MLOps Assignment 1

SHIVA SURYA C M - PH21B009

SHIVA SURYA CM

02.02.2025

Module 1

- In Module 1 , a function called `"module_1_web_scrapping_with_lazy_loading"` has been created to accept an url as a parameter and return the scrapped website
    - Input : URL
    - Output : HTML Scrapped (using beautifulsoup)
- Parameters:
    - url - The url of the page to be scrapped (here it is google news website)
- Lazy Loading: ***Lazy loading is handled and selenium with chrome driver*** is used to handle lazy loading ***via scrolling action***

Module 2

- In Module 2, a function called `"module_2_stories_link"` has been created to get the links from the top stories and also all the stories under the section
    - Input : Scrapped bs4 object from m1
    - Output : Links for all the stories under top stories section
- Parameters :
    - soup - bs4 object from previous module
    - section - The section under which we want to get the links (Here it is "stories")

Module 3

- In Module 3 , we have two functions `"module_3_thumbnail_img_extraction" and "download_and_store_image"`
- ***`module_3_thumbnail_img_extraction :`***
    - This function gets the list of extracted urls and loads the corresponding website and outputs the list containing links of thumbnails and the headline
    - **Lazy loading is handled using selenium and scrolling action**
    - Input Parameters:
        - List[top stories links]
        - class name of the article
        - class name of the thumbnail url

        - class name of the headline
    - Output: (output is a list of dict)
        - List[{“headline” : $headline text$, “image_url” : url}]
- ***`download_and_store_image:`***
    - This function is used to download the all images and also ***assigns a unique id for each image***, such that it won’t collide with any precious ids (basically it will be unique for all)
    - It stores the images in the given root directory for future use in m4
    - Input Parameters:
        - output from the previous function
        - root_img - (root dir where you will store the images)
    - Output:
        - List[dict]
        - dict - {"headline" : image_headline, "image_id" : image_id, "image_url" : image_url}

Module 4 and 5

- MongoDB is being used for storing all the headlines and images (in binary and hashed format)
- It also has two functions : `”connect_database”,` `“module_4_and_5_store_in_database”`
- `connect_database`:
    - It accepts a host parameter as input and created two table headline_table and img_table
    - Input Parameters:
        - host (default = `"mongodb://localhost:27017/"`)
    - Output:
        - tuple : (headline_table, img_table)
        - Structure of ***headline_table***
            - {
            - "headline": current_headline,
            - "image_url": current_url,
            - "image_id": current_image_id (name of the image),
            - "image_index": index of the image in image_table,
            - }

        - Structure of ***image_table***
            - {
            - "image_headline_hash" : current_img_hash + current_headline_hash,
            - "image_bin" : binary_image
            - }
    - To avoid duplicates, the image_headline_hash is set as unique
    - *Duplicate checking* : Using ***hashlib library in python , the sha256 hash of the headline and image is created*** , so duplicate rows are avoided
    - The “image_headline_hash” is set a unique
- **`module_4_and_5_store_in_database:`**
    - This function is used to populate the database
    - Input Parameters:
        - root_img (from which the stored images will be taken)
        - Outputs from the previous function
        - Extracted_data (the data used to populate the rows)

Module 6

- In Module 6 , all 5 modules are imported and connected together which automates the entire process
- Python logger is created to log the status like info, error, exceptions encountered in the process
- M6 is created in such a way that , it can be easily incorporated with cron job or task scheduler