Indian Institute of Technology Gandhinagar

Osborg

Assignment-3 Report

CS 432: Databases

Date:20/03/2023

Group Members

Vishal Ghoniya: 20110229

Dhruv Patel: 20110129

Prakram Rathore: 20110141

Meet Vankar: 20110112

Juhil Aslaliya: 20110030

Pratik Raj: 20110144

Rahul Kumar: 20110153

Tejas Parmar: 20110125

Dablu kumar: 20110049

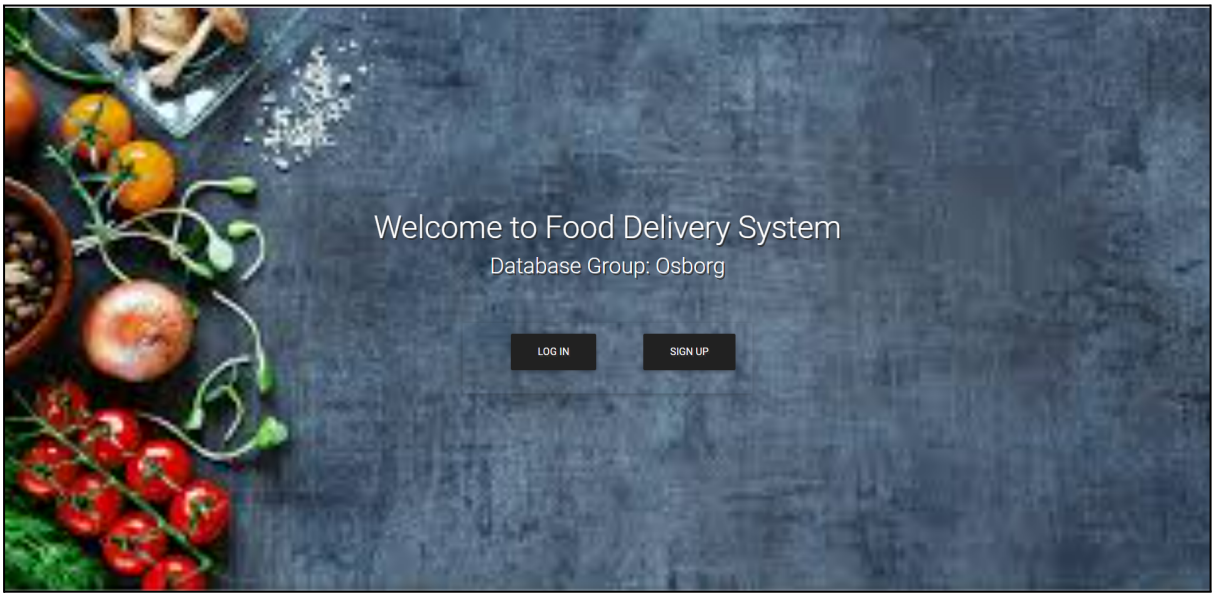
Prey Patel: 20110132

Dhiren Agarwal: 20110057

Github_repo link: https://github.com/Vishal987595/fooddeliverysystem
Video_link:
https://drive.google.com/file/d/1BrPiArpvNygDpygwt5aaDxWbDvjJVzE1/view?usp=sharing

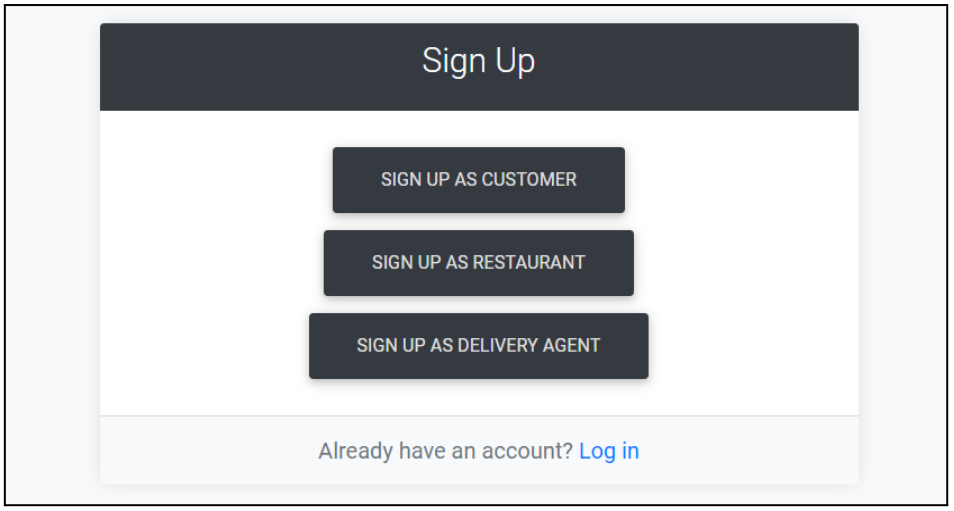
1 : Main and Standard Pages

The first page of our frontend, for login and sign-up for any stakeholder looks like as shown below. Two different buttons are given for login and sign-up purposes.

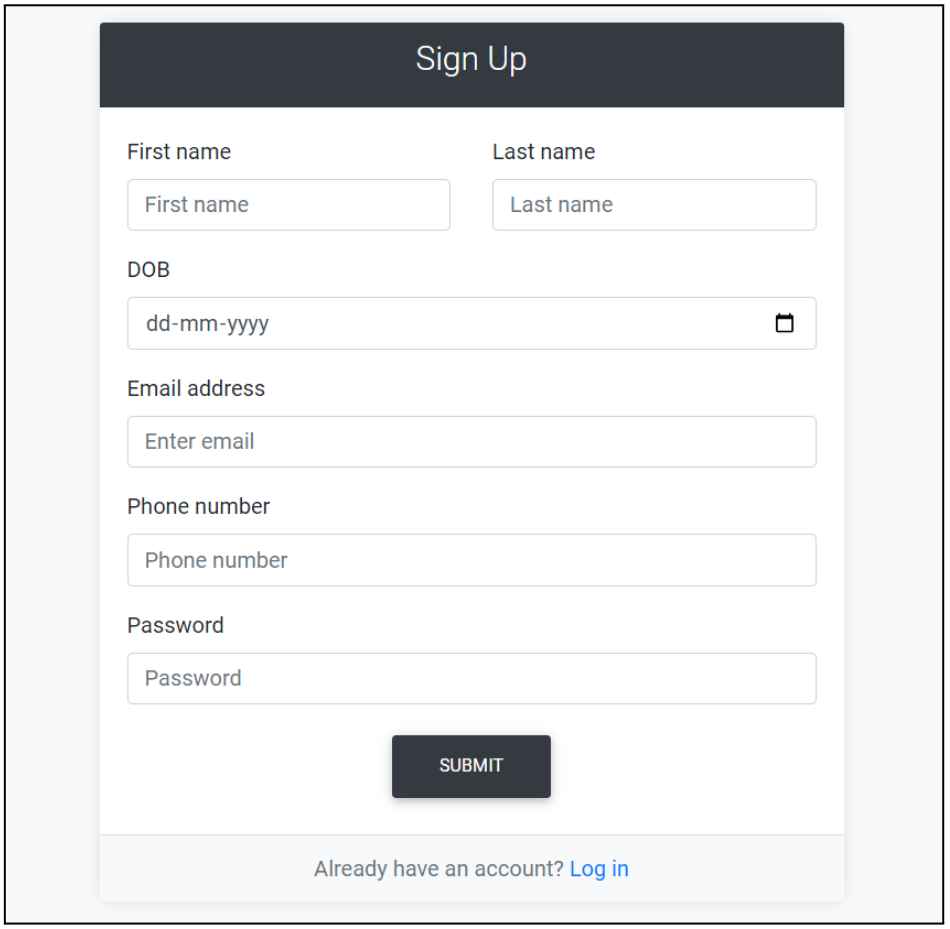



```
@app.route('/home')
def home():
    session.clear()
    return render_template('index.html')
```

The home page for our system looks as follows. When we come to the home page we perform "session.clear()" which clears up all variables we set up in a session like restbool, customerbool, agentbool. We use these variables for user authentication on different pages of our web app wherever we feel necessary.
The sign-up page for our system looks as follows. We could move forward with any of the options. Each button takes the user to their respective sign up page.
Note: If it doesn't work on one click. Go to home and refresh the home page and retry it.



Filling up signup details. All the mentioned fields are mandatory. Each signup page is customized as per the user authorization. Like if it wants to sign up as a Customer and the page will be shown accordingly.

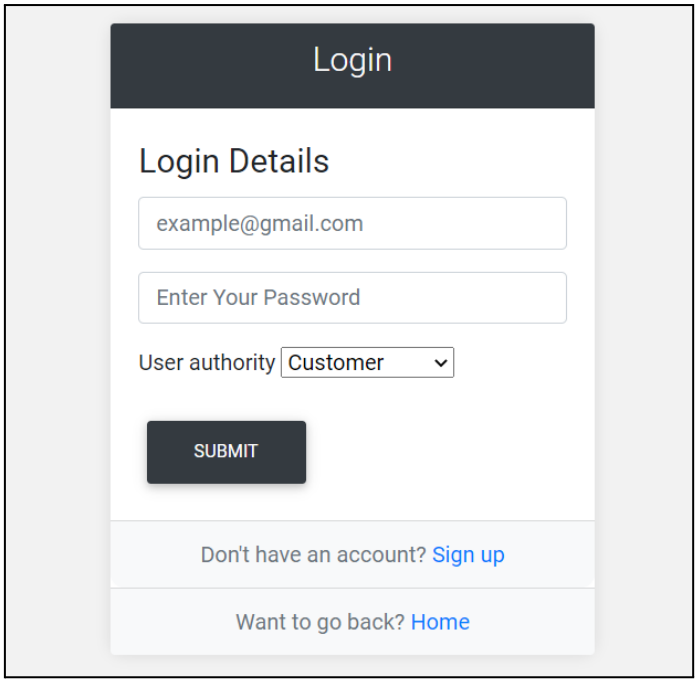


```python
@app.route('/signupcustomer',methods=['GET', 'POST'])
def signupcustomer():
    if request.method == 'POST':
        msg = 'CUSTOMER: Please fill out the form again'
        userdetails = request.form
        firstname = userdetails['firstname']
        lastname = userdetails['lastname']
        email = userdetails['email']
        DOB = userdetails['DOB']
        phone_number = userdetails['phone_number']
        password = userdetails['password']
        cur = mysql.connection.cursor()
        cur.execute("select max(customer_ID) from customers")
        ID = cur.fetchone()
        ID = str(int(ID[0]) + 1)
        cur.execute("insert into customers(customer_ID, first_name, last_name, email, phone
        mysql.connection.commit()
        cur.close()
        msg = 'CUSTOMER: sigup successfully!!'
        flask.flash(msg)
        return redirect(url_for('login'))
    return render_template('customersignup.html')
```

We establish the connection and execute our insert query and commit the changes and close the connection. We write up the flash message and redirect the user to the login page. This is POST request. When there is GET request we just render the page.

In the following image, it is the login page. User needs to select the user authority and based on that we look into the corresponding table to check if any such account with provided password and user email exists or not.

If the user authority is a restaurant or delivery agent they will be redirected to their respective user details page if successfully logged in. If user authority is a customer then it is redirected to the dashboard page. If login fails they need to re-enter the details.




```python
@app.route('/login', methods=['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST' and 'useremail' in request.form and 'password' in request.form and 'authority' in request.form:
        useremail = request.form['useremail']
        password = request.form['password']
        authority = request.form['authority']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        if (authority == "Customer"):
            cursor.execute("SELECT * FROM Customers WHERE email = % s AND password = % s", (useremail, password,))
            account = cursor.fetchone()
            if account:
                session['customerbool'] = True
                session['restbool'], session['agentbool'] = False, False
                session['customer_ID'] = str(account['customer_ID'])
                msg = 'Logged in successfully !'
                flask.flash(msg)
                return redirect(url_for('customer.dashboard'))
            else:
                msg = 'Incorrect username / password !'
```

Login page has 2 methods get and post:

When the login details are submitted i.e., username, password and authority (customer/restaurant/delivery agent). We add several boolean values in session which are checked wherever necessary to validate the user authority and provide response accordingly.

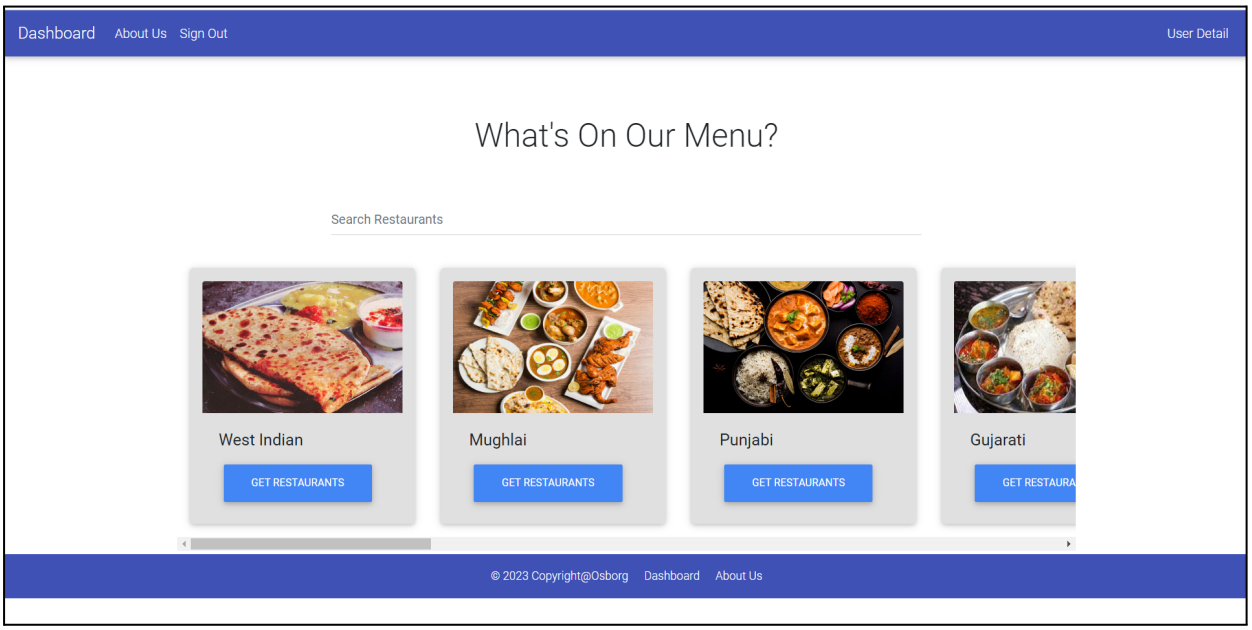
Here in the above pic, for customers we set the value of ‘customerbool’ as ‘True’ and ‘restbool’, ‘agentbool’ as ‘False’.

We also record their ID as per the authority so that we could easily get access to their details.

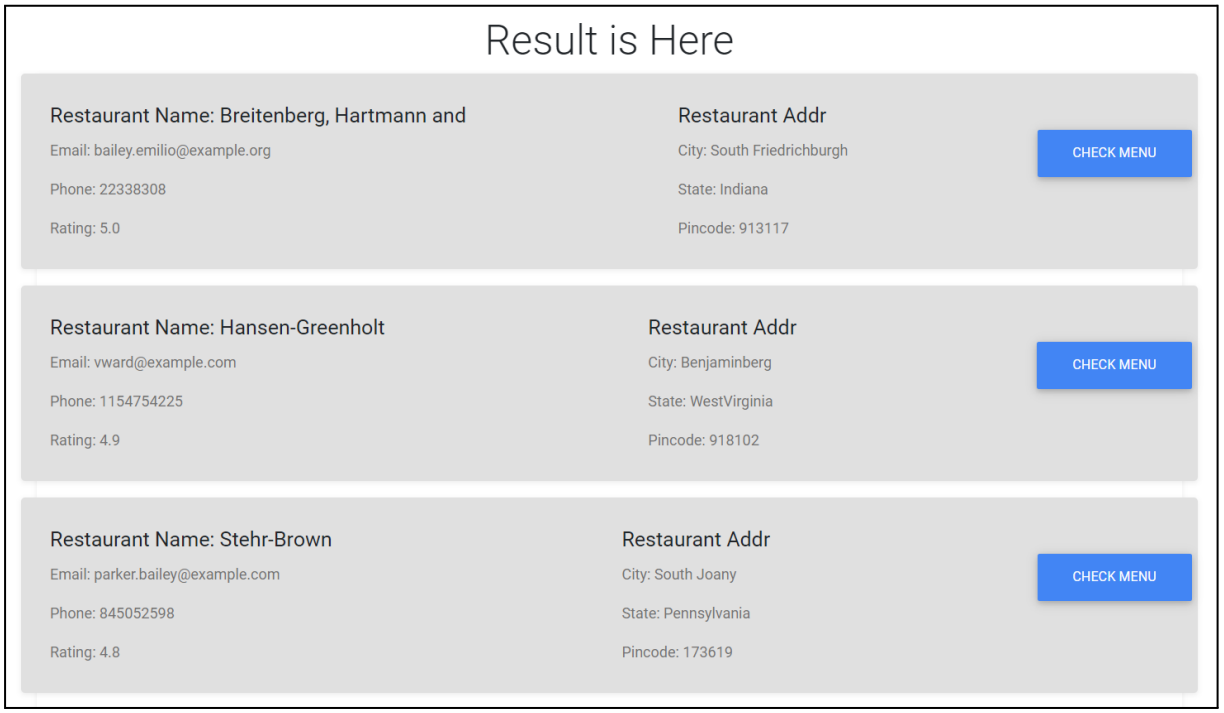
Next step, we put up a message and redirect to the dashboard of Customer blueprint where this flash message shows up for about 5 sec if logged in successfully. Else we again get the login page and show an alert with a message of details incorrect.

2 : Customer Pages

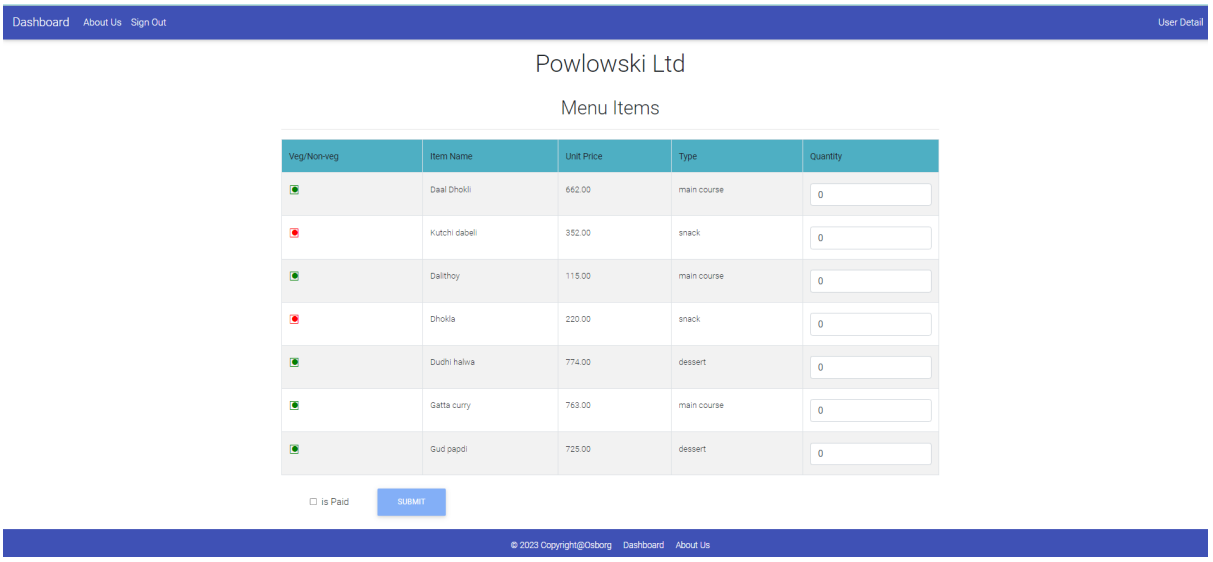
Dashboard for customers. It contains a search bar and cuisine types being served by restaurants.



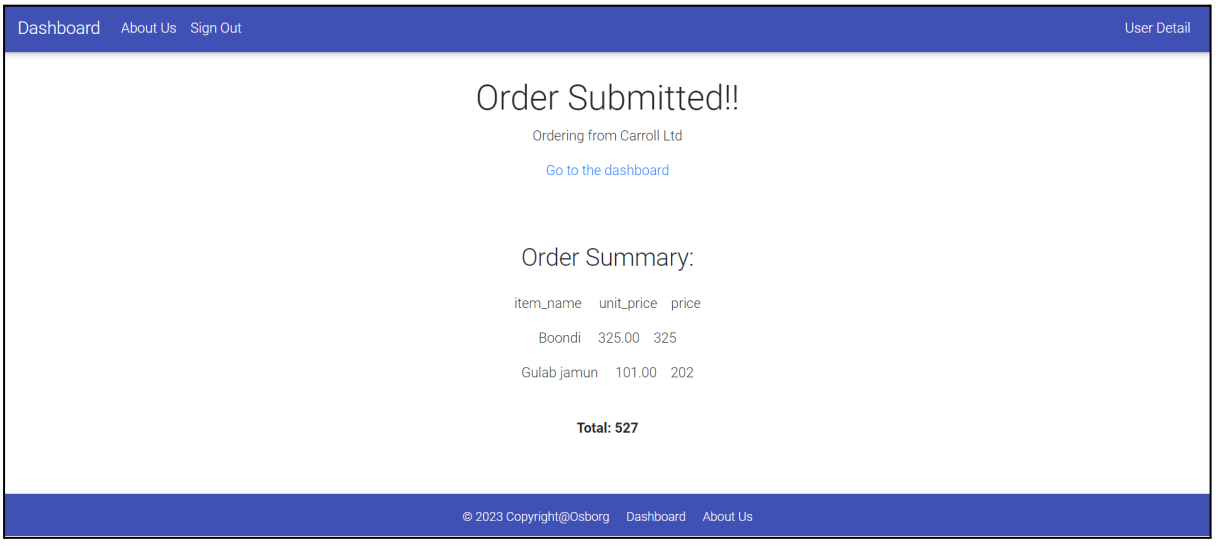
Upon selecting any category from the previous page (Customer Dashboard) by clicking on the "Get Restaurants" button or by making any search for the restaurant's name. We are directed to the "restlist" page. [WHERE clause]

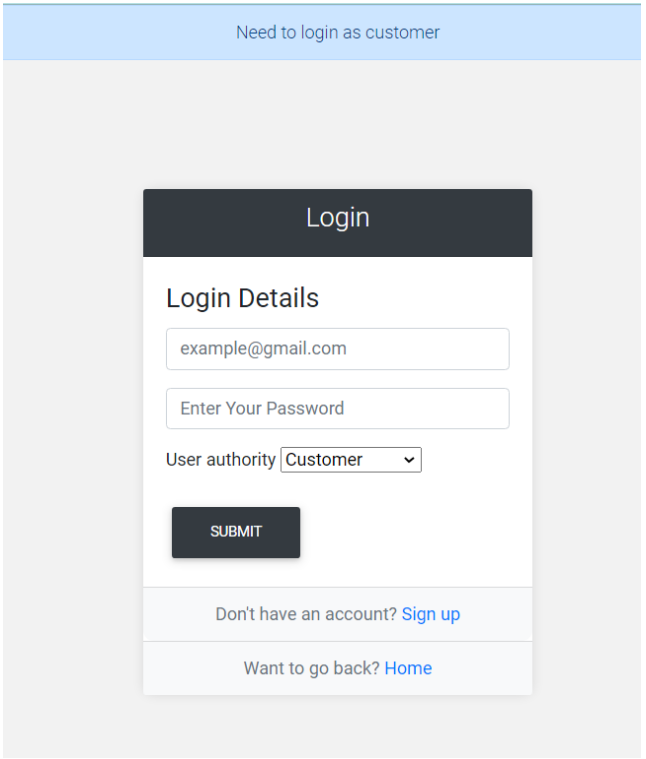


Click on “Check Menu” in the above page, of the restaurant that you like to check the menu of. You will get to see all the menu-items below. Here you can check “Veg/Non-veg”, “Item Name”, “Per Unit Price” of the item, “Type” of item. Then, you have to change the “Quantity” field of the items that you want to order.
Quantity is constrained from zero to 20 (for now).

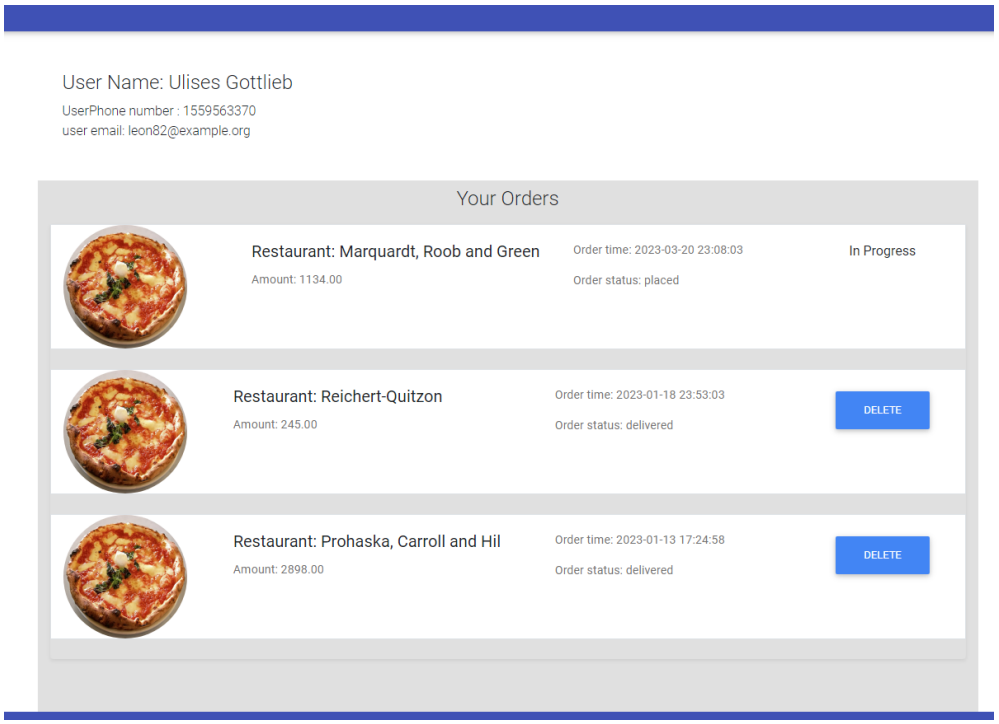


Menu page has a submit button which can be clicked only after the “is Paid” checkbox is ticked.
After clicking on “Submit”, you will be shown the order confirmation page, if your order is placed successfully else you will be redirected to the login page if placing order through user authority other than customer. **(INSERT)**

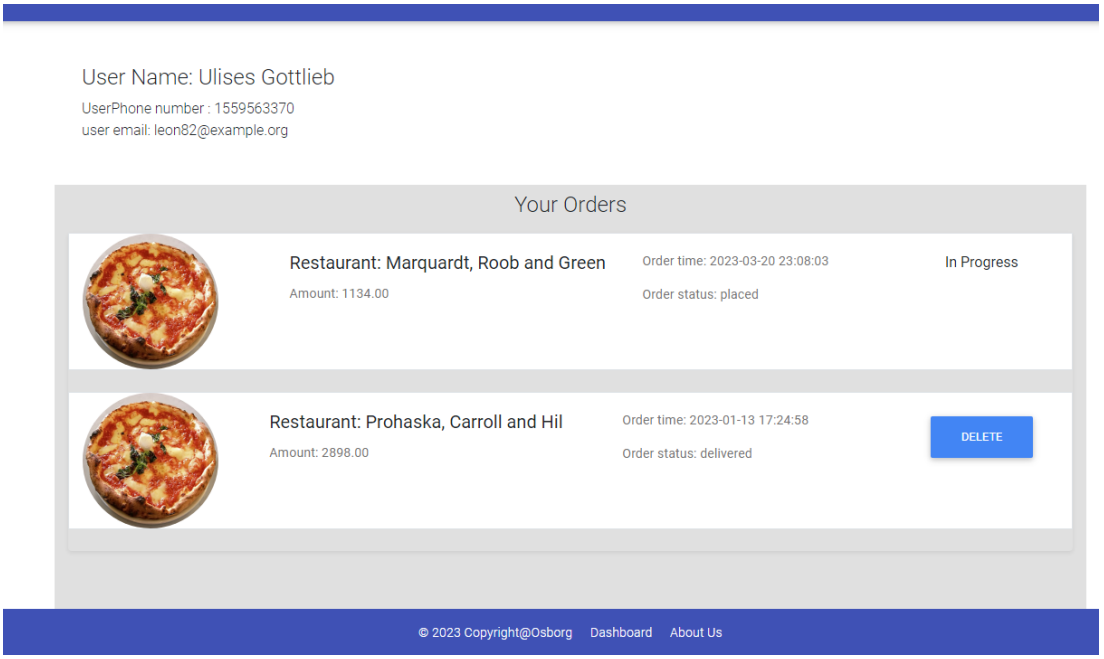




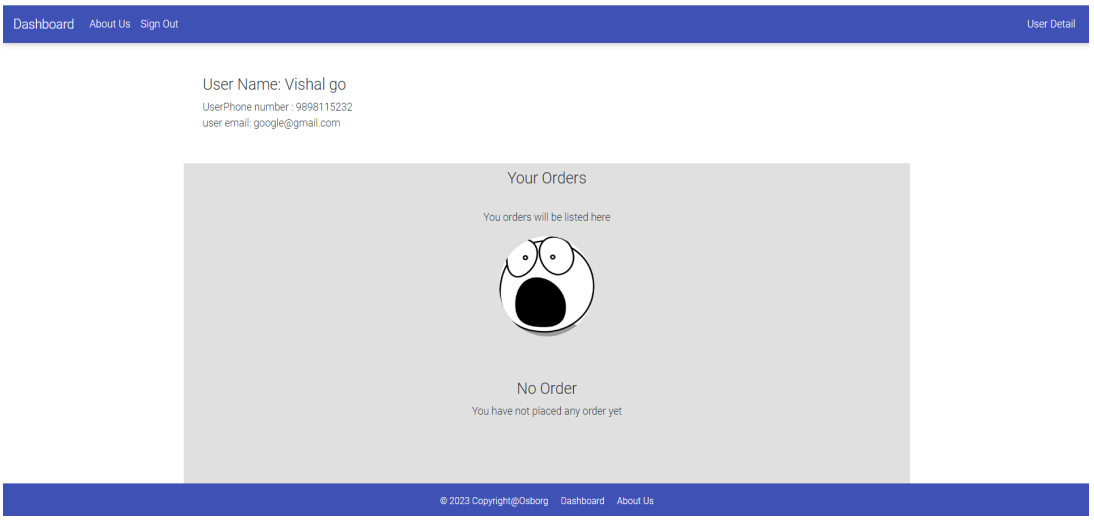
The above page shows up when placing orders using different user authority.
You can see the user's details by clicking on the "User Details" tab on the top-right corner of the screen. It will be shown in the image below. It shows "Username", "Phone Number", "Email", and all the "Orders" made by the user in a descending order of order_placing_time.



On this page, Delete option is provided only if order_status is "delivered". If you click on "Delete" for any such order, the corresponding record will be immediately removed from the database and the page will be reloaded. Besides, the orders that are place but not yet delivered will show "In progress", as shown in the following snapshot. (DELETE)

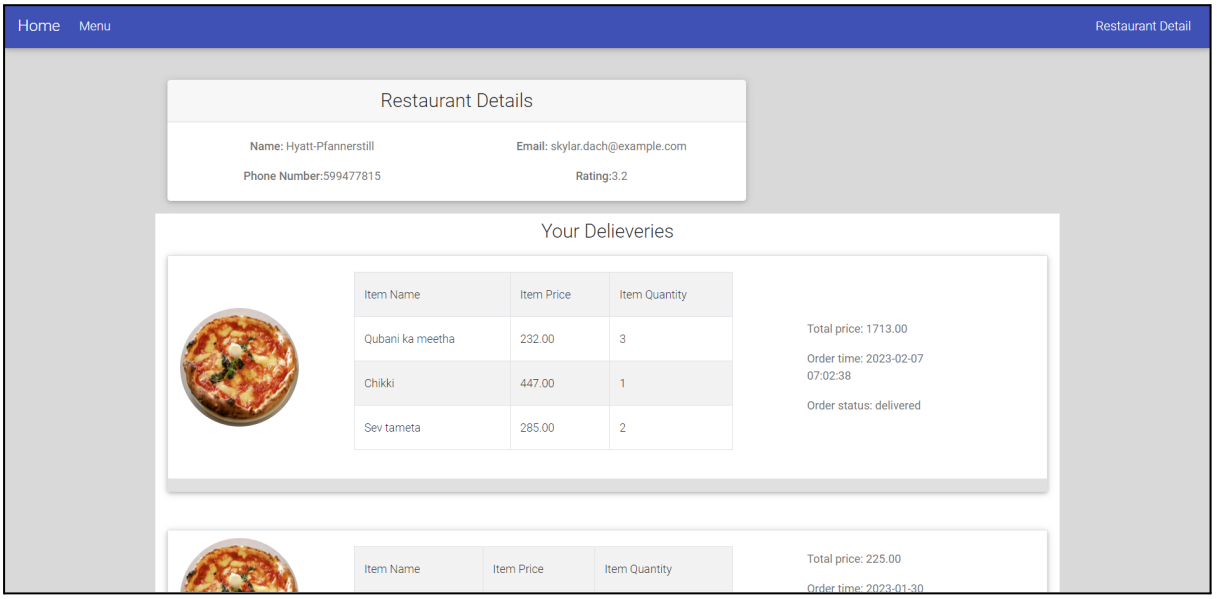


When users have no orders placed the following page will be displayed.



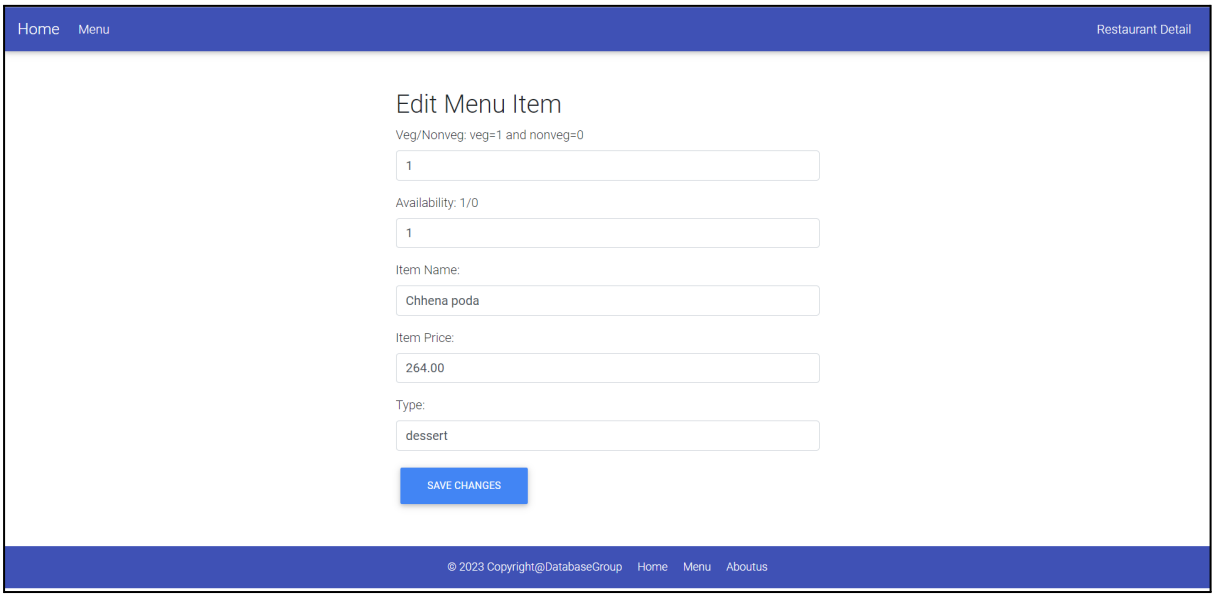
3 : Restaurant Pages

Restaurant owners and related persons with their credentials can open these pages. They will be able to see “Name”, “Address”, “Email”, and “Rating” of the restaurant. They can view the latest 10 orders given to this restaurant.



When they click on the “Menu” tab in the top-left corner of the screen, they can see all of the menu-items that their restaurant serves. They are able to edit the item details and also are able to add and remove the items, by clicking on “Edit” button. They can also add new menu_item for the logged in restaurant.

When you click on the edit button, you will see the following interface. There, you can edit the mentioned details about the items. To save (propagate) your changes to the database, click on the “Save Changes” button. **(UPDATE)**



When you click on it, it will show a confirmation message on top of the screen, in the green highlighted background - “Edited menu successfully!!”.

4 : Delivery-Agent Pages

When the delivery agent logs in he/she will be directed to this page. They will be able to see “Name”, “Email”, and “Phone Number” of themselves. They can view the orders’ delivery details and also their current status.
Delivery agents could watch out for the “AboutUs” page or go backwards to the homepage.
New orders placed will be visible to all delivery agents because it hasn’t been assigned to any particular agent till now. In the below image record with order_ID “1501” is a new order.

Logged in successfully !

SIGN OUT

Delivery Agent Page

Agent Details

Name: Kathryn Little

Email: roselyn.durgan@example.org

Phone: 497110173

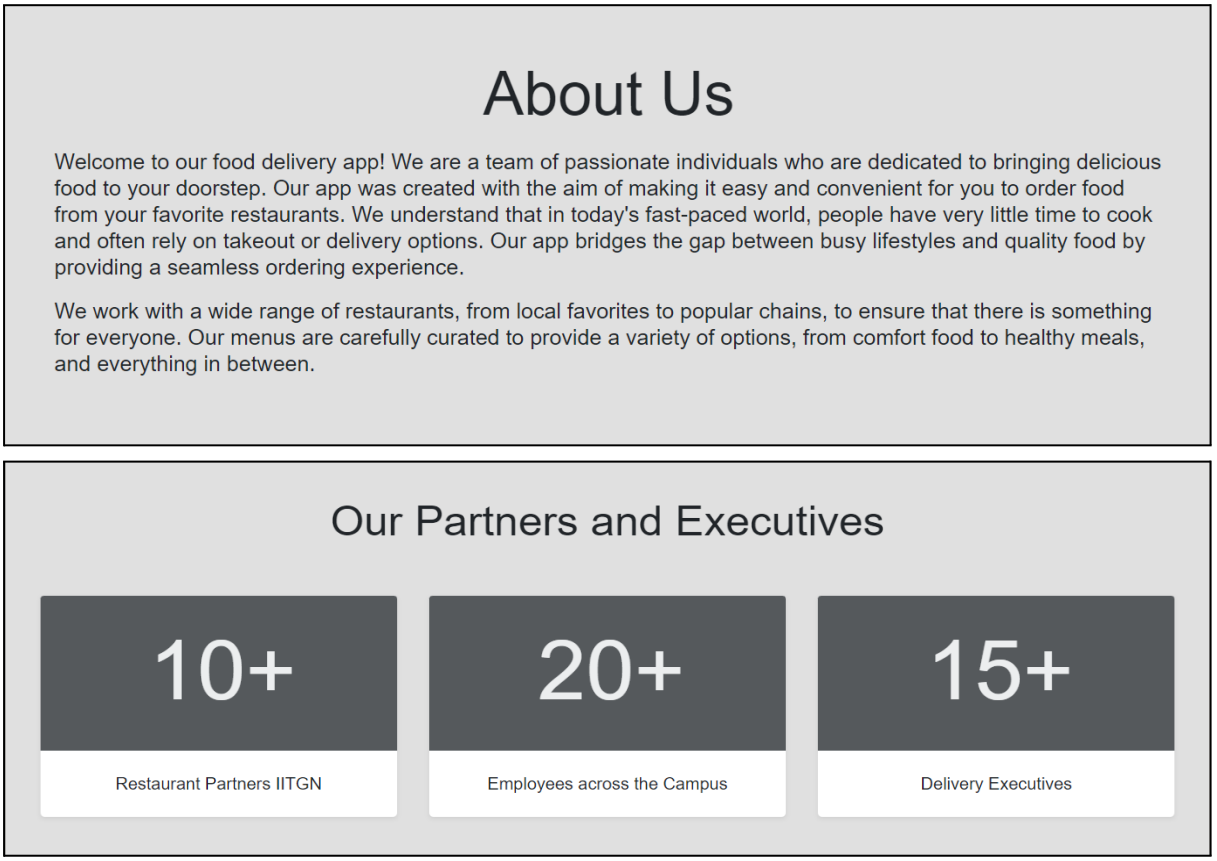
Orders

Order ID	Customer Phone number	Restaurant Adress	Delivery Address	Delivery Charge	Pickup Time	Delivery Time	Current Order Status
1501	535960267	y785 Nata Ernestoborough NewYork 687998	Missing for now	None	None	None	placed
244	932962480	z774 Doug Fridabury Michigan 161719	s775 Manu Kendallton Pennsylvania 967481	289	2023-02-10 11:14:09	2023-02-10 13:08:36	delivered
1010	665548462	o927 Hami Savanahview NorthCarolina 680341	p160 Donn Waelchiberg Ohio 488578	385	2023-02-10 09:05:25	2023-02-10 11:17:29	delivered
800	1317250753	g683 Mitc Myrtieburgh Wyoming 750649	w534 Blan Gloverville NorthDakota 143913	894	2023-02-10 09:06:23	2023-02-10 10:26:50	delivered
3	23056547	x809 Creo Lake Friedrich NorthCarolina 186118	y952 Okun Denesikmouth Idaho 159482	407	2023-02-10 08:51:18	2023-02-10 09:28:03	delivered
872	368638774	c654 Boga East Maymie SouthCarolina 164434	e606 Summ Palmaview Arkansas 216564	205	2023-02-10 06:57:07	2023-02-10 09:01:28	delivered
848	1445065110	z259 Shay Hickleborough Arizona 110576	x595 Cleo East Pamela Idaho 157300	410	2023-02-10 06:59:32	2023-02-10 09:57:53	delivered
809	1400295995	o927 Hami Savanahview NorthCarolina 680341	z311 Geov Lake Piper District of Columbia 101829	323	2023-02-10 04:46:36	2023-02-10 05:57:15	delivered
210	1082541047	e787 Pasc Lake Abdielville WestVirginia 639057	s436 Soph Wolfstad Vermont 786299	772	2023-02-10 04:32:16	2023-02-10 07:15:46	delivered
1118	164318055	j739 Alys East Camille Nevada 116925	d575 Alba West Hymanchester Connecticut 107834	313	2023-02-10 01:24:49	2023-02-10 04:18:26	delivered

© 2023 Copyright@Osborg Home About Us

5 : About-Us Page:

The “**RENAME**” command execution is handled on this page. It displays a team details table containing the details of the group members at the bottom.
This page gives information "aboutUs”.



“Roll_number” is the primary key of the relation. It is newly added in our database to showcase our team members' details. Below the table there are options to change the name of a particular column of this database.

roll_number	first_name	last_name	email_id
20110030	Juhil	Aslaliya	aslaliya.juhil@iitgn.ac.in
20110049	Dablu	Kumar	dablu.kumar@iitgn.ac.in
20110057	Dhiren	Aggarwal	dhiren.aggarwal@iitgn.ac.in
20110112	Meet	Vankar	vankarmeet@iitgn.ac.in
20110125	Tejas	Parmar	tejas.parmar@iitgn.ac.in
20110129	Dhruv	Patel	patel.dhruv@iitgn.ac.in
20110132	Prey	Patel	patel.prey@iitgn.ac.in
20110141	Prakram	Rathore	prakram@iitgn.ac
20110144	Pratik	Raj	pratik.raj@iitgn.ac.in
20110153	Rahul	Kumar	kumar.rahul@iitgn.ac.in
20110229	Vishal	Ghoniya	vishal.ghoniya@iitgn.ac.in

Column to rename roll_number Column Name RENAME COLUMN NAME

Here, we have renamed the “last_name” column to “surname”.

roll_number	first_name	surname	email_id
20110030	Juhil	Aslaliya	aslaliya.juhil@iitgn.ac.in
20110049	Dablu	Kumar	dablu.kumar@iitgn.ac.in
20110057	Dhiren	Aggarwal	dhiren.aggarwal@iitgn.ac.in
20110112	Meet	Vankar	vankarmeet@iitgn.ac.in
20110125	Tejas	Parmar	tejas.parmar@iitgn.ac.in
20110129	Dhruv	Patel	patel.dhruv@iitgn.ac.in
20110132	Prey	Patel	patel.prey@iitgn.ac.in
20110141	Prakram	Rathore	prakram@iitgn.ac
20110144	Pratik	Raj	pratik.raj@iitgn.ac.in
20110153	Rahul	Kumar	kumar.rahul@iitgn.ac.in
20110229	Vishal	Ghoniya	vishal.ghoniya@iitgn.ac.in

Column to rename roll_number Column Name RENAME COLUMN NAME

Responsibilities and Contributions:

Design of application done in a meeting.

Dablu, Rahul, Tejas	Design and making up of front end
Pratik, Prakram and Juhil	Coming up with appropriate SQL queries
Dhruv and Dhiren	Check the web app functionalities and make readme file accordingly
Vishal	Backend integration of database and frontend
Rahul	put up all files of frontend together including bugs removal and made it compatible
Tejas, Prey, Meet, Dhiren	Making up the final report, pointing bugs and record demo