to travel: "))

    print("Select Coach Type:")
    print("1 General (₹0.5 per km)")
    print("2 Sleeper (₹1.5 per km)")
    print("3 3AC     (₹3.0 per km)")
    print("4 2AC     (₹4.0 per km)")
    print("5 1AC     (₹5.0 per km)")

    option = input("Enter your option (1-5): ")

    return distance, option
```

MODULE2 : FARE CALCULATION MODULE



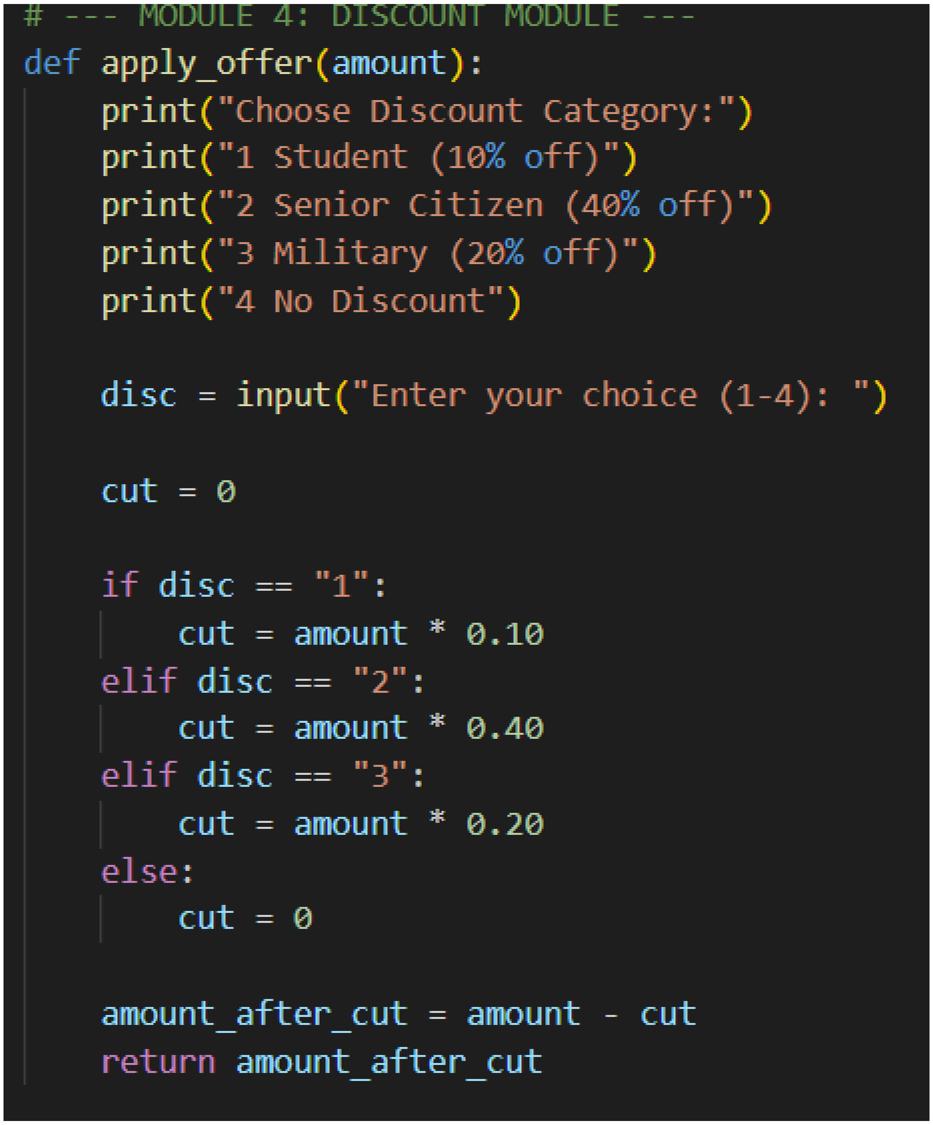
```
# --- MODULE 2: FARE CALCULATION MODULE ---
def basic_fare(distance, option):
    price = 0

    if option == "1":
        price = 0.5
    elif option == "2":
        price = 1.5
    elif option == "3":
        price = 3.0
    elif option == "4":
        price = 4.0
    elif option == "5":
        price = 5.0
    else:
        print("Wrong option selected!")

    amount = distance * price
    return amount
```

MODULE3: OUTPUT MODULE



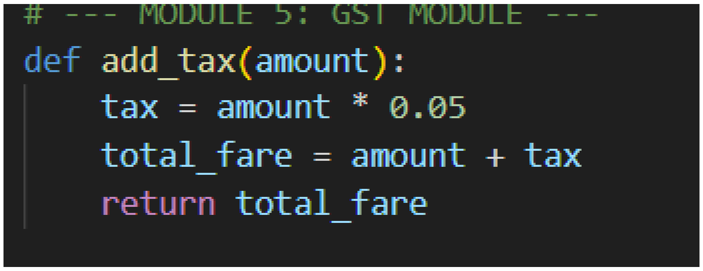
```
# --- MODULE 3: OUTPUT MODULE ---
def show_total(total_fare):
    print("Your Total Fare is: ₹", total_fare)
```

MODULE4: DISCOUNT MODULE



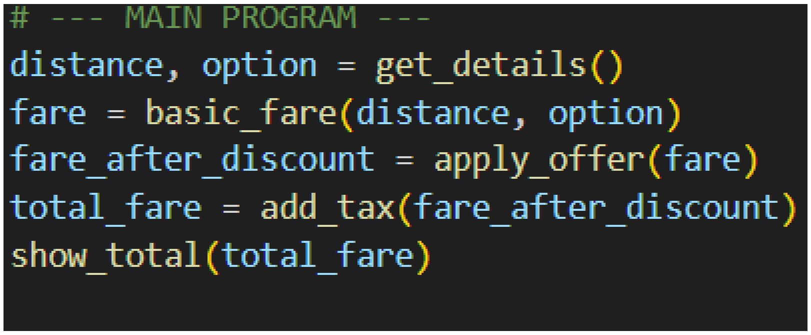
```python
# --- MODULE 4: DISCOUNT MODULE ---
def apply_offer(amount):
    print("Choose Discount Category:")
    print("1 Student (10% off)")
    print("2 Senior Citizen (40% off)")
    print("3 Military (20% off)")
    print("4 No Discount")

    disc = input("Enter your choice (1-4): ")

    cut = 0

    if disc == "1":
        cut = amount * 0.10
    elif disc == "2":
        cut = amount * 0.40
    elif disc == "3":
        cut = amount * 0.20
    else:
        cut = 0

    amount_after_cut = amount - cut
    return amount_after_cut
```

MODULE4: GST MODULE



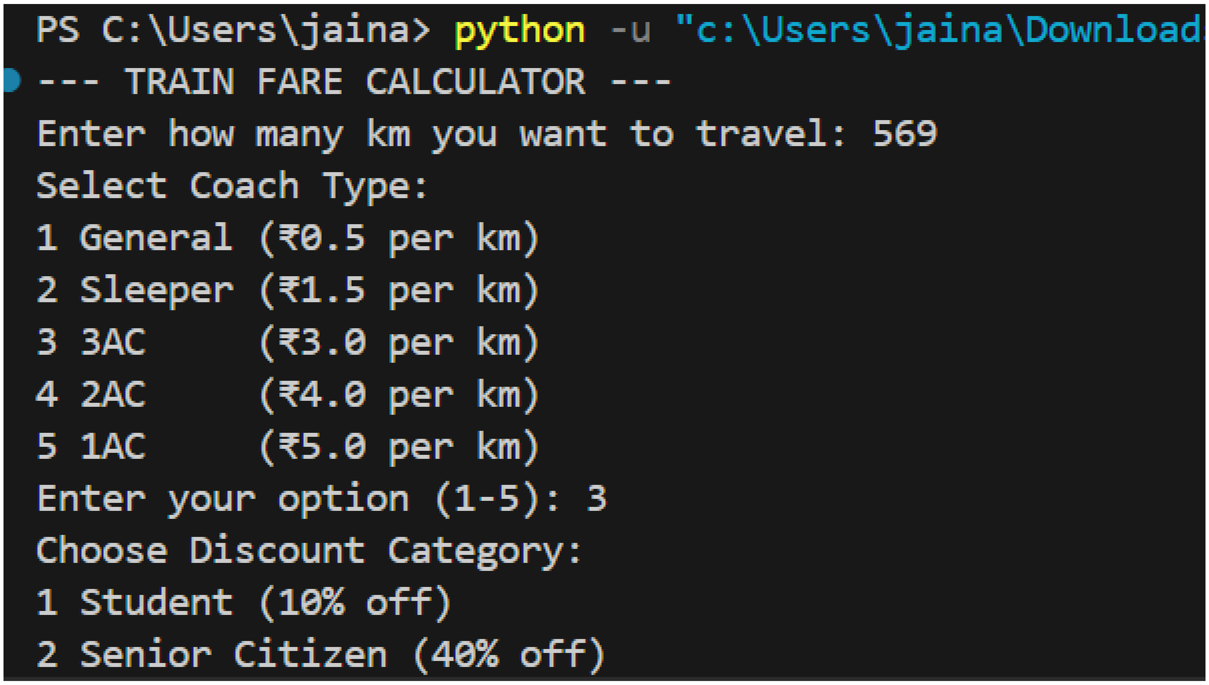
```python
# --- MODULE 5: GST MODULE ---
def add_tax(amount):
    tax = amount * 0.05
    total_fare = amount + tax
    return total_fare
```

MAIN PROGRAM:

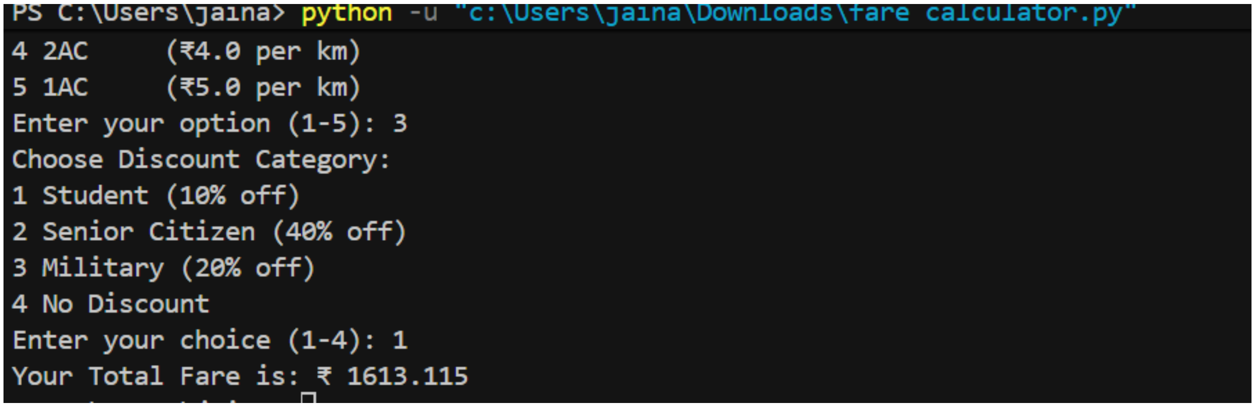


```
# --- MAIN PROGRAM ---
distance, option = get_details()
fare = basic_fare(distance, option)
fare_after_discount = apply_offer(fare)
total_fare = add_tax(fare_after_discount)
show_total(total_fare)
```

RESULTS : FOR EXAMPLE WE HAVE TO TRAVEL 569 KM IN 3RD AC COACH AND I AM A STUDENT SO IT WILL CALCULATE THE FARE AFTER DISCOUNTING 10% AND ADDING GST TO IT



```
PS C:\Users\jaina> python -u "c:\Users\jaina\Download
--- TRAIN FARE CALCULATOR ---
Enter how many km you want to travel: 569
Select Coach Type:
1 General (₹0.5 per km)
2 Sleeper (₹1.5 per km)
3 3AC     (₹3.0 per km)
4 2AC     (₹4.0 per km)
5 1AC     (₹5.0 per km)
Enter your option (1-5): 3
Choose Discount Category:
1 Student (10% off)
2 Senior Citizen (40% off)
```



```
PS C:\Users\jaina> python -u "c:\Users\jaina\Downloads\fare calculator.py"
4 2AC     (₹4.0 per km)
5 1AC     (₹5.0 per km)
Enter your option (1-5): 3
Choose Discount Category:
1 Student (10% off)
2 Senior Citizen (40% off)
3 Military (20% off)
4 No Discount
Enter your choice (1-4): 1
Your Total Fare is: ₹ 1613.115
```

11. Testing Approach

Manual test cases were performed using:

By entering different distances

Each coach type

All discount categories

GST verified using formula:
total = (fare − discount) × 1.05

12. Challenges Faced

Handling multiple functions for each module

Creating proper flowchart and diagrams

Understanding discount logic

Ensuring calculations are correct

13. Learnings

Learn to use the major module

Learn to use Python input/output usage

Learn to use Fare, discount, and GST calculation

Learn to Creating basic diagrams for documentation

14. Future Enhancements

Add multiple passenger support

Save ticket details to a file

Handle wrong input using loops

15. References

Vityarthi

Classroom notes

Online Python beginner tutorials