Report

Advance Image Downloader/Extractor

By : Vipin kumar

Introduction:

The purpose of this design is to design a job to download thousands of images from the internet based on given requirements. The job will use Python and web scraping ways to extract the images. The user will input the details similar as the name of the images to be downloaded, the schedule time, and the email id. The job will run at the specified schedule time and after its completion, the requester will receive an email with a URL to download the extracted images.

Approach:

The project will follow the following approach to accomplish its objectives

- Designing a python script to extract the images from the internet.
- Creating a scheduler to perform the scraping process.
- Accepting inputs from the user similar as the name of the images, schedule time, and email id.
- Running the job at the specified schedule time.
- After the completion of the job, sending an email to the requester with a URL to download the extracted images.

Design and Implementation:

The following way were taken to apply the design:

- Designing the Python script: A python script was created to extract the images from the internet with the help of selenium and request library based on the input requirements. The script used web scraping techniques to extract the images.
- Creating the Scheduler: A scheduler was created to run the scraping process at the specified schedule time. The user inputs similar as the name of the images, schedule time, and email id were accepted by the scheduler. We created a scheduler with the help of Apscheduler library.
- Running the Job: The job was run at the specified schedule time, and the extracted images were saved in a position specified by the user.

- Sending the Email: After the completion of the job, an email was sent to the requester with a URL to download the extracted images.(As of now we stored the extracted images to the local machine so in the URL its only providing the location of the images.)

Conclusion:

The design was successful in accomplishing its objects. A job was created to download thousands of images from the internet based on the given conditions. The job used Python and web scraping ways to extract the images. The scheduler was created to run the job at the specified schedule time, and after its completion, the requester received an email with a URL to download the extracted images. The design demonstrated the potential of Python and web scraping ways in automating image extraction processes.