PYTHON · LANGGRAPH · SERPAPI · PANDAS

Agentic SEO Ranking System

Table of contents

3 Project Overview

4 Problem Statement

5 Solution Approach

6 Agentic Architecture (LangGraph)

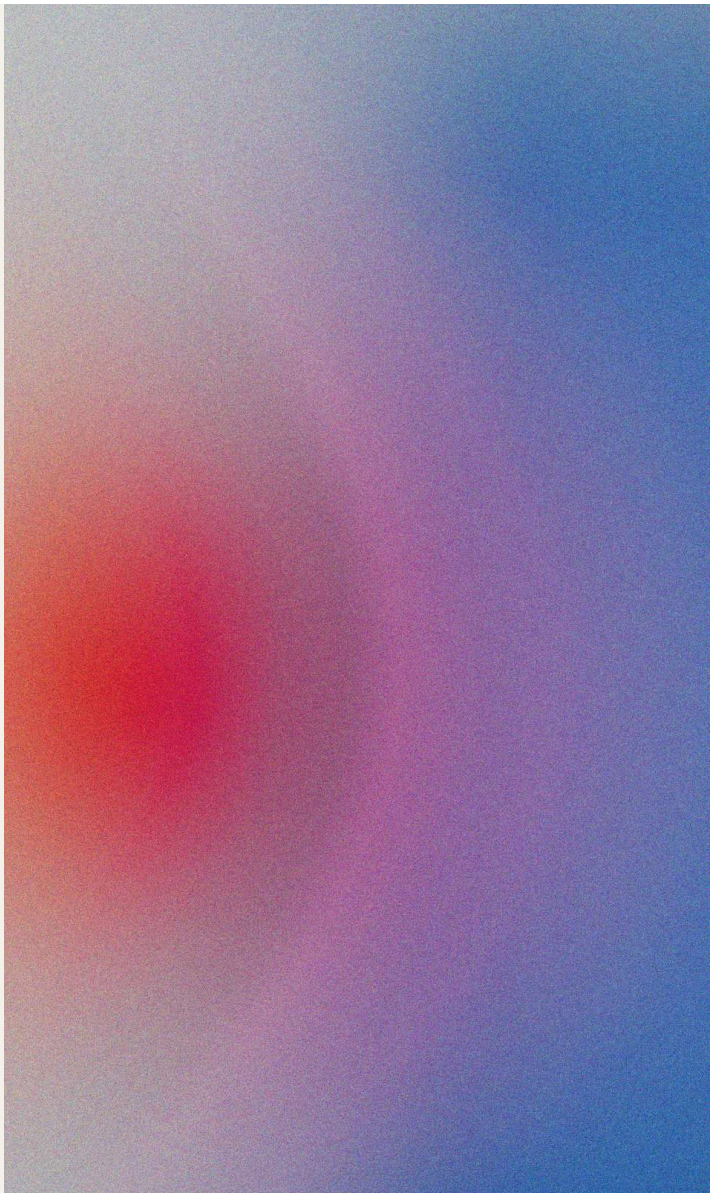
7 Google Places Ranking Logic

8 Google Search Ranking Logic

9 Results & Key Learnings

PROJECT OVERVIEW

- The Agentic SEO Rank Analyzer is an automated system designed to determine:
- Google Places (Local) ranking
- Google Organic Search ranking
- for a website across multiple keywords using an agent-based execution flow.
- The system dynamically adapts to Google’s ranking behavior instead of relying on fixed scraping logic.



The Unspoken Failure of SEO Scripts

- Google's ranking behavior is not static; it's based on user intent.
- Brand keywords ('Starbucks near me') are treated differently than category keywords ('coffee shop').
- A single, rigid API call (e.g., to Google Maps) will inevitably fail, often silently.
- Result: Bad data, wasted resources, and a constant maintenance cycle.

What if a system could think, adapt, and re-route?

Instead of a script that breaks, imagine a system that:

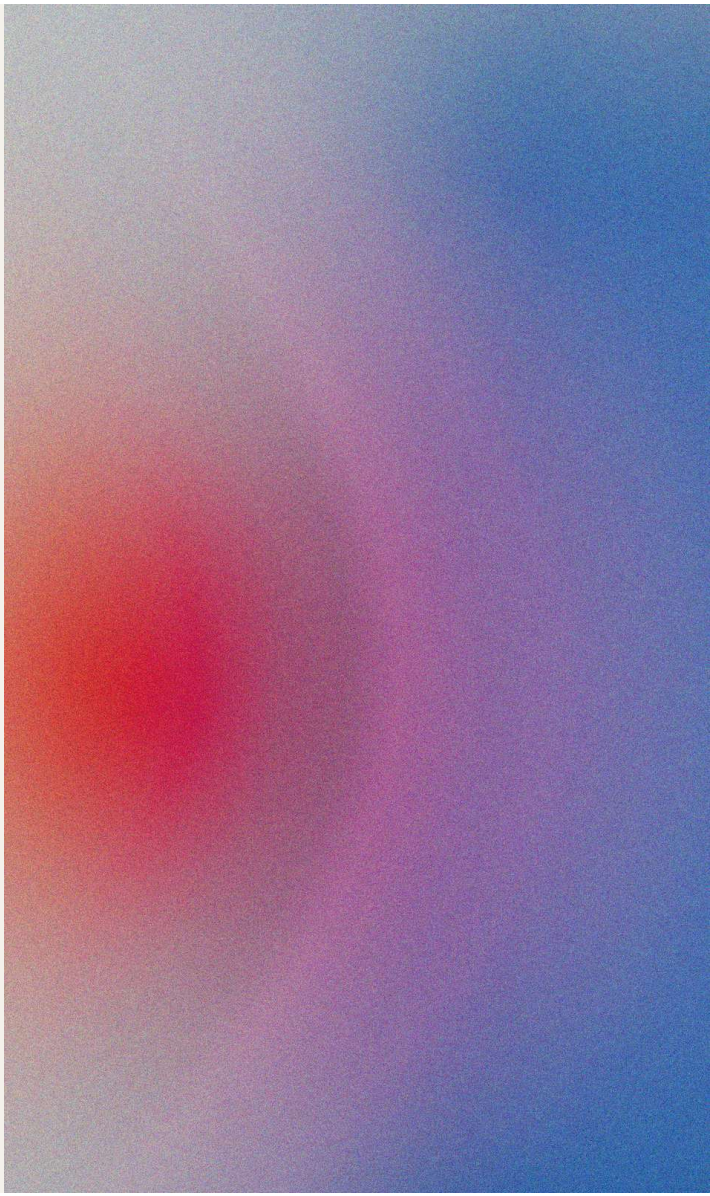
1. Understands its goal (Find the rank).
2. Evaluates the terrain (The API response).
3. Chooses the best path forward (Selects the right tool).

PROBLEM STATEMENT

Traditional keyword ranking scripts fail because:

- Google Places does not return results for all keywords
- Brand and category keywords behave differently
- Google Maps and Local Finder are separate systems
- Rigid logic results in incorrect “NA” rankings

Hence, an adaptive and intelligent ranking system is required.

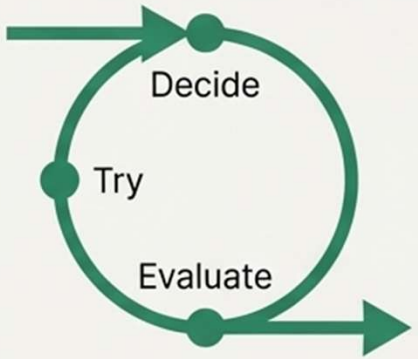


Traditional Script

1. Call Google Maps API
2. Parse Response
3. Return Result

Fails when Maps returns no results.

Agentic Approach



1. Agent decides strategy
2. Tries Google Maps
3. Agent *evaluates* result
4. If needed, Agent *switches* to Local Finder
5. Agent validates business
6. Returns confident result

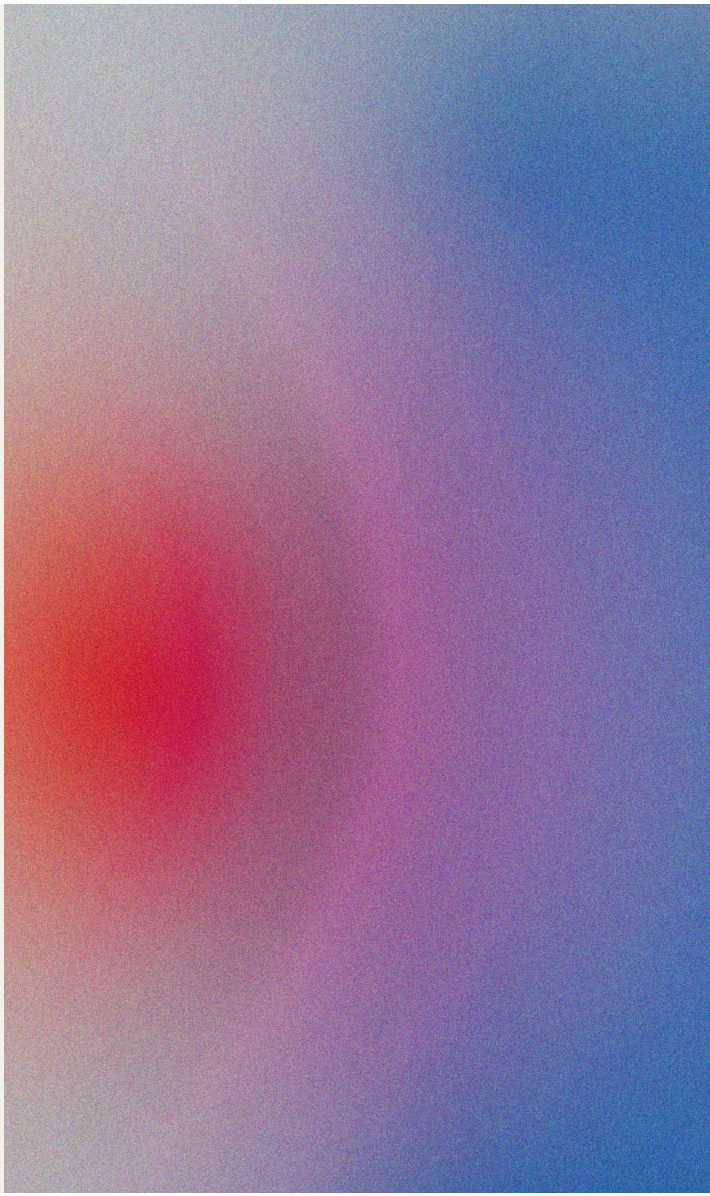
Adapts strategy based on results.

SOLUTION APPROACH

The solution uses a hybrid ranking strategy:

- Google Maps → for strong local discovery keywords
- Google Local Finder → for weak or mixed local intent
- Google Search → for organic website ranking

This ensures accurate ranking detection for all keyword types.



Why Brittle Logic Leads to False Negatives

The Traditional Script Workflow

Fails when Google Maps returns no results for certain keyword types.

AGENTIC ARCHITECTURE

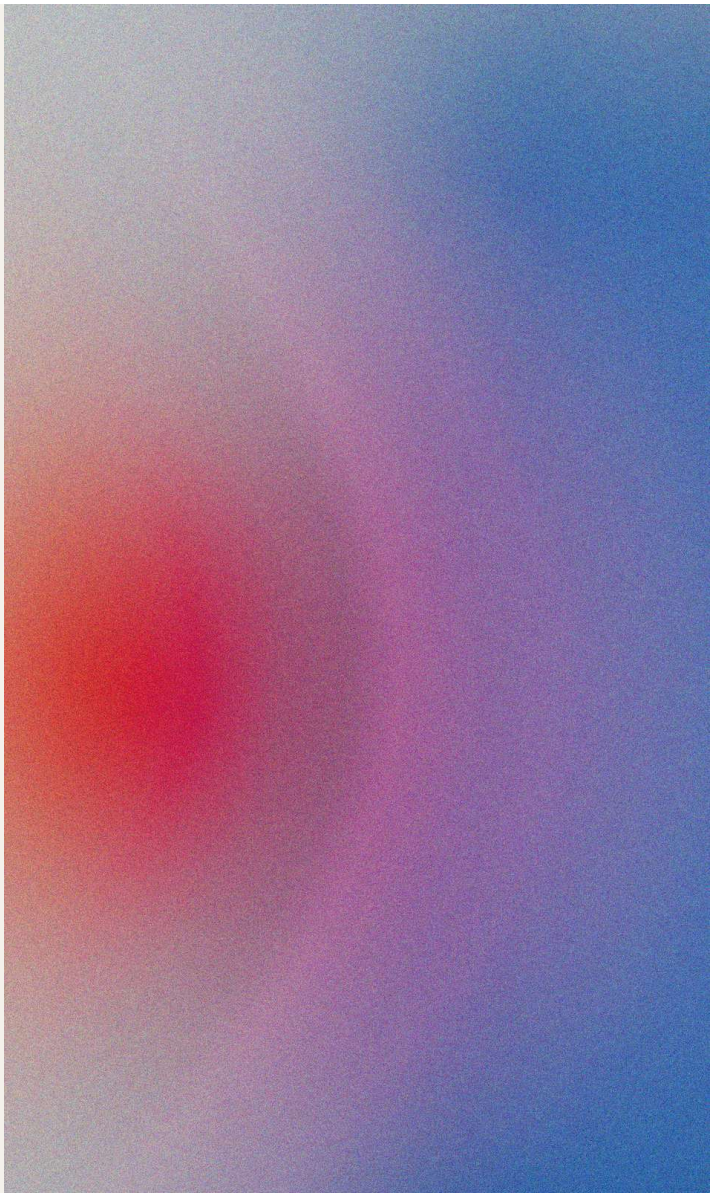
The system is implemented using LangGraph with modular agents:

- Places Rank Agent
- Search Rank Agent

Each agent:

- Performs a single responsibility
- Shares state through LangGraph
- Executes in a controlled, deterministic workflow

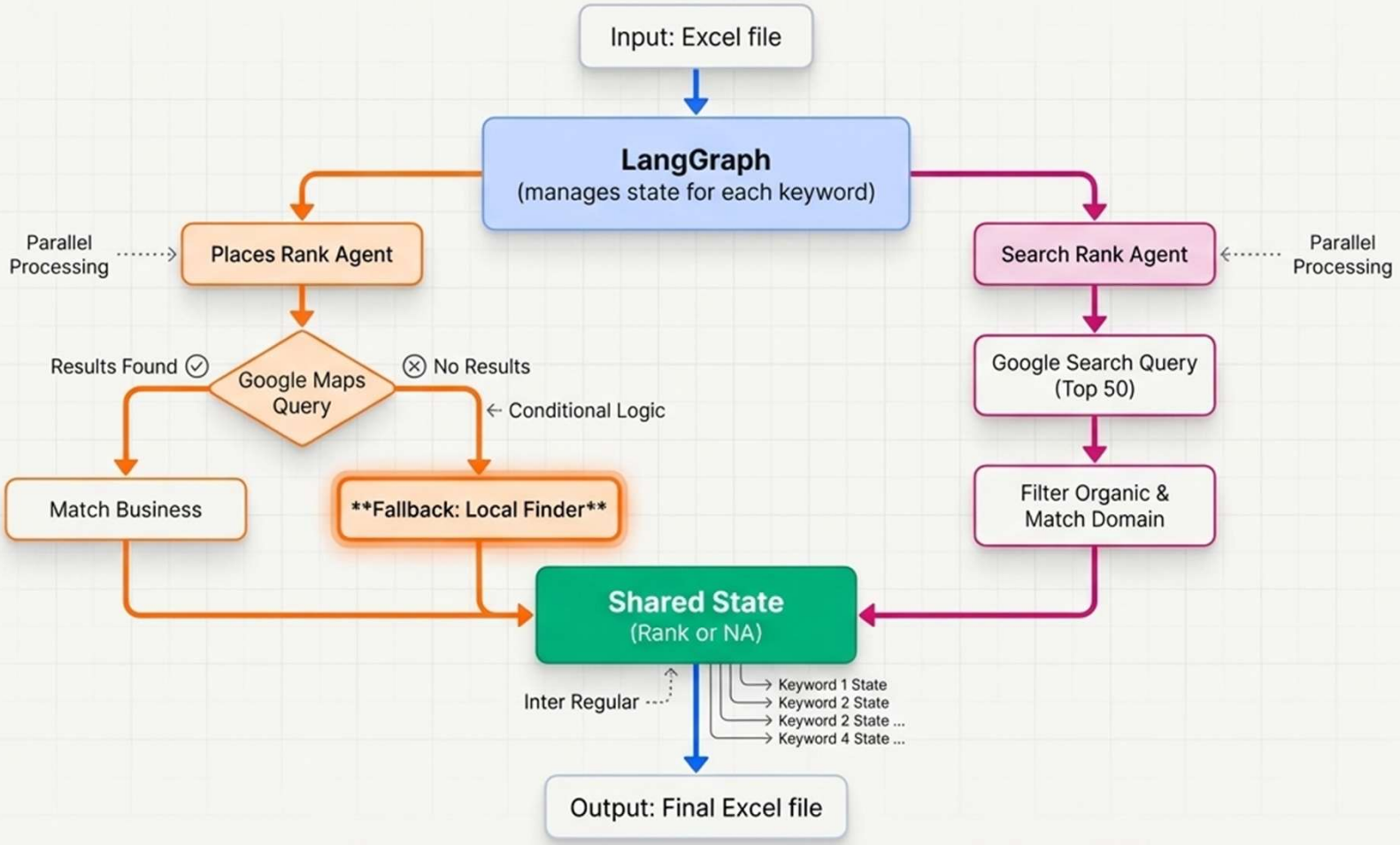
This design improves scalability and maintainability.



The Agentic Approach: Adapt, Don’t Fail

The Adaptive Workflow

Orchestrating Intelligence with LangGraph

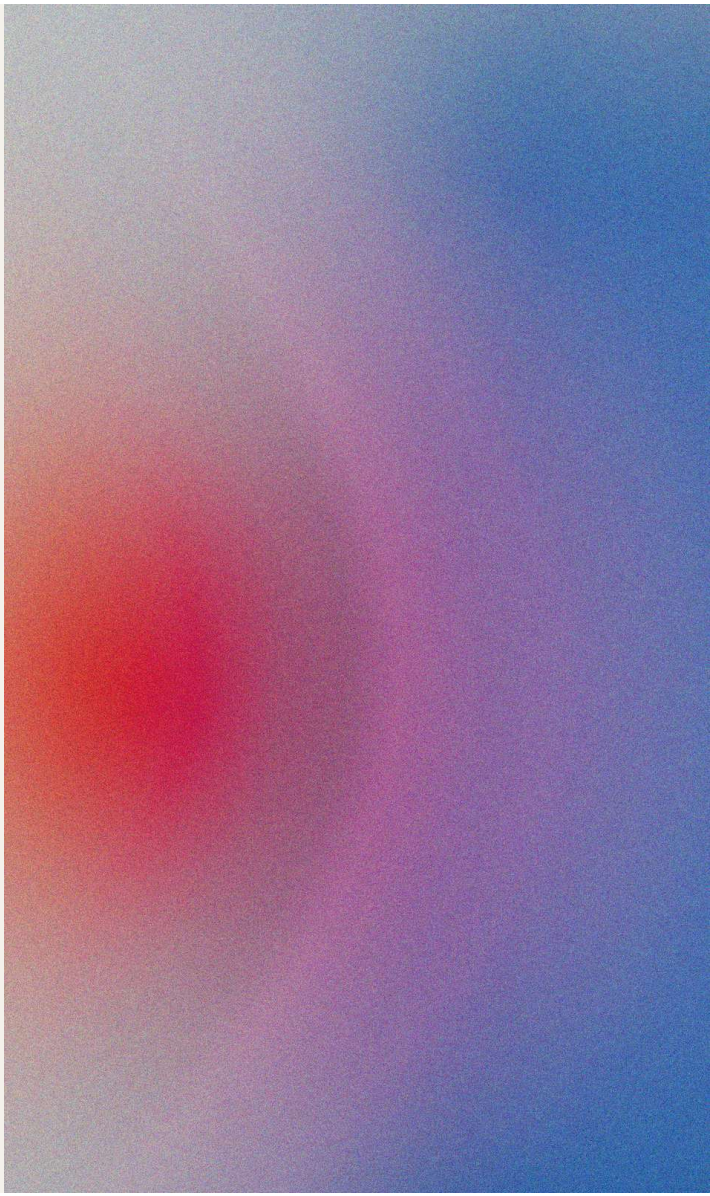


GOOGLE PLACES RANKING LOGIC

Hybrid Google Places Logic:

1. Query Google Maps
2. If no results → fallback to Local Finder
3. Match business using:
    - Business name
    - Website domain

This approach reflects real Google Places behavior and avoids false NA results.



Two Specialists, Two Missions

Places Rank Agent

Mission: Determine the Google Places (Local) ranking.

Superpower: The ‘Hybrid Fallback’—intelligently switching between the Google Maps API and the Local Finder API to avoid false negatives.

Search Rank Agent

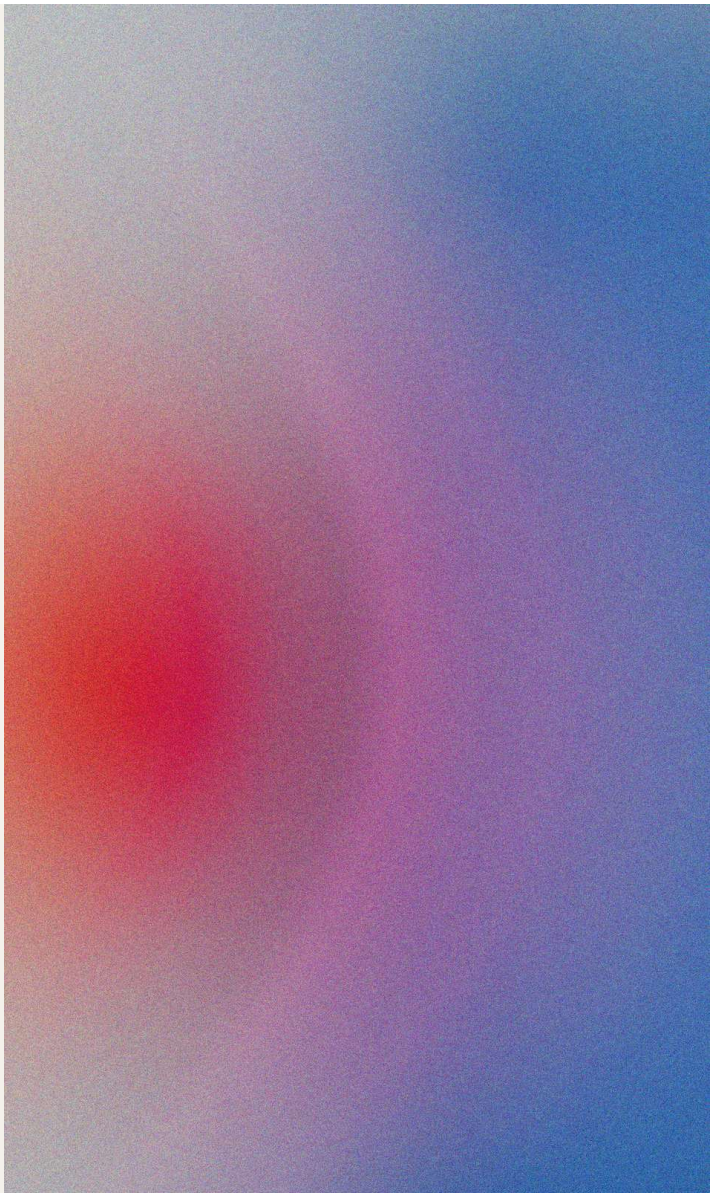
Mission: Determine the Google Organic Search ranking.

Superpower: Precision—sifting through the top 50 organic results, ignoring ads, to find the exact domain position.

GOOGLE SEARCH RANKING LOGIC

- Fetches up to top 50 organic results
- Ignores ads and sponsored links
- Matches website domain
- Returns exact rank or NA

This logic is independent and does not interfere with Google Places ranking.

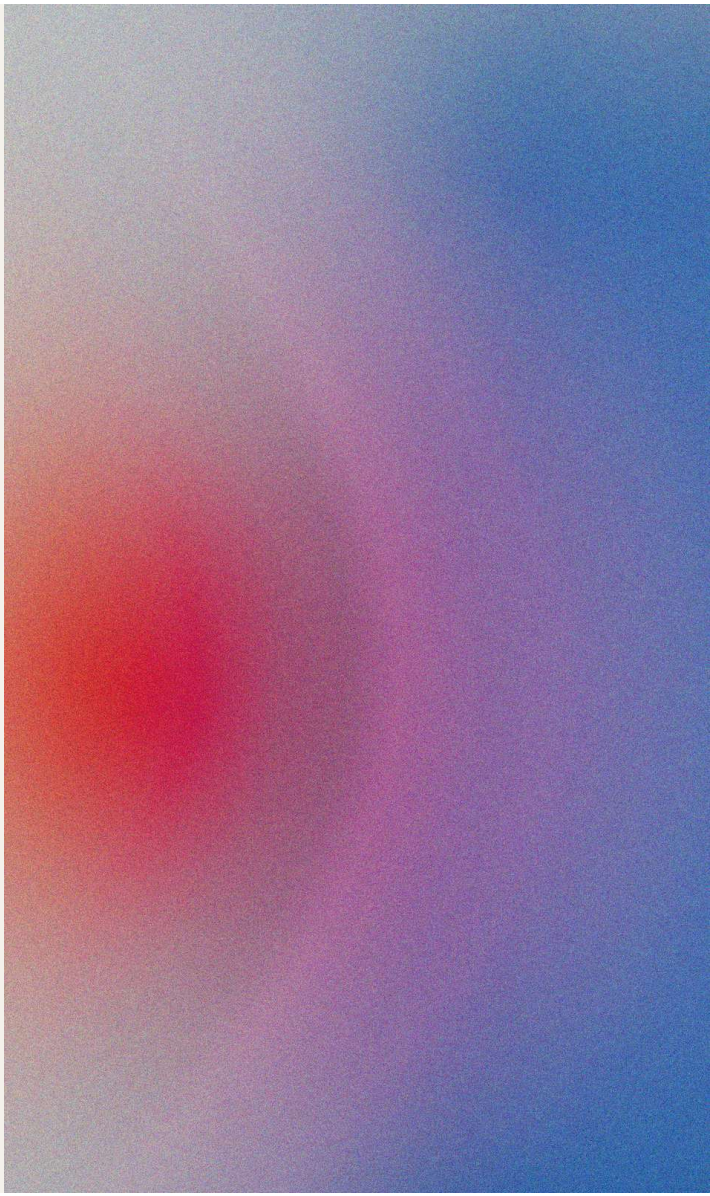


RESULTS & KEY LEARNINGS

This project demonstrates how Agentic AI systems can solve real-world SEO challenges by:

- *Making intelligent decisions*

- Brand keywords rank higher in Places
- Category keywords show varied rankings
- Not all keywords trigger Google Maps results
- Hybrid fallback logic ensures accuracy
- Key Learning:
- Google ranking is intent-based, not static.



Ready to Run

1

Input

Ranking_Website.xlsx

sn	Keyword	Website
1	coffee shop	example.com
2	best pizza	pizzeria.net

2

Execute

```
python agentic_ranking.py
```

3

Output

Agentic_Ranking_Website_Results.xlsx

sn	Keyword	Website	google rank	google rank-1
1	coffee shop	example.com	3	5
2	best pizza	pizzeria.net	1	2

Thank you

BY - HARSHIT WALDIA