This Product Requirements Document (PRD) outlines the "AGI Workforce" platform, a system designed to automate complex tasks by coordinating a team of specialized AI agents, referred to as "AI Employees."

---

1. Introduction

1.1 Purpose

This document defines the scope, features, and functionality of the AI Workforce platform. It serves as a guide for development, design, and testing to ensure the final product aligns with the project's vision and goals.

1.2 Product Overview

The AI Workforce is an autonomous system that interprets user requests, breaks them down into a series of tasks, assigns those tasks to the most suitable AI Employees, and orchestrates their collaboration to achieve the user's goal. This platform mimics the structure of a human company, where different specialists work together on a project. The user interacts with the system through a chat interface, describes their desired outcome in natural language, and the AI Workforce handles the rest, providing real-time, visual updates on its progress.

1.3 Goals

- **Automation of Complex Tasks:** To enable users to automate complex, multi-step tasks that traditionally require a team of specialists.
- **Intuitive User Experience:** To provide a simple, chat-based interface that allows users to initiate and monitor tasks using natural language, supplemented by a rich visual interface.
- **Transparent and Interactive Collaboration:** To create a system where specialized AI agents can work together, hand off tasks, and share information to complete a project, with the user able to observe and intervene in the process.
- **Scalability and Extensibility:** To build a platform that can be easily extended with new AI Employees, tools, and capabilities.

---

2. Core Concepts

2.1 AI Workforce

The entire ecosystem of AI agents, orchestrators, and tools. The workforce is responsible for the end-to-end process from receiving a user request to delivering the final result.

2.2 AI Employees

Specialized AI agents designed to perform specific roles, such as "Frontend Engineer," "Backend Engineer," or "QA Engineer". Each employee has a unique system prompt that defines its persona, skills, and communication style, as well as a visual avatar for representation in the UI.

2.3 Orchestrators

The "brains" of the operation, responsible for managing the workflow. There are two primary orchestrators:

- **Workforce Orchestrator:** The main entry point that manages the entire pipeline from user input to task completion.
- **Multi-Agent Orchestrator:** Intelligently coordinates the AI Employees to work together autonomously.

2.4 Execution Plan

A detailed plan of action generated by the `TaskDecomposer`. It breaks down the user's request into a series of tasks with dependencies, estimates the time required, and defines the execution order.

---

3. User Personas

- **Developers:** Software engineers who want to automate coding tasks, debugging, and deployment.
- **Project Managers:** Individuals who need to coordinate and execute complex projects without a large human team.
- **Entrepreneurs/Solopreneurs:** Individuals who need to build and launch products or services with limited resources.

---

4. Functional Requirements

4.1 User Interaction and Task Initiation

- **Chat-Based Interface:** The primary user interaction will be through a chat interface where users can submit requests in natural language.

- **Visual Collaboration Interface:** The chat interface will be enhanced with a visual representation of the AI Workforce. This interface will display the avatars of the AI Employees as they "talk" to each other, showing the flow of communication and task handoffs. This allows the user to understand the plan's execution, and provides an opportunity for the user to interrupt and make changes to the plan if needed.
-
- **Preview Execution:** Users can preview the execution plan, including the tasks, estimated time, and cost, before committing to the full execution.
-
- **Real-time Updates:** The UI will provide real-time updates on the status of the execution, including which tasks are in progress, completed, or have failed, both in the chat and through the visual collaboration interface.
-
- **Execution Controls:** Users will have the ability to pause, resume, and cancel an ongoing execution.
-

4.2 Task Processing Pipeline

1. **Input Validation and Analysis (NLP):** The user's input is first validated and then analyzed by an NLP processor to determine the intent, domain, complexity, and other parameters.
2.
3. **Task Decomposition:** The `TaskDecomposer` breaks down the user's intent into a structured `ExecutionPlan` with a dependency graph of tasks.
4.
5. **Agent Selection:** For each task, the `AgentSelector` evaluates and selects the optimal AI Employee based on their capabilities, cost, performance, and reliability.
6.
7. **Execution:** The `ExecutionCoordinator` starts the execution of the plan, managing the state and flow of tasks.
8.

4.3 Multi-Agent Collaboration

-

- **Autonomous Coordination:** The `MultiAgentOrchestrator` enables AI Employees to work together autonomously, running continuously until the task is complete.
-
-
- **Visualized Communication:** The interactions between agents will be visualized in the UI, showing the "conversation" as they collaborate and hand off tasks.
-
-
- **Task Handoffs:** Agents can hand off tasks to other agents, for example, from a "Backend Engineer" to a "Frontend Engineer".
-
-
- **Execution Strategies:** The orchestrator can determine the best execution strategy, such as sequential, parallel, or hybrid, based on the task dependencies and complexity.
-

4.4 Workforce Management

- **AI Employee Marketplace:** Users can "hire" AI Employees from a marketplace. The `purchased_employees` table in the database schema confirms this feature.
-
-
- **Workforce Dashboard:** A dashboard will provide an overview of the AI workforce, including the number of employees, their status, and performance metrics.
-
-
- **Status Monitoring:** The system will track the status of each AI Employee (e.g., available, busy, offline) and the overall health of the workforce.
-

4.5 Extensibility and Integrations

- **Tool Integration:** The platform will support a variety of tools that AI Employees can use, such as code executors, web search, and browser automation.
-
-
- **Multiple LLM Providers:** The system is designed to work with various LLM providers, including OpenAI, Anthropic, and Google.
-
-
- **Agent Integration:** The system supports integration with various code agents like `claude-code`, `cursor-agent`, `gemini-cli`, and `replit-agent`.
-

---

5. Non-Functional Requirements

- **Performance:** The system should be responsive, with minimal latency in the chat interface and efficient task processing.
- **Scalability:** The architecture should be able to handle a growing number of users, AI Employees, and concurrent tasks. The use of Supabase for the backend suggests a scalable architecture.
- **Reliability:** The system should be resilient to failures. If an AI Employee or a tool fails, the system should be able to retry the task or assign it to a fallback agent.
- **Security:** User data and API keys must be stored and handled securely. The database schema includes tables for user authentication and authorization, as well as row-level security policies.

---

6. System Architecture Overview

The system is built on a modular, service-oriented architecture.

- **Frontend:** A React-based web application provides the user interface, including the chat interface, visual collaboration view, and workforce management dashboard.
- **Backend:** Supabase is used for the database, authentication, and serverless functions.
- **Core Services:** A set of TypeScript services handle the core logic of the AI Workforce, including orchestration, reasoning, and agent selection.
- **Integrations:** The system integrates with external services for LLMs, code execution, web search, and more.

---

7. Data Model

The database schema is designed to support the core features of the platform. Key tables include:

- `users`: Stores user information and their subscription plan.
- `purchased_employees`: Tracks which AI Employees a user has "hired".
- `chat_sessions` and `chat_messages`: Store the conversation history between users and the AI Workforce.
- `automation_workflows` and `automation_executions`: Manage and track the execution of complex workflows.
- `token_usage`: Tracks the number of tokens used for billing and analytics.

---

8. Future Considerations

- **Advanced Collaboration:** Implement more sophisticated collaboration protocols between AI Employees, allowing for negotiation and dynamic task allocation.
- **Self-Improving Workforce:** Enable the workforce to learn from past executions and improve its performance over time.
- **Expanded Toolset:** Integrate a wider range of tools and services to enhance the capabilities of the AI Employees.
- **Team Creation:** Allow users to create and manage teams of AI Employees for specific projects.