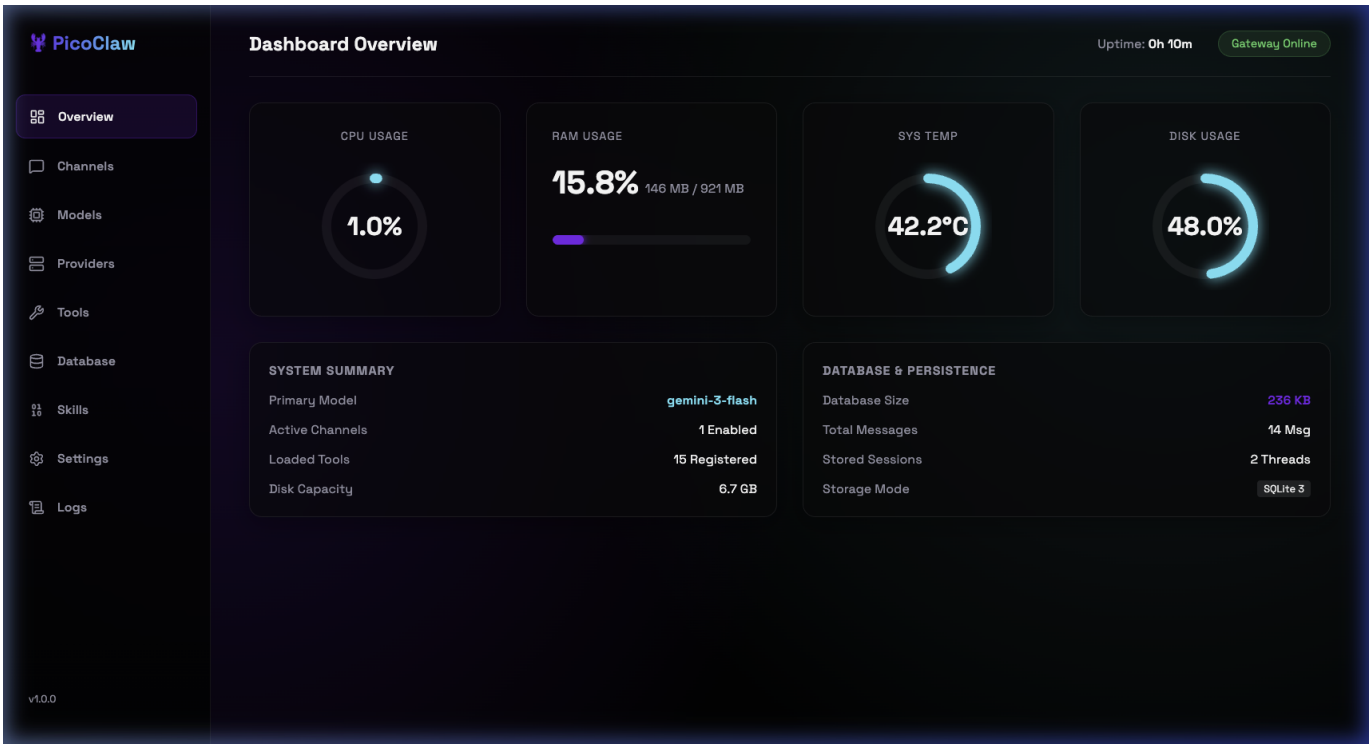
PicoClaw Dashboard Documentation

Overview

The PicoClaw Dashboard is a web-based management interface for the PicoClaw personal AI assistant. It provides a centralized view of the system status, configuration, and extensibility options.

Dashboard Pages

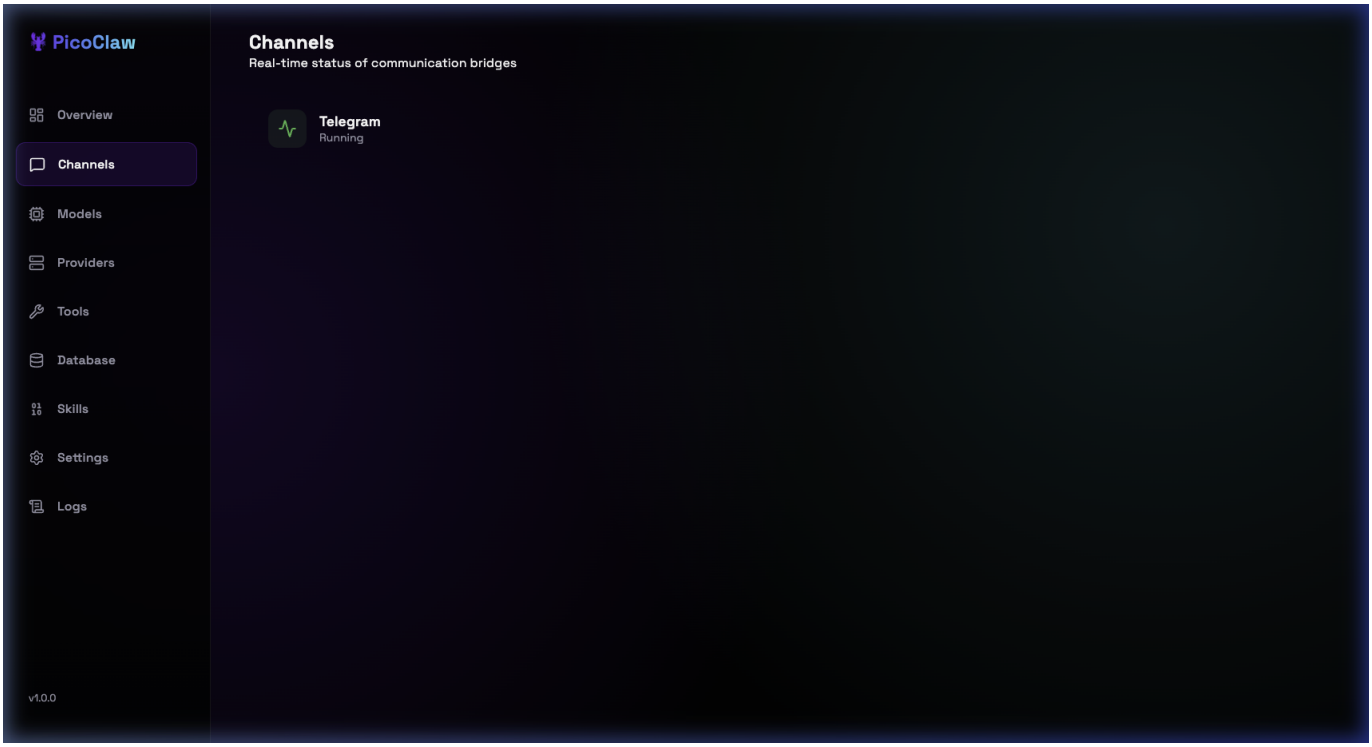
1. Overview



The landing page of the dashboard. It displays real-time system metrics including:

- - **System Stats**: Uptime, CPU Usage, RAM Usage, and Temperature.
- - **Disk Usage**: Total, used, and percentage usage.
- - **Database Metrics**: Total message count, session count, and database file size.
- - **Metadata**: The primary AI model currently in use and the count of active channels and tools.

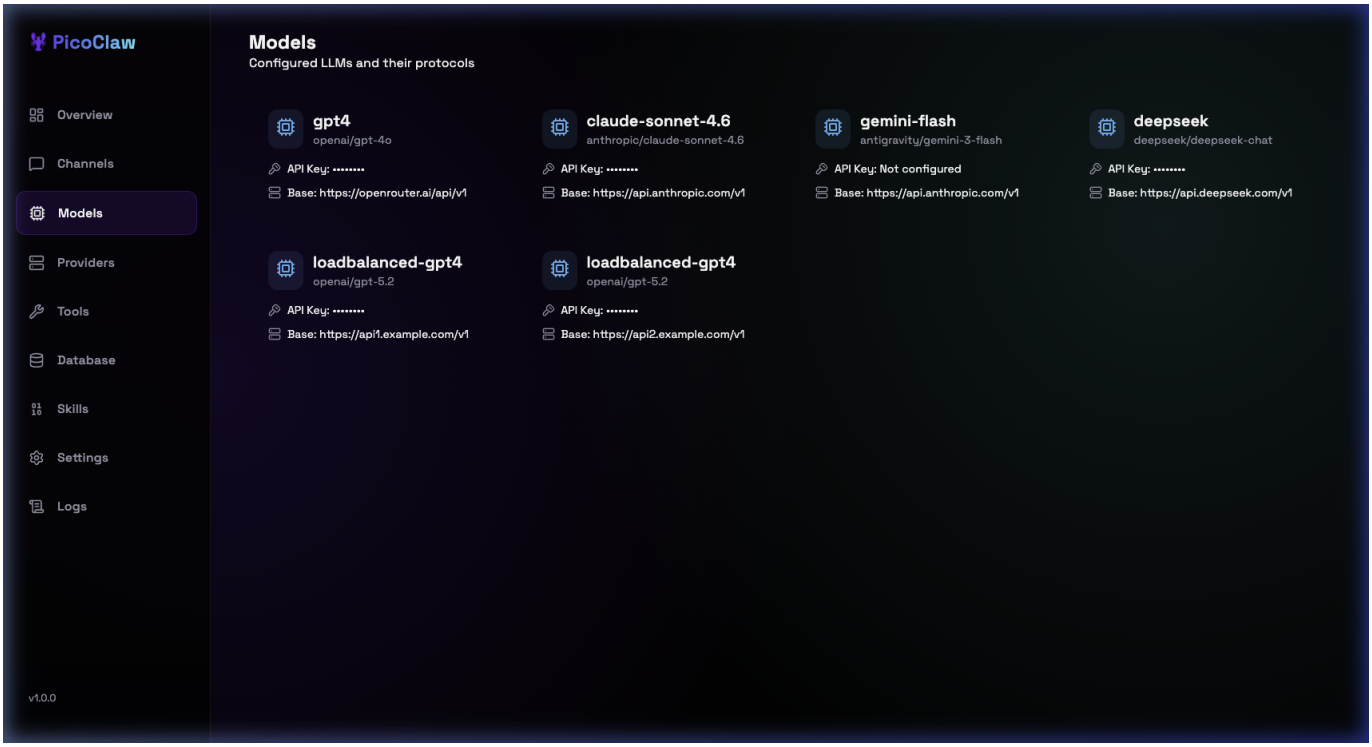
2. Channels



Displays the status of communication channels.

- - **Active Channels**: Lists channels like Telegram, indicating if they are running.
- - **Connection Status**: Real-time status of the bot connections.

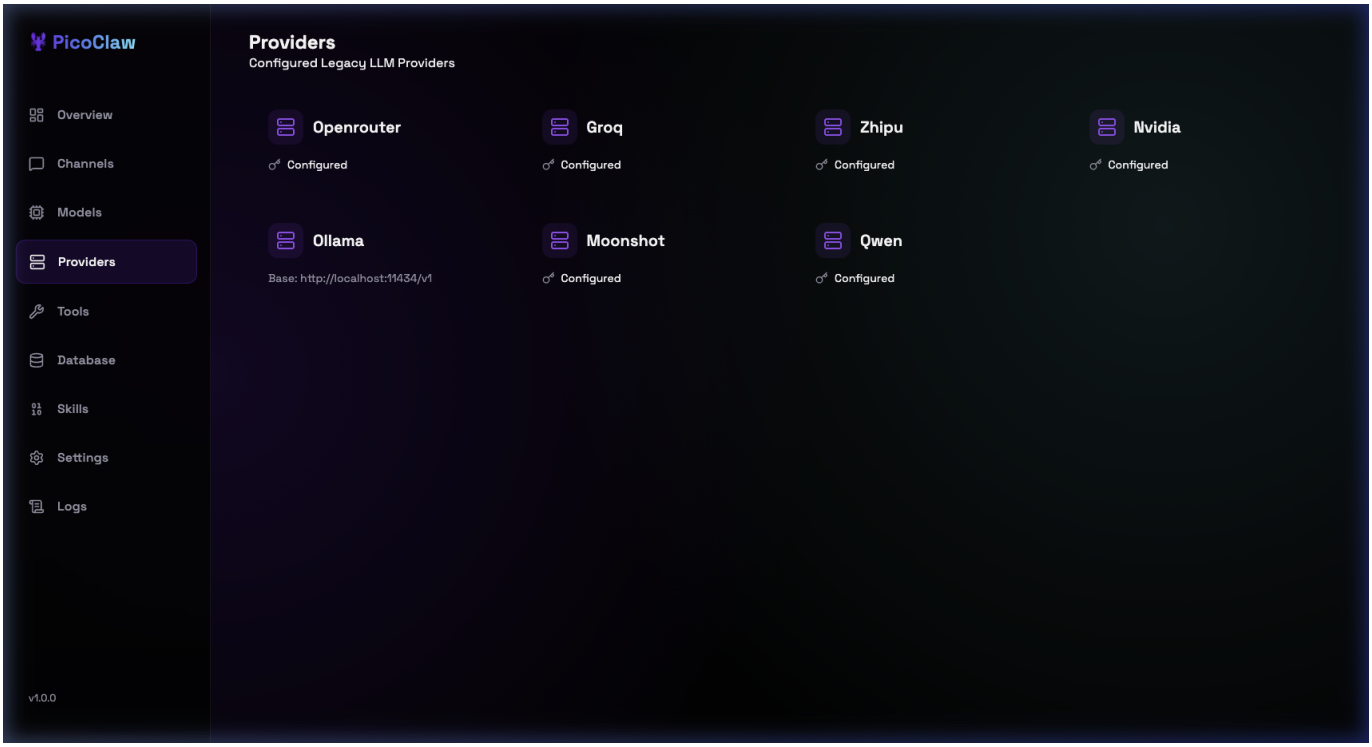
3. Models



Lists all available Large Language Models (LLMs) configured in the system.

- - Shows the model name, provider, and parameters.

4. Providers



Displays the configuration of LLM providers (e.g., Gemini, OpenAI, Claude).

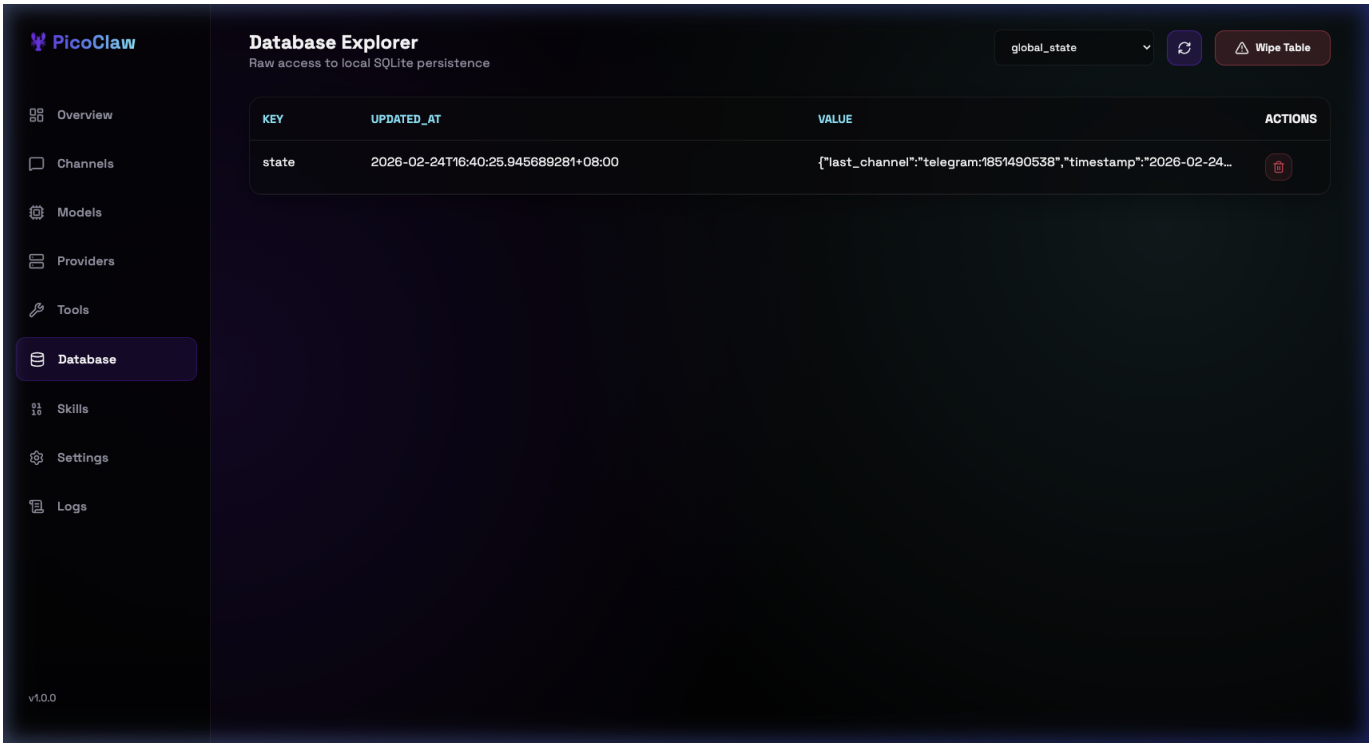
- - It allows verifying provider connectivity and base API URLs.

5. Tools

A library of all tools currently registered with the agent.

- - Each tool entry includes its name, description, and the parameters it accepts.

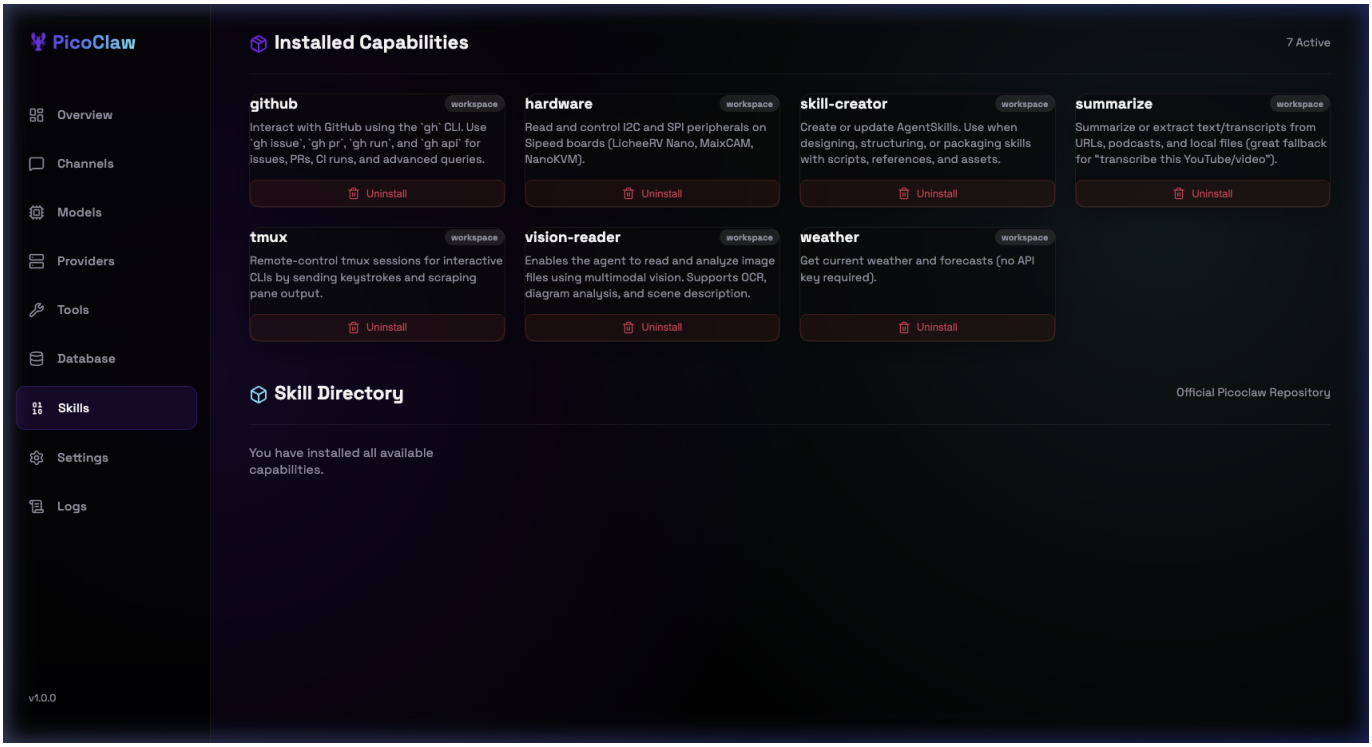
6. Database



Allows direct interaction with the underlying SQLite database.

- - **Tables**: List all tables in the system.
- - **Query/View**: Browse rows within tables.
- - **Maintenance**: Delete specific rows or wipe entire tables for cleanup.

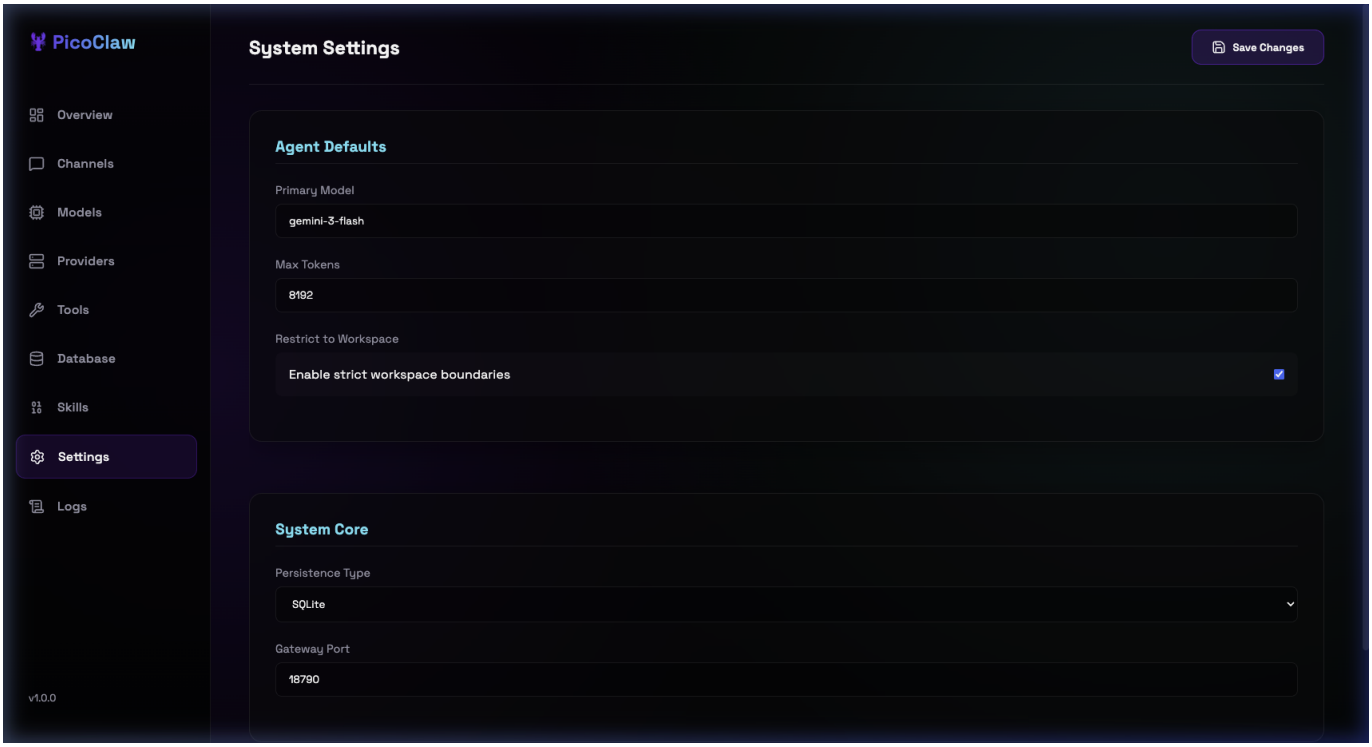
7. Skills



The extensibility hub for PicoClaw.

- - **Installed Skills**: Lists all skills currently loaded from `~/.picoclaw/workspace/skills`.
- - **Installation**: Install new skills directly from GitHub repositories.
- - **Management**: Uninstall skills that are no longer needed.

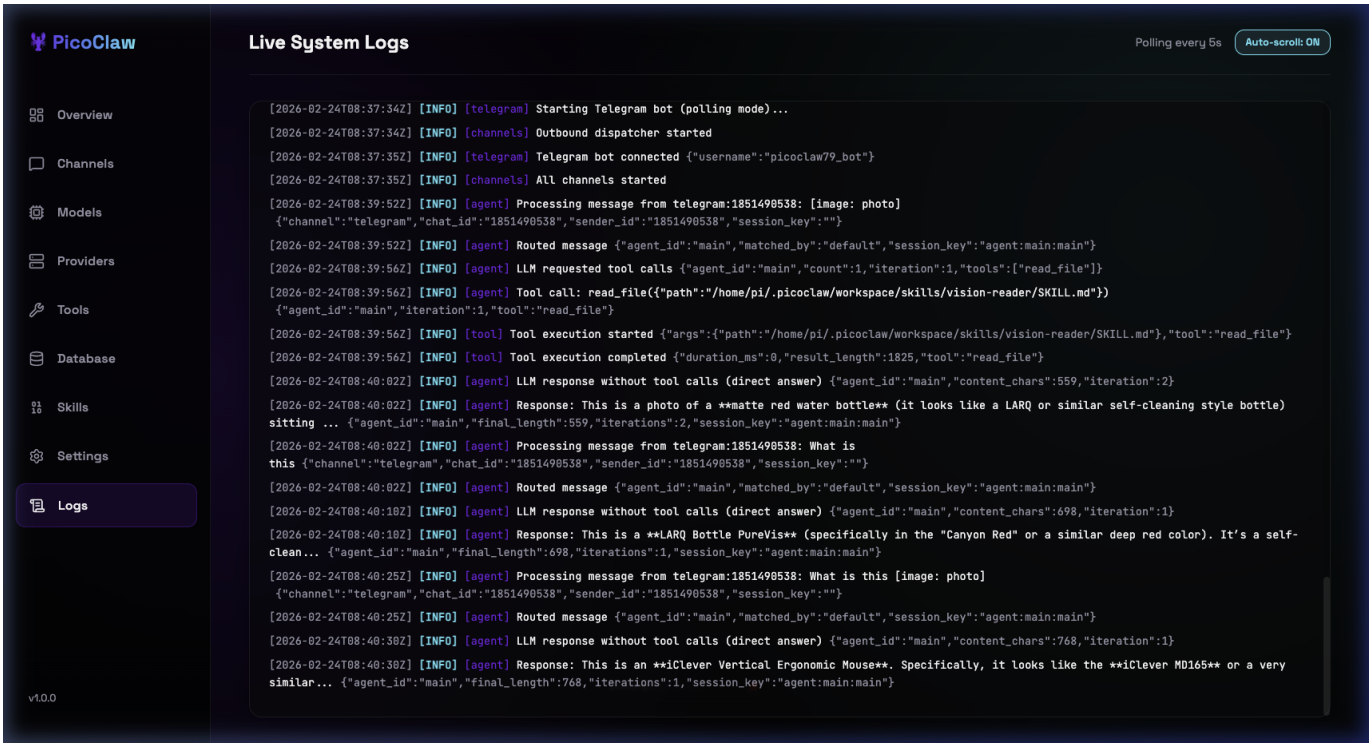
8. Settings



Advanced configuration portal.

- - View and modify the `config.json` file directly from the UI.
- - Update agent behavior, provider keys, and system thresholds.

9. Logs



A live view of the system logs.

- - Displays the last 100 lines of the `picoclaw.log` file for debugging and monitoring.

Backend API Reference

	Description	
	Returns system metrics and metadata.	
	Retrieves or updates the system configuration.	
	Returns the last 100 lines of system logs.	
	Lists all installed skills.	
	Fetches available skills from the registry.	
	Triggers a skill installation from GitHub.	
	Uninstalls a specific skill.	
	Lists all database tables.	
	Retrieves rows from a specified table.	
	Deletes a specific row by ID.	

	Description	
	Wipes an entire table.	
	Lists configured AI models.	
	Returns channel status information.	
	Lists configured LLM providers.	
	Lists all registered tool definitions.	

Technical Architecture

- - **Backend**: Built with Go, utilizing a modular architecture for providers, channels, and tools.
- - **Frontend**: A React SPA (Single Page Application) built with TypeScript and Vite, served via an embedded filesystem in the Go binary.
- - **Persistence**: SQLite for message history and agent state.
- - **Service Management**: Deployable via Systemd on Linux (Raspberry Pi).