ClarifyCoder Demo (Local): Setup & Usage Guide

September 21, 2025

Abstract

This guide documents the uploaded `clarifycoder-demo` project (FastAPI backend + Streamlit UI), with clean, unboxed code blocks and platform-specific steps. It is Overleaf-ready and does not require `--shell-escape`.

1 Project Overview

The demo exposes a FastAPI API at `/clarify` that analyzes a problem statement and optional context to propose clarification questions. A Streamlit UI calls the API and displays the questions, an ambiguity score, and extracted facts.

2 Folder Layout (from your ZIP)

```
clarifycoder-demo/
 .gitignore
 README.md
 requirements.txt

backend/
 agents.py
 main.py
 schema.py

demo_data/
 examples.json

frontend/
      streamlit_app.py
```

3 Prerequisites

- Windows, macOS, or Linux.

- Python 3.10+ installed (`python --version` or `python3 --version`).

- The ZIP extracted so you can `cd` into the project root.

4   Install Dependencies

Windows (Command Prompt)

```
cd C:\path\to\clarifycoder-demo
python -m venv .venv
.\.venv\Scripts\activate
python -m pip install --upgrade pip
pip install -r requirements.txt
```

macOS / Linux (Terminal)

```
cd ~/path/to/clarifycoder-demo
python3 -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
pip install -r requirements.txt
```

5   Run the Backend (FastAPI)

In an activated venv:

```
python -m uvicorn backend.main:app --host 127.0.0.1 --port 8000 --reload
```

Health check (browser):

```
http://127.0.0.1:8000/docs
http://127.0.0.1:8000/clarify (use POST with JSON body)
```

6   Run the Frontend (Streamlit)

Open a second terminal / command prompt, activate the venv again, then:

```
streamlit run frontend/streamlit_app.py
```

If the app asks for the API base URL, enter:

```
http://127.0.0.1:8000
```

7   API Contract

Endpoint

```
POST /clarify
```

Request

```
{
  "problem": "string",
  "context": "string or null",
  "role": "developer | pm | qa | data scientist | null"
}
```

Response (example fields)

```json
{
  "needs_clarification": true,
  "ambiguity_score": 0.42,
  "questions": [
    {
      "question": "Which authentication provider(s) should we support?",
      "rationale": "Security scope impacts architecture.",
      "category": "constraints",
      "priority": 0.92
    }
  ],
  "extracted_facts": {
    "languages": ["python"],
    "frameworks": ["fastapi", "streamlit"]
  }
}
```

8 Common Issues & Fixes

- **ModuleNotFoundError: No module named 'backend'**
  Make sure you `cd` into the project root (where `backend/` lives) before running Uvicorn.

- **No module named 'uvicorn' or 'fastapi'**
  Dependencies not installed in the venv: run `pip install -r requirements.txt` after activating the venv.

- **Address already in use**
  Change the port: `--port 9000` and point Streamlit to `http://127.0.0.1:9000`.

- **Streamlit can't reach API**
  Ensure the API window shows Uvicorn running; confirm the URL in the Streamlit config expander.

9 Stopping & Cleanup

- Stop API/UI: `Ctrl + C` in each terminal.

- Deactivate venv: `deactivate`.

- Remove venv: delete the `.venv` folder.