

Table 1: Revision History

Date	Developer(s)	Change
April 1, 2023	Alyssa	Addition of Sayyara demo link, how to run source code

User Guide Sayyara

Team 31
SFWRENG 4G06
Christopher Andrade
Alyssa Tunney
Kai Zhu
Ethan Vince-Budan
Collin Kan
Harsh Gupta
April 1, 2023

1 Using the Sayyara Application

To use the various features on the Sayyara website, a video demo has been created to demonstrate the flow of a customer creating an account and signing in, and following the process to look up shops, and then create a quote request to send to the desired shops. There is also the view of the shop owner/employee when a customer sends them a quote request, and how the flow resumes from their point of view. The video demo can be found [here on YouTube](#).

2 Running the Sayyara Application

The Sayyara application currently runs locally due to deployment not being part of the project's MVP. In order to run the source code, the user needs to have the [Docker](#) application installed on their machine. Once Docker is installed, a terminal can be opened and navigated to the `src\` directory of this repository where the `docker-restart.sh` script is located. After running this script, the Docker container should be populated as such:

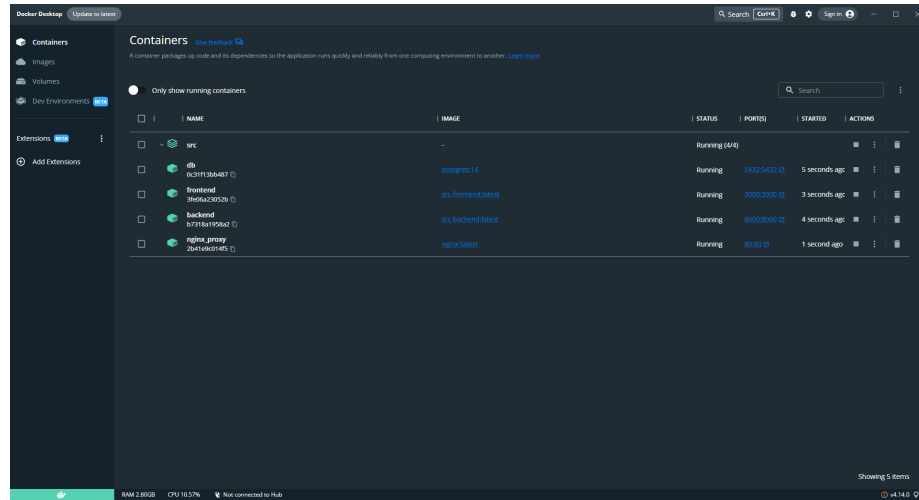
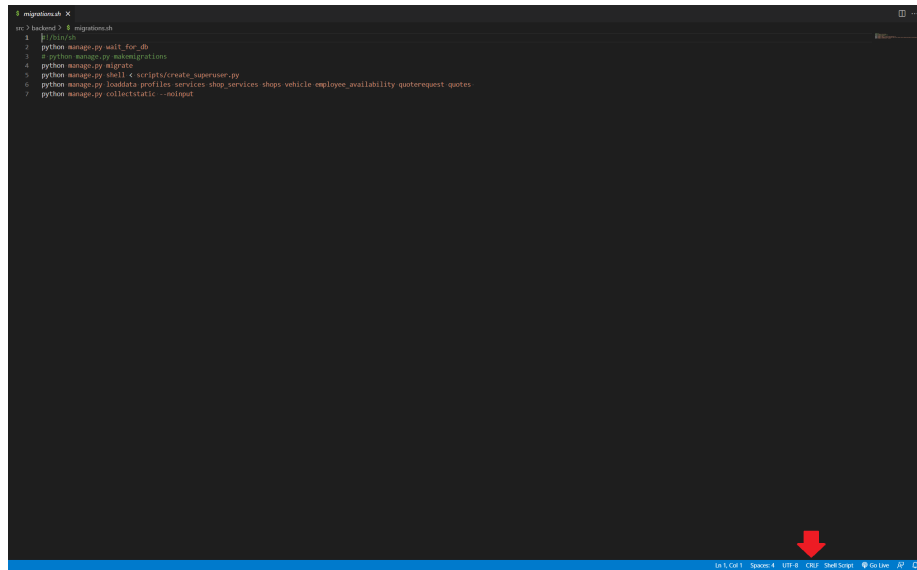


Figure 1: The Docker container housing the Sayyara application

Once running, the frontend can be found at <http://localhost/>. The backend APIs can be viewed at <http://localhost/api/swagger/>, and the related admin panel viewed at <http://localhost/api/admin/> (Username: admin; Password: admin). Any additional information can be found in the README file located in `src\`.

2.1 Common Issues

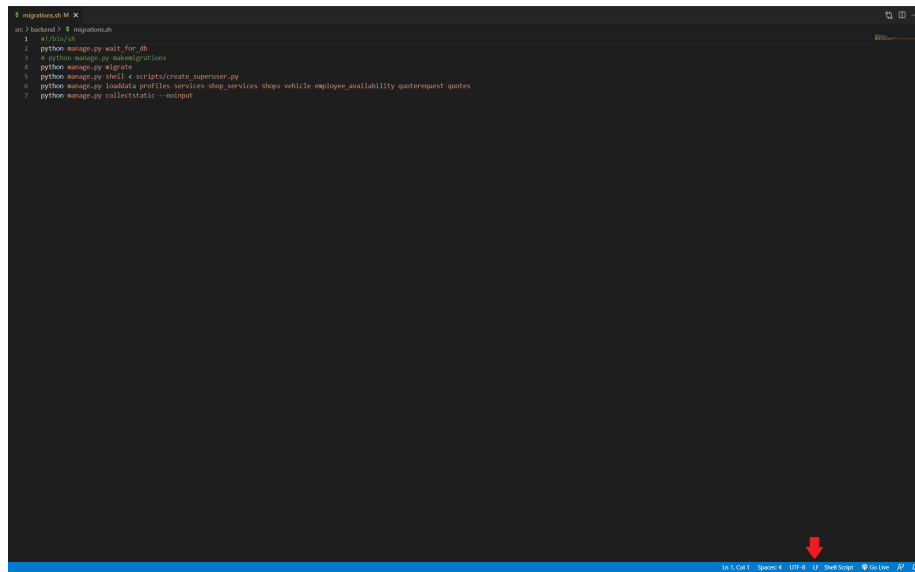
An issue that some of the team has run into when running the Docker container is that the `migrations.sh` script, located in `src\backend\` has compatibility issues due to being in CRLF mode. If VSCode is used, this can easily be modified to LF mode, which allows the backend in the Docker container to run properly.



```
migrations.sh X
src\backend> . migrations.sh
1. python manage.py migrate
2. python manage.py wait_for_db
3. python manage.py seedmigrations
4. python manage.py migrate
5. python manage.py shell < scripts/create_superuser.py
6. python manage.py loaddata profile services shop_services shops vehicle employee_availability questionnaire quotes
7. python manage.py collectstatic --output
```

VS Code status bar: In 1 Col 1 | Search 4 | 100.0% | CRLF | Shell Script | Python 3.8

Figure 2: migrations.sh running in CLRF



```
migrations.sh M X
%%> backend > ! migrations.sh
1 #!/bin/sh
2 python manage.py wait_for_db
3 # python manage.py makemigrations
4 python manage.py migrate
5 python manage.py shell < scripts/create_superuser.py
6 python manage.py loaddata profile.service.shop.services shops vehicle_employee_availability.quoterequest.quotes
7 python manage.py collectstatic --noinput
```

Figure 3: migrations.sh running in LF