2124UCEM1041\_Gaurav\_Bodkhe

Unit 2 : case study

**Problem Statement=**You are to develop a system that should allow users to add, view, update, delete, and search student records, which include the student's name, roll number, and marks. Additionally, the system should generate a grade report based on marks, assigning grades like A, B, and C. You need to implement error handling for invalid inputs such as non-numeric marks or incorrect roll numbers. The data should be stored efficiently in a dictionary or list, and the system should be able to handle multiple student records.

**Code=**

# Dictionary to store student records

students = {}

# Function to add a student record

def add\_student():

try:

roll\_number = int(input("Enter Roll Number: "))

if roll\_number in students:

print("Error: Roll number already exists!")

return

name = input("Enter Name: ")

marks = float(input("Enter Marks: "))

if marks < 0 or marks > 100:

print("Error: Marks must be between 0 and 100!")

return

students[roll\_number] = {"Name": name, "Marks": marks}

print("Student added successfully!")

except ValueError:

print("Error: Invalid input! Roll number must be an integer, and marks must be a number.")

# Function to view all student records

def view\_students():

if not students:

print("No student records found!")

else:

print("\nStudent Records:")

for roll\_number, details in students.items():

print(f"Roll Number: {roll\_number}, Name: {details['Name']}, Marks: {details['Marks']}")

# Function to update a student record

def update\_student():

try:

roll\_number = int(input("Enter Roll Number to update: "))

if roll\_number not in students:

print("Error: Roll number not found!")

return

name = input("Enter new Name (leave blank to keep current): ")

marks = input("Enter new Marks (leave blank to keep current): ")

if name:

students[roll\_number]["Name"] = name

if marks:

try:

marks = float(marks)

if marks < 0 or marks > 100:

print("Error: Marks must be between 0 and 100!")

return

students[roll\_number]["Marks"] = marks

except ValueError:

print("Error: Marks must be a number!")

return

print("Student record updated successfully!")

except ValueError:

print("Error: Roll number must be an integer!")

# Function to delete a student record

def delete\_student():

try:

roll\_number = int(input("Enter Roll Number to delete: "))

if roll\_number not in students:

print("Error: Roll number not found!")

return

del students[roll\_number]

print("Student record deleted successfully!")

except ValueError:

print("Error: Roll number must be an integer!")

# Function to search for a student record

def search\_student():

try:

roll\_number = int(input("Enter Roll Number to search: "))

if roll\_number not in students:

print("Error: Roll number not found!")

return

student = students[roll\_number]1

print(f"\nStudent Found:")

print(f"Roll Number: {roll\_number}, Name: {student['Name']}, Marks: {student['Marks']}")

except ValueError:

print("Error: Roll number must be an integer!")

# Function to generate a grade report

def generate\_grade\_report():

if not students:

print("No student records found!")

else:

print("\nGrade Report:")

for roll\_number, details in students.items():

marks = details["Marks"]

if marks >= 90:

grade = "A"

elif marks >= 75:

grade = "B"

elif marks >= 60:

grade = "C"

else:

grade = "D"

print(f"Roll Number: {roll\_number}, Name: {details['Name']}, Marks: {details['Marks']}, Grade: {grade}")

# Main menu

def main():

while True:

print("\nStudent Management System")

print("1. Add Student")

print("2. View Students")

print("3. Update Student")

print("4. Delete Student")

print("5. Search Student")

print("6. Generate Grade Report")

print("7. Exit")

choice = input("Enter your choice: ")

if choice == "1":

add\_student()

elif choice == "2":

view\_students()

elif choice == "3":

update\_student()

elif choice == "4":

delete\_student()

elif choice == "5":

search\_student()

elif choice == "6":

generate\_grade\_report()

elif choice == "7":

print("Exiting the system. Goodbye!")

break

else:

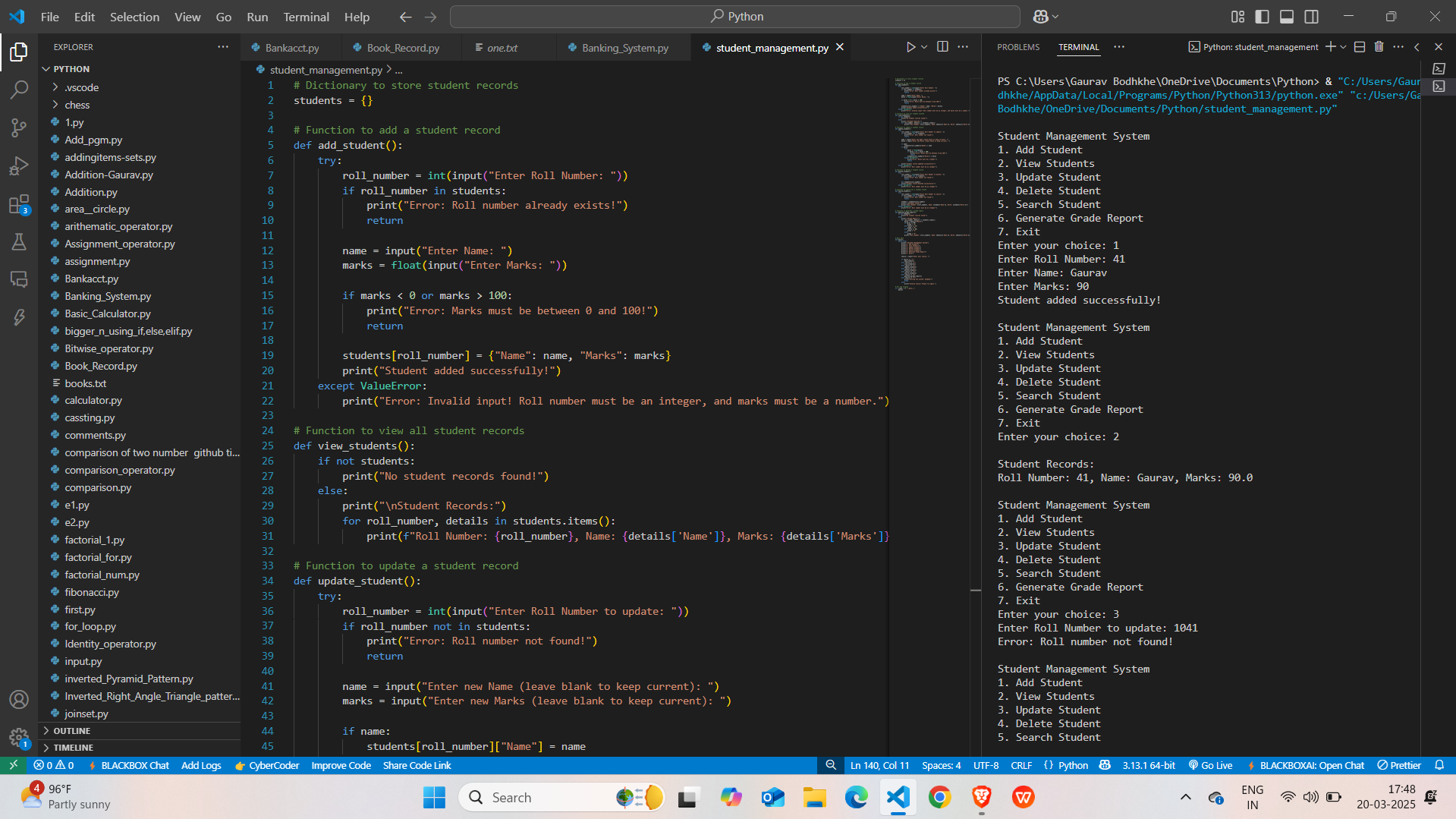
print("Invalid choice! Please try again.")

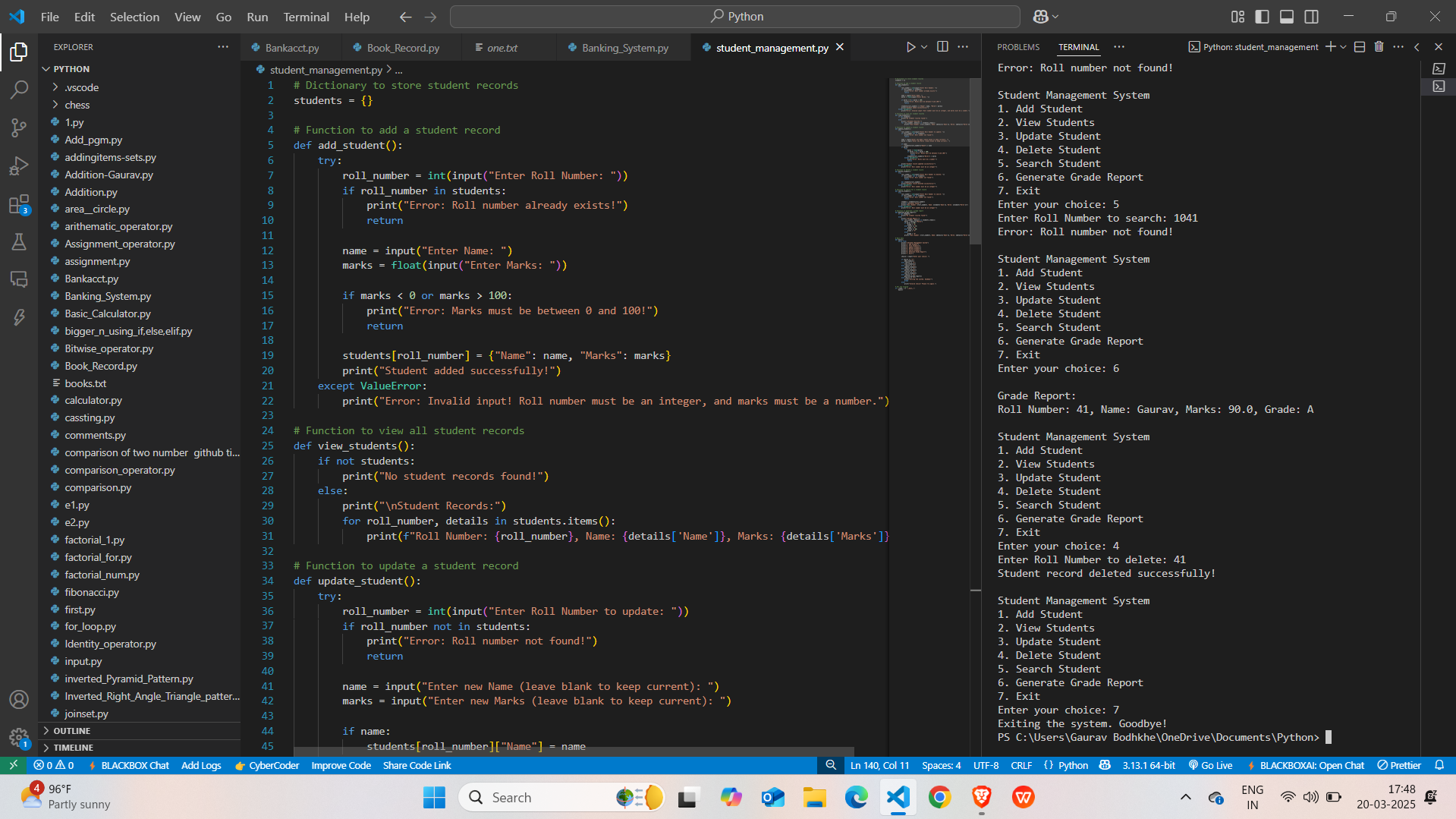
# Run the program

if \_\_name\_\_ == "\_\_main\_\_":

main()

**O/P=**

****

****