

Configuration Manual

MSc Research Project
M.Sc. Artificial Intelligence

Shreeraj Santosh Sangle
Student ID: x23283254

School of Computing
National College of Ireland

Supervisor: Lavish Thomas

National College of Ireland
MSc Project Submission Sheet



School of Computing

Student Name: Shreeraj Santosh Sangle

Student ID: 23283254

Programme: M. Sc Artificial Intelligence

Year: 2024-25

Module: MSc (Research) Practicum

Lecturer: Lavish Thomas

Submission

Due Date: 11th August 2025

Project Title: SoulSense

Word Count:

572

Page Count: 7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Shreeraj Sangle

Date: 11/08/25

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

SoulSense

Shreeraj Santosh Sangle
X23283254

1 Introduction

This configuration manual provides step-by-step instructions for deploying and running the **SoulSense AI** on a local workstation for development, testing, or assessment. It assumes you have the project ZIP extracted in a working directory

2 System Specification

2.1 The following is the hardware configurations of the machine.

Host Machine Specification

Requirements	Specifications	Recommended
Operating System	macOS 15.5	Windows 10/11, macOS 13+, Ubuntu 22.04+
Python Version	3.12.2	Python 3.10 or above
Node.js	18.x LTS	20.x (Latest)
PostgreSql	15	15
Disk Space	2 GB free	Upto 4 GB free

3 Required Tools and Libraries

1. Visual Code Studio: <https://code.visualstudio.com/>
2. Node.js 20.x LTS – <https://nodejs.org/>
3. PostgreSQL 16 – <https://www.postgresql.org/download/>
4. pgAdmin 4 – bundled with PostgreSQL installer
5. Python Latest Version: <https://www.python.org/downloads/>
6. Notepad++: <https://notepad-plus-plus.org/>

4 Project Structure

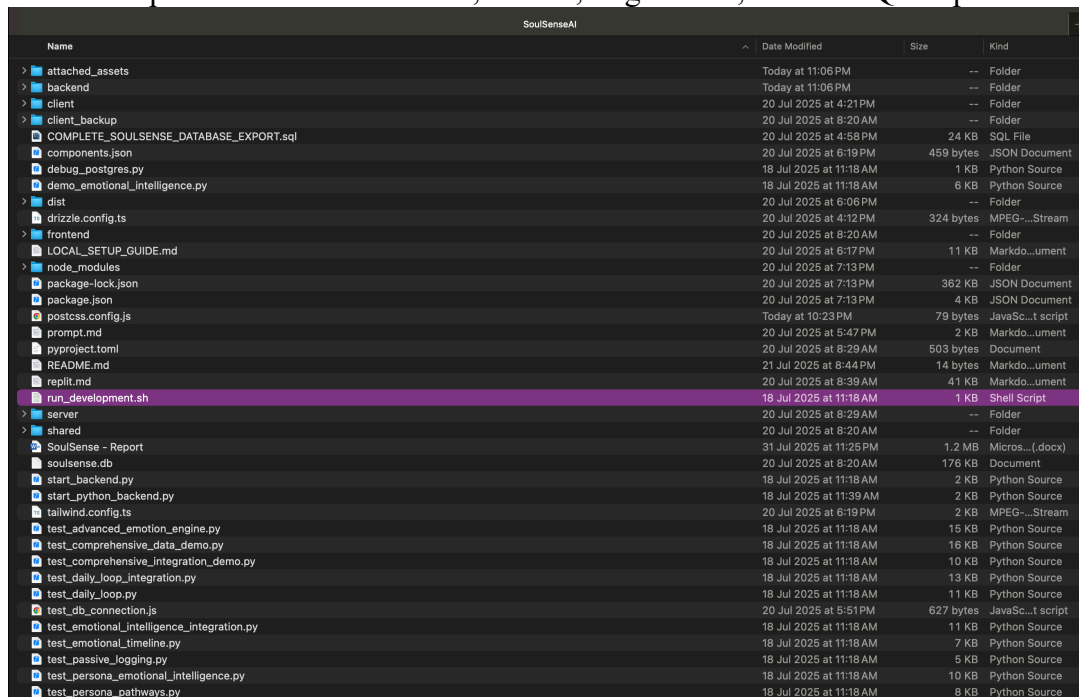
```
soulsense-ai/  
├── client/  
│   ├── src/  
│   │   ├── components/  
│   │   │   └── ui/  
│   │   ├── lib/  
│   │   ├── hooks/  
│   │   ├── pages/  
│   │   ├── App.tsx  
│   │   ├── main.tsx  
│   │   └── index.css  
│   ├── index.html  
│   └── server/  
│       ├── personas/  
│       ├── db.ts  
│       ├── storage.ts  
│       ├── routes-clean.ts  
│       ├── index.ts  
│       └── vite.ts  
├── shared/  
│   └── schema.ts  
├── package.json  
├── tsconfig.json  
├── vite.config.ts  
├── tailwind.config.ts  
├── postcss.config.js  
├── components.json  
├── drizzle.config.ts  
└── .env
```

Figure 1: Project Structure

5 Initial Setup

Step 1: Unzip the Project

Unzip or clone the repository into `~/Projects/soulsense-ai` (or any workspace you prefer). This creates the root folder that the remainder of the instructions assume. A quick directory tree at this point should show `client/`, `server/`, `migrations/`, and the SQL export file.



It should look similar to this.

Step 2: Installing the Dependencies

Open your terminal and change the project root with SoulSense by doing `cd SoulSense`, then there are 2 steps one the lengthy step and other the quick step.

```
Last login: Sun Aug 10 21:08:46 on ttys004
[(base) shreerajsangle@Shreerajs-Air ~ % cd desktop
[(base) shreerajsangle@Shreerajs-Air desktop % cd SoulSenseAI
```

By first Step :

Install all these dependencies manually present in package.json file

Using Step 2 :

Run the command called “`npm install`”. This command resolves every dependency listed in the package.json file like React, Vite, Drizzle-Kit, Tailwind, etc.—creating a local `node_modules/` folder and a lockfile that pins exact versions.

```
SoulSenseAI -- zsh -- 102x58
Last login: Sun Aug 10 21:08:46 on ttys004
[(base) shreerajsangle@Shreerajs-Air ~ % cd desktop
[(base) shreerajsangle@Shreerajs-Air desktop % cd SoulSenseAI
[(base) shreerajsangle@Shreerajs-Air SoulSenseAI % npm install
(node:69857) ExperimentalWarning: CommonJS module /opt/homebrew/lib/node_modules/npm/node_modules/debu
g/src/node.js is loading ES Module /opt/homebrew/lib/node_modules/npm/node_modules/supports-color/inde
x.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

up to date, audited 589 packages in 2s

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

5 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
(base) shreerajsangle@Shreerajs-Air SoulSenseAI %
```

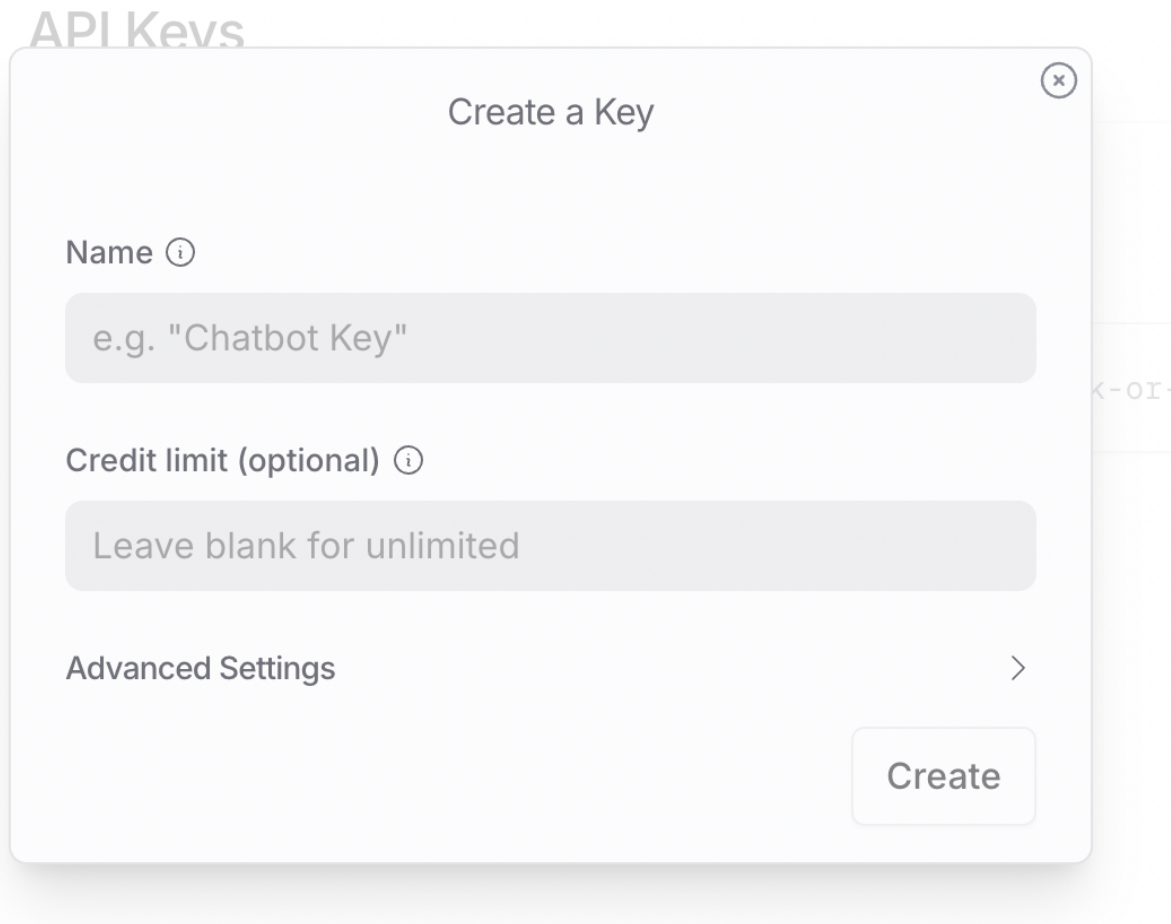
6. Create Environment

Open the new `.env` in VS Code (or any editor) and paste your OpenRouter key, local PostgreSQL connection string, and optionally set `ENVIRONMENT=development` plus `DEBUG=true` for verbose logs. The `dotenv` package will load these values at runtime so they never reach version control.

Your `.env` file should look similar to this :

```
.env
Import to Postman | You, 2 weeks ago | 1 author (You)
1 OPENROUTER_API_KEY=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
2 DATABASE_URL=postgresql://soulsense_ai:admin@localhost:5432/soulsense_ai
3 NODE_ENV=development
```

Creating Api key got to OpenRouter on your Web Browser <https://openrouter.ai/>
Open the API KEYS : <https://openrouter.ai/settings/keys>



API Keys

Create a Key

Name ⓘ

e.g. "Chatbot Key"

Credit limit (optional) ⓘ

Leave blank for unlimited

Advanced Settings >

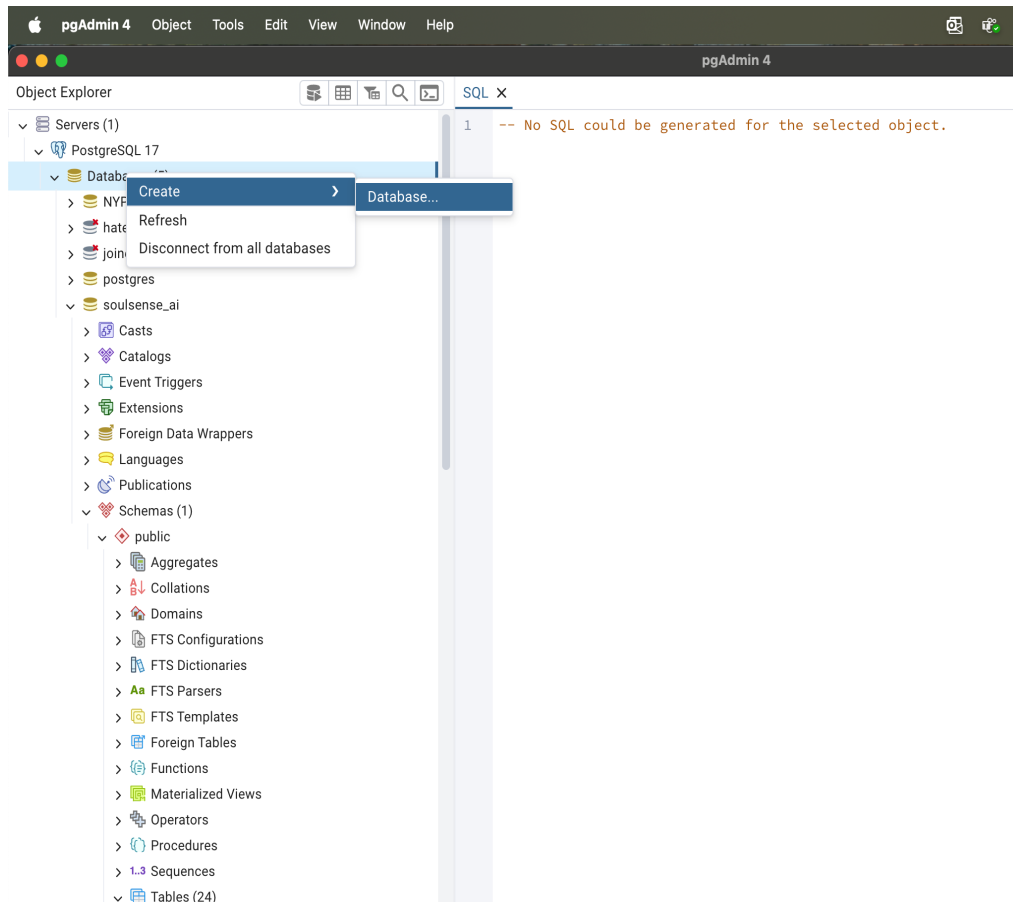
Create

Create a key and then paste it into the .env file.

7. Creating an empty local PostgreSQL database:

Launch your preferred postgres client. I have used a pgAdmin 4 where as you open the application you can see the option of server, right clicl on the server →Postggre17→Database

→ Create and then enter the name of the database named SoulSense. This step creates a new database called SoulSense which is ready for schema import.



Restore schema and seed data

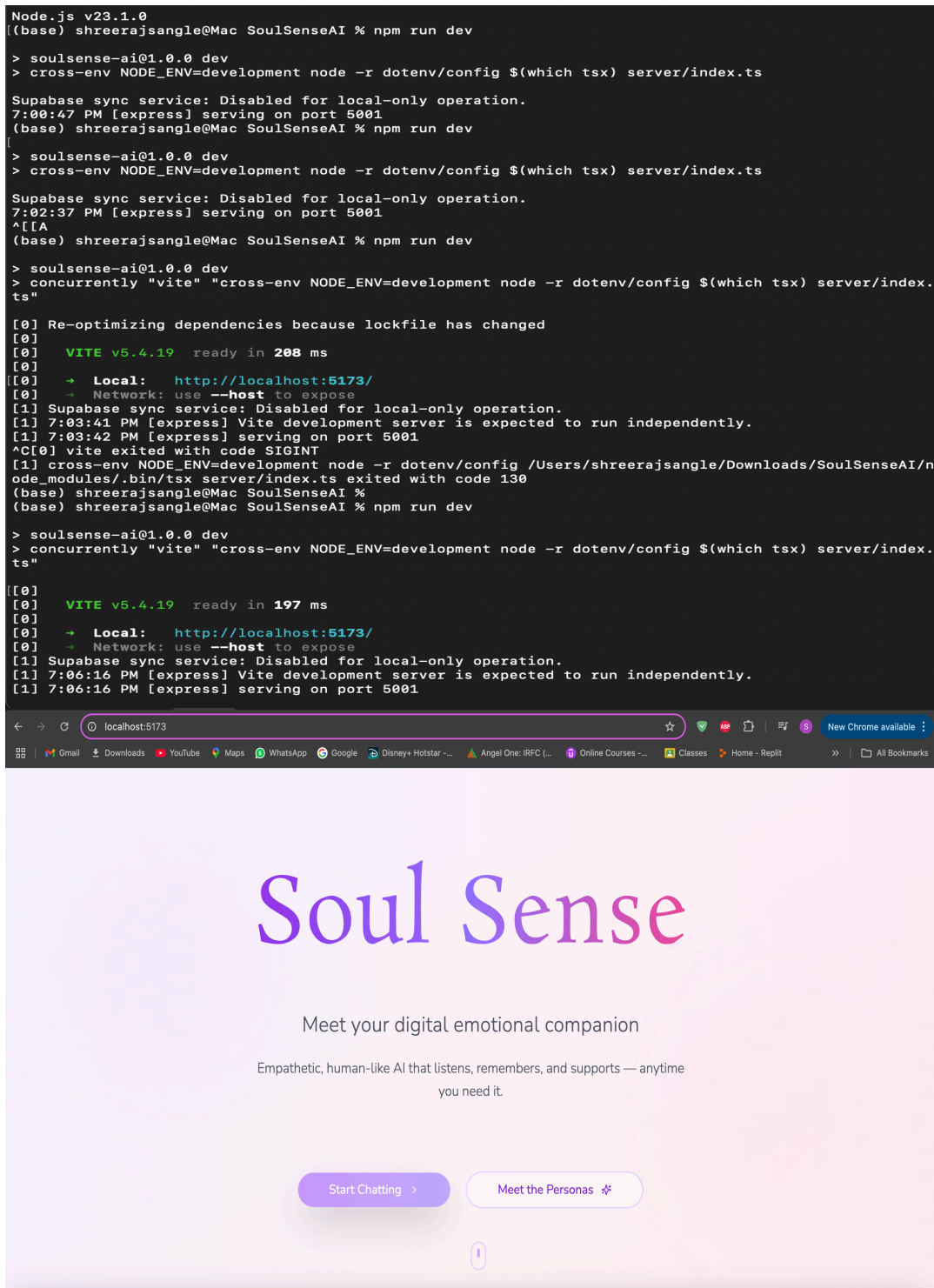
With the database selected in pgAdmin 4, choose *Restore...*, point to `COMPLETE_SOULSENSE_DATABASE_EXPORT.sql`, and run the job. The SQL dump builds every table, index, and seed record exactly as they exist in the reference environment.

A command-line alternative is:

`psql soulsense < COMPLETE_SOULSENSE_DATABASE_EXPORT.sql`. and paster the database url in the `.env` file .

8. Launch the development servers

Execute `npm run dev`. Concurrent scripts spin up Vite on <http://localhost:5173> or <http://localhost:5174> depending on condition if 5173 doesn't work, for the React UI and Express on <http://localhost:5000> for the REST API. Hot-reload is enabled, so code changes update the browser instantly. Visit the front-end URL; you should see the SoulSense landing page and watch the API heartbeat at `/api/health`.



If not open directly just click the <https://localhost:5173/> it will open only when the server is working this is if it doesn't open directly.

9. Key Configuration Parameters

Parameters	File	Default	Notes
OPENROUTER_A PI_KEY	.env	-	Obtain from https://openrouter.ai

DATABASE_URL	.env	-	postgresql://user:pwd@localhost:5432/soulsense
PORT	server/index.ts	5000	Change if port conflict
CORS_ORIGIN	server/config.ts	http://localhost:5173	Whitelist front-end origin
SESSION_TTL	server/index.ts	-	Express-session cookie life