Installation and Running Guide

Running the project (Front-end offline, back-end online)

1. Install Node.js and npm

1.1 For Windows and macOS

1. Visit the official Node.js website.
2. Download the latest stable version of Node.js suitable for your operating system.
3. Execute the downloaded installer and follow the installation guide.
4. After the installation, open the command prompt or terminal and input the following commands to verify the installation:

```
node -v
npm -v
```

1.2 For Linux (Debian and Ubuntu as examples)

```
sudo apt update
sudo apt install nodejs npm
```

Verify the installation:

```
node -v
npm -v
```

2. Run React Project

Download all necessary packages:

```
npm install
```

Navigate to the project directory and start it:

```
cd ui-test
npm run start
```

(Make sure the API_BASE_URL in the config.js file is "https://perksummit.club:5000")

Running the project without docker (Both front-end and back-end, both offline)

1. Install Node.js and npm

1.1 For Windows and macOS

1. Visit the official Node.js website.
2. Download the latest stable version of Node.js suitable for your operating system.
3. Execute the downloaded installer and follow the installation guide.
4. After the installation, open the command prompt or terminal and input the following commands to verify the installation:

```
node -v
npm -v
```

1.2 For Linux (Debian and Ubuntu as examples)

```
sudo apt update
sudo apt install nodejs npm
```

Verify the installation:

```
node -v
npm -v
```

2. Install Python and pip

2.1 For Windows

1. Visit the official Python website.
2. Download the latest stable version of Python suitable for your OS.
3. Run the downloaded installer. Ensure you select the option "Add Python to PATH" or a similar option during the installation process.
4. After the installation, open the command prompt and verify with:

```
python --version
pip --version
```

2.2 For macOS

macOS typically comes preinstalled with Python, but it might be Python 2. To install Python 3 and pip, it's recommended to use Homebrew:

```
brew install python3
```

Verify the installation:

```
python3 --version
pip3 --version
```

2.3 For Linux (Debian and Ubuntu as examples)

```
sudo apt update
sudo apt install python3 python3-pip
```

Verify the installation:

```
python3 --version
pip3 --version
```

3. Run React Project

Navigate to the project directory and start it:

```
cd ui-test
npm run start
```

Make sure the API_BASE_URL in the config.js file is "http://localhost:5000"

4. Run Flask Project

4.1 Install dependencies

Before running the Flask project, make sure to install all required packages listed in `requirements.txt`.

Navigate to the backend directory:

```
cd backend
```

Install the dependencies:

```
pip install -r requirements.txt
```

4.2 Run the Flask application

Now, run the Python script:

```
python run.py
```

5. Configure MySQL Database

5.1 Install MySQL:

For Ubuntu:

```
sudo apt-get update
sudo apt-get install mysql-server
```

For CentOS:

```
sudo yum install mysql-server
sudo systemctl start mysqld
```

For Windows:

Please visit the MySQL Official Download Page and download the appropriate installer for your system version, then follow the prompts to install.

5.2 Configure MySQL Security:

For Linux Systems:

Run MySQL's security script for initial security configuration.

```
sudo mysql_secure_installation
```

If you haven't set a MySQL password before, set it as `chen526` during this step. If you prefer a different password, remember the one you set and replace `chen526` with your MySQL password in the `config.py` file.

For Windows:

During the installation process, you will usually be guided through the security setup.

5.3 Log in to MySQL and Set Up the Database:

For Linux Systems:

Use the following command to log into MySQL:

```
mysql -u root -p
```

After entering your password, you'll be in MySQL's command-line mode.

For Windows:

Log in using MySQL Workbench or another MySQL client tool of your choice.

Then, create a new database named `UI_test`:

```
CREATE DATABASE UI_test;
```

To exit the MySQL command line:

```
exit;
```

5.4 Import SQL Data:

Import your `init.sql` file into the `UI_test` database you just created.

For Linux Systems:

```
mysql -u root -p UI_test < init.sql
```

For Windows:

You can use phpMyAdmin to accomplish this task.

Note:

This SQLAlchemy connection is as follows:

```
SQLALCHEMY_DATABASE_URI = 'mysql+pymysql://root:chen526@localhost/UI_test'
```

If necessary, make modifications in the `config.py` file of the Python project.

Running the project with docker

Deploying a Python Flask Application Using Docker and Docker Compose

In this guide, we will walk you through deploying a Python Flask application using Docker and Docker Compose.

1. Install Docker:

- If you haven't installed Docker yet, visit the Docker Official Website and follow the installation guide suitable for your operating system.

2. Install Docker Compose:

- Docker Compose lets you define and run multi-container applications using a `docker-compose.yml` file. Head over to the Official Installation Guide for Docker Compose and follow the steps.

3. Build and Launch Containers:

- Open your terminal or command prompt and navigate to the project directory that has the `Dockerfile` and `docker-compose.yml`.
- Execute the command `docker-compose build` to build your web service.
- Follow it up with `docker-compose up` to start your web and db services.

4. Access the Application:

- Launch your web browser and navigate to http://localhost:5000. You should now see your Python Flask application up and running.

5. Shutdown Services:

- When you're done and wish to shut down the services, use the command `docker-compose down` to stop all associated services.

Building a Docker Image of front end project

From the root directory of the project, execute the following command to build a Docker image:

```
docker build -t UI_test .
```

This will construct a Docker image named `UI_test` based on your `Dockerfile`.

Running the Docker Container

To run your application, use the command below:

```
docker run -p 3000:3000 UI_test
```

This initiates a new Docker container instance based on the `UI_test` image. The `-p 3000:3000` flag maps the container's port 3000 to port 3000 on the host machine.

Accessing the Application

Visit http://localhost:3000 in your browser, and you should see your application running.