

CSE471: System Analysis and Design

Project Report

Project Title: TankupNow(Petrol delivery system)

Table of Contents

Section No.	Content	Page No.
1	System Request	2-5
2	Functional Requirements	5-6
3	Technologies	6
4	Backend Development	6-13
5	User Interface Design and User manual	14-19
6	Frontend Development	19-29
7	GitHub Repo Link	29-30
8	Individual Contribution	30-32

Introduction

The Petrol Delivery Management System addresses the growing need for convenient and efficient fuel delivery services. Traditional refueling methods often involve significant time and effort, particularly in busy urban areas. This project provides a comprehensive solution with dedicated dashboards for users, drivers, and administrators to streamline the entire process.

The system allows users to order fuel types such as petrol, diesel, and octane while specifying quantities and delivery details. Real-time status updates keep users informed at every stage of their order. Drivers benefit from a structured process for managing deliveries, including options for accepting, picking up, and completing orders. Administrators maintain oversight, ensuring smooth operations and resolving potential issues. Additional features like order animations, notifications, and detailed order histories enhance user experience and transparency. By automating and optimizing fuel delivery workflows, this system provides a reliable and user-friendly solution for modern fuel logistics.

System Request

Business Need

Fuel delivery services are becoming increasingly important in today's fast-paced and convenience-driven society. Traditional methods of refueling vehicles at petrol stations often involve waiting in long queues, dealing with limited operating hours, or traveling significant distances, which can be time-consuming and inconvenient for customers. Businesses also face challenges in managing the logistics of fuel delivery effectively, particularly in urban areas with high traffic or remote locations where accessibility is an issue.

This project addresses these needs by providing a digital platform that allows users to place fuel orders online, specify their requirements, and have the fuel delivered to their desired location. Drivers can manage their delivery tasks more efficiently, reducing idle time and ensuring faster service. For businesses, the system offers centralized management, better order tracking, and real-time updates, which lead to improved decision-making and resource allocation. The platform also meets the growing demand for contactless services, ensuring safety and convenience for users during fuel transactions.

By bridging the gap between fuel suppliers and consumers, the Petrol Delivery Management System aligns with modern consumer expectations and business goals, creating a win-win scenario for all stakeholders.

Business Value

1. **Customer Convenience:** Enables users to order fuel from the comfort of their homes, avoiding the hassle of visiting fuel stations.
2. **Time Efficiency:** Saves time for both customers and drivers through streamlined delivery processes and real-time order tracking.
3. **Market Expansion:** Provides fuel providers with a platform to reach a wider audience, including residential and commercial customers.

4. **Operational Optimization:** Enhances logistics management for fuel deliveries, improving overall efficiency and reducing operational costs.
5. **Customer Retention:** Builds loyalty by offering reliable and transparent services, fostering trust and repeat business.
6. **Scalable Solution:** Offers a foundation for future expansion, such as adding more fuel types or integrating new features like subscription-based services.

Business Requirements

1. Customer Interaction:

- Allow users to place fuel orders with details like type, quantity, and delivery address.
- Provide real-time updates on order status (pending, accepted, picked up, delivered).

2. Driver Efficiency:

- Enable drivers to view and accept assigned orders.
- Provide tools for updating order status at various stages (accept, pick up, deliver).

3. Admin Oversight:

- Allow administrators to monitor and manage all orders and system users.
- Maintain records for auditing and operational insights.

4. Scalability and Security:

- Ensure a secure login and authentication system for users, drivers, and admins.
- Enable the system to handle increased demand and integrate additional features in the future.

5. Customer Satisfaction:

- Provide seamless navigation, order history, and notifications for enhanced user experience.

- Include dynamic pricing based on fuel type and quantity for transparency.

Functional Features:

Module 1:

1. User/Driver/Admin Signup
2. User/Driver/Admin Login
3. User/Driver/Admin edit profile
4. Driver can register a vehicle

Module 2 :

1. User can order varieties of fuel (like petrol, diesel, octane)
2. User can select the quantity of fuel (cannot order less than 3 if a user input a value which is less than 3 an invalid message will be pop out)
3. User can give the address of fuel station and confirm order
4. An order history has been shown were the detailed info of order is popping after confirming an order by user
5. A table regarding assigned order is showing were all the info about order is showing in driver dashboard (like order id, status, price, quantity, fuel type and userinfo)
6. Pricing is handled in user dashboard according to the quantity of fuel

Module 3:

7.An option for accepting order is given which updates the status of order

8.After accepting order pickup option appear which eventually updates the status of order

9.Lastly after picking up order order delivered button appears which also updates the status of order

10. Status in user dashboard is updating in real time by accepting,picking up and order delivered by driver

11.An animation for accepting order is given in user dashboard

12.Notification is shown in user dashboard when the order is delivered

13.An animation for picking up order is showing in user dashboard

Technologies used : Html ,Css(tailwind),Flask,DB Browser(SQL cipher)

Backend Development

API - 01

The /dashboard API determines the dashboard view based on the logged-in user's role. If the user is a driver, it displays pending and assigned orders along with their registered vehicles. For regular users, it shows their order history with detailed statuses. Admin users can view all orders in the system. The API ensures authentication, validates roles, and redirects unauthorized users to the login or logout page as needed. It also dynamically renders the appropriate dashboard template for each role.

```
@app.route('/dashboard')
def dashboard():
    if 'username' not in session:
        flash('You need to log in first.', 'info')
        return redirect(url_for('login'))

    # Fetch the logged-in user
    user = User.query.filter_by(username=session['username']).first()

    if not user:
        flash('User not found.', 'error')
        return redirect(url_for('logout'))

    # Driver dashboard
    if user.role == 'driver':
        pending_orders = FuelOrder.query.filter(
            (FuelOrder.status == 'Pending') | (FuelOrder.driver_id == user.id)
        ).all()

        registered_vehicles = Vehicle.query.filter_by(user_id=user.id).all()

    return render_template(
        'dashboard_driver.html',
        driver=user,
        pending_orders=pending_orders,
        vehicles=registered_vehicles
```

```

# User dashboard
elif user.role == 'user':
    # Fetch all orders for the logged-in user
    user_orders = FuelOrder.query.filter_by(user_id=user.id).all()

    # Debugging: Print user orders in the server logs
    if not user_orders:
        print(f"No orders found for user: {user.username}")
    else:
        print(f"Orders for user {user.username}:")
        for order in user_orders:
            print(f"Order ID: {order.id}, Status: {order.status}")

    return render_template(
        'dashboard_user.html',
        user=user,
        orders=user_orders
    )

# Admin dashboard
elif user.role == 'admin':
    all_orders = FuelOrder.query.all()

    return render_template(
        'dashboard_admin.html',

```

API - 02

This API handles user fuel orders by validating inputs like fuel type and quantity (minimum 3 liters), calculating the total price, and storing order details with a "Pending" status. It ensures users are logged in before placing an order and redirects them to their dashboard upon success.

```

def order_fuel():
    if 'username' not in session:
        flash("Please log in to place an order.", "error")
        return redirect(url_for('login'))

    user = User.query.filter_by(username=session['username']).first()

    if request.method == 'POST':
        # Retrieve form data
        fuel_type = request.form['fuel_type']
        quantity = float(request.form['quantity'])
        fuel_station_address = request.form['address'] # Get fuel station address

        # Validation: Quantity must be at least 3 liters
        if not fuel_type or quantity < 3:
            flash("Please select a fuel type and order at least 3 liters of fuel.", "error")
            return redirect(url_for('order_fuel'))

        # Define price per liter for each fuel type
        fuel_prices = {
            'octane': 125,
            'diesel': 110,
            'petrol': 120
        }

        # Ensure the selected fuel type is valid
        if fuel_type not in fuel_prices:
            flash("Invalid fuel type selected.", "error")
            return redirect(url_for('order_fuel'))

        # Calculate total price based on the selected fuel type
        price_per_litre = fuel_prices[fuel_type]
        total_price = quantity * price_per_litre

        # Create and save the new FuelOrder
        new_order = FuelOrder(
            user_id=user.id,
            fuel_type=fuel_type,
            quantity=quantity,
            address=fuel_station_address,
            total_price=total_price,
            status="Pending" # Default status
        )

        db.session.add(new_order)
        db.session.commit()

        # Flash success message
        flash(f"Order placed successfully! Total Price: {total_price:.2f} tk", "success")
        return redirect(url_for('dashboard'))

```

API - 03

This API lets drivers update the status of their orders (e.g., "Accepted," "Picked Up," "Delivered"). It validates the driver's identity and permissions before updating the order in real-time.

```
# Update Order Status Route
@app.route('/update_order_status/<int:order_id>/<string:status>', methods=['POST'])
def update_order_status(order_id, status):
    if 'username' not in session:
        return jsonify(success=False, message='Please log in first.')

    driver = User.query.filter_by(username=session['username'], role='driver').first()
    order = FuelOrder.query.get_or_404(order_id)

    if order.driver_id == driver.id:
        order.status = status
        db.session.commit()
        return jsonify(success=True)
    else:
        return jsonify(success=False, message='Invalid order or permission denied.')

# Delete Order Route
```

API - 04

Drivers use this API to mark orders as "Delivered." It validates the driver's permissions and updates the status, ensuring the delivery workflow is complete.

```
# Deliver Order Route
@app.route('/deliver_order/<int:order_id>', methods=['POST'])
def deliver_order(order_id):
    if 'username' not in session:
        flash('Please log in first.', 'error')
        return redirect(url_for('login'))

    driver = User.query.filter_by(username=session['username'], role='driver').first()
    order = FuelOrder.query.get_or_404(order_id)

    if order.status == 'Picked Up' and order.driver_id == driver.id:
        order.status = 'Delivered'
        db.session.commit()
        flash(f'Order {order.id} has been delivered.', 'success')
    else:
        flash('Order cannot be delivered.', 'warning')

    return redirect(url_for('dashboard'))
```

API - 05

This API finalizes a user's order by saving all details (fuel type, quantity, address, and price) into the database and confirming it for processing.

```
@app.route('/confirm_order', methods=['POST'])
def confirm_order():
    if 'username' not in session:
        flash('Please log in to place an order.', 'error')
        return redirect(url_for('login'))

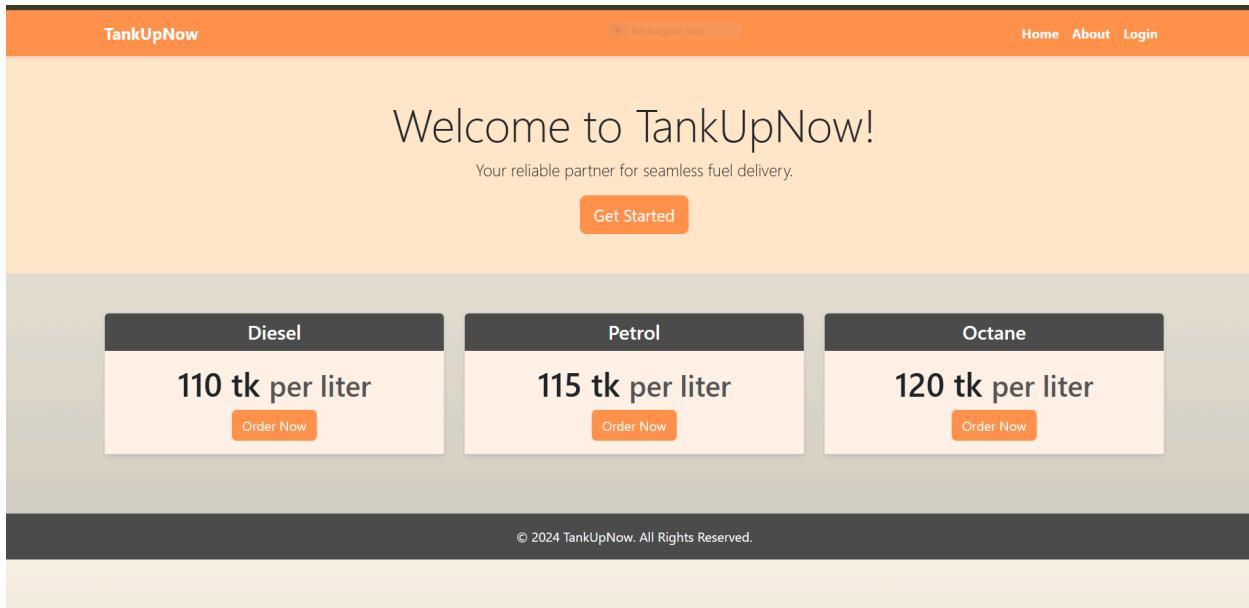
    # Extract form data
    fuel_type = request.form['fuel_type']
    quantity = int(request.form['quantity'])
    fuel_station_address = request.form['fuel_station_address']
    total_price = float(request.form['total_price'])

    # Create a new fuel order
    user = User.query.filter_by(username=session['username']).first()
    new_order = FuelOrder(
        user_id=user.id, # Use logged-in user's ID
        fuel_type=fuel_type,
        quantity=quantity,
        address=fuel_station_address,
        total_price=total_price
    )

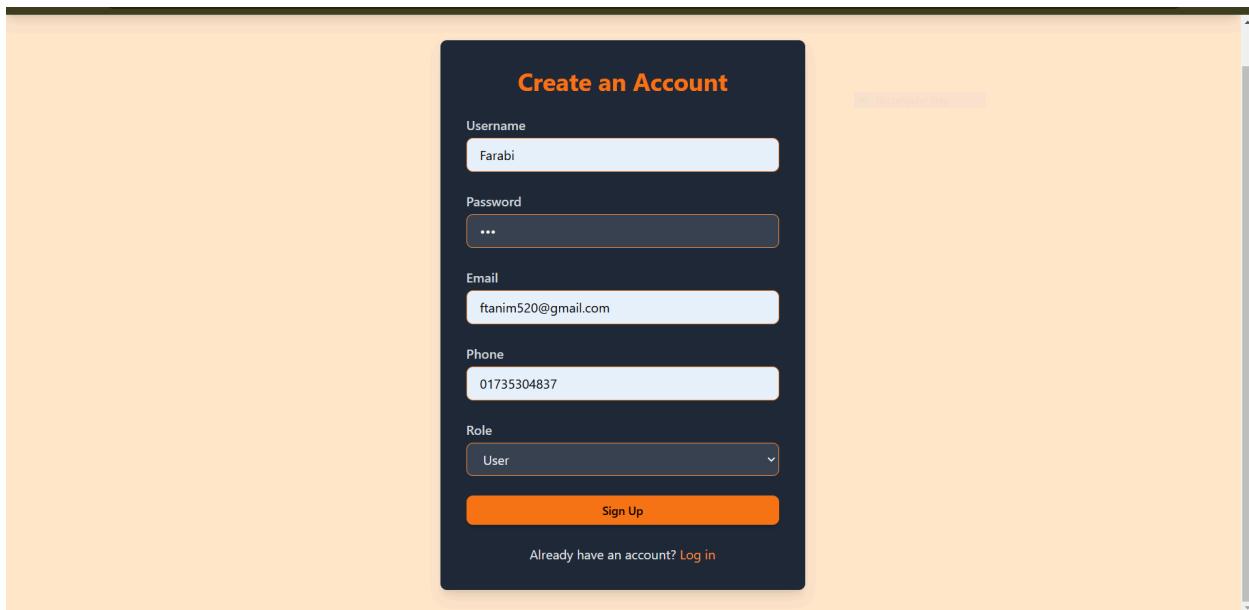
    db.session.add(new_order)
    db.session.commit()
    flash('Order confirmed successfully!', 'success')
    return redirect(url_for('user_dashboard'))
```

User Interface Design and User manual

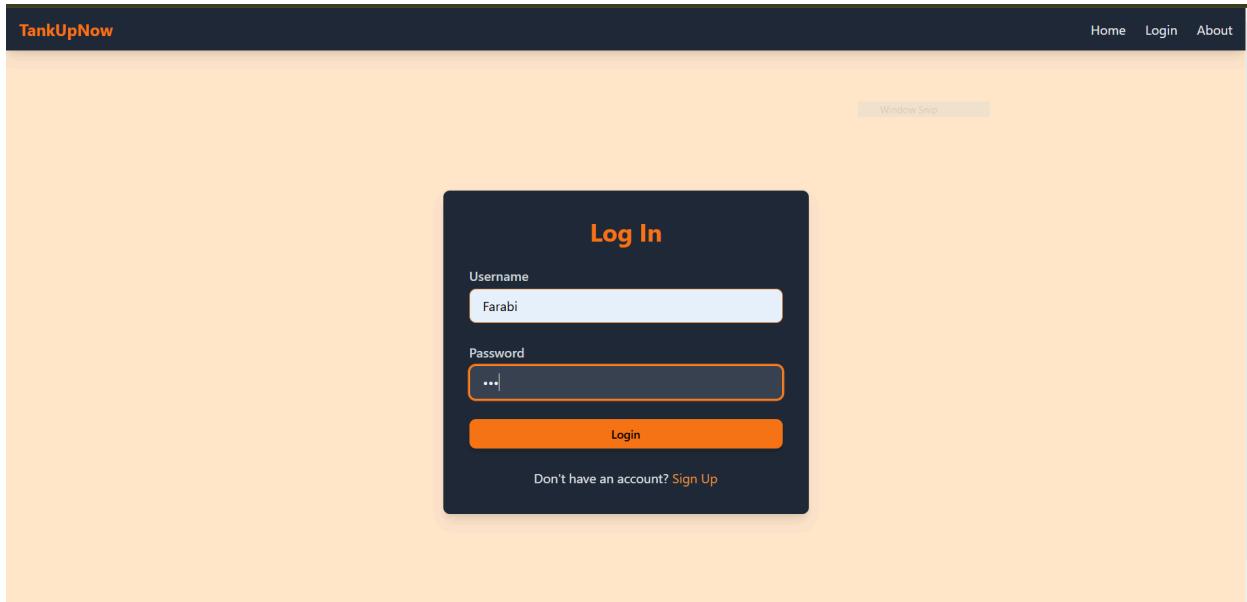
Home page: This is the view of home page



Signup: Users/Driver/Admin can sign up like this



Login Page : User/Admin/Driver can login after signing up



User dashboard: User can select varieties of fuel ,select the quantity of fuel and select the fuel station address. User cannot select the quantity of fuel less than 3 it will pop up a message if the quantity is less than 3 and there is table for order history were user can see the order details like price fuel

type status etc.

The screenshot shows a fuel ordering interface. On the left, there's a form titled "Place Your Fuel Order" with fields for "Fuel Type" (Petrol), "Quantity (Litres)" (5), and "Address" (Moghbazar Fuel station). A "Confirm Order" button is at the bottom. On the right, a "Your Order History" table lists three orders:

Order ID	Fuel Type	Price	Status	Action
1	petrol	600.0	Pending	N/A
2	diesel	660.0	Pending	N/A
3	octane	2500.0	Pending	N/A

Below the order history is a "Dashboard Options" section with "Go to Profile" and "Logout" buttons. The taskbar at the bottom shows various application icons and the date/time (1/6/2025).

Driver dashboard:

The driver dashboard has a purple header with a driver profile picture, the text "Welcome, sami!", and "You are logged in as a driver." It includes "Edit Profile" and "Logout" buttons. Below the header is a section titled "Your Registered Vehicles" with the message "No vehicles registered. Register a vehicle". The main content area is titled "Your Assigned Orders" and contains a table:

Order ID	User	Fuel Type	Quantity	Total Price	Status	Action
1	Farabi	petrol	5 Litres	600.0 tk	Accepted	pick-Up
2	Farabi	diesel	6 Litres	660.0 tk	Delivered	Delivered
3	Farabi	octane	20 Litres	2500.0 tk	Pending	Accept

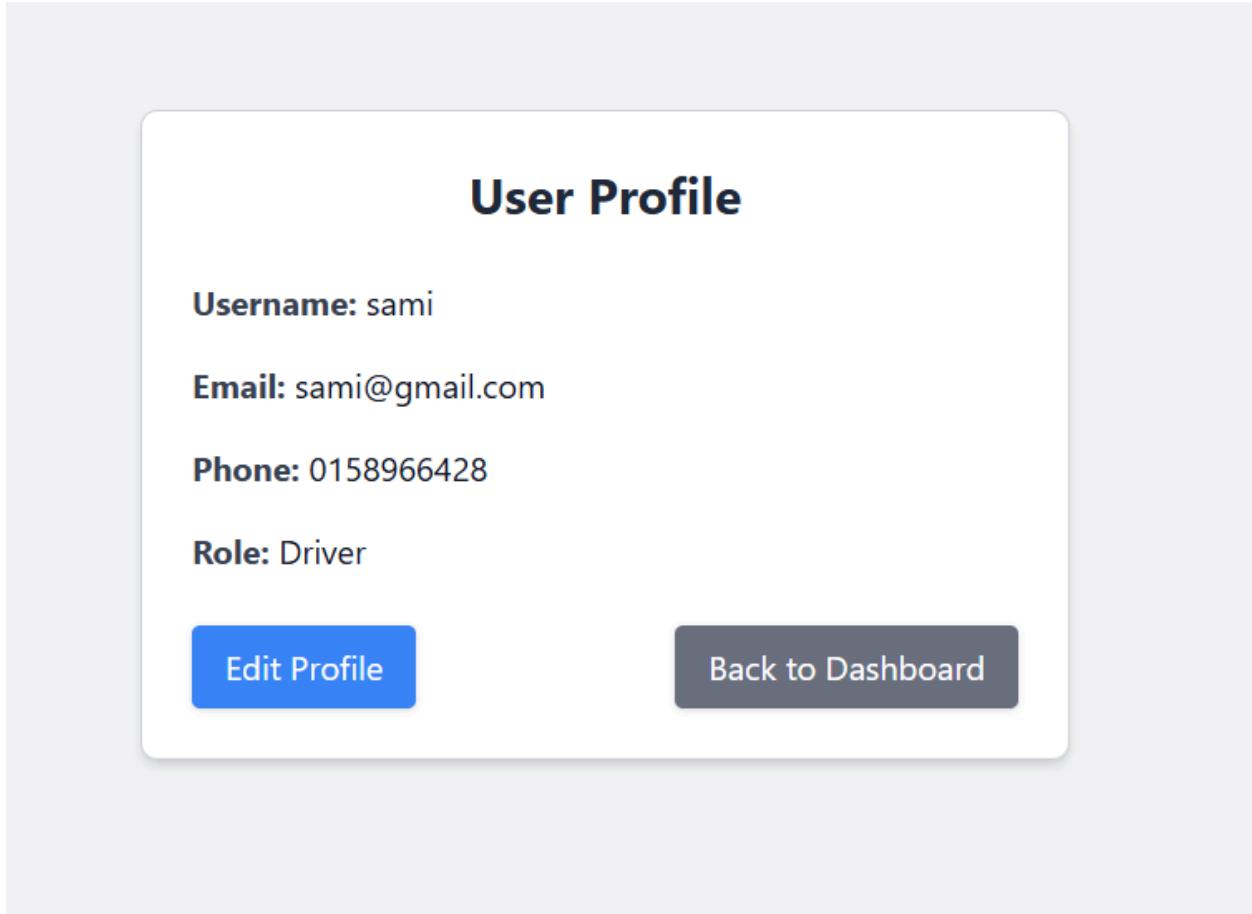
Register vehicle: There is an option for register vehicle in driver dashboard were driver can register a vehicle through a form and it will show in the driver dashboard .

The screenshot shows a registration form titled "Register Your Vehicle". It includes fields for "Vehicle Model" (with a placeholder "Rectangular Snip"), "Vehicle Year" (with a placeholder "3"), and "License Plate" (with a placeholder "Rectangular Snip"). At the bottom left is a "Back to Dashboard" link.

The screenshot shows a driver dashboard with a purple header. It features a "Driver Picture" placeholder, a welcome message "Welcome, sami!", and a note "You are logged in as a driver." Below this are "Edit Profile" and "Logout" buttons. The main content area is titled "Your Registered Vehicles" and contains a table with one row:

Model	Year	License Plate
bmw	2012	DK-1234-AB

Edit profile: User and driver can update their profile. There is an option in both user and driver dashboard to update their profile.



Animation and status update in user dashboard: After accepting and picking up order by driver we have added a particular animation which is our special feature for this project. And the status in user dashboard is also updating so that the user can understand if it is accepted , picked up or delivered

TankUpNow

Place Your Fuel Order

Fuel Type

Quantity (Litres)

Address

Confirm Order

Your Order History

Order ID	Fuel Type	Price	Status	Animation
1	petrol	600.0	Accepted	
2	diesel	660.0	Delivered	✓ Order completed Your order is delivered. If you need any help please contact with us. Thankyou for staying with us. Give Rating Dismiss
3	octane	2500.0	Picked Up	

Frontend development : Following codes are for home page

```
/our_package > templates > home.html > ...
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>TankUpNow - Home</title>
7      <link rel="stylesheet" href="/static/css/bootstrap.min.css">
8      <style>
9          /* Custom Colors */
10         body {
11             background: linear-gradient(to bottom, #f7f1e3, #d1ccc0);
12         }
13         .navbar {
14             background-color: #ff914d; /* Orange shade */
15         }
16         .navbar-brand, .nav-link {
17             color: white !important;
18             font-weight: bold;
19         }
20         .nav-link:hover {
21             color: #4b4b4b !important;
22         }
23         .card-header {
24             background-color: #4b4b4b; /* Ash color */
25             color: white;
26         }
27         .card-body {
28             background-color: #fff4e6;
29         }
30         .btn-primary {
31             background-color: #ff914d;
32             border-color: #ff914d;
33         }
34         .btn-primary:hover {
35             background-color: #e67732;
36             border-color: #e67732;
37         }

```

```
37     }
38     footer {
39         background-color: #4b4b4b;
40         color: white;
41     }
42 
```

Rectangular Snip

```
43 </style>
44 </head>
45 <!-- Navigation Bar -->
46 <nav class="navbar navbar-expand-lg navbar-dark shadow-sm">
47     <div class="container">
48         <a class="navbar-brand" href="#">TankUpNow</a>
49         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-e
50             <span class="navbar-toggler-icon"></span>
51         </button>
52         <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
53             <ul class="navbar-nav">
54                 <li class="nav-item">
55                     <a class="nav-link" href="/">Home</a>
56                 </li>
57                 <li class="nav-item">
58                     <a class="nav-link" href="/about">About</a>
59                 </li>
60                 <li class="nav-item">
61                     <a class="nav-link" href="/login">Login</a>
62                 </li>
63             </ul>
64         </div>
65     </div>
66 </nav>
67
68 <!-- Hero Section -->
69 <div class="text-center py-5" style="background-color: #ffe6cc;">
```

```

70   <div class="text-center py-5" style="background-color: #ffe6cc;">
71     <h1 class="display-4">Welcome to TankUpNow!</h1>
72     <p class="lead">Your reliable partner for seamless fuel delivery.</p>
73     <a href="/signup" class="btn btn-primary btn-lg">Get Started</a>
74   </div>
75
76   <!-- Fuel Pricing Cards -->
77   <main class="container mt-5">
78     <div class="row">
79       <!-- Diesel Card -->
80       <div class="col-md-4">
81         <div class="card mb-4 shadow-sm">
82           <div class="card-header text-center">
83             <h4 class="my-0 font-weight-normal">Diesel</h4>
84           </div>
85           <div class="card-body text-center">
86             <h1 class="card-title pricing-card-title">110 tk <small class="text-muted">per liter</small></h1>
87             <a href="/login" class="btn btn-primary btn-block">Order Now</a>
88           </div>
89         </div>
90       </div>
91
92       <!-- Petrol Card -->
93       <div class="col-md-4">
94         <div class="card mb-4 shadow-sm">
95           <div class="card-header text-center">
96             <h4 class="my-0 font-weight-normal">Petrol</h4>
97           </div>
98           <div class="card-body text-center">
99             <h1 class="card-title pricing-card-title">115 tk <small class="text-muted">per liter</small></h1>
100            <a href="/login" class="btn btn-primary btn-block">Order Now</a>
101          </div>
102        </div>
103      </div>
104
105      <!-- Octane Card -->
106      <div class="col-md-4">
107        <div class="card mb-4 shadow-sm">
108          <div class="card-header text-center">
109            <h4 class="my-0 font-weight-normal">Octane</h4>
110          </div>
111          <div class="card-body text-center">
112            <h1 class="card-title pricing-card-title">120 tk <small class="text-muted">per liter</small></h1>
113            <a href="/login" class="btn btn-primary btn-block">Order Now</a>
114          </div>
115        </div>
116      </div>
117    </div>
118  </main>
119
120  <!-- Footer -->
121  <footer class="text-center py-3 mt-5">
122    <p class="mb-0">&copy; 2024 TankUpNow. All Rights Reserved.</p>
123  </footer>
124
125  <!-- Bootstrap Scripts -->
126  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
127  <script src="/static/js/bootstrap.bundle.min.js"></script>
128 </body>
129 </html>
130

```

Signup page: User/Driver can signup manually

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up</title>
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-[#ffe6cc] text-gray-800 min-h-screen">
    <!-- Navbar -->
    <nav class="bg-gray-800 shadow-lg">
        <div class="container mx-auto px-4">
            <div class="flex justify-between items-center py-3">
                <a href="/" class="text-orange-500 text-xl font-bold">TankUpNow</a>
                <div class="flex items-center space-x-6">
                    <a href="/" class="text-gray-100 hover:text-orange-400">Home</a>
                    <a href="/login" class="text-gray-100 hover:text-orange-400">Login</a>
                    <a href="/about" class="text-gray-100 hover:text-orange-400">About</a>
                </div>
            </div>
        </div>
    </nav>

    <!-- Sign-Up Form -->
    <div class="flex items-center justify-center min-h-screen">
        <div class="bg-gray-800 shadow-lg rounded-lg p-8 max-w-md w-full">
            <h2 class="text-3xl font-bold text-center text-orange-500 mb-6">Create an Account</h2>
            <form method="POST" action="/signup" class="space-y-6">
                <div>
                    <label class="block font-medium text-gray-300 mb-1">Username</label>
                    <input type="text" name="username" required
                           class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus">
                </div>
                <div>
                    <label class="block font-medium text-gray-300 mb-1">Password</label>
                    <input type="password" name="password" required
                           class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus">
                </div>
                <div>
                    <label class="block font-medium text-gray-300 mb-1">Email</label>
                    <input type="email" name="email" required
                           class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus">
                </div>
                <div>
                    <label class="block font-medium text-gray-300 mb-1">Phone</label>
                    <input type="text" name="phone" required
                           class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus">
                </div>
                <div>
                    <label class="block font-medium text-gray-300 mb-1">Role</label>
                    <select name="role" required
                           class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus">
                        <option value="user">User</option>
                        <option value="driver">Driver</option>
                        <option value="admin">Admin</option>
                    </select>
                </div>
                <div>
                    <button type="submit"
                           class="w-full bg-orange-500 hover:bg-orange-600 text-black py-2 rounded-lg shadow-md text-sm font-semibold transition">
                        Sign Up
                    </button>
                </div>
            </form>
            <div class="mt-6 text-center">
                <p class="text-gray-200">Already have an account?</p>
                <p> <a href="/login" class="text-orange-400 hover:underline">Log in</a></p>
            </div>
        </div>
    </div>
</body>
</html>

```

Login page: User/Driver can Login after signing up

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Login</title>
7      <script src="https://cdn.tailwindcss.com"></script>
8  </head>
9  <body class="bg-[#ffe6cc] text-gray-800 min-h-screen">
10     <!-- Navbar -->
11     <nav class="bg-gray-800 shadow-lg">
12         <div class="container mx-auto px-4">
13             <div class="flex justify-between items-center py-3">
14                 <a href="/" class="text-orange-500 text-xl font-bold">TankUpNow</a>
15                 <div class="flex items-center space-x-6">
16                     <a href="/" class="text-gray-100 hover:text-orange-400">Home</a>
17                     <a href="/login" class="text-gray-100 hover:text-orange-400">Login</a>
18                     <a href="/about" class="text-gray-100 hover:text-orange-400">About</a>
19                 </div>
20             </div>
21         </div>
22     </nav>
23
24     <!-- Login Form -->
25     <div class="flex items-center justify-center min-h-screen">
26         <div class="bg-gray-800 shadow-lg rounded-lg p-8 max-w-md w-full">
27             <h2 class="text-3xl font-bold text-center text-orange-500 mb-6">Log In</h2>
28             <form method="POST" action="/login" class="space-y-6">
29                 <div>
30                     <label class="block font-medium text-gray-300 mb-1">Username</label>
31                     <input type="text" name="username" required
32                         |   class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus:<br/>
33                         |   <div>
34                         |       <label class="block font-medium text-gray-300 mb-1">Password</label>
35                         |       <input type="password" name="password" required
36                         |           |   class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus:<br/>
37

```

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Login</title>
7      <script src="https://cdn.tailwindcss.com"></script>
8  </head>
9  <body class="bg-[#ffe6cc] text-gray-800 min-h-screen">
10     <!-- Navbar -->
11     <nav class="bg-gray-800 shadow-lg">
12         <div class="container mx-auto px-4">
13             <div class="flex justify-between items-center py-3">
14                 <a href="/" class="text-orange-500 text-xl font-bold">TankUpNow</a>
15                 <div class="flex items-center space-x-6">
16                     <a href="/" class="text-gray-100 hover:text-orange-400">Home</a>
17                     <a href="/login" class="text-gray-100 hover:text-orange-400">Login</a>
18                     <a href="/about" class="text-gray-100 hover:text-orange-400">About</a>
19                 </div>
20             </div>
21         </div>
22     </nav>
23
24     <!-- Login Form -->
25     <div class="flex items-center justify-center min-h-screen">
26         <div class="bg-gray-800 shadow-lg rounded-lg p-8 max-w-md w-full">
27             <h2 class="text-3xl font-bold text-center text-orange-500 mb-6">Log In</h2>
28             <form method="POST" action="/login" class="space-y-6">
29                 <div>
30                     <label class="block font-medium text-gray-300 mb-1">Username</label>
31                     <input type="text" name="username" required
32                         class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus:<br/>
33                         <div>
34                             <label class="block font-medium text-gray-300 mb-1">Password</label>
35                             <input type="password" name="password" required
36                                 class="w-full bg-gray-700 border border-orange-400 rounded-lg py-2 px-4 text-gray-100 focus:outline-none focus:<br/>
37

```

Order fuel.html: Here the user can select 3 types of fuel like octane diesel and petrol and they can also select the quantity and address

```

<!-- Fuel Order Form -->
<form method="POST" action="{{ url_for('order_fuel') }}>
    <label for="fuel_type">Select Fuel Type:</label>
    <select name="fuel_type" id="fuel_type" required>
        <option value="">-- Select Fuel Type --</option>
        <option value="Diesel">Diesel</option>
        <option value="Petrol">Petrol</option>
    </select>
    <br><br>

    <label for="quantity">Enter Quantity (Liters):</label>
    <input type="number" step="0.1" name="quantity" id="quantity" min="0.1" required>
    <br><br>

    <button type="submit">Place Order</button>
</form>

<!-- Back to Dashboard -->
<p><a href="{{ url_for('dashboard') }}>Back to Dashboard</a></p>
</body>
</html>

```

Profile.html: Here the user can update their profile after log in

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Dashboard</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}>
</head>
<body>
    <h2>User Profile</h2>
    <form method="POST" action="/profile">
        <label>Username:</label>
        <input type="text" name="username" value="{{ user.username }}" disabled>

        <label>Email:</label>
        <input type="email" name="email" value="{{ user.email }}" required>

        <label>Phone:</label>
        <input type="text" name="phone" value="{{ user.phone }}>

        <button type="submit">Update Profile</button>
    </form>

    <a href="/dashboard">Back to Dashboard</a>

```

Register vehicle: Here a register vehicle option is created in driver dashboard were the driver can register vehicle by filling up a form and it will show in the driver dashboard

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Register Vehicle</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <h2>Register Your Vehicle</h2>

    <form method="POST">
        <label for="model">Vehicle Model:</label>
        <input type="text" id="model" name="model" required><br>

        <label for="year">Vehicle Year:</label>
        <input type="number" id="year" name="year" required><br>

        <label for="license_plate">License Plate:</label>
        <input type="text" id="license_plate" name="license_plate" required><br>

        <button type="submit"><a href="{{ url_for('dashboard') }}>Register Vehicle</a></button>
    </form>

    <a href="{{ url_for('dashboard') }}>Back to Dashboard</a>
</body>
</html>

```

● Rectangular Snip

User dashboard: User can select varieties of fuel ,select the quantity of fuel and select the fuel station address. User cannot select the quantity of fuel less than 3 it will pop up a message if the quantity is less than 3 and there is table for order history were user can see the order details like price fuel

type status etc. And some animations were given according to the updated status the status is updating by the driver.

```

    height: 100px;
    margin-bottom: 6px;
    animation: motion 1s linear infinite;
}
/* truck suspension animation*/
@keyframes motion {
  0% {
    transform: translateY(0px);
  }
  50% {
    transform: translateY(3px);
  }
  100% {
    transform: translateY(0px);
  }
}
/* truck's tires */
.truckTires {
  width: 130px;
  height: fit-content;
  display: flex;
  align-items: center;
  justify-content: space-around;
}



<!-- Left Side: Order Placement -->
  <div class="bg-gray-100 shadow-lg p-8 flex flex-col justify-center">
    <h2 class="text-3xl font-bold mb-6 text-orange-500">Place Your Fuel Order</h2>
    <form method="POST" action="{{ url_for('order_fuel') }}" class="space-y-6">
      <div>
        <label class="block font-medium text-gray-700">Fuel Type</label>
        <select name="fuel_type" required
          class="w-full bg-white border border-orange-500 rounded py-2 px-3 text-gray-900">
          <option value="petrol">Petrol</option>
          <option value="diesel">Diesel</option>
          <option value="octane">Octane</option>
        </select>
      </div>
      <div>
        <label class="block font-medium text-gray-700">Quantity (Litres)</label>
        <input type="number" name="quantity" min="3" required
          class="w-full bg-white border border-orange-500 rounded py-2 px-3 text-gray-900">
      </div>
      <div>
        <label class="block font-medium text-gray-700">Address</label>
        <input type="text" name="address" required
          class="w-full bg-white border border-orange-500 rounded py-2 px-3 text-gray-900">
      </div>
    </form>
  </div>


```

Driver dash: Drivers can see the order details which are assigned by the user. Here user info, fuel quantity price can be seen. And there are 3 buttons which Accept order, pickup order and deliver order the driver can

press those options after performing a particular action and the status will be automatically updated over here after pressing those buttons.

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Driver Dashboard</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <style>
        button {
            width: 9em;
            height: 3em;
            border-radius: 30em;
            font-size: 15px;
            font-family: inherit;
            border: none;
            position: relative;
            overflow: hidden;
            z-index: 1;
            box-shadow: 6px 6px 12px #c5c5c5,
                        -6px -6px 12px #ffffff;
        }
        button::before {
            content: '';
            width: 0;
            height: 3em;
            ...
        }
        <!-- Assigned Orders Section -->
        <div class="bg-gradient-to-r from-purple-900 via-black to-purple-900 shadow-lg rounded-lg p-8">
            <h3 class="text-2xl font-semibold mb-4">Your Assigned Orders</h3>
            {% if pending_orders %}
                <table class="w-full text-left border-collapse border border-purple-700 text-white">
                    <thead class="bg-purple-800">
                        <tr>
                            <th class="border border-purple-700 px-4 py-2">Order ID</th>
                            <th class="border border-purple-700 px-4 py-2">User</th>
                            <th class="border border-purple-700 px-4 py-2">Fuel Type</th>
                            <th class="border border-purple-700 px-4 py-2">Quantity</th>
                            <th class="border border-purple-700 px-4 py-2">Total Price</th>
                            <th class="border border-purple-700 px-4 py-2">Status</th>
                            <th class="border border-purple-700 px-4 py-2">Action</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for order in pending_orders %}
                            <tr class="hover:bg-purple-700">
                                <td class="border border-purple-700 px-4 py-2">{{ order.id }}</td>
                                <td class="border border-purple-700 px-4 py-2">{{ order.user.username }}</td>
                                <td class="border border-purple-700 px-4 py-2">{{ order.fuel_type }}</td>
                                <td class="border border-purple-700 px-4 py-2">{{ order.quantity }} Litres</td>
                            </tr>
                        {% endfor %}
                    </tbody>
                </table>
            {% endif %}
        </div>
    </style>
</head>
<body>
    <button>
        ...
    </button>
</body>

```

Github repo link:

<https://github.com/samiifty/Tank-up-Now/blob/main/Rideshare.zip>

Individual Contribution

Group member - 01	
Name: MD Farabi Tanim	Student ID: 21301513
Functional Requirements which are developed by this member:	
<ol style="list-style-type: none">1. Driver can register a vehicle and it will pop up in a table format in driver dashboard2. Pricing is handled in user dashboard according to the quantity of fuel3. An option for accepting order is given which updates the status of order4. After accepting order pickup option appear which eventually updates the status of order5. Lastly after picking up order order delivered button appears which also updates the status of order6. An animation for accepting order is given in user dashboard	

Group member - 02	
Name: samiul huq ifty	Student ID: 21
Functional Requirements which are developed by this member:	
1. User/Driver/Admin edit profile 2. User can order varieties of fuel (like petrol, diesel, octane) 3. User can select the quantity of fuel (cannot order less than 3 if a user input a value which is less than 3 an invalid message will be pop out) 4. User can give the address of fuel station and confirm order 5. An animation for picking up order is showing in user dashboard 6. Status in user dashboard is updating in real time by accepting, picking up and order delivered by driver	

Group member - 03	
Name: Ahmed Tahlil Kadir	Student ID: 21101138
Functional Requirements which are developed by this member:	
1. User/Driver/Admin Signup 2. User/Driver/Admin Login 3. An order history has been shown where the detailed info of order is popping after confirming an order by user 4. A table regarding assigned order is showing where all the info about order is showing in driver dashboard (like order id, status, price, quantity, fuel type and userinfo)	

5.Notification is shown in user dashboard when the order is delivered