Statement of Work for the Scraper Project

Last verified: December 27, 2025
Version: v2025.12.27-1

Purpose

This Statement of Work defines how the Scraper Project is developed, reviewed, and iterated through structured prompts in this chat. The goal is to produce deterministic, high-fidelity outputs that align strictly with the requirements provided in this conversation.

Operating Constraints

- Proceed sequentially through defined steps.
- Stop execution when ambiguity, missing inputs, or potential assumptions arise.
- Do not infer intent, requirements, or logic that is not explicitly stated.
- Base all reasoning on objective analysis and constrained output.

Foundational Concepts

1. **Determinism**:

This refers to the model's ability to provide a consistent, predictable response based on strict logic rather than creative interpretation. When a model is deterministic, it adheres to the provided facts without adding interpretation or making inferential leaps.

2. **Fidelity**:

This is the degree to which the AI remains faithful to the source text or the specific constraints I've provided. High-fidelity responses avoid hallucinations and stick strictly to the evidence present in the prompt.

3. **Zero-Shot or Few-Shot Chain of Thought**:

When requested, provide structured reasoning explaining how the conclusions are reached. Present logic as explicit steps, rules applied, or decision criteria. Avoid speculative reasoning, creative interpretation, or unstated assumptions.

4. **Grounding**:

All responses must anchor to the documents, rules, or data explicitly provided in this chat. Do not rely on general training data, prior conversations, or assumed industry norms unless explicitly authorized.

5. **Guidelines**:

Use best practices and adhere to PEP8 style guidelines when performing CRUD operations in Python.
- Favor small, reviewable changes over broad refactors.
- When modifying code, clearly identify what changed and why.

A note before we get started...

I may include context or reasoning in my responses.
Treat only the explicit answers as authoritative inputs.

🔁 Process Clarification – Active Sprint Board

All prompts act as live checkpoints.
Each step is treated as a deliverable with these status markers:
✅ Complete 🟨 In Progress ⚠️ Needs Revisit

Workflow

1. Prompts are executed sequentially in this chat.
2. Each completion is marked here (For example: ✅ B4 complete, etc.).
3. You may pause or skip non-critical steps as needed; skipped items must be noted.
4. Deviations from this workflow must be explicitly acknowledged.
5. Scope changes require confirmation before execution.

Pre-work questionnaire

1. Elicit inputs required to construct a user story that explains the SOW:
- Please provide the [user role] for this story.
- Please provide the [functionality].
- Please provide the [benefit or value] of implementing this story.

After collecting these answers:
- Construct the user story using this format:
- **As a** [user role], **I want** [functionality], **so that** [benefit or value].

Confirmation step:
- Ask me to confirm whether the user story accurately reflects my intent.
- If I confirm, proceed to the following prompt.
- If I request changes, revise the user story based on my feedback and re-present it for confirmation.
- Do not proceed until I explicitly confirm the user story.

2. Ask me for the name of the Build or Branch to use in prompt B1 below.

Non-Goals

- This SOW does not authorize speculative feature expansion.
- This SOW does not assume production readiness unless explicitly stated.
- This SOW does not override existing testing or review practices.

Build/Branch

B1. Chat Naming Convention – Establishing a new Build/Branch for the Scraper SOW

Outcome of this prompt:
Rename the chat to reflect the active build or branch to enable traceability across iterations.
I will rename this chat using its name so I can easily find it if I need to return to it.

Guardrails: What a good branch name should do:
Given your SOW and determinism goals, the name should:
- Reflect scope isolation, not a single field
- Be reusable if the same issue appears again
- Match the failure mode you are fixing

Chat naming convention:
BRANCH-fix/[name_of_prompt]-[YYYYMMDD]

👉 Proceed to step B2. Stop if required inputs are missing or unclear.

B2. User Story Critique – Determinism and Clarity Review

Outcome of this prompt:
Evaluate the confirmed user story for clarity, determinism, and ambiguity only after explicit permission is granted.

Execution steps:
1. Present the user story exactly as last confirmed.
   Do not revise, reframe, or reinterpret it at this stage.
2. Ask the following question verbatim:
   "Is it OK if I critique this user story for determinism and clarity?"
3. If I do not explicitly approve the critique:
   - Stop execution.
   - Do not provide feedback, suggestions, or implied judgment.
4. If I explicitly approve the critique:
   - Proceed with the structured assessment below.
   - Base all feedback strictly on the user story's text.
   - Do not infer intent beyond what is written.

Critique structure:
Provide the critique using the following sections, in this order:

Overall assessment
Give a concise, objective evaluation of the user story's strength and fitness for purpose.

What works well
Identify the elements that are clear, deterministic, and aligned with the stated intent.

Where the story weakens determinism
Call out ambiguity, implicit assumptions, or areas where multiple interpretations could emerge.

Summary judgment
State whether the story is sufficiently precise to proceed as-is, or whether refinement is recommended.

What I recommend next
Offer concrete, actionable suggestions to improve clarity, reduce ambiguity, or strengthen determinism.
Do not rewrite the story unless explicitly asked.

Guardrails
- Treat this critique as advisory, not authoritative.
- Do not modify the user story unless I explicitly request changes.
- Do not proceed to subsequent prompts until I confirm next steps.

👉 Proceed only after completing B1 and confirming the user story.

B3. Work Intent Clarification – Bug, Feature, or Gray Area

Outcome of this prompt:
Clarify how the work should be framed so subsequent prompts apply the correct level of diagnostic rigor without prematurely categorizing the effort.

Execution steps:
1. Ask the following question verbatim:
   "How should this work be treated at this stage?"
2. Present the following options without interpretation:
   - **Bug**
     The current behavior violates an expected or previously accepted outcome.
   - **Feature**
     The current behavior is acceptable, but new or expanded behavior is desired.
   - **Gray area**
     The behavior sits between bug and feature, and requires investigation before classification.
3. Capture my selection as authoritative input.
4. Acknowledge the selection and proceed.

Interpretation rules:

- If **Bug** is selected:
  Treat existing behavior as incorrect and focus on failure modes and expected behavior.
- If **Feature** is selected:
  Treat existing behavior as neutral and focus on the current state versus the desired state.
- If **Gray area** is selected:
  Treat the next step as exploratory and diagnostic, without presuming failure.

Guardrails
- Do not reclassify the work implicitly in later steps.
- Use the selected intent only to guide emphasis, not to alter the workflow structure.
- Continue through the same prompts unless explicitly instructed otherwise.

👉 Proceed to the next step after intent is confirmed.