A PROJECT REPORT ON

# STUDENT WELFARE SYSTEM

SUBMITTED IN PARTIAL

FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

UNDER THE GUIDANCE OF

Mr. Pratik Dhole

PRESENTED BY


| | |
|---|---|
| 220940120088 | Ishita Rajora |
| 220940120122 | Muskan Singh |
| 220940120175 | Sapkale Nilesh Sanjay |
| 220940120200 | Shubham Goswami |
| 220940120203 | Shubham Siddharam Vibhute |
| 220940120209 | Suryakant Shankar Shintre |


AT

CENTER FOR DEVELOPMENT OF ADVANCED

COMPUTINGC-DAC, PUNE

# ACKNOWLEDGEMENT

The project **Student Welfare System** was a great learning experience for us and we are submitting this work to the Advanced Computing Training School (C-DAC ACTS, Pune). We are very glad to mention the name of Mr. Pratik Dhole for his valuable guidance to work on this project. Our heartfelt thanks go to Ms. Namrata Mam (Course Coordinator, PG-DAC) who gave us all the required support and kind coordination to provide all the necessities to complete the project and throughout the course up to the last day of the course.

We would like to express our sincere gratitude towards Mrs. Madhura Anturkar, our faculty for Advanced Java, who was always there for us. Her guidance and Student Welfare System us overcome various obstacles and intricacies during the course of our project work. Without her tremendous support, guidance, and efforts, this project would not have been possible.

From,

220940120088 Ishita Rajora
220940120122 Muskan Singh
220940120175 Sapkale Nilesh Sanjay
220940120200 Shubham Goswami
220940120203 Shubham Siddhram Vibhute
220940120209 Suryakant Shankar Shintre

# ABSTRACT

In the present situation, the web applications which are used by the user for food services and other day-to-day activities are present in different applications with their respective modules. These modules are not only complicated but also discreet which make the whole task time-consuming and lethargic. In order to eradicate these problems, we need a solution such that the student is able to complete as well as modify his/her rental accommodations.

**Student Welfare System** is a web application specially developed for students to facilitate easy search and ordering of food from local kitchens and hotels along with other related services. This application will help students to save time and money by eliminating the dependence on the offline facilities. The user interface (UI) of the application is developed using the React.js library and the server-side programming is done in the Spring Boot framework of Java Enterprise Edition (J2EE) along with MySQL as the database technology.

# INDEX

# LIST OF TABLES



# LIST OF FIGURES

## *1.* INTRODUCTION TO PROJECT

Some of the applications in this field are ZOMATO, FOODPANDA etc. These applications are perfectly fine in their respective fields but they are discrete. Due to this nature of isolation, a particular user when trying to access both these features in a single platform finds him in a spot of bother. In order to remove this issue of redundancy, we are generating an application which will merge all the features into one single application. This integrated platform will help the student to save data, time and money. Let us consider an example where a CDAC student belonging to a different state or locality comes to a particular place. Certainly, he doesn't have any knowledge about the place. He needs to roam around for hours in order to search a place for shelter, a place where edible food is available. During the time of examination or assignments he /she have to look out for various stationary shops and other day-to-day activities. If all these problems are solved by browsing a particular application, the world of the students will turn out to be very easy and accessible.

Our application will provide food delivery as the first service to students along with other services such as paying guest rooms, and stationary shops in the proximity of the particular student (which are considered for future scope and development). The idea is very simple but will turn out to be very helpful and time saver for a particular user because it is completely based on real time issues which a common man faces as an immigrant in a new locality. The admin updates the availability of local kitchens and hotels in different areas and students check the availability of food service in a specified location. They should be able to order the food to their needs in advance to make their stay comfortable. Also, we are keeping the future scope for searching the other services as well.

In the present situation, the web applications which are used by the user for food services and other day-to-day activities are present in different applications with their respective modules. These modules are not only complicated but also discreet which make the whole task time-consuming and lethargic. In order to eradicate these problems, we need a solution such that the student is able to complete as well as modify his/her rental accommodations.

Student Welfare System is a web application specially developed for students to facilitate easy search and ordering of food from local kitchens and hotels along with other related services. This application will help students to save time and money by eliminating the dependence on the offline facilities. The user interface (UI) of the application is developed using the React.js library and the server-side programming is done in the Spring Boot framework of Java Enterprise Edition (J2EE) along with MySQL as the database technology.

The web-based Student Welfare System project is an attempt to stimulate the basic concepts of food shopping. The system enables the customer to do the things such as search for menu items category wise, choose menu items based on description and add that item into cart. The system provides you details about food items. If user want to buy food items, he must have registered account. The system shows the food items that are available. The system displays price, image and quantity of food items to user. Here we provided menu items by category wise that allows customer to choose a particular item easily. If the menu items are available then the system allows the user to add food items into cart.

To place order system ask user to select the address and payment mode. Single customer can save multiple addresses for his account but while placing order he can select only one address. If address is not provided the user can't place order, Customer have to specify the address before placing order. After selecting address and payment mode customer can place order and the same updates will be done in database. The System have admin who can add new menu types and menu items or can remove menu types and menu items and he also can see the availability of menu items.

# *2.* PROJECT OVERVIEW AND SUMMARY

## PURPOSE

To develop a web application convenient for newly admitted students in CDAC centers to minimize efforts and save time in acquiring needful services. These services include food, paying guest, etc. In particular, food delivery service is taken into consideration as the initial service and described henceforth as a reference. Our system is flexible, so that it can be used for different rooms and services.

## SCOPE

- Currently Purchasing food items has become a tedious job near to CDAC center.
- Small and medium scale restaurants, have to manage data about customers, services offered to them.
- It is difficult for small scale businesses to maintain data for longer time as they are using paper-based system.
- Students also need to find nearest restaurants which provide authentic service.
- Using this system, they will be able to maintain customer and services data.
- We are also solving the problem from customer's end by making ease of choice. They can choose the products from different category and from different Restaurants.

# USER CLASSES AND CHARACTERISTICS

In this software, there is an Admin, Admin can add new category of menu type. Customer can use the software for registering to the system. Customer can purchase different food items and can place order. Restaurants Can Register and list there products. delivery person can see order list and order status.

# TECHNOLOGIES USED

- MySQL
- React-JS
- Spring Boot

# REQUIREMENTS

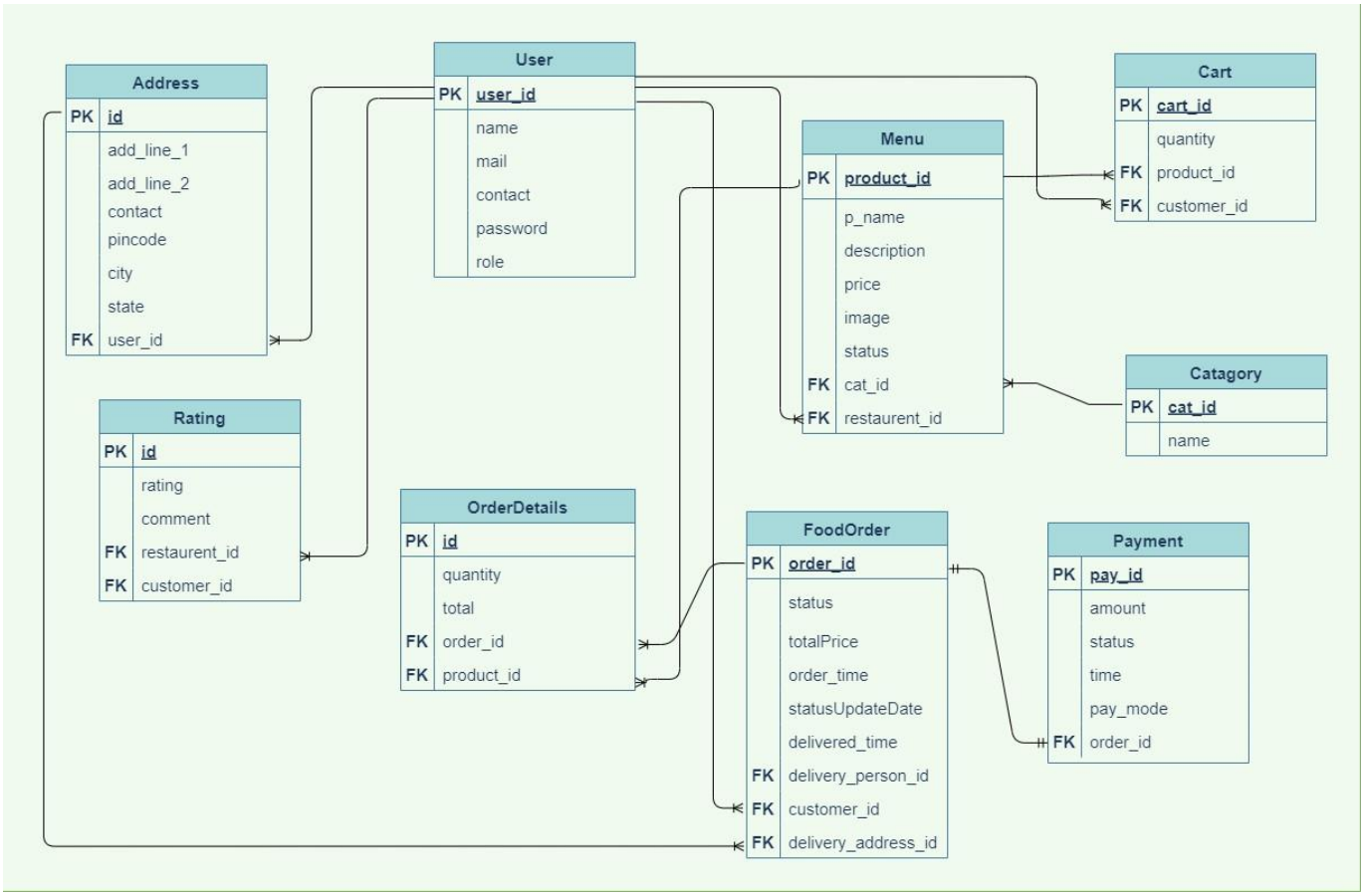
## FUNCTIONAL REQUIREMENTS

The major functionality of this project is divided into four categories.

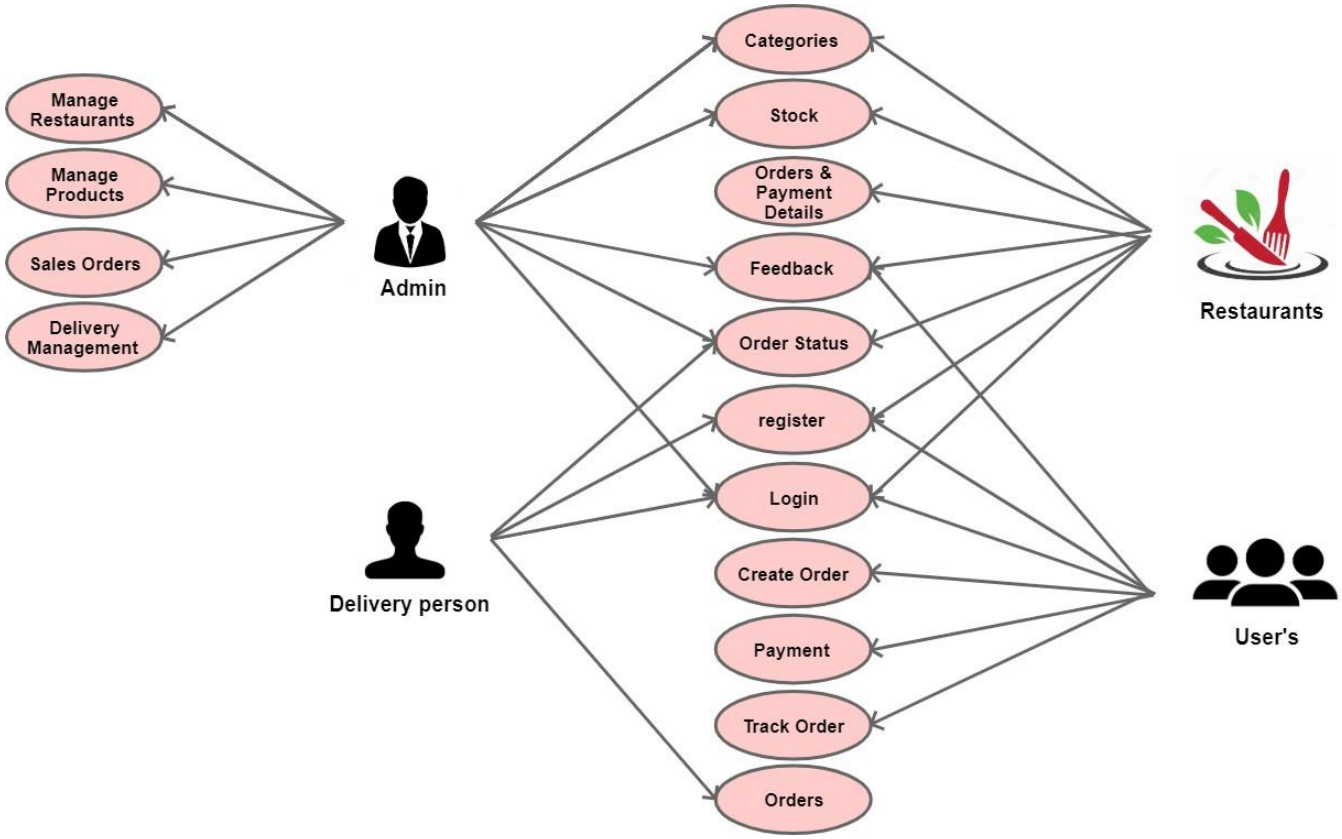
- Administrative Functions.
- Student Functions.
- Restaurant Functions.
- Delivery Boy Functions.

In this application each and every user must have their own Email ID and Password, using these Email ID and Password only they can directly enter into their corresponding Login forms. System analysis will be performed to determine if it is feasible to design information based on policies and plans of the organization and on user requirements and to eliminate the weaknesses of the present system.

# ER-DIAGRAM


Address
PK id
add_line_1
add_line_2
contact
pincode
city
state
FK user_id
User
PK user_id
name
mail
contact
password
role
Cart
PK cart_id
quantity
FK product_id
FK customer_id
Menu
PK product_id
p_name
description
price
image
status
FK cat_id
FK restaurent_id
Catagory
PK cat_id
name
Rating
PK id
rating
comment
FK restaurent_id
FK customer_id
OrderDetails
PK id
quantity
total
FK order_id
FK product_id
FoodOrder
PK order_id
status
totalPrice
order_time
statusUpdateDate
delivered_time
FK delivery_person_id
FK customer_id
FK delivery_address_id
Payment
PK pay_id
amount
status
time
pay_mode
FK order_id

# USE-CASE


Manage Restaurants
Manage Products
Sales Orders
Delivery Management
Admin
Delivery person
Categories
Stock
Orders & Payment Details
Feedback
Order Status
register
Login
Create Order
Payment
Track Order
Orders
Restaurants
User's

# DATABASE DESIGN

### Users

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | No | PRI | NULL | auto_increment |
| email | varchar | No | UNI | NULL | |
| name | varchar | Yes | | NULL | |
| password | varchar | No | | NULL | |
| contact | varchar | Yes | | NULL | |
| role | varchar | Yes | | NULL | |

### Addresses

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | No | PRI | NULL | auto_increment |
| address_line_1 | Varchar | Yes | | NULL | |
| address_line_2 | Varchar | Yes | | NULL | |
| city | Varchar | Yes | | NULL | |
| contact | Varchar | No | | NULL | |
| pin_code | Varchar | Yes | | NULL | |
| state | varchar | Yes | | NULL | |
| user_id | int | No | MUL | NULL | |

## Cart

| Field | Type | NULL | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | NO | PRI | NULL | auto_increment |
| quantity | int | NO | | NULL | |
| customer_id | int | YES | MUL | NULL | |
| menu_id | int | YES | MUL | NULL | |

## Category

| Field | Type | NULL | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar | YES | | NULL | |

## Menu

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | No | PRI | NULL | auto_increment |
| description | varchar | Yes | UNI | NULL | |
| name | varchar | Yes | | NULL | |
| image | varchar | Tes | | NULL | |
| price | double | Yes | | NULL | |
| Status | tinyInt | yes | | NULL | |
| category_id | int | Yes | MUL | NULL | |
| rest_id | int | No | MUL | NULL | |

## Payments

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | No | PRI | NULL | auto_increment |
| amount | double | No | | NULL | |
| payment_time | datetime | Yes | | NULL | |
| status | varchar | Yes | | NULL | |
| Pay_mode | varchar | Yes | | NULL | |
| order_id | int | Yes | MUL | NULL | |

## Food_order

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | NO | PRI | NULL | auto_increment |
| order_date | Datetime | Yes | | NULL | |
| order_status | Varchar | Yes | | NULL | |
| status_update_date | datetime | Yes | | NULL | |
| total_price | double | NO | | NULL | |
| user_id | int | NO | MUL | NULL | |
| delivery_addresses_id | int | NO | MUL | NULL | |
| delevery_boy_id | int | NO | MUL | NULL | |

## Rating

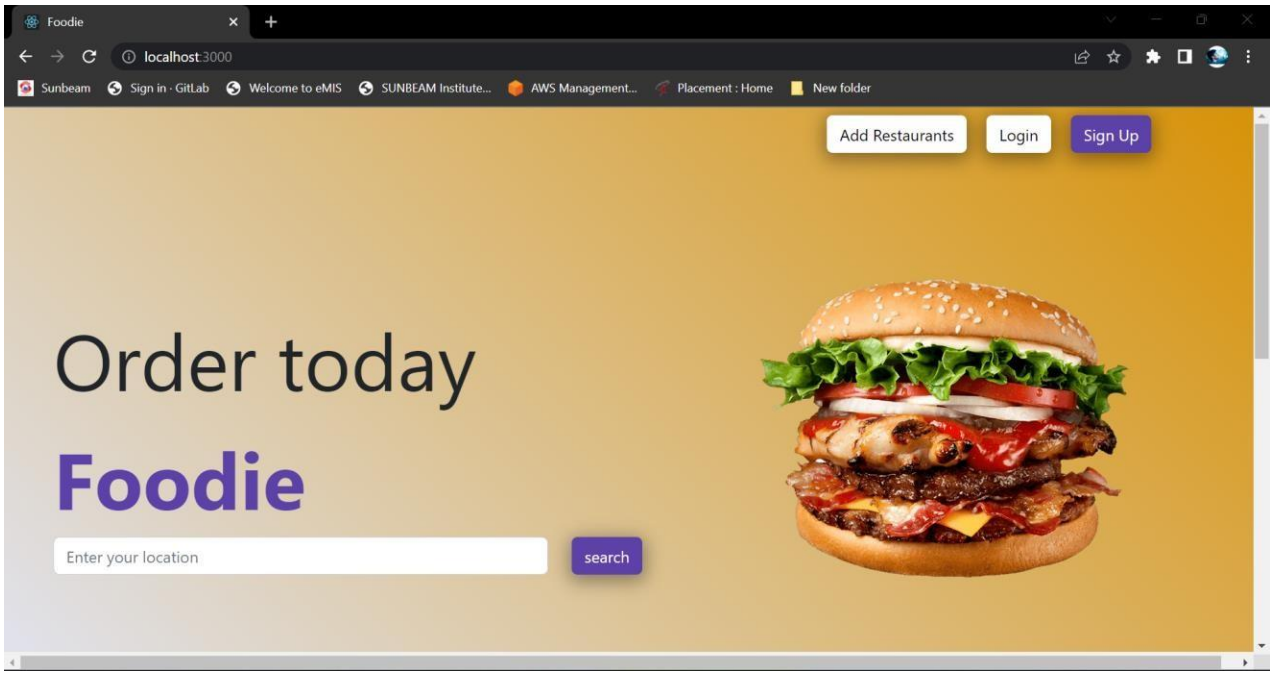
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | No | PRI | NULL | auto_increment |
| comment | Varchar | Yes | | NULL | |
| rating | int | Yes | | NULL | |
| customer_id | int | No | MUL | NULL | |
| rest_id | int | No | MUL | NULL | |

## Order_Details

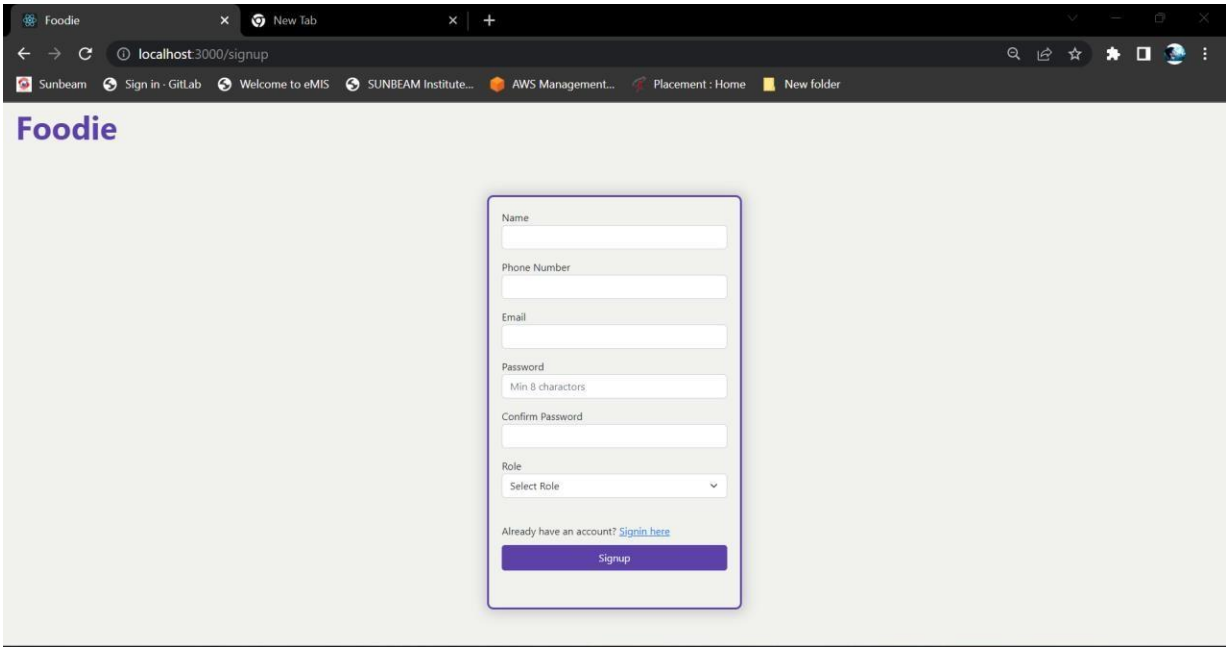
| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| id | int | No | PRI | NULL | auto_increment |
| total | double | No | | NULL | |
| quantity | int | No | | NULL | |
| order_id | int | No | MUL | NULL | |
| product_id | int | No | MUL | NULL | |

# PROJECT SCREENSHOTS

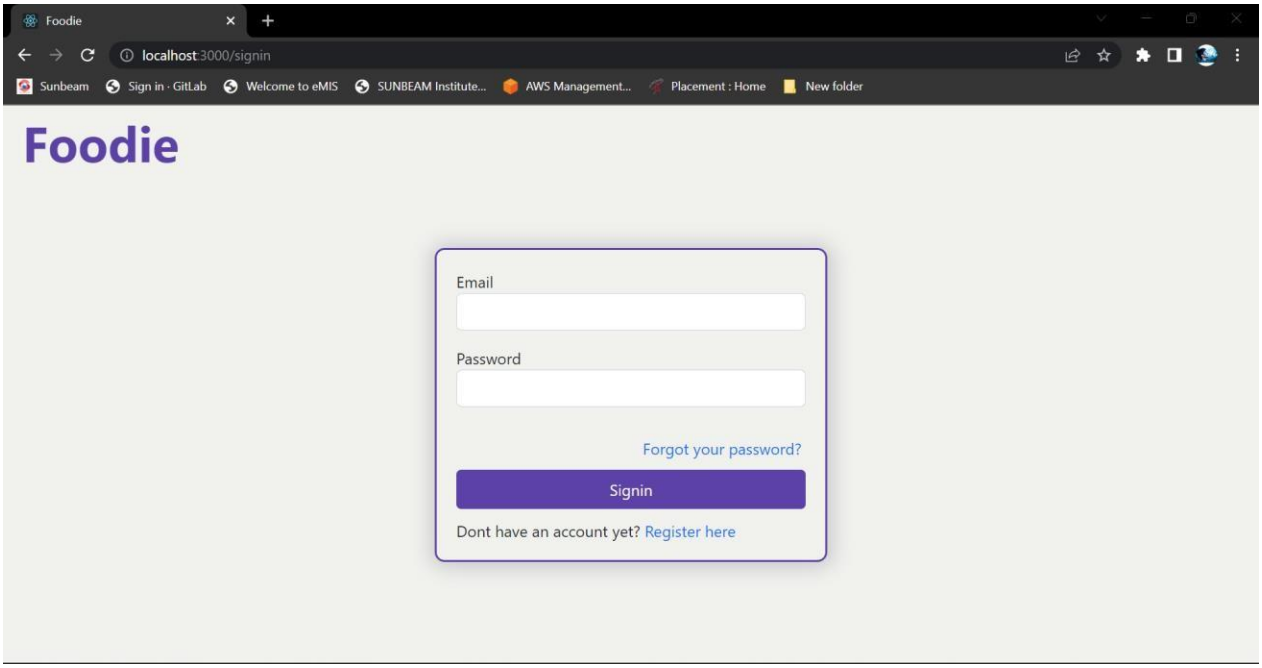
## HOME PAGE



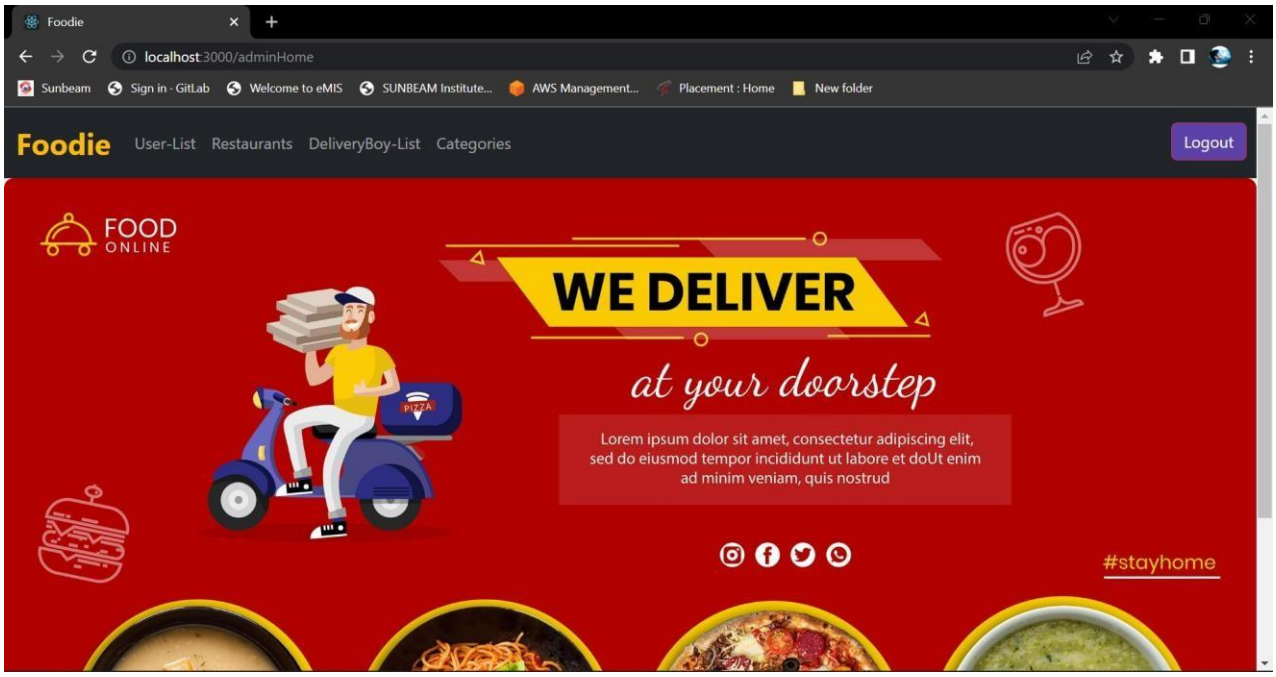

## SIGN UP PAGE



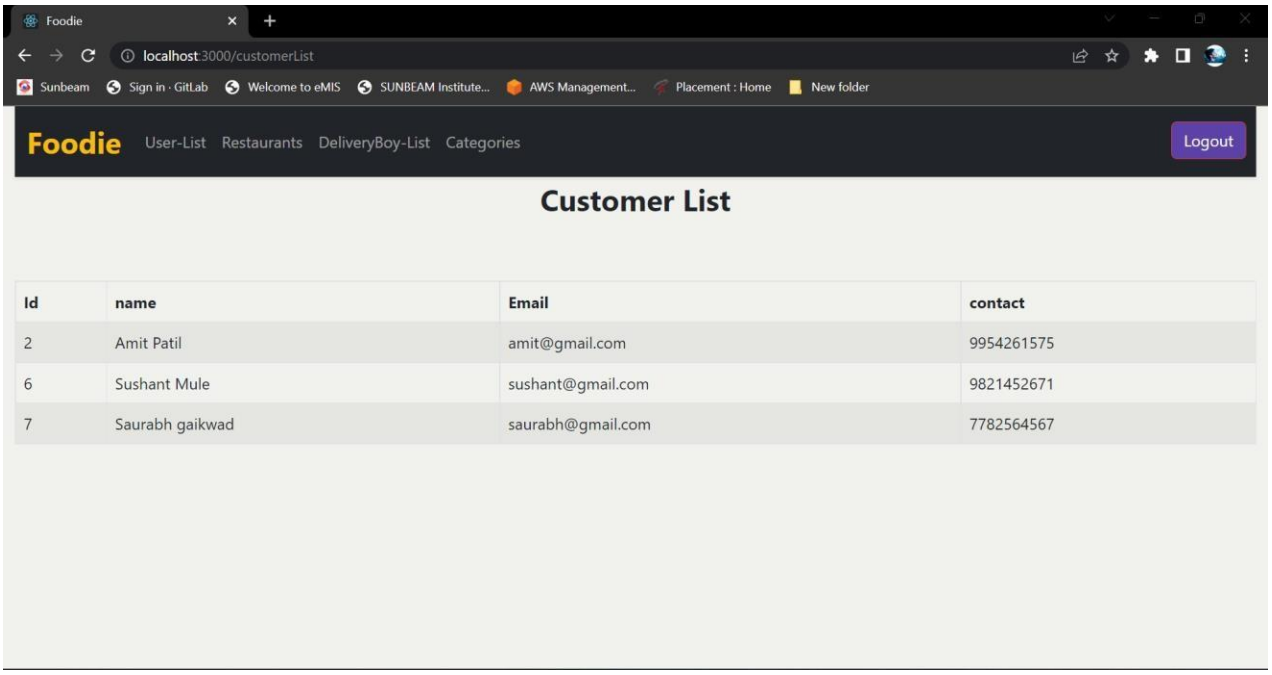
## LOGIN PAGE


Foodie
localhost:3000/signin
Sunbeam
Sign in · GitLab
Welcome to eMIS
SUNBEAM Institute...
AWS Management...
Placement : Home
New folder
Foodie
Email
Password
Forgot your password?
Signin
Dont have an account yet? Register here

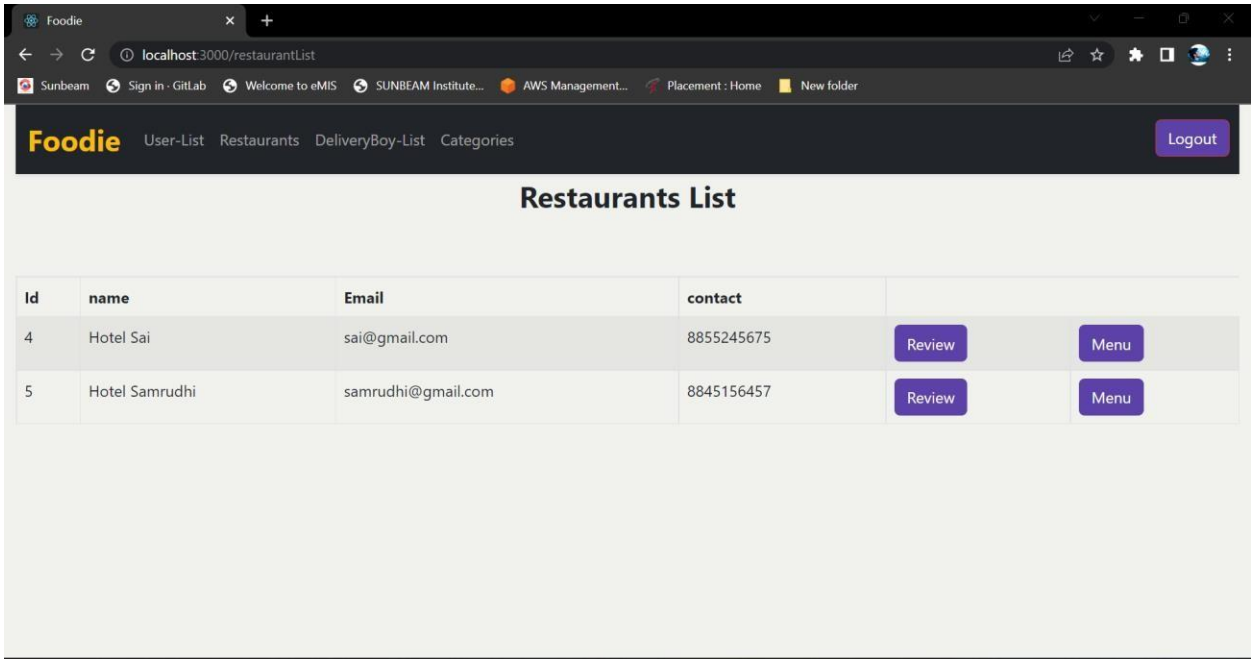
# ADMIN HOMEPAGE



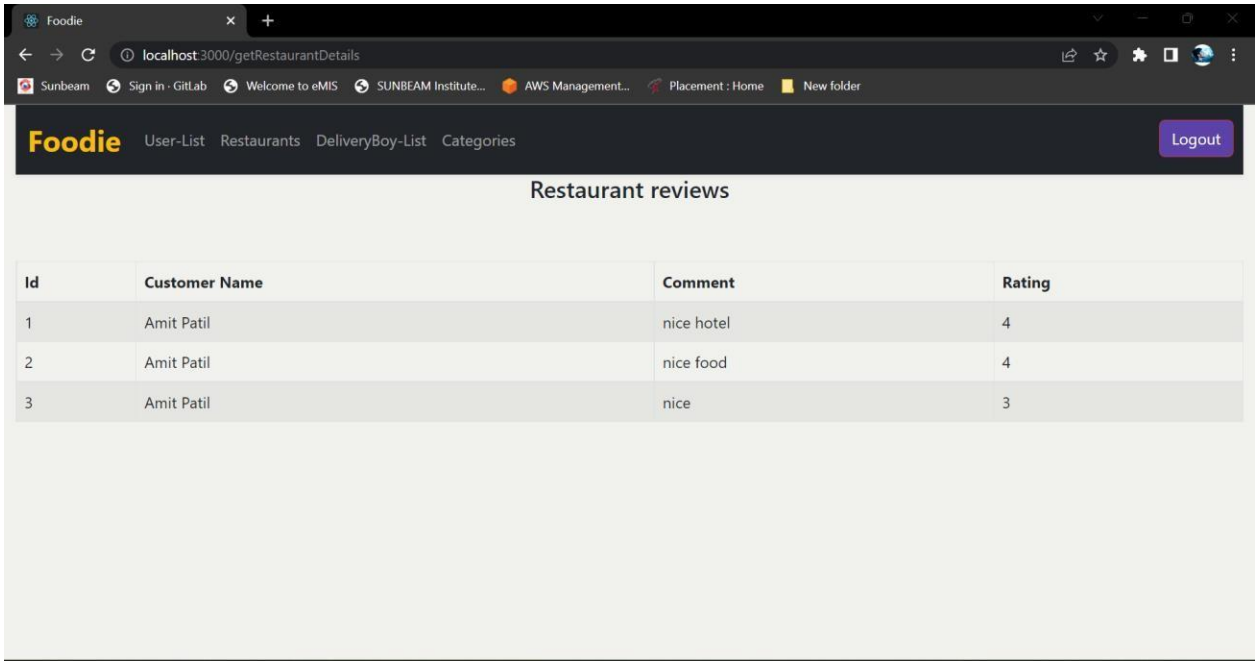

# CUSTOMERS LIST



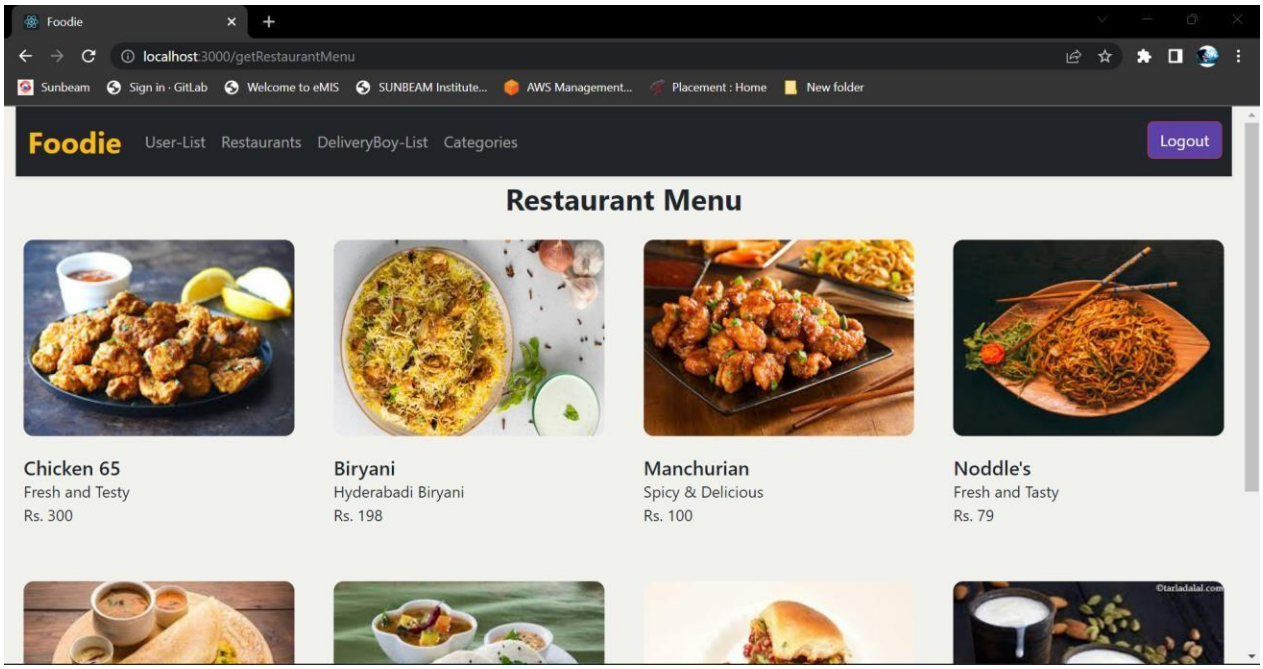
# RESTAURANTS LIST



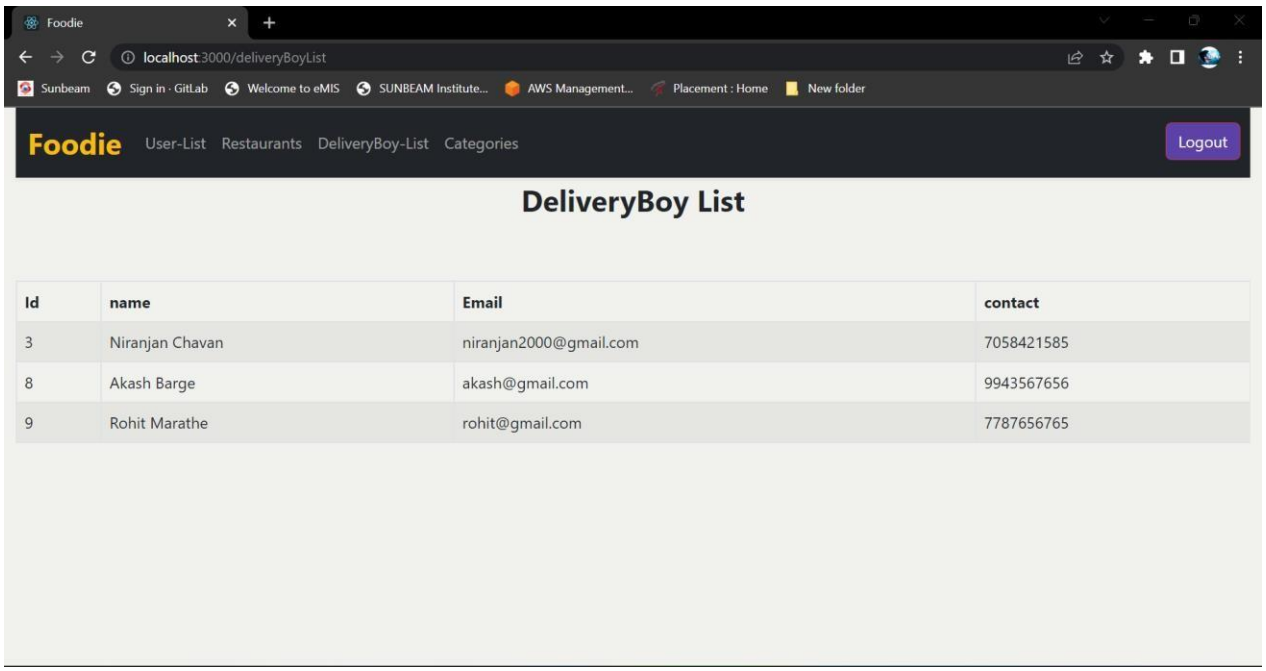

# RESTAURANT REVIEWS



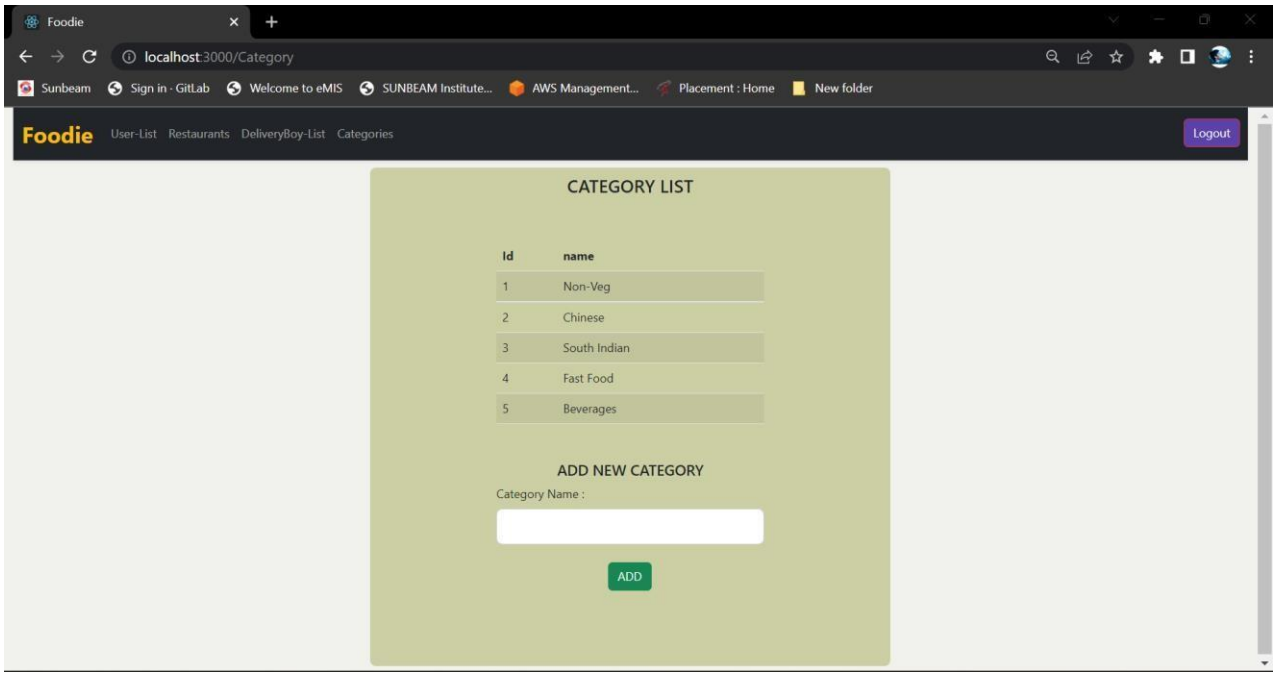
# RESTAURANT MENU



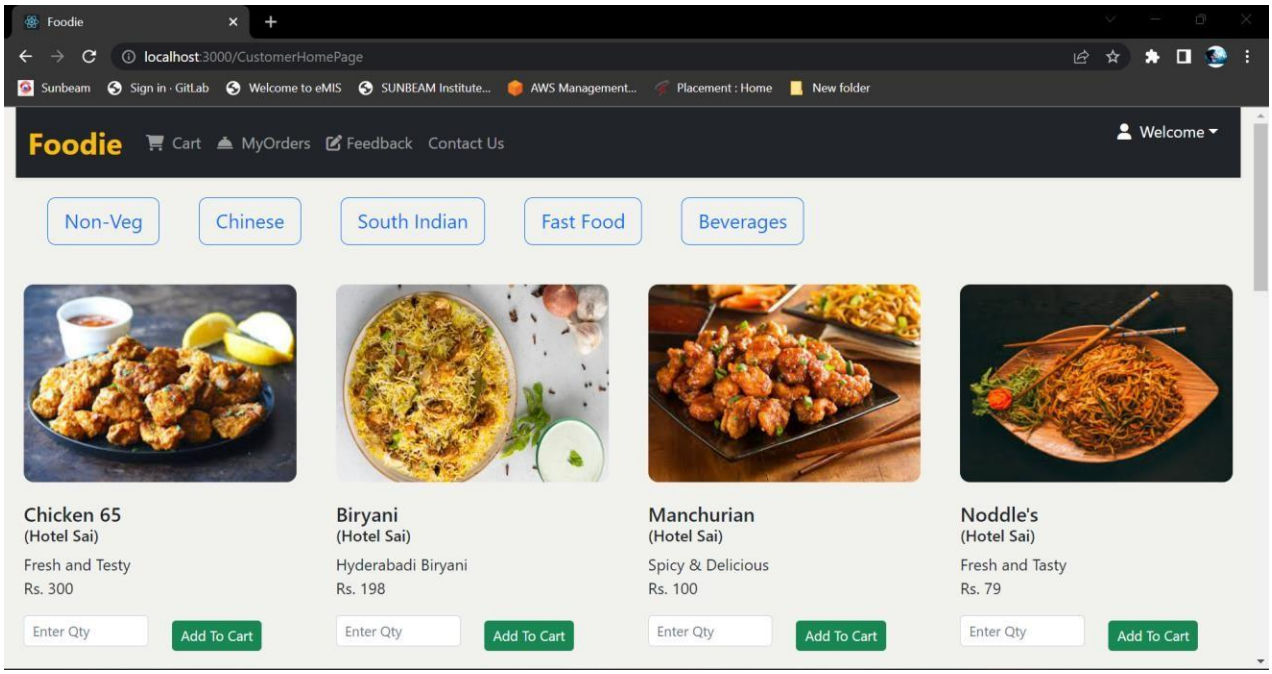

# DELIVERYBOY LIST



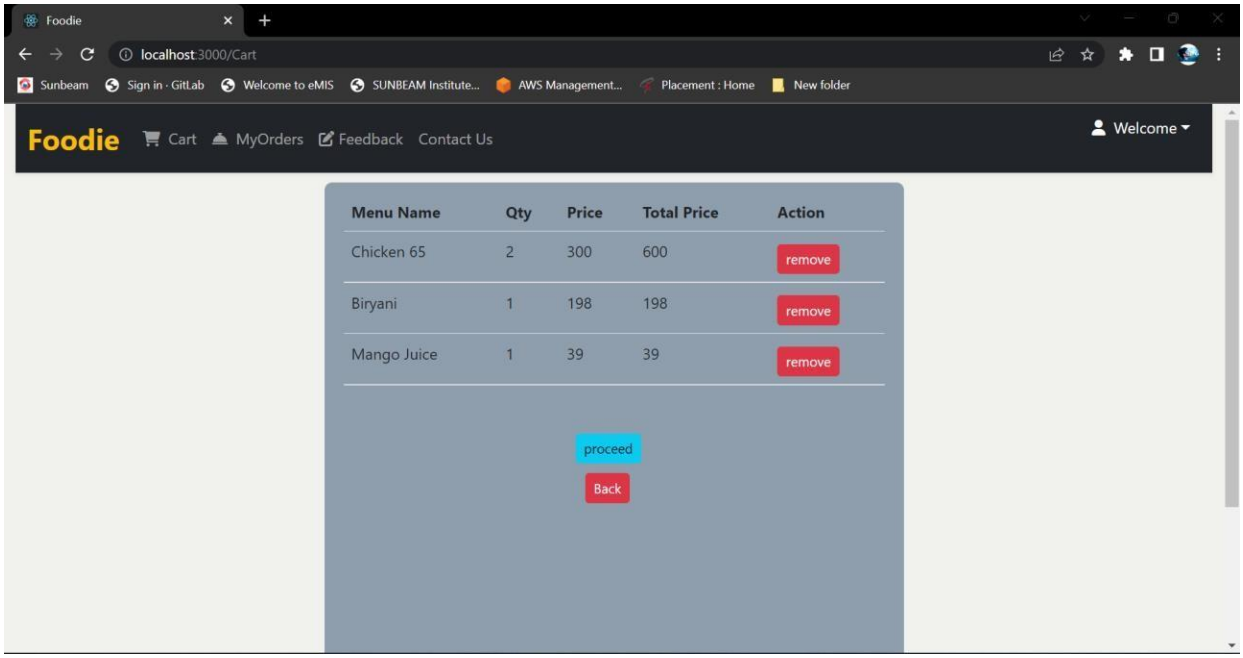
# CATEGORY



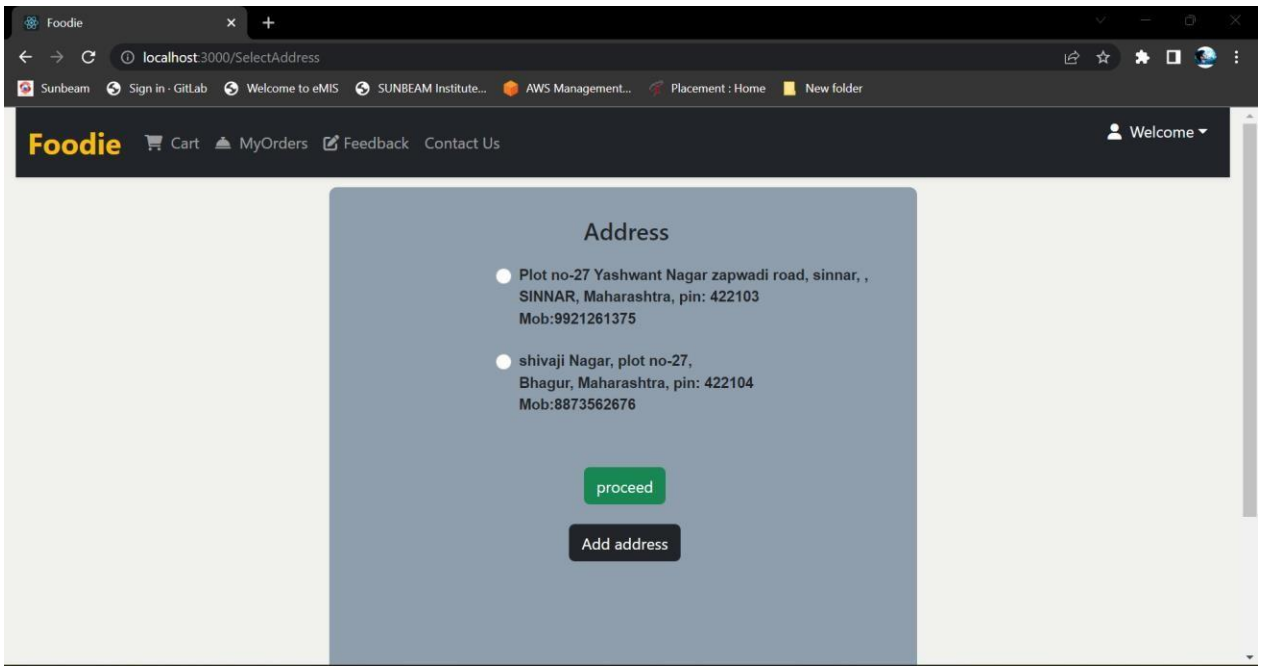

# CUSTOMER HOMEPAGE



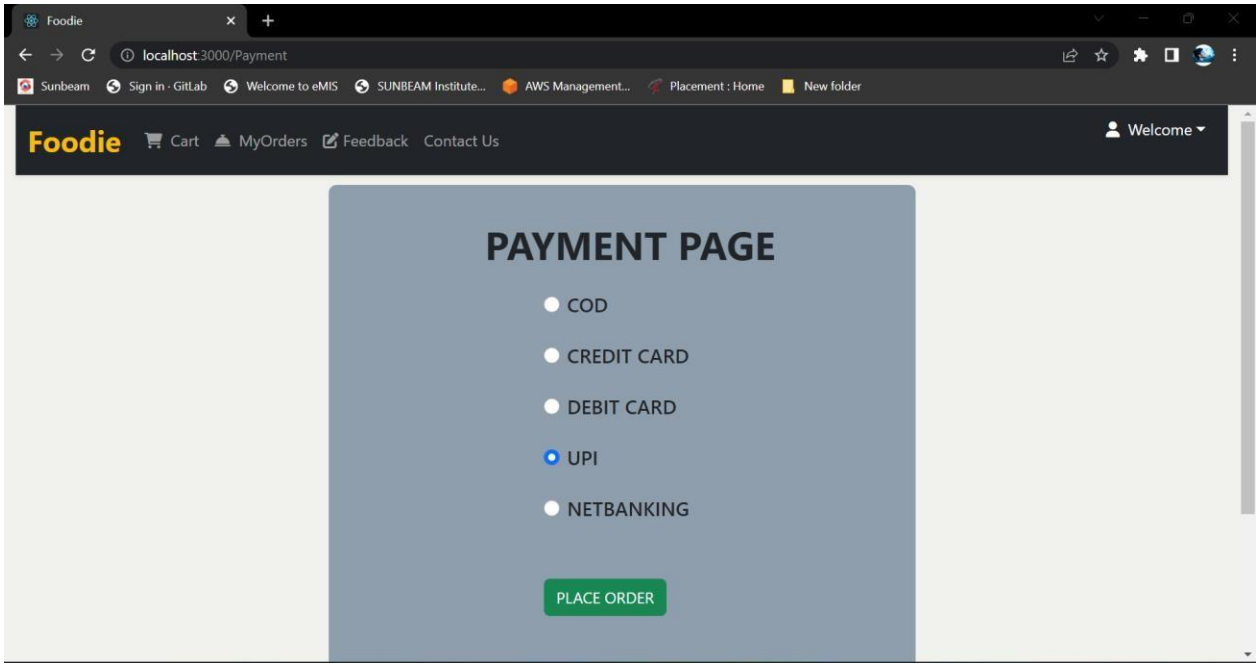
# ADD TO CART



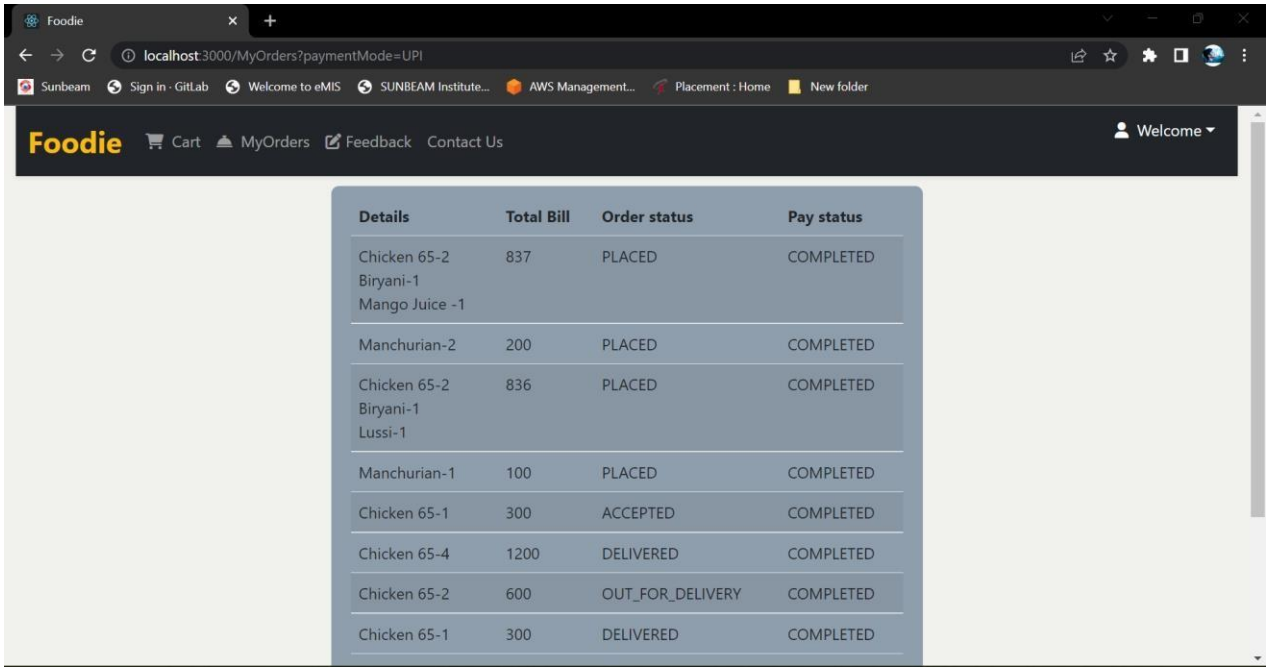

# ADD ADDRESS



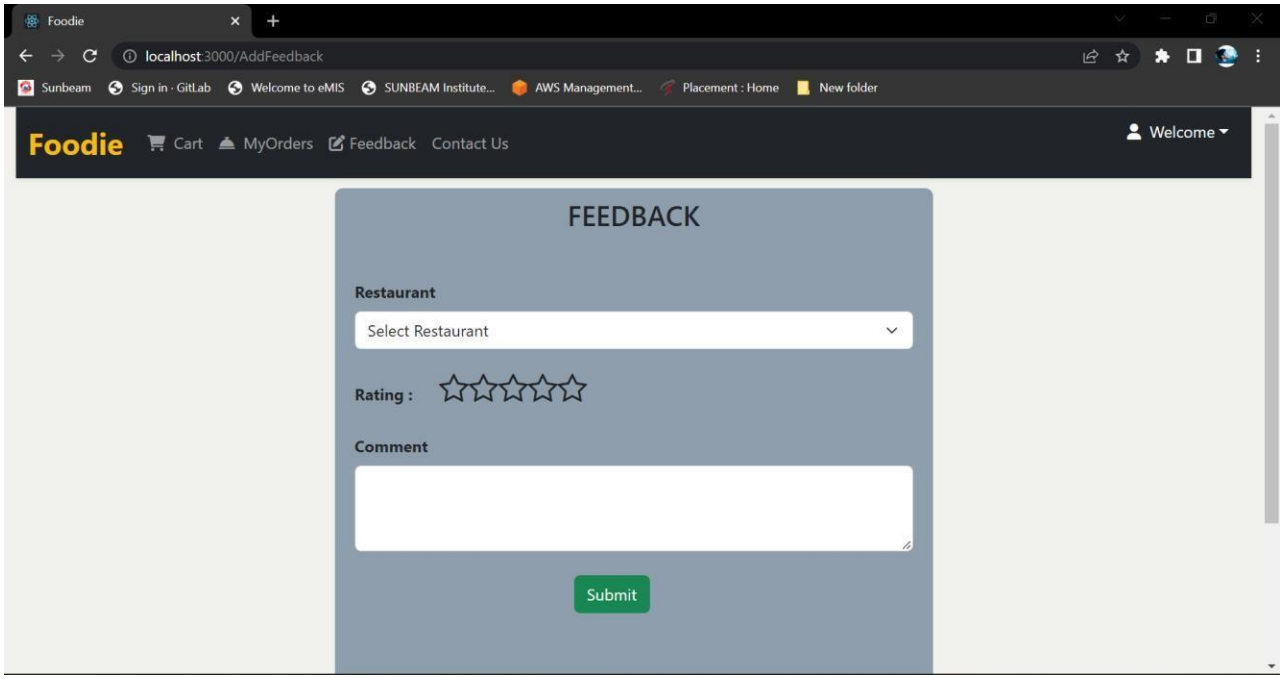
# PAYMENT METHODS



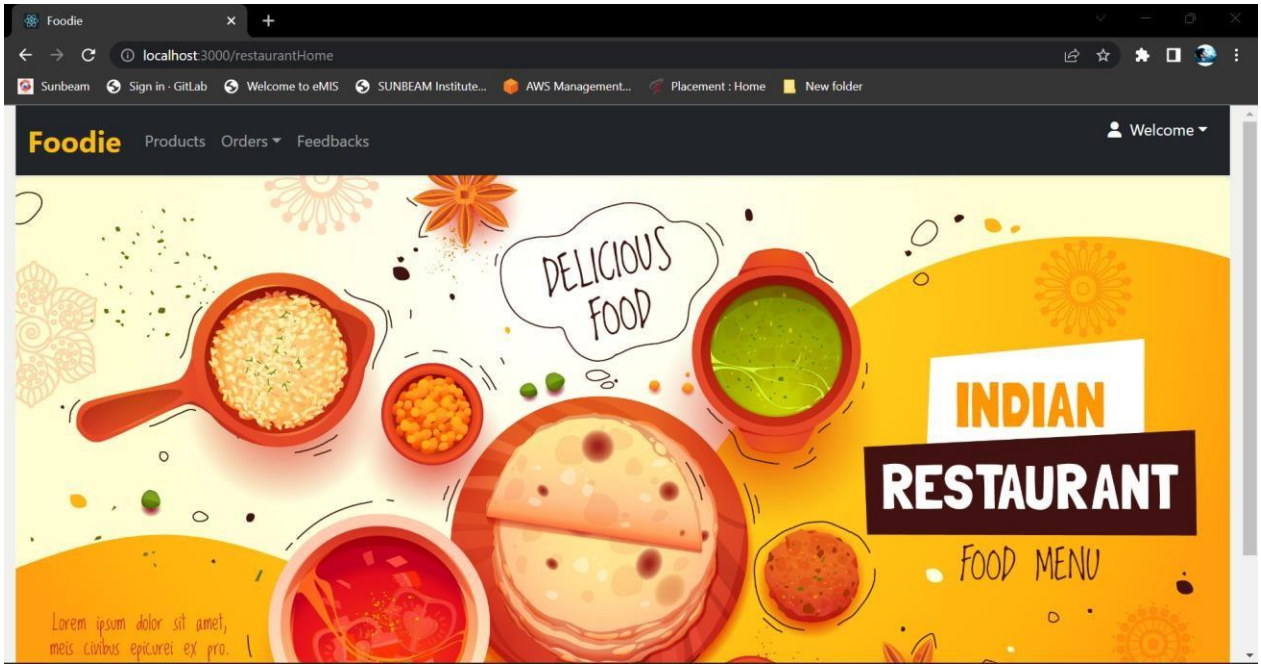

# MY ORDER DETAILS



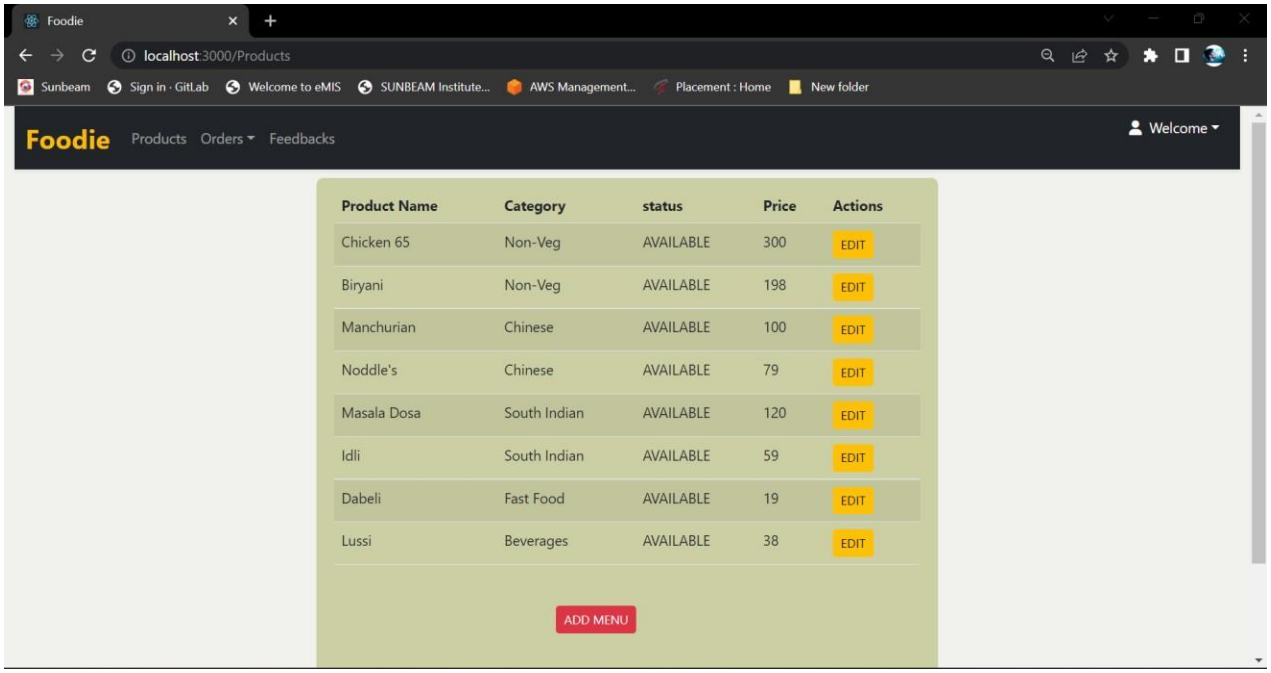
# FEEDBACK PAGE



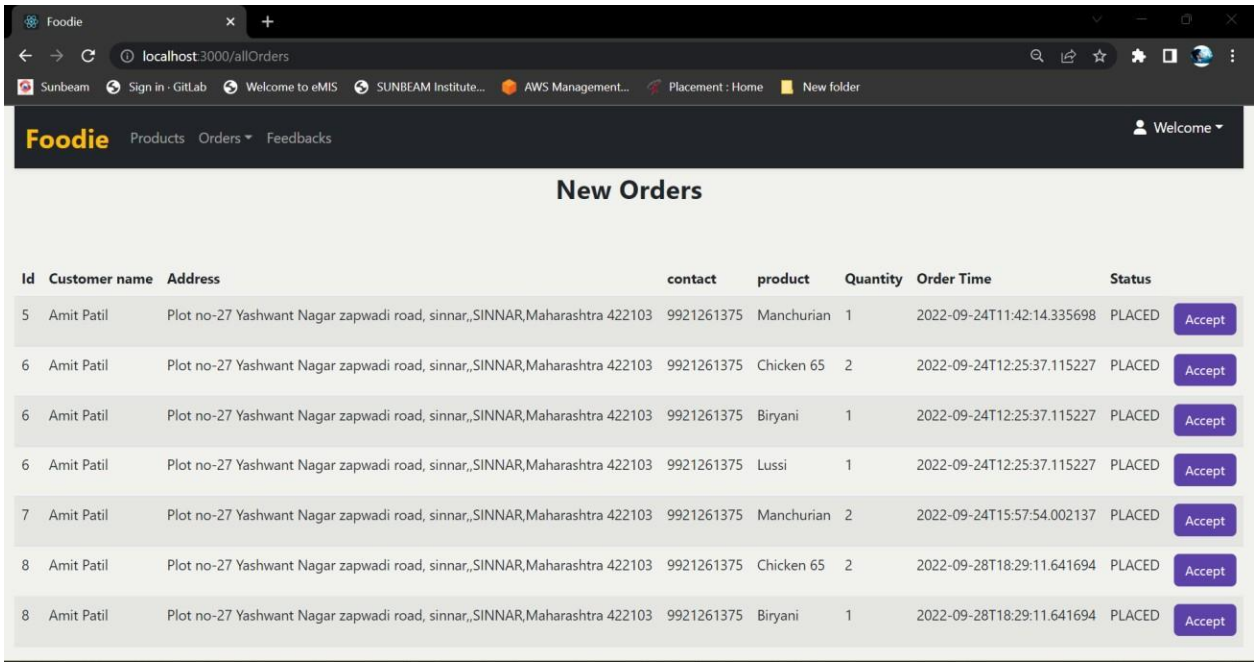
# RESTAURANT HOMEPAGE



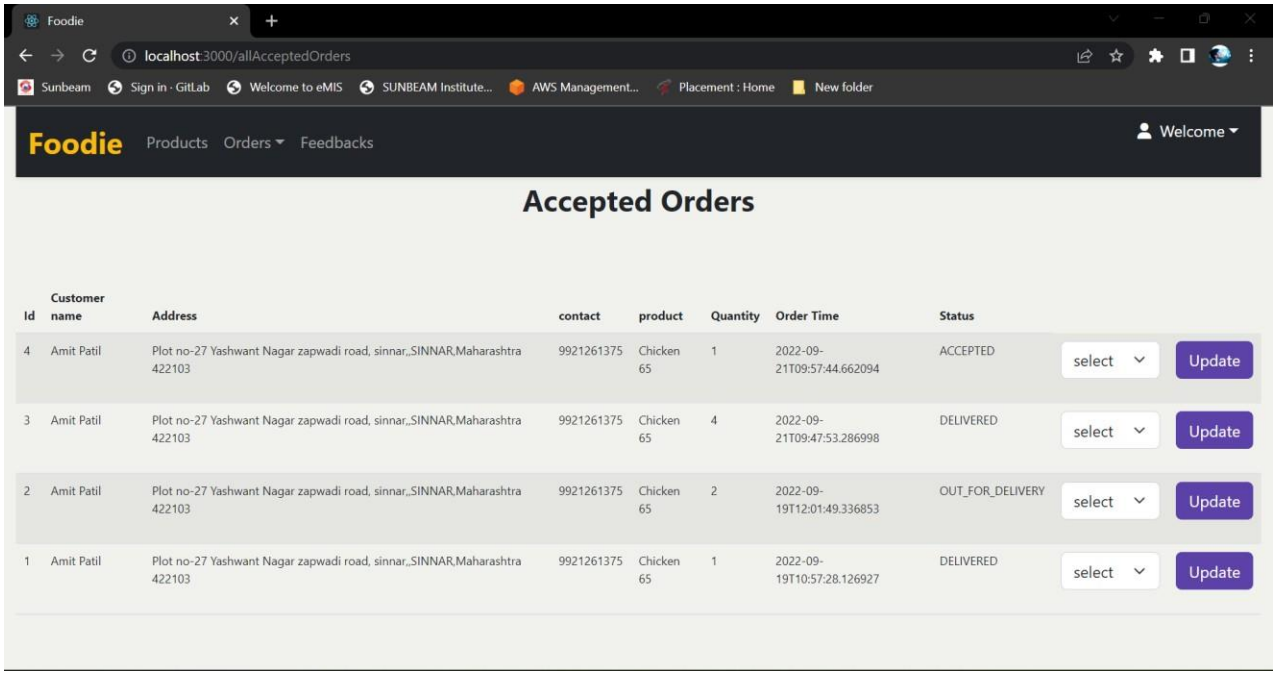

# ADD/EDIT MENU



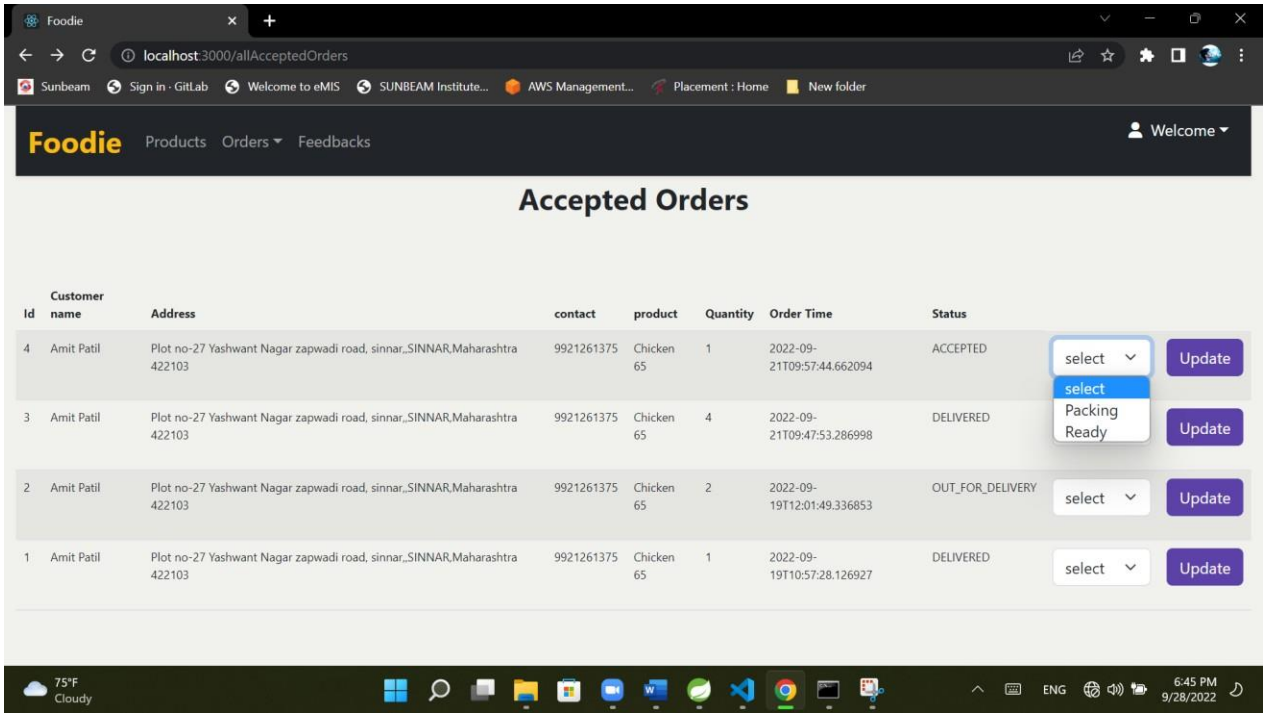
# ARRIVED ORDERS



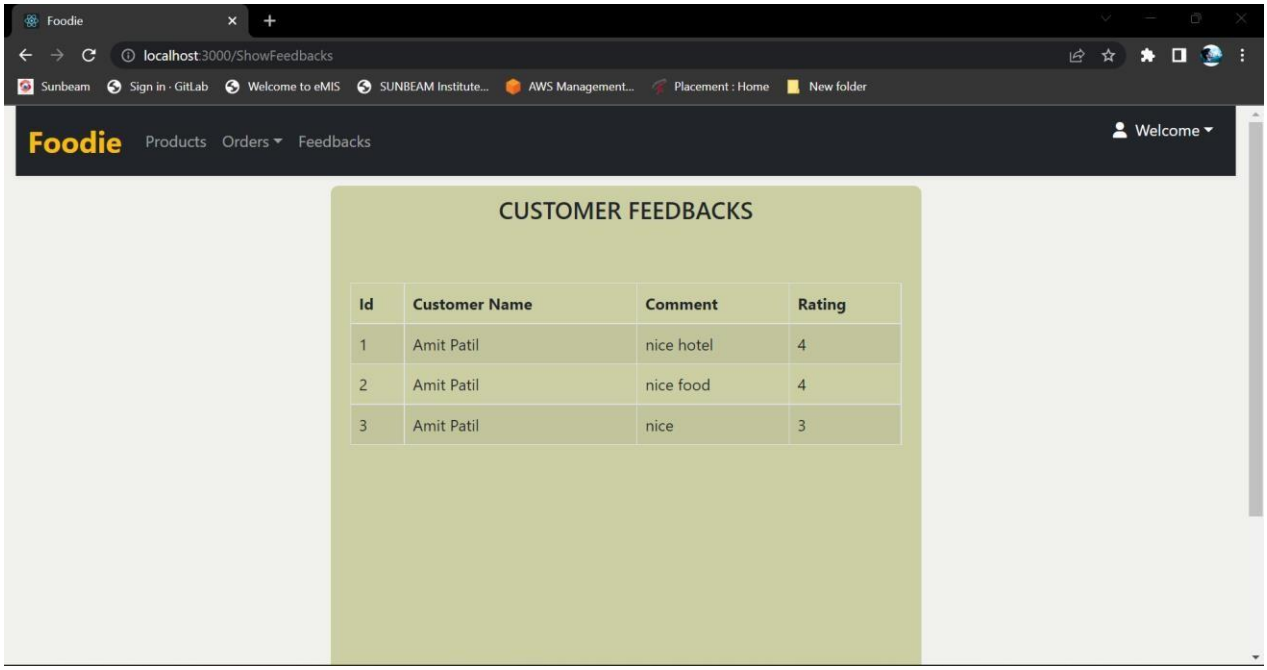

# ACCEPTED ORDERS



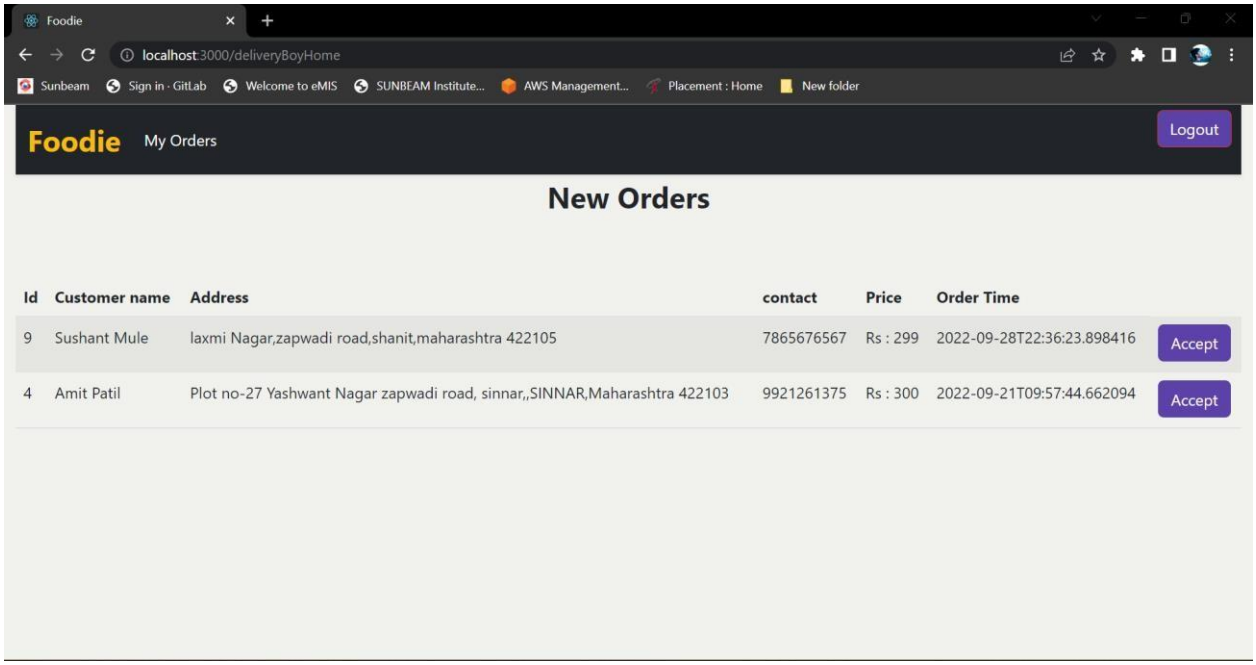
# UPDATE ORDERS STAUS



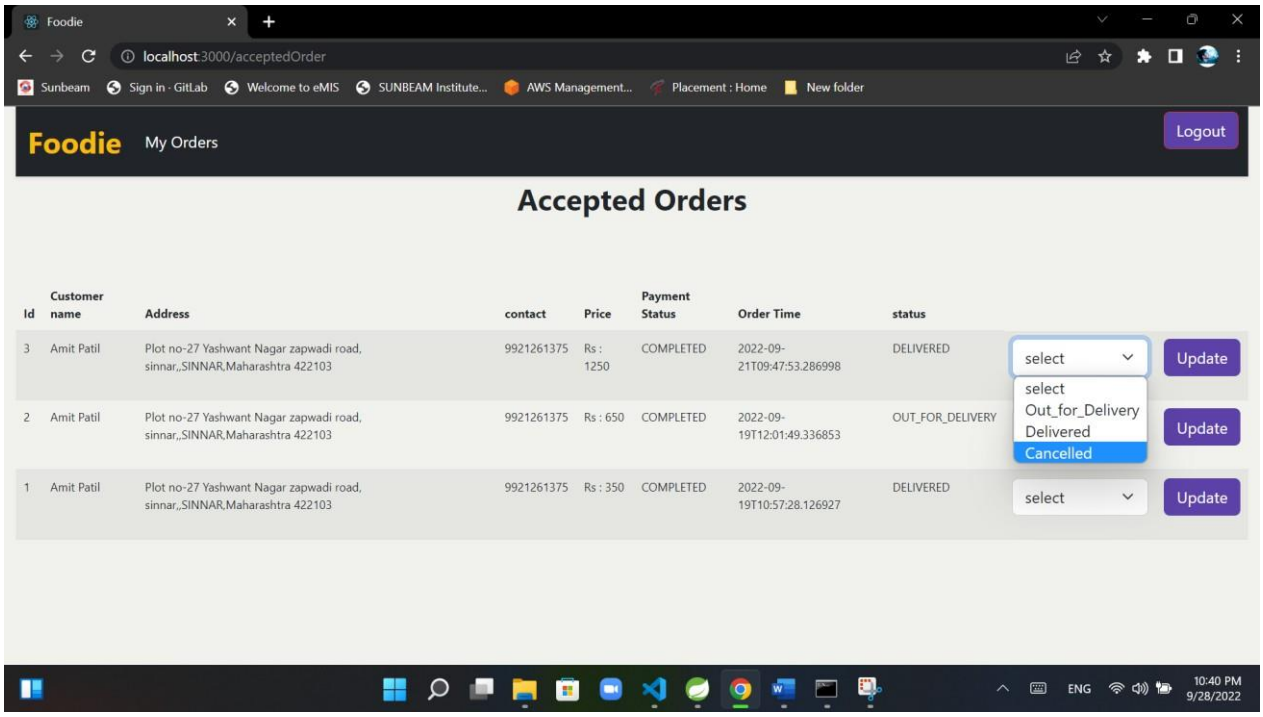

# SHOWING CUSTOMER FEEDBACKS



# DELIVER BOY HOME




# DELIVER BOY - UPDATE STATUS



# TESTING

To build up our project we used software testing process for executing a program with the intent of finding error that is uncovering errors in a program makes it a feasible task and also typing to find the errors (whose presence is assumed) in a program. As it is a destructive process. Types of testing we use in our project Here we just mentioned that how the testing is related to this software and in which way we have test the software? In our project we have used five types of testing this are listed below

**UNIT TESTING**

- Unit testing where individual program units or object class are tested here byusing this testing we have focus on testing functionality of the methods.

**MODULE TESTING**

- Where this is the combination of unit program is called module. Here wetested unit program is where the module program have dependency.

**SUB SYSTEM TESTING**

- Then we combined some module for the preliminary system testing in outproject.

**SYSTEM TESTING**

- Where it is combination of two or more sub system and then it is tested here we tested the entire system a per requirement.

## ACEEPTANCE TESTING

- Normally this type of testing is done to verify if system meets the customer specified requirements. After submitting this project to the user then they tested and to determine whether to accept the application. It is the system of testing performed by the customer to determine where they should accept thedelivery of system.

# CONCLUSION

Currently small and medium scale restaurants don't have synchronization between their task and customer. By making online food delivery system we have solved the problem from food store and customer end and more convenience is added to theexisting system.

Web application abridges the gap between the user and the hosts. This integrates basic amenities for the users especially the students in one platform. It further extends the feature of advertising the hosts' accommodation facilities over the website application. All in all this application will turn out to be a boon for all the students by providing them with a portable all-in-one application. None of the applications in the existing system support such a user friendly atmosphere where all the three features are merged into one integrated platform. There are many conclusive features in the website which suggests there can be further development and an outlook can be created for business perspective using various hosting platforms. The inclusion of cloud services makes it all the more remarkable. The integration of these platforms can make a subtle environment where a user can incur less data and also save time.

In future scope this system will be available with large scale database and can accommodate many services such as laundry, paying guest rooms. This system can also be developed on mobile application so that it can be access remotely.