

INFSCI 2710: DBMS PROJECT REPORT

FOOD DELIVERY APPLICATION

Basic introduction:

Our Food Delivery application enables a user to place orders for delivery. It allows them to pick menu items, place orders and specify a delivery address and delivery time. The user can select menu items based on different restaurant and cuisine choices.

There are two types of logins:

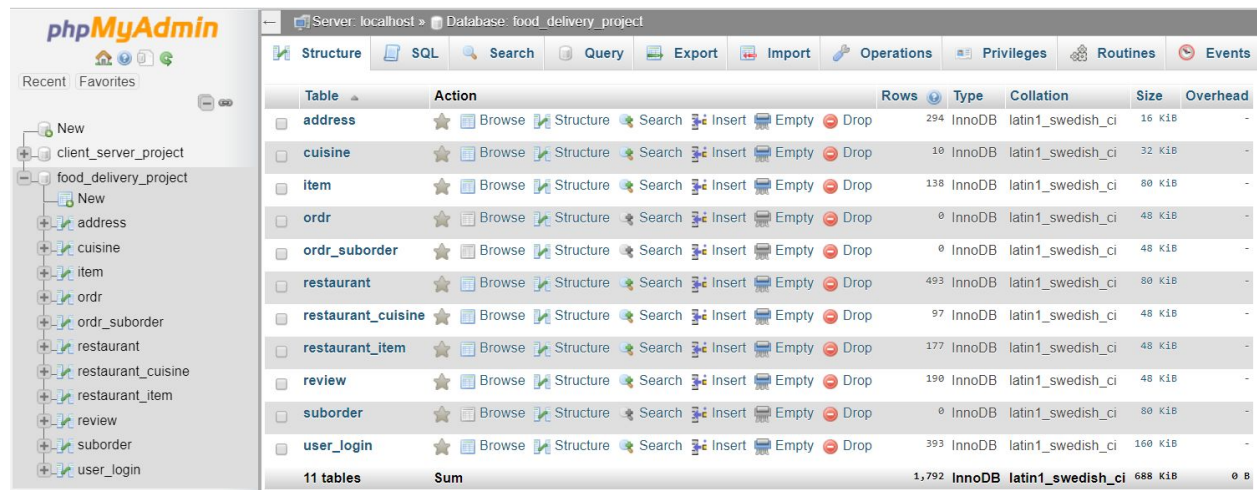
1. User login(customer).
2. Restaurant owner

The restaurant owner can add cuisines and different items under his restaurant.

The user can choose from different cuisines and select items from those restaurants.

The user can review the restaurants.

We have created the following tables:



The screenshot shows the phpMyAdmin interface for a database named 'food_delivery_project'. On the left, a tree view shows the database structure with tables: address, cuisine, item, ordr, ordr_suborder, restaurant, restaurant_cuisine, restaurant_item, review, suborder, and user_login. The main panel displays a table of 11 tables with their respective actions and statistics.

Table	Action	Rows	Type	Collation	Size	Overhead
address	★ Browse Structure Search Insert Empty Drop	294	InnoDB	latin1_swedish_ci	16 KiB	-
cuisine	★ Browse Structure Search Insert Empty Drop	10	InnoDB	latin1_swedish_ci	32 KiB	-
item	★ Browse Structure Search Insert Empty Drop	138	InnoDB	latin1_swedish_ci	80 KiB	-
ordr	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48 KiB	-
ordr_suborder	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48 KiB	-
restaurant	★ Browse Structure Search Insert Empty Drop	493	InnoDB	latin1_swedish_ci	80 KiB	-
restaurant_cuisine	★ Browse Structure Search Insert Empty Drop	97	InnoDB	latin1_swedish_ci	48 KiB	-
restaurant_item	★ Browse Structure Search Insert Empty Drop	177	InnoDB	latin1_swedish_ci	48 KiB	-
review	★ Browse Structure Search Insert Empty Drop	190	InnoDB	latin1_swedish_ci	48 KiB	-
suborder	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	80 KiB	-
user_login	★ Browse Structure Search Insert Empty Drop	393	InnoDB	latin1_swedish_ci	160 KiB	-
11 tables	Sum	1,792	InnoDB	latin1_swedish_ci	688 KiB	0 B

We have used the following technologies-

Database: MySQL

Backend: Java , Framework-Spring Boot

Frontend: HTML, CSS, Javascript

We implemented our server-side functionalities using REST web services and exposed them to the front-end. We used Hibernate to map Java objects to the MySQL database. We then consumed these services from the front-end by making REST calls in Javascript and built our HTML using this.

Assumptions:

1. We are assuming that the users are already registered as one of the three type of possible membership:

Silver- Get 5% discount on 1 order per month by paying a monthly fee of \$10.

Gold- Get 5% discount of 2 orders per month by paying a monthly fee of \$20

Platinum- Get 5% discount on every order by paying a monthly fee of \$28.

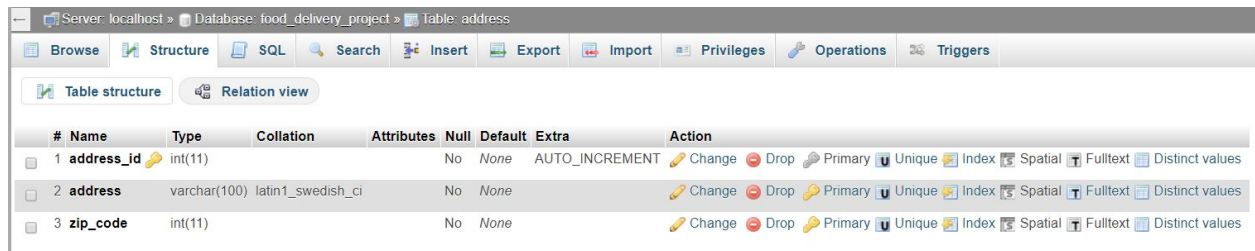
2. We have not creating a super admin. We are assuming that there exists a super admin who add and delete restaurants.

3. We are assuming that the payment gateway is already implemented.

Database Design:

The detailed description of each table is given below:-

1.address



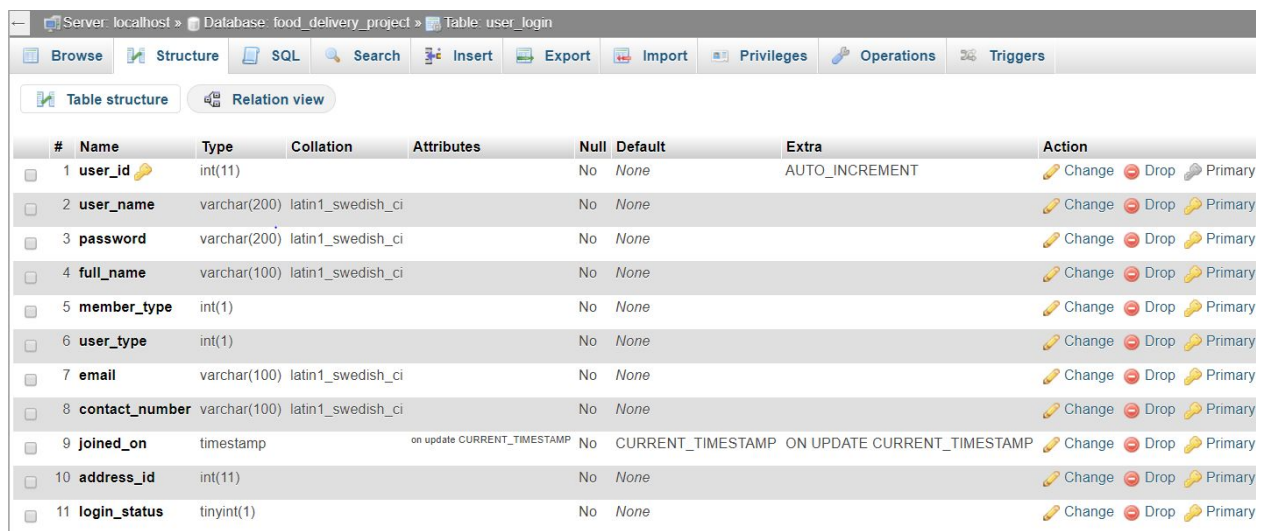
The screenshot shows the MySQL Table structure for the 'address' table. The table has three columns: 'address_id' (int(11), primary key, auto-increment), 'address' (varchar(100), latin1_swedish_ci), and 'zip_code' (int(11)). The 'address_id' column is the primary key and has a unique index. The 'address' column has a unique index. The 'zip_code' column has a unique index. The table is located in the 'food_delivery_project' database on the 'localhost' server.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	address_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	address	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	zip_code	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

This table will contain all the addresses of the restaurants and the users.

We have 299 address rows in our database

2. user_login



The screenshot shows the MySQL Table structure for the 'user_login' table. The table has 11 columns: 'user_id' (int(11), primary key, auto-increment), 'user_name' (varchar(200), latin1_swedish_ci), 'password' (varchar(200), latin1_swedish_ci), 'full_name' (varchar(100), latin1_swedish_ci), 'member_type' (int(1)), 'user_type' (int(1)), 'email' (varchar(100), latin1_swedish_ci), 'contact_number' (varchar(100), latin1_swedish_ci), 'joined_on' (timestamp, on update CURRENT_TIMESTAMP), 'address_id' (int(11)), and 'login_status' (tinyint(1)). The 'user_id' column is the primary key and has a unique index. The 'user_name' column has a unique index. The 'password' column has a unique index. The 'full_name' column has a unique index. The 'member_type' column has a unique index. The 'user_type' column has a unique index. The 'email' column has a unique index. The 'contact_number' column has a unique index. The 'joined_on' column has a unique index. The 'address_id' column has a unique index. The 'login_status' column has a unique index. The table is located in the 'food_delivery_project' database on the 'localhost' server.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	user_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	user_name	varchar(200)	latin1_swedish_ci		No	None		Change Drop Primary
3	password	varchar(200)	latin1_swedish_ci		No	None		Change Drop Primary
4	full_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary
5	member_type	int(1)			No	None		Change Drop Primary
6	user_type	int(1)			No	None		Change Drop Primary
7	email	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary
8	contact_number	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary
9	joined_on	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Change Drop Primary
10	address_id	int(11)			No	None		Change Drop Primary
11	login_status	tinyint(1)			No	None		Change Drop Primary

This table consist of all the login details. The user or the restaurant will use the user_name and password to login into their account. While registration they have to fill the details like full_name, address, contact_number.

user_id is the primary key.

address_id is the foreign key which is used here from the address table.

user_type field can have 2 values(0 or 1)

0= user

1= restaurant owner

2=super admin (assumption)

member-type field can have 3 values(1,2,3).

1= silver club member

2=gold club member

3=platinum club number

login_status shows us if the user or restaurant owner is logged in or not.

joined_on field shows the date when the user first created his or her account.

We have total 400 rows of user data in this table.

Our implementation might result in duplicate entries in the address column in the user_login table. So,we decided to create a new table called 'address' with foreign key(address_id) in user_login table, so that the second Normal form8 is satisfied.

3.cuisine

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	cuisine_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

This table will contain the list of the different cuisines from which the user can choose.

In this table we have listed 10 cuisines.

4.restaurant

Server: localhost » Database: food_delivery_project » Table: restaurant

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	rest_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial More
2	user_id	int(11)			No	None		Change Drop Primary Unique Index Spatial More
3	name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
4	city	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial More
5	zip_code	int(11)			No	None		Change Drop Primary Unique Index Spatial More

This table contains the list of restaurants along with their names and addresses. This also contains the user_id(foreign key) who is the owner of the particular restaurant. There are total 500 restaurants listed in this table.

5.item

Server: localhost » Database: food_delivery_project » Table: item

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	item_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	cuisine_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	item_name	varchar(100)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	description	varchar(500)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	price	double(20,2)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

This table consist of the list of items available in all the restaurants. Along with their name,description and the price.

Item_id is the primary key in this table.

cuisine_id is the foreign key used in this table from the cuisine table to show which item belongs to which cuisine.

We have 140 items listed in this table which belong to different cuisines.

6.suborder

Server: localhost » Database: food_delivery_project » Table: suborder									
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers									
Table structure Relation view									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	so_id	int(11)			No	None	AUTO_INCREMENT	Change	Drop Primary Unique Index More
2	user_id	int(11)			No	None		Change	Drop Primary Unique Index More
3	rest_id	int(11)			No	None		Change	Drop Primary Unique Index More
4	cuisine_id	int(11)			No	None		Change	Drop Primary Unique Index More
5	item_id	int(11)			No	None		Change	Drop Primary Unique Index More
6	quantity	int(50)			No	None		Change	Drop Primary Unique Index More
7	total_price	double(20,2)			No	None		Change	Drop Primary Unique Index More
8	spice_level	int(2)			No	None		Change	Drop Primary Unique Index More
9	comment	varchar(500)	latin1_swedish_ci		No	None		Change	Drop Primary Unique Index More
10	status	tinyint(1)			No	None		Change	Drop Primary Unique Index More

This table will consist each suborder that means if a user chooses one item then it will be treated as a suborder. Along with the order the user will provide the quantity, spice level and comments if required for that specific item. Total price here means the item price multiplied by the quantity.

Status field is a boolean type which tells the status of the items added to the cart

This table consist of 4 foreign keys :

user_id from the user_login table,session_id from the session_details table,3rest_id from the restaurant table and cuisine_id from the cuisine table.

7.ordr

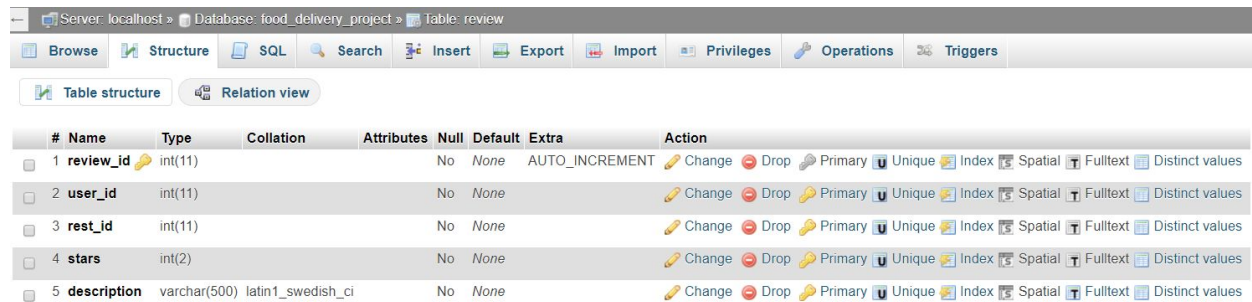
Server: localhost » Database: food_delivery_project » Table: ordr									
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers									
Table structure Relation view									
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action	
1	ordr_id	int(11)			No	None	AUTO_INCREMENT	Change	Drop Primary Unique More
2	user_id	int(11)			No	None		Change	Drop Primary Unique More
3	rest_id	int(11)			No	None		Change	Drop Primary Unique More
4	total_price	double(20,2)			No	None		Change	Drop Primary Unique More
5	delivery_address	varchar(100)	latin1_swedish_ci		No	None		Change	Drop Primary Unique More
6	estimated_delivery_time	datetime			No	None		Change	Drop Primary Unique More
7	delivery_flag	tinyint(1)			No	1		Change	Drop Primary Unique More

This table will consist of the final order of the user along with the total price, delivery address and the estimated delivery time.

order_id id the primary key in this table.

user_id and rest_id are foreign keys from the user_login and restaurant tables respectively.

8.review



The screenshot shows the MySQL Table structure for the 'review' table. The table has five columns: review_id, user_id, rest_id, stars, and description. review_id, user_id, and rest_id are primary keys. The description column has a collation of latin1_swedish_ci.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	review_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	user_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	rest_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	stars	int(2)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	description	varchar(500)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

This table consists the reviews given by the user to the different restaurants. The user can give upto 10 stars to a restaurant. The user can also add any comments in description.

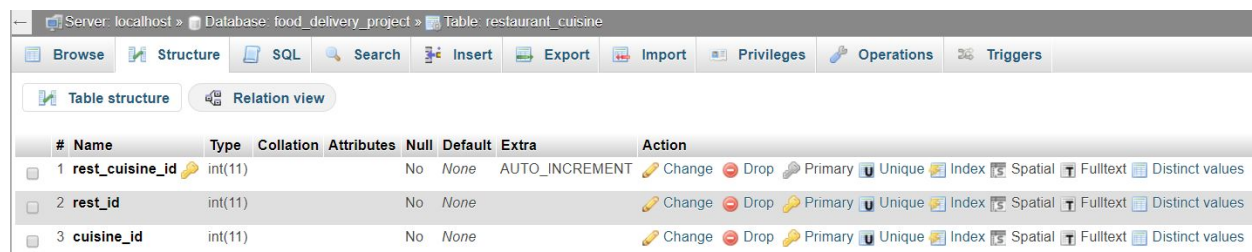
Here there are 2 foreign keys :

user_id from the user_login table

rest_id from the restaurant table

There are total 200 reviews in the table.

9.restaurant_cuisine



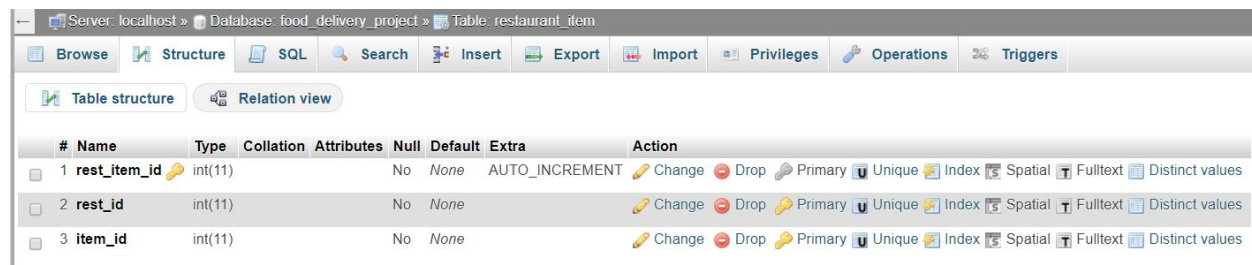
The screenshot shows the MySQL Table structure for the 'restaurant_cuisine' table. The table has three columns: rest_cuisine_id, rest_id, and cuisine_id. rest_cuisine_id and rest_id are primary keys.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	rest_cuisine_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	rest_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	cuisine_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

This is the junction table which we have created with both the foreign keys rest_id from the restaurant table and the cuisine_id from the cuisine table. This shows which cuisines are available in the different restaurants.

There are 100 rows in this table.

10. restaurant_item



The screenshot shows the MySQL Table structure for the 'restaurant_item' table. The table has three columns: rest_item_id, rest_id, and item_id. rest_item_id and rest_id are primary keys.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	rest_item_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	rest_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	item_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

This is a junction table which consists of the foreign keys rest_id from the restaurant table and the item_id from the item table. This table shows the list of items which are available in different restaurants.

There are 180 entries in this table.

11. ord_r_suborder

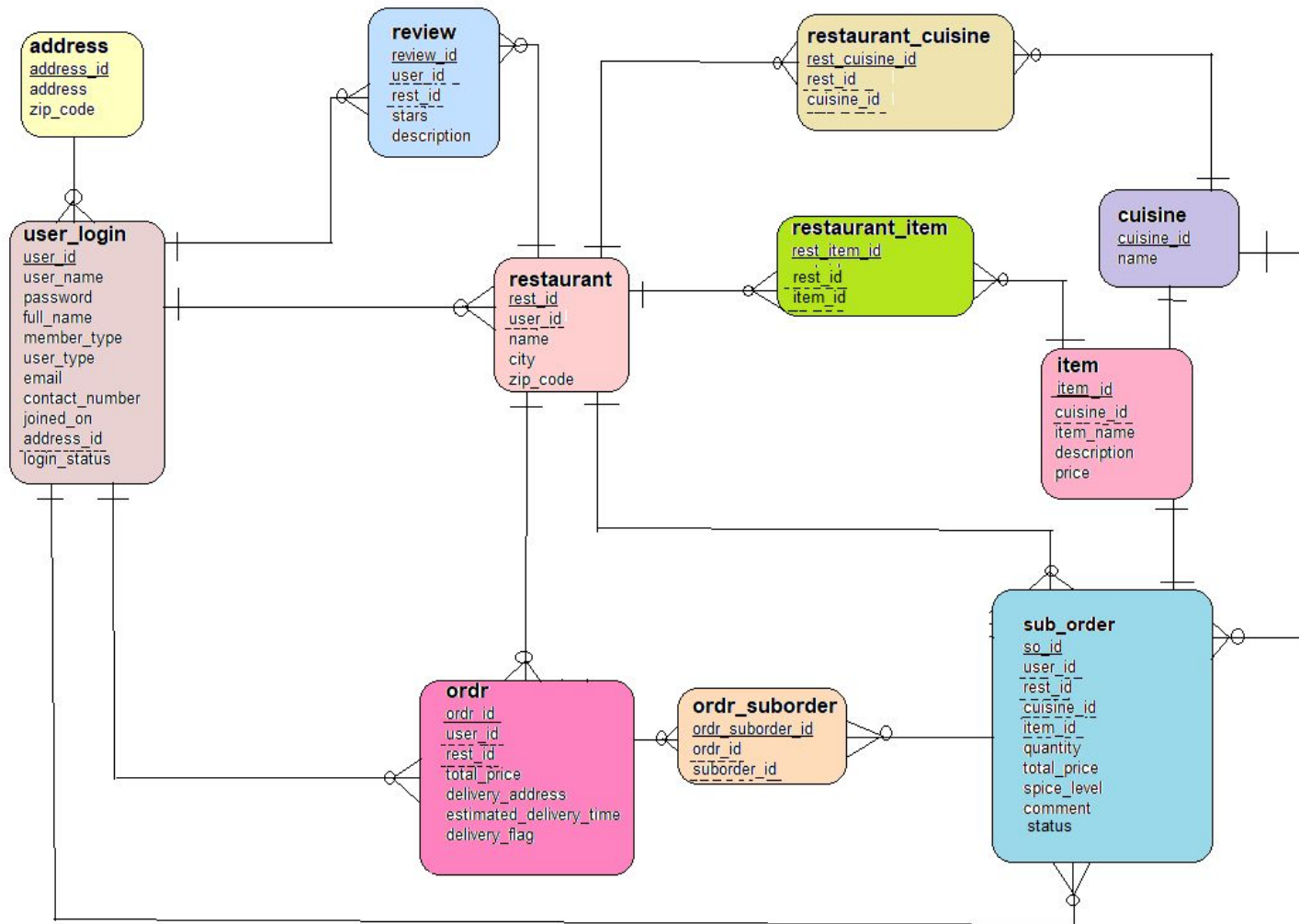


The screenshot shows a database management interface with a toolbar at the top containing buttons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below the toolbar, there are tabs for 'Table structure' and 'Relation view'. The 'Table structure' tab is active, displaying a table with three columns: #, Name, and Type. The table lists three fields: 1. 'ord_r_suborder_id' of type 'int(11)', 2. 'ord_r_id' of type 'int(11)', and 3. 'suborder_id' of type 'int(11)'. Each field has a set of icons for actions like Change, Drop, Primary, Unique, Index, Spatial, Fulltext, and Distinct values.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ord_r_suborder_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	ord_r_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	suborder_id	int(11)			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values


This is a junction table which consist of two primary keys ord_r_id which belongs to ord_r table and the suborder_id which belongs to the suborder table. This is a list which shows that which suborders belong to which order.

ER Diagram:



Application screenshots and flow:

Baked Food Delivery App Home About us Contact Order Food? Logout



Register User

Username:

Password:

Full Name:

User Type:
☒ User
☐ Restaurant Owner
☐ Admin

Email ID:

Contact No:


Address:

Zip code:

[Already a member? Sign in](#)

User registration page where the user can sign up by entering his/her details

Baked Food Delivery App Home About us Contact Order Food? Edit Profile Logout



Login

Password:

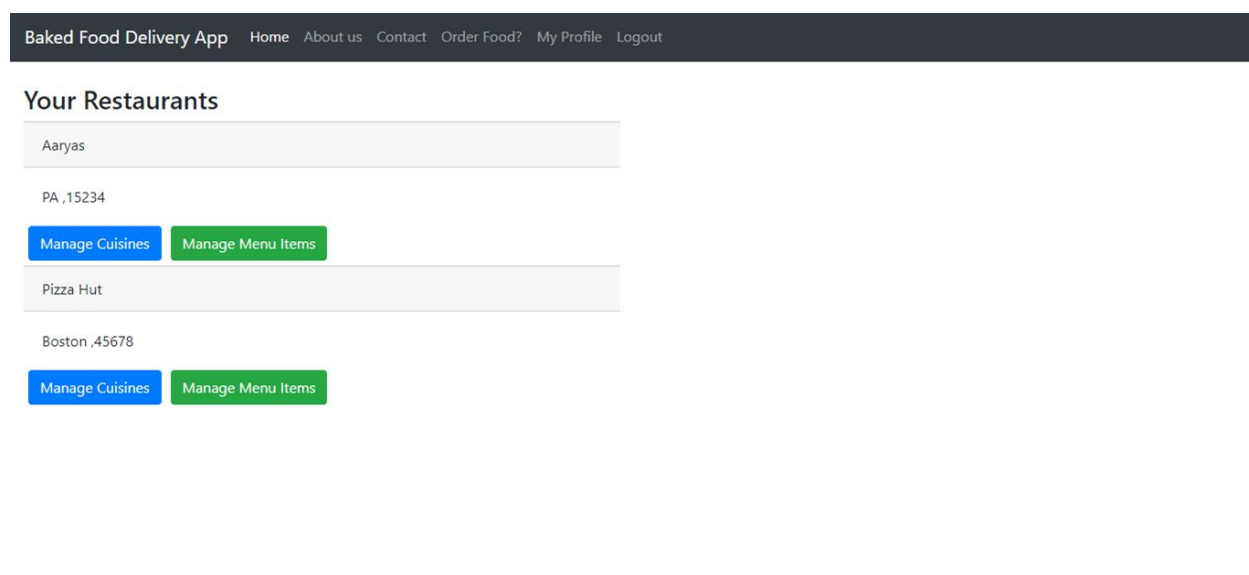
☒ User
☐ Restaurant Owner
☐ Admin

[Not a member? Register here](#)

Login page for the user. Username and Password are validated against the database.

User type: Restaurant owner

Restaurant owner home page. View restaurants page, where the restaurant owner can view all the restaurants owned by him.



Fig(a)

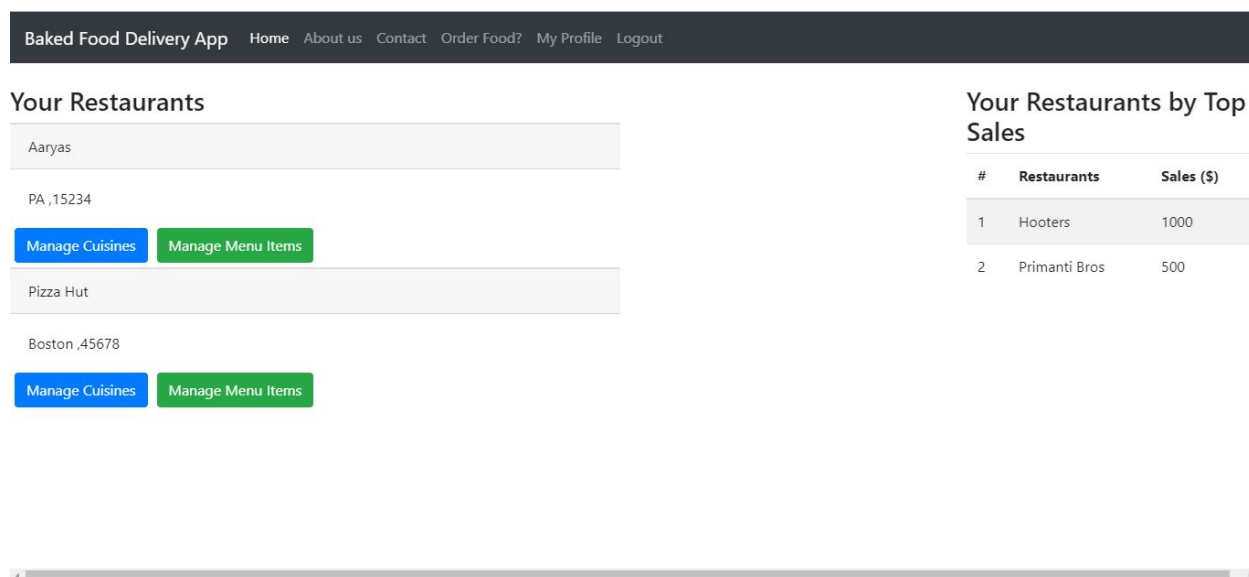


Fig. To view the restaurant with the top sales

Baked Food Delivery App
Home
About us
Contact
Order Food?
Edit Profile
Logout

Add Cuisine

Cuisine:

Your Cuisines

Continental	<input type="button" value="Update Cuisine"/>	<input type="button" value="Delete Cuisine"/>
Indian	<input type="button" value="Update Cuisine"/>	<input type="button" value="Delete Cuisine"/>
English	<input type="button" value="Update Cuisine"/>	<input type="button" value="Delete Cuisine"/>

Fig(b)

Add/manage cuisine. The restaurant owner can add/view/update/delete cuisines under a restaurant here.

Baked Food Delivery App
Home
About us
Contact
Order Food?
Edit Profile
Logout

Add Menu Items

Select Cuisine:

Menu Item:

Description:

Price:

Your Menu Items

Chinese

Idli: \$15	<input type="button" value="Update Item"/>	<input type="button" value="Delete Item"/>
Dosa: \$7	<input type="button" value="Update Item"/>	<input type="button" value="Delete Item"/>

Indian

Butter Chicken: \$50		
----------------------	--	--

Fig(a)

Add/manage items. The restaurant owner can add/view/update/delete items under a cuisine and a restaurant here.

A typical implementation:

1. Restaurant Owner homepage has a list of restaurants that he owns. For every restaurant, the owner can add Cuisines and menu items to a given restaurant as shown in Fig (a)
2. On selecting Manage Cuisines, the restaurant owner is redirected to a page where he can create new cuisines as well as manage existing cuisines as shown in Fig(b)
3. On selecting Manage Items, the restaurant owner is redirected to a page where he can create new items as well as manage existing items under a cuisine

User Type: User

Baked Food Delivery App [Home](#) [About us](#) [Contact](#) [Order Food?](#) [View Cart](#) [Past Orders](#) [My Profile](#) [Logout](#)

Order Food!

Select City:

Cranbrook ▾

Find Food!

Baked Food Delivery App

[Home](#) [About us](#) [Contact](#) [Order Food?](#) [View Cart](#) [Past Orders](#) [View Reviews](#) [My Profile](#) [Logout](#)

Order Food!

Select City:

Capena

Find Food!

Restaurants

Et Arcu Imperdiet PC

Montreal 54963

View Cuisines

Top Rated Restaurants

Siachen

Kalken Mahal

Vivamus Euismod Urna Associates

Fig. To view the top rated restaurants

(a) Select city for placing order

Baked Food Delivery App

[Home](#) [About us](#) [Contact](#) [Order Food?](#) [View Cart](#) [Past Orders](#) [My Profile](#) [Logout](#)

Order Food!

Select City:

Cranbrook

Find Food!

Restaurants

Et Arcu Imperdiet PC

Montreal 54963

View Cuisines

(b) Select restaurants under the chosen city

Baked Food Delivery App

[Home](#)[About us](#)[Contact](#)[Order Food?](#)[View Cart](#)[Past Orders](#)[My Profile](#)[Logout](#)

Order Food!

Select City:

Cranbrook

Find Food!

Restaurants

Et Arcu Imperdiet PC

Montreal 54963

View Cuisines

Cuisines

Continental

View items

Indian

View items

English

View items

(c) Choose from a list of cuisines for a particular restaurant

Baked Food Delivery App

[Home](#)[About us](#)[Contact](#)[Order Food?](#)[View Cart](#)[Past Orders](#)[My Profile](#)[Logout](#)

Order Food!

Select City:

Cranbrook

Find Food!

Restaurants

Et Arcu Imperdiet PC

Montreal 54963

View Cuisines

Cuisines

Continental

View items

Indian

View items

English

View items

Items

Noodles \$20

Noodles

Add

Idli \$15

Idli

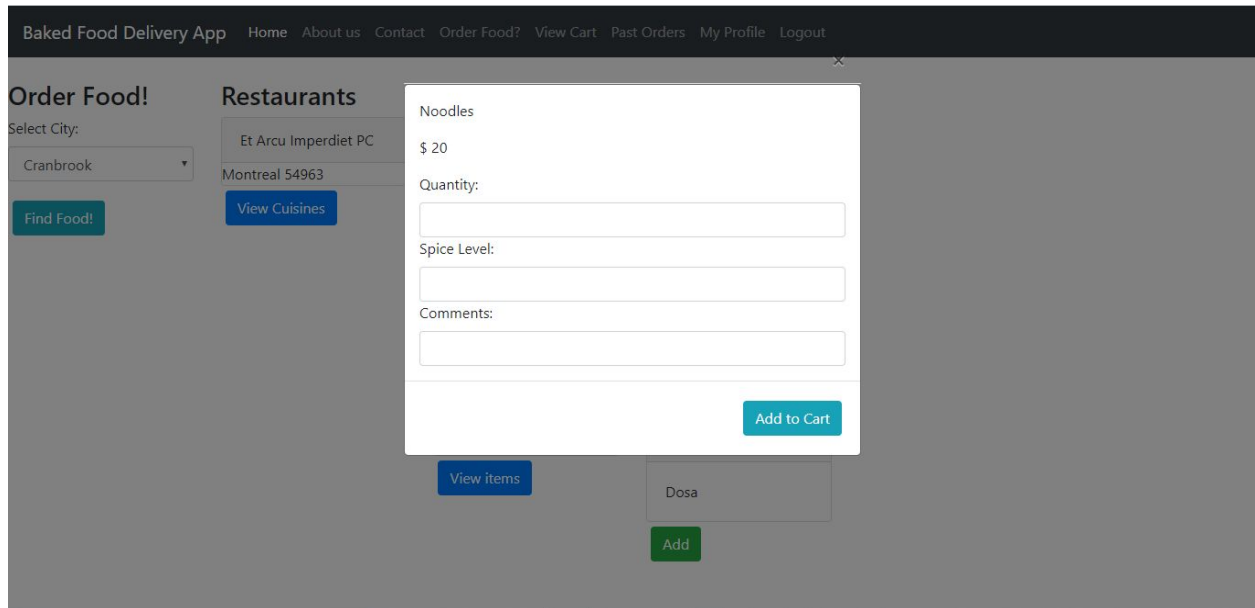
Add

Dosa \$7

Dosa

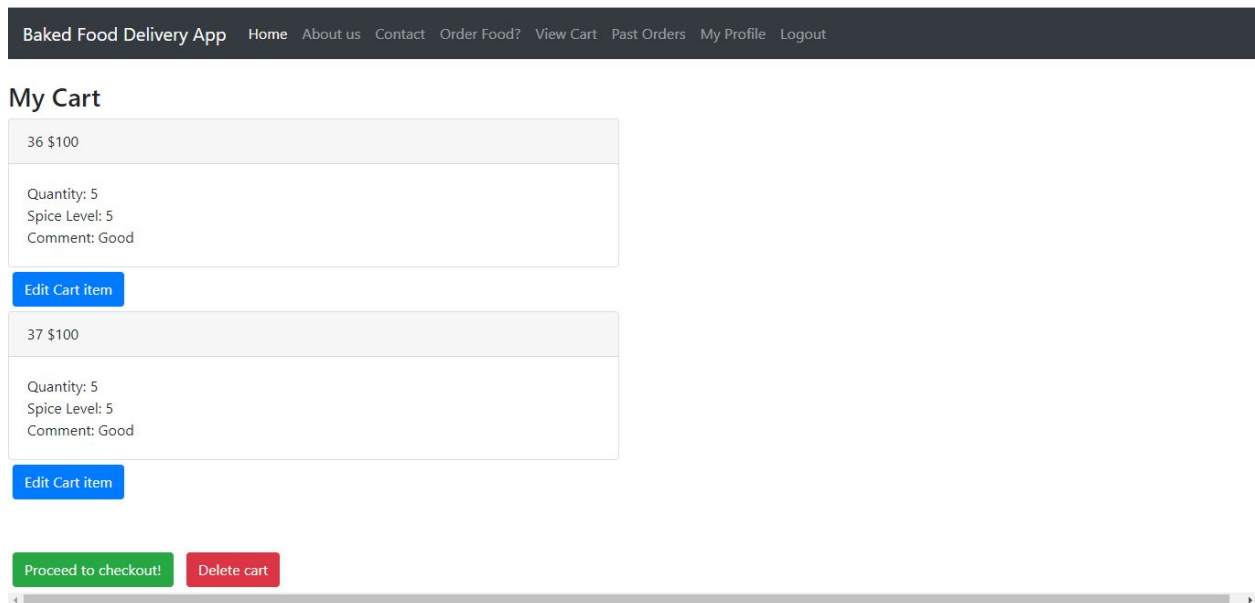
Add

(d) Choose from a list of menu items under a cuisine and restaurant



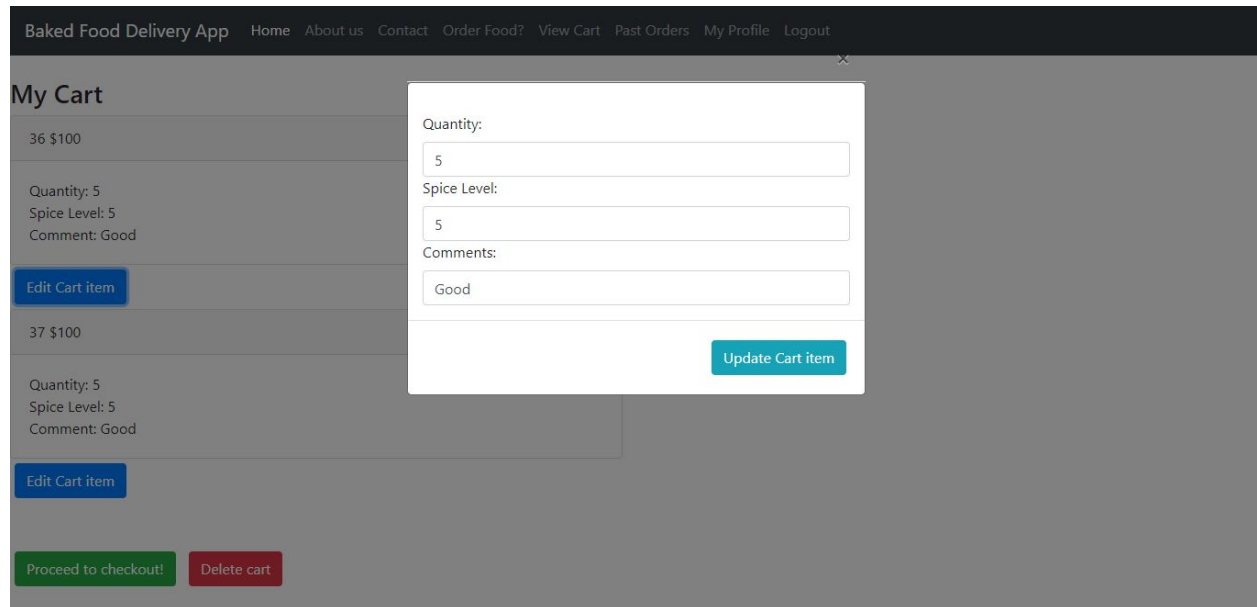
fig(e)

On clicking add to cart for a particular menu item, the user can decide upon the quantity of the order, the spice level and any special comments with respect to that particular menu item. Add to Cart allows users to add items to cart for future purposes as shown in fig(e).

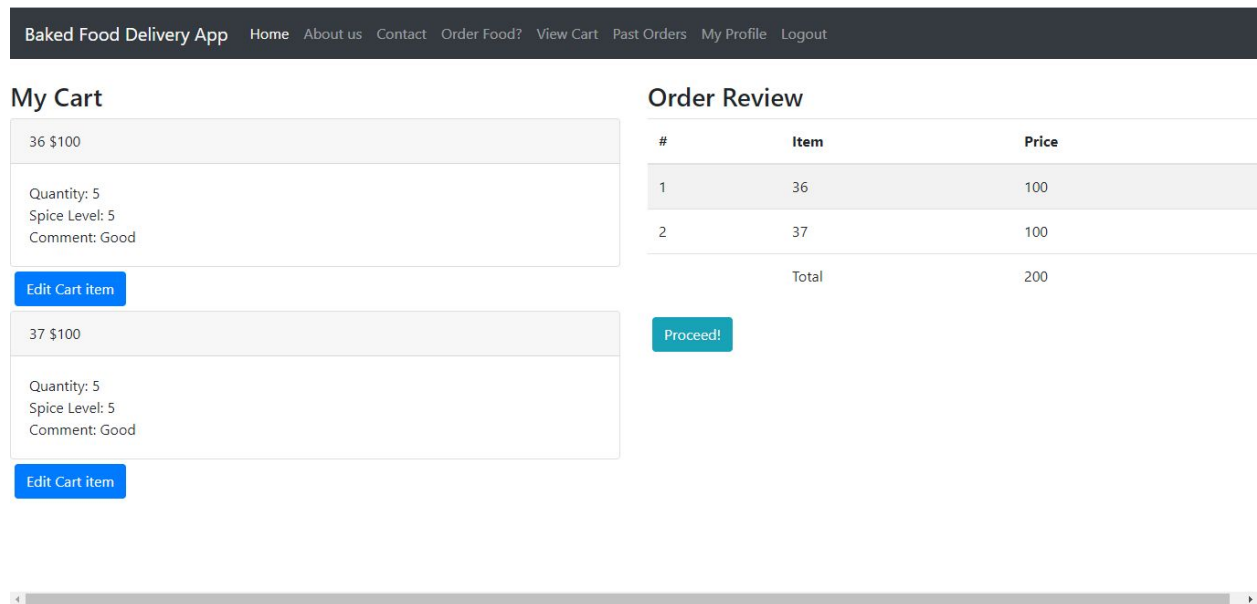


fig(f)

The user cart functionality gives the user the opportunity to edit items before placing the order. This page also allows the user to proceed to checkout if he finds that there are no changes required with respect to the menu items in the cart. The user also has the option to delete the cart as seen in fig(f)



fig(g)



fig(h)

After clicking Proceed to Checkout, Order Review section shows the final order comprising of items and price associated with each item. It also shows the total price of the order and gives the user the option to go back or proceed with the order.

The screenshot shows the 'Baked Food Delivery App' interface. At the top, there is a navigation bar with links: Home, About us, Contact, Order Food, My Cart, Past Orders, My Profile, and Logout. The main content area is divided into two sections. On the left, under the heading 'My Cart', there are two items. Each item shows a quantity of 5, a spice level of 5, and a comment of 'Good'. The first item has a total price of \$100, and the second item has a total price of \$100. Each item has an 'Edit Cart item' button. On the right, there is a table with a header 'Price' and three rows of prices: 100, 100, and 200. A modal window is open in the center, titled 'Delviery address:' (note the typo). It contains two input fields: one for the address and one for the zip code. Below the input fields are two buttons: 'Create Order' and 'Proceed!'. The 'Proceed!' button is highlighted in teal.

fig(i)

After Clicking Proceed, the user has to specify the delivery address and zip code before placing the order. On clicking Create Order, delivery time is estimated on the basis of the distance between the restaurant and delivery address zip codes by consuming a public REST API which returns the distance between the two zip codes. The user is prompted with a message stating whether the order was successful and the user is directed to the Past Orders page as shown in fig (i) and fig(j).

Baked Food Delivery App

[Home](#) [About us](#) [Contact](#) [Order Food?](#) [View Cart](#) [Past Orders](#) [View Reviews](#) [My Profile](#) [Logout](#)

Create Review

Select Restaurant:

Stars:

Description:

Create Review

fig(j)

To create review of a restaurant

Baked Food Delivery App

[Home](#) [About us](#) [Contact](#) [Order Food?](#) [View Cart](#) [Past Orders](#) [Edit Profile](#) [Logout](#)

Your Past Orders

Delivered- 2019-04-16 at 04:00:00 PM
70 Quantity: 2 Price: \$10
70 Quantity: 2 Price: \$10
70 Quantity: 2 Price: \$10
Total Price: 45
Delivered- 2019-04-16 at 04:00:00 PM
70 Quantity: 2 Price: \$10
70 Quantity: 2 Price: \$10
70 Quantity: 2 Price: \$10
Total Price: 120

fig(k)

Past Orders page displays the orders placed by the user.

A typical implementation:

The order food functionality is present on the user home page itself for ease of access. (Primary function of user after logging in is to order food)

As seen from fig(a), user first selects the city to begin his order. Once a city is selected, the page displays the list of restaurants present in that city under the “Restaurants” section.

As seen from fig (b), user then selects the restaurant of his choice from a list of a restaurants available by selecting “Choose cuisines” functionality.

As seen from fig(c), Cuisines section shows the list of available cuisines for a selected restaurant. Upon selecting a particular cuisine,a list of items belonging to a particular cuisine is displayed under Items section as shown in fig(d)

The Add option allows user to add individual items to cart.

Testing Efforts:

- 1) We performed unit testing on the different pages in the application.

Limitations

- 1) Only 1 item can be added to a cart at a time.

Future Improvements:

- 1) Implementing recommendation systems employing Data mining and machine learning techniques.