

BUSINESS REQUIREMENT DOCUMENT

PRODUCT TITLE: Agentic Playground

PRODUCT COACH: Yajur Thapar

TECH COACH: Boddeda Prudhvi Raj, Aditya Tulsian

TEAM MEMBERS:

Abhay Bhadauria (Team Captain)

Aditya Nautiyal

Saswat Jyoti Das

V.S. Kiran

1. Problem Statement

In the current landscape, managing agents and workflows can be a cumbersome and fragmented process. Existing solutions often lack integration, real-time feedback, and user-friendly interfaces, leading to inefficiencies and increased complexity in workflow management. Users need a cohesive platform that not only simplifies the management of agents but also provides robust tools for creating, testing, and debugging workflows in an interactive and real-time environment.

2. Proposed Solution

The *"Agentic Playground"* application aims to provide users with a comprehensive platform to manage agents, create and test workflows, and monitor real-time terminal output for debugging purposes. The proposed solution leverages modern technologies to ensure a seamless, efficient, and user-friendly experience. The application will be developed using React for the frontend and FastAPI for the backend, with additional tools and technologies like web sockets to support real-time communication and data storage.

3. Project Objectives

- Develop a user-friendly web application for managing agents and workflows.
- Implement backend functionalities to support agent management, workflow creation, and execution.
- Implement an interactive playground where users can test workflows in a simulated environment.
- Allow human interaction in workflows, to get desired output.
- Provide real-time debugging capabilities through a terminal output connected via web socket.

4. Project Scope

Agentic Playground aims to assist the users in creating and testing workflows using multiple LLM agents in nested and sequential manner using an interactive user interface.

Category	In-Scope	Out-of-Scope
GUI Application	- Develop a user-friendly GUI for managing agents and workflows.	- Development of mobile applications.
Backend Services	- Design and develop a robust backend system.	- Integration with external third-party services not specified in the requirements.
Agent Management	- Implement features to add user specified agents - Allow users to assign agents to specific workflows.	
Workflow Management	- Develop functionalities to create workflows. - Enable users to assign tasks and agents to workflows.	-Edit Workflow is not specified in the requirement
Interactive Playground	- Implement an interactive playground for testing workflows. -Allow human interaction in workflows, to get desired output	

Category	In-Scope	Out-of-Scope
Terminal Output	- Include a terminal output connected via WebSocket for real-time debugging information.	

5. Functional Requirements

5.1 Agent Management

- The Agent Management screen is a crucial component of the *Agentic Playground* application, designed to facilitate the creation of agents.
- Users can create new agents by filling out a form with relevant details, such as agent name and its system prompt, and then save these details to the backend.

5.2 Workflow Creation

- The workflow creation is designed to enable users to build their own workflows.

5.3 Playground

- Workflow playground offers users an interactive environment to test the workflows they have created.
- With the input data which is provided, users can initiate the workflow and observe the output in a designated output display section.
- Workflow Playground offers human interaction in order to get desired results.

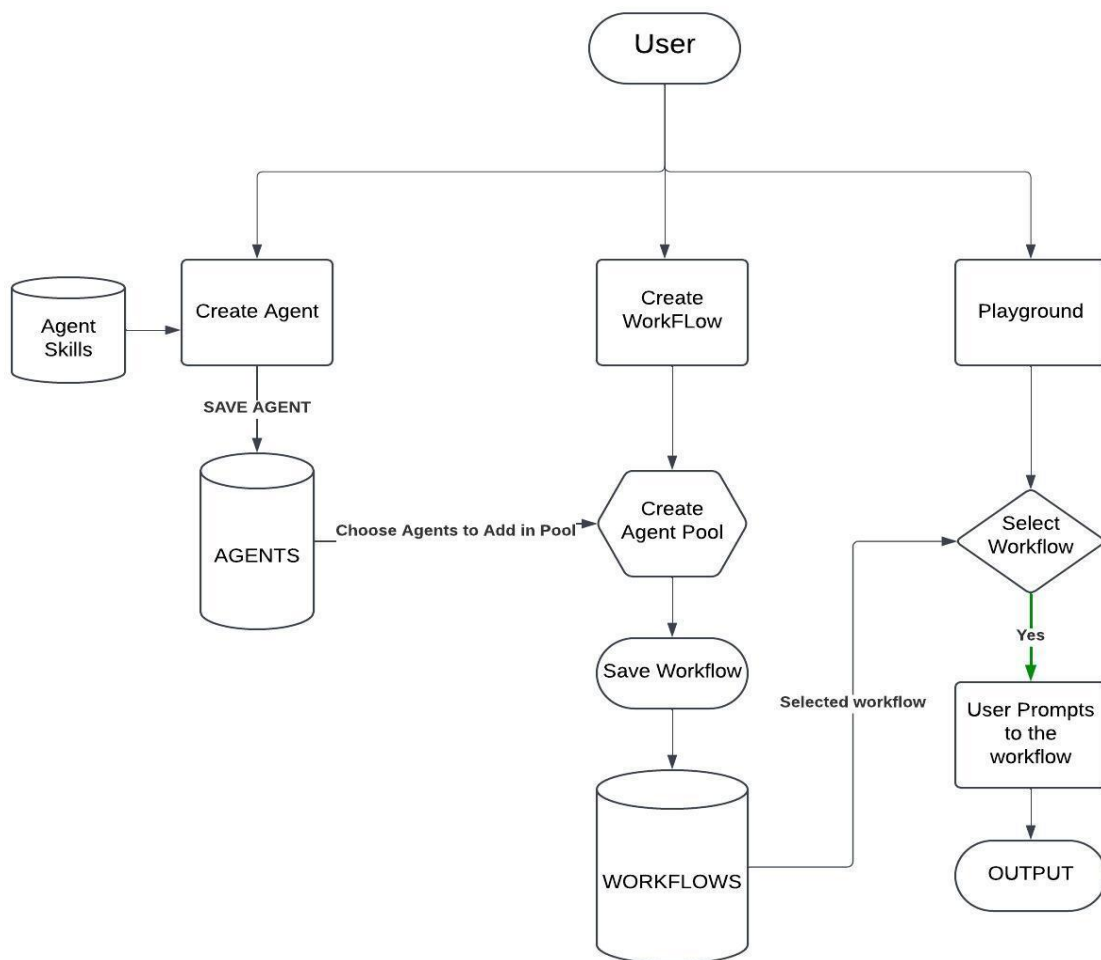
5.4 Terminal Output

- This features a terminal window that streams output directly from the backend processes via WebSocket technology.

6. Non-Functional Requirements

- **Usability:** The application should have an intuitive and user-friendly interface.
- **Performance:** The application should perform efficiently with minimal latency.
- **Scalability:** The system should be scalable to handle an increasing number of agents and workflows.

7. Architecture Diagram



8. Technical Stack

- **Frontend:** React.js for UI display of agents, workflow and playground
- **Backend:** FastAPI for handling backend processes.

9. Key Stakeholders

Development Team

- **Role:** Designs, develops, tests the application.
- **Interest:** Builds a high quality, functional product.

End Users

- **Role:** Uses the application for managing agents and workflows.
- **Interest:** Needs a user-friendly, efficient, and reliable tool.

10. Assumptions

- Users have basic knowledge of managing agents and creating workflows.
- The backend system will be developed using a suitable technology stack that supports web socket connections.

11. Project Constraints

- The project must be completed within the allocated timeline.
- The application should be compatible with major web browsers.
- The application must feature an intuitive and user-friendly interface, enabling users to navigate and perform tasks efficiently.
- The project must manage changes to the scope, requirements, or design effectively to avoid increased costs, and extended timelines.

12. Acceptance Criteria

- Successful creation, editing, and display of agents.
- Functional playground for executing workflows and receiving outputs.
- Real-time terminal output connected via web socket for debugging.