Milestone Report for Workflow Automation

# Step 1: Analyze and Filter Keywords from the Ranking Sheet

Objective:  
 - Extract and filter the necessary columns (RID and Keywords) from the Ranking Sheet and store them in a new Excel file called Final\_keywords.  
  
 Achieves:  
 - Data Cleansing: Extracts and prepares essential data from a large ranking sheet, focusing only on RID (Report ID) and Keywords for further analysis.  
 - Efficient Data Extraction: Automates the extraction process, avoiding manual handling and saving time when dealing with large datasets.  
  
 Tech Stack:  
 - Language: Python  
 - Libraries:  
 - pandas: For reading and writing Excel files.  
 - Agno/Phidata (AI Agent): To automate extraction based on predefined criteria (RID and Keywords).  
   
 Task Achieved:  
 - Keyword Extraction: Only the relevant RID and Keyword data is selected and stored in a new file, Final\_keywords.  
 - Preparation for Keyword Analysis: Sets up the data for the next step (SerpAPI analysis).

# Step 2: Analyze Keywords Using SerpAPI (Conditions and Filters)

Objective:  
 - Use SerpAPI to analyze each keyword in Final\_keywords and apply two conditions:  
 1. Condition 1: The "Interest over time" must cross the Y-axis value of 50 at least twice in a 7-day period.  
 2. Condition 2: "Interest by region" must include at least one country in the top 5 from the list (China, USA, UK, France, South Korea, Japan, Brazil).  
   
 Achieves:  
 - Validating Keyword Relevance: The agent only selects keywords that meet both conditions, ensuring that only highly relevant keywords are retained for further processing.  
 - Data Cleansing: Filters out less relevant keywords, significantly improving the quality of the data being processed in the following steps.  
   
 Tech Stack:  
 - SerpAPI: For querying keyword data.  
 - Python: To interact with the API, process data, and filter.  
 - Libraries:  
 - requests: For making API requests to SerpAPI.  
 - pandas: For reading, filtering, and writing Excel files.  
   
 Task Achieved:  
 - Keyword Validation: Filters and retains only those keywords that meet the conditions, ensuring that the final data is relevant and adheres to the specified criteria.

# Step 3: Process the ROB Excel File

Objective:  
 - After receiving the ROB Excel file from the backend team, remove unnecessary columns and clean the data.  
   
 Achieves:  
 - Data Cleanup: This step ensures that only the necessary data is retained in the ROB file by removing redundant or irrelevant columns.  
 - Preparation for Web Application: The cleaned data is now ready to be used in the in-house web application for generating V4 articles with CTAs.  
   
 Tech Stack:  
 - Python: The core language for automating data cleaning.  
 - Libraries:  
 - pandas: To load, clean, and save Excel files.  
   
 Task Achieved:  
 - Data Reduction: Simplifies the ROB file, making it easier to work with and reducing the processing load for the next steps.

# Step 4: Generate CTAs and Article via Web App

Objective:  
 - The cleaned ROB file is passed to the in-house web app, which generates V4 articles along with CTAs (Call to Actions).  
   
 Achieves:  
 - Automating Article Generation: The web app generates high-quality articles based on the cleaned ROB data. This saves time and ensures consistency in the content creation process.  
 - CTA Integration: Each article generated contains CTAs, which are essential for guiding the readers toward actionable steps (e.g., purchasing or requesting more information).  
   
 Tech Stack:  
 - Web Application: The backend processing and article generation logic are handled by the web app.  
 - Framework: Could be built using frameworks like Flask or Django (depending on your tech stack).  
   
 Task Achieved:  
 - Content Creation: Automatically generates articles with integrated CTAs, ensuring that all generated content meets the required standards.

# Step 5: Automate Article Publishing on OpenPR Platform

Objective:  
 - Use the provided OpenPR credentials to automate the publishing of the article generated in Step 4.  
   
 Achieves:  
 - Article Publication Automation: Automatically publishes the article to the OpenPR platform, reducing manual effort and ensuring the timely release of the article.  
 - Integration with OpenPR: Streamlines the entire publishing process, from article generation to live publication on the platform.  
   
 Tech Stack:  
 - Selenium: For automating web interactions (e.g., logging into OpenPR, submitting the article).  
 - Python: For scripting the automation process.  
 - Libraries:  
 - selenium: For automating web browser tasks.  
 - requests: If OpenPR offers API access.  
   
 Task Achieved:  
 - Seamless Publishing: Ensures that the generated articles are automatically published without manual intervention.

# Summary of Workflow Achievements

|  |  |  |
| --- | --- | --- |
| Step | Objective | What It Achieves |
| Step 1 | Extract \*\*RID\*\* and \*\*Keywords\*\* from \*\*Ranking Sheet\*\* | Cleanses and prepares the data for further analysis. |
| Step 2 | Analyze keywords using \*\*SerpAPI\*\* with two defined conditions | Filters out irrelevant keywords, ensuring high-quality data for publication. |
| Step 3 | Remove unnecessary columns from \*\*ROB Excel\*\* | Reduces the size and complexity of the data, making it easier to work with. |
| Step 4 | Generate \*\*V4 articles\*\* with \*\*CTAs\*\* from \*\*ROB data\*\* | Automates the article creation process, saving time and ensuring consistency. |
| Step 5 | Use \*\*OpenPR credentials\*\* to automate article publishing | Ensures timely and efficient publishing of articles without manual effort. |