

# Project Galapagos

A Python web application built with Flask, Neo4j, MongoDB, and GraphQL.

## Overview

This app serves HTML views via REST endpoints and provides a public API using GraphQL. It leverages both Neo4j and MongoDB for data storage, each optimized for specific use cases.

---

## Tech Stack

Component	Technology
Backend	Python, Flask
Database	Neo4j, MongoDB
API	GraphQL (Public API)
Views	REST (HTML)

---

## Project Structure

```
Galapagos/
  app/                      # Main application directory
    templates/                # HTML templates for REST views
    static/                   # Static files (CSS, JS, images)
    resolvers/                # GraphQL resolvers
    schemas/                 # GraphQL schemas
    models/                   # Data models and database interactions
      Mongo/                  # MongoDB models
      Neo4j/                  # Neo4j models
    services/                 # Business logic (including route optimization)
    utils/                    # Utility functions
    data/                     # Data migration scripts
    Dockerfile                # Dockerfile to build the app
    app.py                    # App entrypoint (includes Flask routes & GraphQL setup)
    requirements.txt          # Python dependencies
  Sujet.pdf                  # Project requirements document
  compose.yml                # Docker compose file
  README.md                  # Project documentation
```

---

## Data Storage

### Neo4j

**Data:** - Islands - Ports - Lockers - Warehouse - Clients - Seaplanes type - Seaplanes

### MongoDB

**Data:** - Seaplanes cargo hold - Lockers cargo - Warehouse cargo - Scientific equipment - Orders - Deliveries

---

## Getting started

install the libraries with `pip install -r ./app/requirements.txt`

start the project with `docker compose up -d` run migrations with `docker compose exec web flask migrate` run `docker compose up -d --build web` to rebuild the python web app

Frontend GUI is exposed on `http://localhost:5000/` GraphQL playtest interface is exposed on `http://localhost:5000/graphql` Mongo express interface is exposed on `http://localhost:8081/db/galapagos/` Neo4j admin interface is exposed on `http://localhost:7474/browser/`