Summary and Challenges

1. Before running the main code please install the necessary dependencies mentioned in *requirements.txt* file by running the following command “*pip install –r requirements.txt*”. This will help us to install all the dependencies in one go.
2. **Imports**:  
   The code starts by importing essential libraries: *requests, BeautifulSoup, urllib.parse*, csv, *and time*. These libraries are crucial for tasks like making HTTP requests, parsing HTML content, encoding URLs, writing data to CSV files, and generating timestamps.
3. **Function Definitions**:
   * *search\_google(query)*: This function constructs a Google search URL with a given query, sends a GET request to Google, parses the HTML response to extract URLs from the search results, and returns a list of extracted URLs. It relies on **requests** for making the *HTTP request and BeautifulSoup* for parsing the HTML response.
   * *scrape\_website(url):* Given a URL, this function sends a GET request, parses the HTML content of the page using *BeautifulSoup,* and extracts the title and paragraphs. It returns a dictionary containing the URL, title, and paragraphs.
4. **Main Function** (**main()**):
   * It defines a list of search queries related to Canoo and generates a timestamp using *time.strftime()* to create a unique name for the CSV file.
   * Opening a CSV file with the specified name in write mode *('w')* ensures consistent line endings across platforms.
   * Creating a *csv.DictWriter* object with specified field names writes the header row to the CSV file.
   * Iterating through each query, it searches Google for the query and extracts relevant URLs, filters out irrelevant URLs, and iterates through each URL to scrape data.
   * Scraped data (query, URL, title, and paragraphs) is written row by row to the CSV file using the *writerow()* method of *csv.DictWriter.*
   * Exception handling is implemented to capture and print any errors that occur during scraping.
   * A message indicating the completion of scraping and the name of the generated CSV file is printed.
5. **Execution**:
   * The code checks if it is being run as the main program using *if \_\_name\_\_ == "\_\_main\_\_":* and then calls the *main()* function to initiate the scraping process.
6. **Challenges**:
   * Challenges arose with DuckDuckGo providing *HTTP status codes like 403 and 401* for different URLs, causing issues. To overcome this, *switching to the Google Search API* was considered a solution, which likely helped bypass such limitations