DBS

Dublin
Business
School

Electronic Assignment Cover sheet

Student (s) Number as per your student card:

- **20068451 (Wai Aung)**
-

Course Title: M.Sc. Cyber Security

Lecturer Name: Swati Dongre

Module/Subject Title: Advanced Programming Techniques

Assignment Title: CA_ONE_(30%)

No of Words: 553 Words

Github Repo:

https://github.com/20068451/ADVANCED-PROGRAMMING-TECHNIQUES.git

1. Introduction

For this part of the assignment, I was asked to build a small Python program that collects hotel room prices from two different establishments, saves the information in a CSV file, and then loads it back again at the end. At first, I tried to scrape real hotel websites online, such as booking pages for Dublin hotels. However, this did not work as expected. Later on, my classmate Haruna shared two simple static HTML pages that he had created for practice. These two pages provided the room descriptions and prices needed to complete the assignment.

Using these sample sites, I wrote a Python program that scrapes the room information, stores it in a structured CSV file, and finally reads the CSV back into the terminal. This report explains how the program works, the Python modules I used, and what I learned while completing the task.

2. How the Program Works

My program runs in three main stages:

1. Scraping the hotel webpages
2. Saving the extracted data to a CSV file
3. Reading the CSV file and displaying the results again in the terminal

Scraping the Webpages

I created a list of two hotel websites:

- https://hotel1.tiiny.site
- https://booking-hotels2.tiiny.site/

Both of these were the static pages shared by Haruna.

Saving Data to a CSV File

After collecting the room information, the program writes everything into a file called hotel_prices.csv.

The CSV file includes:

- Hotel_name
- Source_url

- Season_period
- Room_description

Reading the CSV File and Displaying the Results

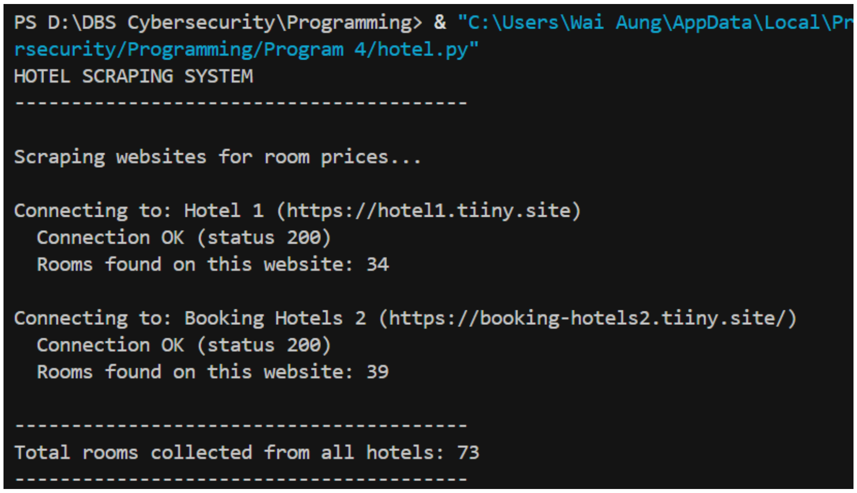
At the end of the program, the CSV file is opened again using csv.DictReader.

The program loops through each row and prints the details clearly in the terminal window.

3. Outputs and Testing

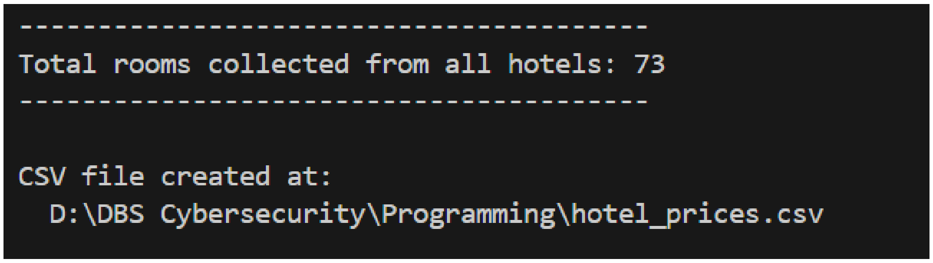
When I ran the program:

Both websites connected successfully.

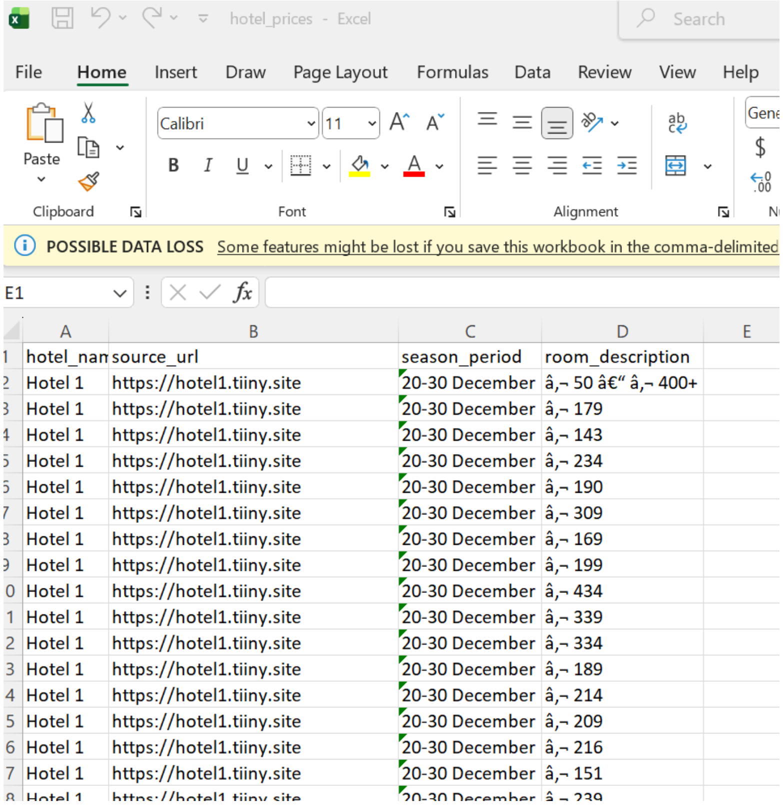


```
PS D:\DBS Cybersecurity\Programming> & "C:\Users\Wai Aung\AppData\Local\Pr
rsecurity/Programming/Program 4/hotel.py"
HOTEL SCRAPING SYSTEM
----------------------------------------

Scraping websites for room prices...

Connecting to: Hotel 1 (https://hotel1.tiiny.site)
  Connection OK (status 200)
  Rooms found on this website: 34

Connecting to: Booking Hotels 2 (https://booking-hotels2.tiiny.site/)
  Connection OK (status 200)
  Rooms found on this website: 39

----------------------------------------
Total rooms collected from all hotels: 73
----------------------------------------
```

All extracted room details were saved into hotel_prices.csv.



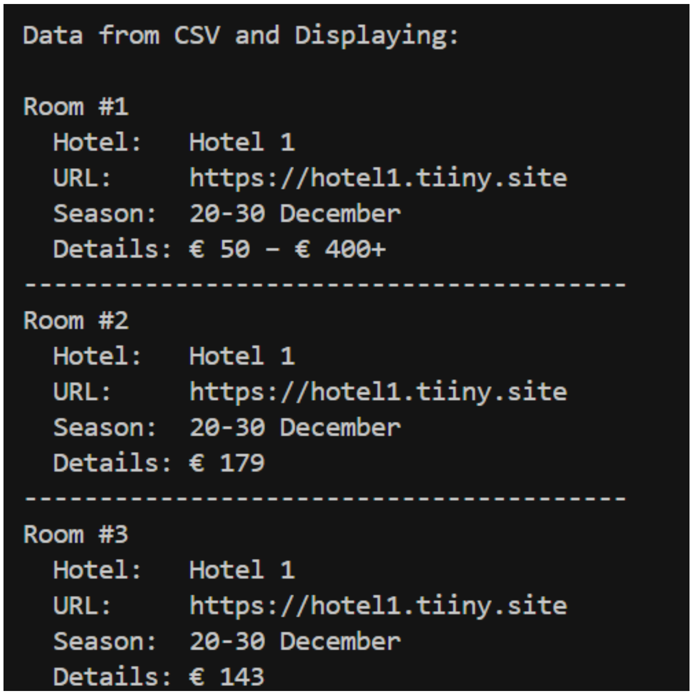
```
----------------------------------------
Total rooms collected from all hotels: 73
----------------------------------------

CSV file created at:
  D:\DBS Cybersecurity\Programming\hotel_prices.csv
```



A	B	C	D	E
hotel_nan	source_url	season_period	room_description	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 50 â€“ â‚¬ 400+	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 179	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 143	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 234	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 190	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 309	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 169	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 199	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 434	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 339	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 334	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 189	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 214	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 209	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 216	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 151	
Hotel 1	https://hotel1.tiiny.site	20-30 December	â‚¬ 239	

The CSV file was created in the same folder where the script was executed.

When I reopened the CSV using the program, every row was displayed correctly in the terminal.



```
Data from CSV and Displaying:

Room #1
  Hotel:   Hotel 1
  URL:     https://hotel1.tiiny.site
  Season:  20-30 December
  Details: € 50 – € 400+
----------------------------------------
Room #2
  Hotel:   Hotel 1
  URL:     https://hotel1.tiiny.site
  Season:  20-30 December
  Details: € 179
----------------------------------------
Room #3
  Hotel:   Hotel 1
  URL:     https://hotel1.tiiny.site
  Season:  20-30 December
  Details: € 143
```

4. What I Learned

While completing this task, I learned several new things:

- How Python connects to websites using the requests module
- How to extract readable text from HTML using BeautifulSoup
- How to clean and filter lines of text to find useful information
- How to create and manage CSV files in Python

This assignment gave me more confidence in handling basic automation tasks and understanding how data can be collected and processed in Python.

6. External Help

I mainly learned from the lecture notes, class discussions, and the Python documentation. I also used Google Gemini to help me understand error messages and improve my scraping logic.

All coding was typed, tested, and run manually by me.

References

BeautifulSoup Documentation, 2024. BeautifulSoup 4.12 – HTML Parsing Guide. Available at: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Mitchell, R., 2018. Web Scraping with Python: Collecting Data from the Modern Web. 2nd ed. Sebastopol: O'Reilly Media.

Python Software Foundation, 2024. Python CSV File Reading and Writing. Available at: https://docs.python.org/3/library/csv.html