**# Requirements Documentation**

## **1. Introduction**

### **1.1 Purpose**

This document outlines the requirements for a web-based Kanban board designed to facilitate task tracking and team collaboration. The platform enforces a "single-task focus" approach by allowing only one active task in the "Doing" column at a time, which triggers a focus timer. Additionally, the system integrates an AI-powered chatbox that enables users to analyze task data and gain insights for improved productivity.

### **1.2 Scope**

The system will provide:

* A Kanban board for task management
* Enforcement of single-task focus in the "Doing" column
* A focus timer triggered when a task is moved to "Doing"
* An AI-powered chatbox for querying task data and generating insights
* Support for team collaboration and progress tracking

### **1.3 Goals and Objectives**

* Enhance user productivity by preventing multitasking
* Help users achieve "Inbox Zero" by maintaining a streamlined workflow
* Provide AI-driven insights to optimize task completion and efficiency
* Foster effective team collaboration through a structured Kanban workflow

## **2. Functional Requirements**

### **2.1 User Roles**

**1. Unified Permissions for All Users**

* All users will have the same permissions to create and update tasks.
* They can move tasks freely between columns (Backlog, To-Do, Doing, Done).
* They can add comments and files to assist in task progress.
* They can view task progress and the time spent on each task.

**2. Access Control**

* **Content Access Monitoring**: Access to projects can be limited so that users can only view projects they've been invited to.
* **Prevent Edits on Tasks Not Created by the User**: Restrictions can be enforced to prevent users from modifying tasks they didn't create.

**3. Logging and Monitoring**

* A simple mechanism can be added to track changes made by users, allowing the team to review actions without the need for roles.

### **2.2 Kanban Board Features**

* Task creation, editing, and deletion
* Drag-and-drop task management across columns
* Column states: "Backlog," "To-Do," "Doing," and "Done"
* Restriction: Only one active task in the "Doing" column
* Focus timer activation when a task is moved to "Doing"

### **2.3 AI-Powered Chatbox**

* Integration with OpenRouter, Gemini API, or similar AI models
* Natural language processing to answer user queries
* Task analytics, predictions, and recommendations
* Insights based on stored task data in the database

### **2.4 Team Collaboration**

* Multi-user support with real-time updates
* Task assignment and notifications
* Commenting and file attachment support

### **2.5 User Authentication & Registration**

* **Sign Up and Login**:  
  + Users can sign up and log in via Supabase Auth, using email/password authentication.
  + Upon successful login, the user is issued an authentication token or session cookie that maintains their session for secure access.
  + User credentials and session data are securely stored and managed by Supabase.
* **Backend Authentication Process**:  
  + Upon login, the backend validates user credentials, retrieves user data from Supabase, and sets up a session for the user.
  + The system ensures that only authenticated users can access their team tasks, AI chatbot interactions, and other private data.
  + Token-based authentication is used to maintain the user’s session securely across the application.
* **Session Management**:  
  + Supabase handles session management through cookies or JWT tokens, ensuring users stay logged in without repeated logins.
  + The session expires after a predefined period, and users will be asked to reauthenticate after logging out or session expiration.

## **3. Non-Functional Requirements**

### **3.1 Performance**

* The system should support concurrent users with minimal latency
* AI responses should be generated within a few seconds

### **3.2 Security**

* User authentication and role-based access control
* Secure storage and transmission of task data

### **3.3 Usability**

* Intuitive and user-friendly UI/UX
* Responsive design for desktop and mobile devices

### **3.4 Scalability**

* Ability to support growing teams and task volumes

## **4. Technical Requirements**

### **4.1 Frontend**

* React.js + Next.js for a dynamic UI
* Tailwind CSS or Material UI for styling

### **4.2 Backend**

* Python or Next.js (Express) for backend logic

### **4.3 Database**

* Supabase for database management

### **4.4 Authentication**

* Supabase Auth for user authentication

### **4.5 AI Integration**

* OpenRouter (DeepSeek, Qwen, or Mistral) or Gemini (2.5) Pro for AI-powered analysis

### **4.6 Hosting & Deployment**

* Vercel for frontend deployment

## **5. Assumptions and Constraints**

* The system assumes stable internet connectivity for real-time updates and AI interactions
* AI insights depend on the quality and structure of task data stored
* User roles will be predefined and managed by an administrator

## **6. Future Enhancements (Optional)**

* Mobile app version
* Advanced AI-driven task prioritization
* **Task Dependencies** – Ability to set dependencies between tasks
* **Reporting & Analytics** – Dashboards for task progress and performance metrics
* **User Notifications** – Email or push notifications for updates and deadlines
* **Offline Mode** – Support for working without an internet connection, with sync on reconnection

This document serves as a foundation for the design and development of the Kanban board system, ensuring clarity in requirements and expectations.