## 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)



# **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 <a href="http://localhost:5173">http://localhost:5173</a>
- 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

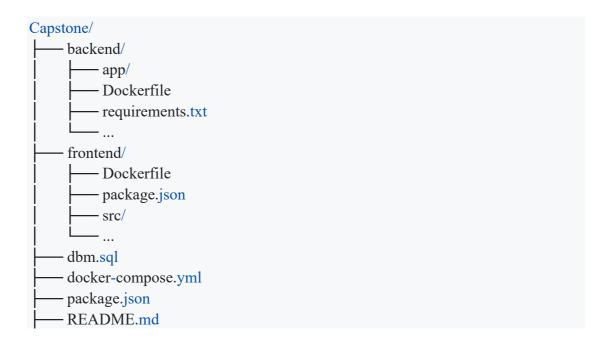
uvicorn app.main:app --reload
```

### 2. Start Frontend Dev Server

In another terminal:

cd frontend npm run dev

# Project Structure



# Notes

- The backend runs on <a href="http://localhost:8000">http://localhost:8000</a>
- The frontend runs on <a href="http://localhost:5173">http://localhost:5173</a>

#### 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.

```
Network capstone_default Created
Volume "capstone_pgdata"
Container postgres_db
Container migration
Container backend
Container frontend
```

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

```
2025-08-21 23:13:40.140 UTC [41] LOG: database system is shut down
                                                                                                   server stopped
                                                                                                   PostgreSQL init process complete; ready for start up.
          | Stgres_db | 2025-08-21 23:13:40.213 UTC [1] LOG: | starting PostgreSQL 15.13 on aarch64-unknown-linux-musl, compiled by gcc (Alpine 14.2.0) 14.2.0, 64-bit | stgres_db | 2025-08-21 23:13:40.213 UTC [1] LOG: | listening on IPv4 address "0.0.0.0", port 5432 | stgres_db | 2025-08-21 23:13:40.213 UTC [1] LOG: | listening on IPv6 address ":", port 5432 | stgres_db | 2025-08-21 23:13:40.215 UTC [1] LOG: | listening on Unix socket "var/run/postgresql/.s.PGSQL.5432" | stgres_db | 2025-08-21 23:13:40.217 UTC [59] LOG: | database system was shut down at 2025-08-21 23:13:40 UTC | database system is ready to accept connections
                                                                                  VITE v7.0.0 ready in 514 ms

- Local: http://localhost:5173/
- Network: http://172.18.0.5:5173/
INFO: Started server process [1]
INFO: Waiting for application startup.
Waiting for migration to create required tables... Attempt 1/10
Connection to db (172.18.0.2) 5432 port [tcp/postgresql] succeeded!
Postgres is up - continuing
INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
INFO [alembic.runtime.migration] Will assume transactional DDL.
INFO [alembic.autogenerate.compare] Detected added table 'process_categories'
INFO [alembic.autogenerate.compare] Detected added table 'roles'
INFO [alembic.autogenerate.compare] Detected added table 'roles'
INFO [alembic.autogenerate.compare] Detected added index 'ix_roles_id' on '('id',)'
INFO [alembic.autogenerate.compare] Detected added index 'ix_users_avatar' on '('avatar',)'
INFO [alembic.autogenerate.compare] Detected added index 'ix_users_avatar' on '('avatar',)'
INFO [alembic.autogenerate.compare] Detected added index 'ix_users_username' on '('username',)'
INFO [alembic.autogenerate.compare] Detected added index 'ix_users_username' on '('id',)'
backend
```

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
```