Web Scraping Bot

Overview

The Web Scraping Bot is a Python application that automates the process of scraping product information from an e-commerce website. It uses the Selenium library for web scraping and data extraction. The scraped data is then stored in a MySQL database using the MySQL Connector library.

Installation

To run the web scraping bot, follow these steps:

1. Clone the project repository from GitHub:

```
git clone https://github.com/cls-talha/Daraz-Product-Scrapper.git
```

2. Install the required Python packages by running the following command:

```
pip install -r requirements.txt
```

3. Ensure that you have the Chrome web browser installed on your system.
4. Set up a MySQL database and provide the necessary credentials in the `database.py` file.

Usage

To use the web scraping bot, follow these steps:

1. Open the `bot.py` file and configure the logging options as needed.
2. In the `bot()` function, create an instance of the `Scrapper` class by providing the URL of the target e-commerce website and the product name to be scraped.
3. Run the `scrap_product()` method of the `Scrapper` instance to start scraping product information. Specify the number of pages to scrape (optional, defaults to 3).
4. After scraping, the product information is stored in a Pandas DataFrame.
5. Use the `to_dataframe()` method to convert the scraped data into a Pandas DataFrame.
6. Use the `show_dataframe()` and `show_dataframe_info()` methods to display the DataFrame and its information, respectively.
7. In the `bot()` function, create an instance of the `Database` class by providing the MySQL database credentials.
8. Run the `insert_data()` method of the `Database` instance to insert the scraped data into the database.

9. Use the `show_database()` method to display the contents of the database table.
10. Open the `app.py` file and set the `run_all_time` variable to `True` if you want the bot to run immediately, or set it to `False` to schedule the bot to run every Tuesday at 9:00 AM.
11. Run the `app.py` file to start the bot and schedule it to run periodically.

Limitations

- The web scraping bot is designed to work with the specific structure and elements of the target e-commerce website. Any changes to the website's structure may require modifications to the code.
- The bot is currently configured to scrape product information from a single website. To scrape from a different website, you may need to modify the code accordingly.
- The database connection is assumed to be local. If you're using a remote database, update the host information in the `Database` class.

Conclusion

The Web Scraping Bot provides a convenient way to automate the process of scraping and storing product information from an e-commerce website. By using the provided classes and scripts, you can easily customize and extend the functionality to suit your specific requirements.

Please note that web scraping should be done responsibly and in accordance with the website's terms of service. Be mindful of the website's policies and ensure that you have the necessary permissions before scraping any data.