Name: Karnishwar Ravikrishna (20086868)

Course Title: M.Sc. Cyber Security

Lecturer Name: Swati Dongre

Module/Subject Title: Advanced Programming Techniques

Assignment Title: CA_ONE_(30%)

No. of Words: 740

Table Of Contents:

Introduction:

This report is demonstrated how I implemented the coding tasks given.

This report consists of two parts, in which the first part consists of two tasks that need to be completed with **C#** coding, and the second part consists of two tasks that needs to be completed with **Python** coding.

In PART I the first task given to me was to create a phone contact book that allows users to add, update, view, and delete contacts. Important concepts from object-oriented programming, including classes, objects, encapsulation, properties, and method overloading, were used in the development of this system. The second task was to create a C# program to identify different file extension types.

In PART II the first task given to me was to create Creating a client-server application via TCP is the first Python task. Before returning a unique registration number, the server saves the admission data that the client gathers from applicants in a database. The second task was to Web Scrap two Hotel sites and get the information like room rent and return all the details as a CSV files.

The report also includes Screenshots of the outputs of all the codes and also the link for GitHub repository used for this project.

Part 1 – C# Programming:

Question 1:

In this task, I was asked to create a Contact Book program where a user can add a contact, view all contacts, search for a contact, update details, and delete a contact. The program needed to store at least 20 contacts and validate that the mobile number entered is a non-zero, 9-digit number. I used Object-Oriented Programming by creating a Contact class to hold the details and a PhoneBook class to manage all contacts. I also used method overloading and properties to show encapsulation. A list was used to store multiple Contact objects.

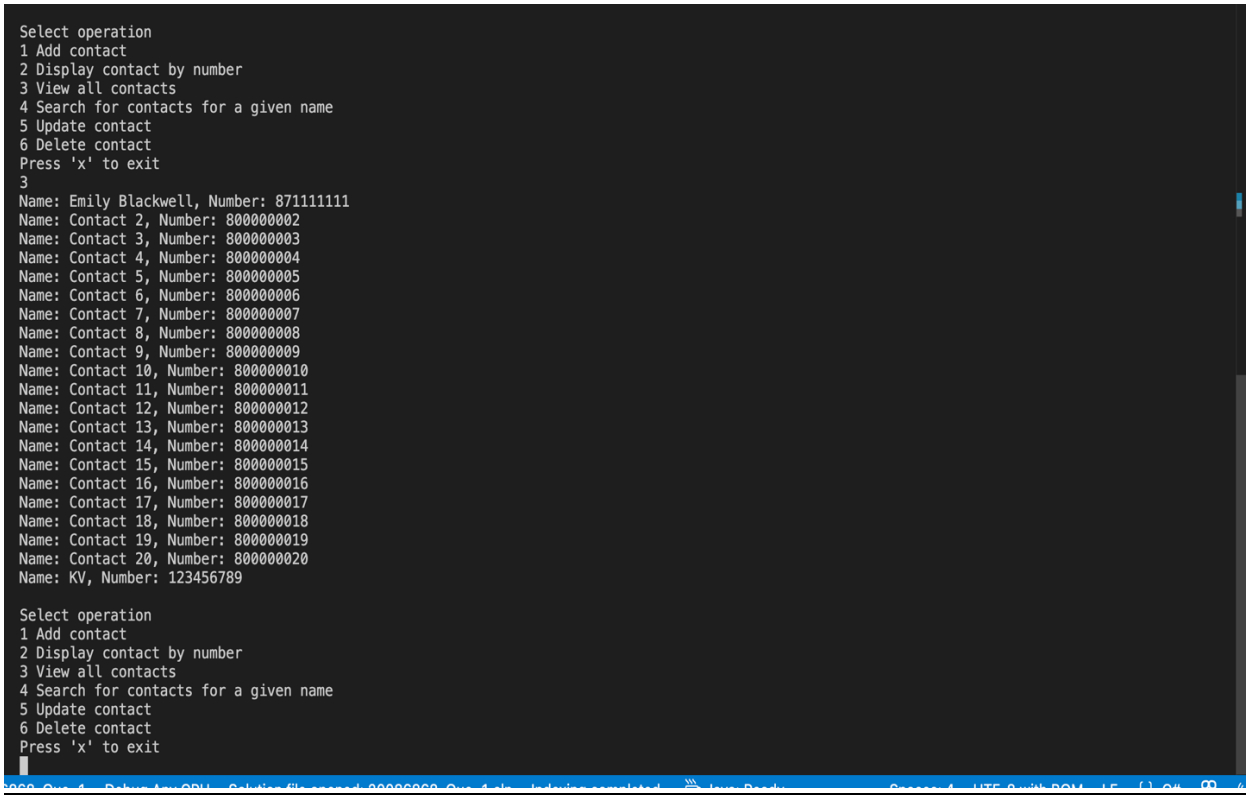

Image 1 – ContactBook Result

```
DELL@MacBook-Air-5 20086868_Que_1 %  /Users/DELL/.vscode/extensions/ms-dotnettools.csharp-2.100.11-darwin-arm64/.debugger/arm64/vsdbg --interpreter
=vscode --connection=/var/folders/2y/5rv8w6js5kzbspxt5cygbfjm0000gn/T/CoreFxPipe_vsdbg-ui-63c5db1fe36d466daf44d27e1c725462
Welcome to 1Console PhoneBook app

Select operation
1 Add contact
2 Display contact by number
3 View all contacts
4 Search for contacts for a given name
5 Update contact
6 Delete contact
Press 'x' to exit
1
Contact name:
KV
Contact number (9-digit, non-zero):
123456789
Contact added.

Select operation
1 Add contact
2 Display contact by number
3 View all contacts
4 Search for contacts for a given name
5 Update contact
6 Delete contact
Press 'x' to exit
2
Contact number to search:
123456789
Name: KV, Number: 123456789
```

Image 2 – ContactBook Result

```
Select operation
1 Add contact
2 Display contact by number
3 View all contacts
4 Search for contacts for a given name
5 Update contact
6 Delete contact
Press 'x' to exit
4
Name search phrase:
KV
Name: KV, Number: 123456789

Select operation
1 Add contact
2 Display contact by number
3 View all contacts
4 Search for contacts for a given name
5 Update contact
6 Delete contact
Press 'x' to exit
5
Enter current contact number to update:
123456789
New name (leave blank to keep same):
KA
New number (leave blank to keep same):
123456788
Contact updated.

Select operation
1 Add contact
2 Display contact by number
3 View all contacts
4 Search for contacts for a given name
5 Update contact
6 Delete contact
Press 'x' to exit
6
Enter contact number to delete:
1234567898
Contact not found.
```

Image 3 – ContactBook Result

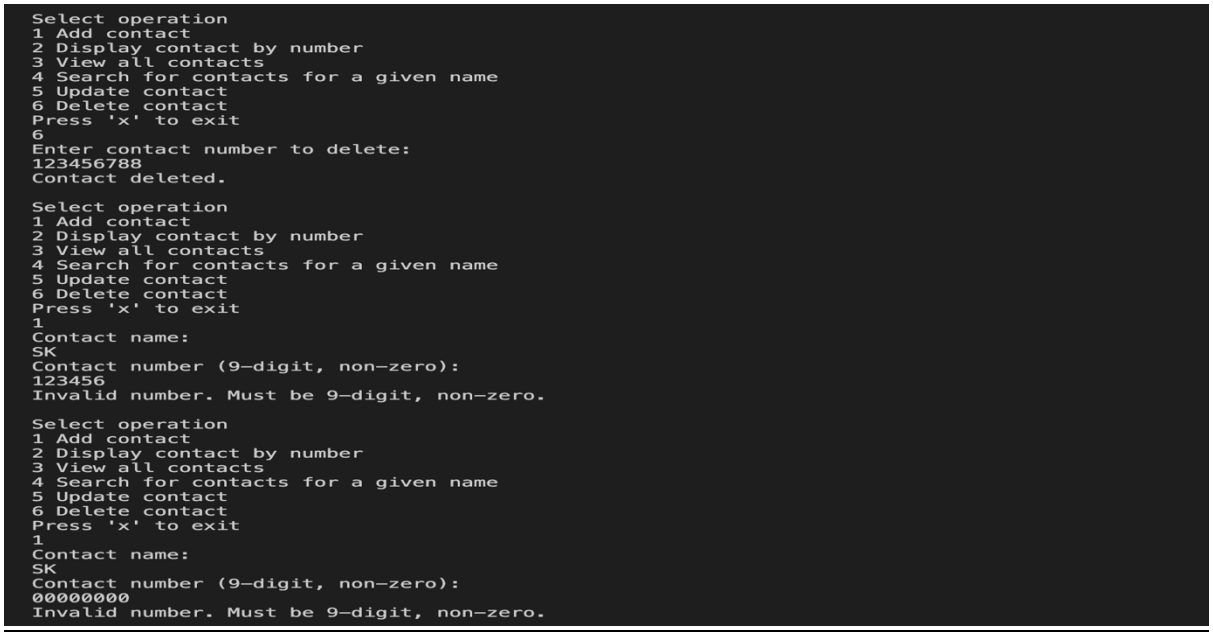

Image 4 – ContactBook Result

Part 1 – C# Programming:

Question 2:

In this task I was asked to write a code that gives information about file extension types. So, if a user types **.mp3,** the code displays the full form of .mp3. It was also mentioned that the code should contain minimum 20 extension types and the exit command exits the operation. I used a Python Dictionary which is a key-value pair. Here, the key is the extension and Value is the full form of the Extension. Dictionary was used as it is simple to integrate and fast.

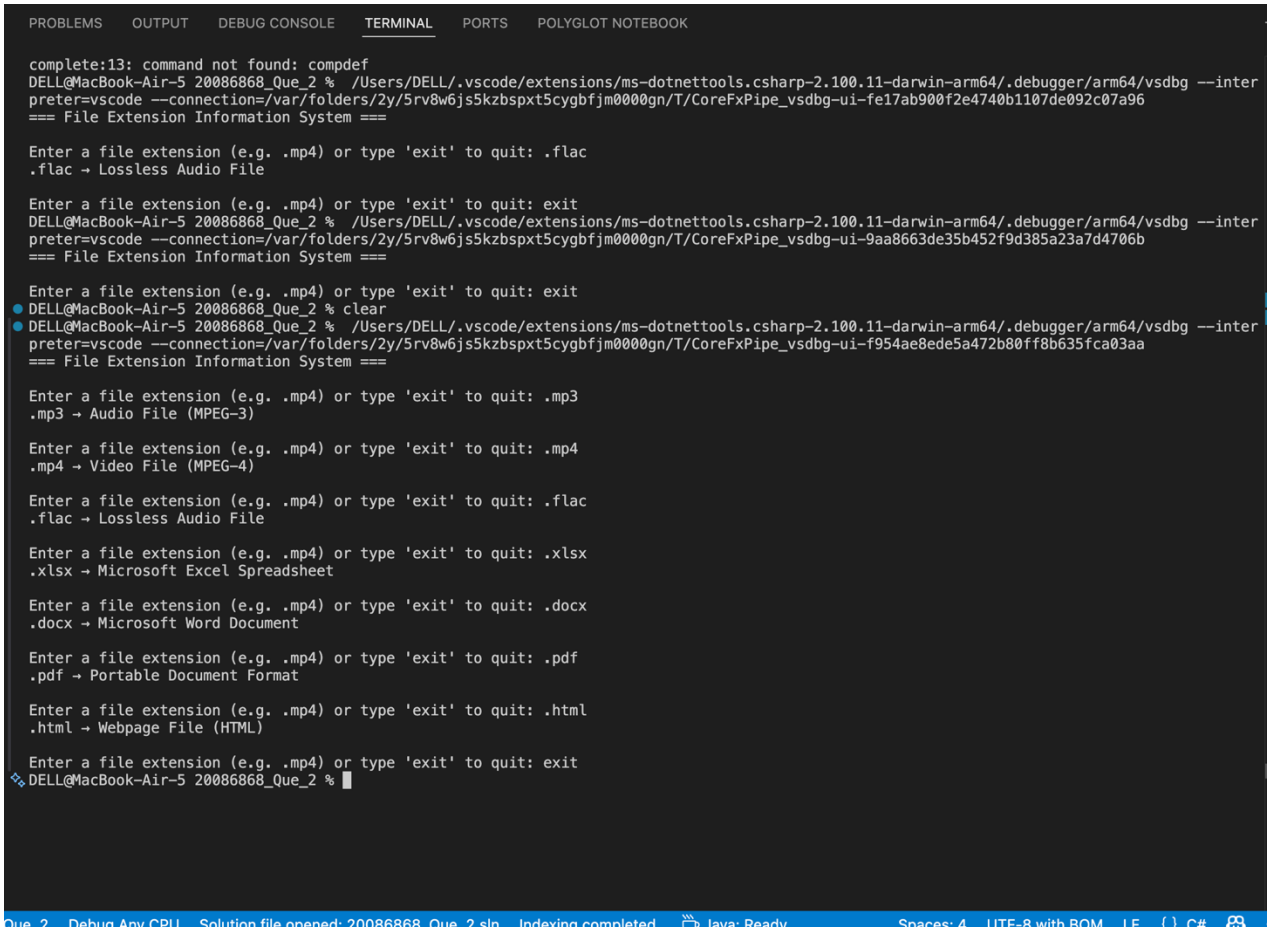

Image 5 – File Extension Types Result

Part 2 – Python Programming:

Question 3:

In this task, I was asked to create a client–server application where the client collects admission details from a student and sends them to the server. The server receives the information, stores it, and sends back a unique registration number. I used TCP because TCP is reliable and ensures that all data sent from the client is delivered safely and in order. The client gets the applicant's name, address, qualifications, course choice, and intended start date, then sends this information to the server. The server stores the data in a file and generates a unique ID using Python UUID library. This unique ID is returned to the client as confirmation.

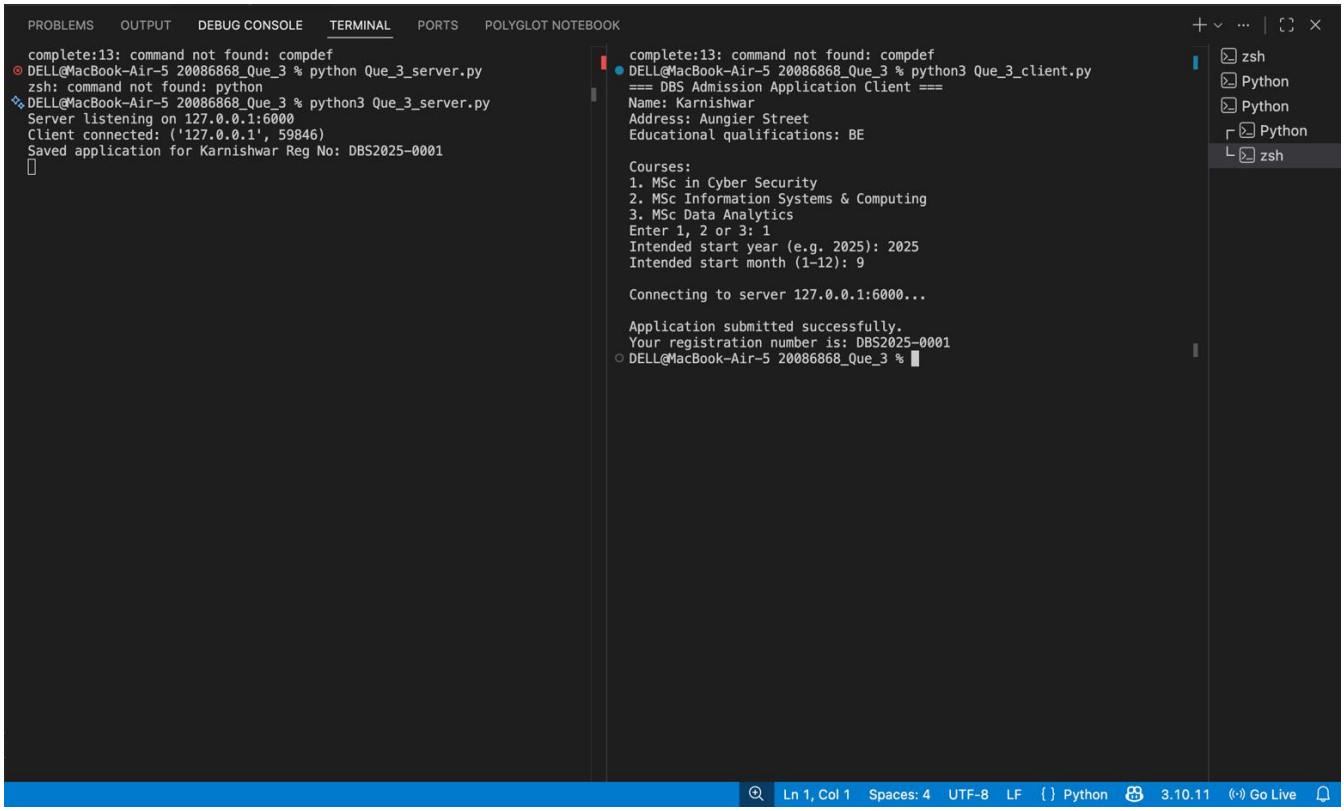

Image 6 – Client-server Result [Left side = server, Right side = client]

Part 2 – Python Programming:

Question 4:

In this task, I was asked to scrape hotel room price information from two websites and present the data in a csv file in order. Since most sites did not allow scraping, I created two HTML webpages that acted as hotel sites. I used Python requests feature to fetch the webpage content and BeautifulSoup to extract the room names and prices using their HTML class names. After collecting the data, I stored it in a CSV file using Python csv module. At the end of the program, the CSV file was read and displayed in the terminal. HTML code for the hotel pages are also included in the project.

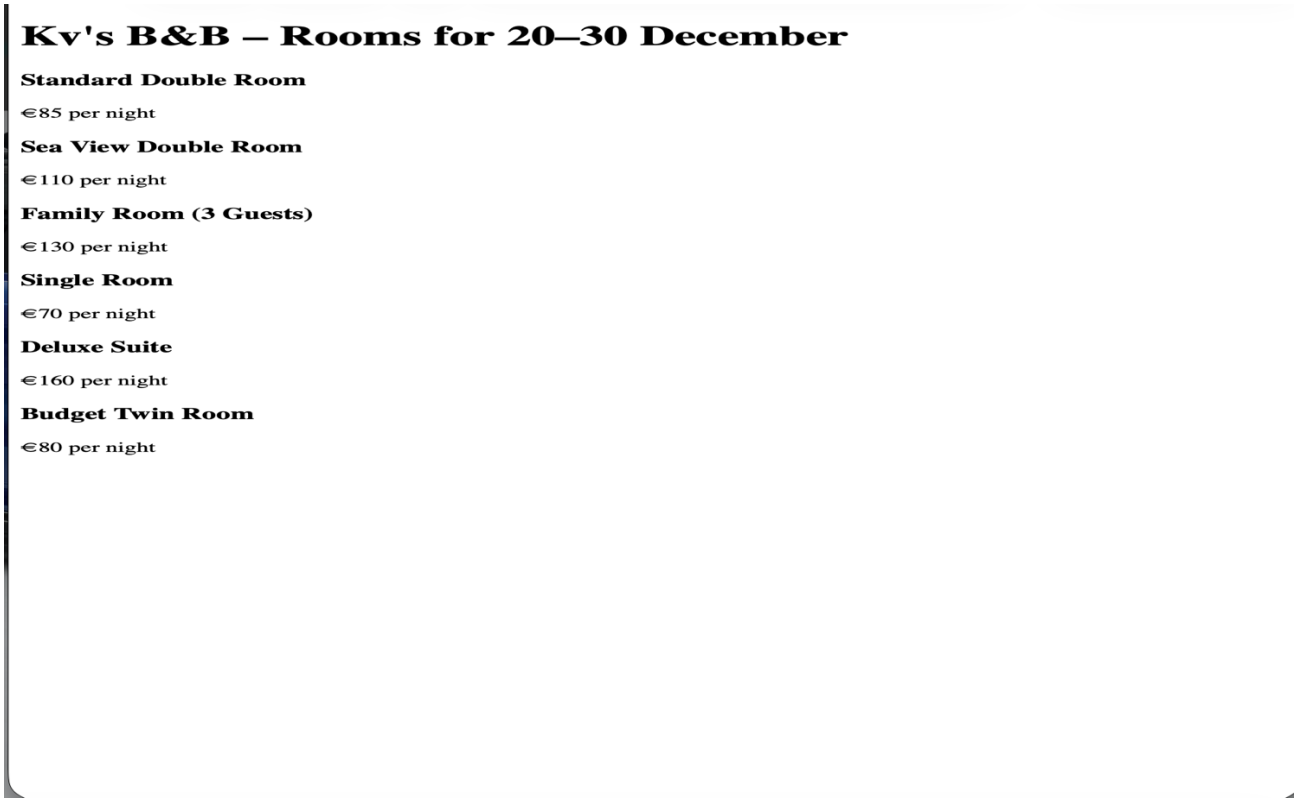

Image 7 – Hotel Website 1

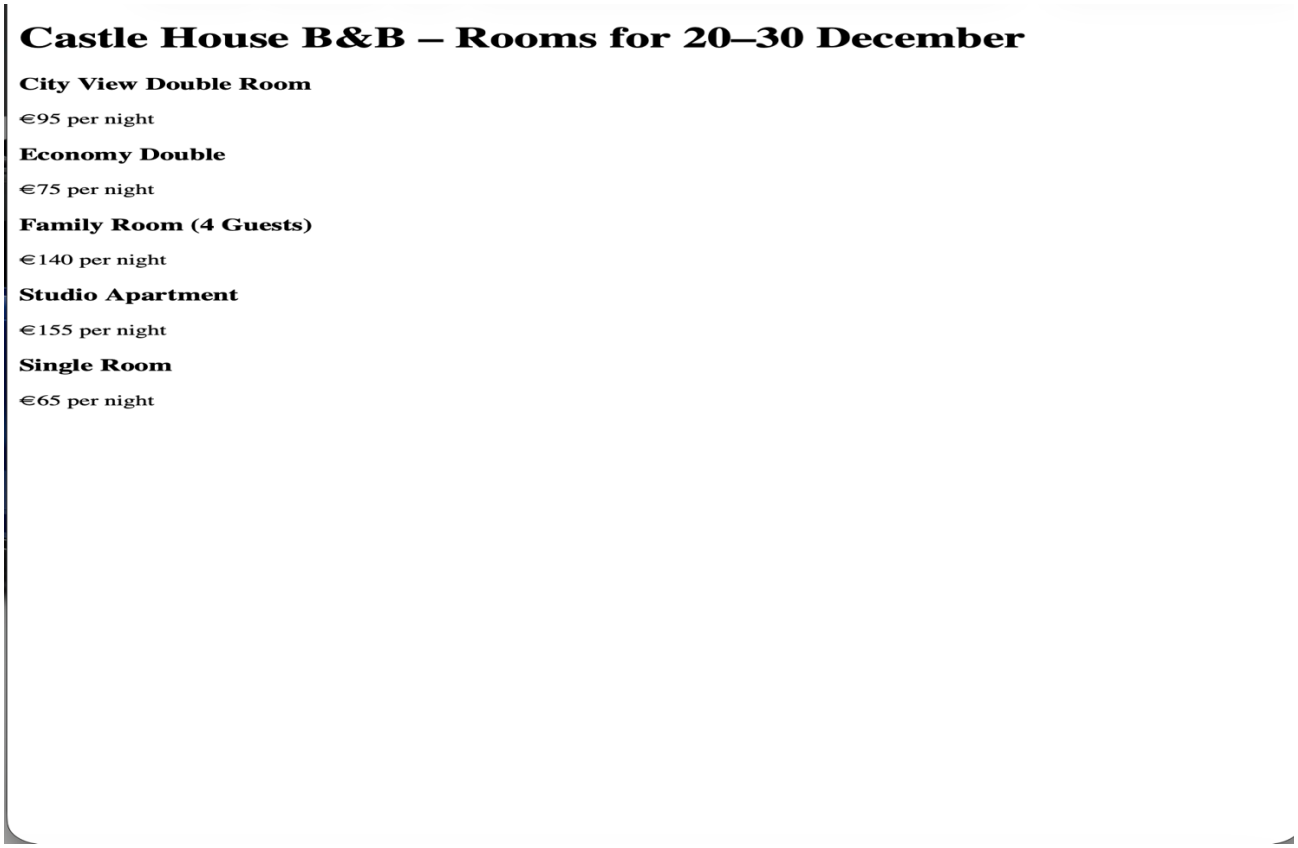

Image 8 – Hotel Website 2

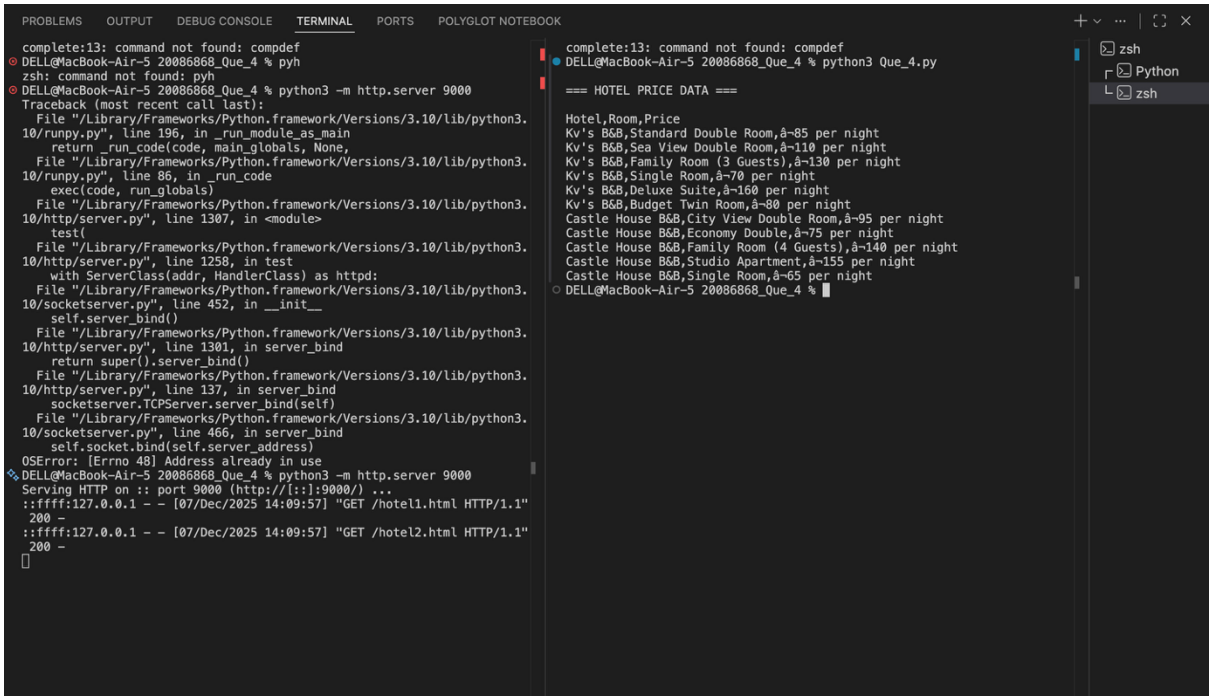

Image 9 – Web Scraping Result (Hotel details displayed in the Terminal)

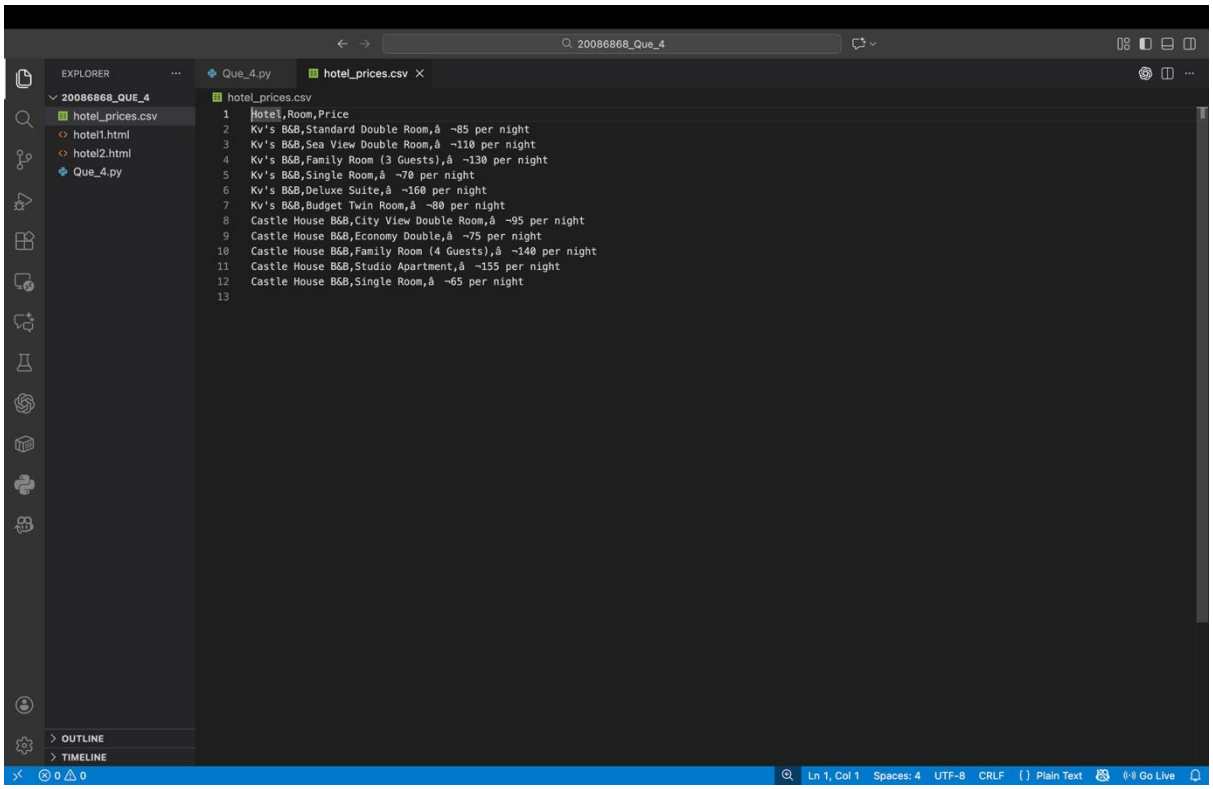

Image 10 – Web Scraping Result (Downloaded CSV File)

Conclusion:

Overall, the tasks in this assignment allowed me to apply different programming concepts. In Question 1, I created a Contact Book system where users can add, view, update, and delete contact information. I used Object-Oriented Programming concepts such as classes and objects and I applied encapsulation by storing contact details inside a Contact class with properties. In Question 2, I used a dictionary to efficiently store and retrieve file extension information. In Question 3, I implemented a TCP-based client–server system that fetched user data and generated a unique registration ID. In Question 4, I created two local webpages and used web scraping to extract hotel room prices and store them in a CSV file.

GitHub Repository Link:

https://github.com/karnishwar1725/CA-1.git