Statement of Work (SOW): StarshipChatbot Phase 1 Agentic Web Navigation and Semantic Pathing

Prepared For: StarshipChatbot Project Team
Date: September 14, 2025
Project Name: Agentic Web Navigator

1. Project Overview

The StarshipChatbot project aims to develop advanced AI-driven chatbot capabilities. Phase 1 focuses on creating the foundational data layer: an intelligent agent capable of autonomously exploring a target website, understanding the navigational structure, and generating human-readable "semantic paths" that describe the interaction required to reach specific URLs.

2. Problem Statement

Traditional web crawling methods identify *what* URLs exist, but they fail to capture the *context* of how a user navigates to them. For a chatbot to effectively guide a user or understand website structure, it needs to know the sequence of interactions (clicks) required to reach a destination.

For example, knowing that https://www.cinemark.com/movies/now-playing exists is insufficient. The system must understand that this URL is reached by navigating to "Cinemark," clicking "Movies," and then clicking "Now Playing."

The challenge is developing a system that can reliably navigate complex, dynamic websites and intelligently determine the appropriate label for each navigation step, even when links are ambiguous (e.g., icons, images, or generic text like "Learn More").

3. Objectives and Scope

The primary objective of Phase 1 is to develop and deploy the Agentic Web Navigator.

In Scope:

• Development of a headless browser automation system to navigate target websites.
• Implementation of robust crawling logic to handle dynamic content and prevent navigation errors.
• Integration of a Large Language Model (LLM) via the Groq API to analyze HTML context for semantic labeling.
• Development of a standardized JSON output format mapping URLs to their corresponding semantic paths.
• The agent will handle standard navigation links (primarily <a> tags).

Out of Scope:

• Form submissions (e.g., logging in, entering search queries).
• Extraction of page content (focused only on navigation structure).
• Handling complex anti-bot measures or CAPTCHAs.
• Navigation outside of the starting domain.

4. Technical Approach

We will employ a strategy of **Iterative Crawling with Isolated Navigation** powered by a hierarchical AI analysis system.

4.1. Robust Navigation (The Crawler)

The crawler will utilize an iterative approach rather than recursion to maintain stability. For every URL analyzed, a new, isolated browser context (tab) will be opened using Playwright. This prevents state pollution, JavaScript errors (like stale element references), and ensures that each page is analyzed in a clean environment. The crawler will manage a queue of URLs and track visited links to ensure comprehensive coverage within a configurable maximum depth.

4.2. The Agentic Labeling System (The Brain)

When the crawler identifies a navigational link, it will determine the semantic label using a hierarchical approach:

1. **Heuristic Analysis (Primary):** The system will first check high-confidence indicators such as Inner Text, ARIA labels, Title attributes, and Alt text. If a clear, concise, and non-generic label is found, it is used immediately.
2. **LLM Interpretation (Secondary):** If the heuristics are ambiguous or fail, the system will extract the HTML context surrounding the element and send it to an LLM hosted on the Groq infrastructure.

The Groq Advantage: Groq's LPU (Language Processing Unit) infrastructure provides extremely low latency inference. This allows the agent to make potentially thousands of contextual decisions quickly, which is essential for efficient web crawling. The LLM will be prompted to analyze the HTML and the destination HREF to determine a concise (1-3 words) descriptive name for the link.

5. Technology Stack

Layer	Technology	Purpose
Intelligence (AI/ML)	Groq API	Ultra-fast inference for contextual analysis of web elements.
LLM Model	Llama 3 (or similar)	The foundational model used for understanding HTML and generati
Orchestration/Core Logic	Python 3.10+	Core programming language for managing the crawler, API commun

Web Automation	Playwright (Python)	Controlling headless browsers (e.g., Chromium) for reliable, modern
Data Handling	Python Libraries	asyncio, json, urllib.parse.
Environment	Docker (Recommended)	Containerization for consistent deployment and execution.

6. Deliverables

Upon completion of Phase 1, the following deliverables will be provided:
1. **Agentic Web Navigator Application:** Fully functional Python source code.
2. **Dependencies List:** A requirements.txt file.
3. **Documentation:** Setup instructions, usage guide, and technical architecture overview.
4. **Example Output:** A sample JSON file demonstrating the output format.

7. Estimated Timeline and Milestones

The estimated timeline for Phase 1 is approximately 2-3 weeks.

Milestone	Description	Duration
M1: Setup and Basic Crawler	Project setup, Playwright initialization, basic navigation logic, isolated context manageme	Week 1
M2: Heuristic Labeling	Implementation of heuristic extraction (Inner Text, ARIA, etc.) and URL normalization.	Week 1
M3: Groq Integration	Integration of the Groq SDK, LLM prompt strategy development, API error handling.	Week 2
M4: Hybrid Agent Logic	Combining heuristics and LLM interpretation; implementing the decision hierarchy.	Week 2
M5: Testing and Delivery	Testing against complex websites, refining prompts, optimization, documentation.	Week 3

8. Assumptions and Dependencies

• **Groq API Access:** The project team will provide valid API keys and access to the Groq infrastructure with sufficient rate limits.
• **Target Websites:** The target websites are publicly accessible. Navigation requiring authentication or CAPTCHA solving is out of scope.
• **LLM Availability:** The chosen LLM (e.g., Llama 3) remains available on the Groq platform.