# Deployment Playbook

## Back end Setup

1. Clone the repo <https://github.com/ewfx/gaipl-train-test-split/tree/main> and get into folder ‘gaipl-train-test-split/tree/main/code/src’
2. Run the following commands to install dependencies:

* install python
* install pip- python -m pip install --upgrade pi
* install uvicorn  - pip install uvicorn
* pip install crewai\_tools
* pip install fastapi
* pip install jsons
* pip install -U langchain
* pip install -U langchain langchain-community
* pip install langchain-google-genai
* pip install langchain-cohere
* pip install chromadb
* pip install autogen
* pip install tensorflow
* pip install transformers
* pip install tf-keras
* pip install flask\_socketio

1. We need to run the service class as its main method generates the embeddings for the incident data. So run the command “py .\incqueryservice.py”. This will be a one time task which might need to be periodically re-run to refresh the incident data.
2. Now, start the server using command python –m uvicorn:main app –reload

## Front end setup

1. The only requirement is having Node.js & npm installed - [install with nvm](https://github.com/nvm-sh/nvm#installing-and-updating)
2. Install the necessary dependencies. - npm i
3. Run front end server using command ‘npm run dev’

## Health Check Module

1. Install Python 3.11 and set up the virtual environment.
2. Activate the virtual environment and install dependencies using the requirements.txt file.
3. Get inside folder ‘gaipl-train-test-split/tree/main/code/src’
4. Start the Flask application locally by running the appropriate Python script or by using docker.
5. Start Prometheus using the provided configuration file.
6. If using Docker, build and start all services with Docker Compose.
   1. docker-compose build
   2. docker-compose up
7. Run the health check polling script using ‘python health\_api.py’
8. Access the Flask application through the designated localhost URL.
9. Access Prometheus through its assigned localhost port.