**Mini Project Documentation**

**Title of the Project – *Super Gas***

**Name: K. YASWANTH KUMAR**

**Regd number: 122010322024**

**Batch number of the student: 18**

1. ***Requirement Analysis/Abstract of the Project***

***Description:***

In India most of us use gas for cooking. There are many companies like Hindustan Petroleum, Bharat Petroleum, Super gas are some of the prominent suppliers in India. In the pre internet era, people use to book the gas refill by physically going to the nearest dealer. But today they are offering you the flexibility and convenience of booking your refill cylinder at anytime, from anywhere, when you are on the move, on a holiday or at home through multiple modes:

The consumer has a Unique number and it is connected with a cell number.

***Abstract:***

* **The abstract of our project is to book the gas cylinder in India through online.**
* **With the help of our super gas website, anyone can book the gas cylinder from anywhere and at anytime.**
* **Easy supervision by the admin.**

**REQUIREMENT ANALYSIS:**

**Entities:**

1. **Customer**
2. **Booking**
3. **Payment**
4. **Bill**

**Attributes:**

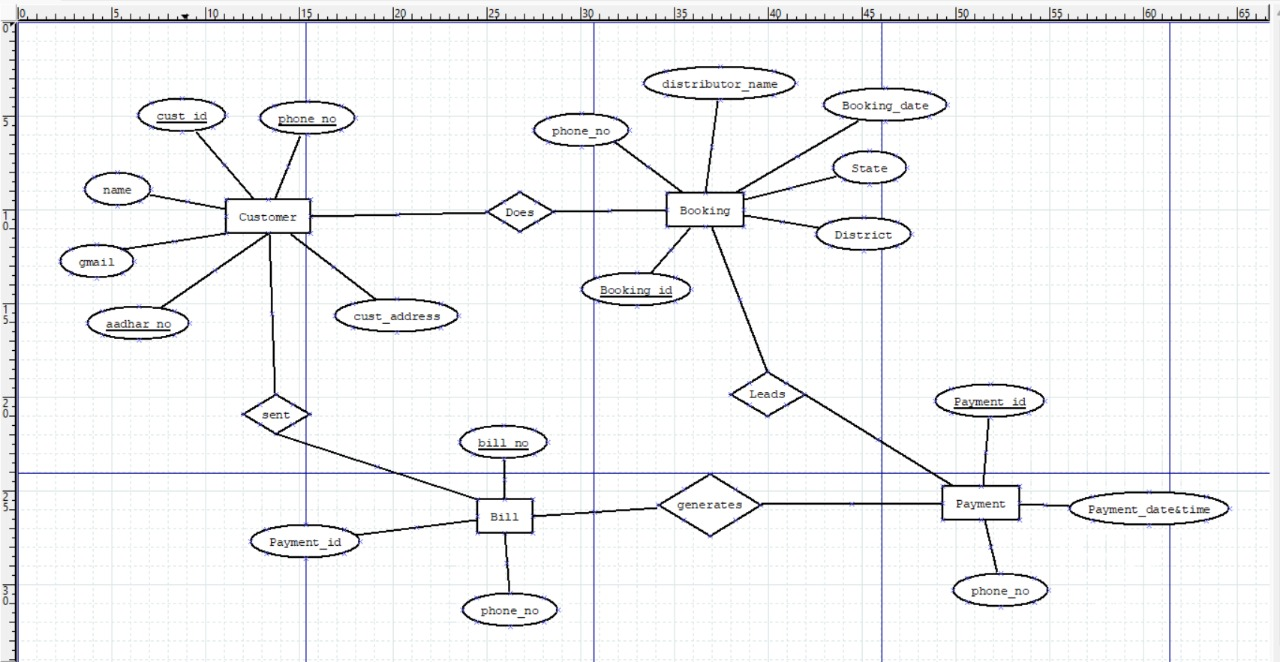
* **Customer – Cust\_ID(Primary key), Name(Not Null), Gmail(Not Null), Phne\_Number(Primary key) ,Aadhar\_number(unique), Address(Unique).**
* **Booking – Booking\_ID(primary key), State(Not Null), District(Not Null), Phone\_Number(Foreign key),Distributor \_Name(Not Null), Booking\_date(Not Null).**
* **Payment – Payment\_ID(Primary key), Payment\_Time(Not Null),Phne\_Number(Foreign key).**
* **Bill - Bill\_Number(Primary key), payment\_ID(Foreign key),Booking\_ID(Foreign key),Phne\_Number(Foreign key).**

|  |  |  |
| --- | --- | --- |
| **TABLES** | **PRIMARY KEYS** | **FOREIGN KEYS** |
| **Customer** | Cust\_ID,Phne\_Number | - |
| **BOOKING** | Booking\_ID | Phne\_Number |
| **Payment** | Payment\_ID | Phne\_Number |
| **BILL** | Bill\_Number | Phne\_Number,Payment\_ID,Booking\_ID |

**Relationships:**

* **Customer will do booking**
* **Booking leads to payment**
* **Payment generates bill**
* **Bill sent to customer**

1. ***ER diagram for Super Gas:***



1. ***Conceptual Schema***

**Customer:**

**CREATE TABLE IF NOT EXISTS customer(  
 cust\_id int not null,  
 name varchar(20) not null,  
 phno int not null,  
 aadhar\_no int not null,  
 gmail varchar(40) not null,  
 primary key (phno)  
 )**

**Payment :**

**CREATE TABLE IF NOT EXISTS payment\_details (  
 payment\_id varchar(30) not null,  
 payment\_date varchar(30) not null,  
 phno int not null,  
 FOREIGN KEY(phno) REFERENCES customer(phno)  
 )**

**Booking:**

**CREATE TABLE IF NOT EXISTS booking\_details (  
 state varchar(30) not null,  
 district varchar(30) not null,  
 distributor\_name varchar(30) not null,  
 booking\_id int unique,  
 booking\_date varchar(40) not null,  
 phno int not null,  
 FOREIGN KEY(phno) REFERENCES customer(phno)  
 )**

1. ***Logical Database Design***

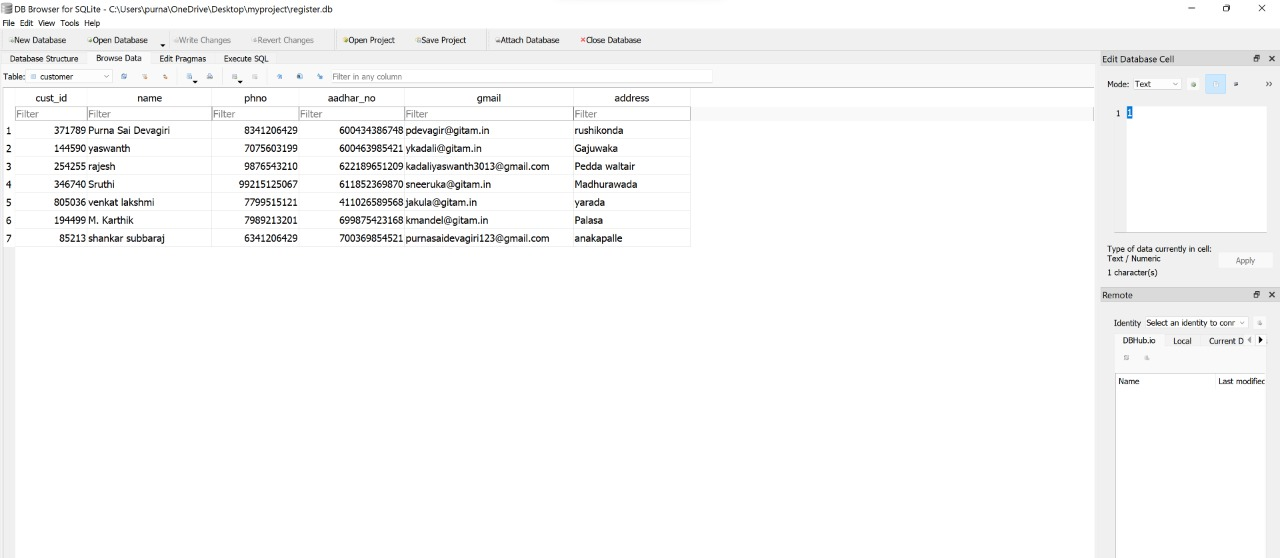
* **Values stored in database in each table:**

1. **Customer Table:**

**Code:**

**def login():  
 global phno  
 global cust\_id   
 global name  
 global gmail  
 if request.method == "GET":  
 cust\_id = random.randint(10000,999999)  
 with sqlite3.connect("register.db") as con:   
 con.execute('''CREATE TABLE IF NOT EXISTS customer(  
 cust\_id int not null,  
 name varchar(20) not null,  
 phno int not null,  
 aadhar\_no int not null,  
 gmail varchar(40) not null,  
 primary key (phno)  
 )''')  
 con.commit()  
 with sqlite3.connect("register.db") as con:   
 con.execute('''CREATE TABLE IF NOT EXISTS payment\_details (  
 payment\_id varchar(30) not null,  
 payment\_date varchar(30) not null,  
 phno int not null,  
 FOREIGN KEY(phno) REFERENCES customer(phno)  
 )''')  
 con.commit()  
 return render\_template('dbms\_pg1.html')  
 else:  
 name = request.form['name']  
 phno = request.form['phno']  
 aadhar = request.form['aadhar']  
 gmail = request.form["gmail"]  
 with sqlite3.connect("register.db") as con:  
 con.execute(  
 "INSERT into customer (cust\_id,name, phno , aadhar\_no,gmail) values (?,?,?,?,?)“**

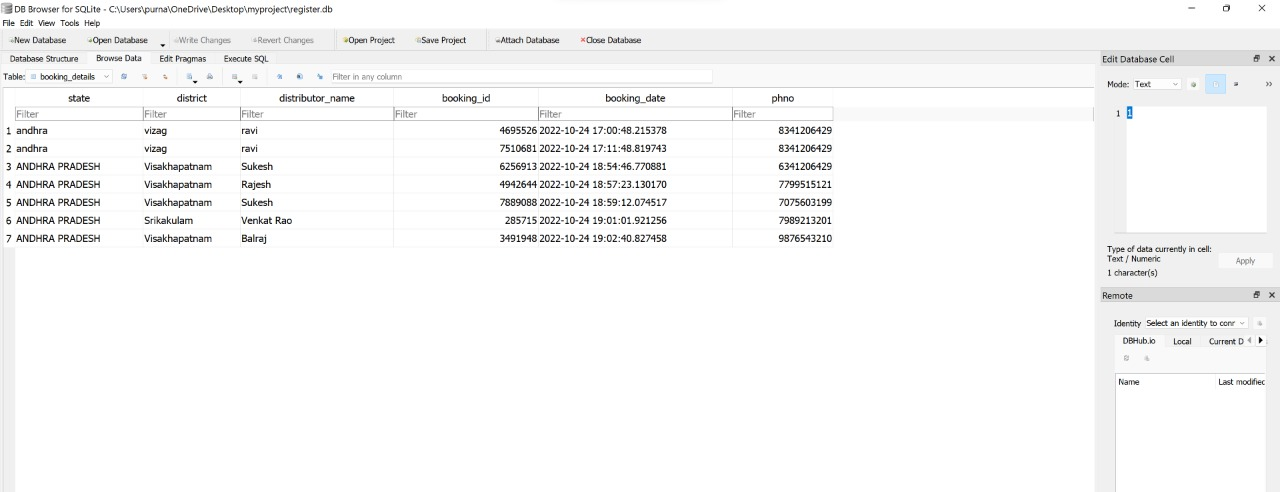
**,(cust\_id,name, phno, aadhar, gmail))  
 mail.email(gmail,cust\_id)  
 return render\_template('dbms\_pg1.html')**

****

1. **Booking Table:**

**Code:**

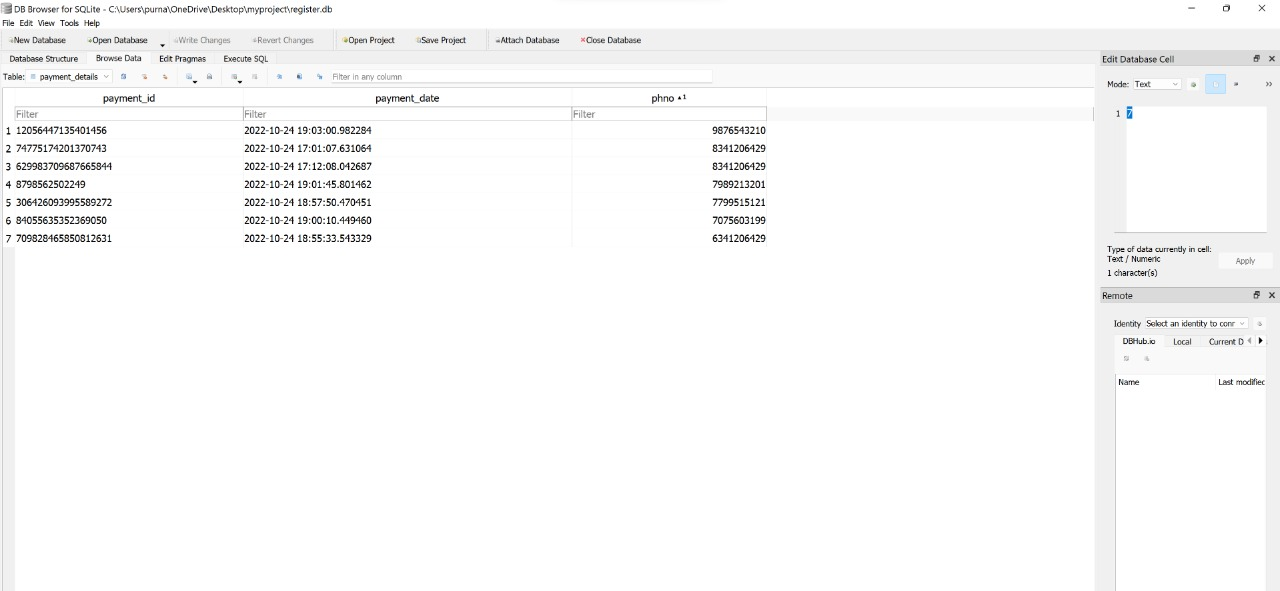
**def quick\_book():  
 if request.method == "GET":  
 with sqlite3.connect("register.db") as con:   
 con.execute('''CREATE TABLE IF NOT EXISTS booking\_details (  
 state varchar(30) not null,  
 district varchar(30) not null,  
 distributor\_name varchar(30) not null,  
 booking\_id int unique,  
 booking\_date varchar(40) not null,  
 phno int not null,  
 FOREIGN KEY(phno) REFERENCES customer(phno)  
 )''')  
 con.commit()  
 return render\_template('quick\_pg4.html')   
 else:  
 return render\_template('pay\_pg8.html')  
  
  
@app.route("/payment",methods = ["post"])**

****

1. **Payment Table:**

**Code:**

**def payment():  
 booking\_id = random.randint(100000,9999999)  
 if request.method == "post":  
 state = request.form['state']  
 district = request.form['district']  
 distributor\_name = request.form['distributor']  
 with sqlite3.connect("register.db") as con:  
 con.execute(  
 "INSERT into booking\_details(state, district , distributor\_name, booking\_id,booking\_date, phno) values (?,?,?,?,?,?)",(state, district, distributor\_name, booking\_id, datetime.now(),phno))  
 return render\_template("pay\_pg8.html")  
 else:  
 state = request.form['state']  
 district = request.form['district']  
 distributor\_name = request.form['distributor']  
 with sqlite3.connect("register.db") as con:  
 con.execute(  
 "INSERT into booking\_details(state, district , distributor\_name, booking\_id,booking\_date,phno) values (?,?,?,?,?,?)",(state, district, distributor\_name, booking\_id, datetime.now(),phno))  
 return render\_template("pay\_pg8.html")  
  
  
  
@app.route("/subsidy")**

****

***TECHNOLOGIES USED:***

* **In Frontend**
* **Html(hypertext markup language) – used for content writing**
* **CSS(cascading style sheets) - used for styling**

**HTML:**

* **In html, we used many tags like:**
* **<h1></h1> tag for heading – heading tag**
* **<div></div> tag for combining into a container – division tag**
* **<p></p> tag for writing a paragraph – paragraph tag**
* **<button></button> tag for creating a button – button tag**
* **<form></form> tag for taking the inputs from user – form tag**
* **<a></a> tag for inserting links – anchor tag**
* **<br></br> used for breaking the line – Line break tag**
* **<hr></hr> used to draw a horizontal line in between – horizontal rule tag**
* **<span></span> tag is an inline container used to mark up a part of a text. – span tag**
* **<label></label> tag is used to provide a usability improvement for mouse users – label tag**
* **<body></body> tag for the whole body of html – body tag**
* **<html></html> tag for the whole page of html – html tag**
* **In html, we used html void elements like:**
* **<img src =””/> used for image – image void element (A void element is a html tag which has no end tag, it specifies both in start tag itself)**

**CSS:**

1. **For text styling:**
   * + **font-size**
     + **font-family**
     + **font-weight**
     + **color**
2. **For background styling:**

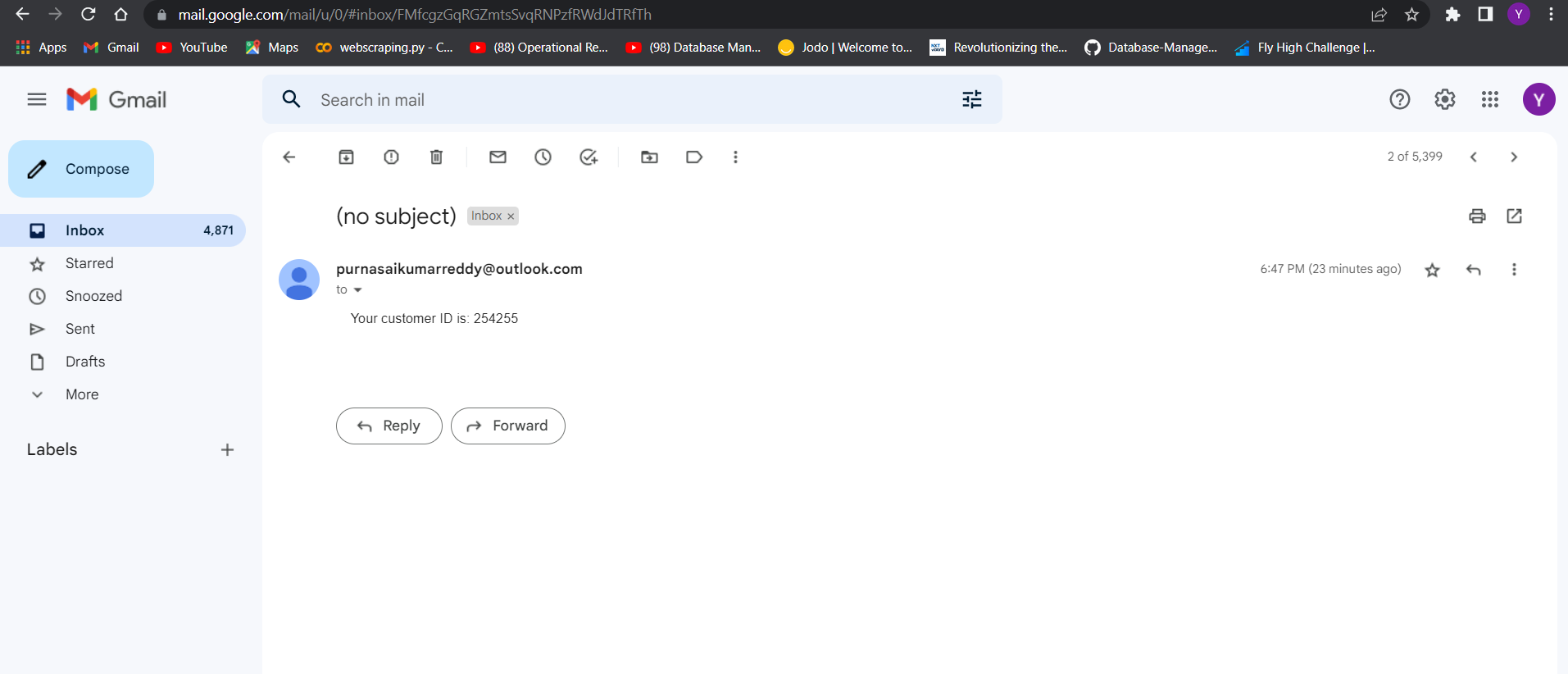
* **Background-image**
* **Background-size**
* **Background-color**

1. **For button styling:**

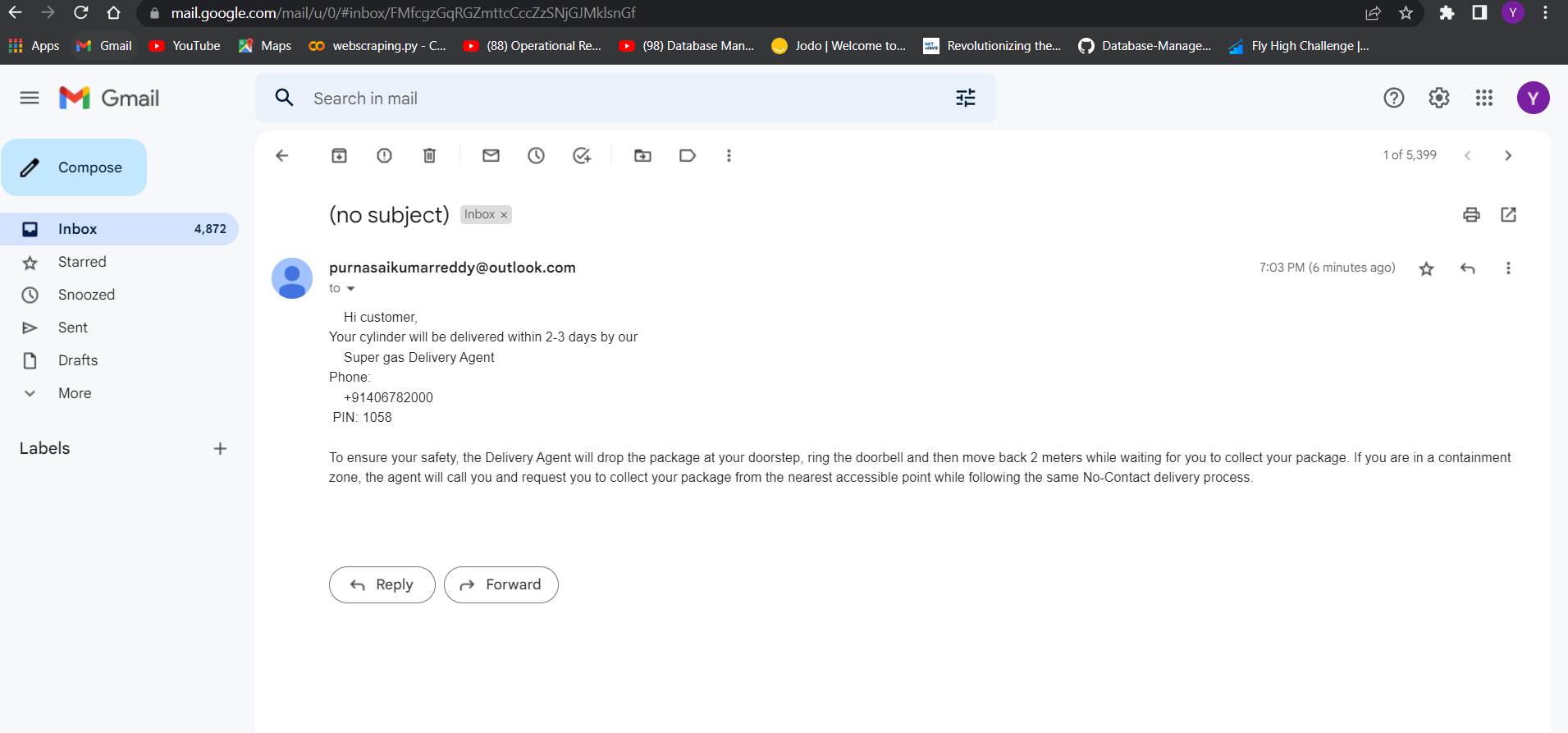
* **Width**
* **Height**
* **Color**
* **Border-radius**
* **Border-outline**
* **Border-color**
* **We also used bootstrapping, bootstrap provides reusable code snippets like**

**d-flex flex-row justifify-content-center**

* **Backend**
* **In backend, we imported flask web framework to connect with frontend**
* **We also used cursors to retrieve the tuples**
* **We used @app.route routers to move to next page of website**
* **We used sqlite3 for database**
* **We used render template to render specific data**
* **We used GET and POST methods , as GET is used to retrieve the data from database and POST is used to insert the data into database**
* **We also used random module to randomly generate transaction id during payment**
* **We imported smtplib to sent the mail for OTP after verification. Login(), startls() functions are used from smtplib**
* ***Steps to use our Website:***
* **Firstly, we will open our website link in the browser. Then our login page will be displayed.**
* **If the user is existing user, he can login with his phone number and customer id**
* **Otherwise, the customer should click on the new user button which redirects to the signup page of our website, where the customer needs to provide the customer details like name, phone number, gmail, aadhar\_number, address.**
* **When the customer fills all the above details and clicks on sign up button then he/she will receive the unique customer id to his/her email as shown in the below image.**

****

* **Using this customer id and his/her phone number, customer can login to our website to book the gas cylinder.**
* **For booking the cylinder he needs to click on the quick and pay button, so that it will be redirected to the booking page.**
* **Customer should fill the booking details like state,district,distributor name.. to proceed.**
* **Once he/she clicks on the proceed button,which redirects to the payment page.**
* **The customer should provide the payment details like customer name, card number, expiry date, cvv.. to make the payment.**
* **After clicking on the pay button, if the payment details are correct, After payment is successful, a unique transaction id will be generated.**
* **The customer will receive a confirmation mail which consists of Delivery agent number, pin, booking id, customer phone number, payment id… as shown in the below image.**

****

**The full code is available at the below link**

**Github link: https://github.com/yaswanth3013/DBMS\_MINI\_PROJECT\_FILES.git**