SEO Opportunity Automator: User Guide

Version 1.0

Welcome to your new SEO Opportunity Automator! This guide will walk you through everything you need to know, from the initial technical setup to strategic daily use, all in simple, easy-to-understand language.

Note: This automation is a powerful tool designed to run on your local machine. Please follow the setup instructions carefully.

Sheet Link: Page Optimizing Myntra

Table of Contents

Chapter 1: Introduction to the Tool

What is this Automation Tool?

Imagine you have a list of 100 keywords you want to rank for on Google. For each one, you need to ask:

1. Are we already ranking for this keyword? If so, where?
2. If we are ranking, is the page fully optimized?
3. If not, what specific optimization does it need (Title, Content, etc.)?

Doing this manually is incredibly time-consuming and repetitive. This tool solves that problem.

It is a smart system built with Python and Selenium that automates the entire process. It reads a keyword from your Google Sheet, finds its current Google rank for your website, and then visits the ranking page to perform a detailed on-page SEO analysis. It identifies specific opportunities—like missing content or a poorly written title—and records its findings directly back into your Google Sheet.

Who is this Tool For?

This tool is designed for anyone involved in improving a website's search engine performance. You do not need to be a professional coder to use it, but you should be comfortable following setup instructions.

It's perfect for:

- **SEO Analysts:** To quickly audit hundreds of keywords and identify low-hanging fruit for optimization.
- **Content Strategists:** To find pages that need content expansion or updates.
- **Digital Marketers:** To track ranking progress and its correlation with on-page factors.
- **Website Managers:** To get a clear, actionable to-do list for improving site SEO.

Key Features at a Glance

- **Fully Automated Ranking Checks:** Just provide the keyword and target domain; the tool finds the rank.

- **Intelligent On-Page Analysis:** Automatically checks for common SEO issues related to content, titles, meta descriptions, and product availability.
- **Centralized Management:** Your entire workflow—inputs and results—is managed from a single Google Sheet.
- **Human-like Browsing:** Uses a real browser with delays and a persistent profile to minimize the risk of being blocked by Google.
- **Progress Tracking:** Automatically marks keywords as "Completed" so you never process the same one twice unless you want to.
- **Detailed Logging:** Creates a local log file (`main_automator.log`) that records every step, making it easy to see what the tool is doing.
- **Error Notifications:** Can be configured to send you an email if the script crashes or encounters a CAPTCHA, so you're always in the loop.

How It Works: A Simple Overview

Think of the tool as a very diligent junior SEO assistant. You give it a list of tasks in a Google Sheet, and it gets to work.

1. **You Provide the List:** You add keywords and your website's domain to the 'kwd optimization' sheet.
2. **The Automation Wakes Up:** You run the script from your computer.
3. **It Checks a Keyword:** The script picks the first keyword on your list that isn't marked "Completed".
4. **It Searches Google:** It opens a Chrome browser, searches for the keyword, and looks for your website in the results.
5. **It Records the Rank:** Once found, it notes the rank and the exact URL and updates the Google Sheet.
6. **It Analyzes the Page:** It then visits that URL and performs an internal search for the keyword. It checks the resulting page against a list of SEO rules (e.g., "Does this page have enough content?").
7. **It Recommends an Action:** Based on its analysis, it decides if the keyword is irrelevant ("Deletion"), if the page needs title/meta work ("T&M"), or if it needs more content ("Content"). This is also updated in the sheet.
8. **It Logs Its Work:** Every major action is recorded in a local text file for your review.
9. **Repeat:** The script moves to the next keyword and starts the process all over again.

This cycle repeats until every keyword on your list has been processed.

Chapter 2: Getting Started: The Complete Setup Process

This chapter will guide you through the one-time setup of your automation tool. Follow these steps carefully to ensure everything works perfectly.

Prerequisites: What You Need Before You Start

1. **Python:** Ensure you have Python installed on your computer (version 3.8 or newer is recommended). You can download it from python.org.
2. **Code Editor:** A code editor like VS Code is highly recommended for editing configuration files.
3. **Google Account:** You will need a standard Google account.

Part A: Google & Cloud Setup

Step 1: Make a Copy of the Google Sheet

This sheet is the brain of your operation.

1. Open this link to the master Google Sheet template: https://docs.google.com/spreadsheets/d/1Ssi1XtOXP7AF_9xk1aV6WuIbKOC9-TB83Hl1hsB3cOs/edit?gid=0#gid=0
2. In the menu, click **File > Make a copy**.
3. Give your copy a new name (e.g., "My Website - SEO Automator") and save it to your own Google Drive.
4. **Note:** We have provided a sample CSV file (`Page Optimizing Myntra - kwd optimization.csv`) in the project folder. You can use this to understand the required format or upload it to your sheet by going to **File > Import > Upload**.

Step 2: Set Up Your Google Cloud Project (GCP)

The script needs API access to read and write to your Google Sheet. This is free.

1. Go to the Google Cloud Console.
2. Click the project dropdown menu at the top of the page and click **NEW PROJECT**.
3. Give your project a name (e.g., "My SEO Automator") and click **CREATE**.

4. Make sure your new project is selected in the dropdown.
5. In the search bar at the top, search for and enable these two APIs:
    - **Google Drive API** (Click "Enable")
    - **Google Sheets API** (Click "Enable")

Step 3: Get Your GCP Credentials (The

This step creates a special "robot" user (a service account) that the script will use.

1. In the Google Cloud Console, click the navigation menu (☰) in the top-left.
2. Go to **IAM & Admin > Service Accounts**.
3. Click **+ CREATE SERVICE ACCOUNT**.
4. Give it a name (e.g., "sheets-editor-bot") and a description. Click **CREATE AND CONTINUE**.
5. For the role, select **Project > Editor**. Click **CONTINUE**.
6. Skip the last step and click **DONE**.
7. You will now see your new service account in the list. Click the three dots under "Actions" and select **Manage keys**.
8. Click **ADD KEY > Create new key**.
9. Choose **JSON** as the key type and click **CREATE**.
10. A `.json` file will be downloaded to your computer. **This file is very important.**

Step 4: Share Your Google Sheet with the Service Account

You need to give your new "robot" user permission to edit your sheet.

1. Open the downloaded `.json` file with a text editor.
2. Find the line that says `"client_email"` and copy the email address next to it (it will look something like `...gserviceaccount.com`).
3. Go to your copy of the Google Sheet.
4. Click the **Share** button in the top-right corner.
5. Paste the copied email address into the sharing box, give it **Editor** permissions, and click **Send**.

Part B: Local Machine & Python Setup

Step 5: Set Up Your Project Folder

1. Download and unzip all the project files (`main_automator.py`, `config.py`, etc.) into a single folder on your computer.
2. Move the `.json` credentials file you downloaded in Step 3 into this same folder.
3. Rename the `.json` file to `gcp_credentials.json`.

Step 6: Create a Python Virtual Environment (venv)

This creates an isolated space for your project's libraries.

1. Open a terminal or command prompt.
2. Navigate into your project folder using the `cd` command. (e.g., `cd C:\Users\YourName\Documents\SEO_Automator`).
3. Run the following command to create a virtual environment named `venv`:
4. codeBash
5. python -m venv venv
6.
7. Activate the environment:
    - **On Windows:** `venv\Scripts\activate`
    - **On macOS/Linux:** `source venv/bin/activate`
8. You will see `(venv)` appear at the beginning of your command prompt line.

Step 7: Install Required Libraries

With your virtual environment active, run this command to install all the necessary Python packages from the `requirements.txt` file:

```
pip install -r requirements.txt
```

Part C: Final Configuration & Activation

Step 8: Configure the

This file contains all the important settings for the script. Open `config.py` in your code editor and update the following:

- `SHEET_NAME`: The exact name of your Google Sheet copy (e.g., "My Website - SEO Automator").

- `WORKSHEET_NAME`: The name of the tab within your sheet you want to use (default is "kwd optimization").
- `ENABLE_EMAIL_NOTIFICATIONS`: Set to `True` if you want email alerts for crashes or CAPTCHAs, or `False` to disable them.
- `SENDER_EMAIL` & `SENDER_PASSWORD`: If email is enabled, enter the Gmail address and "App Password" you want to send alerts from. (Note: You need to generate an App Password from your Google Account security settings, not your regular password).
- `RECIPIENT_EMAIL`: A list of email addresses that should receive the alerts.

Step 9: Create Your Master Browser Profile

This one-time step creates a dedicated Chrome profile with your Google login saved, which helps avoid detection.

1. In your terminal (with the `venv` still active), run the `create_master_profile.py` script:
2. python create_master_profile.py
3. A new Chrome browser window will open. **You have 90 seconds to perform the following actions manually:**
   - Go to `google.com`.
   - Sign in to your Google account.
   - Accept any cookie pop-ups.
   - When you are finished, **close the browser window manually**.
4. The script will create a `Chrome-Master-Profile` folder in your project directory. Do not delete this folder.

Step 10: Activating the Automation

You are now ready to run the main tool.

1. Make sure your `venv` is active in the terminal.
2. Run the `main_automator.py` script:
3. codeBash
4. python main_automator.py

Congratulations! Your SEO Opportunity Automator is now fully set up and running. You will see its progress being logged in the terminal and the results appearing in your Google Sheet.

Chapter 3: Using the Automation Day-to-Day

Now that the setup is complete, using the tool is simple.

Your Daily Workflow

1. **Add New Keywords:** Open your Google Sheet. Add new keywords to the `Keyword` column and the target domain to the `Company1` column.
2. **Leave Status Blank:** For new keywords, make sure the `Processing Status` column is empty. The script uses this to find new work.
3. **Run the Script:** Open your terminal, navigate to the project folder, activate your `venv`, and run the main script: python main_automator.py
4. **Check Your Progress:** Watch the terminal for real-time updates. You can also check your Google Sheet to see the `Rankings`, `Ranking URL`, and optimization columns get filled in. The `Processing Status` will change to "Completed" once a row is finished.
5. **Stop the Script:** To stop the automation at any time, simply press **Ctrl + C** in the terminal window.

Understanding the Log File

A file named `main_automator.log` is created in your project folder. This is your detailed activity record. If you're wondering why a keyword was processed a certain way, or if an error occurred, this is the first place to look. It contains timestamps and detailed messages for every step the script takes.

How to Handle CAPTCHAs

Occasionally, Google may present a "I'm not a robot" CAPTCHA. The script is designed to handle this gracefully.

1. **The Script Pauses:** The script will detect the CAPTCHA and pause its execution.
2. **You Get an Alert:** If you enable email notifications, you will receive an email alerting you that your action is required.
3. **Solve it Manually:** Go to the Chrome browser window that the script opened. Solve the CAPTCHA puzzle.
4. **The Script Resumes:** Once solved, the script will automatically detect that the CAPTCHA is gone and will resume its work.

Re-processing Specific Keywords

If you want the script to re-analyze a keyword that is already marked "Completed":

1. Go to the Google Sheet.
2. Find the row for that keyword.
3. Simply **delete the text "Completed"** from the `Processing Status` column.
4. The next time you run the script, it will see the empty status and process that row again.

Chapter 4: Strategies & Customization

This tool is powerful out-of-the-box, but with a few strategic tweaks, you can tailor it perfectly to your needs.

Strategic Keyword Selection

The quality of your output depends on the quality of your input. For best results, focus on:

- **High-Intent Keywords:** Prioritize keywords that are likely to lead to a conversion (e.g., "buy black running shoes" vs. "shoes").
- **Long-Tail Keywords:** Target more specific, lower-competition phrases (e.g., "waterproof trail running shoes for men"). These often present easier optimization opportunities.
- **"Striking Distance" Keywords:** Use a tool like Google Search Console to find keywords where you already rank on page 2 or 3. A small on-page optimization can often push these to page 1.

How to Customize: Adapting for a Different Website

The tool is pre-configured for Myntra, but the on-page analysis logic can be adapted for almost any e-commerce or content site. This is an advanced step that requires changing the code.

1. **Open** This file contains all the website-specific logic.
2. **Update the CSS Selectors:** At the top of the file, you will find variables like `INTERNAL_SEARCH_SELECTOR`, `DELETION_SELECTOR`, `PRODUCT_COUNT_SELECTOR`, and `SEO_CONTAINER_SELECTOR`.
3. **Find the New Selectors:** Go to the new target website. Use your browser's "Inspect" tool to find the correct CSS selectors for that site's search bar, "no results" message, product count display, and main content area.
4. **Replace the Values:** Update the variables in the script with the new selectors. You may also need to adjust the logic inside the functions (`check_for_deletion`, etc.) if the new site behaves differently.

How to Customize: Adjusting On-Page Analysis Rules

You can easily change the thresholds for the on-page analysis.

- **To change the minimum product count:**
  1. Open `page_optimizer.py`.
  2. Find the `is_product_count_sufficient` function.
  3. Change the line `if product_count < 13:` to your desired number (e.g., `if product_count < 10:`).

- **To change the minimum content word count:**
    1. Open `page_optimizer.py`.
    2. Find the `check_for_content_optimization` function.
    3. Change the line `if word_count < 250:` to your desired word count (e.g., `if word_count < 300:`).

How to Customize: Modifying Scraping Behavior

You can control how many pages of Google results the script will check.

1. Open `google_rank_finder.py`.
2. Find the variable `MAX_PAGES_TO_SCRAPE` at the top of the file.
3. Change its value. For example, setting it to `3` will make the script check the first three pages of Google results before giving up.

Chapter 5: How the Magic Happens (A Simple Look at the Scripts)

You don't need to understand code to use this tool, but it can be helpful to know what's happening behind the scenes.

What are the Core Python Scripts?

- **(The Orchestrator):** This is the main script you run. Its job is to connect to Google Sheets, read the keywords, and call the other scripts to do the actual work. It manages the overall process.
- **(The Detective):** This script's only job is to perform the Google search. It's designed to act like a human, with realistic typing speeds and delays, to find the rank of your URL.
- **(The Analyst):** Once the Detective finds the ranking URL, this script takes over. It visits the page and runs the on-page SEO analysis funnel to identify optimization opportunities.
- **(The Settings Menu):** This file holds all your personal settings, like the sheet name and email details, so you don't have to change them in the main code.
- **&** These scripts help create and refresh your dedicated Chrome browser profile to ensure a stable, logged-in state.

The Journey of a Keyword: The Script's Logic

When the script runs, it follows a logical, step-by-step process for each keyword:

1. **Wake Up & Read:** The script starts and reads all the data from your Google Sheet.
2. **Pick a Task:** It looks for the first row where the `Processing Status` is empty.
3. **Ask Question 1: Should I process this?** If the status is "Completed", it skips to the next row. If it's empty, it proceeds.
4. **Launch Browser & Search:** It opens a Chrome window using the master profile and searches Google for the keyword.
5. **Ask Question 2: Is the URL on this page?** It scans the search results. If it finds your domain, it records the rank and the exact URL. If not, it clicks to the next page (up to the `MAX_PAGES_TO_SCRAPE` limit).
6. **Update the Sheet (Part 1):** It writes the found rank and URL back to the Google Sheet.
7. **Analyze the Page:** It navigates to the ranking URL, performs an internal search, and then runs its analysis checks in a specific order:
   - Is this a "no results" page? If yes, mark for **Deletion** and stop.
   - Is the title or meta description bad? If yes, mark for **T&M** and stop.
   - Are there fewer than 13 products? If yes, stop analysis.
   - Is the SEO content block too short? If yes, mark for **Content**.
8.

9. **Update the Sheet (Part 2):** It writes the final recommendation ("Deletion", "T&M", or "Content") into the sheet.
10. **Mark as Done:** It updates the `Processing Status` to "Completed".
11. **Repeat:** It moves to the next keyword and starts the entire process over again.

Why Delays and a Browser Profile are Important

Google is very good at detecting bots. If a script sends requests too quickly, it will be blocked. The randomized delays in the code make the script's behavior appear more human. The master browser profile saves your login cookies, making the script look like a regular user who is already logged in, which further reduces the chance of being blocked or seeing CAPTCHAs.

Appendix: Troubleshooting & Frequently Asked Questions (FAQ)

Q: I ran the script, but it crashed immediately. What should I do?
A: This is usually a setup issue. Check the following:

- **Paths:** Make sure the `SHEET_NAME` and `WORKSHEET_NAME` exactly match your Google Sheet.
- **GCP Credentials:** Ensure the `gcp_credentials.json` file is in your project folder and is named correctly.
- **Python Environment:** Make sure your `venv` is active. You should see `(venv)` in your terminal prompt. If not, run the activation command again.

Q: I'm getting a
A: This means the script can't find your sheet.

- **Check the Name:** Double-check that the `SHEET_NAME` in `config.py` is spelled exactly right.
- **Check Sharing:** The most common cause is forgetting to share the Google Sheet with the `client_email` from your `gcp_credentials.json` file. Follow Step 4 again.

Q: The script always reports "Not Found" for my rankings, but I know I'm ranking.
A:

- **Check to** make sure the domain in your sheet is correct (e.g., `https://www.myntra.com/`).
- **Increase Page Search:** Your rank might be on page 2 or 3. Try increasing the `MAX_PAGES_TO_SCRAPE` value in `google_rank_finder.py` to `3` or `4`.

Q: I'm getting a lot of CAPTCHAs or Google is blocking me.
A: Your browser profile might be "stale" or logged out.

- Run the `refresh_profile.py` script to delete the old profile and create a fresh, logged-in one. Follow the on-screen instructions to log in to Google again in the browser that opens.

Q: How do I stop the automation completely?
A: Simply press **Ctrl + C** in the terminal window where the script is running.

Q: Can I change the follow-up days from 3 and 7 days to something else?
A: For this SEO tool, you can change analysis rules like product count or word count by editing the `page_optimizer.py` file as described in Chapter 4.