

Devansh

G-23 (2010991860)

# Gas Booking Website

12, February, 2023

## Overview

This project aims to build a web-based application for booking and managing gas deliveries for customers. The application will allow users to create an account, view their booking history, and place new orders for gas deliveries. The application will also have an admin panel for managing customer accounts and monitoring delivery orders.

## Technology Stack

The application will be built using the MERN stack (MongoDB, Express.js, React.js, and Node.js).

**MongoDB:** MongoDB will be used to store customer and order information.

**Express.js:** Express.js will be used as the back-end web framework for building the API endpoints for the application.

**React.js:** React.js will be used as the front-end framework for building the application's user interface.

**Node.js:** Node.js will be used as the runtime environment for the application.

## Outline Of The Project

The application will have two main parts: the customer-facing part and the admin part.

#### Customer-Facing Part:

- Users will be able to create an account.
- The user will be able to log in.
- Users have a user-dashboard which displays their information and recently placed orders.
- Users will be able to view their booking history.
- Users will be able to place new orders for gas deliveries.
- Users will be able to give feedback.

#### Admin Part:

- Admin will be able to view all customer accounts.
- Admin will be able to monitor delivery orders.
- Admin will be able to update the status of orders.
- Admin will be able to manage the inventory of gas.
- Admin is able to view user feedbacks on their dashboard.

#### Endpoints:

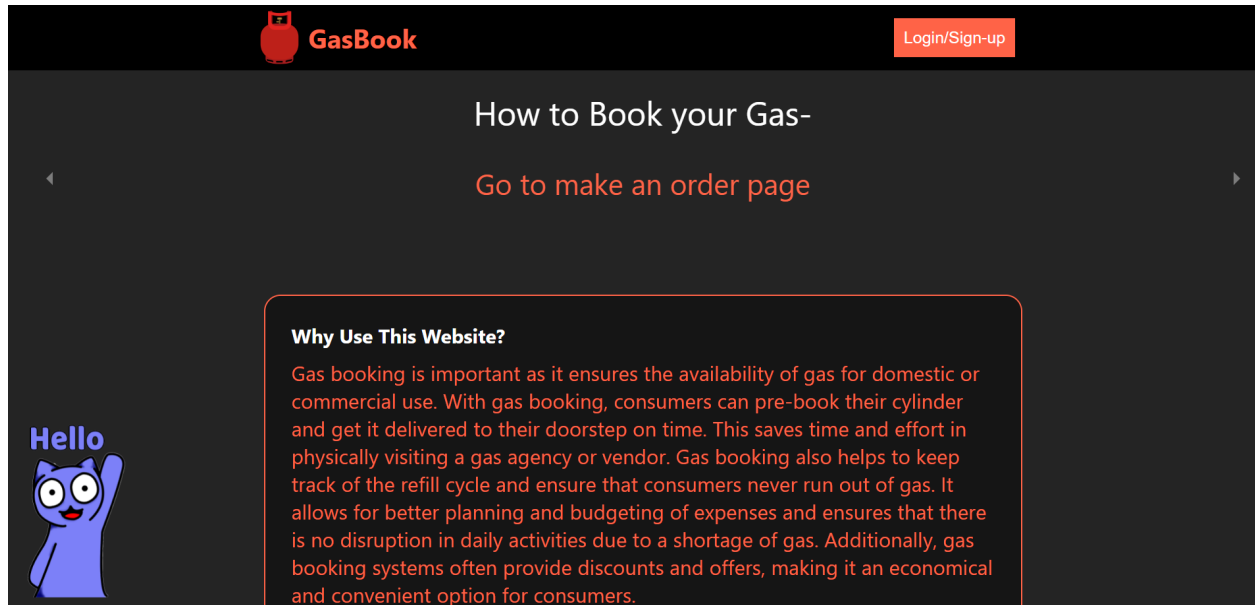
- **POST /register:** for customer registration
- **POST /login:** for customer login
- **GET /orders:** for fetching customer orders
- **POST /orders:** for creating a new order
- **PUT /orders/:id:** for updating order status
- **GET /admin/customers:** for fetching all customer
- **GET /admin/orders:** for fetching all orders
- **PUT /admin/inventory:** for updating the inventory
- **GET /feedback:** for fetching all the feedbacks

- **POST /feedback:** for creating a new feedback from a user

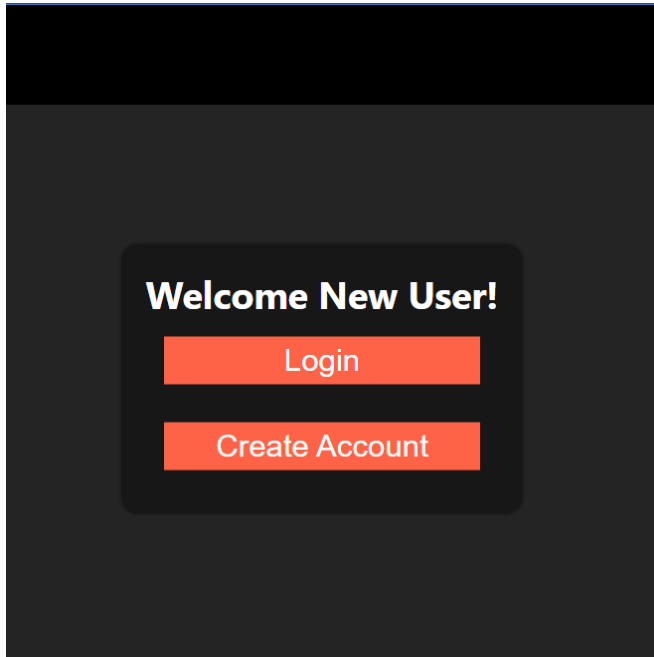
## Working Of The Website:

### General Part-

#### Home Page:



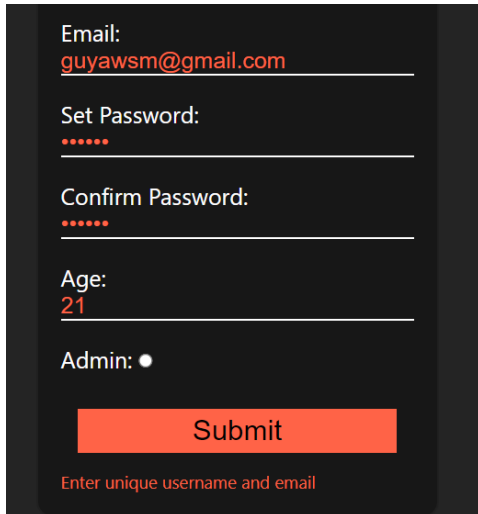
The simple home page contains a carousel, which guides people on how to book gas cylinders on the website. To access the user functions, one must make sure they are logged in, which is why the navbar currently has the Login/Sign-up button.



## Register Page:

A dark-themed registration form. The form is contained within a dark gray rectangular area. At the top of this area, the word "Register" is written in a bold, orange, sans-serif font and is underlined. Below this title are several input fields, each preceded by a label in a white, sans-serif font: "Full Name:", "User Name:", "Email:", "Set Password:", "Confirm Password:", "Age:", and "Admin:". Each label is followed by a white horizontal line representing the input field. The "Admin:" label is followed by a small white radio button. At the bottom of the form is an orange rectangular button with the word "Submit" in white, sans-serif font.

This is the common Register page, wherein a user can enter his/her details and create an account, the user name, and email fields must be **unique**, or the application throws a warning-



The image shows a registration form with the following fields and values:

- Email: guyawsm@gmail.com
- Set Password: .....
- Confirm Password: .....
- Age: 21
- Admin: ☐

Below the fields is a red "Submit" button. At the bottom, a warning message reads: "Enter unique username and email".

The password entered by the user is encrypted and then stored on MongoDB using a manually written encryption algorithm, which uses the Unicode of given characters in the password string, and a secret string, and then takes the XOR of the two to create an encrypted string. Example-

```

    ault in Mongoose 7. Use `
    e.set('strictQuery', true
    (Use `node --trace-depre
    http://localhost:4000/
    connected with mongo!
    qwerty
    ↕↕↕L▶
    qwerty
    ↕↕↕L▶
    []
  
```

Here the string qwerty

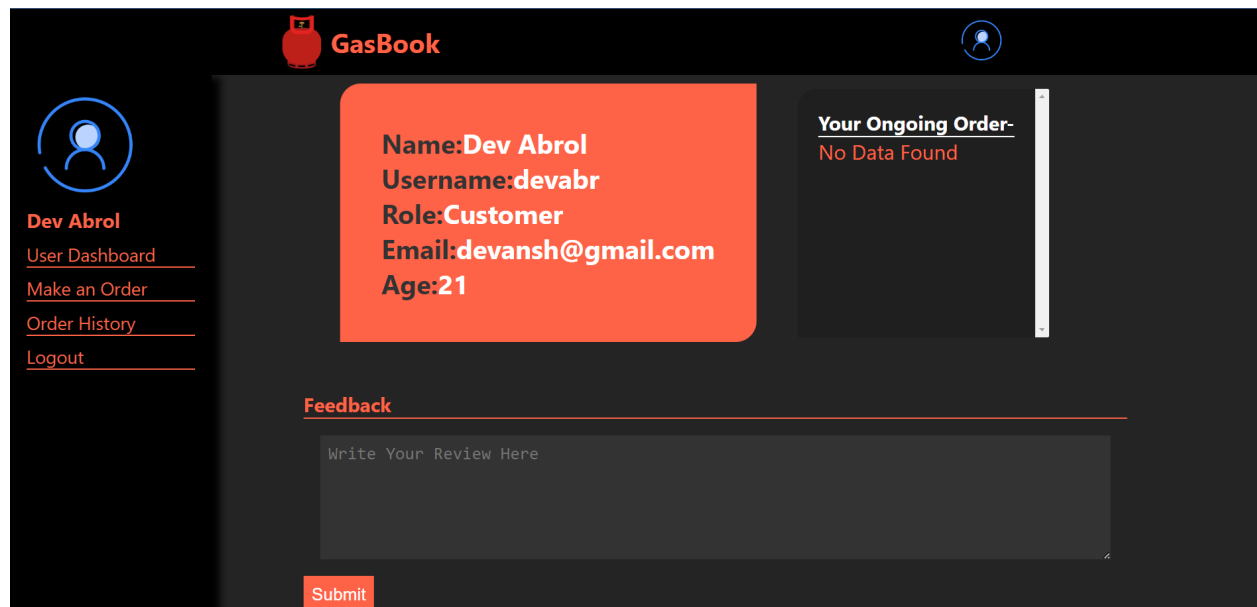
(the password here) gets encrypted, before being stored on MongoDB.

Login Page:

The simple login page checks user credentials and returns a message if any of the credentials entered are wrong. Once logged in, the username, and his/her role is saved in the local storage so that when they enter the site again, they remain logged in already. Once they log out, their information is also deleted from local storage.

## User Side-

### User Dashboard-



The user dashboard contains three things- User information, their recently placed orders (which aren't delivered yet), and a feedback form.





To make an order, one must fill out the form providing their details like Address, and select the weight of the cylinder they wish to order. The process doesn't proceed, if the user hasn't made any payments, and displays a message until the payment is made-

**Booking Details**

Date:  
2023-2-12

Cylinder Weights:  
15.9kg

Rs. 1100 **Make Payment**

**Place Order!**

Please make payment before order

**GasBook**

**User Details**

Name:  
Dev Abrol

UserId:  
devabr

Address:  
14 Bharat Nagar

**Booking Details**

Date:  
2023-2-12

Cylinder Weights:  
15.9kg

Rs. 1100 **Make Payment** ✓

**Place Order!**

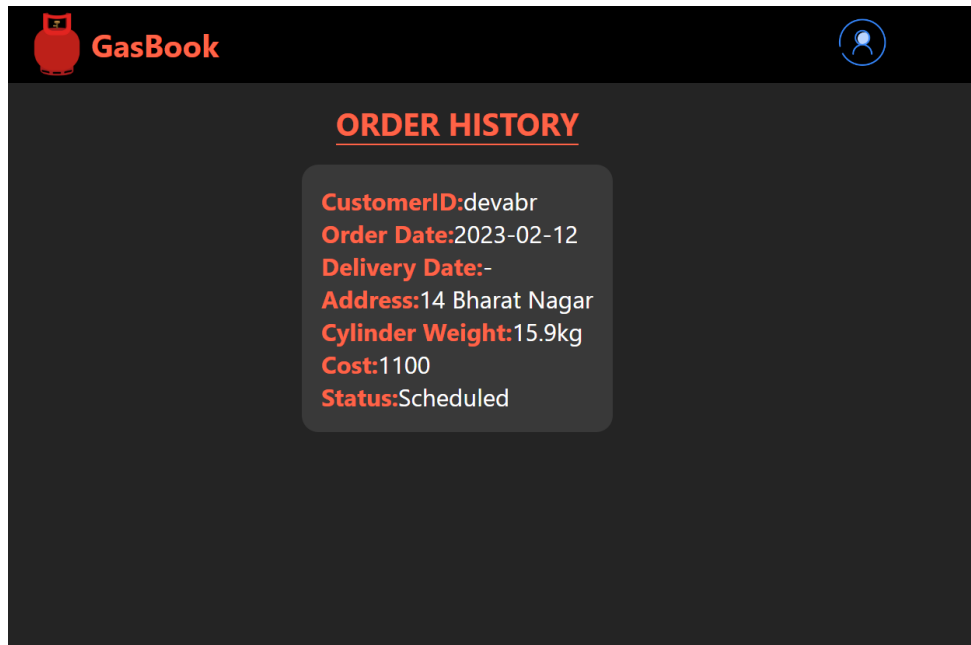
localhost:3000 says  
order placed!

OK

Once, the payment is made, the user can make the order, and an alert box gets displayed.

Order History-

The user can view their order history to gather information on all the orders they made in the past. The orders which aren't been delivered yet display no delivery date and are status-scheduled.



## Admin Part-

### User Dashboard:

The user dashboard is almost the same as that of the user, with a few differences that the admin has no current orders placed, and they can view all feedback given by various users-

**GasBook**

**Devansh Abrol**  
User Dashboard  
All Orders  
Customers  
Inventory  
Logout

**Name: Devansh Abrol**  
**Username: devansh1503**  
**Role: admin**  
**Email: guyawsm@gmail.com**  
**Age: 21**

**Your Ongoing Order-**  
No Data Found

**User Feedbacks-**

- rishabh007**  
Please make the orders a bit fast
- devabr**  
The order is getting too late!

## Monitor Orders:

The admin can monitor orders, update their status to delivered and update the delivery date on the current day (which is assumed to be their delivery date)-



**GasBook**

Customer ID	Order Date	Delivery Date	Cylinder Weight	Address	Cost	Status
rishabh007	2023-02-08	2023-02-08	90kg	Zirakpur, Punjab	400	Delivered
dipanshu1866	2023-02-10	2023-02-10	210kg	Ganganagar	700	Delivered
dipanshu1866	2023-02-10	2023-02-10	210kg	Gandhinagar	700	Delivered
rishabh007	2023-02-11	Update for today	90kg	Zirakpur, Punjab	400	Update Status
devanshabr	2023-02-12	2023-02-12	90kg	14 Bharat Nagar Ferozepur	400	Delivered
devabr	2023-02-12	Update for today	15.9kg	14 Bharat Nagar	1100	Update Status

devabr	2023-02-12	2023-02-12	15.9kg	14 Bharat Nagar	1100	Delivered
--------	------------	------------	--------	-----------------	------	-----------

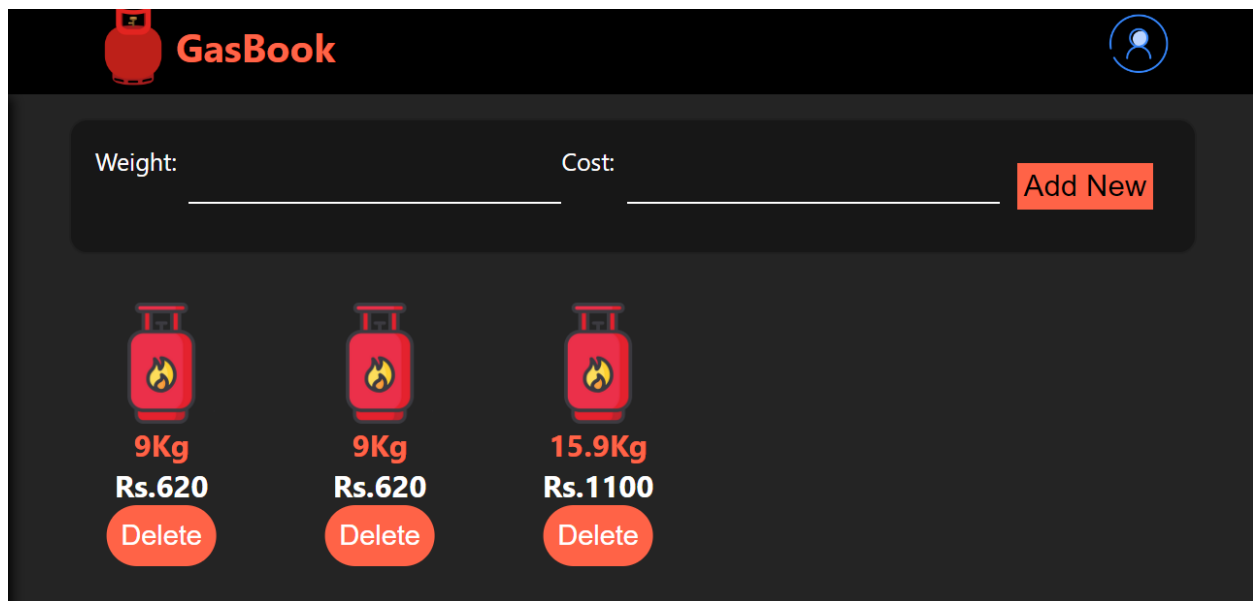
### View All Customers-

The admin is able to view all the customers and their data (except their passwords)-

 <b>GasBook</b> 			
Full-name	Age	Email	User-name
Rishabh Arora	19	rishabh@gmail.com	rishabh007
Dipanshu Gupt Maurya	21	dipanshu@gmail.com	dipanshu1866
Raghu Ram	43	raghuram@gmail.com	andrewtate_69
Dev Abrol	21	devansh@gmail.com	devabr

### Inventory-

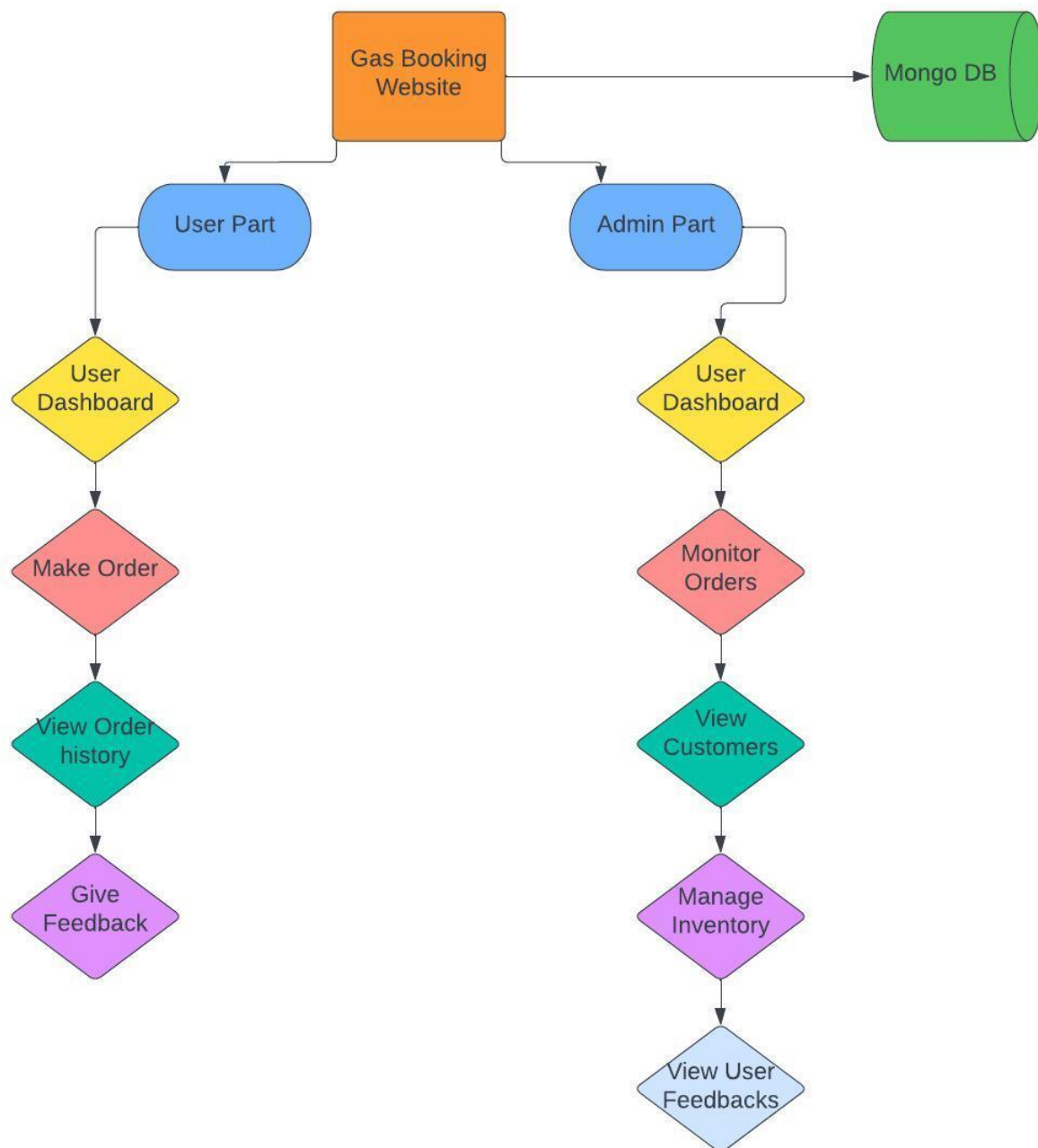
The admin can update the inventory and view all the cylinders in inventory, they can manually update the number of cylinders by adding their weights and costs as per the orders monitored by them-



The image shows a mobile application interface for "GasBook". At the top, there is a header bar with a red gas cylinder icon and the text "GasBook" in orange. To the right of the header is a circular profile icon. Below the header, there is a dark gray section with two input fields: "Weight:" and "Cost:", each followed by a horizontal line. To the right of these fields is an orange button labeled "Add New". Below this section, there is a list of three gas cylinders, each represented by a red cylinder icon with a yellow flame. Under each icon, the weight and cost are displayed in orange and white text, respectively. Below the cost is an orange button labeled "Delete".

Weight	Cost	Action
9Kg	Rs.620	Delete
9Kg	Rs.620	Delete
15.9Kg	Rs.1100	Delete

Flow Diagram:



## Conclusion:

The Gas Booking Web Project is a platform that will make the process of buying gas more convenient and efficient for users. The platform's user-friendly interface and various features will make it easy for users to book and purchase gas online. The platform will be built to the highest standards of security and performance and will be regularly maintained to ensure that it remains reliable and efficient.