AthenaAI: User Manual & System Installation Guide

Introduction

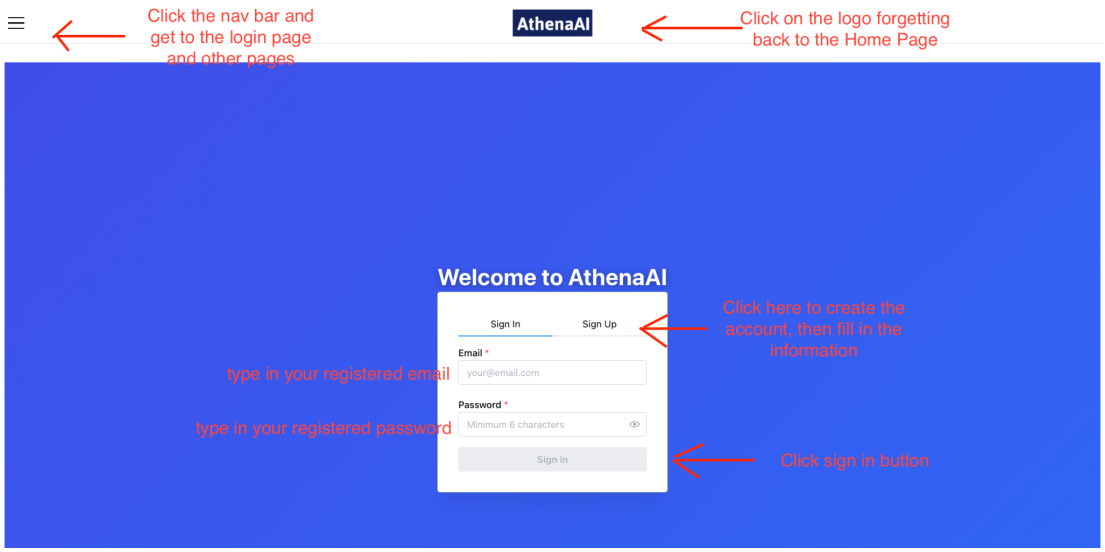
- **Project Name:** Virtual Psychologist
- **Overview:** This project aims to build a personalized and accessible mental health support system using Generative Artificial Intelligence (GenAI). The system is designed to provide therapeutic interactive sessions based on Cognitive Behavioral Therapy (CBT).
  The project's architecture consists of the following main components:
  - **Frontend:** The user interface is built with **React** and **TypeScript** to ensure a safe, efficient, and interactive user experience.
  - **Backend:** The server is developed with **Node.js**, responsible for handling business logic, managing user data, and acting as a communication intermediary.
  - **AI Service:** A dedicated AI service built with **Python**, tasked with integrating and orchestrating multiple API keys from various Large Language Models (LLMs) to generate intelligent therapeutic responses.

Part 1: User Manual

- **Audience:** This guide is intended for end-users and does not require technical knowledge.

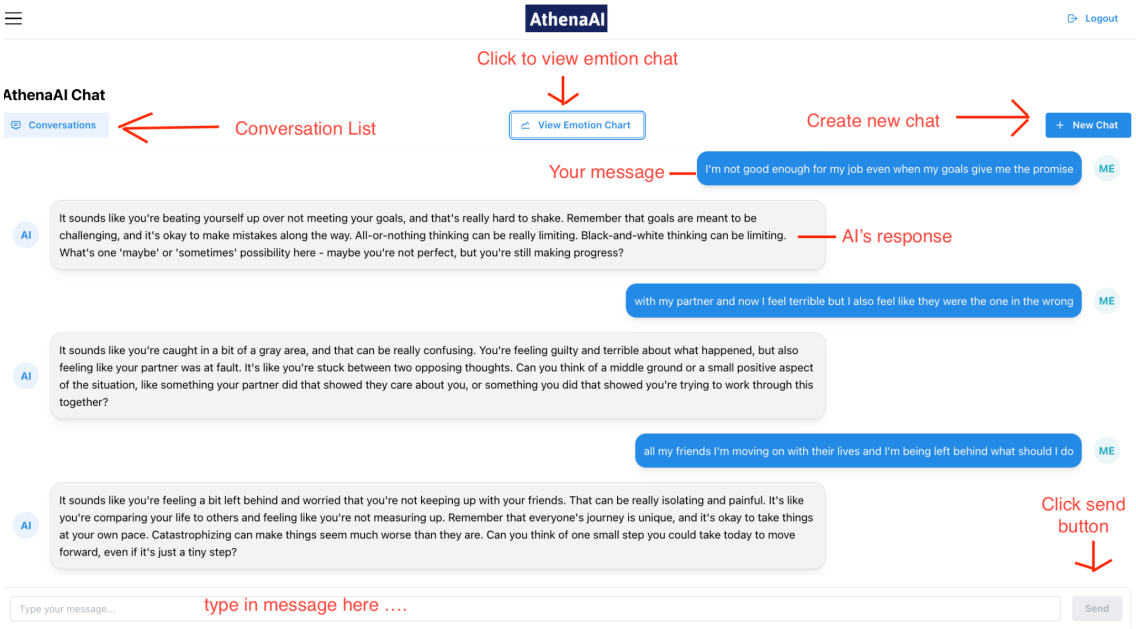
1.1. Getting Started

- **Creating an Account & Logging In:** Step-by-step instructions for registering a new account and logging into the platform.

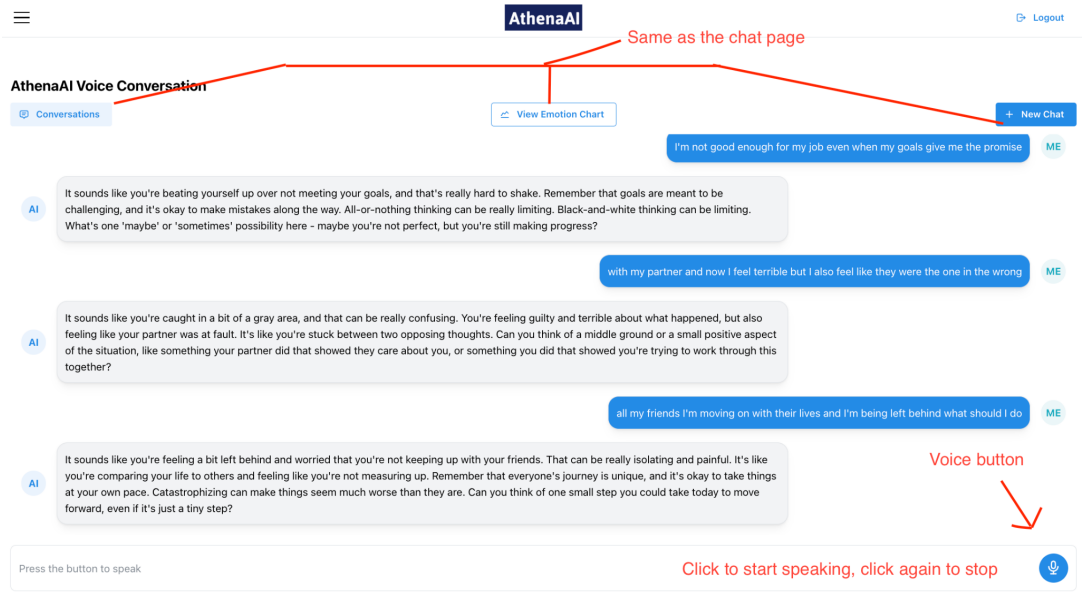


1.2. Main Features and Pages

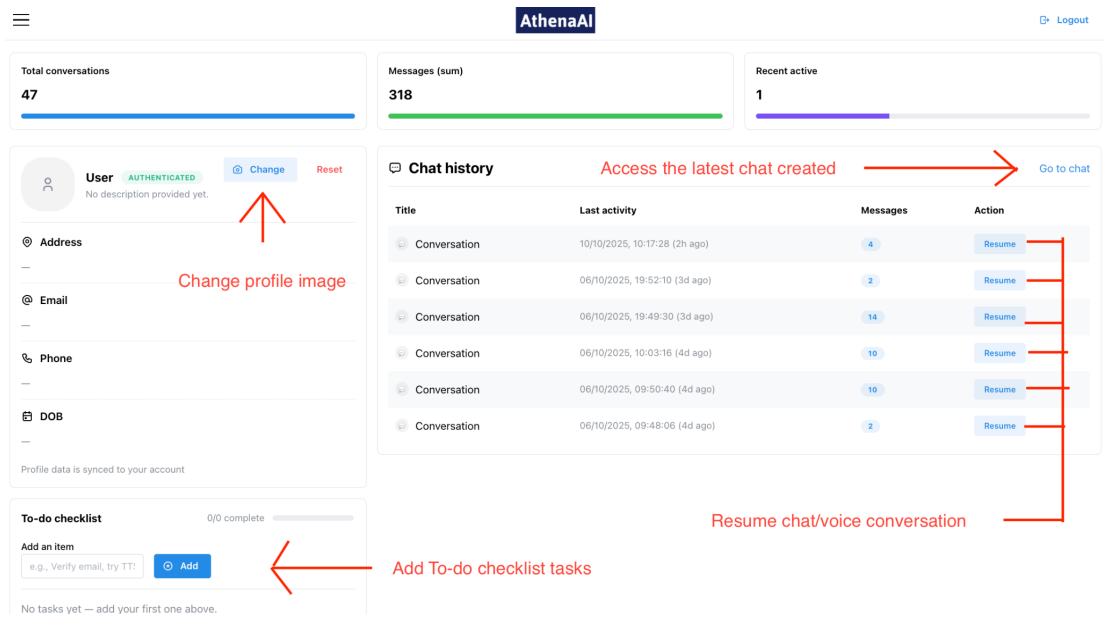
- **Informational Pages:**
  - A brief overview of the site's static pages: Home, Blog, Service, and Contact, which provide product and project information.
- **Chat Page:**
  - The primary feature for interacting with the virtual psychologist via text.
  - How to send a message and receive a CBT-based response.
  - How to use the "View Emotion Chart" feature to see an emotional analysis of the conversation.



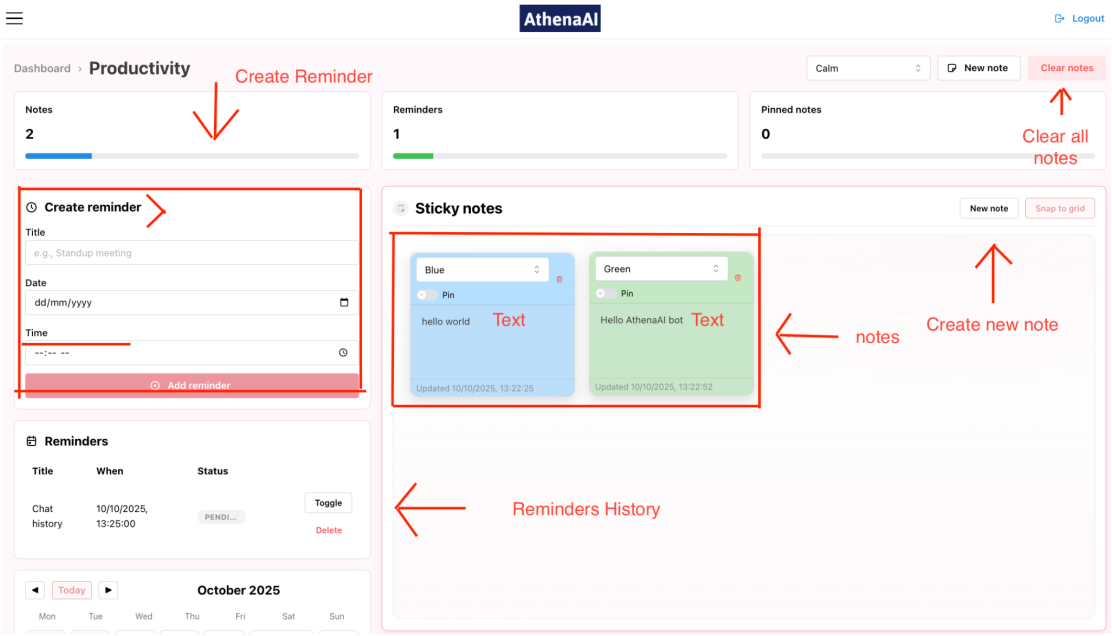
- **Voice Page:**
  - The main feature for voice-based interaction.
  - How to press the button to speak and receive an empathetic, spoken response from the bot.
  - Includes the "View Emotion Chart" feature for conversation analysis.



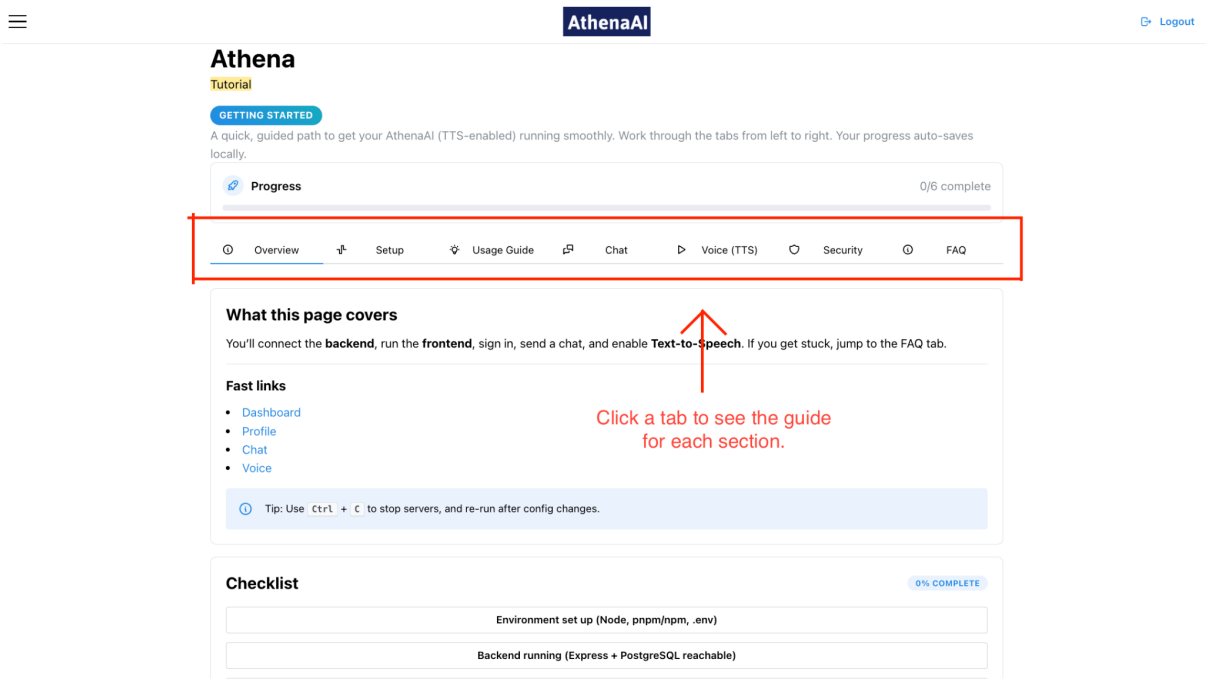
- **Profile Page:**
  - How to view your account information.
  - How to access and review your entire conversation history.



- **Productivity Page (base on your request):7**
    - A dedicated space for users to write and manage notes and comments.



- **Tutorial Page**
    - An in-depth manual embedded within the application, providing detailed instructions for all features.



Part 2: System Installation & Deployment Manual

- **Audience:** This guide is intended for developers who wish to run the project in a local environment.

2.1. Architecture Overview

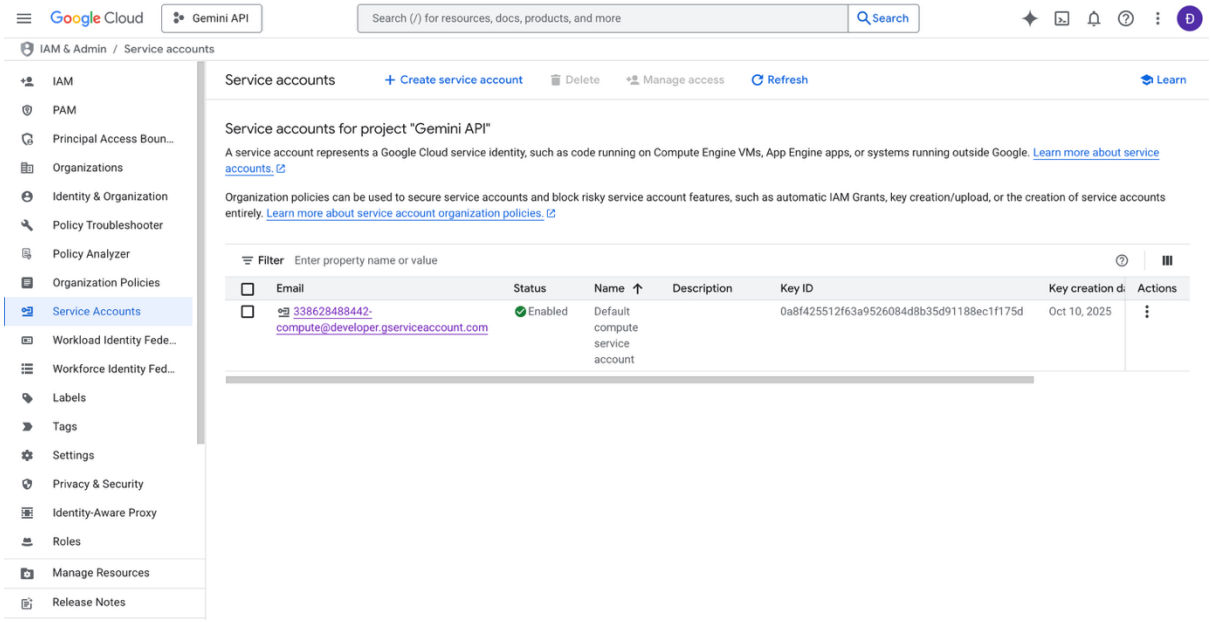
- A brief description of the project's 3-service architecture:
    - athenaos-frontend: Role, technologies (React, TypeScript).
    - athenaos-backend: Role, technologies (SQLite).
    - athenaos-ai-service: Role, technologies (Python, Fast API/Flask).

2.2. Prerequisites

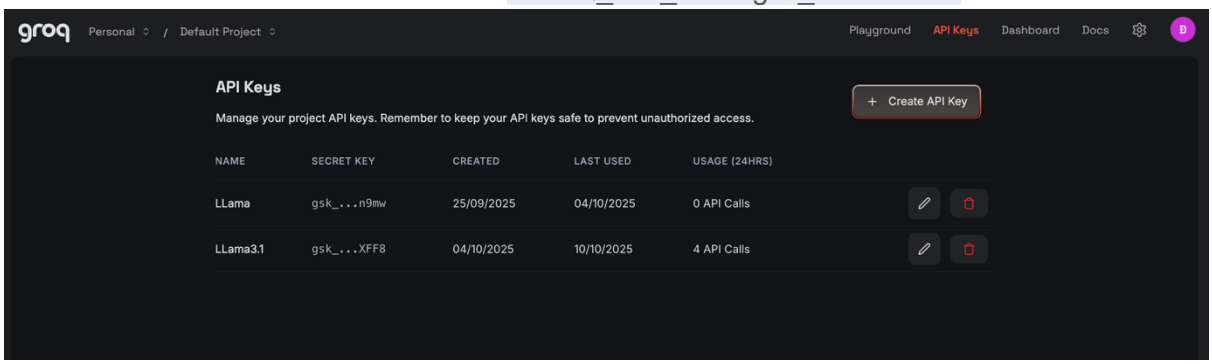
- **Software:**
    - Node.js: v20.14.0
    - Python: 3.13.5
    - pnpm (or npm/yarn).
- **Database:** SQLite (allocate in athenaos-backend)

2.3. Local Installation Guide

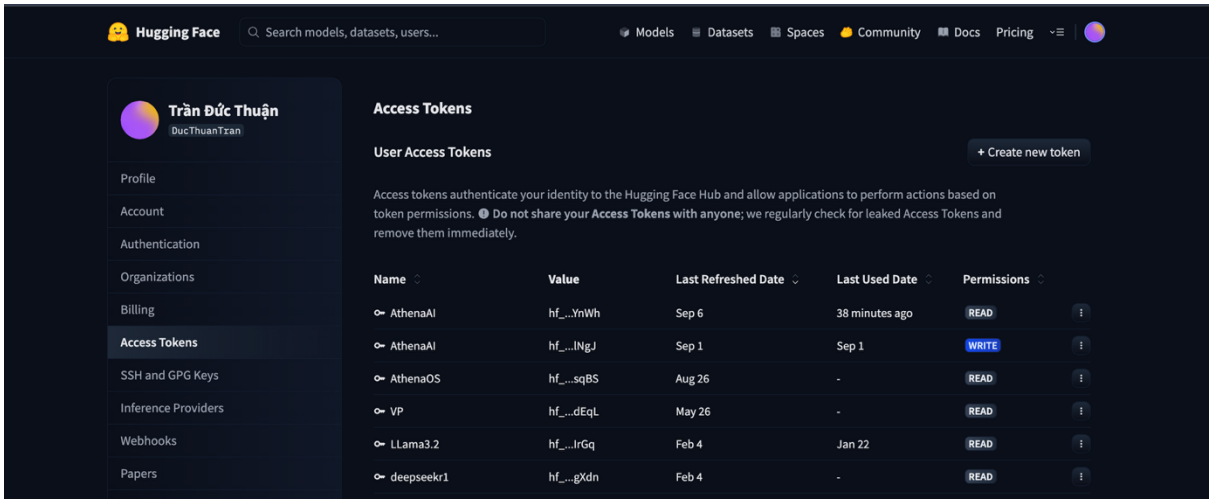
- **Step 1: Download and Unzip the Source Code**
    - Download the project's `.zip` file from the provided Google Drive link.
    - Unzip the file to a location of your choice on your local machine. This will create the main project folder containing the three service directories.
- **Step 2: Configure Environment Variables**
    - For each service (**athenaos-frontend**, **athenaos-backend**, **athenaos-aii-service**), locate the **.env** file and follow the instructions below to fill in the required values.
    - **For the Backend (athenaos-backend):**
        - **Standard Variables:** Based on the provided `.env` file, ensure the following variables are set. For security, it is highly recommended to change the `JWT_SECRET` to your own unique, complex string.
        - **Google Cloud TTS Credentials:** This service requires a JSON credential file for voice recognition and Text-to-Speech.
            1. Go to the **Google Cloud Console**, select your project, and navigate to **IAM & Admin > Service Accounts**.
            2. Select an existing service account or create a new one.
            3. Go to the **KEYS** tab, click **ADD KEY**, and select **Create new key**.
            4. Choose the key type **JSON** and click **CREATE**.
            5. A .json file will be downloaded. Rename it to **google-tts-credentials.json** and move it directly into the athenaos-backend folder.



- **For the AI Service (`athenaos-ai-service`):**
  - The `.env` file for this service requires two API keys.
  - **Groq API Key (for Llama 3.1 Model):**
    1. Go to the **Groq Console** (https://console.groq.com/) and log in.
    2. Navigate to the **API Keys** tab and click **Create API Key**.
    3. Name your key and copy the generated key string (it will only be shown once).
    4. In the `.env` file, paste the key into the `GROQ_API_KEY` variable: `GROQ_API_KEY=gsk_xxxxxxxxxx`.



  - **Hugging Face Access Token (for supporting models):**
    1. Go to **Hugging Face** (https://huggingface.co/) and log in.
    2. Navigate to your profile **Settings > Access Tokens**.
    3. Click **New token**, give it a name, and select the **Read** role.
    4. Copy the generated token (it will only be shown once).
    5. In the `.env` file, paste the token into the `HUGGINGFACE_API_KEY` variable: `HUGGINGFACE_API_KEY=hf_xxxxxxxxxx`.



    - **For the Frontend (athenaos-frontend):**
        - In the .env file, ensure the VITE_API_BASE_URL variable is set to the correct address of your running backend service (e.g., http://localhost:8888).
- **Step 3: Run the Backend Service**
    - Open terminal 1, navigate to athenaos-backend, install dependencies (npm install), and run the start command (npm start).

```
thuanduc@Thuans-MacBook-Pro athenaos-backend % npm install

up to date, audited 286 packages in 711ms

37 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
thuanduc@Thuans-MacBook-Pro athenaos-backend % npm start

> start
> node server.js

Running in development mode, using SQLite.
[dotenv@17.2.2] injecting env (3) from .env -- tip: 🔐 encrypt with Dotenvx: https://dotenvx.com
Executing (default): SELECT 1+1 AS result
Database connection has been established successfully.
Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Users';
Executing (default): PRAGMA INDEX_LIST(`Users`)
Executing (default): PRAGMA INDEX_INFO(`sqlite_autoindex_Users_1`)
Executing (default): PRAGMA INDEX_INFO(`sqlite_autoindex_Users_2`)
Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Conversations';
Executing (default): PRAGMA INDEX_LIST(`Conversations`)
Executing (default): SELECT name FROM sqlite_master WHERE type='table' AND name='Messages';
Executing (default): PRAGMA INDEX_LIST(`Messages`)
All models were synchronized successfully.
Server is running on port 8888
```

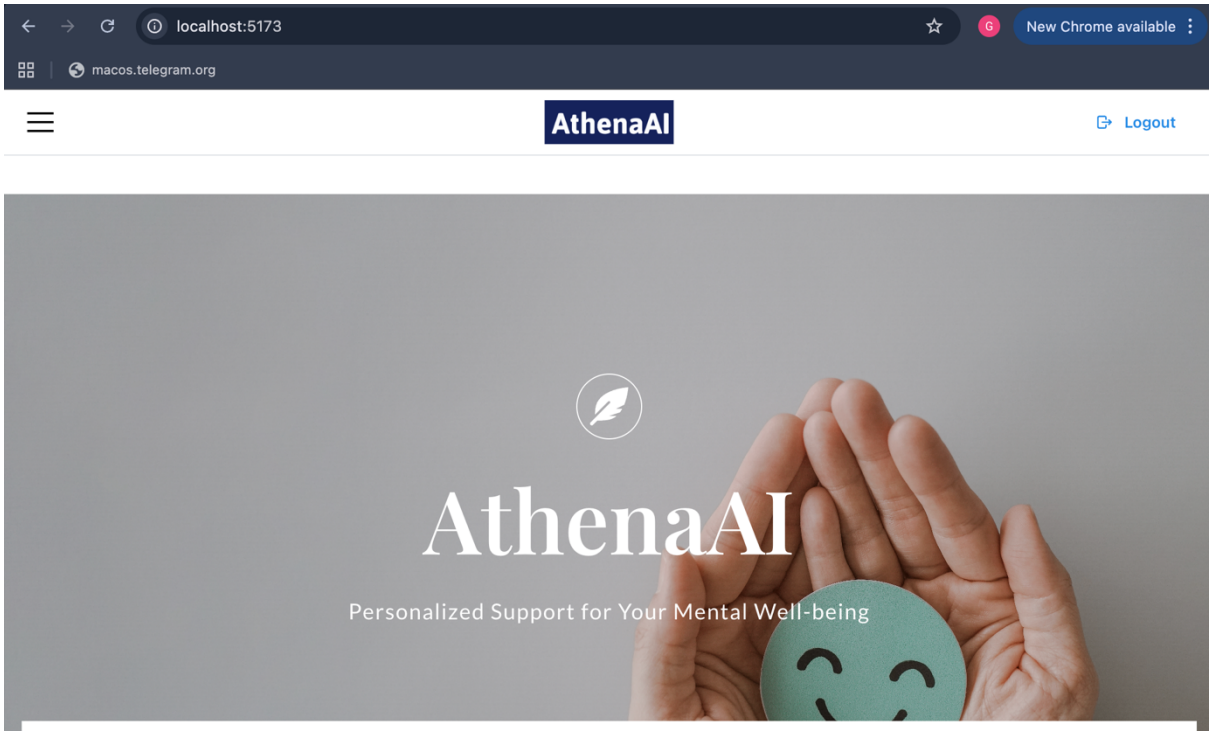
- **Step 4: Run the Frontend Service**
    - Open terminal 2, navigate to athenaos-frontend, install dependencies (npm install), and run the start command (npm run dev).

```
thuanduc@Thuans-MacBook-Pro AthenaAI % cd athenaos-frontend
thuanduc@Thuans-MacBook-Pro athenaos-frontend % npm install
npm warn EBADENGINE Unsupported engine {
npm warn EBADENGINE   package: 'vite@7.1.5',
npm warn EBADENGINE   required: { node: '^20.19.0 || >=22.12.0' },
npm warn EBADENGINE   current: { node: 'v20.14.0', npm: '10.7.0' }
npm warn EBADENGINE }

up to date, audited 264 packages in 612ms

59 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
thuanduc@Thuans-MacBook-Pro athenaos-frontend % npm run dev

> athenaos-frontend@0.0.0 dev
> vite

You are using Node.js 20.14.0. Vite requires Node.js version 20.19+ or 22.12+. Please upgrade your Node.js version.

  VITE v7.1.5  ready in 173 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

- **Step 5: Run the AI Service**
  - Open terminal 3, navigate to athenaos-ai-service, install dependencies (pip install -r requirements.txt), and run the start command (uvicorn main:app --reload).

```
thuanduc@Thuans-MacBook-Pro AthenaAI % cd athenaos-ai-service
thuanduc@Thuans-MacBook-Pro athenaos-ai-service % pip install -r requirements.txt
Requirement already satisfied: fastapi in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from -r require
ments.txt (line 4)) (0.115.14)
Requirement already satisfied: uvicorn in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from -r require
ments.txt (line 5)) (0.35.0)
Requirement already satisfied: requests in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from -r requir
ements.txt (line 8)) (2.32.3)
Requirement already satisfied: python-dotenv in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from -r r
equirements.txt (line 9)) (1.0.1)
Requirement already satisfied: groq in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from -r requiremen
ts.txt (line 10)) (0.32.0)
Requirement already satisfied: starlette<0.47.0,>=0.40.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-package
s (from fastapi->-r requirements.txt (line 4)) (0.46.2)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4 in /Library/Frameworks/Python.framework/Versio
ns/3.12/lib/python3.12/site-packages (from fastapi->-r requirements.txt (line 4)) (2.9.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages
 (from fastapi->-r requirements.txt (line 4)) (4.14.1)
Requirement already satisfied: annotated-types>=0.6.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (
from pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 4)) (0.7.0)
Requirement already satisfied: pydantic-core==2.23.3 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (f
rom pydantic!=1.8,!=1.8.1,!=2.0.0,!=2.0.1,!=2.1.0,<3.0.0,>=1.7.4->fastapi->-r requirements.txt (line 4)) (2.23.3)
Requirement already satisfied: anyio<5,>=3.6.2 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from st
arlette<0.47.0,>=0.40.0->fastapi->-r requirements.txt (line 4)) (4.9.0)
Requirement already satisfied: idna>=2.8 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from anyio<5,
>=3.6.2->starlette<0.47.0,>=0.40.0->fastapi->-r requirements.txt (line 4)) (3.10)
Requirement already satisfied: sniffio>=1.1 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from anyio
<5,>=3.6.2->starlette<0.47.0,>=0.40.0->fastapi->-r requirements.txt (line 4)) (1.3.1)
Requirement already satisfied: click>=7.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from uvicorn
->-r requirements.txt (line 5)) (8.1.7)
Requirement already satisfied: h11>=0.8 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from uvicorn->
-r requirements.txt (line 5)) (0.14.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages
 (from requests->-r requirements.txt (line 8)) (3.4.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from
 requests->-r requirements.txt (line 8)) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from
 requests->-r requirements.txt (line 8)) (2025.1.31)
Requirement already satisfied: distro<2,>=1.7.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from g
roq->-r requirements.txt (line 10)) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from g
roq->-r requirements.txt (line 10)) (0.28.1)
Requirement already satisfied: httpcore==1.* in /Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/site-packages (from http
x<1,>=0.23.0->groq->-r requirements.txt (line 10)) (1.0.7)
thuanduc@Thuans-MacBook-Pro athenaos-ai-service % uvicorn main:app --reload

INFO:     Will watch for changes in these directories: ['/Users/thuanduc/AthenaAI/athenaos-ai-service']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [60500] using WatchFiles
API-based configuration file loaded successfully.
INFO:main:Groq client configured for model: llama-3.1-8b-instant
INFO:     Started server process [60502]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

- **Step 6: Verification**
  - Open a web browser and navigate to the frontend's address (e.g., http://localhost:5173) to confirm that everything is working.


localhost:5173
New Chrome available
macos.telegram.org
AthenaAI
Logout
AthenaAI
Personalized Support for Your Mental Well-being