

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.

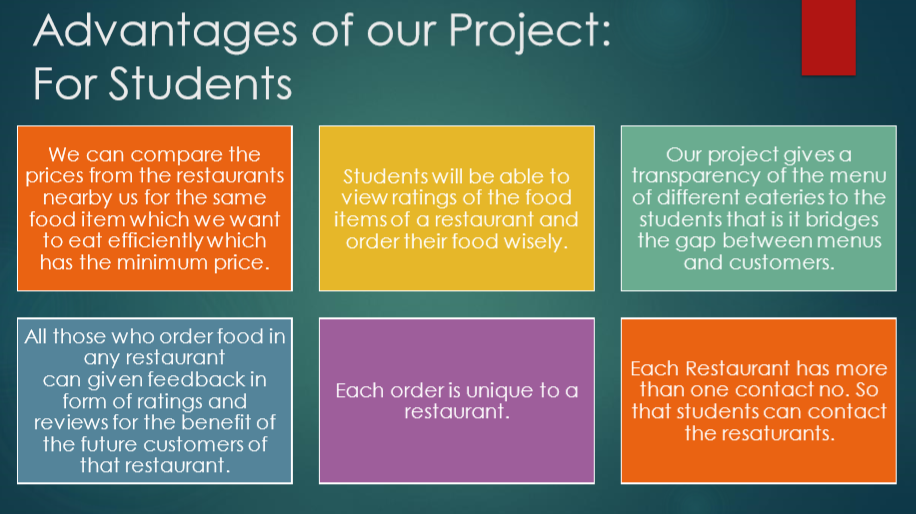
A pocket-friendly person wants that he gets his desired food item in his budget which has nice ratings and reviews and is currently a trending item of that restaurant. There are about 10 restaurants which generally makes the customer confused in choosing a particular place to eat. Hence, we came up with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

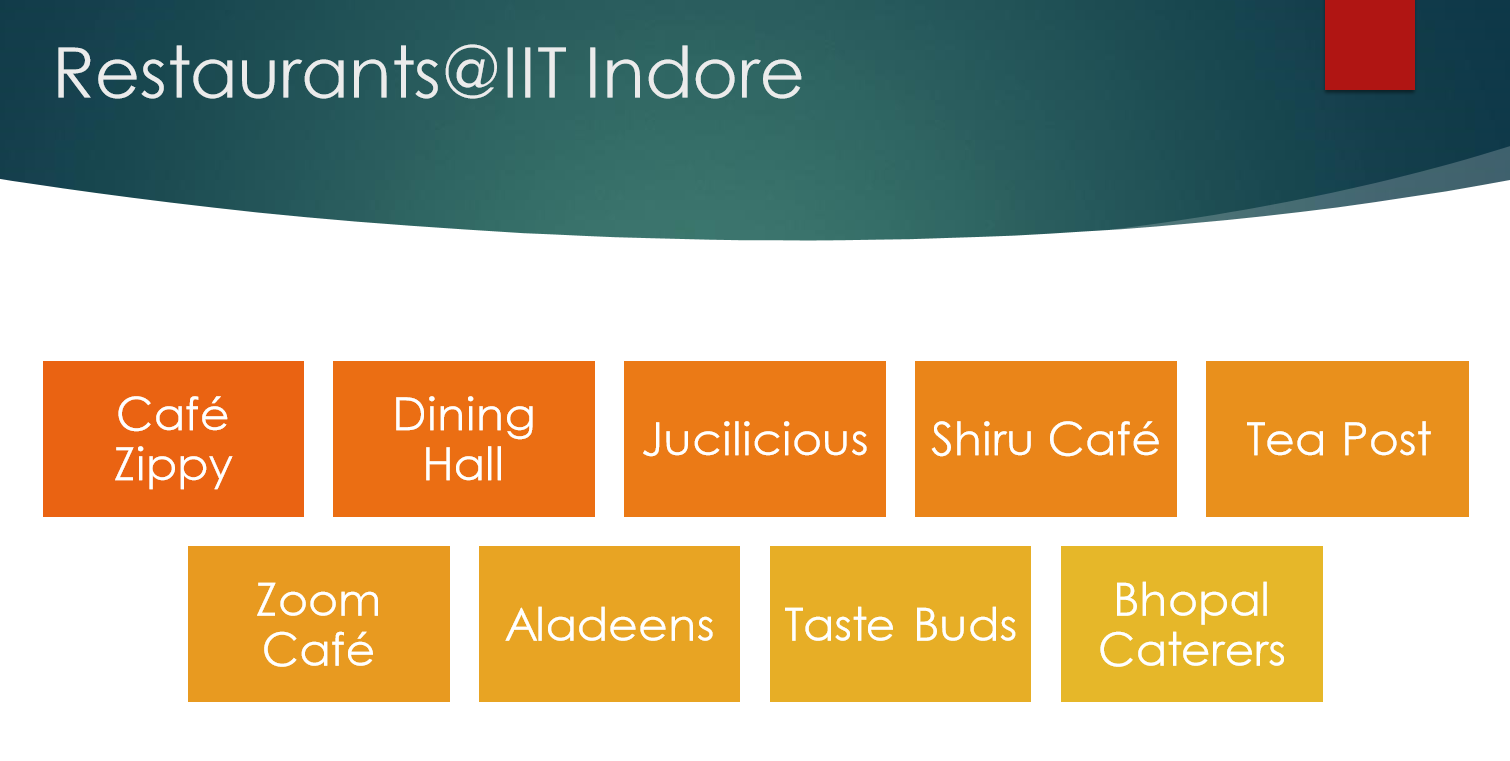
2)Change restaurant timing

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.

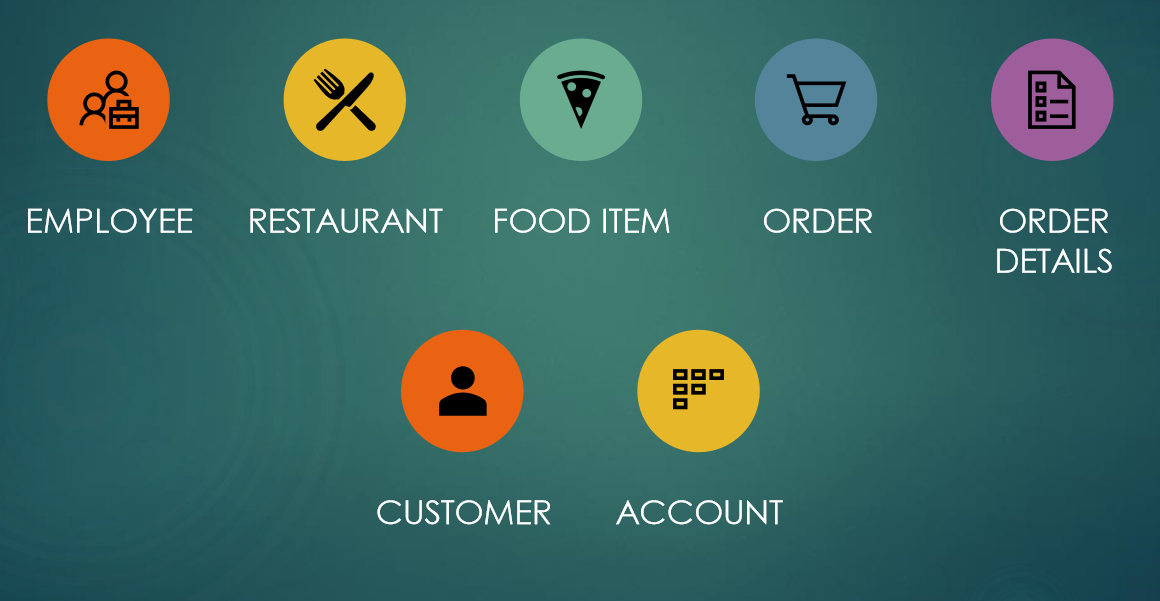
**Restaurants:**

** ER Analysis: Identifying Entity Sets :**

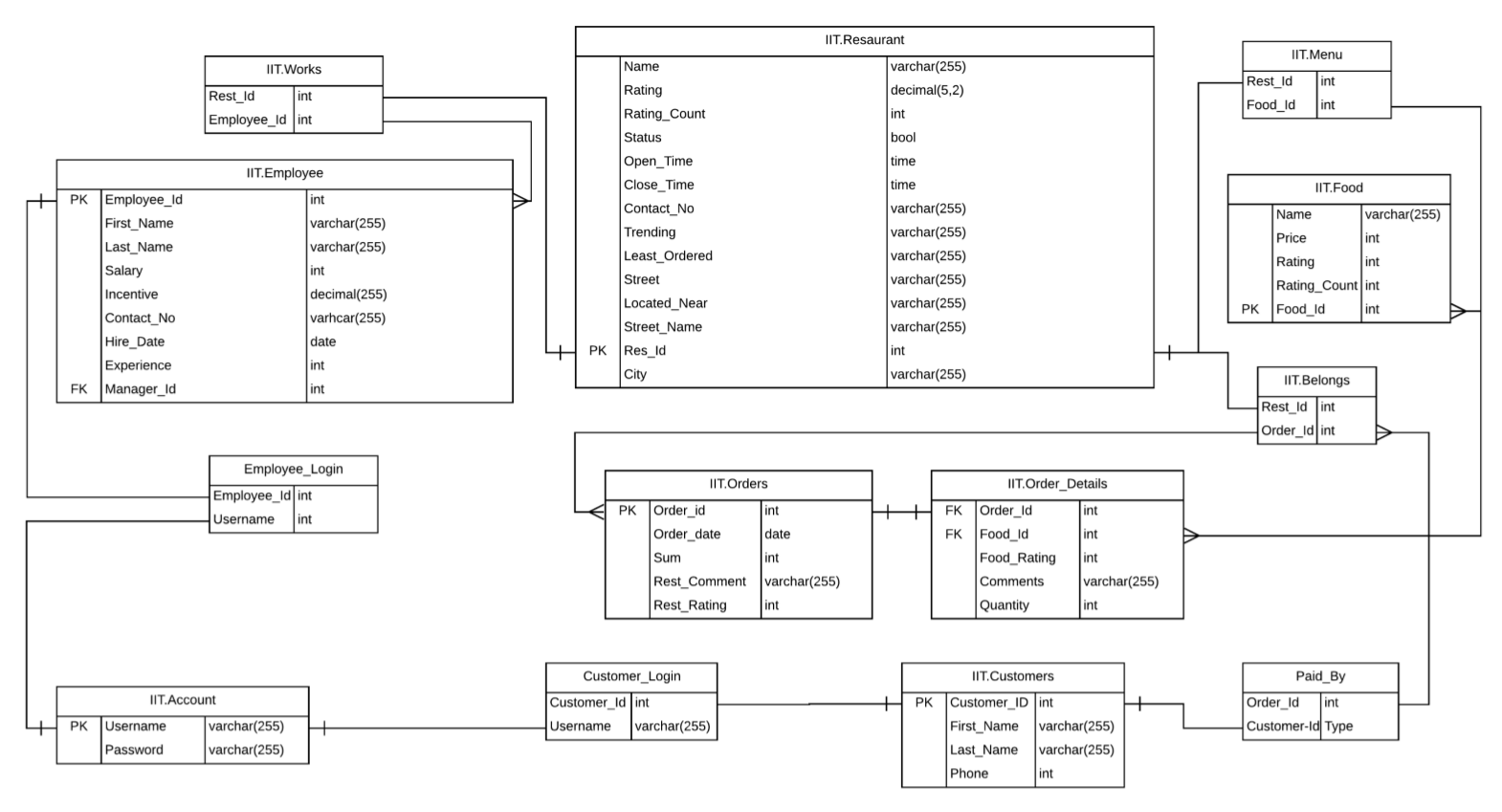
1. Employee(entity set)
2. Restaurant(entity set)
3. Orders(entity set)
4. Order\_Details(partial entity)
5. Customer(entity set)
6. Food(entity set)
7. Account(entity set)

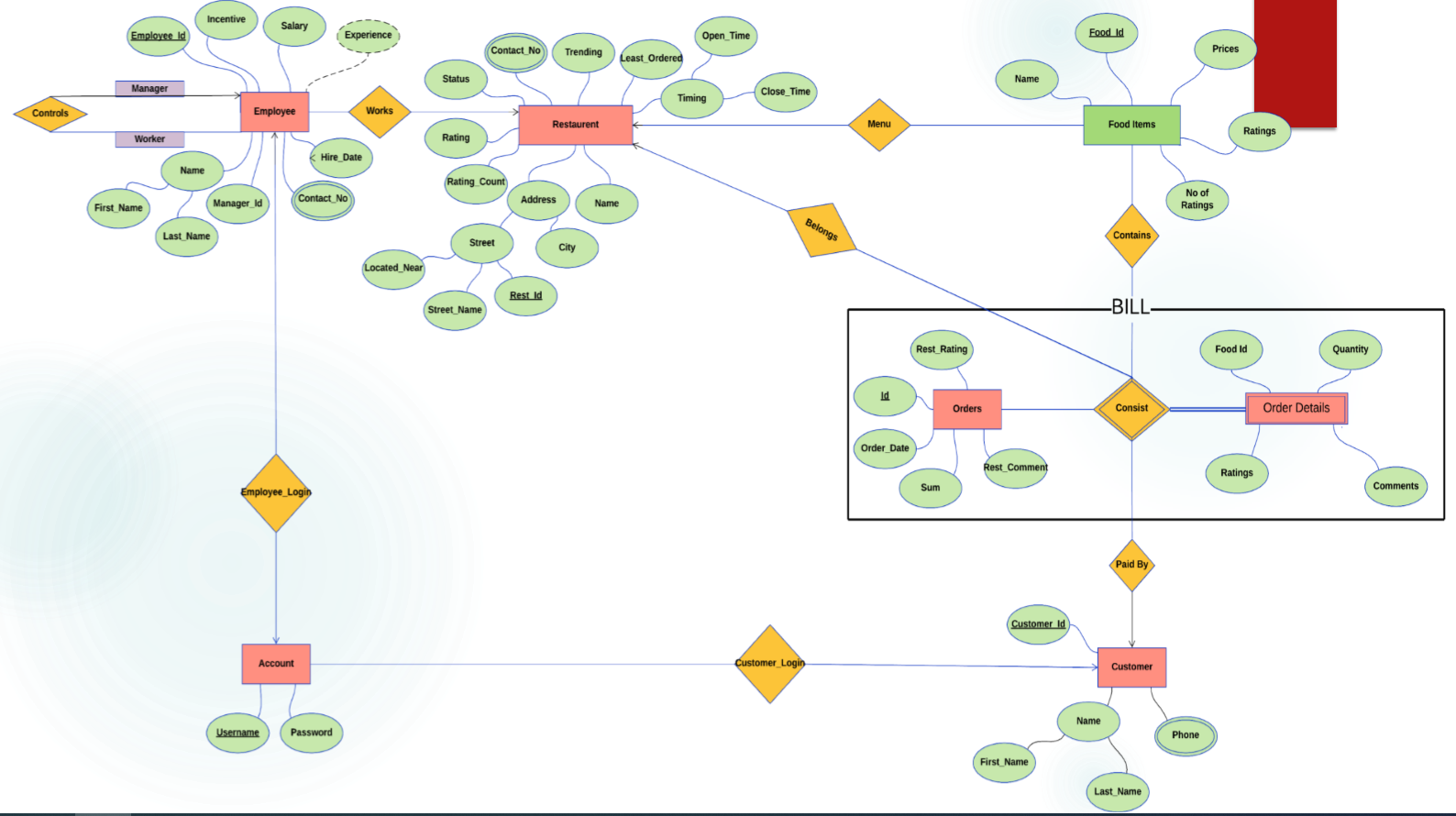
**Relationship Sets:**

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:**





**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

);

1. **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

);

1. **Food**

CREATE TABLE food (

name VARCHAR (255),

price INTEGER,

rating INTEGER,

rating\_count INTEGER,

food\_id INTEGER PRIMARY KEY AUTOINCREMENT

);

1. **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') )

);

1. **Customers**

CREATE TABLE customers (

username VARCHAR PRIMARY KEY,

first\_name VARCHAR,

last\_name VARCHAR,

phone INTEGER

);

1. **Menu**

CREATE TABLE Menu (

Rest\_id INTEGER,

Food\_id INTEGER PRIMARY KEY AUTOINCREMENT

);

1. **Works**

CREATE TABLE works (

rest\_id INTEGER,

employee\_id INTEGER PRIMARY KEY AUTOINCREMENT

);

1. **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

);

1. **Account**

CREATE TABLE account (

username VARCHAR (255),

password VARCHAR (255)

);

1. **Belongs**

CREATE TABLE belongs (

rest\_id INTEGER,

order\_id INTEGER PRIMARY KEY AUTOINCREMENT

);

1. **Employee\_login**

CREATE TABLE employee\_login (

employee\_id INTEGER,

username VARCHAR (255)

);

1. **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;

1. 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;

1. select \* from restaurant where name=rest\_name;
2. select \* from food where food\_id in ("+food\_id+");
3. insert into orders(sum) values(sum);
4. SELECT \* FROM orders ORDER BY order\_id DESC LIMIT 1;
5. Insert into belongs(rest\_id,order\_id) values(restaurant.rest\_id,order\_detail.order\_id);
6. insert into paid\_by(order\_id,username) values(order\_detail.order\_id,req.session.username);
7. for(let index=0;index<food\_item.length; index+ +)

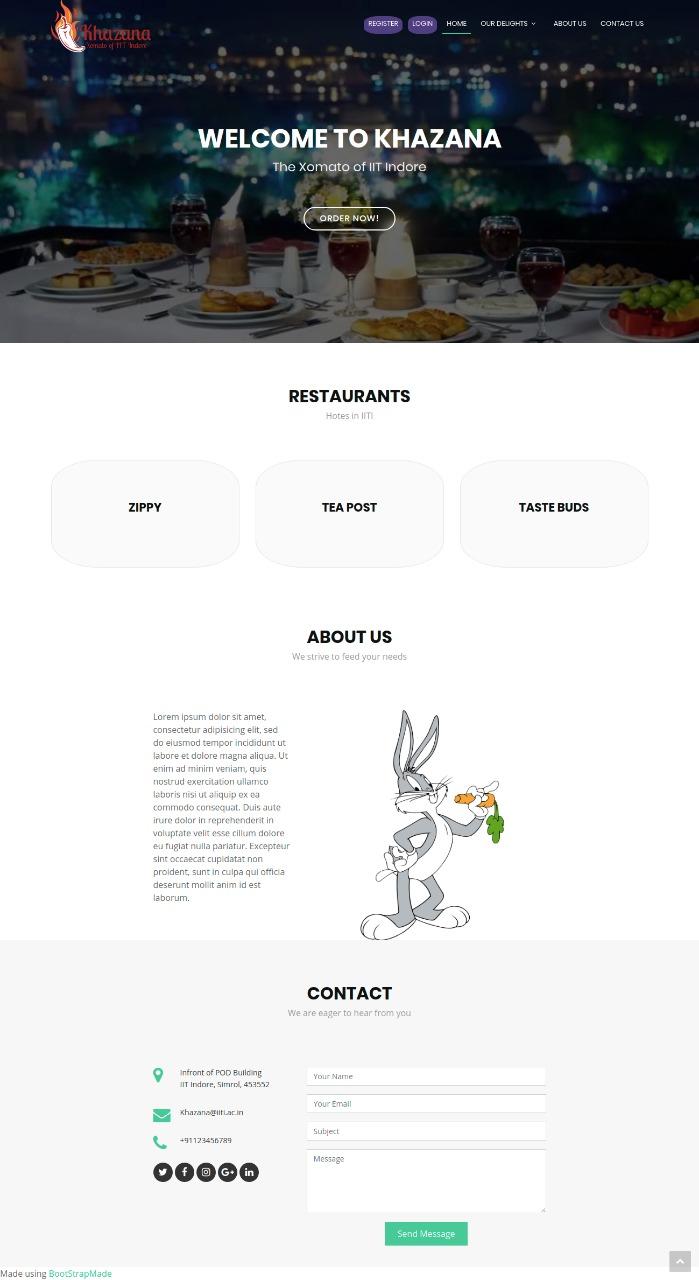
{ db.run("insert into order\_details(order\_id,food\_id,quantity) values(?,?,?)",[order\_detail.order\_id,food\_item [index].food\_id,m.get(food\_item[index].food\_id)],function(err){}); }

1. select \* from customers where username = req.session.username;
2. 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;
3. select \* from order\_details inner join food on food.food\_id=order\_details.food\_id where order\_details.order\_id=order\_id
4. SELECT \* FROM account WHERE username = username AND password = password;
5. insert into account values(username,password)
6. insert into customers values(?,?,?,?)",[username,firstname,lastname,phone]
7. select \* from customers inner join account on customers.username=account.username where customers.username = ?",[req.session.username]
8. update customers set first\_name=?,last\_name=?,phone=? where username=?",[firstname,lastname,phone,username]
9. "update account set password=? where username=?",[password,username]

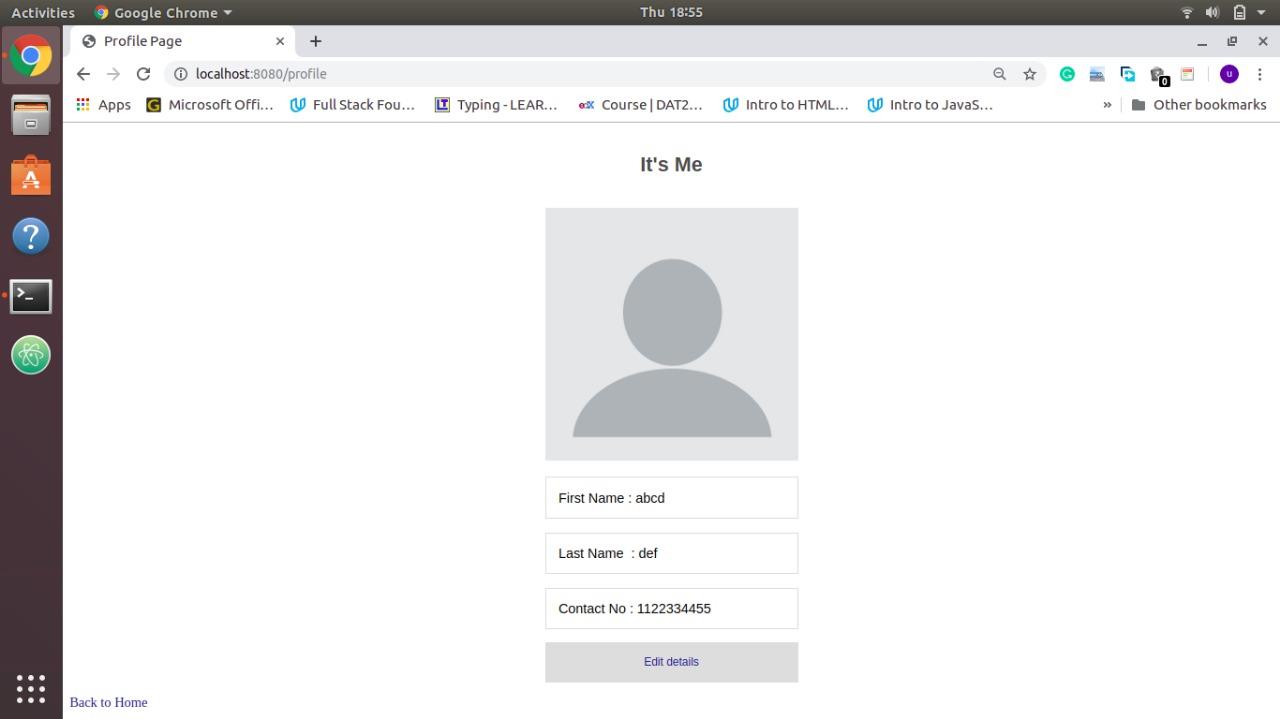
**Implementation Of the Above Database**

***Web Pages***

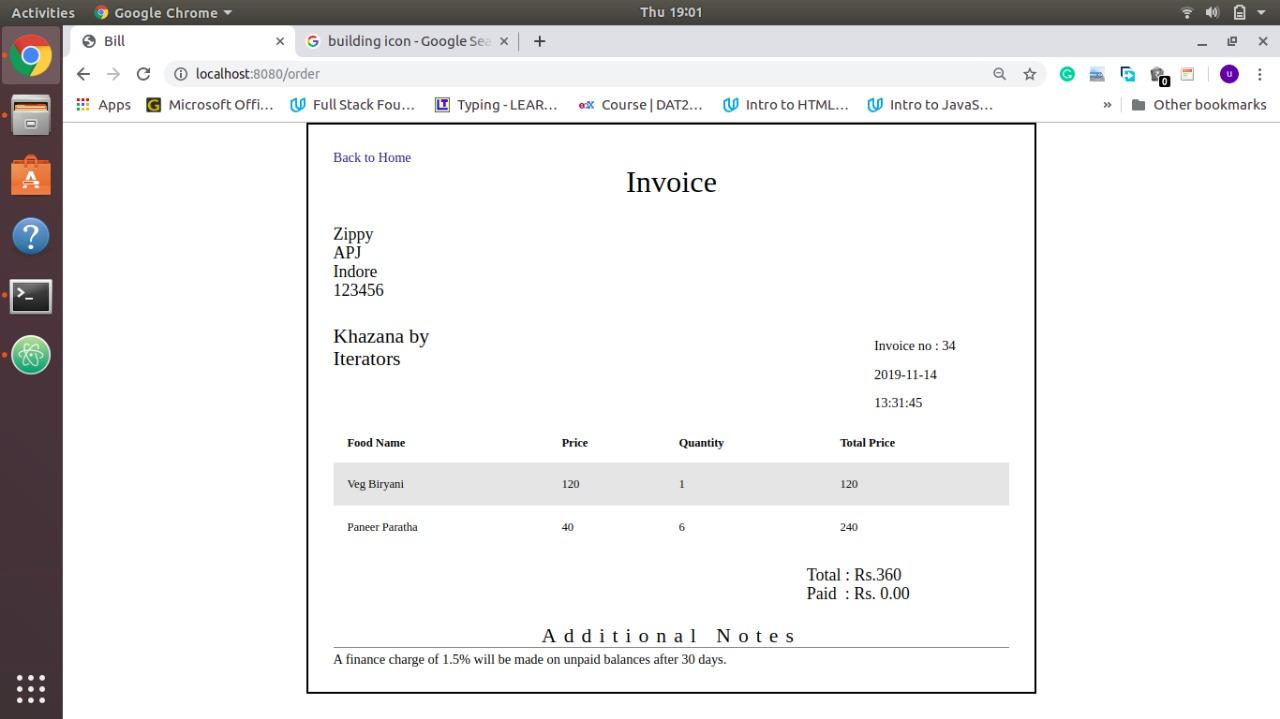
* ***Homepage***

******

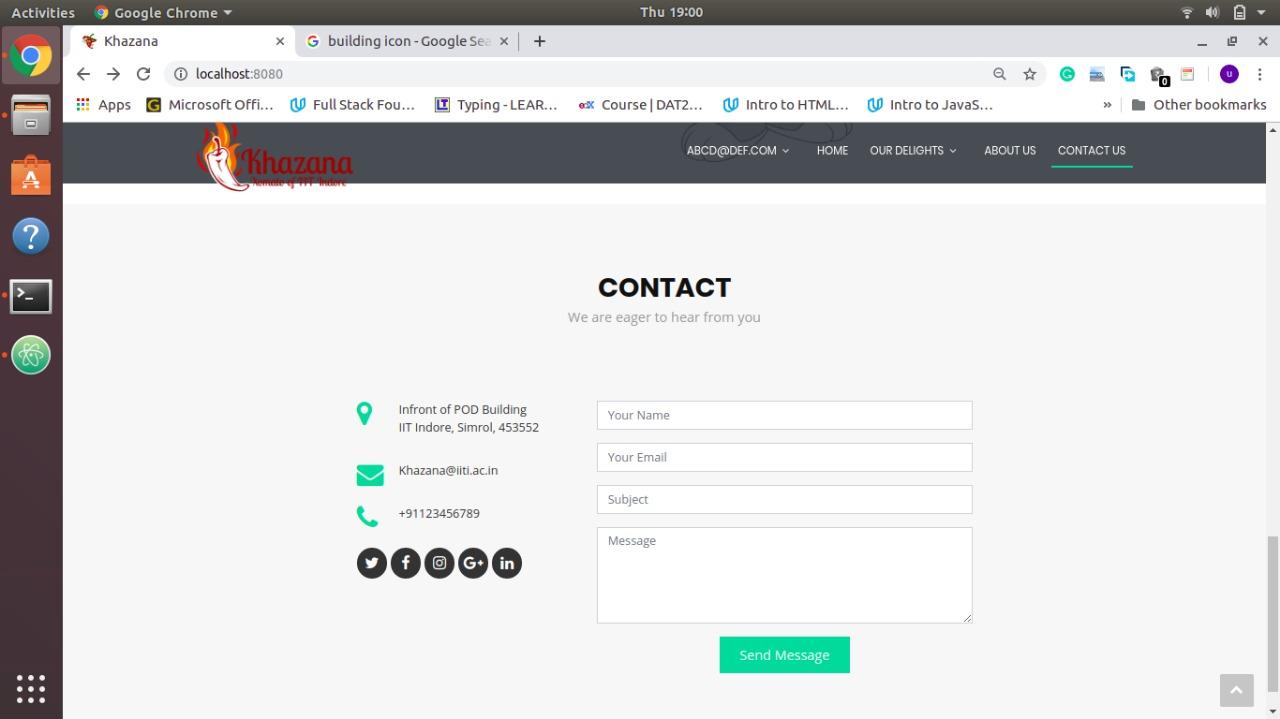
* ***Profile Page***

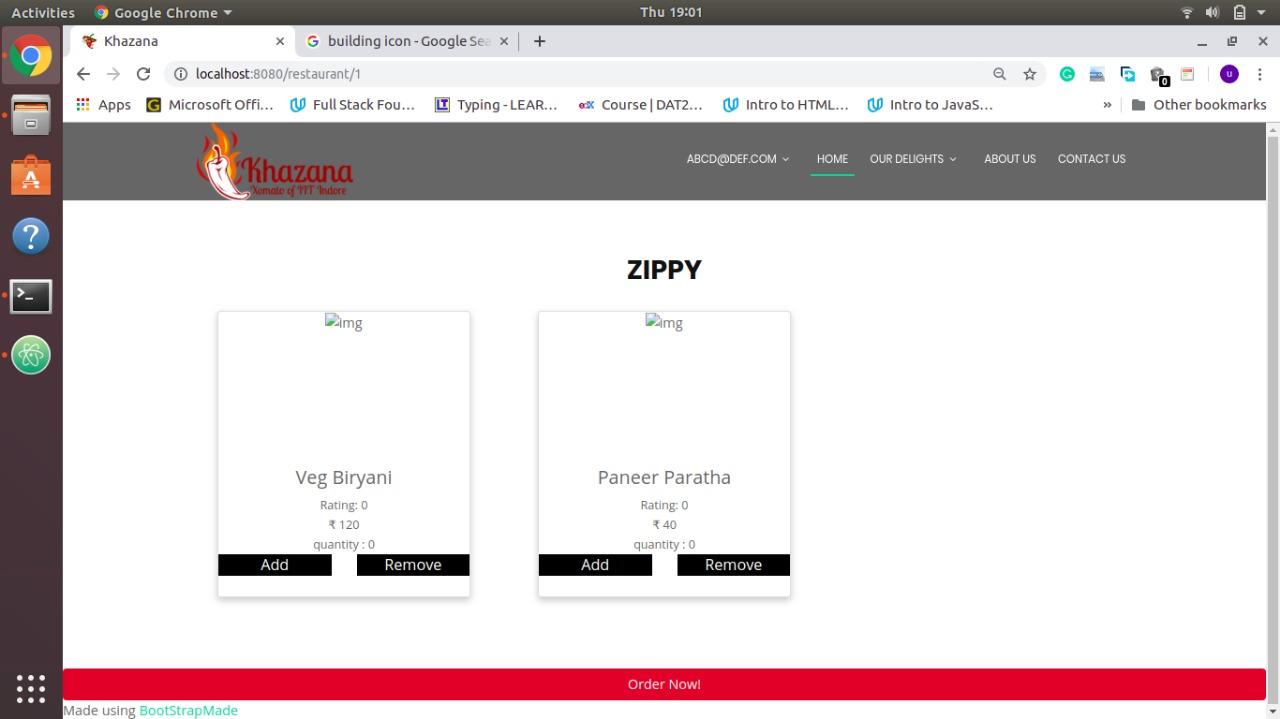
******

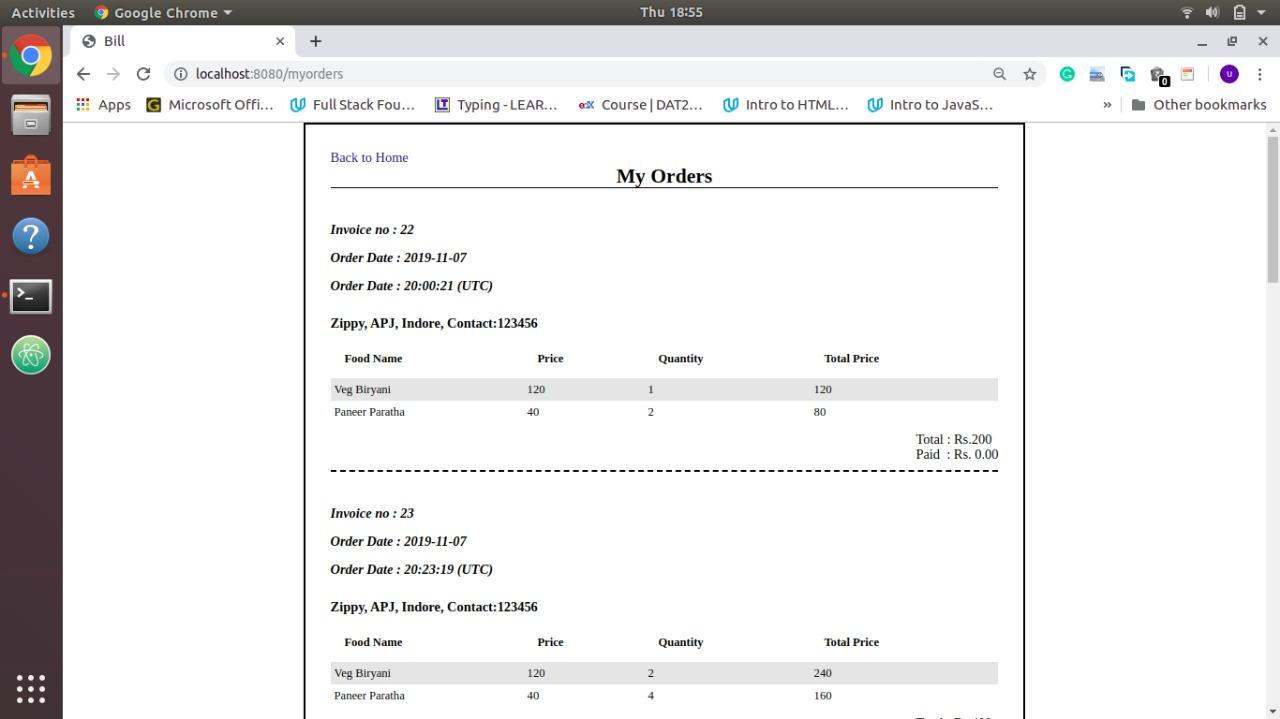
* ***Invoice:***



* ***Contact:***

******

* ***Menu:***
* ***My Orders Page:***



--Thank You--