2124UCEM1041\_Gaurav\_Bodkhe

Unit 3 : case study

**Problem Statement=**

A company maintains a large database of book records in plain text files, including details such as the title, author, publication year, and price. The management has tasked you with automating certain tasks to improve efficiency. Create a book records text file named “books.txt”. Each line in the file should represent a book, with fields separated by commas (e.g., Title,Author,Year,Price). Using the concepts of File handling using python implement the following: Read data from the text file, allow users to search for books by title or author and display matching records, Provide an option to update details of a book, Append new book details to the file without overwriting existing records.

**Code=**

# Function to read data from the file

def read\_books():

books = []

try:

with open("book.txt", "r") as file:

# Skip the header line

next(file)

for line in file:

title, author, year, price = line.strip().split(",")

books.append({

"Title": title,

"Author": author,

"Year": int(year),

"Price": float(price)

})

return books

except FileNotFoundError:

print("Error: File 'books.txt' not found!")

return []

# Function to search for books by title or author

def search\_books(books):

search\_term = input("Enter title or author to search: ").lower()

results = []

for book in books:

if search\_term in book["Title"].lower() or search\_term in book["Author"].lower():

results.append(book)

if results:

print("\nSearch Results:")

for book in results:

print(f"Title: {book['Title']}, Author: {book['Author']}, Year: {book['Year']}, Price: ${book['Price']:.2f}")

else:

print("No matching books found.")

# Function to update details of a book

def update\_book(books):

title = input("Enter the title of the book to update: ")

found = False

for book in books:

if book["Title"].lower() == title.lower():

print(f"Current Details: {book}")

new\_author = input("Enter new author (leave blank to keep current): ")

new\_year = input("Enter new year (leave blank to keep current): ")

new\_price = input("Enter new price (leave blank to keep current): ")

if new\_author:

book["Author"] = new\_author

if new\_year:

try:

book["Year"] = int(new\_year)

except ValueError:

print("Error: Year must be an integer!")

return

if new\_price:

try:

book["Price"] = float(new\_price)

except ValueError:

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

return

found = True

print("Book details updated successfully!")

break

if not found:

print("Book not found!")

# Function to append new book details to the file

def append\_book():

title = input("Enter title: ")

author = input("Enter author: ")

year = input("Enter year: ")

price = input("Enter price: ")

try:

year = int(year)

price = float(price)

except ValueError:

print("Error: Year must be an integer, and price must be a number!")

return

with open("books.txt", "a") as file:

file.write(f"\n{title},{author},{year},{price}")

print("New book added successfully!")

# Main menu

def main():

while True:

print("\nBook Records Management System")

print("1. Search Books")

print("2. Update Book Details")

print("3. Add New Book")

print("4. Exit")

choice = input("Enter your choice: ")

if choice == "1":

books = read\_books()

search\_books(books)

elif choice == "2":

books = read\_books()

update\_book(books)

# Write updated data back to the file

with open("books.txt", "w") as file:

file.write("Title,Author,Year,Price\n")

for book in books:

file.write(f"{book['Title']},{book['Author']},{book['Year']},{book['Price']}\n")

elif choice == "3":

append\_book()

elif choice == "4":

print("Exiting the system. Goodbye!")

break

else:

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

# Run the program

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

main()

**Book.txt=**

Title,Author,Year,Price

The Great Gatsby,F. Scott Fitzgerald,1925,15.99

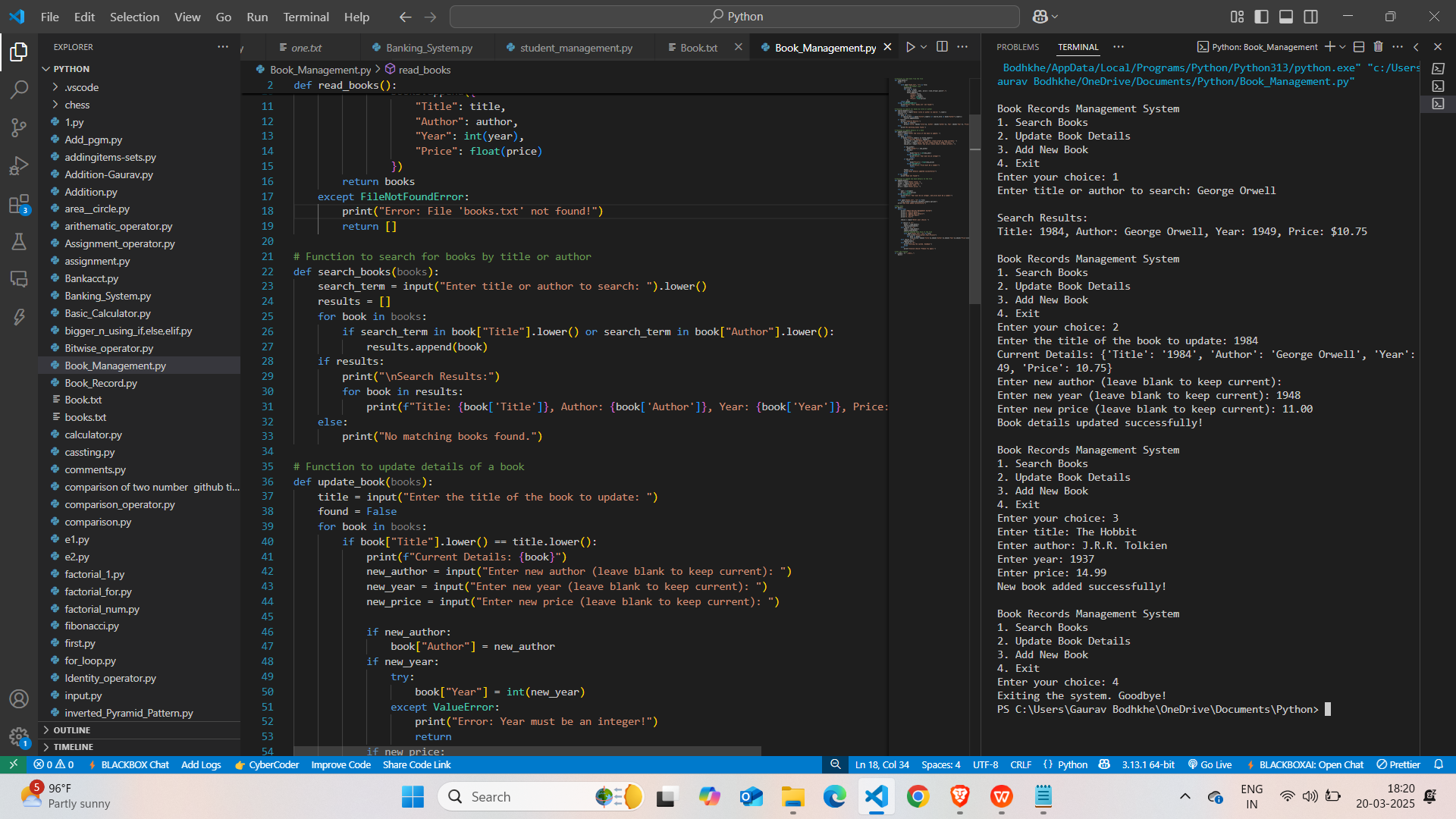
To Kill a Mockingbird,Harper Lee,1960,12.50

1984,George Orwell,1949,10.75

Pride and Prejudice,Jane Austen,1813,9.99

The Catcher in the Rye,J.D. Salinger,1951,11.25

**O/P=**

****