CSC301 AI Assignment Report Template

Team Name	WeWantTwo	Sub-Team Name	n8n Research Agent
Sub-Team Members	Kevin Umaiyalan, Akshat Oza, Abuzar Ansari		
Video Presentation Link	https://youtu.be/l7fqEGLT_O8		
PR Link	https://github.com/csc301-2025-y/6-SEOSearchEngine/pull/9		
README Link	https://github.com/csc301-2025-y/6-SEOSearchEngine/blob/main/README.md (added an n8n workflow explanation section)		

Part 1 – Planning:

1. Identifying the problem/improvement

When users request blog posts, our original workflow relied on a single overloaded AI agent (Mistral 8x7b via OpenRouter), resulting in vague research and poor content quality due to weak prompts and inefficient SerpAPI usage. By introducing multiple specialized AI agents—Planner, Researcher, Editor, and Title Generator—we improved task focus, automated research with better prompts, and structured output using AI-generated Tables of Contents. This modular design enhances SEO, reduces irrelevant content, and simulates a human editorial workflow, improving quality and efficiency without human intervention.

2. Criteria, Research, Selection

Criteria for our AI solution:

- Must enhance our researching capabilities
- Integrate smoothly with n8n – has to do mostly with a no-code environment
- Customizable via prompt engineering

Potential AI Models/Frameworks:
- OpenAI GPT Models
- SerpAPI

Alternatives identified/Reasons not used:
- Tavily: need to create another workflow because it is not available in n8n
- Google Ads API: complex and mostly gives analytics – would decrease speed heavily
- Claude: excellent for research but is not as flexible to use as OpenAI on n8n

Limiting ourselves to OpenAI and SerpAPI ensures that our web application does not have the client waiting more than 5-7 minutes for their content to be generated. Introducing unnecessary complexity to the workflow compromises both content quality and time management—two key priorities of our project.

Part 2 – Implementation:

Plan on using OpenAI gpt-4o mini as it is the most cost-effective without sacrificing quality of work done, especially with researching. Serp integration is simply done on the n8n connecting API, free for 100 searches/month.

Development steps:

1. Identify value model, one that is cost-effective with high-quality output
2. Integrate a workflow that works around chosen model – prompt engineering is the most important aspect
3. Test out workflow with dog park prompt (user is a pet store owner for example)
4. Evaluate tests and confirm with partner for evaluation (second perspective)

Obstacles:

1. Usage of SerpAPI has to be conservative as we have limited tokens for free for the time being (since we're only testing how well it works before we present our MVP) – addressed this with our partner and we will look into getting the paid version
2. Workflow took around 8 minutes to execute initially because multiple AI agents were doing redundant work — addressed this by modifying prompts and cross-checking between teammates to verify that redundancy has been eliminated. Workflow now takes approximately 2-3 minutes to execute.

Part 3 – Impact Analysis:

Our metric is based on how the blog posts are generated, keeping in mind latency and consumption of tools like OpenAI and Serp. We have also evaluated our improvements with our partner and asked for their input.

Blog post structure, tone and engaging manner were all holistically tested by having frontend/full-stack team members and our partner evaluate the before and after results of using the same dog parks in Toronto prompt. Positive feedback was given with minor adjustments to some of the prompts for AI agents when specifically talking to our partner.

Trade-offs:

1. Increased time: our workflow initially took around 30-40 seconds to execute but keeping in mind that the output was very generic and not SEO-optimized, we can sacrifice some time to generate high-quality output. It takes about 2-3 minutes now with very high-quality content generated.
2. Consumption of API resources: Tokens for OpenAI API used for each execution can vary from 1000-7000 which is about $0.002 to $0.007 (partner provided API) used plus another 7-15 searches using SerpAPI which does not cost anything as of now but when potentially commercialized, we have to look into making this a cheaper cost as we will have to adopt the paid version of Serp. Again, the quality of output is far better than it was with our original workflow so we are satisfied with the results as of now.