All set up instruction can be found on GitHub page:

https://github.com/MikeLiang2/Main-Capstone

Capstone Project

A fullstack application with a Python FastAPI backend and a Vue 3 + Vite frontend.

Requirements

- **Python**: >= 3.13
- **Node.js**: >= 22.17
- **npm**: (comes with Node.js)
- **Docker** and **Docker Compose** (for containerized setup)

🐋 Run the Project with Docker

Quick Start

If you're new to Docker, the easiest way to run the entire project is to execute the startup script:

```
./start.sh
```

This script handles building, migrating, and starting all the necessary containers.

Manual Docker Compose Commands

First Time Setup

Before running the containers manually, make sure
the backend/alembic/versions directory exists. If it doesn't, create it:

```
mkdir -p backend/alembic/versions
```

Also, make sure Docker and Docker Compose are installed and running on your machine.
Then, from the project root directory, run:

```
docker compose up --build
```

This will:

- Build and start the backend FastAPI server on http://localhost:8000

- Build and start the frontend Vue app served via nginx on http://localhost:5173
- Start a PostgreSQL database container
- Start the database migration service to apply schema changes

Subsequent Runs (After Initial Build)

To start the containers without rebuilding, simply run:

```
docker compose up
```

Stop the containers

```
docker compose down
```

Remove the containers

```
docker compose down -v
```

🔧 Setup Instructions (Only if without Docker)

1. Backend Setup (Local development)

```
cd backend
python -m venv venv
# Activate virtual environment:
# Windows:
venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate
pip install -r requirements.txt
```

2. Frontend Setup (Local development)

```
cd ../frontend
npm install
```

▶️ Run the Project Locally (without Docker)

1. Start Backend Server

In a new terminal:

```
cd backend
```

```
# Activate virtual environment:
# Windows:
venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate
uvicorn app.main:app --reload
```

2. Start Frontend Dev Server

In another terminal:

```
cd frontend
npm run dev
```

📁 Project Structure

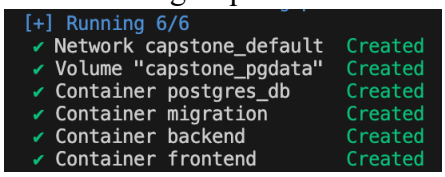
```
Capstone/
├── backend/
│   ├── app/
│   ├── Dockerfile
│   ├── requirements.txt
│   └── ...
├── frontend/
│   ├── Dockerfile
│   ├── package.json
│   ├── src/
│   └── ...
├── dbm.sql
├── docker-compose.yml
├── package.json
└── README.md
```

📝 Notes

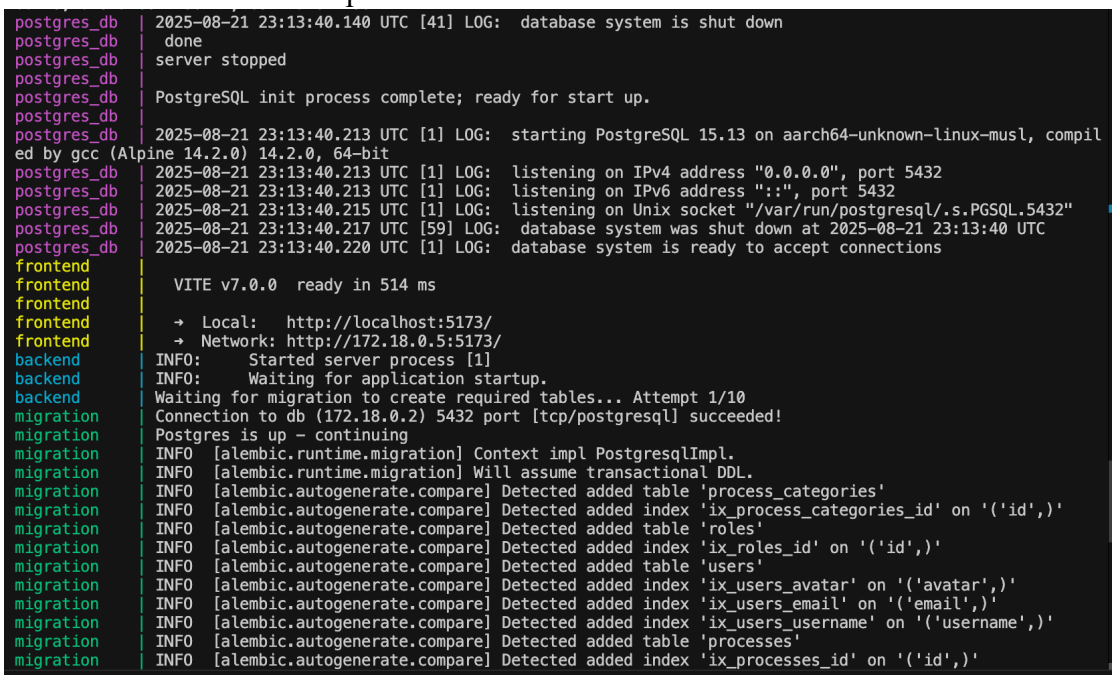
- The backend runs on http://localhost:8000
- The frontend runs on http://localhost:5173

Validation (how to confirm the setup is successful).

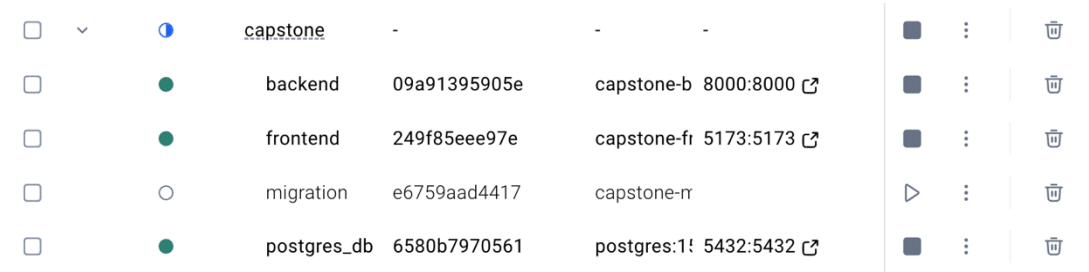
All the deployment status will be reflected in the terminal. For detail check the terminal status.
For instance:
The following steps monitor the container build status in Docker.



Other start status will also posted in the terminal in different colors.



The services status (backend, frontend, database) can be viewed in Docker.



Local deployment is easy, one lines of command should complete everything.

For cloud deployment:
Since different cloud service providers have different service policy and compacity, right now, auto deployment feature is only supported by Microsoft Azure.
For auto deployment on Microsoft Azure, run:

```
./deploy_azure.sh
```