# Obsidian Project Handbook

This document is your complete guide to setting up, running, and maintaining the Obsidian AI Security Advisor project. Keep this file handy for future reference.

## 1. Project Architecture Overview

Before you run the project, it's helpful to understand how the pieces fit together.

1. **Frontend (.html, .css, .js files):** This is the website you see and interact with in your browser. When you click a button (e.g., "Ask Obsidian"), it sends a request to the backend.
2. **Backend (app.py):** This is the "brain" of the operation. It's a Flask server that listens for requests from the frontend. It uses the AI and the database to process those requests and sends back a response.
3. **Vector Database (db folder):** This is the AI's specialized memory. The rag\_setup.py script reads your knowledge\_base.txt, converts it into a format the AI can understand quickly, and stores it here.

## 2. Initial Setup & Running the Backend

Follow these steps the first time you set up the project on a new machine.

### Step 1: Set Up Your API Key

* Create a file named .env in the main project folder.
* Inside this file, add the following line, replacing "YOUR\_API\_KEY\_HERE" with your actual Gemini API key:  
  GEMINI\_API\_KEY="YOUR\_API\_KEY\_HERE"

### Step 2: Set Up Python Environment

* Open a terminal in your main project folder.
* Create a virtual environment to keep your project's packages separate:  
  python -m venv venv
* Activate the environment:
  + On Windows: venv\Scripts\activate
  + On macOS/Linux: source venv/bin/activate

### Step 3: Install Required Packages

* With your virtual environment active, install all the necessary Python libraries from the requirements.txt file:  
  pip install -r requirements.txt

### Step 4: Build the Initial Database

* This is a one-time step for the initial setup. Run the rag\_setup.py script to create your vector database:  
  python rag\_setup.py
* Wait for it to complete. You will see a new folder named db appear in your project.

### Step 5: Run the Backend Server

* Now, start your Flask server. This is the command you will use every time you want to run your project.  
  python app.py
* The terminal will show that the server is running (usually on http://127.0.0.1:5001). **Keep this terminal window open while you use the website.**

### Step 6: Launch the Frontend

* Open the index.html file in your web browser. The website is now fully functional and can communicate with your running backend.

## 3. How to Update the Knowledge Base (IMPORTANT)

**This is the process you must follow every time you make changes to the knowledge\_base.txt file.**

The AI does not automatically know when you change the text file. You need to "re-train" it by rebuilding the database.

### Step 1: Stop the Backend Server

* Go to the terminal window where app.py is running.
* Press Ctrl + C to stop the server.

### Step 2: Delete the Old Database Folder

* In your project folder, find the directory named db.
* **Delete the entire db folder.** This step is crucial to ensure a clean rebuild.

### Step 3: Re-run the Database Setup Script

* In your terminal (make sure your virtual environment is still active), run the rag\_setup.py script again:  
  python rag\_setup.py
* This will create a new db folder based on your updated knowledge\_base.txt file.

### Step 4: Restart the Backend Server

* Once the setup script is finished, you can restart your backend server:  
  python app.py
* Your website is now running with the updated knowledge.

## 4. Quick Troubleshooting

* **"My website says 'Could not connect to the Obsidian server'."**
  + This means your backend server is not running. Make sure you have the app.py script running in an open terminal window.
* **"I'm getting an API key or credentials error."**
  + Double-check that your .env file is named correctly and that you have pasted your API key inside the quotes.
* **"My new information isn't showing up in the AI's answers."**
  + You need to update the database. Follow the three steps in the **"How to Update the Knowledge Base"** section above.