



# ONLINE FOOD DELIVERY APP

**Presented by group 7**



# GROUP MEMBERS

Aliza Zaheer 2587565 (Team Lead & Frontend Lead)

Aileni Ranadeep 2587425 (Backend Lead )

Abhishek Sharma 2587324

Abothu Divya 2587778

Adepu Dharani 2587384

Shubham kumar singh 2587631

Aradhana Kumari 2587416

Arjit Pradhan 2587392

# contents

- INTRODUCTION
- PROJECT OVERVIEW
- TOOLS & TECHNOLOGY
- FUNCTIONALITY HIGHLIGHTS
- ER DIAGRRAM
- MODULES
- CONCLUSION

# Introduction

In today's fast-paced world, convenience and accessibility are driving significant transformations across industries. One such notable evolution is in the way we indulge in our favorite culinary delights.

The age-old practice of dining out has evolved into a seamless online experience, and FoodMunch, a renowned restaurant chain, is at the forefront of this transformation. Welcome to our presentation on "FoodMunch"

Our platform is designed with users in mind, providing a user-friendly experience that makes ordering food easy and enjoyable. From our intuitive search function to our secure payment system, we've thought of everything to ensure satisfaction.

# PROJECT OVERVIEW



[!\[\]\(7e0d6a31a51eb3952a6a6daebf7e401c\_img.jpg\) About Us](#) [!\[\]\(67f2870f14660d0c1a2612f703dff40d\_img.jpg\) Contact Us](#)

## Enjoy Our Delicious Meal

Indulge in the flavors of our exquisite dishes. Treat yourself to a gastronomic delight that will tantalize your taste buds and leave you craving for more.

**Domain: Food Delivery**

**Project Objective: Develop a dynamic and responsive online food delivery web application for ordering food items of different cuisines from a restaurant.**

# TOOLS AND TECHNOLOGIES

## FRONT-END DEVELOPMENT:

- Angular
- HTML, CSS, Bootstrap

## DATABASE MANAGEMENT:

- MySQL
- MySQL Workbench

## BACK-END DEVELOPMENT:

- Spring Boot
- Hibernate (JPA - ORM)
- Java

## TESTING:

- JUnit Testing
- Postman Testing

## SERVER:

- Tomcat

## VERSION CONTROL:

- Git

## INTEGRATED DEVELOPMENT ENVIRONMENTS (IDE'S):

- Spring STS
- Visual Studio Code (VS Code)

# FUNCTIONALITY HIGHLIGHTS

## Enhanced User Experience:

- Responsive Design
- Contact Us and About Us Sections

## User Interaction:

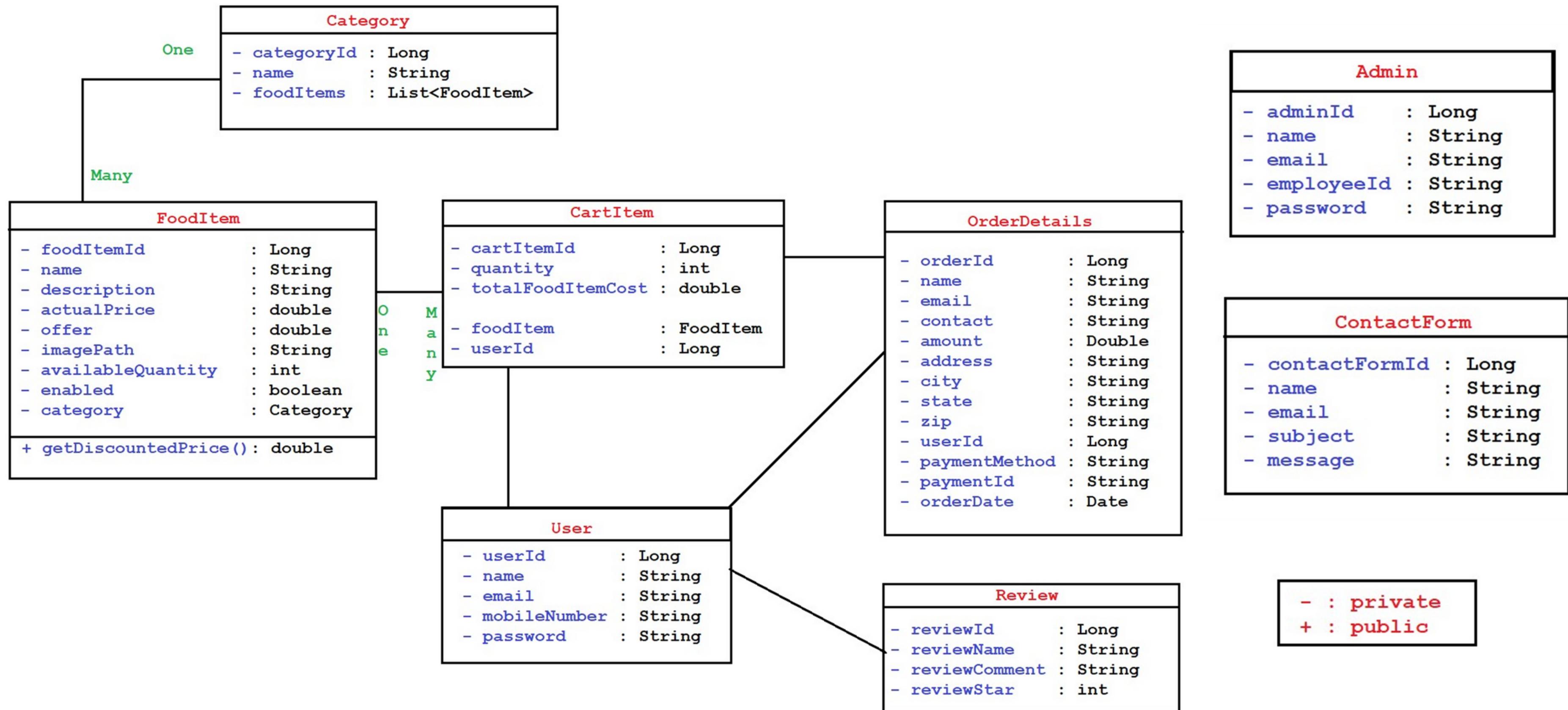
- Raise Queries

## Admin Portal:

- Efficient Management
- Cuisine and Food Management
- Query Management
- Review Management
- Order Details

## User Portal:

- User Registration
- Customized Orders
- Hassle-Free Payment

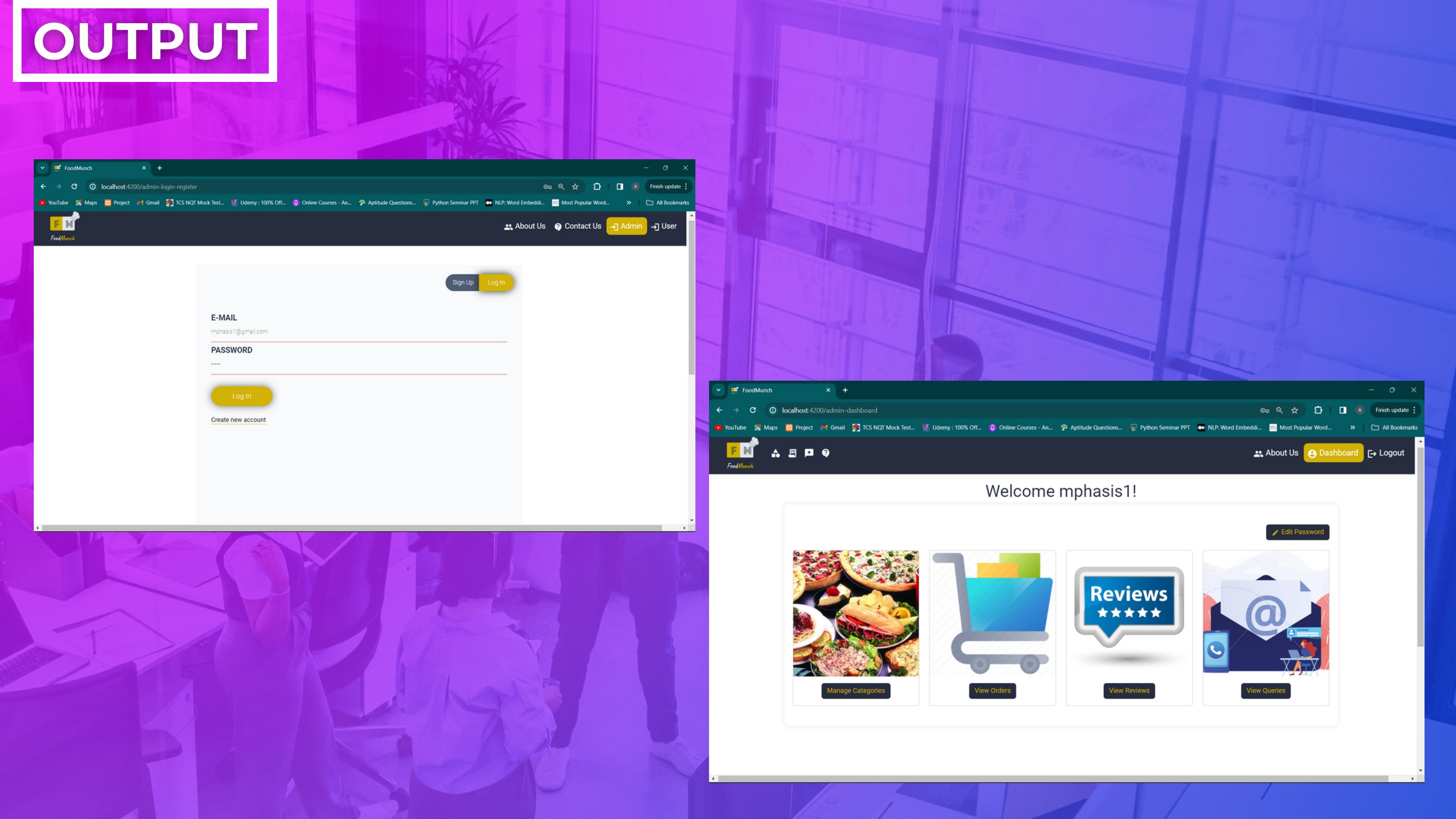


# ADMIN

## Functionalities

- Login to the system
- Add ,view, delete and update restaurants
- GET list order placed by user
- Manage rating of users
- View and manage queries

# OUTPUT



FoodMunch

localhost:4200/admin-login-register

About Us Contact Us Admin User

Sign Up Log In

E-MAIL  
mphasis1@gmail.com

PASSWORD  
.....

Log In Create new account

This screenshot shows the FoodMunch admin login page. It features a dark header with the brand logo and navigation links for About Us, Contact Us, Admin (which is highlighted in yellow), and User. Below the header is a form with fields for E-MAIL and PASSWORD. The E-MAIL field contains a placeholder email address. The PASSWORD field is represented by a series of red dots. At the bottom of the form are two buttons: a yellow 'Log In' button and a smaller link 'Create new account'.

FoodMunch

localhost:4200/admin-dashboard

About Us Dashboard Logout

Welcome mphasis1!

Edit Password

Manage Categories

View Orders

Reviews

View Reviews

View Queries

This screenshot shows the FoodMunch admin dashboard. The top navigation bar includes the brand logo, navigation links for About Us, Dashboard (highlighted in yellow), and Logout. A welcome message 'Welcome mphasis1!' is displayed above a grid of four cards. The first card, 'Manage Categories', features an image of various food items like pizzas and sandwiches. The second card, 'View Orders', shows a shopping cart icon. The third card, 'Reviews', displays a speech bubble with a 5-star rating. The fourth card, 'View Queries', depicts a person working at a computer with a phone and an envelope icon. Each card has a corresponding button at the bottom: 'Manage Categories', 'View Orders', 'View Reviews', and 'View Queries'.

# DB

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator admin admin

SCHEMAS

- book
- company
- exercise
- facial\_recognition
- food\_database
- foodbox**
  - Tables
  - admin
  - cart\_item
  - category
  - contact\_form
  - food\_item
  - orders
  - review
  - user
  - Views
  - Stored Procedures
  - Functions
- partymaster
- sys

Result Grid

```
1 •  SELECT * FROM foodbox.admin;
2 •  use foodbox;
3 •  select * from admin;
4 •  select * from contact_form;
5 •  select * from review;
6 •  select * from category;
7 •  select * from food_item;
8 •  select * from user;
9 •  select * from orders;
10 •  select * from cart_item;
```

admin_id	email	employee_id	name	password
1	alz@gmail.com	2587565	aliza	alz@123
2	mphasis1@gmail.com	1234	mphasis1	za@123
3	mphasis2	2345	mphasis2	lz@123
4	alz1@gmail.com	2587564	Aliza	Akz@12
*	HULL	HULL	HULL	HULL

Administration Schemas

Information

Table: food\_item

Columns:

food_item_id	actual_price	available_quantity	description	enabled	image_path	name	offer	category_id
bigint	double	int	varchar	bit(1)	varchar	varchar	double	bigint

Object Info Session

Output

Action Output

#	Time	Action	Message	Duration / Fetch
77	12:28:24	select * from orders LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
78	12:28:32	select * from cart_item LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
79	13:00:10	select * from contact_form LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
80	13:48:45	select * from admin LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec

Context Help Snippets

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

# Testing

The screenshot shows the Postman application interface. On the left, a sidebar titled "My Workspace" displays "Collections", "Environments", and "History". A modal dialog titled "Create a collection for your requests" is open, showing a preview of a collection named "Your collection" with an "Authorization" section containing an "API Key". Below the modal, there is descriptive text about collections and a "Create Collection" button.

The main workspace shows a POST request to "http://localhost:8084/admin/register". The "Body" tab is selected, showing raw JSON data:

```
1  {
2    ...
3  "name": "Mphasis3",
4  "email": "mphasis3@gmail.com",
5  "employeeId": 258746,
6  "password": "Mp@123"
7
8 }
```

The status bar at the bottom indicates "Status: 200 OK Time: 188 ms Size: 300 B".

At the bottom of the screen, there are various status icons and links: "Online", "Find and replace", "Console", "Postbot", "Runner", "Start Proxy", "Cookies", "Trash", and a question mark icon.

# USER

## Functionalities

- Login or Register to our App easily by one step.
- View Restaurants and add their favourite food from their preferred Restaurants.
- Manage their cart according to their needs.
- Give rating to food and the Restaurant
- Ask queries about the food and restaurant.

# SIGNUP PAGE

Sign Up Log In

FULL NAME

Bishal

E-MAIL

bishal@mail.com

Phone Number:

9342567855

CREATE PASSWORD

.....

Sign Up

I'm already a member

 FoodMunch

X    

About Us   Contact Us    Dashboard    Logout

# Welcome Bishal!



 Edit Password

[View Food Items](#)



[Manage Cart](#)



[Add review](#)

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1

SCHEMAS

- coursedb
- foodbox**
  - Tables: admin, cart\_item, category, contact\_form, food\_item, orders, review, user
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys
- world

SQLAdditions: Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Query 1

```
9
10 •   SELECT * FROM contact_form;
11
12 •   SELECT * FROM food_item;
13
14 •   SELECT * FROM orders;
15
16 •   SELECT * FROM review;
17
18 •   SELECT * FROM user;
```

Result Grid

user_id	email	mobile_number	name	password
1	beta@mail.com	0123456789	Krishna	Krish@123
2	bishal@mail.com	9342567855	Bishal	<b>bishal@123</b>
*	HULL	HULL	HULL	HULL

Administration Schemas

Information

Schema: foodbox

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	15:49:21	CREATE DATABASE foodbox	1 row(s) affected	0.016 sec
10	15:55:08	SELECT * FROM admin LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SC...	0.000 sec
11	15:55:24	SELECT * FROM admin LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
12	15:58:56	SELECT * FROM admin LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
13	16:03:01	SELECT * FROM user LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
14	16:15:27	SELECT * FROM user LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

! Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%.

Learn more  X

Overview Getting started GET localhost:8080/course ● POST localhost:8084/user/register ● + No environment

localhost:8084/user/register

POST localhost:8084/user/register

Params Authorization Headers (8) Body  Pre-request Script Tests Settings

None form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Abhishek",
3   "email": "abcd@mail.com",
4   "mobilenumber": "9471380928",
5   "password": "Abhi@30"
6 }
7 }
```

Cookies Beautify

Body Cookies Headers (8) Test Results Status: 200 OK Time: 19 ms Size: 300 B Save as example

Pretty Raw Preview Visualize Text 

1 Registered Successfully! Login to continue

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash

# Category (Adding Restaurant Name)

## Functionalities

- It's Admin based Functionality.
- The Admin will perform all the Crud Operations:
  - 1.The Admin will add the Restaurant by the HTTP Post Request.
  - 1.The Admin is able to view all the restaurant will HTTP Get request.
    - By using CategoryId to get the Particular restaurant
    - We can get all the restaurant Names
    - The Admin can Delete the Restaurant names by the HTTP Delete request
  - 3.By passing the CategoryId , the Admin will delete the particular Restaurant Name.

output

## Frontend Admin (Category):

Restaurant Management

Add New Restaurant

Peacock + Add

Restaurant List

Restaurant Name	Actions
Aardhanki Biryani	<span>Manage Food Items</span> <span>Delete</span>
Paradise	<span>Manage Food Items</span> <span>Delete</span>

- Admin is adding the Peacock Restaurant.
- Now we can check in the database.

	category_id	name
▶	3	Aardhanki Biryani
▶	2	Paradise
▶	1	Peacock
*	HULL	HULL

# Backend Admin Category:

- Here we are sending the data through the HTTP Post Method
- http://localhost:8084/category/add/Tool\_Gate

- Here We going to get a particular restaurant with HTTP Get method
- http://localhost:8084/category/1

Result Grid | Filter Rows: [ ]

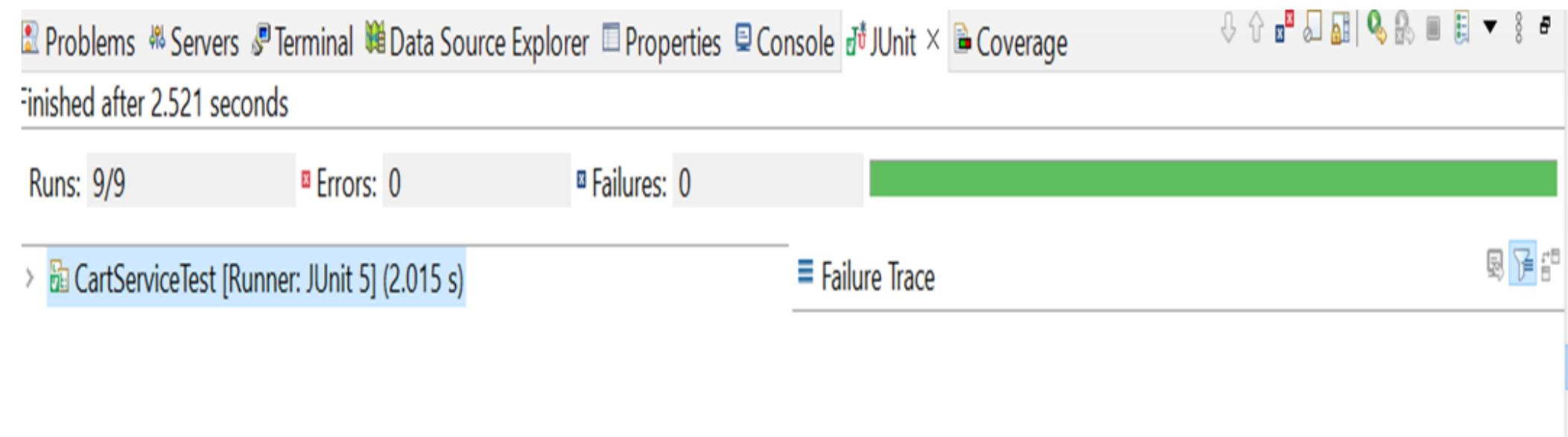
	category_id	name
▶	3	Aardhanki Biryani
	2	Paradise
	1	Peacock
	4	Tool_Gate
*	NULL	NULL

DUMP COOKIES HEADERS (0) TEST RESULTS

Pretty Raw Preview Visualize J

```
1 {  
2   "categoryId": 1,  
3   "name": "Peacock"  
4 }
```

# Admin Category Testing(Junit):



# FOODITEM

## Functionalities

- - Adding Food Items
- - Availability Status
- - Display List of Food Items
- - Detailed Item Descriptions with images
- - Search and Filter

# OUTPUT

### List of Food Items - Hot-N-Hot

Name & Description	Available Quantity	Actual Price	Offer %	Total Price	Status	Toggle Status	Image Preview
Butter Chicken Creamy tomato-based chicken curry	12	₹350.75	10%	₹315.68	Enabled	<button>Disable</button>	
Chocolate cake cake flavored with melted chocolate, cocoa powder, or both. Chocolate cake is made with chocolate	5	₹600.75	15%	₹510.64	Enabled	<button>Disable</button>	

### Update Butter Chicken Details

Name: Butter Chicken  
Description: Creamy tomato-based chicken curry  
Actual Price: 350.75  
Offer %: 10  
Available Quantity: 12  
 Enabled  
Image URL: <https://www.liicious.in/blog/wp-content/uploads/2020/10/butter-chicken-.jpg>

### FoodMunch

All Aardhanki Biryani Hot-N-Hot Pradise Tool\_Gate

Search for food items

  
Butter Chicken  
Hot-N-Hot  
Creamy tomato-based chicken curry  
Actual Price- ₹350.75  
Price: ₹315.68  
Available Quantity: 12

  
Chocolate cake  
Hot-N-Hot  
cake flavored with melted chocolate, cocoa powder, or both. Chocolate cake is made with chocolate  
Actual Price- ₹600.75  
Price: ₹510.64  
Available Quantity: 5

MySQL Workbench

Local instance MySQL

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- ecomap25
- ecommerce
- ecommerce
- ecommerce2
- ecommercea
- ecommercea
- ecommercea
- ecommercecp
- foodmunch**
- Tables
- Views
- Stored Pr
- Function
- railway
- sakila
- springboot
- sql\_practice

Administration

Information

No object selected

Query 1

SQLAdditions

My Snippets

1 • create database foodmunch;

2 • use foodmunch;

3 • select \* from admin;

4 • select \* from user;

5 • select \* from category;

6 • select \* from food\_item;

7 • select \* from cart\_item;

8 • select \* from orders;

9 • select \* from review;

10 • select \* from contact\_form;

11 • drop database foodmunch;

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

food_item_id	actual_price	available_quantity	description	enabled	image_path	name	offer	category_id
1	350.75	12	Creamy tomato-based chicken curry	1	https://www.licious.in/blog/wp-content/uploads...	Butter Chicken	10	1
2	600.75	5	cake flavored with melted chocolate, cocoa pow...	1	https://upload.wikimedia.org/wikipedia/common...	Chocolate cake	15	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

food\_item 5

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	07:13:51	use foodmunch	0 row(s) affected	0.000 sec
2	07:13:57	select * from food_item LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
3	08:17:01	select * from user LIMIT 0, 1000	1 row(s) returned	0.031 sec / 0.000 sec
4	08:25:26	select * from admin LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
5	10:36:31	select * from food_item LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec
6	11:36:35	delete from food_item where food_item_id = 4	1 row(s) affected	0.016 sec
7	11:36:39	select * from food_item LIMIT 0, 1000	2 row(s) returned	0.016 sec / 0.000 sec

≡ ← → Home Workspaces API Network Search Postman + Invite ⚙️ 🔔 👤 Upgrade — ×

POST http://localhost:8084/1 ● +

HTTP http://localhost:8084/foodItems/2

POST http://localhost:8084/foodItems/2

Save 🔗 📝

Params Authorization Headers (8) **Body** • Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1
2 "name": "Dosa",
3 "description": "A dosa is South Indian, fermented crepe made from rice batter and black lentils.",
4 "actualPrice": 145,
5 "offer": 12,
6 "imagePath": "https://sukhis.com/app/uploads/2020/01/Dosa.jpg",
7 "categoryId": 2,
8 "availableQuantity": 100,
9 "enabled": true
10
```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 153 ms Size: 258 B Save as example

Pretty Raw Preview Visualize Text

1 1

Online Find and replace Console Postbot Runner Start Proxy Cookies Trash ?

# CART

## Functionalities

- Its User based module
- • The user can food items to the cart, he/her can increase or decrease food items in the cart
- • User can add the N number of food items.
- • The admin can update the quantity.
- • Admin can view the userId.
- • User can delete the cart items.

# OUTPUT

FoodMunch

localhost:4200/cart

YouTube Maps Project Gmail TCS NQT Mock Test... Udemy : 100% Off... Online Courses - An... Aptitude Questions... Python Seminar PPT NLP: Word Embeddi... Most Popular Word... All Bookmarks

About Us Contact Us Dashboard Logout

Your Cart

Item	Preview	Quantity	Price	Total
Chessy 7 pizza		<button>-</button> 2 <button>+</button>	₹223.20×2	₹446.40
Burger		<button>-</button> 3 <button>+</button>	₹85.00×3	₹255.00

Cart Summary

Total Unique Items: 2

Total Cost: ₹701.40

[Checkout](#)

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 1 admin

SCHEMAS

- book
- company
- exercise
- facial\_recognition
- food\_database
- foodbox**
  - Tables
  - admin
    - Columns
    - Indexes
    - Foreign Keys
    - Triggers
  - cart\_item
  - category
  - contact\_form
  - food\_item
  - orders
  - review
  - user
- Views
- Stored Procedures
- Functions
- partymaster
- sys

Administration Schemas

Information

Table: admin

Columns:

<b>admin_id</b>	bigint AI PK
<b>email</b>	varchar(255)
<b>employee_id</b>	varchar(255)
name	varchar(255)
password	varchar(255)

Result Grid

cart_item_id	distance	quantity	total_food_item_cost	user_id	food_item_id
8	0	1	223.2	1	1
*	HULL	HULL	HULL	HULL	HULL

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
1 • use foodbox;
2 select * from admin;
3 • select * from contact_form;
4 • select * from review;
5 • select * from category;
6 • select * from food_item;
7 • select * from user;
8 • select * from orders;
9 • select * from cart_item;
```

Context Help Snippets

Action Output

#	Time	Action	Message	Duration / Fetch
9	10:08:14	select * from admin LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
10	10:12:07	select * from user LIMIT 0, 1000	3 row(s) returned	0.015 sec / 0.000 sec
11	12:27:45	select * from cart_item LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
12	12:28:52	select * from cart_item LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with sections for Collections, Environments, and History. A modal window titled 'Create a collection for your requests' is open, showing a template for a collection named 'Your collection' with an 'Authorization' tab selected. The main workspace shows a POST request to `http://localhost:8084/cart/add?userId=1&foodItemId=1&quantity=1`. The 'Body' tab is selected, showing the following JSON payload:

```
1 {
2   "name": "Smith",
3   "email": "smith.doe@example.com",
4   "subject": "Example2",
5   "message": "This is an example message for the contact form."
6 }
```

The response status is 200 OK, Time: 35 ms, Size: 289 B. The response body contains the message: Item added to cart successfully.

# ORDER

## Functionalities

- Order placement
- Delivery address management
- Promotions and discounts
- Payment processing

# OUTPUT

The screenshot shows a web browser window with two tabs open, both titled "FoodMunch". The active tab displays a "PAYMENT SUCCESSFUL" message with a delivery truck icon. Below the message, there is a button labeled "Please provide your Review". The page also shows "User: Rajani" and "Order Details" and "Payment Details" sections, each containing a table of data.

**User: Rajani**

**Order Details:**

Field	Value
address	Near Baba Temple
amount	510.64
city	Guntur
contact	9674464349
distance	5
email	ch.rajani@gmail.com
name	Rajani
orderDate	1711477800000
orderId	6
paymentId	C_O_D080718498471
paymentMethod	cash
state	Andhra Pradesh
totalAmount	570.64
userId	1
zip	183132

**Payment Details:**

Field	Value
Amount	₹570.64
Distance	5
Payment Method	cash

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sys

Tables Views Stored Procedures Functions

SQL File 1\*

Limit to 1000 rows

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows:

Export/Import:

Wrap Cell Content:

Result Grid

Form Editor

Context Help Snippets

No object selected

Output

Action Output

#	Time	Action	Message	Duration / Fetch
86	23:09:29	select * from orders LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
87	23:25:23	select * from admin LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
88	23:50:59	select * from orders LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

78°

ENG IN

01:06

27-03-2024

PRE

The screenshot shows the MySQL Workbench interface. In the SQL editor, several SELECT statements are listed, with the 10th statement being the current focus. The results grid displays a table with columns: order\_id, address, amount, city, contact, distance, email, name, and order\_date. The table contains five rows of data. The status bar at the bottom shows system information like battery level (78%), network connectivity, and system time (01:06, 27-03-2024).

≡ ← → Home Workspaces ▾ API Network ▾

Search Postman

Invite Upgrade

My Workspace New Import

PUT http://localhost:8084/api/order/createOrder POST http://localhost:8084/api/order/createOrder +

No environment

Collections + ⌂ New Collection

Environments

History

HTTP http://localhost:8084/order/createOrder

POST http://localhost:8084/order/createOrder

Save Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Body none  form-data  x-www-form-urlencoded  raw  binary  GraphQL Beautify

```
1 {  
2   "name": "Mphasis1",  
3   "email": "mphasis1@gmail.com",  
4   "contact": "7773458778",  
5   "amount": 652.0,  
6   "address": "Near b.b.high school field civil lines",  
7   "city": "Buxar",  
8   "state": "Bihar",  
9   "zip": 802131,  
10  "userId": 1,  
11  "paymentMethod": "cash",  
12  "distance": "4",  
13  "totalAmount": 682,  
14  "paymentId": "C_O_D1334694238572",  
15  "orderDate": "20240324"  
}
```

Body Cookies Headers (8) Test Results

200 OK 49 ms 1.77 KB Save as example

Pretty Raw Preview Visualize JSON

Online Find and replace Console

Postbot Runner Start Proxy Cookies Trash

78° ENG IN 01:06 27-03-2024 PRE

# REVIEW

## Functionalities

- Its User based module
- Only the user is able to give Review, based on their food he/her ordered.
- The Admin (Restaurant Owner) is able to view all reviews given by the user

“ good worth mawa worth



Thank you for your order!!  
Please Leave a Review

### Review Form

Name

Eshwar

Comments

Awesome

Review

4

Submit

<

Result Grid | Filter Rows: Edit:

	review_id	review_comment	review_name	review_star
▶	1	good worth mawa worth	Raj Kumar	5
●	2	Awesome	Eshwar	4
*	HULL	HULL	HULL	HULL

Result Grid | Filter Rows: Edit: Export

	review_id	review_comment	review_name	review_star
▶	1	good worth mawa worth	Raj Kumar	5
●	HULL	HULL	HULL	HULL

[Overview](#)[PUT http://localhost:8084/review](#)[PUT http://localhost:8084/review](#)[POST http://localhost:8084/review](#)[POS](#)<http://localhost:8084/review>[POST](#)<http://localhost:8084/review>[Params](#)[Authorization](#)[Headers \(8\)](#)[Body](#) ●[Pre-request Script](#)[Tests](#) none form-data x-www-form-urlencoded raw binary Gr

```
1 {  
2   ..... "reviewName": "Raj Kumar",  
3   ..... "reviewComment": "good worth mawa worth",  
4   ..... "reviewStar": 5  
5 }
```

[Body](#)[Cookies](#)[Headers \(8\)](#)[Test Results](#)[Pretty](#)[Raw](#)[Preview](#)[Visualize](#)[JSON](#)

```
1 {  
2   "reviewId": 1,  
3   "reviewName": "Raj Kumar",  
4   "reviewComment": "good worth mawa worth",  
5   "reviewStar": 5  
6 }
```



# CONTACTFORM

## Functionalities

- A structured contact form where users can easily submit their queries or messages
- This feature enhances user engagement by allowing direct communications
- Users receive an immediate notification confirming that their query has been received
- Features an integrated map (Google Maps) that provides location-based information

# OUTPUT

localhost:4200/contact-us

Your Query has been submitted successfully! We'll get back to you soon

OK

Contact Us

## Any Query? Feel free to drop a message

**Booking**  
✉ book@foodmunch.com

**General**  
✉ info@foodmunch.com

**Technical**  
✉ tech@foodmunch.com

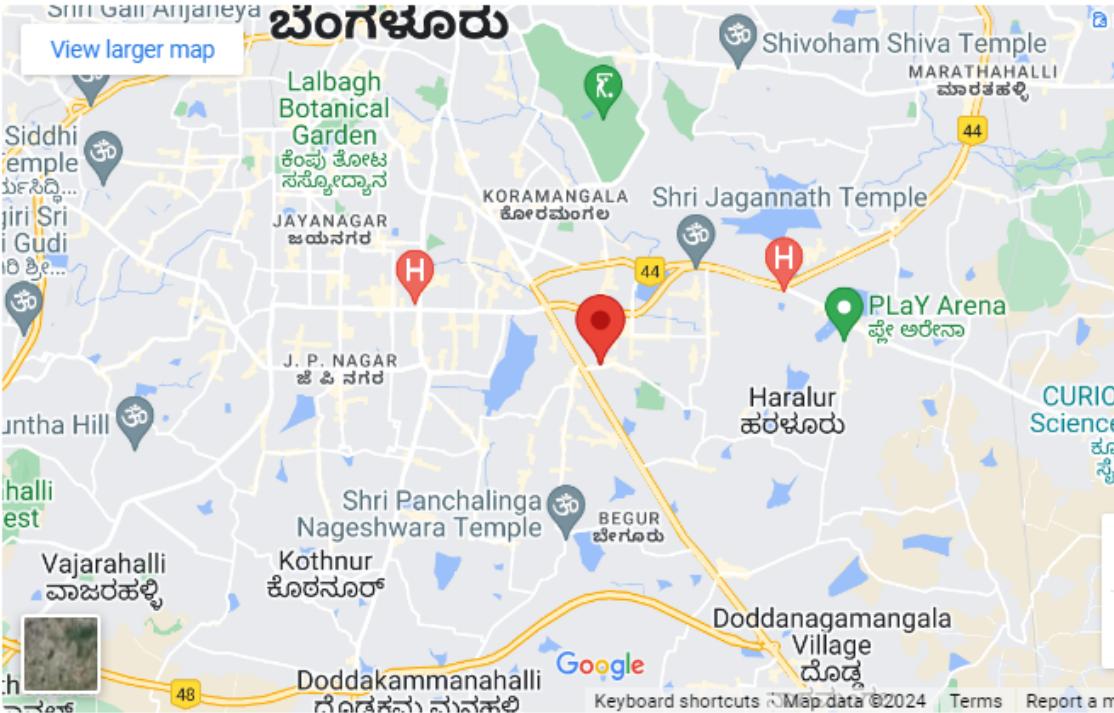
Your Name  
Aliza

Your Email  
aliza@gmail.com

Subject  
Example

Message  
This is an example query for contact form

Send Message



# DB

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator admin admin

SCHEMAS

- book
- company
- exercise
- facial\_recognition
- food\_database
- foodbox**
  - Tables
  - Views
  - Stored Procedures
  - Functions
- partymaster
- sys

Query 1 admin admin

1 • `SELECT * FROM foodbox.admin;`  
2 • `use foodbox;`  
3 • `select * from admin;`  
4 • `select * from contact_form;`  
5 • `select * from review;`  
6 • `select * from category;`  
7 • `select * from food_item;`  
8 • `select * from user;`  
9 • `select * from orders;`  
10 • `select * from cart_item;`

Result Grid

contact_form_id	email	message	name	subject
6	john.doe@example1.com	It is an example 2 message for the contact form.. issue with the food delivered	John Doe 2	Example Subject 2
7	scott@gmail.com		scott	food related
9	Kin@example.com	it is an example message for the contact form f...	Kin	Example2
10	aliza@gmail.com	This is an example query for contact form	Aliza	Example
*	HULL	HULL	HULL	HULL

Administration Schemas

Information

Table: food\_item

Columns:

food_item_id	bigint
actual_price	double
available_quantity	int
description	varchar
enabled	bit(1)
image_path	varchar
name	varchar
offer	double
category_id	bigint

Object Info Session

admin 2 contact\_form 72

Output

Action Output

#	Time	Action	Message	Duration / Fetch
76	12:28:18	select * from user LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
77	12:28:24	select * from orders LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
78	12:28:32	select * from cart_item LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
79	13:00:10	select * from contact_form LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

# Testing

The screenshot shows the Postman application interface. On the left, the sidebar displays 'My Workspace' with sections for Collections, Environments, and History. A collection named 'My first collection' is expanded, showing two folders: 'First folder inside collection' and 'Second folder inside collection', each containing several requests. Below the collection list, there is a button labeled 'Create a collection for your requests' and a descriptive text about collections. At the bottom of the sidebar, there is a 'Create Collection' button.

In the main workspace, a POST request is being configured for the URL `http://localhost:8084/contact-form`. The 'Body' tab is selected, showing the raw JSON payload:

```
1 {  
2   ...  
3   "name": "Kin2",  
4   "email": "Kin2@example.com",  
5   "subject": "Example3",  
6   "message": "message for the contact form"  
7 }  
8  
9  
10 }  
11  
12 }
```

The 'Headers' section shows 8 headers. The 'Body' section is set to 'raw' and 'JSON'. The 'Tests' and 'Settings' tabs are also visible. At the bottom of the request configuration, there are tabs for Body, Cookies, Headers (8), and Test Results, along with status information: Status: 200 OK, Time: 31 ms, Size: 328 B, and a 'Save as example' button.

At the very bottom of the interface, there are several status indicators and links: Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Trash, and Help.

# CONCLUSION

- In conclusion, our project "FoodMunch: Online Food Delivery System" aims to provide a seamless and user-friendly platform for users to order their favorite food items from various cuisines.
- By leveraging technologies such as Angular for the front-end and Spring Boot for the back-end, we've created a dynamic and responsive application that caters to both users and administrators.
- We've highlighted the significance of our application in addressing the changing dynamics of the food industry. The ability to manage cuisines, food items, reviews, and orders through the admin portal showcases the system's robustness.



**Thank you**