Deliverable 4 – Project Phase 2 Report

FuelXpress

Team: Hackelite

Team Members:

Sahithi Regalla

Uday Vandanapu

Karthik Setti

Pranay Pendyala

Mounika Eddala

Kavya Jangapalli

Durga Devi Pampani

Hemanth Janapala

Vaishnavi Adapa

INDEX

1) Requirements:

a. Order Confirmation

Description:

- After submitting an order, users can conveniently review the order confirmation on the user interface page. However, to finalize the confirmation, administrative approval is necessary.
- Users are required to provide specific details such as the quantity of fuel, the delivery address, and the type of fuel needed during the ordering process.
- The confirmation page provides a comprehensive overview of the order, detailing specifics such as the ordered items and associated information.
- Users' orders will be confirmed upon approval by the administration, ensuring accuracy and adherence to user specifications.

b. Order History

Description:

- Users are granted access to a comprehensive order history feature, presenting a detailed record of their prior transactions.
- This history includes specific details like the type of fuel, quantity ordered, and payment information, offering users a thorough and organized overview of their transaction history.

c. Gas Station

Description:

- This functionality enables to oversee and manage incoming fuel orders effectively. Also receive alerts for new orders, allowing them to have control over the process.
- They can access comprehensive order details, such as delivery addresses, fuel types, and requested quantities.
- In cases of insufficient gasoline availability, users have the option to cancel orders. Additionally, the system provides suggestions for the nearest alternative gas station, enhancing user convenience.
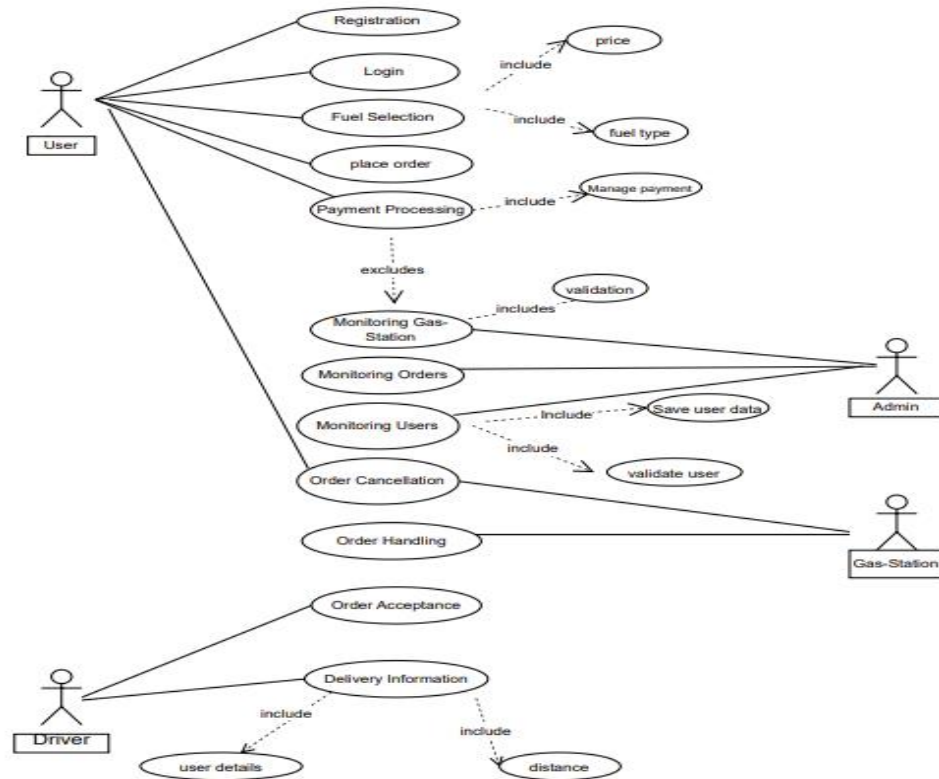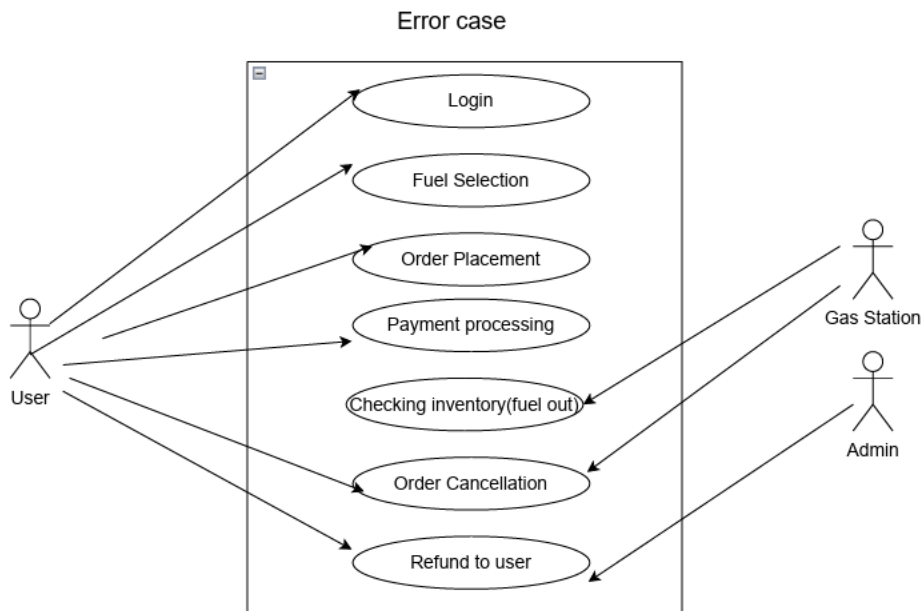
d. Driver Dashboard

Description:

- The Drivers Page serves as a centralized hub for drivers, streamlining the process of accepting incoming orders and organizing them in a queue for efficient service.
- It's noteworthy that drivers are registered exclusively by administrators.
- Once registered, drivers gain access to complete delivery information, encompassing the client's location, client data, and the vehicle identification number.
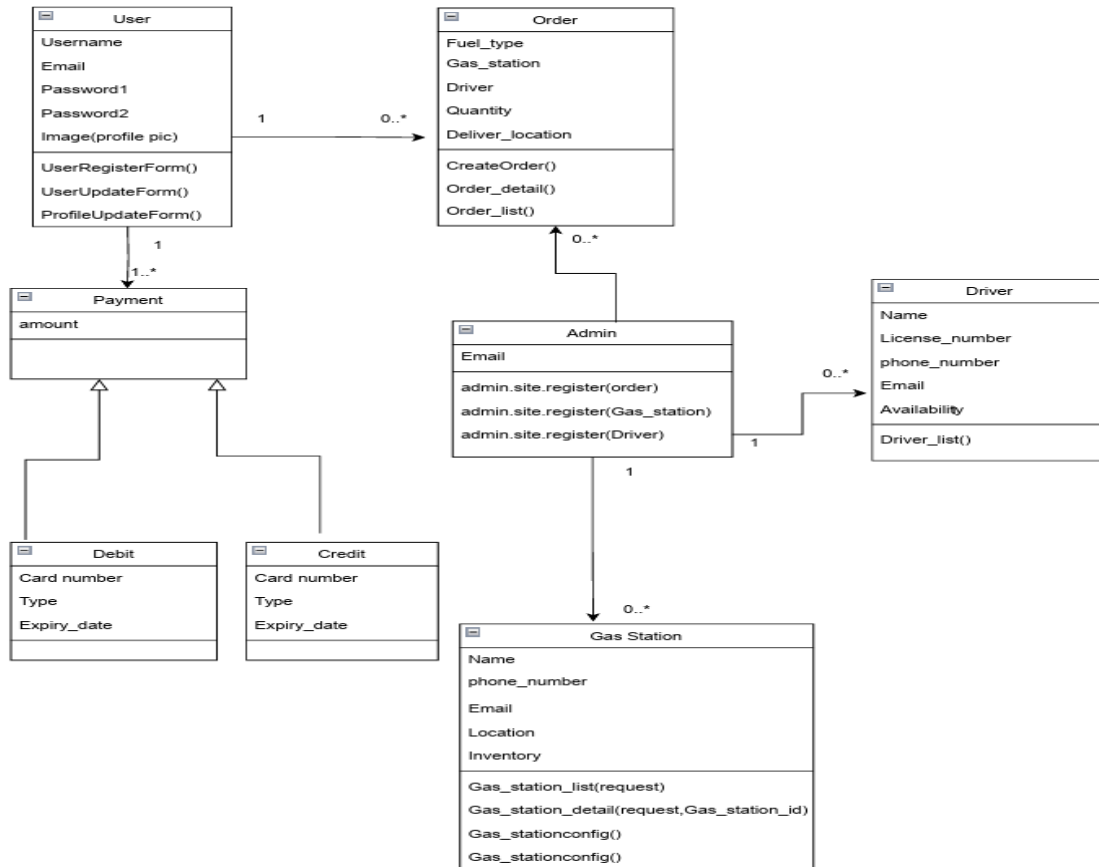- This ensures a smooth and organized delivery process

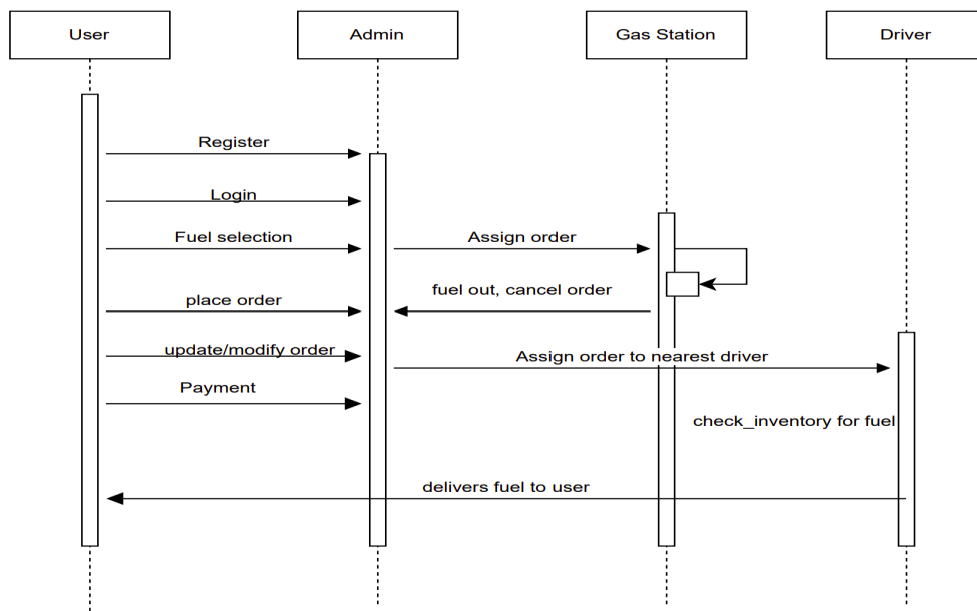## 2) UML Diagrams:

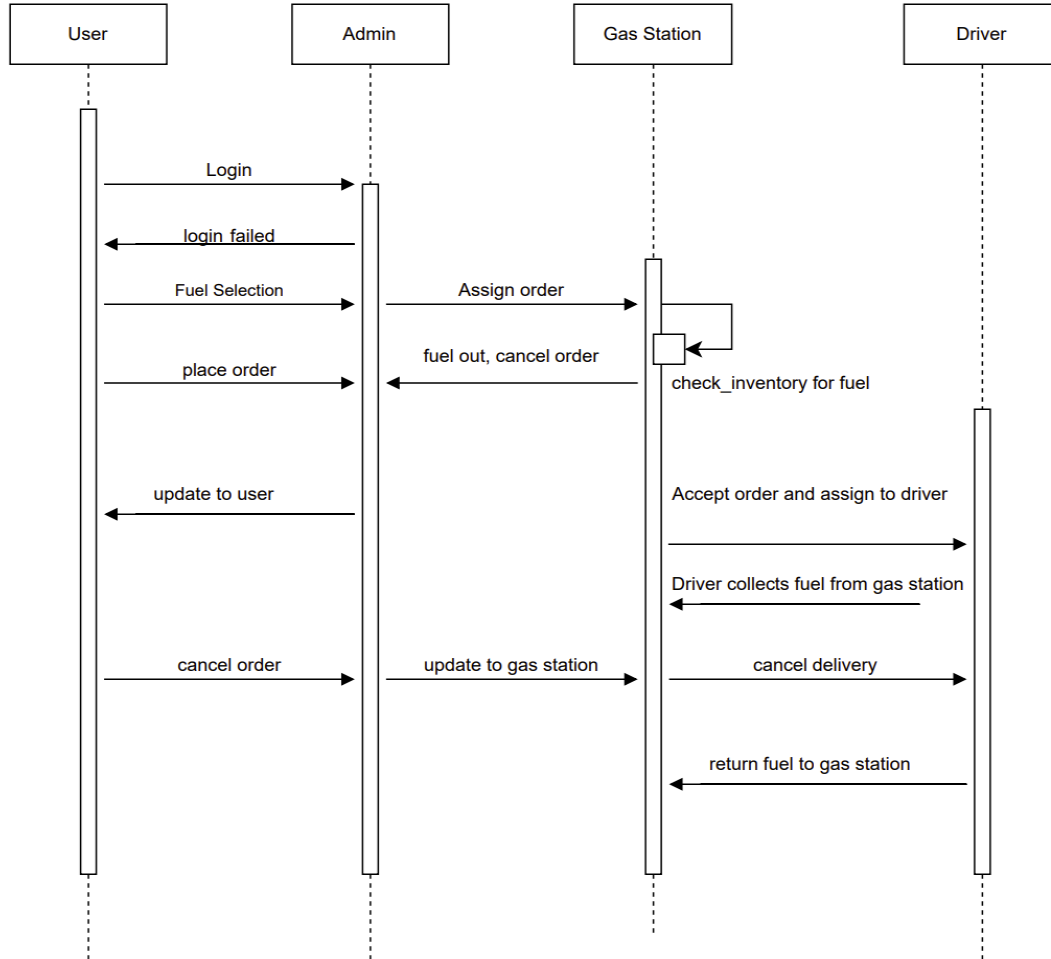### a) Use Case Diagram

Use Case Diagram (Error Case)

b) Class Diagram

User
- Username
- Email
- Password1
- Password2
- Image(profile pic)

- UserRegisterForm()
- UserUpdateForm()
- ProfileUpdateForm()

Order
- Fuel_type
- Gas_station
- Driver
- Quantity
- Deliver_location

- CreateOrder()
- Order_detail()
- Order_list()

Payment
- amount

Admin
- Email

- admin.site.register(order)
- admin.site.register(Gas_station)
- admin.site.register(Driver)

Driver
- Name
- License_number
- phone_number
- Email
- Availability

- Driver_list()

Debit
- Card number
- Type
- Expiry_date

Credit
- Card number
- Type
- Expiry_date

Gas Station
- Name
- phone_number
- Email
- Location
- Inventory

- Gas_station_list(request)
- Gas_station_detail(request,Gas_station_id)
- Gas_stationconfig()
- Gas_stationconfig()

c) Sequence Diagram

5

Sequence Diagram (Error case)

3)Test Cases:

For Phase-2 implementation, we have created the following tests.

Test Case ID	Test Module Name	Test Scenario	Test Data	Test Case Steps	Expected Results
TC001	Order Creation	Verify a new order is Created		a. Open the application. b. Dashboard is displayed. c. Click on Add Orders d. Displays Order form. e. Select from drop User (required field with alphabets), Gas Station, Driver f. Enter Fuel type, Quantity and Location (all fields need to be filled to create an order.) g. Auto updates Order Status and Order date (fills date and time	User should be able get success message on order creation and is able to view order in the list.

TC002	Order Creation	Verify one or more Orders are added		a. Open the application. b. Dashboard is displayed. c. Click on Add Orders d. Displays Order form. e. Select from drop User (required field with alphabets), Gas Station, Driver f. Enter Fuel type, Quantity and Location (all fields need to be filled to create an order.) g. Auto updates Order Status and Order date (fills date and time) h. Click on 'Save and add another'.	One or more orders should be created and user should be able to view the Order list and success message.
TC003	Order Creation	Verify that the Existing Order can be modified or deleted.		a. Existing Order list is opened. b. Select the Order which needs to be modified. c. For order updation, do the required modifications. click on save. d. Click on 'Go' e. For deleting order, Select Action Dropdown 'Delete selected Order' f. Click on 'Go'	Existing Order should be updated/deleted successfully and Success message is displayed.
TC004	View Orders	Verify the order details		a. Open the application. b. Dashboard is displayed. c. Click on Orders d. Displays Order form.	Order details should be displayed successfully
TC005	View Orders	Verify user is able to Process or Cancel Orders		a. Open the application. b. Dashboard is displayed. c. Click on Orders d. Displays Order form. e. For order status updating/deletion, select appropriate actions from dropdown	Order status should be updated successfully. When delete order is selected, order should be removed from the list.
TC006	Add Gas station	Verify a new Gas station is added successfully.		a. Open the application. b. Dashboard is displayed. c. Click on Add Gas station d. Displays Gas station form. e. Enter Name, Phone Number (valid numeric), Email (valid mail ID ex: abc@gmail.com), Location (valid location and mandatory field) and Inventory (Mandatory field) e. Click on save	Gas station should be added successfully.
TC007	Add Gas station	Verify Gas station is modified or deleted successfully.		a. Existing Gas station list is opened. b. Select the Gas station which needs to be modified. c. Do the required modifications. click on save. d. For removing gas station, Select Action Dropdown 'Delete selected gas stations' e. Click on 'Go'	Gas station should be modified or removed from the list successfully.

TC008	Add Driver	Verify a new Driver is added successfully.		a. Open the application. b. Dashboard is displayed. c. Click on Add Driver d. Displays Driver details form. e. Enter Name, License Number (valid numeric)¬¬¬¬¬¬¬, Phone Number(valid numeric) and Email (valid mail ID ex: abc@gmail.com) Fill all the fields e. Click on save	Driver should be added successfully.
TC009	Add Driver	Verify the Drivers list is modified or Deleted successfully.		a. Existing Driver list is opened. b. Select the Driver field which needs to be modified. c. Do the required modifications. click on save. D. For deletion of Drivers from list, select Action Dropdown 'Delete selected Driver' c. Click on 'Go'	Driver details should be modified or removed from the list successfully.

4) Compilation/Running program instructions

Frontend: -

Step1: To begin with, we must download Visual Studio Code, which will serve as the IDE   for the frontend repository.

Here we can able to download visual studio code by using this link, select Windows or Mac based on your system.https://code.visualstudio.com/download.

Step 2:Next we need to install python version above 3.12.0 or above suggested.

 https://www.python.org/downloads/.

Step 3: Install Django Packages inside visual studio code.

Step 4: Go the file menu in Visual Studio and select our folder ,we named our folder name as FuelExpress.

Step 5: Select the new terminal and select GitBash which is located in the powershell below in visual studio code.

Note: If you don't have GitBash use the below link to download.

https://git-scm.com/downloads.

Step 6:

After the installation of gitbash, open gitbash terminal in visual Studio Code to create a virtual environment and installing the packages required.Follow the below bash commands to do so.

```
sahit@NaSahi MINGW64 ~/OneDrive/Desktop/CSCE Software Engineering/Updated fuelxpress work with tests/fuelxpre
ss
$ python -m venv venv

sahit@NaSahi MINGW64 ~/OneDrive/Desktop/CSCE Software Engineering/Updated fuelxpress work with tests/fuelxpre
ss
$ source venv/Scripts/activate
(venv)
sahit@NaSahi MINGW64 ~/OneDrive/Desktop/CSCE Software Engineering/Updated fuelxpress work with tests/fuelxpre
ss
$ pip install -r requirements.txt
```

Step7:

We apply the database migrations now using 'manage.py make migration' and 'manage.py migrations', it installs missing dependencies.

Step8:

To run the development server follow the below command

```
sahit@NaSahi MINGW64 ~/OneDrive/Desktop/CSCE Software Engineering/Updated fuelxpress work with tests/fuelxpre
ss
$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 23, 2023 - 19:25:13
```

Step9:

For Accessing the developed panel

 URL: http://localhost:8000

Step10:

For running the tests

   manage.py test fuelexpress_gasstations

   manage.py test fuelexpress_orders

   manage.py test fuelexpress_drivers

Backend: -

Sql Lite: Sql lite can be used as web browser which contains a self-contained serverless and embedded relational data management system

1. Authentication of Users and User Registration

   Save user account data in a SQLite database, Users can register and log in this database.

   Now, login using User credentials:(for example)

   User=" Hackelite"

Password=" India@77"

2. **Django:** web framework that was used to create the application, runs with python.
3. **CrispyBootStrap4:** we used it to render forms in simple & elegant manner.
4. **Pillow:** Python imaging library for processing pictures.
5. **Python_Decouple:** we can segregate source code from settings of configuration.

5) User Manual

a. Getting Started

To get started with accessing Fuel Xpress website, access the URL http://localhost:8000/home.

Once user lands on homepage, Click on the "Sign Up" or "Register" button to create an account if not already registered. Fill in the required registration details, such as your name, email, and password.

After successfully registered, user will log using appropriate credentials. On Dashboard page, User will be able to Create a fuel order, View List of Orders.

After logging in with Admin Credentials, you'll be directed to the admin dashboard. On Admin Dashboard, following details are displayed:

Drivers, Gas stations, Users - This will be used to assign access level to other users using Groups.

Admin will be able to monitor orders created for all the users here, view the data and Add/Delete/Update Gas Station or the Drivers list.

b. Code Compilation Results

Above is the image of sign up, where the user has to create Username, and give his email address and has to create the password, then the user is successfully signed up.

Above the picture of webpage which has orders list, gas stations information, Drivers and a log in page

In the extreme right corner, there is an Xpress Sidebar which has three options they are "order fuel here", "Announcements", "calendars".

After sign in when we log in, we can now click on "order fuel here"

The above picture shows the Order's details which asks the type of fuel, Gas station, Drivers, Quantity and Delivery Location. After filling up the details, you may now proceed and click on "create order"

Above is the image of "Your Orders", here we can also see the order date and time.
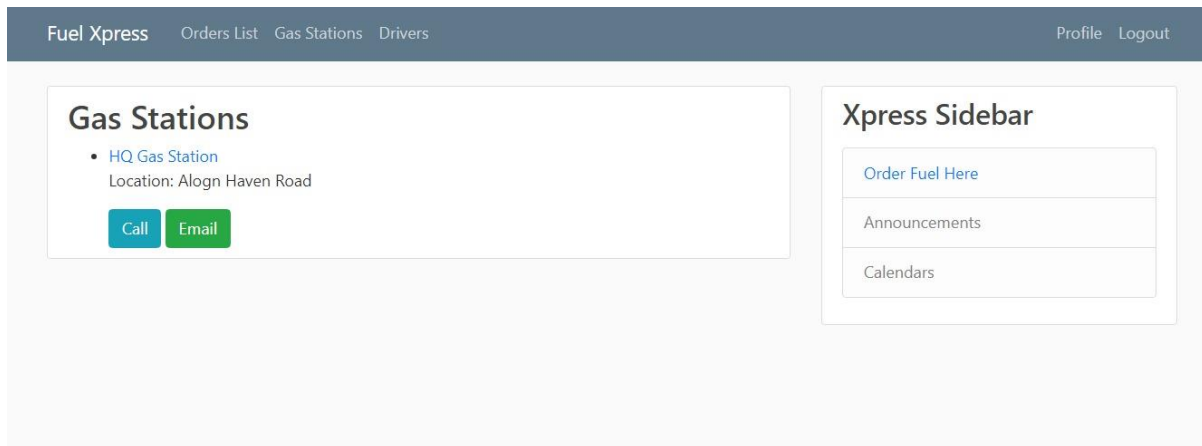
The gas stations nearby will be displayed.
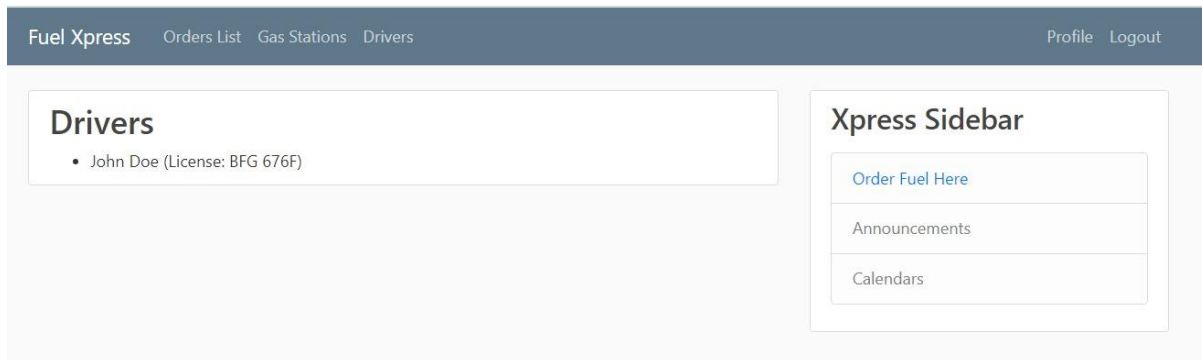
After successfully placing the order, the drivers will be assigned and the fuel will be delivered to the respective customers.

URL: http://localhost:8000/admin/users/profile/

Below is the Fuel Xpress Admin Dashboard Which has Authentication and Authorization info, Driver's info, Gas stations information, Orders and Users information (Profiles).

Recent Actions will let us know about all the recent orders and driver's information.
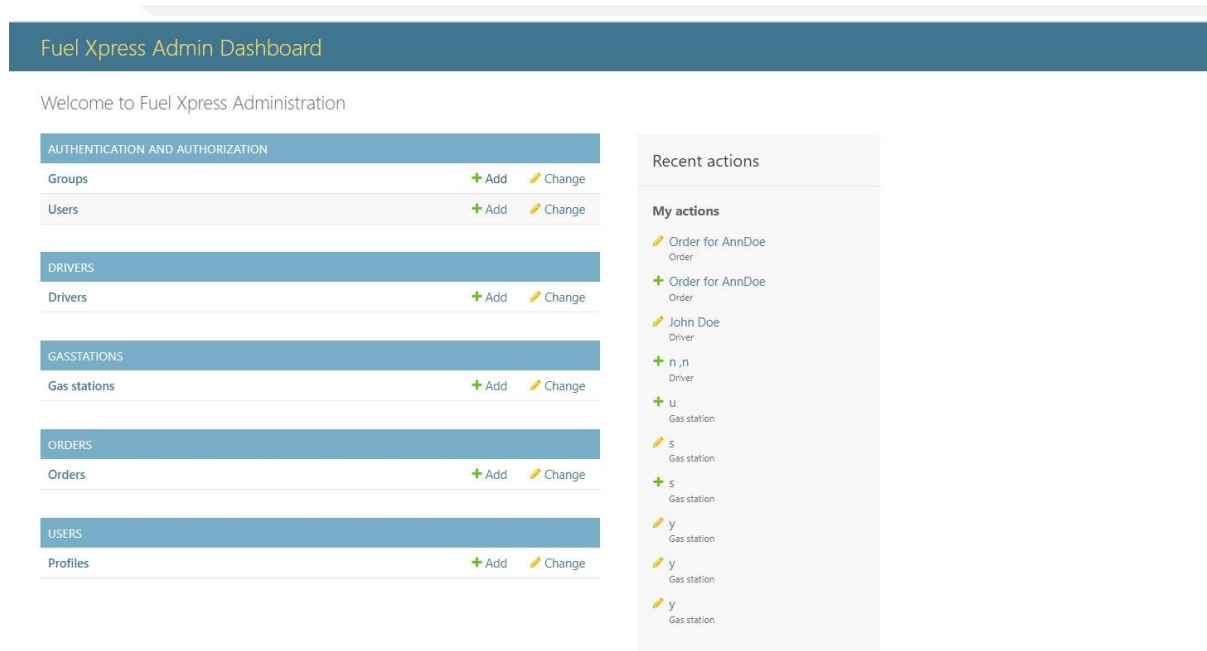
Below is the image of Users, we can also add user by clicking on "ADD USER", In the Admin dashboard we can also filter by the staff status, by superuser status and active.
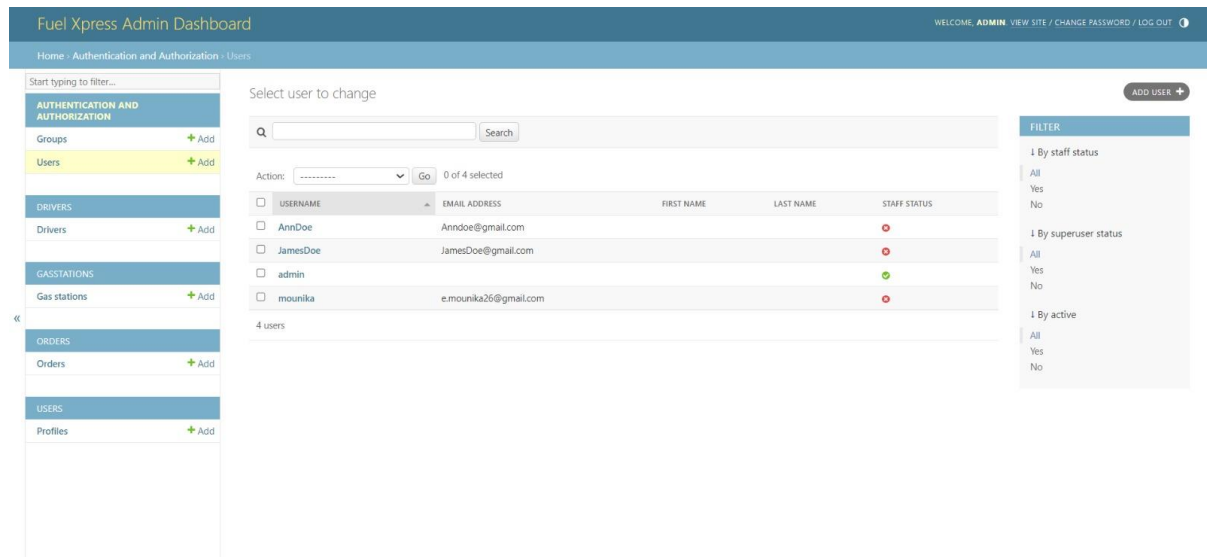
Upon clicking the add user it will ask us to enter the username and password as shown in the image below

In "ADD DRIVER" we make sure to take the license number and phone number and email address, name .

Below is the image for selecting the drivers to change

Below is the image for adding a gas station

Select order to change

Add order

These are the profiles and we can make changes to it by adding new profiles

Upon clicking add profile it will ask user and image then it will ask us to save. Finally, it gets saved.

c. Test Results:

6) Code Inspection Review Feedback

The code inspection session with team Tech Geeks proved to be a source of valuable insights and enabled us as a team to evaluate the overall code quality. They have appreciated the way we have implemented and organized the code as per standards.

We have received an feedback to improve on code readability by following standard practices for class and method details, provide a client side validation during registration, which already has been sufficed.

7) Report Reflection

As part of Phase-2 implementation, we have improved on the feedback received from our peer team to improve code readability by providing the standard names for classes, methods and user defined variables. Python provides a vast range of predefined libraries which are incorporated in our code to improve the coverage.

We are planning to ahead in a similar fashion and completed the project in a well fashioned manner.

8) Member Contribution Table:

Member name	Contribution description	Overall contribution (%)	Note (if applicable)
Sahithi Regalla	Code for view orders, Word Document.	12	
Uday Vandanapu	Code for view orders, Code Inspection.	11	
Kavya Jangapalli	Code for order creation, Note Deliverable, Member Contribution table.	11	
Durga Devi Pampani	Code for order processing, UML Diagrams.	11	
Karthik Setti	Code for adding gas station and driver details, Code Inspection.	11	
Hemanth Janapala	Requirement changes, Code Inspection Document, Code for adding gas station and driver details.	11	
Pranay Pendyala	Code for adding gas station and driver details Meeting Minutes.	11	
Mounika Eddala	Code for order creation, Test Cases.	11	
Vaishnavi Adapa	Code for order processing, User Manual.	11	