Report

This document presents recommended approaches to fulfill the tasks for a prompt-driven microsite generation engine, specifically targeting the "Promote" use case. Each method is aligned with its respective deliverable, followed by observed outcomes and impact based on implementation results. The solution incorporates trend-aware context retrieval using a web search API with Retrieval-Augmented Generation (RAG) and leverages a lightweight language model for generating concise, skimmable microsites, optimized for cost and designed without bio-link dependencies.

The implemented tasks are:

1. Build prompt templates per intent (Promote, Sell, Educate).
2. Implement a layout logic engine to assemble microsite structure based on intent.
3. Set up API orchestration logic for dynamic calls to GPT-4o Mini and Serper.
4. Build caching middleware for prompt responses.

Approaches and Results

Build Prompt Templates per Intent

Approach:

- Design a modular function to craft structured prompts for Promote, Sell, and Educate intents, specifying a 5-element structure (e.g., Informational, Benefits, Trend, CTA, Testimonial for Promote) with JSON output, keeping each element under 150 words.
- Embed dynamic layout instructions in the prompt to guide GPT-4o Mini, ensuring consistent element ordering.
- Incorporate Serper API search snippets as RAG context for the Trend element to enhance relevance (e.g., trends from Reddit or Quora).
- Test prompts with Promote intent input to validate output quality and alignment with user intent.

Results and Impact:

- **Results**: The notebook generates structured prompts for Promote, Sell, and Educate intents, with Promote fully tested. Prompts include Serper context (e.g., "Customers love personalized, handcrafted cakes") and produce JSON microsites with 5 elements (e.g., Informational: 34 words, Trend: 27 words, CTA: 37 words), all under 150 words.
- **Impact**: The approach ensures high-quality, relevant content tailored to user intent (e.g., promoting a baking studio), with Serper-driven trends enhancing the Trend element’s

appeal. Structured JSON output supports seamless integration into microsite rendering, improving user engagement.

Implement a Layout Logic Engine

Approach:

- Define a rules-based dictionary for element structures per intent (e.g., Promote: Informational, Benefits, Trend, CTA, Testimonial) and a function to retrieve the appropriate layout.
- Include layout instructions in prompts to align GPT-4o Mini output with the intended structure.
- Develop a reordering function to ensure generated elements match the specified layout, adding placeholders for missing elements to maintain consistency.
- Validate outputs to confirm 5-element structures, each under 150 words, with no bio-link.

Results and Impact:

- **Results**: The notebook implements a layout engine that produces Promote microsites with 5 elements in the correct order (e.g., Informational: 34 words, Benefits: 38 words, Trend: 27 words, CTA: 37 words, Testimonial: 31 words), all under 150 words, with no bio-link, viewable in <25 seconds.
- **Impact**: The consistent, skimmable structure enhances user experience on mobile devices, ensuring quick content consumption. The automated reordering and validation reduce errors, supporting scalable microsite deployment.

Set Up API Orchestration Logic

Approach:

- Create a function to call GPT-4o Mini, parse JSON output, and handle errors, integrating Serper RAG context for Promote intent's Trend element.
- Build an orchestration function to combine Serper API calls, GPT-4o Mini generation, parsing, validation, and reordering into a streamlined workflow.
- Implement retry logic (e.g., multiple attempts with delays) to ensure reliable API interactions.
- Test orchestration with Promote intent input to verify element structure and RAG integration.

Results and Impact:

- **Results**: The notebook orchestrates Serper and GPT-4o Mini to generate 5-element Promote microsites (e.g., Informational: 46 words, Trend: 33 words, CTA: 37 words) with Serper-driven trends (e.g., summer baking demand) and Instagram-focused CTAs, viewable in <25 seconds, with no bio-link. Retry logic ensures reliability.

- **Impact**: The integrated workflow delivers relevant, action-oriented content, boosting user engagement (e.g., driving Instagram orders). Reliable orchestration minimizes downtime, supporting consistent microsite delivery.

Build Caching Middleware

Approach:

- Implement caching for GPT-4o Mini outputs using a hashing mechanism (e.g., SHA-256) to store JSON responses in a cache directory, enabling reuse of identical prompt configurations.
- Include retry logic in the caching workflow to handle API failures, ensuring uninterrupted content generation.
- Monitor cost savings by comparing uncached and cached runs, targeting <$0.002 per cached run.

Results and Impact:

- **Results**: The notebook caches GPT-4o Mini outputs, reducing costs from $0.00954 (uncached, 588 tokens) to $0.001–$0.002 per cached run. Outputs maintain 5-element microsites (e.g., Informational: 46 words, Benefits: 40 words), with retry logic ensuring reliability.
- **Impact**: Cost optimization aligns with company's budget, enabling scalable microsite generation. Caching improves response times, enhancing user experience, while retries ensure robust performance under varying API conditions.

Conclusion

The approaches modular prompt design with RAG, a rules-based layout engine, streamlined API orchestration with retries, and efficient caching. Results include 5-element microsites with concise, relevant content (e.g., Serper-driven trends, Instagram CTAs), viewable in <25 seconds, with no bio-link, and costs optimized to $0.001–$0.002. The impact is a scalable, user-focused solution that enhances engagement and operational efficiency, aligning with the social marketing AI product goals.