Hotel B2B Lead Scraper Project

Project Source

- **GitHub Repository:** GitHub Link For code(index.py is the main file Only)
- **Separate Excel File:** Excel File Link For User

Excel File Explanation

1. **Combined Sheet**:
    - **Name**: `combined`
    - **Content**: This sheet contains both hotel and leadership data combined.
    - **Structure**: Each row represents a unique leadership team member, with the corresponding hotel information duplicated for each leadership detail.
    - **Columns**: Includes all hotel information (e.g., name, address, phone number) along with leadership details (e.g., leader name, title, contact information).
2. **Hotel Offices Sheet**:
    - **Name**: `Hotel Offices`
    - **Content**: This sheet contains only the hotel information.
    - **Structure**: Each row represents a unique hotel.
    - **Columns**: Includes hotel details such as name, address, phone number, and a unique hotel ID in the first column.
3. **Leadership Details Sheet**:
    - **Name**: `Leadership Details`
    - **Content**: This sheet contains only the leadership team information.
    - **Structure**: Each row represents a unique leadership team member.
    - **Columns**: Includes leadership details such as name, title, contact information, and the corresponding hotel ID to match the `Hotel Offices` sheet.

Misunderstanding About No of Hotels Data

The website mentions that there are 1,061 hotel details, but this is incorrect. There are only 750 hotel details, spread across 75 pages with each page containing information on 10 hotels.

Project Overview

- **Purpose**: To scrape detailed information from over 750 hotels, including their names, locations, addresses, and more. Additionally, scrape leadership details for each hotel.
- **Output**: The scraped data is stored in two separate tables in an SQLite database and then combined using a SQL query to generate a comprehensive CSV file.

Features

- **Hotel Details Scraping**: Extracts information such as hotel name, location, address, and phone number.
- **Leadership Details Scraping**: Retrieves leadership information for each hotel, including names, titles, and contact details.
- **Database Storage**: Saves the scraped data into two separate tables in an SQLite database.
- **Data Joining and Export**: Combines the data from the hotel and leadership tables using a SQL query and exports the combined data to a CSV file.

Prerequisites

- **Software**: Python 3.x
- **Libraries**:
  - `requests`: For making HTTP requests.
  - `beautifulsoup4`: For parsing HTML and extracting data.
  - `json`: For handling JSON data.
  - `dataset`: For managing SQLite database operations.
  - `pandas`: For data manipulation and CSV export.

Installation

1. **Clone the Repository**:
   - Run the command: `git clone https://github.com/yourusername/HotelB2BLeadScraper.git`
   - Navigate to the project directory: `cd HotelB2BLeadScraper`
2. **Install Required Packages**:
   - Run the command: `pip install -r requirements.txt`

Usage Instructions

1. **Manual Cookie Update**:
   - Before running the script, manually update the cookies in the `headers` dictionary in the `index1.py` file. These cookies must be copied from your browser's request headers when accessing the target website.
2. **Run the Script**:
   - Execute the script using the command: `python index3.py`
3. **Script Operations**:
   - The script will scrape hotel details and leadership profiles.
   - It will save the data to an SQLite database.
   - Finally, it will join the tables and export the data to `joined_data.csv`.

Code Flow and Understanding

1. **Initialize Headers and Session**:
   - Headers containing necessary request details and cookies are defined.
   - A session object is created to handle the requests and maintain the session.
2. **Scrape Hotel List**:
   - The script iterates over multiple pages to scrape hotel details.

    - For each page, the following details are extracted: name, location, address, phone number, and more.
    - This information is stored temporarily for further processing.
3. **Scrape Leadership Profiles**:
    - For each hotel, the script extracts the hotel ID and uses it to scrape leadership details.
    - Leadership information includes names, titles, email addresses, phone numbers, and personal webpage links.
    - These details are associated with their respective hotels.
4. **Save to Database**:
    - The scraped hotel and leadership details are saved into two separate tables in an SQLite database: `offices` and `leaders`.
    - The `offices` table stores the hotel information.
    - The `leaders` table stores leadership information and includes a foreign key reference to the `offices` table.
5. **Join Tables and Export to CSV**:
    - A SQL query is used to join the `offices` and `leaders` tables based on the hotel ID.
    - The combined data is then exported to a CSV file named `joined_data.csv`.

Important Notes

- **Manual Cookie Update**: Each time you run the script, you must manually update the cookies in the `headers` dictionary. This step is necessary to ensure successful requests to the target website.
- **Error Handling**: The script includes basic error handling for failed requests and data extraction issues, ensuring that the scraping process continues even if some requests fail.

Conclusion

This project efficiently scrapes and processes data from multiple hotel web pages, stores the data in a structured manner, and provides a comprehensive output file for further analysis or use. The manual cookie update requirement ensures that the requests are authenticated and successful.

Request for Feedback

If you require any changes or have any feedback, please let me know. I am happy to improve the project and meet your requirements, as I offer unlimited revisions in my package.