Project Proposal – UTI

By Ayesha Rahman
Summer Internship

The project is being done on behalf of UTI Mutual Fund company for a summer internship of 2 months in order to make the work of the analysts at the company easier.
This is PROJECT 1 – Hotels at UTI.

Contents

Introduction:

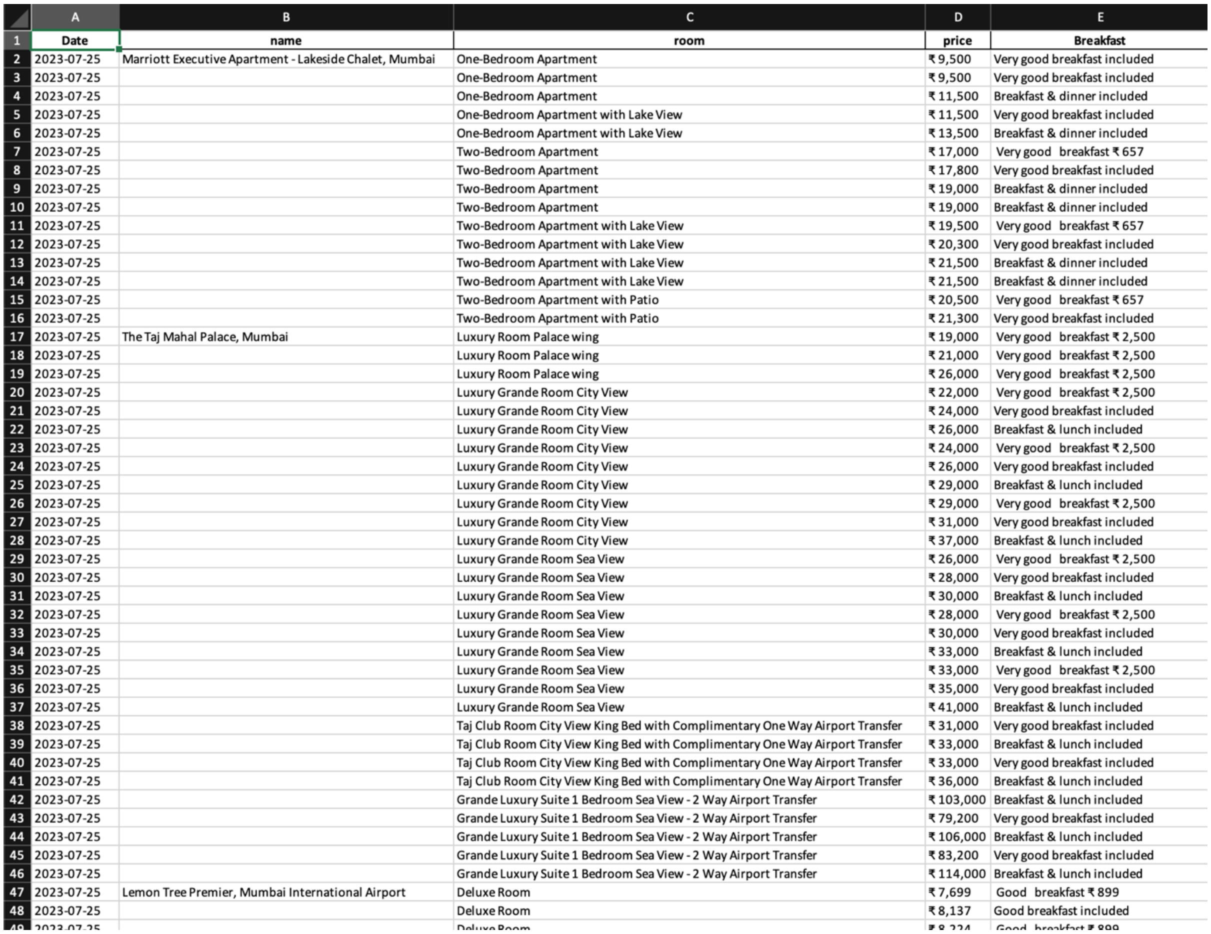
In the past, researchers and scientists faced challenges in accessing relevant data for their ideas or thoughts. The scarcity of available information made it difficult for them to conduct comprehensive research (1). However, with the advent of the World Wide Web, a vast amount of data has become easily accessible. To utilize this information, researchers often resort to web scraping or manual data extraction through traditional copy-and-paste methods.

Nevertheless, certain scenarios, such as extracting financial data or hotel prices, can be tiresome and repetitive when done manually. To alleviate this issue, various methods, including APIs and web scraping, have been employed. This proposal aims to focus on utilizing Python to facilitate legal and ethical web scraping techniques, particularly for individuals who lack access to APIs or cannot afford their usage.

Objectives

The initial phase of the project involves developing a Python-based web scraping code to gather hotel price data, distinguishing between different room types and breakfast availability. The code will be designed to collect data from the present day up to a week in the future. I began my research based on the guidance provided by my mentors for the project, which encompasses two areas: Finance and Computer Science. This proposal focuses specifically on the Computer Science aspect. The desired outcome is to generate a comprehensive table that presents all the pricing details in a structured format.

Date (when code run)	Date for room booking	Hotel	Room Type	Breakfast (yes/no)	Rent
17/7/2023	18/7/2023	Tai Mahal Palace	Type 1	Yes	
	18/7/2023	Tai Mahal Palace	Type 1	No	
	18/7/2023	Tai Mahal Palace	Type 2	Yes	
	18/7/2023	Tai Mahal Palace	Type 2	No	
	18/7/2023	Tai Mahal Palace	Type 3	Yes	
	18/7/2023	Tai Mahal Palace	Type 3	No	
	18/7/2023	Marriot	…	…	
	…	…	…	…	
	19/7/2023	…	…	…	
	19/7/2023	…	…	…	
	…	…	…	…	
	24/7/2023	…	…	…	
18/7/2023	19/7/2023	Taj Mahal Palace	…	…	
…	…	…	…	…	



	A	B	C	D	E
1	Date	name	room	price	Breakfast
2	2023-07-25	Marriott Executive Apartment - Lakeside Chalet, Mumbai	One-Bedroom Apartment	₹ 9,500	Very good breakfast included
3	2023-07-25		One-Bedroom Apartment	₹ 9,500	Very good breakfast included
4	2023-07-25		One-Bedroom Apartment	₹ 11,500	Breakfast & dinner included
5	2023-07-25		One-Bedroom Apartment with Lake View	₹ 11,500	Very good breakfast included
6	2023-07-25		One-Bedroom Apartment with Lake View	₹ 13,500	Breakfast & dinner included
7	2023-07-25		Two-Bedroom Apartment	₹ 17,000	Very good breakfast ₹ 657
8	2023-07-25		Two-Bedroom Apartment	₹ 17,800	Very good breakfast included
9	2023-07-25		Two-Bedroom Apartment	₹ 19,000	Breakfast & dinner included
10	2023-07-25		Two-Bedroom Apartment	₹ 19,000	Breakfast & dinner included
11	2023-07-25		Two-Bedroom Apartment with Lake View	₹ 19,500	Very good breakfast ₹ 657
12	2023-07-25		Two-Bedroom Apartment with Lake View	₹ 20,300	Very good breakfast included
13	2023-07-25		Two-Bedroom Apartment with Lake View	₹ 21,500	Breakfast & dinner included
14	2023-07-25		Two-Bedroom Apartment with Lake View	₹ 21,500	Breakfast & dinner included
15	2023-07-25		Two-Bedroom Apartment with Patio	₹ 20,500	Very good breakfast ₹ 657
16	2023-07-25		Two-Bedroom Apartment with Patio	₹ 21,300	Very good breakfast included
17	2023-07-25	The Taj Mahal Palace, Mumbai	Luxury Room Palace wing	₹ 19,000	Very good breakfast ₹ 2,500
18	2023-07-25		Luxury Room Palace wing	₹ 21,000	Very good breakfast ₹ 2,500
19	2023-07-25		Luxury Room Palace wing	₹ 26,000	Very good breakfast ₹ 2,500
20	2023-07-25		Luxury Grande Room City View	₹ 22,000	Very good breakfast ₹ 2,500
21	2023-07-25		Luxury Grande Room City View	₹ 24,000	Very good breakfast included
22	2023-07-25		Luxury Grande Room City View	₹ 26,000	Breakfast & lunch included
23	2023-07-25		Luxury Grande Room City View	₹ 24,000	Very good breakfast ₹ 2,500
24	2023-07-25		Luxury Grande Room City View	₹ 26,000	Very good breakfast included
25	2023-07-25		Luxury Grande Room City View	₹ 29,000	Breakfast & lunch included
26	2023-07-25		Luxury Grande Room City View	₹ 29,000	Very good breakfast ₹ 2,500
27	2023-07-25		Luxury Grande Room City View	₹ 31,000	Very good breakfast included
28	2023-07-25		Luxury Grande Room City View	₹ 37,000	Breakfast & lunch included
29	2023-07-25		Luxury Grande Room Sea View	₹ 26,000	Very good breakfast ₹ 2,500
30	2023-07-25		Luxury Grande Room Sea View	₹ 28,000	Very good breakfast included
31	2023-07-25		Luxury Grande Room Sea View	₹ 30,000	Breakfast & lunch included
32	2023-07-25		Luxury Grande Room Sea View	₹ 28,000	Very good breakfast ₹ 2,500
33	2023-07-25		Luxury Grande Room Sea View	₹ 30,000	Very good breakfast included
34	2023-07-25		Luxury Grande Room Sea View	₹ 33,000	Breakfast & lunch included
35	2023-07-25		Luxury Grande Room Sea View	₹ 33,000	Very good breakfast ₹ 2,500
36	2023-07-25		Luxury Grande Room Sea View	₹ 35,000	Very good breakfast included
37	2023-07-25		Luxury Grande Room Sea View	₹ 41,000	Breakfast & lunch included
38	2023-07-25		Taj Club Room City View King Bed with Complimentary One Way Airport Transfer	₹ 31,000	Very good breakfast included
39	2023-07-25		Taj Club Room City View King Bed with Complimentary One Way Airport Transfer	₹ 33,000	Breakfast & lunch included
40	2023-07-25		Taj Club Room City View King Bed with Complimentary One Way Airport Transfer	₹ 33,000	Very good breakfast included
41	2023-07-25		Taj Club Room City View King Bed with Complimentary One Way Airport Transfer	₹ 36,000	Breakfast & lunch included
42	2023-07-25		Grande Luxury Suite 1 Bedroom Sea View - 2 Way Airport Transfer	₹ 103,000	Breakfast & lunch included
43	2023-07-25		Grande Luxury Suite 1 Bedroom Sea View - 2 Way Airport Transfer	₹ 79,200	Very good breakfast included
44	2023-07-25		Grande Luxury Suite 1 Bedroom Sea View - 2 Way Airport Transfer	₹ 106,000	Breakfast & lunch included
45	2023-07-25		Grande Luxury Suite 1 Bedroom Sea View - 2 Way Airport Transfer	₹ 83,200	Very good breakfast included
46	2023-07-25		Grande Luxury Suite 1 Bedroom Sea View - 2 Way Airport Transfer	₹ 114,000	Breakfast & lunch included
47	2023-07-25	Lemon Tree Premier, Mumbai International Airport	Deluxe Room	₹ 7,699	Good breakfast ₹ 899
48	2023-07-25		Deluxe Room	₹ 8,137	Good breakfast included
49	2023-07-25		Deluxe Room	₹ 8,224	Good breakfast ₹ 899

Project Scope

The project's main objective is to develop a comprehensive Python code that enables efficient extraction of hotel prices based on room type and breakfast availability from multiple websites. To achieve this, the code will incorporate some key functionalities.

1. Firstly, it will be designed to read and process website links, which are to be extracted from an Excel sheet. Additionally, the code will also extract the corresponding HTML classes for each website's source pages from the same Excel sheet. This approach ensures a flexible and user-friendly experience, as users can easily update and manage the input data without altering the code directly.
2. Once the necessary inputs are obtained, the Python code will utilize web crawling techniques to navigate the specified webpages and gather the relevant hotel price information. By implementing intelligent data scraping methods, the code will accurately extract data related to room types and breakfast availability from each website, ensuring a comprehensive collection of data points.
3. The output of the Python code will be presented in the form of an Excel sheet. This output format is chosen for its simplicity and compatibility, allowing analysts to easily manipulate and analyse the extracted data using familiar spreadsheet tools.

4. The resulting Excel sheet will be structured in a tabular form, closely resembling the table defined under the objectives heading in the project report. This well-organized presentation will include hotel names, room types, corresponding prices, and breakfast availability details, providing analysts with a clear overview of the hotel options available across the websites.

The development of this Python project aims to streamline the process of gathering hotel price data, saving users considerable time and effort in the data collection and compilation phase. By automating the extraction and organization of hotel prices according to room types and breakfast availability, the code facilitates informed decision-making. The code's robustness, flexibility, and user-friendly output format make it a valuable tool for travellers, researchers, and anyone involved in the hospitality industry seeking to access and analyse hotel price data efficiently.

Imports and Functions:

Imports Used:

1. from bs4 import BeautifulSoup
2. import pandas as pd
3. from datetime import datetime, timedelta
4. import tkinter as tk
5. from tkinter import filedialog, messagebox, simpledialog
6. import logging
7. import os
8. import time

Functions Created:

1. Functions for Handling Lock File:
    a. `create_lock_file()`: Creates a lock file named 'scraping.lock' to indicate that the scraping process is running.
    b. `remove_lock_file()`: Removes the lock file 'scraping.lock' if it exists, indicating that the scraping process is completed or encountered an error.
2. Custom Function for Displaying Popup:
    a. `show_popup(title, message)`: Displays a popup window with a custom style, containing a title, message, and "OK" button.
3. Web Scraping Function:
    a. `scrape_hotel_data(target_url, class_name_dict, tag_dict, checkin_date, data_dict, num_days): Performs web scraping on the target_url for a specified number of days (num_days) starting from the checkin_date. It retrieves hotel data such as name, address, rating, room details, price, and breakfast options and stores the scraped data in data_dict.
4. Function to Save Data to Excel:
    a. `save_to_excel(data_dict, filename)`: This function saves the scraped data stored in data_dict to an Excel file with the specified filename.
5. Functions for User Input and Output:
    a. `get_city()`: Asks the user to input the city name from a list of cities.

b. `choose_url_city_file(city)`: Asks the user to choose the 'url{city}.xlsx' file containing target URLs for scraping.
c. `choose_directory(title="Choose a directory")`: Asks the user to choose a directory using a dialog box.
d. `choose_file(title="Choose a file")`: Asks the user to choose any file using a dialog box.

6. `main()`: The main function drives the entire process. It starts by asking the user for the city name, constructing the target URL, and reading target URLs and corresponding classes and tags from 'url{city}.xlsx'. It then asks the user to choose the output directory and the number of days to scrape data. The web scraping is performed for each target URL and saves the data to an Excel file named '{date}-{city}.xlsx' in the chosen output directory.

Methodology

The methodology employed for developing the Python code aimed at extracting hotel room prices based on room type and breakfast availability involved several key steps. Initially, a sample code from a reliable web scraping source was utilized as a reference to understand the fundamentals of data extraction from Booking.com. Subsequently, the code was carefully edited and customized to meet the specific requirements of our project. Extensive modifications were made to ensure seamless integration with the target website's structure and data layout.

Iteration 1

To begin the scraping process, the BeautifulSoup library was leveraged to parse the HTML content of the booking.com website and retrieve relevant information about hotel rooms, room types, and breakfast options. The code was designed to handle dynamic content loading and pagination, ensuring comprehensive data collection across multiple pages. After successfully extracting the necessary data, the Python code was optimized for performance and efficiency. This included implementing caching mechanisms and reducing unnecessary HTTP requests to prevent overloading the server.

As the scraping process matured, the extracted data was efficiently organized and cleaned before being saved in a CSV file format, allowing for easy data manipulation and analysis. The CSV output provided a solid foundation for validating the accuracy of the extracted information and served as a crucial intermediate step in refining the code further.

Based on feedback and specific project requirements, the next iteration of the Python code will be to adapt to produce the output in the form of an Excel file rather than a CSV. This change enabled seamless integration with existing data management workflows and facilitated more comprehensive data visualization. Throughout the development process, rigorous testing and validation were conducted to ensure the code's reliability and accuracy. Any discrepancies were addressed promptly, and refinements were made to enhance the overall performance.

Iteration 2

In Iteration 2 of the project, significant enhancements have been implemented to improve the code's functionality and usability. One major improvement is the transformation of the output format from CSV to Excel, providing users with a more versatile and familiar way to access and analyse the extracted data. The transition to Excel allows for better data organization, formatting options, and compatibility with other spreadsheet tools, making it easier for analysts to work with the extracted hotel price information.

Another substantial enhancement in Iteration 2 is the dynamic handling of website URLs and HTML classes. Unlike in the previous iteration, where the URLs were hardcoded directly into the code, now, in Iteration 2, the URLs are stored in an Excel sheet. This improvement empowers analysts to manually input the website URLs directly into the Excel sheet, thereby enabling them to easily update and manage the list of websites to be scraped without modifying the code itself.

Moreover, the Excel sheet now also contains the corresponding HTML div classes for each website's source pages. These div classes are essential for targeting and extracting the specific hotel price data accurately. The code in Iteration 2 reads these HTML div classes from the Excel sheet, ensuring a more adaptable and user-friendly approach. With this dynamic configuration, the code can easily adapt to changes in the website's HTML structure or accommodate new websites without requiring code modifications.

The iterative approach in the project development ensures that the code evolves with the needs and feedback of the analysts, leading to an optimized and efficient solution. The combination of Excel output format and dynamic URL and HTML class handling in Iteration 2 streamlines the hotel price data extraction process, enhancing the code's overall usability and making it a valuable tool for analysts in the hospitality industry and related research domains.

Iteration 3

With the release of Iteration 3, I have introduced a plethora of new features and improvements to our hotel price data extraction tool, based on invaluable user feedback and specific requirements. This iteration has been focused on delivering a truly customizable and user-centric experience, catering to the diverse needs of analysts in the hospitality industry and related research domains.

One of the most significant advancements in this iteration is the implementation of a user-configurable popup that allows analysts to specify the number of days' worth of data they wish to extract from booking.com. Previously, the code was set to extract data for a fixed period of 7 days, which might not always align with the analysts' research objectives. The introduction of the popup brings an unprecedented level of flexibility and customization to the data collection process. Analysts can now effortlessly input their desired duration, ranging from a few days to several weeks, allowing them to gather precisely the data they need for their research and analysis.

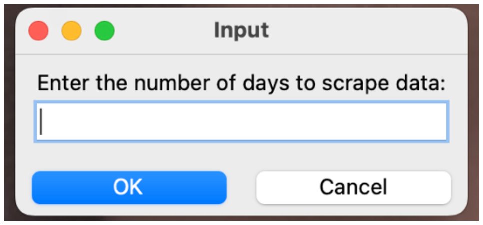
By empowering analysts to customize the data extraction duration, I have significantly improved the relevance and accuracy of the extracted data. Researchers can now delve into specific timeframes, explore trends, or analyse seasonal variations in hotel prices, leading to deeper and more insightful conclusions. This newfound ability to tailor data collection is a game-changer, enabling analysts to make well-informed decisions and strategic recommendations based on data that closely align with their unique research objectives.

Beyond the user-configurable popup, Iteration 3 also introduces a range of additional precautions to ensure a seamless and secure data extraction process. A prominent addition is the creation of a lock file that appears when the code initiates execution. This lock file serves as a visual indicator, promptly notifying analysts that the code is actively running and extracting data. It prevents inadvertent interruptions during the scraping process, ensuring that analysts are aware of the ongoing activity and can avoid accidental termination or interference.

Moreover, the lock file acts as a safeguard against potential concurrent runs of the code. In instances where multiple analysts inadvertently launch the code simultaneously, the lock file serves as an alert, preventing data corruption or conflicts. This protective mechanism ensures that the data extraction process remains reliable, efficient, and devoid of any unexpected issues.

The seamless integration of the user-configurable popup and the lock file mechanism reflects our commitment to providing a refined and efficient user experience. By offering analysts the freedom to extract data within their preferred timeframe and by instilling a sense of confidence and security during code execution, I aim to optimize the data extraction workflow.

The iterative development approach continues to be a driving force behind the project's progress, allowing it to consistently evolve and refine the code based on user needs and changing industry dynamics. The introduction of the user-configurable popup and the lock file mechanism in Iteration 3 elevates the code's usability, reliability, and overall user experience. It empowers researchers to seamlessly access, analyze, and derive meaningful insights from hotel price data, reinforcing our commitment to developing a cutting-edge tool that resonates with the needs of our valued users in the ever-evolving hospitality industry.



Enter the number of days to scrape data:

Iteration 4

In the latest Iteration 4 of our hotel price data extraction tool, I have made significant strides towards elevating the user experience and optimizing the efficiency of data retrieval. Our primary focus has been on enhancing user-friendliness and accessibility, ensuring that analysts can harness the full potential of the tool with unparalleled ease.

One of the standout features introduced in this iteration is the implementation of a user-friendly directory selection interface. Upon initiating the application, the code now presents a prompt that enables analysts to effortlessly choose the directory where the input Excel file resides. This file contains crucial website URLs and their corresponding HTML div classes, essential for the data extraction process. With this enhancement, analysts can easily manage and update the list of websites to be scraped, tailoring data collection to their specific research requirements.

Moreover, we recognize the significance of providing analysts with control over the output location of the extracted data. Therefore, Iteration 4 includes an additional prompt that allows users to specify the directory where they wish to save the output Excel file. This new feature streamlines the data export process, eliminating the need for manual file transfers. By seamlessly saving the extracted data to their preferred location, analysts can rapidly access the results for further analysis or integration into existing workflows.

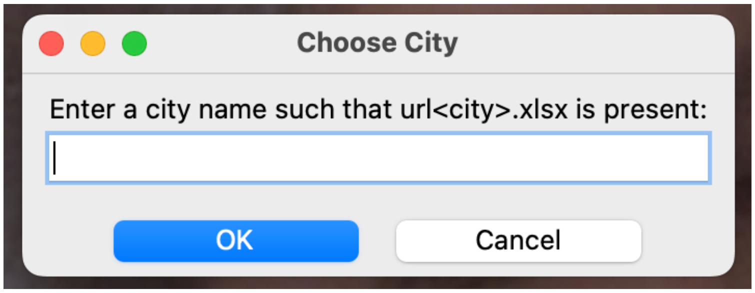
To further enhance the tool's accessibility and user-friendliness, I have transformed the code into an executable file (exec file). This strategic move means that analysts no longer need to execute the code using a Python interpreter; they can directly run the exec file, making the entire process more efficient and straightforward. This significant improvement empowers researchers to leverage the data extraction tool across various computing systems, without the need for Python installation. I have ensured that the tool is now truly platform-independent, broadening its reach to cater to a wider range of users.

As we progress with our iterative development approach, I remain dedicated to refining and innovating the tool to meet the evolving needs of analysts in the dynamic hospitality industry and related research domains. By allowing users to easily select input and output directories and converting the code into an exec file, I have removed barriers and streamlined the data extraction process, providing an unparalleled level of convenience to our valued users.

Our commitment to delivering an exceptional user experience and empowering analysts with unparalleled insights remains unwavering. Through Iteration 4, we have taken a significant leap forward in making our hotel price data extraction tool a reliable and indispensable asset for professionals in the hospitality industry and beyond. As we look to the future, we will continue to embrace the iterative development approach, incorporating valuable feedback and cutting-edge advancements to ensure our tool remains at the forefront of data extraction technology.

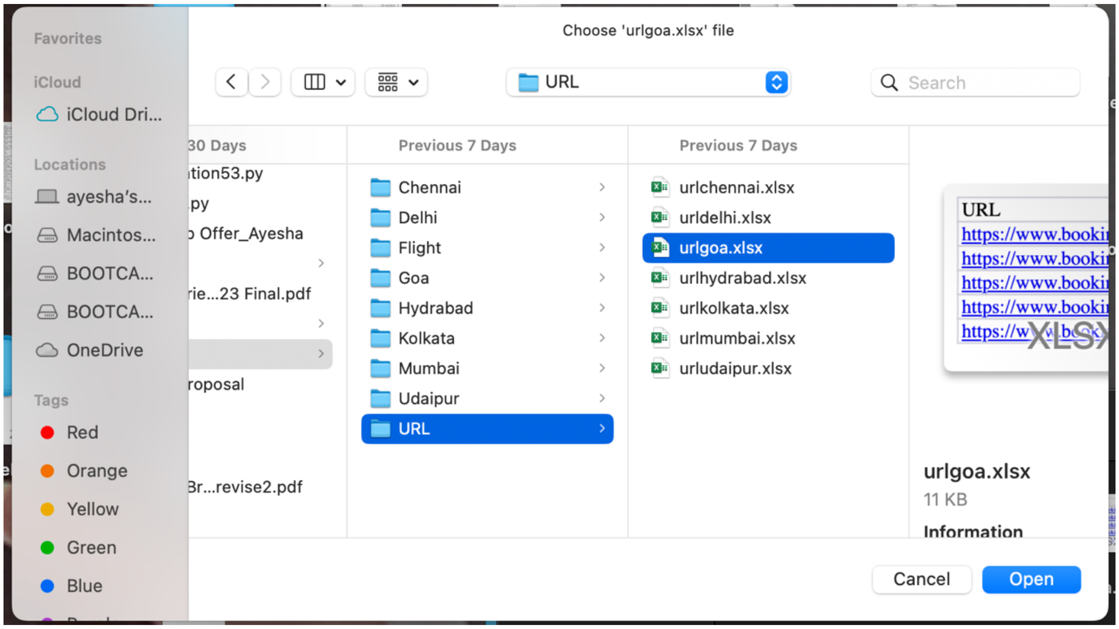
Iteration 5:

In Iteration 5 of our advanced hotel price data extraction tool, I have introduced a range of new features aimed at enhancing user flexibility and further streamlining the data extraction process. Acknowledging the diverse research needs of analysts, the code now offers the capability for users to input the specific city they want to scrape data for. This functionality enables analysts to target their research to a particular city of interest, tailoring the data extraction to match their research objectives precisely.



Enter a city name such that url<city>.xlsx is present:

Upon providing the desired city, the code prompts the user to select the appropriate url<city>.xlsx file. This user-friendly interface simplifies the data selection process, ensuring that analysts can easily choose the relevant input data specific to their city of interest.



As part of the iterative development approach, I have also made improvements to the naming convention of the output Excel file to enhance data traceability. Now, the output file is named in the format <starting date>-<city>.xlsx, where the starting date corresponds to the date when the data extraction process was initiated. This standardized naming convention streamlines data management, making it easier for analysts to track and organize extracted data for different time periods and cities.

To improve transparency and facilitate error tracking, we have also introduced the creation of a log file. This log file captures essential information about the data extraction process, including any potential issues encountered during the scraping process. The log file serves as a valuable resource for analysts, offering insights into the data extraction process and facilitating troubleshooting in case of any unforeseen challenges.

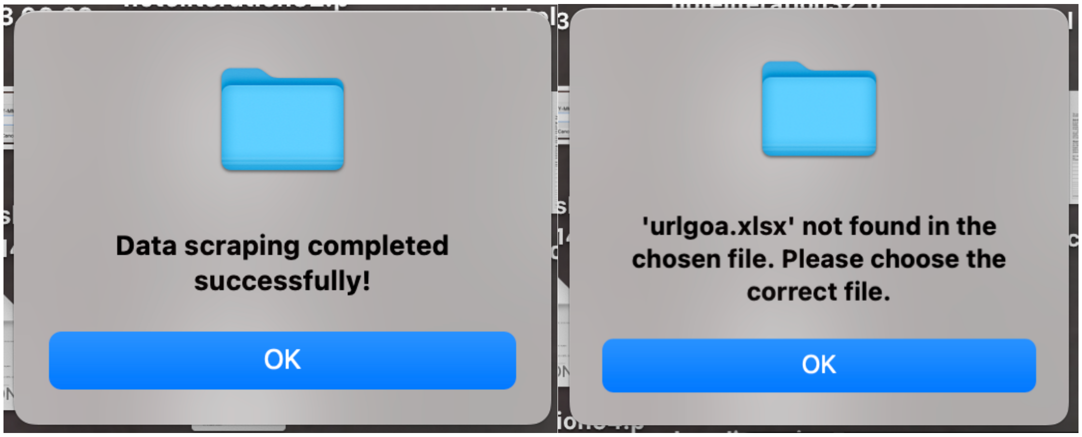
With the introduction of these new features in Iteration 5, our hotel price data extraction tool has evolved into an even more robust and user-centric solution. By enabling users to select a specific city and choose the corresponding input file, we empower analysts to conduct targeted research with utmost precision. The standardized naming convention for the output file and the implementation of a log file ensure data integrity and facilitate smoother data management and analysis.

As we continue our iterative development journey, we remain committed to staying at the forefront of data extraction technology. Iteration 5 stands as a testament to our dedication to meeting the evolving needs of analysts in the dynamic hospitality industry and related research domains. Through user feedback and cutting-edge advancements, we strive to elevate our tool's performance and usability, empowering analysts to unlock unparalleled insights from hotel price data with ease and confidence.

Iteration 6:

In the final and most comprehensive Iteration 6 of our hotel price data extraction tool, we have prioritized user-centricity and improved the overall user experience to an unprecedented level. We have meticulously addressed user prompts and error handling, ensuring a smoother and more intuitive interaction with the application. Our objective was to create a seamless and error-free experience for analysts, empowering them to extract data with utmost confidence and ease.

One of the significant enhancements introduced in this iteration is the improved user prompts and error-handling mechanisms. The code now provides clearer and more informative prompts, guiding analysts through each step of the data extraction process. In cases where users encounter errors or input invalid data, the error handling system offers comprehensive and helpful feedback, allowing users to quickly rectify their input and proceed with confidence.



Data scraping completed successfully!

'urlgoa.xlsx' not found in the chosen file. Please choose the correct file.

As we embark on the final phase of our iterative development approach, we have transformed our hotel price data extraction tool into a truly user-centric and robust solution. The improved user prompts, error handling, and expanded city-input capabilities offer analysts an unparalleled level of flexibility and confidence in the data extraction process. The incorporation of a terms and conditions page exemplifies our commitment to ethical and responsible data usage, placing user awareness at the core of our development principles.

Iteration 6 marks the culmination of our efforts to create a powerful and reliable tool that caters to the diverse needs of analysts in the hospitality industry and beyond. By providing a seamless and intuitive user experience, we aim to empower researchers to harness the full potential of hotel price data, driving innovation and insights that shape the future of the hospitality industry. As we conclude this iteration, I extend our sincere gratitude to our users for their invaluable feedback and support, which has been instrumental in shaping the evolution of our data extraction tool. With a firm commitment to continuous improvement and innovation, we look forward to a future where data-driven insights power transformative advancements in the global hospitality landscape.

Running the application

After the successful implementation of Iteration 6, we envision a user-friendly and self-contained data analysis application that allows analysts to run it without any prior knowledge of the underlying Python code. With the completion of Iteration 6, we have transformed the Python code into a fully functional application that includes a proper GUI, making it easier for users to interact with the tool intuitively. The installation process will be made seamless through the development of a user-friendly installer, eliminating the need for users to manually handle any code or dependencies. Once installed, users can simply launch the application and be greeted by a visually engaging interface. The GUI will prompt analysts to input the specific date range from which they want to retrieve data and inquire about the desired data duration for analysis. By providing these inputs, the application will dynamically generate insightful data visualizations, presenting trends, patterns, and relationships in an easy-to-understand tabular format. The application's user-centric design empowers analysts to explore their data effortlessly, perform in-depth analysis, and make data-driven decisions without worrying about the intricacies of the underlying code. This approach ensures that users, regardless of their technical background, can efficiently utilize the application to its full potential, democratizing data analysis and enabling a wider audience to leverage the power of data-driven insights for their diverse projects and applications.

Legality and Ethics of Web Scraping:

Web scraping involves the automated extraction and organization of data from the web, using technical tools, with the purpose of analysing the data further (2). Although the use of web scraping software and tools has become widespread, the legal and ethical considerations associated with it are often overlooked (1). The abundance of web scraping tools has created the misconception that web scraping is entirely legal, when in reality, its legality remains uncertain and falls into a "grey area" within the legal field (3).

Since there are no specific laws directly addressing web scraping, it is governed by a set of related legal theories and laws, including copyright infringement, the Computer Fraud and Abuse Act (CFAA), breach of contract, and trespass to chattels (3). To effectively prevent web scraping, website owners can explicitly prohibit it in their publicly accessible "terms of use" policy. Violating these terms and conditions may constitute a breach of contract, enabling the website owner to pursue legal action against the user (2). Furthermore, republishing scraped data or information that is owned or explicitly copyrighted by a website can result in a copyright infringement case (2).

Website owners possess the capacity and authority to establish and enforce a wide range of measures, including but not limited to imposing comprehensive restrictions or erecting robust firewalls, with the specific aim of impeding the activities of web scrapers and web crawlers in their quest to extract data from their websites. These measures effectively serve as formidable barriers, meticulously designed to thwart the efforts of automated tools, even when the data being targeted is seemingly accessible to the public and falls within the legal boundaries (2).

By proactively implementing these restrictions and fortified firewalls, website owners assert their rights and exercise control over their digital domains, actively safeguarding the valuable assets contained within. These protective measures encompass a multifaceted approach, combining cutting-edge technologies, advanced algorithms, and meticulously crafted protocols that scrutinize and evaluate incoming requests in real-time. Through the strategic implementation of IP blocking mechanisms, stringent rate limiting strategies, multifactor authentication protocols, and other sophisticated security measures, website owners effectively raise the barriers for unauthorized data extraction, diminishing the chances of successful scraping endeavours.

The authority bestowed upon website owners enables them to institute a myriad of sophisticated measures that effectively deter and hinder web scrapers and web crawlers from extracting data from their websites, even if that data is ostensibly available to the public and meets legal criteria. Through the implementation of comprehensive restrictions, fortified firewalls, and cutting-edge security measures, website owners actively assert their control over their digital realms, safeguarding their valuable assets, complying with legal obligations, protecting user privacy, and fortifying their intellectual property rights.

Limitations

While developing the Python code for extracting hotel room prices based on room type and breakfast availability, several limitations have been encountered:

1. **Website Structure Changes:** Booking.com, like any other website, may undergo frequent updates and changes in its structure. These changes could affect the HTML layout, class names, or IDs used to identify the relevant data, causing the scraper to break or return incorrect results.
2. **Anti-Scraping Mechanisms:** Booking.com may implement anti-scraping mechanisms to prevent automated data extraction. These can include CAPTCHAs, rate limiting, IP blocking, or user agent detection. Overcoming these measures ethically requires implementing strategies to avoid detection while scraping.
3. **Data Availability:** The availability and presentation of data on booking.com may vary depending on the specific hotel, location, or date range. Some hotels may not provide room-specific pricing or breakfast options, leading to missing data in the scraped results.
4. **Legal Concerns:** Scraping data from websites can raise legal issues, especially if the website's terms of service explicitly prohibit scraping. It is essential to review and comply with the website's terms and conditions to avoid any potential legal repercussions.
5. **Data Accuracy**: Web scraping relies on the assumption that the website's content is up-to-date and accurate. However, there might be instances where the data is outdated or erroneous, leading to inaccuracies in the scraped results.
6. **IP Blocking and Captcha Challenges:** Frequent scraping activities from the same IP address may lead to IP blocking, preventing further data extraction. Captchas might also be encountered, requiring manual intervention to bypass them, which can slow down the scraping process.
7. **Ethical Considerations:** While scraping public data is generally acceptable, scraping too frequently or excessively could strain the website's resources or violate ethical guidelines. Ensuring responsible scraping practices is essential to maintain a positive relationship with the website owners and users.
8. **Data not available?** If data for any hotel which was run is not given, then that means there are no rooms available for that hotel on that website.

Addressing these limitations requires careful planning, continuous monitoring of the scraping process, and adapting the code as needed to handle changes in the website's structure and policies. Regular updates and maintenance are essential to ensure the code remains functional and compliant with both legal and ethical considerations.

Conclusion

Engaging in this project has been instrumental in expanding my knowledge and practical experience in both the fields of Computer Science and Finance. Through this endeavour, I have had the opportunity to establish a meaningful connection between these two domains, gaining valuable insights into how they intersect and complement each other. Prior to embarking on this project, I had limited familiarity with the concept of web scraping. However, through thorough research and hands-on implementation, I have acquired a comprehensive understanding of this technique and its applications. This exposure to the world of web scraping has broadened my skill set and equipped me with a valuable tool for extracting and analysing data from various sources, fostering my growth as a professional in the field.

Throughout the development of the Python code for hotel room price extraction, strict adherence to ethical and legal practices was paramount. To ensure that the legal terms of all websites were not compromised, extensive precautions were taken during the scraping process. The code was designed to respect the website's robots.txt file, ensuring that the scraping activities did not violate any crawling restrictions imposed by the site. Additionally, the scraping frequency was carefully controlled to prevent overloading the server and causing any disruptions to the website's performance.

Furthermore, the data extraction process focused solely on retrieving publicly available information without attempting to access any private or sensitive data. The code was crafted to avoid scraping any personal user information, adhering to strict privacy guidelines. All possible precautions were diligently taken to respect the legal terms and conditions set forth by booking.com, maintaining a high level of integrity and responsibility throughout the scraping project. By adhering to ethical practices, the code's development upheld a strong commitment to data legality and the protection of intellectual property rights.

References:

1. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=need+for+web+scraping&btnG=#d=gs_qabs&t=1689568686149&u=%23p%3DF0to5SeS9PEJ
2. https://www.researchgate.net/profile/Vlad-Krotov/publication/324907302_Legality_and_Ethics_of_Web_Scraping/links/5aea622345851588dd8287dc/Legality-and-Ethics-of-Web-Scraping.pdf
3. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=snell+and+menaldo+2016&btnG=#d=gs_qabs&t=1689570600409&u=%23p%3D018PufGTz_EJ
4. Reference for code:
   https://www.scrapingdog.com/blog/scrape-bookingcom/

About the writer

Name - Ayesha Rahman
Student ID – 201522771

Course details:

Under-graduation in Computer Science specialising in Artificial Intelligence.
Currently in year 3 (last year) at the University of Leeds.

Internship details:

Has done the internship at UTI in the summer of 2023.
(From July 10-2023 to September 03-2023)

Acknowledgement from UTI

Mentors:

- Miss Pradnya S. Ganar: Associate Vice President (Fund Management)

[redacted]

- Mr Parag Chavan: Sr Associate Vice President (Fund Management)

[redacted]