Objective or Purpose:

This is a runbook for a piece of software (The_Twitter_Soul_Harvester_3000) that can scrape the latest tweets for any topic.

Procedure:

1. Download & install the necessary libraries
    a. Selenium
    b. Pandas

2. Selenium uses a bot controlled browser to do all its tasks called the webdriver. You have to install a specific web driver for the kind of browser you want, like different web drivers exist for chrome, firefox, edge etc. (you can find the latest webdrivers here: https://www.selenium.dev/documentation/webdriver/)

3. The path variable in the second cell of jupyter notebook, requires the path in which your webdriver exists.

4. For running the twitter scraper, you must have a twitter account, the scraper will use this account to scrape the tweets. The chrome profile directory variable in the second cell contains the path to this profile. Make sure this profile has autosign enabled for the twitter profile you want your bot to use for scraping. To do this, sign in with the twitter account with this chrome profile and select stay signed in option.

5. The first cell contains imports and function definitions.

6. The second cell opens up the webdriver and opens twitter with the chosen topic from the user. The bot searches that topic on the twitter search bar and goes to the latest tab.

7. The third cell uses the already opened webdriver to scrape the tweets from that page. It will scrape all the tweets for that topic by scrolling all the way to the end.

8. By default, I have set the webdriver to be invisible, but if you wanna see the scraping in action, go to the second cell and comment out the following line “chrome_options.add_argument("--headless")”

9. The second and third cell can be clubbed into a single cell, for a continuous scraping experience.

10. Once the data is scraped, you'll know this when the hourglass icon disappears from the tab icon of jupyter notebook. You can check for the len(data) which shows how many tweets you have. You can also convert the data into a dataframe.

11. After converting the data into a pandas dataframe, the line after it converts it into a csv file.

12. Make sure you change the csv file name each time for new tweets, otherwise older data will be overwritten.

13. The remaining cells are for debugging purposes, which may be useful to you in the future, if twitter decides to change its website structure.

Tools and Resources:

1.Selenium may deprecate its methods, so to keep the code updated check the documentation: https://selenium-python.readthedocs.io/

2. This video helped me with most of my code: https://www.youtube.com/watch?v=3KaffTIZ5II&t=749s&ab_channel=IzzyAnalytics

But keep in mind, that this person uses deprecated method, and the twitter webpage structure has changed since this video, but the overall logic of the scraper remains intact and is contained within this video.

Troubleshooting:

1. Twitter may change how the tweets are organized within a webpage, so use the debugging cells mentioned above to gauge where the tweets in the page are, and then make relevant changes to the main code.
2. If you are using the visible webdriver version, then do not click anything on the bot's browser, this will interrupt the scraping process and result in an error.