🚀 Challenge: AgentDock — Multi-Agent MCP Server with UI & Tool Integrations

Overview

AgentDock is a Model Context Protocol (MCP) server with a clean UI to register, manage, and interact with intelligent agents. It enables multi-agent orchestration, tool integrations (e.g., GitHub, Slack, Jira), and LLM-powered interactions via an LLM (free version / tier).

The goal is to build an extensible platform where users can trigger actions through natural language, monitor agent behavior, and register tools dynamically — all from a web interface.

Core Requirements

UI Capabilities

- **Agent Management**
  - Register/deregister agents with code, description, and config
- **Natural Language Interface**
  - Ask agents questions using Groq (e.g., “Summarize latest PR”)
- **Monitoring & Logs**
  - View recent agent actions and outputs

Backend Functionality

- **MCP-Compatible Agent Server**
- **Multi-Agent Support** (e.g., GitHubSync, SlackAgent, etc.)
- **REST API Tool Registration**
- **Modular & Extensible Architecture**

Sample Integrations

Speech-to-text commands (e.g., "Update Shopify inventory")

Advanced tools or agent chaining

Seamless configuration UI for tool-specific settings

These are some examples, try to make creative ones.

Tool	Example Use
GitHub	PR summaries, repo sync, CI/CD triggers
Jira	List, update, or create tickets
Slack	Send messages, channel updates
Shopify	Inventory updates via natural language
Speech	Transcribe commands into actions