

Project-X: Agent Development Report

[Developer Name]

July 21, 2025

Abstract

This report documents the development of the [Agent Name] Agent, a critical component of the Project-X MVP, designed to operate within a LangGraph environment using pre-trained Ollama models. The agent's role is to [briefly describe agent's purpose, e.g., parse user prompts, generate synthetic data]. This document details the agent's design, implementation, testing, and integration, providing a comprehensive guide for future developers to replicate or extend the work. The report follows a scientific format, adhering to British English standards without contractions, as specified for Project-X documentation.

Keywords: Project-X, LangGraph, Ollama, [Agent-specific keywords, e.g., synthetic data, Py-Caret, business rules].

1 Introduction

This report outlines the development process of the [Agent Name] Agent, one of eight agents in the Project-X MVP, aimed at delivering a system for automated data processing, machine learning model creation, and business rule application within a 6-week timeline starting July 21, 2025. All documentation must be written in British English, avoiding contractions (e.g., use "is not" instead of "isn't") to ensure standardisation across the project.

The [Agent Name] Agent is responsible for [describe agent's role, e.g., parsing user prompts into structured inputs]. The author's additional role in the team is [describe team role, e.g., Frontend Developer, Tech Lead], contributing to [briefly describe additional contributions, e.g., UI design, system integration]. This report provides a rigorous description of the agent's functionality, implementation, and testing, ensuring that another developer can fully understand and replicate the work.

Expected Content: Provide a brief overview of the agent's purpose within the Project-X pipeline, its significance, and how it aligns with the project's goals (e.g., fraud detection, supply chain optimisation). Mention the author's team role and contributions beyond agent development. Include references to relevant literature or tools (e.g., `pycaret`, `faker`, LangGraph) if applicable.

2 Methodology

This section describes the design methodology for the [Agent Name] Agent, including its role in the LangGraph pipeline, inputs, outputs, and integration strategy.

2.1 Agent Role and Goal

The [Agent Name] Agent is designed to [describe specific goal, e.g., generate synthetic datasets based on user specifications]. Its role in the pipeline is to [explain position, e.g., process inputs from the User Prompt Parser Agent and provide outputs for the EDA Agent].

Expected Content: Detail the agent's primary objective and its position in the sequential pipeline. Explain how it contributes to the overall MVP functionality (e.g., enabling automated model training).

2.2 Inputs

The agent receives the following inputs:

- **Source:** [Specify source, e.g., user input, previous agent (name the agent)].
- **Format:** [Describe format, e.g., JSON, CSV, text prompt].

- **Details:** [Explain how inputs are processed, e.g., parsing JSON for column names, validating data types].

Expected Content: Provide a precise description of input sources, formats, and processing steps. Include examples (e.g., sample JSON or CSV structure) and any validation mechanisms to ensure input usability.

2.3 Outputs

The agent produces the following outputs:

- **Format:** [Describe format, e.g., CSV file, JSON report].
- **Details:** [Explain output structure, e.g., columns in a CSV, fields in a JSON].
- **Usability:** [Describe how outputs are used by the next agent, e.g., compatibility with PyCaret].

Expected Content: Specify the output format, structure, and how it ensures compatibility with the next agent in the pipeline. Include examples of expected output.

2.4 Integration with LangGraph and Ollama

The agent is implemented as a LangGraph node, using a pre-trained Ollama model for [specify NLP tasks, e.g., prompt parsing, result interpretation]. The agent’s logic is executed in Python, leveraging [list libraries, e.g., `pandas`, `faker`] for data processing.

Expected Content: Describe how the agent integrates into the LangGraph workflow (e.g., as a node with defined edges). Detail the use of Ollama (e.g., model selection, tasks performed) and Python libraries. Explain how state is managed between agents.

3 Implementation

This section details the technical implementation of the [Agent Name] Agent, including code structure, Ollama configuration, and prompt engineering.

3.1 Code Structure

The agent is implemented in Python, with the following components:

```

1 def parse_prompt(prompt):
2     # Function to parse user prompt
3     pass
4
5 def generate_data(specification):
6     # Function to generate synthetic data
7     pass

```

Listing 1: Main Functions

Expected Content: Provide a high-level overview of the code structure, including key functions, classes, or scripts. Include pseudocode or code snippets if necessary to illustrate critical logic.

3.2 Ollama Configuration

The agent uses a pre-trained Ollama model configured as follows:

- **Model:** [Specify model, e.g., LLaMA variant].
- **Temperature:** [Specify value, e.g., 0.3 for deterministic output].
- **Top-p:** [Specify value, e.g., 0.9 for nuanced responses].

Expected Content: Detail the Ollama model used, configuration parameters, and rationale (e.g., low temperature for precision). Explain how these settings affect output quality.

3.3 Prompt Engineering

The agent’s prompts are designed as follows:

- **Prompt Structure:** [Describe prompt format, e.g., “Generate Python code for...”].
- **Examples:** [Provide sample prompts and expected outputs].
- **Validation:** [Explain how prompts ensure valid outputs].

Expected Content: Provide the exact prompt structure, including instructions and examples. Describe how prompts are tested and refined to ensure reliable Ollama responses.

4 Results

This section presents the results of testing the [Agent Name] Agent on the five predefined use cases (e.g., fraud detection, customer analytics).

Expected Content: Describe the testing process, including use cases tested, input examples, and output results. Include metrics (e.g., accuracy of generated data, model performance) or qualitative assessments (e.g., report clarity). Provide tables or figures if applicable, e.g.:

Use Case	Result
Fraud Detection	[e.g., Generated 1000-row dataset with 95% column type accuracy]
Customer Analytics	[e.g., JSON report confirmed target suitability]

Table 1: Test Results for [Agent Name] Agent

5 Discussion

This section evaluates the agent’s performance, challenges encountered, and lessons learned.

Expected Content: Discuss the agent’s strengths (e.g., robust prompt parsing), limitations (e.g., handling ambiguous inputs), and challenges during development (e.g., debugging Ollama outputs). Suggest improvements for future iterations, such as enhanced prompt engineering or additional validation.

6 Conclusion

The [Agent Name] Agent successfully [summarise key achievement, e.g., generates well-formed synthetic data]. Its integration into the Project-X pipeline ensures [describe contribution, e.g., reliable inputs for model training]. Future work may include [briefly suggest next steps, e.g., supporting larger datasets].

Expected Content: Summarise the agent’s role, key achievements, and alignment with project goals. Highlight its readiness for integration and potential enhancements.

7 References

References

[1] PyCaret Documentation, <https://pycaret.org/>.

[2] LangGraph Framework, <https://langchain-ai.github.io/langgraph/>.

[3] Ollama Documentation, <https://ollama.ai/>.

Expected Content: List all referenced materials, tools, or libraries used in development (e.g., `pycaret`, `faker`, LangGraph, Ollama). Use BibTeX format for consistency.

8 Team Role

The author, [Developer Name], serves as [team role, e.g., Frontend Developer], contributing to [describe additional contributions, e.g., UI design, system integration]. In the context of Project-X, these contributions supported [explain how role aided the project, e.g., seamless API integration].

Expected Content: Detail your additional role in the team and specific contributions beyond agent development. Explain how these efforts supported the broader project.