Indian Institute of Technology, Indore

CS 257

DBIS Project

Restaurant Management (Khazana) Database for IIT Indore

Xomato of IIT Indore

Submitted by- Shaikh Ubaid and Ruchir Mehta

Introduction:

Restaurants are the places where they have to deal with huge databases handling the menu and the order placed by multiple customers simultaneously. We as customers face a lot of problems while choosing a place and food items to dine. We want that we eat our desired food items which also has good rating points and great reviews and would we great if it is among the trending items of that restaurant. To rectify this issue we came up with a platform that will take care of all your needs and provide you with a facility to order food from home after a proper procedure of logging in into your account.

Project Description :

We as students face many problems and even outsiders many times face some difficulties while choosing a restaurant at mealtime on our campus. Out institute offers various degrees viz. BTech, MTech, PhD, MSc and there are different prices for students pursuing different degrees. While choosing a restaurant, we do not know whether it is open or close and even if we know the menu and prices are unknown. Secondly, many items of a restaurant do not become popular since they remain hidden from customers due to unawareness about them among the customers.

A pocket-friendly person wants that he gets his desired food item in his budget which has nice ratings and reviews and is currently a trending item of that restaurant. There are about 10 restaurants which generally makes the customer confused in choosing a particular place to eat. Hence, we came up with an

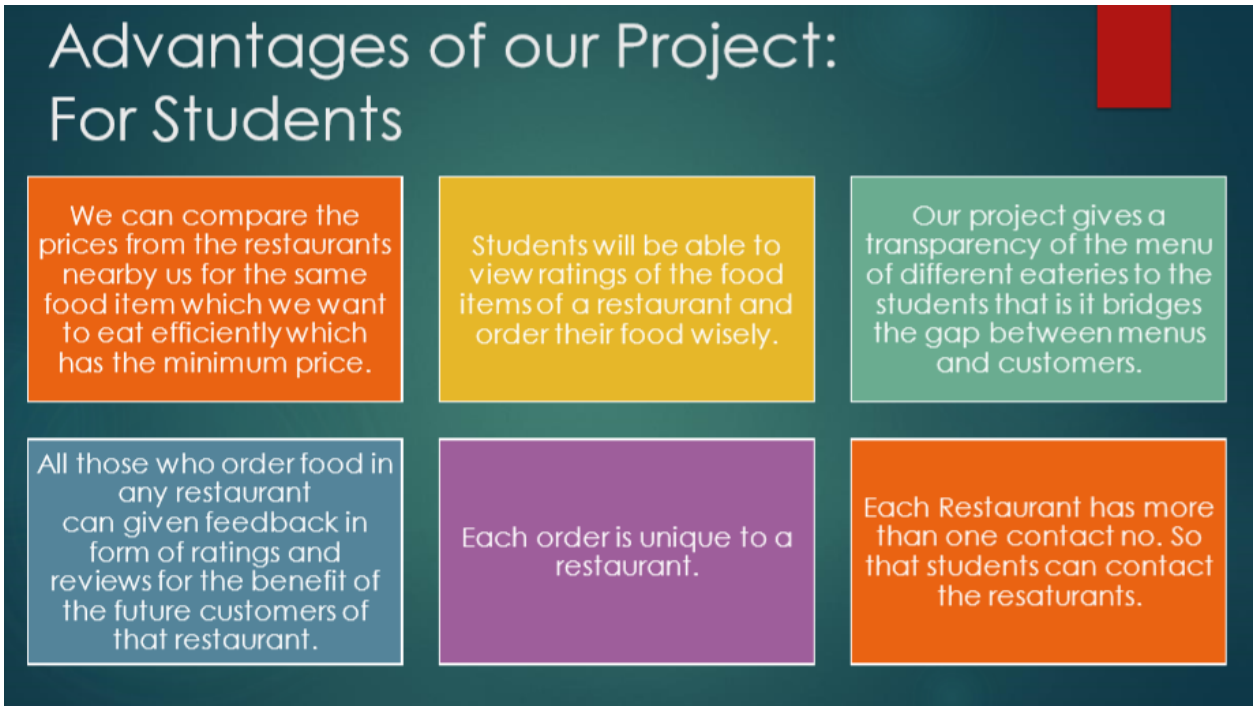
idea to integrate this information about restaurants on our campus which is very useful for all students, faculties, visitors and the shop owner themselves.

For Newcomers and Visitors:

For newcomers, visitors and all the students@ IITI, it will be a great boon which will help them decide. 1. What to eat?

2 . Where to eat?

3. When to eat?



For shop owners:

They can:

1)Edit the menu anytime:

a)add food items

b)delete items

c)modify prices

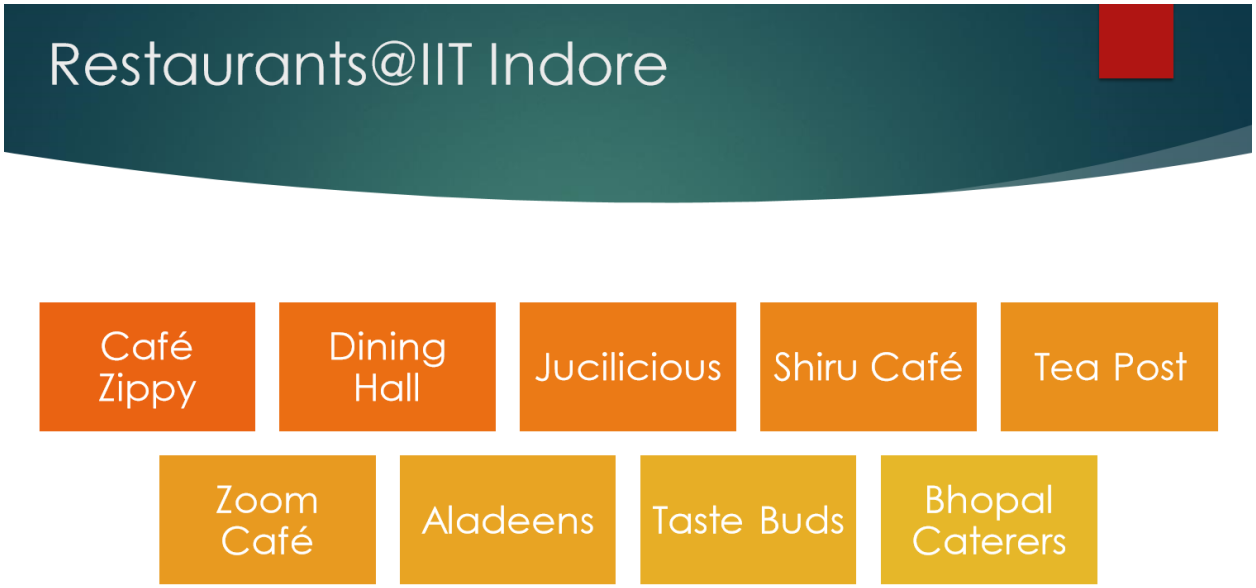
2)Change restaurant timing

a)they can switch their restaurant open/close through a manual process also in our software 3)Manage/maintain a proper information (type/job)about the workers, chef, waiter, etc(their salary, their name, hired date)

4)Manage the amount of food constituents available in their inventory. Here manage involves the amount present and constituents present. Moreover as soon as the quantity of any item reduces below a pre-set limit, it shows warning and gets added to the shopping list of Due Items.

- We will keep a managerial section in this project which keeps a track of all out clients (shopowners) where the upcoming shop owners can join the system through a login channel for which we maintain a login system.
- Customers will get a bill based on the food items ordered. Bill will be printed. They can order food from one restaurant only.

Restaurants:



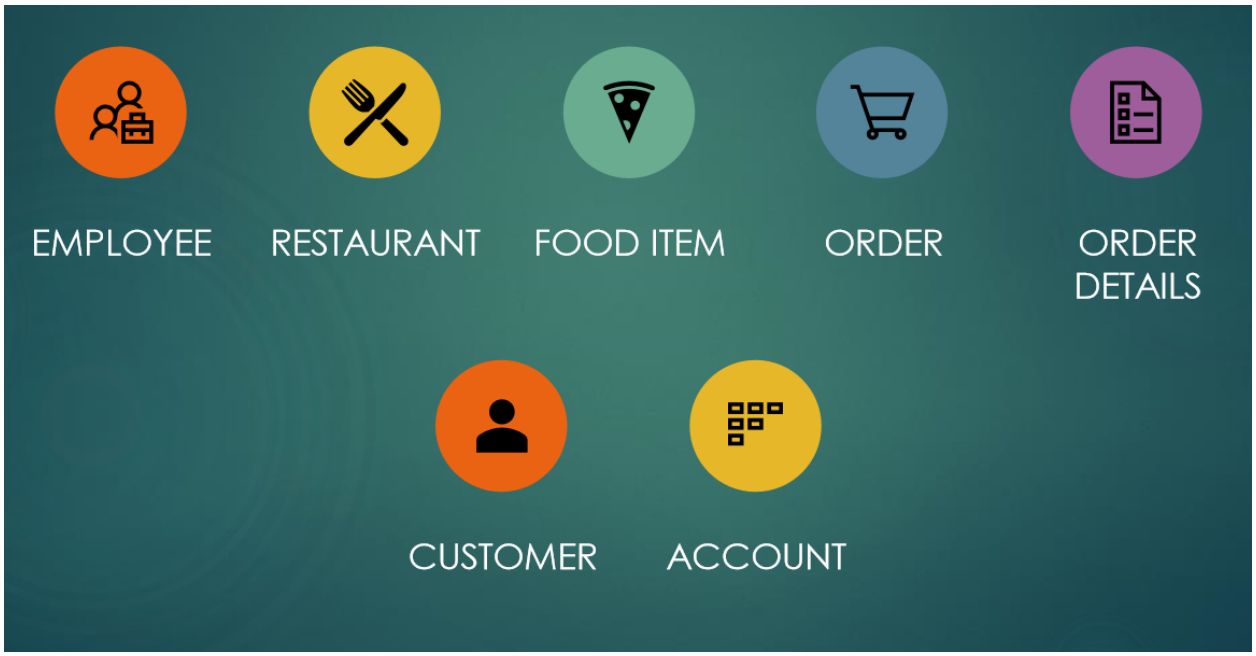
ER Analysis: Identifying Entity Sets :

1) Employee(entity set)
2) Restaurant(entity set)
3) Orders(entity set)
4) Order_Details(partial entity)
5) Customer(entity set)

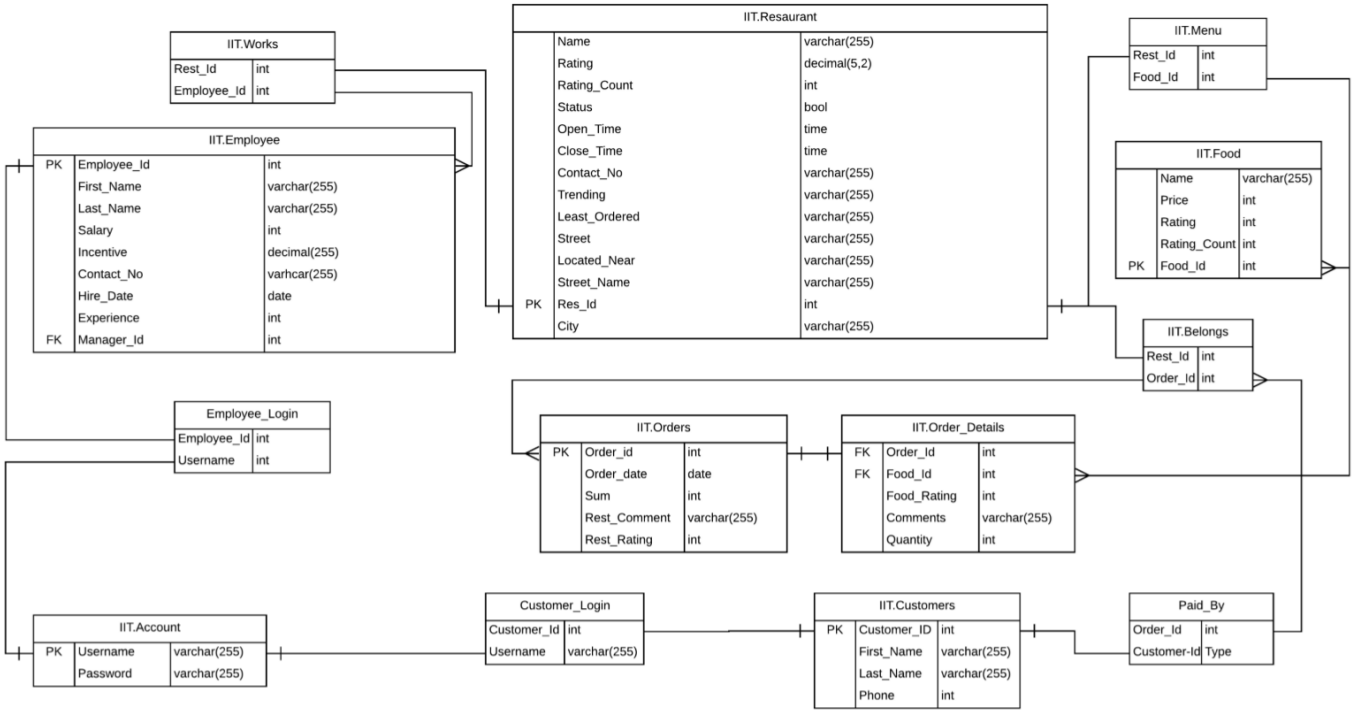
6) Food(entity set)
7) Account(entity set)

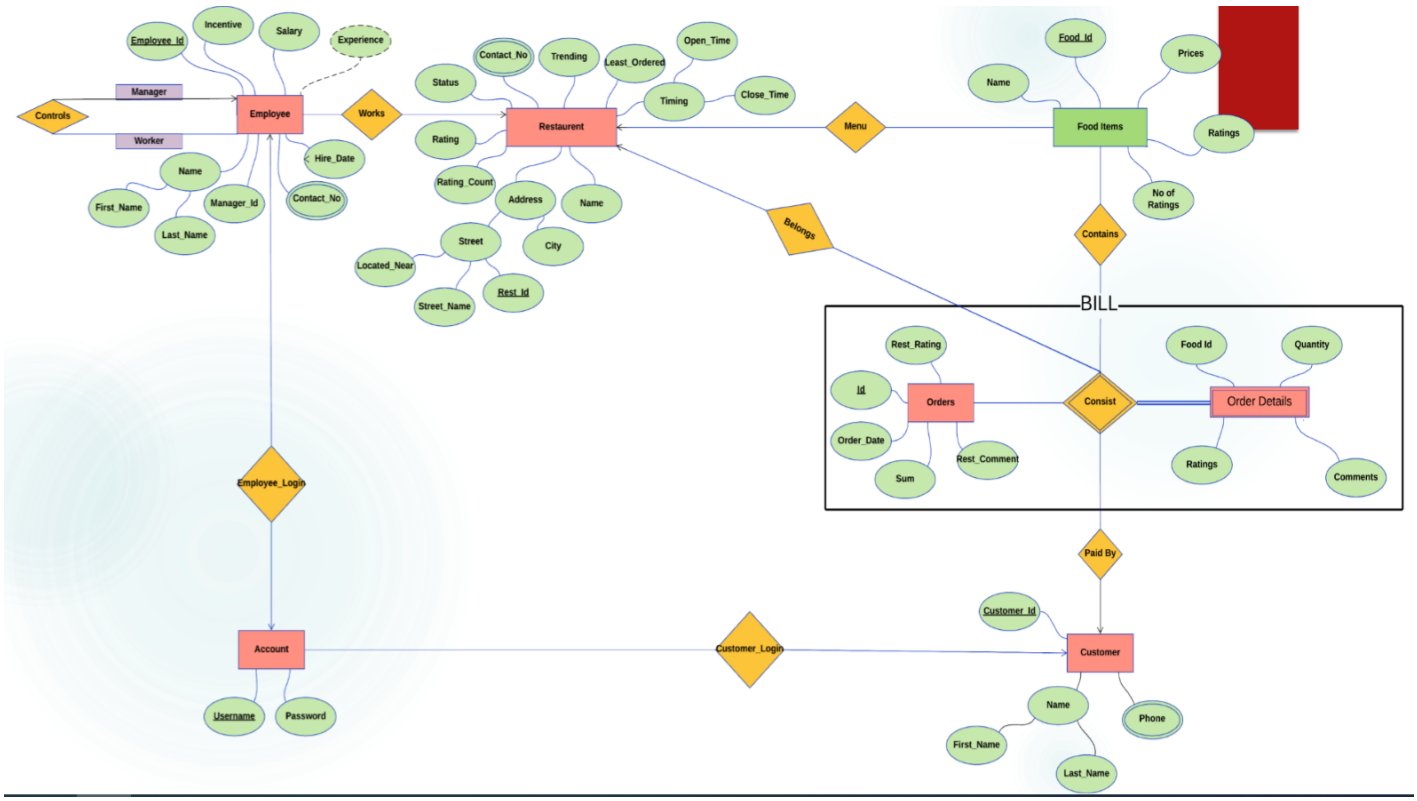
Relationship Sets:

1. Menu – Between Restaurant and Food Item (one to many)
2. Controls – Recursive Relation on Employee Entity Set. A Manager controls its Workers (one to many)
3. Works – Between Employee and Restaurant (many to one)
4. Belongs – Between Bill and Restaurant (many to one)
5. Contains – Between Bill and Food Items (many to many)
6. Consists – Between Order and Order Details (one to many)
7. Employee_Login – Between Manager and Account (one to one)
8. Customer_Login – Between Customer and Account (one to one)
9. Paid_By – Between Bill and Customer (many to one)



Entity and Relationship Sets:


IIT.Works
Rest_Id int
Employee_Id int
IIT.Resaurant
Name varchar(255)
Rating decimal(5,2)
Rating_Count int
Status bool
Open_Time time
Close_Time time
Contact_No varchar(255)
Trending varchar(255)
Least_Ordered varchar(255)
Street varchar(255)
Located_Near varchar(255)
Street_Name varchar(255)
PK Res_Id int
City varchar(255)
IIT.Menu
Rest_Id int
Food_Id int
IIT.Food
Name varchar(255)
Price int
Rating int
Rating_Count int
PK Food_Id int
IIT.Employee
PK Employee_Id int
First_Name varchar(255)
Last_Name varchar(255)
Salary int
Incentive decimal(255)
Contact_No varhcar(255)
Hire_Date date
Experience int
FK Manager_Id int
IIT.Belongs
Rest_Id int
Order_Id int
Employee_Login
Employee_Id int
Username int
IIT.Orders
PK Order_id int
Order_date date
Sum int
Rest_Comment varchar(255)
Rest_Rating int
IIT.Order_Details
FK Order_Id int
FK Food_Id int
Food_Rating int
Comments varchar(255)
Quantity int
IIT.Account
PK Username varchar(255)
Password varchar(255)
Customer_Login
Customer_Id int
Username varchar(255)
IIT.Customers
PK Customer_ID int
First_Name varchar(255)
Last_Name varchar(255)
Phone int
Paid_By
Order_Id int
Customer-Id Type


Employee_Id
Incentive
Salary
Experience
Manager
Controls
Employee
Works
Worker
Hire_Date
Name
First_Name
Manager_Id
Contact_No
Last_Name
Contact_No
Trending
Least_Ordered
Open_Time
Status
Timing
Close_Time
Rating
Restaurent
Rating_Count
Address
Name
Street
City
Located_Near
Rest_Id
Street_Name
Menu
Food_Id
Prices
Name
Food Items
Ratings
No of Ratings
Belongs
Contains
BILL
Rest_Rating
Id
Orders
Consist
Food Id
Quantity
Order Details
Order_Date
Sum
Rest_Comment
Ratings
Comments
Employee_Login
Paid By
Customer_Id
Account
Customer_Login
Customer
Name
Phone
First_Name
Last_Name
Username
Password

Transformation of ER diagrams into set of Tables:

```
create database rs;
```

1) **Employee**

```
CREATE TABLE employee (
        employee_id INTEGER PRIMARY KEY AUTOINCREMENT,
        first_name VARCHAR (255),
        last_name VARCHAR (255),
        salary INTEGER,
        incentive DECIMAL,
        contact_no VARCHAR,
        hire_date DATE,
        experience INTEGER,
        manager_id INTEGER
);
```

2) **Restaurant**

```
        CREATE TABLE restaurant (
                name VARCHAR (255),
                rating DECIMAL,
                rating_count INTEGER,
                status BOOLEAN,
                open_time TIME,
                close_open TIME,
                contact_no VARCHAR,
                trending VARCHAR,
                least_ordered VARCHAR,
                located_near VARCHAR,
                street_name VARCHAR,
                rest_id INTEGER PRIMARY KEY AUTOINCREMENT,
                city VARCHAR
        );
```

3) **Food**

```
        CREATE TABLE food (
                name VARCHAR (255),
                price INTEGER,
```

```
        rating INTEGER,
        rating_count INTEGER,
        food_id INTEGER PRIMARY KEY AUTOINCREMENT
);
```

4) Orders

```
CREATE TABLE Orders (
        order_id INTEGER PRIMARY KEY AUTOINCREMENT,
        order_date DATE DEFAULT (date('now') ),
        sum INTEGER,
        rest_comment VARCHAR,
        rest_rating INTEGER,
        order_time TIME DEFAULT (time('now') )
);
```

5) Customers

```
 CREATE TABLE customers (
        username VARCHAR PRIMARY KEY,
        first_name VARCHAR,
        last_name VARCHAR,
        phone INTEGER
);
```

6) Menu

```
CREATE TABLE Menu (
        Rest_id INTEGER,
        Food_id INTEGER PRIMARY KEY AUTOINCREMENT
);
```

7) Works

```
CREATE TABLE works (
        rest_id INTEGER,
        employee_id INTEGER PRIMARY KEY AUTOINCREMENT
);
```

8) Order_Details

```
CREATE TABLE Order_Details (
        order_id INTEGER REFERENCES Orders (order_id),
        food_id INTEGER REFERENCES food (food_id),
        food_rating INTEGER,
        comments VARCHAR (255),
        quantity INTEGER,
        amount INTEGER
```

```
);
```

9) Account

```
CREATE TABLE account (
        username VARCHAR (255),
        password VARCHAR (255)
);
```

10) Belongs

```
CREATE TABLE belongs (
        rest_id INTEGER,
        order_id INTEGER PRIMARY KEY AUTOINCREMENT
);
```

11) Employee_login

```
CREATE TABLE employee_login (
        employee_id INTEGER,
        username VARCHAR (255)
);
```

12) Paid_By

```
CREATE TABLE paid_by (
        order_id INTEGER PRIMARY KEY AUTOINCREMENT,
        username VARCHAR (255)
);
```

Triggers

```
CREATE TRIGGER cal
AFTER INSERT
ON Order_Details
FOR EACH ROW
BEGIN
UPDATE order_details
SET amount = new.quantity * (
SELECT price
FROM food
WHERE food.food_id = new.food_id
)
WHERE food_id = new.food_id AND
order_id = new.order_id;
END;
```

;

SQL Queries (as implemented in code):

1. HomePage:
   select * from restaurant;
2. MenuPage:
   - select * from restaurant;
   - select * from menu inner join food on menu.food_id=food.food_id where menu.rest_id=rest_id;
   - SELECT name FROM restaurant WHERE rest_id =rest_id;
3. select * from restaurant where name=rest_name;
4. select * from food where food_id in ("+food_id+");
5. insert into orders(sum) values(sum);
6. SELECT * FROM orders ORDER BY order_id DESC LIMIT 1;
7. Insert into belongs(rest_id,order_id) values(restaurant.rest_id,order_detail.order_id);
8. insert into paid_by(order_id,username) values(order_detail.order_id,req.session.username);
9. for(let index=0;index<food_item.length; index+ +)

{ db.run("insert into order_details(order_id,food_id,quantity) values(?,?,?)",[order_detail.order_id,food_item[index].food_id,m.get(food_item[index].food_id)],function(err){}); }

10. select * from customers where username = req.session.username;
11. select * from paid_by inner join orders on paid_by.order_id=orders.order_id inner join belongs on orders.order_id=belongs.order_id inner join restaurant on restaurant.rest_id=belongs.rest_id where username = req.session.username;
12. select * from order_details inner join food on food.food_id=order_details.food_id where order_details.order_id=order_id
13. SELECT * FROM account WHERE username = username AND password = password;
14. insert into account values(username,password)
15. insert into customers values(?,?,?,?)",[username,firstname,lastname,phone]

16. select * from customers inner join account on customers.username=account.username where customers.username = ?",[req.session.username]
17. update customers set first_name=?,last_name=?,phone=? where username=?",[firstname,lastname,phone,username]
18. "update account set password=? where username=?",[password,username]

Implementation Of the Above Database

Web Pages

- ✔ ***Homepage***

RESTAURANTS

Hotes in IITI

ZIPPY

TEA POST

TASTE BUDS

ABOUT US

We strive to feed your needs

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

CONTACT

We are eager to hear from you

Infront of POD Building
IIT Indore, Simrol, 453552

Khazana@IITI.ac.in

+911123456789

Your Name

Your Email

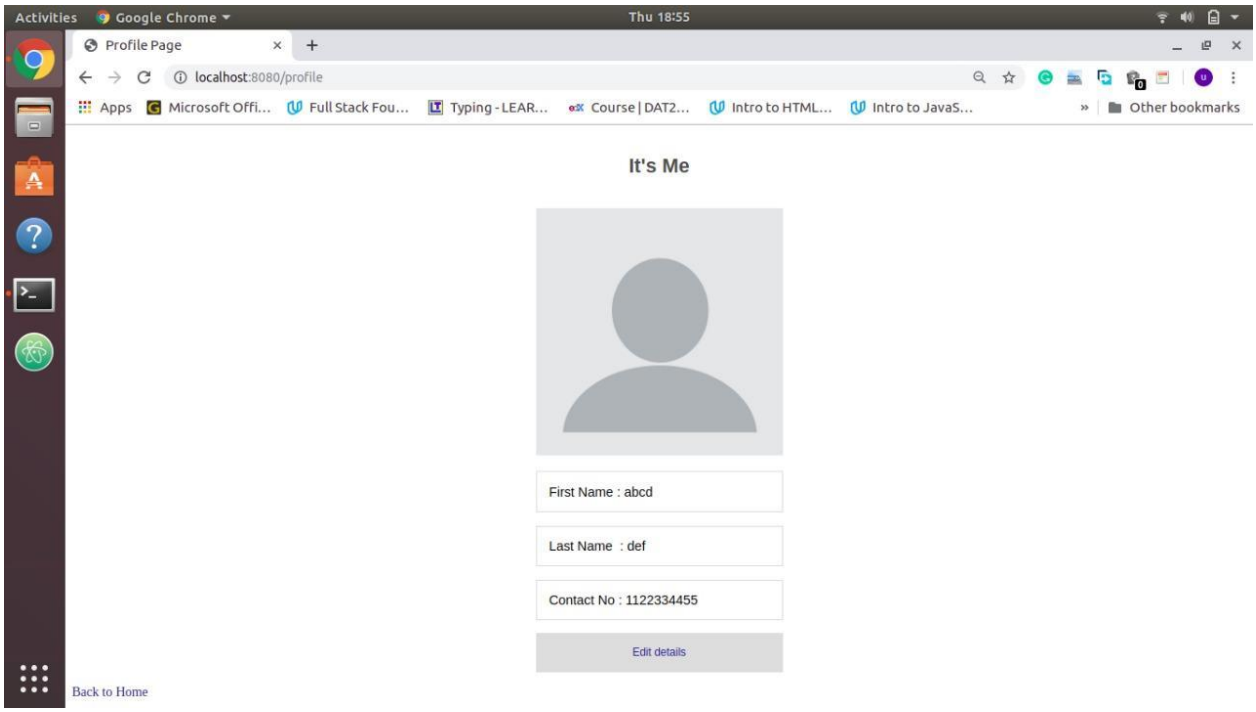
Subject

Message

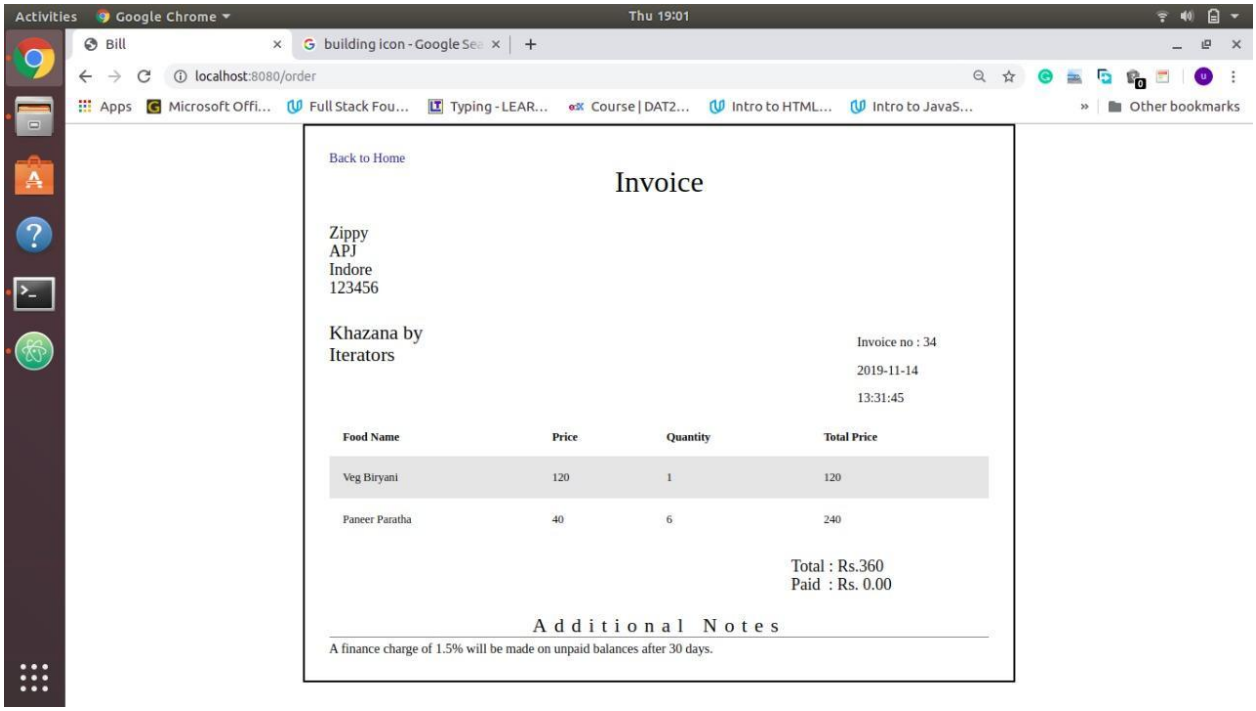
Send Message

Made using BootStrapMade

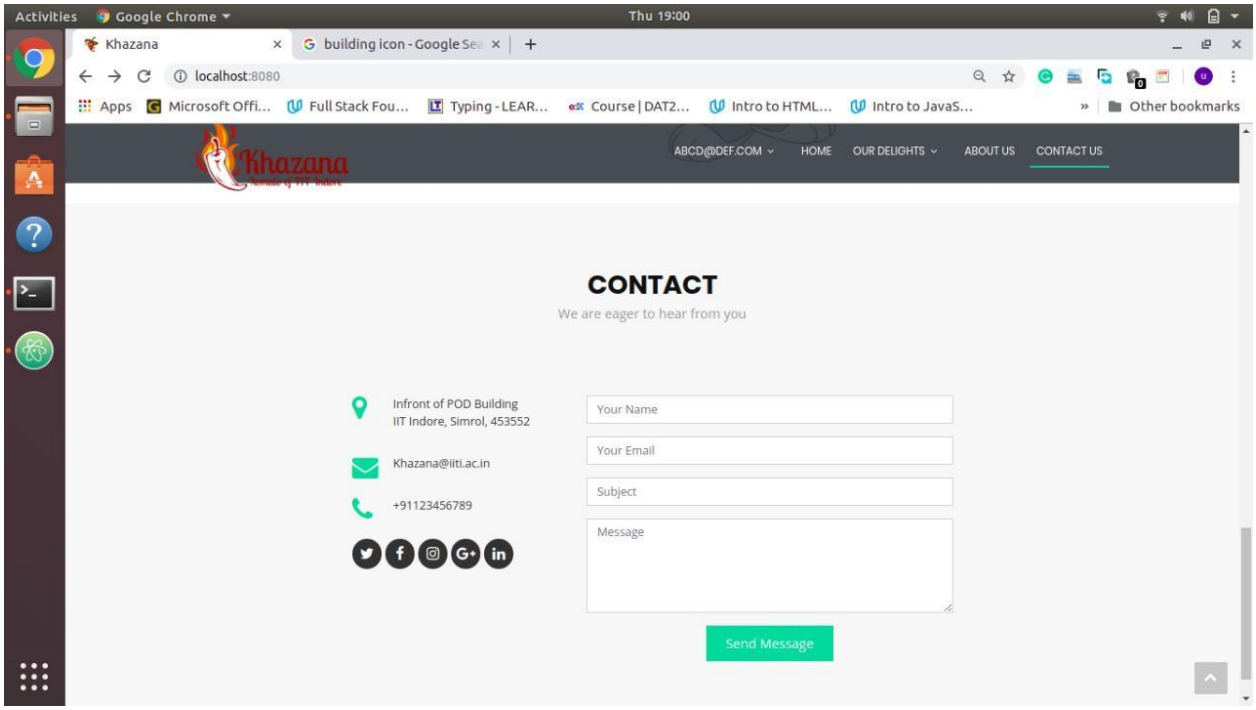
✔ ***Profile Page***



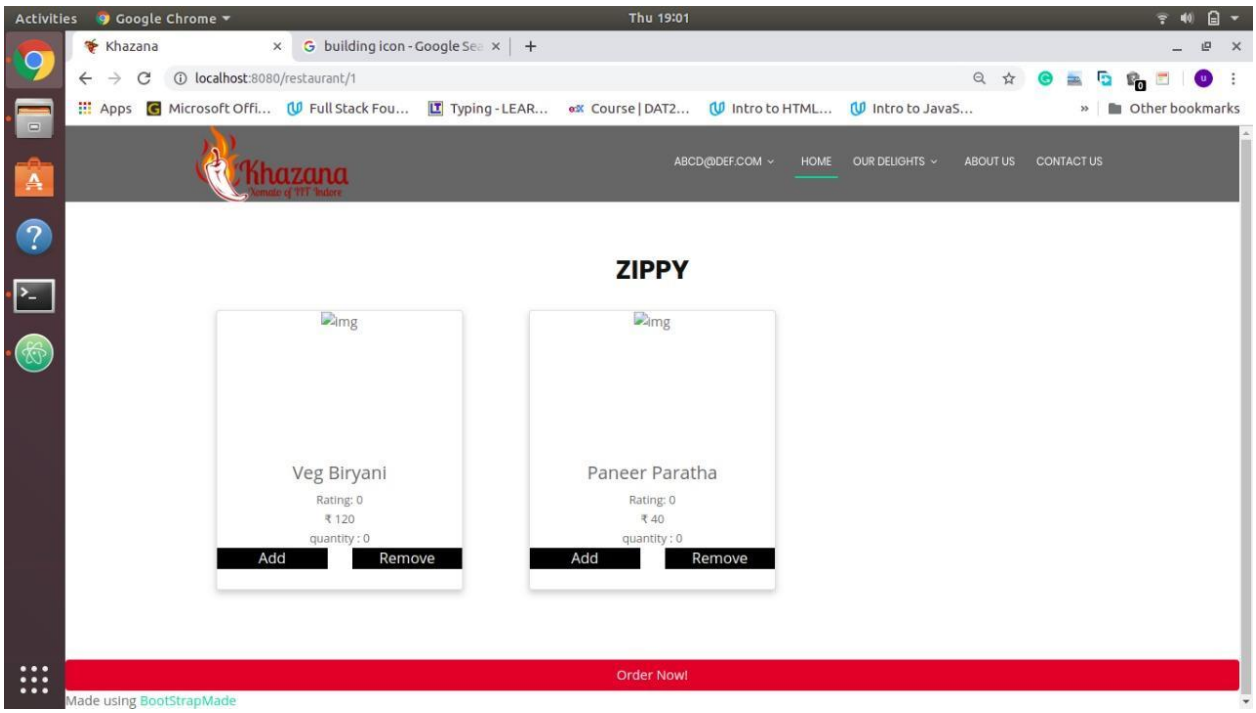
✔ ***Invoice:***



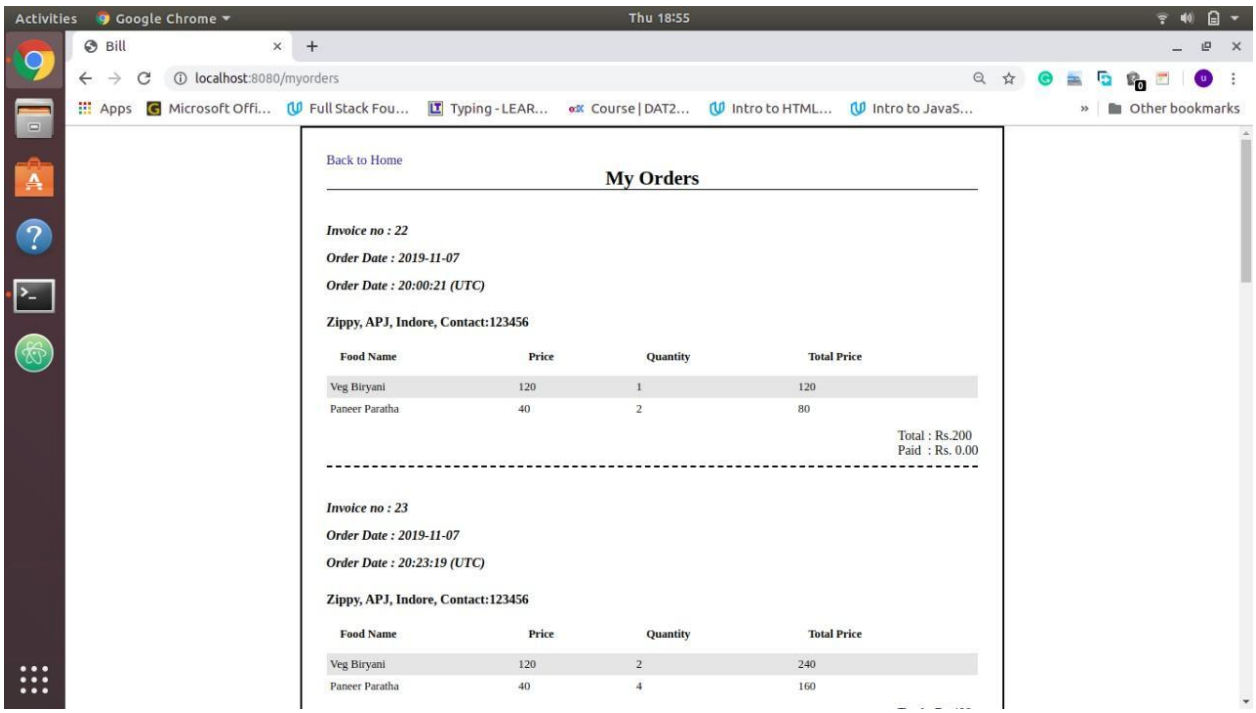
✔ ***Contact:***


Activities
Google Chrome
Thu 19:00
Khazana
building icon - Google Sea
localhost:8080
Apps
Microsoft Offi...
Full Stack Fou...
Typing - LEAR...
Course | DAT2...
Intro to HTML...
Intro to JavaS...
Other bookmarks
Khazana
ABCD@DEF.COM
HOME
OUR DELIGHTS
ABOUT US
CONTACT US
CONTACT
We are eager to hear from you
Infront of POD Building
IIT Indore, Simrol, 453552
Khazana@iiti.ac.in
+91123456789
Your Name
Your Email
Subject
Message
Send Message

✔ ***Menu:***


Activities
Google Chrome
Thu 19:01
Khazana
building icon - Google Sea
localhost:8080/restaurant/1
Apps
Microsoft Offi...
Full Stack Fou...
Typing - LEAR...
Course | DAT2...
Intro to HTML...
Intro to JavaS...
Other bookmarks
Khazana
ABCD@DEF.COM
HOME
OUR DELIGHTS
ABOUT US
CONTACT US
ZIPPY
img
Veg Biryani
Rating: 0
₹ 120
quantity : 0
Add
Remove
img
Paneer Paratha
Rating: 0
₹ 40
quantity : 0
Add
Remove
Order Now!
Made using BootStrapMade

✔ ***My Orders Page:***



--Thank You--