Twitter Scraping System

System Overview:

- **Description:** This system performs three main functions: extracting tweet data from Twitter, checking if the extracted data is already saved in the database, and saving new tweet data to the database if it does not already exist.
- **Components:**
  - **TwitterScraper:** Responsible for scraping tweet data from Twitter.
  - **SaveToDB:** Handles saving tweet data to a MySQL database.
  - **Main:** Executes the main functionality of the system.

Functionality:

1. **Extract Tweet Data from Twitter:**
   - The system utilizes the TwitterScraper component to extract tweet data from Twitter. It navigates to the specified URL and collects relevant tweet information.
2. **Check Database for Existing Data:**
   - Upon extracting tweet data, the system checks the database to determine if the data is already saved. It queries the database using the tweet ID to find matching records.
3. **Save New Tweet Data to Database:**
   - If the extracted tweet data is not already present in the database, the system uses the SaveToDB component to insert the data into the database. It ensures data integrity and avoids duplicates.

Error Handling:

- The system includes robust error handling mechanisms to handle potential exceptions during data extraction, database operations, and network-related issues.

Scalability Considerations:

- Modular design allows for easy scalability by adding additional components or deploying on distributed systems as demand grows.

Usage:

1. Ensure you have the necessary dependencies installed (e.g., mysql-connector-python, selenium, requests.).
2. Run the main.py file to execute the system.
3. The system will scrape tweet data from the specified URL, check if it already exists in the database, and save new data if necessary.

Configuration:

- Update the database connection details (host, user, password, database name) in the SaveToDB class.
- Ensure the appropriate driver (e.g., Chrome WebDriver) is installed and configured for Selenium.

Development Steps:

1. **Analyze the website:** In this step, open the browser, search for a Twitter post, inspect how the request is sent, then take the endpoint URL and call it using an API caller. Repeat this process with multiple requests.
2. **Choose the suitable technique:** Determine that the suitable technique involves extracting cookies and tokens with Selenium, then putting them in request headers and sending them.
3. **Inspect the response and extract desired data from it:** Use an online JSON formatter to display the response in a clear way, then determine the keys to deal with.
4. **Create a connection with MySQL DB** and write code to check if this tweet is in the DB or not.
5. **Test the code repeatedly.**
6. **Refactor the code and test it again.**