Deployment Guide

This guide provides step-by-step instructions to deploy the project in Window 10/11 operating system.

Prerequisites

1. Install Python with necessary libraries

Download and install Python 3.8 or higher from python.org.

Verify installation:

```
python --version
```

In terminal run the following command:

```
#Run at project root
pip install -r requirements.txt
# OR
python -m pip install -r requirements.txt
```

2. Install Docker Desktop

Download and install Docker Desktop from docker.com.

Verify installation:

```
docker --version
docker-compose --version
```

Configuration

3. Create Environment Files

Create three `.env` files with the following content:

3.1 Backend `.env`

Create `backend/.env` with the same content as above:

```
DATA_ENCRYPTION_KEY=c29a02b23662ced73f8c007c877a85c8aab576b1b7f888ac37c364b5a75a681b
```

3.2 Percona Compose `.env`

Create `percona-compose/.env`:

```
MYSQL_ROOT_PASSWORD=supersecurepassword
MYSQL_DATABASE=ComputingU
MYSQL_USER=myuser
MYSQL_PASSWORD=myuserpassword
DATA_ENCRYPTION_KEY=c29a02b23662ced73f8c007c877a85c8aab576b1b7f888ac37c364b5a75a681b
```

3.4 Set encryption key for AES encryption/decryption

In `load_sql/university.sql`, replace the following code at line 4

```sql
SET @encryption_key = 'c29a02b23662ced73f8c007c877a85c8aab576b1b7f888ac37c364b5a75a681b';
```

Database Setup

4. Start Docker Container

Open a terminal and navigate to the `percona-compose` directory:

```
cd <project_root>/percona-compose
docker-compose up -d
```

Wait for the container to start. Verify it's running:

```
docker ps
```

5. Initialize Database

Connect to MySQL:

```
docker exec -it percona-server mysql -u root -p
```

When prompted, enter the password: `supersecurepassword`

6. Setup database

Enter MySQL server,
Set the encryption key first:

```
SET @encryption_key =
'c29a02b23662ced73f8c007c877a85c8aab576b1b7f888ac37c364b5a75a681b';
```

Then, copy and paste the entire content of `<project_root>/load_sql/University.sql` into the MySQL prompt, then press Enter.

7. Verify Database Setup

In the MySQL prompt, verify tables were created:

```
USE ComputingU;
SHOW TABLES;
```

You should see tables like `students`, `guardians`, `staffs`, `courses`, `grades`, `disciplinary_records`, `accountLog`, `dataUpdateLog`, `audit_log`, and `sessions`.

Exit MySQL:

```
EXIT;
```

Backend Setup

8. Setting up the certificate for the application

Open PowerShell as **Administrator Mode !!!!**. Press Win + X, choose Windows PowerShell (Admin).

Enter `<project_root>`. Install Chocolatey (one-time setup)

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

Install mkcert via Chocolatey

```
choco install mkcert
```

Trust mkcert’s local CA

```
mkcert -install
```

Open a new terminal and change directory to `<project_root>/security` ,generate development certificates

```
mkcert -cert-file localhost-cert.pem -key-file localhost-key.pem localhost
127.0.0.1 ::1
```

replace the cert.pem and key.pem with the content in localhost-cert.pem and localhost-key.pem correspondingly

Troubleshooting: Access denined on `mkcert -install`

1(a): Try to grant administration permission (PolyU machines).
1(b): If you cannot install local cert, you need to manual trust the cert and covert the import the self-signed cert.

9. Start Backend Server

Navigate to the backend directory and run:

```
cd <project-root>/backend
python main.py
```

The server will start on `http://127.0.0.1:8000` .

> Make sure you have installed requirements.txt

Frontend Access

11. Open Frontend

Open `frontend/index.html` in your web browser.

You can:

- Double-click the file to open it in your default browser
- Or manually navigate to the file location and open it

Test Users

Use these credentials to test the system:

- **Student**: `test_student@example.com` / `StudentTest123`
- **Guardian**: `test_guardian@example.com` / `GuardianTest123`
- **Staff**: `test_staff@example.com` / `StaffTest123`

Troubleshooting

- **Database connection error**: Ensure Docker container is running (`docker ps`)
- **Port 3306 already in use**: Stop other MySQL instances or change the port in `docker-compose.yml`
- **Encryption key error**: Ensure `.env` , `university.sql` files exist with `DATA_ENCRYPTION_KEY` set
- **Module not found**: Run `pip install -r requirements.txt` again

Stopping the System

To stop the Docker container:

```
cd percona-compose
docker-compose down
```

To stop the backend server, press `Ctrl+C` in the terminal running `main.py` .