Early Care Gateway

Execution Instructions

Contents

1 Prerequisites 2

2 Getting Started (Step-by-Step) 2

1. Clone the Repository 2
2. Configure Environment Variables 2
3. Run the System 3
4. Use the System 4

3 Troubleshooting 4

1 Prerequisites

Prior to running the project, ensure that your system meets the following software and hardware requirements:

- **Python 3**: Generate the JWT secret key
- **Pip**: Install testing dependencies
- **Docker (and WSL for Windows)**: Build containers
- **Docker Compose V2**: Deploy the multi-container system
- **Git**: Clone the repository
- **At least 2 GB of Disk Space**: Host AI models and system components
- **Recommended at least 8 GB RAM**: Ensure stable execution

Verify that the tools are correctly installed:

```
git --version
python --version
pip --version
wsl --version    # Windows only
docker --version
docker compose version
```

2 Getting Started (Step-by-Step)

1. Clone the Repository

```
git clone https://github.com/carrieroroberto/
   ↪ early_care_gateway
cd early_care_gateway
```

2. Configure Environment Variables

The file `.env.example` serves as a template for the environment configuration required by the system. Rename it to `.env`, or create a new `.env` file and fill it with the correct values for your setup (e.g. database user and password).

Generate Keys

If Python is installed on your machine, you can generate a secure SHA256 secret key using:

```
python secret_key_generator.py
```

Copy the generated key and paste it into your `.env` file under `SECRET_KEY`.

If you already have a Gemini API key, place it under GOOGLE_API_KEY. Otherwise, you can obtain a free API key by visiting Google AI Studio. After logging in and creating a project, you will be able to generate your personal API key.

Example .env Configuration

Your .env file should look similar to the following:

```
# PostgreSQL database connection settings
POSTGRES_HOST = postgres
POSTGRES_USER = earlycaregateway
POSTGRES_PASSWORD = your_password
POSTGRES_DB = earlycaregateway
POSTGRES_PORT = 5432

# pgAdmin credentials
PGADMIN_EMAIL = admin@earlycaregateway.com
PGADMIN_PASSWORD = your_password

# Application secret key
SECRET_KEY = your_secret_key

# External API keys
GOOGLE_API_KEY = your_api_key

# Microservices URLs (internal Docker network addresses)
GATEWAY_URL = http://gateway:8000/gateway
AUTHENTICATION_URL = http://auth_service:8000/authentication
DATA_PROCESSING_URL = http://data_service:8000/
   ↪ data_processing
EXPLAINABLE_AI_URL = http://xai_service:8000/explainable_ai
AUDIT_URL = http://audit_service:8000/audit

# Frontend application settings
REACT_APP_API_URL = http://localhost:8002/gateway
```

3. Run the System

You can start the system using the run.bat (Windows) or run.sh (macOS/Linux) scripts, that automatically launch the services and run automated integration tests.

Alternatively, you may manually start the system with:

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

This command will:

- Build all Docker images (Backend, Frontend, Database)
- Start the containers

- Connect services through networks and volumes

4. Use the System

Access the web application at:

`http://localhost:3000`

You may also use an API client, such as Postman, to interact directly with the API endpoints following the provided documentation.

3 Troubleshooting

- Monitor logs to verify that all services start correctly:

  ```
  docker compose logs -f
  ```

- Ensure no other applications are using ports required by the system.
- If Docker images fail to build, try stopping and removing existing containers and volumes using:

  ```
  docker compose down -v
  ```

- On Linux systems, Docker commands may require `sudo`.
- Access pgAdmin at `http://localhost:8080` using the credentials provided in your `.env` file.