Scraping Faculty Information

Introduction:

This document provides an overview of the Python program developed for scraping faculty information from the University of Management and Technology (UMT) website. The program utilizes various libraries such as `requests`, `BeautifulSoup`, `pandas`, and `concurrent.futures` to efficiently extract and structure data from the UMT faculty page.

Program Overview:

The program is designed to perform the following tasks:

1. **Scrape Faculty Names and URLs:**
   - Fetches faculty names and their corresponding profile URLs from the UMT faculty page.
2. **Multithreaded Scraping:**
   - Utilizes multithreading with 30 worker threads for concurrent scraping of faculty profile information.
3. **Retrieve Lecturer Profile Information:**
   - Retrieves detailed information about each lecturer (name, designation, department, and email) from their respective profile pages.
4. **Decoding Encrypted Emails:**
   - Decodes encrypted email addresses on the faculty pages using custom decoding functions.
5. **Data Structuring:**
   - Structures the extracted data into a pandas DataFrame.
6. **Data Export:**
   - Exports the final structured data to a CSV file named `all_faculty.csv`.

Program Components:

1. **Libraries Used:**
   - `requests`: For sending HTTP requests and handling responses.
   - `BeautifulSoup`: For parsing HTML documents.
   - `pandas`: For data structures and analysis.
   - `concurrent.futures`: For multithreading tasks.

2. **Functions:**
    - `cfDecodeEmail(encodedString)`: Decodes an email address encoded with Cloudflare's protection mechanism.
    - `encrypted_email_extraction(enc_email)`: Extracts encrypted email addresses from input strings.
    - `get_lecturer_desc(lecturer_doc)`: Extracts and structures lecturer information from parsed HTML.
    - `get_lecturer_page(lecturer_url)`: Retrieves and parses the HTML document of a lecturer's profile page.
    - `scrape_lecturer_info(row)`: Scrapes detailed information about a lecturer based on their profile URL.
    - `get_faculty_url(doc)`: Extracts faculty URLs from parsed HTML documents.
    - `get_faculty_name(doc)`: Extracts faculty names from parsed HTML documents.
    - `scrape_names()`: Scrapes lecturer names and their corresponding URLs from the UMT faculty page.
    - `main()`: Main function orchestrating the scraping process.

Usage Instructions:

To run the program, execute the `main()` function. Ensure the following:

- The necessary libraries (`requests`, `BeautifulSoup`, `pandas`) are installed in your Python environment.
- Internet connectivity is available to fetch data from the UMT website.
- The program will generate a file named `all_faculty.csv` containing the scraped data.

Conclusion:

This Python program efficiently scrapes faculty information from the UMT website, utilizing multithreading for faster execution. The extracted data is structured and saved in a CSV file for further analysis and use.

Happy Coding!