**Data Engineer Assessment**

**Given**

**Raw Data:** User clicks data in JSON format (user\_click\_data.json).

**Schema of the Raw Data:**

* **click\_event\_id:** Unique identifier for each click event
* **user\_id:** The ID of the user who is clicking the link
* **url:** The link that has been clicked by the user
* **timestamp:** The time when the click event happened
* **city:** The city from which the user accessed the link
* **country:** The country from which the user accessed the link
* **ip\_address:** IP address of the device the user used to access the link
* **device:** The type of device used
* **browser:** The browser from which the link is accessed

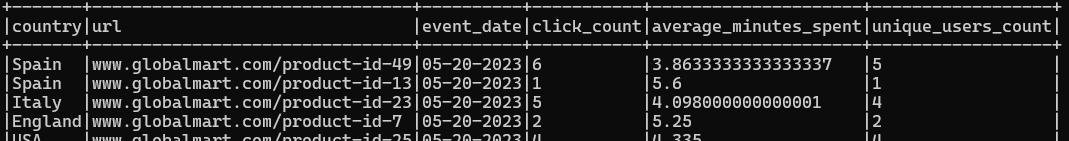
**Task**

Develop a Spark application to perform ETL on the provided raw data:

1. **Data Type Correction:** Ensure all fields have the correct data types.
2. The final output DataFrame should contain the following:
   * The average time (in minutes) users from each country spent on each web page (URL) for each date.
   * The total number of unique users from each country who visited each web page (URL) for each date.
   * The total number of clicks each web page (URL) garnered from each country for each date.

Push the PySpark code to GitHub, upload the final output data after transformation, and share the link of the GitHub repository.

**Output DataFrame:**



**Interpretation of the Table:** The URL '[www.globalmart.com/product-id-49](http://www.globalmart.com/product-id-49)' was accessed from Spain on '2023-05-20' by 5 distinct users. Each of them spent around 3.86 minutes on that webpage. The link garnered 6 clicks on that day from that country.

**Additional Points**

Not mandatory, but preference will be given to candidates who can complete the following tasks:

1. **PostgreSQL Integration:**
   * Host a PostgreSQL server using a Docker image locally.
   * Create a table with a schema that matches the raw data schema (user\_click\_data.json).
   * Load the raw data into PostgreSQL.
   * Read the data from PostgreSQL using Spark and perform the required transformations.
2. **Hive Integration:**
   * Host a Hive server using a Docker image locally.
   * Write the final output DataFrame to Hive, ensuring best practices for query performance and data management are followed.