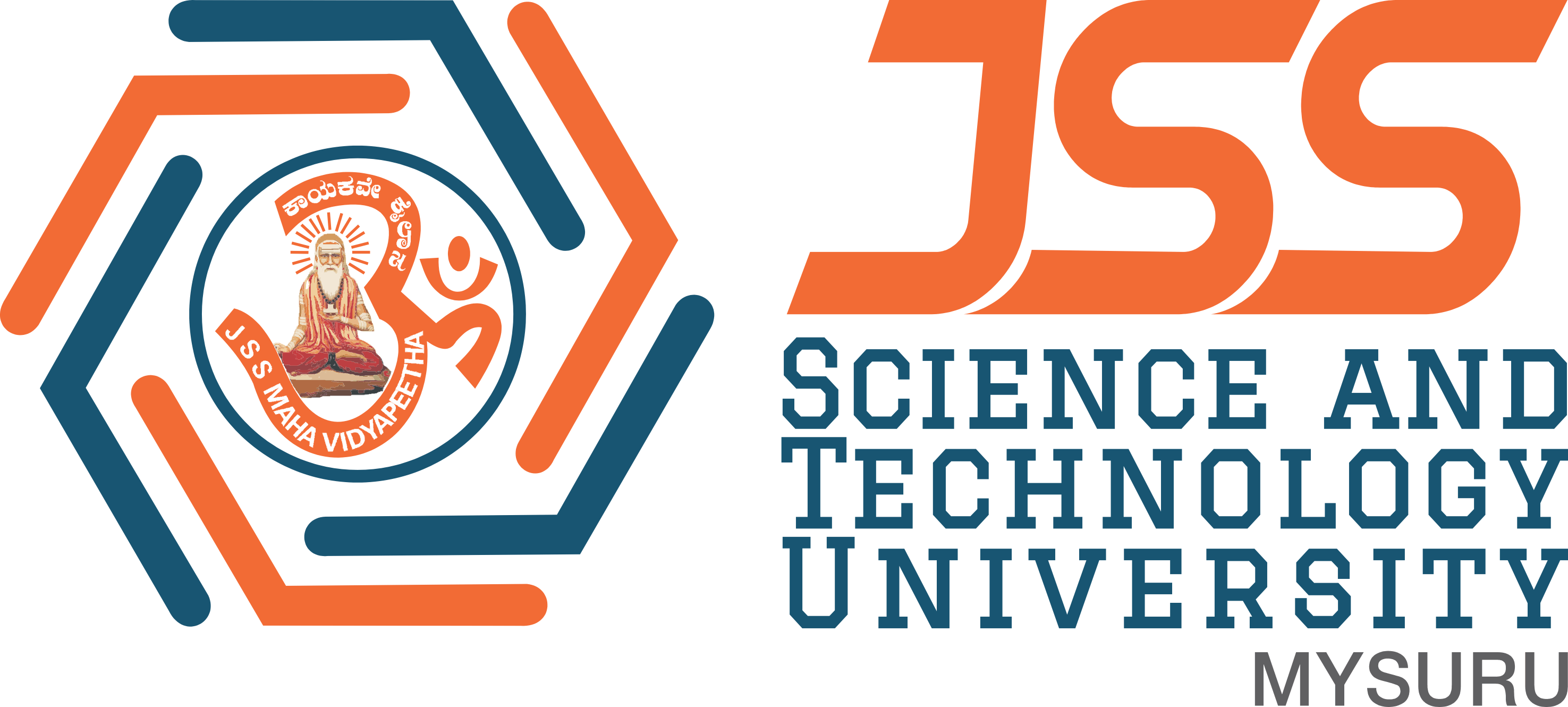
**JSS MAHAVIDYAPEETHA**

**JSS SCIENCE AND TECHNOLOGY UNIVERSITY**

JSS Technical Institutions Campus, Mysuru – 570006



**“ONLINE FOOD ORDER SYSTEM”**

Mini project report submitted in partial fulfilment of curriculum prescribed for the Database Systems (CS510) course for the award of the degree of

#### BACHELOR OF ENGINEERING

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

*by*

|  |  |
| --- | --- |
| **Name** | **USN** |
| **Chithranjan kumar** | **01JST17CS041** |
| **S Hemanth** | **01JST17CS126** |
| **Shankar S P** | **01JST17CS144** |

# 

*Under the Guidance of*

#### Dr. Mahesha P Dr. R. Guru

Assistant Professor, Assistant Professor

Dept of CS & E, Dept. of CS & E,

JSS STU, Mysuru JSS STU, Mysuru

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

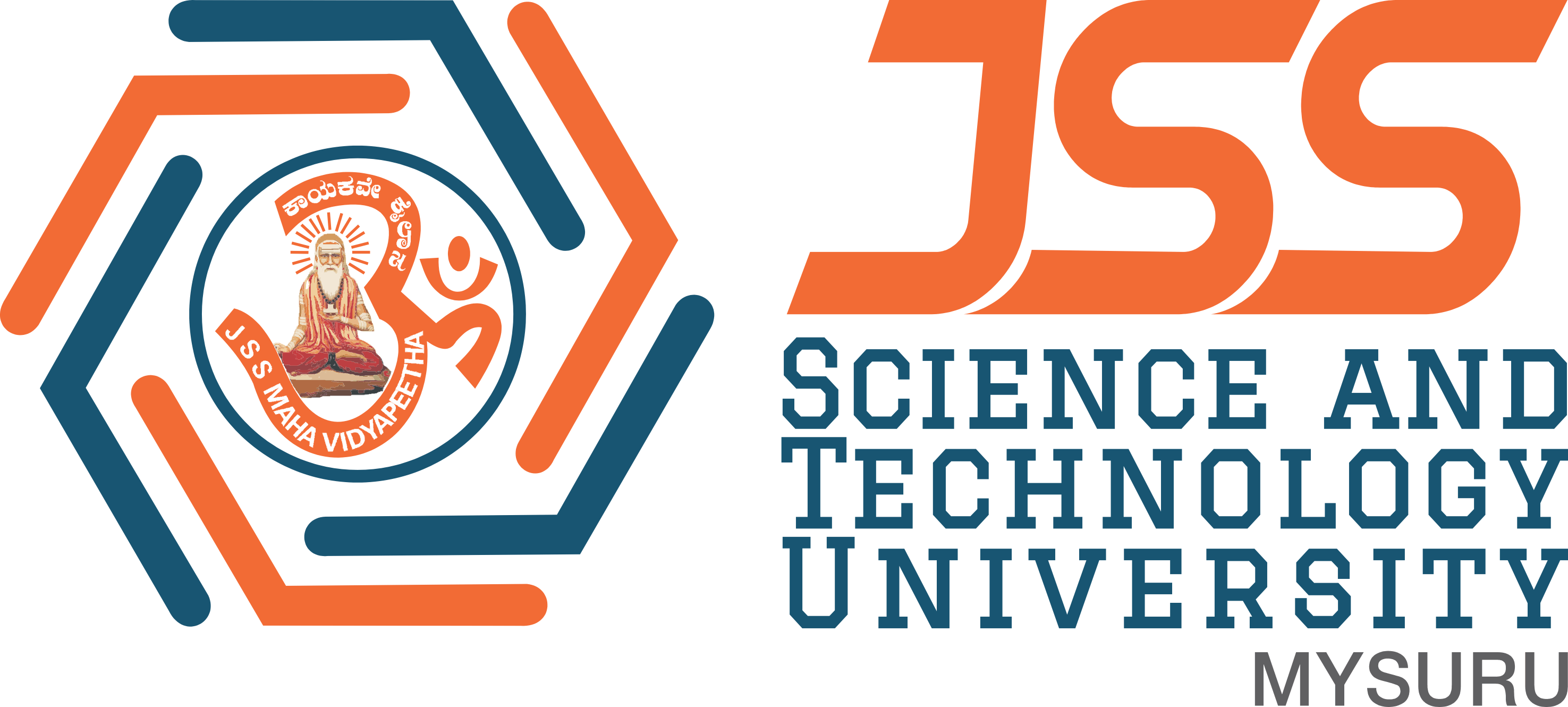
**November 2019**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**JSS MAHAVIDYAPEETHA**

**JSS SCIENCE AND TECHNOLOGY UNIVERSITY**

JSS Technical Institutions Campus, Mysuru – 570006



**CERTIFICATE**

This is to certify that the work entitled “Online Food Order System” is a bonafide work carried out by Chithranjan Kumar, S Hemanth, Shankar S P in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science and Engineering of JSS Science and Technology, Mysuru during the year 2019. It is certified that all corrections or suggestions indicated during CIE have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Database Systems (CS510) course.

## Course In Charge and Guide Lab In Charge and Guide

#### Dr. Mahesha P Dr. R. Guru

Assistant Professor Assistant Professor,

Dept of CS & E, Dept. of CS & E,

JSS STU, Mysuru JSS STU, Mysuru

**Place:** Mysuru **Date :** 22-11-2019

**Contents**

**1. Introduction**

1.1 Objective of the project-------------------------------------------------------------4

1.2 Features of the project--------------------------------------------------------------4

**2. System design**

2.1 ER Diagram-high level data modeling------------------------------------------6

2.2 Schema Diagram-conceptual data modeling---------------------------------7

**3. System Implementation**

3.1 Introduction to SQL/MySQL/MangoDb/DBMS-------------------------------8

3.2 Relational algebraic queries------------------------------------------------------9

3.3 Queries designed using SQL commands------------------------------------10

**4. Conclusion**------------------------------------------------------------------------18

**5. References**------------------------------------------------------------------------19

**1 Introduction**

It is known globally that, in today’s market, it is extremely difficult to start a new small-scale business and live-through the competition from the well-established and settled owners. In fast paced time of today, when everyone is squeezed for time, the majority of people are finicky when it comes to placing a food order. The customers of today are not only attracted because placing an order online is very convenient but also because they have visibility into the items offered, price and extremely simplified navigation for the order.

**1.1 Objective of the project**

Online ordering system that I am proposing here, greatly simplifies the ordering process for both the customer and the restaurant. System presents an interactive and up-to-date menu with all available options in an easy to use manner. Customer can choose one or more items to place an order which will land in the Cart. Customer can view all the order details in the cart before checking out. At the end, customer gets order confirmation details. Once the order is placed it is entered in the database and retrieved in pretty much real time. This allows Restaurant Employees to quickly go through the orders as they are received and process all orders efficiently and effectively with minimal delays and confusion.

**1.2 Features of the project**

**1.2.1 Easy and User-friendly Dashboard**

Customers seem to prefer a more simplified process of ordering food. Hence, a straightforward and user-friendly dashboard must be implemented. While the dashboard must be attractive in terms of look and feel, you need to make sure that the dashboard can effortlessly save the history of orders, restaurant menu and the delivery address. It will reduce the complications and at the same time enrich the user experience.

**1.2.2 Generating a code referral system**

Customer engagement is of utmost experience when you are taking your business online. So, you must give your customers enough opportunities to get engaged. One such facility is providing coupon codes. For example, your customers can avail coupon codes to avail discounts for first-time food orders. On the other hand, your system must have the facility of generating referral codes, through which they can refer your website or app to their friends and acquaintances. This is certainly a great way to market your app and offers a win-win situation for both the parties. While your customers can get a discount on their orders, you can easily reach out to a wider customer base.

###### **1.2.3 Customer Engagement with Push notifications**

###### Today, when the online food ordering is increasing at a staggering rate, it’s quite obvious that people will have more than one food ordering apps on their phone. But the only way you can create visibility and set apart yourself from the competitors is by sending push notifications. These powerful tools and with an engaging message popping up on the mobile screens constantly will keep your potential customers constantly notified about your discounts and special offers. This way, you can instantly grab your customers’ attention and at the same time keep them engaged.

###### **1.2.4 Social Media Integration**

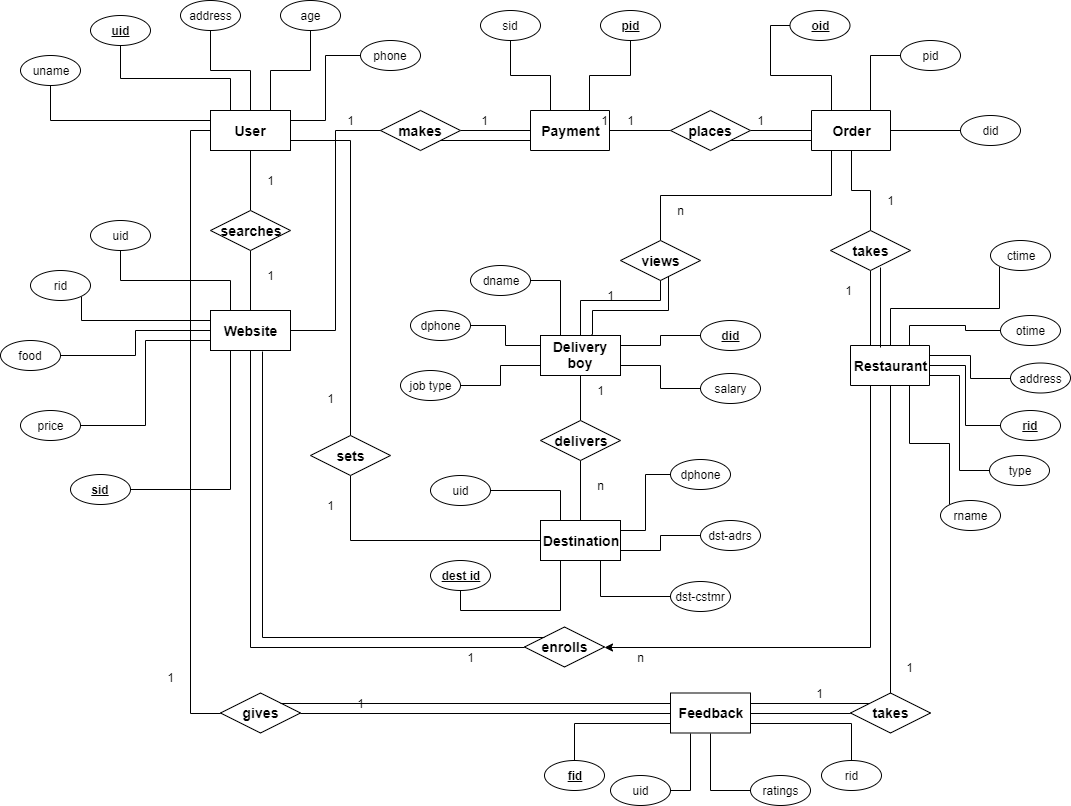
###### In recent times, the contribution of social media for the purpose of marketing is simply undeniable. It is one of those subtle aspects which improve consumer decision. With social media integration, customers can share photos of their favourite dish with their friends. This way, you can even make your app visible in social media and at the same time can attract customers on a regular basis.

###### **1.2.5 Cross-platform compatibility**

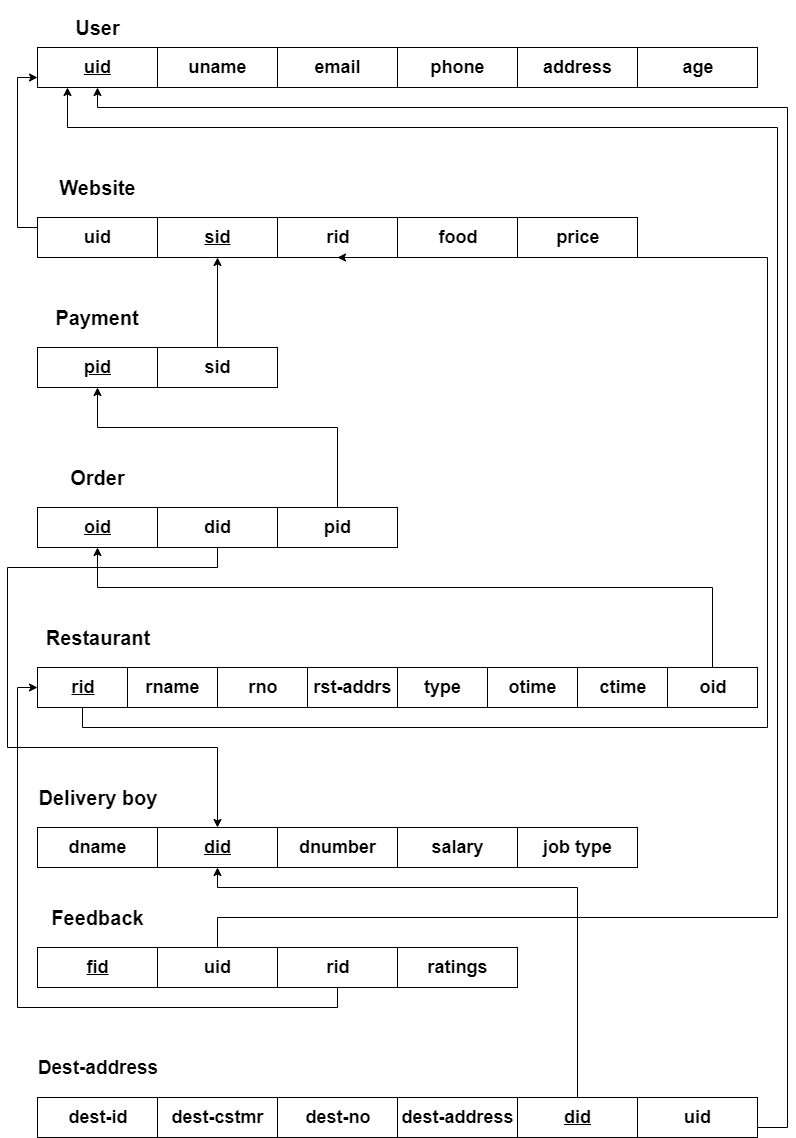
###### With an ever-increasing target market, accessibility should be your main priority. By enabling a cross-platform compatibility feature, you can increase your reach to across different platforms. No wonder, with more reach, you are likely to have more downloads within a short span of time. Food ordering software aids in efficient order and customer management, which further results in increasing sales in the long run.

**2. System design**

**2.1 ER Diagram-high level data modelling**



**2.2 Schema Diagram-conceptual data modelling**



**3 System Implementation**

**3.1 Introduction to SQL / MySQL / MangoDb / DBMS**

**3.1.1 SQL**

SQL(Structured Query Language) is a [domain-specific language](https://en.wikipedia.org/wiki/Domain-specific_language) used in programming and designed for managing data held in a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), or for stream processing in a [relational data stream management system](https://en.wikipedia.org/wiki/Relational_data_stream_management_system) (RDSMS). It is particularly useful in handling [structured data](https://en.wikipedia.org/wiki/Data_model), i.e. data incorporating relations among entities and variables.

SQL offers two main advantages over older read–write [APIs](https://en.wikipedia.org/wiki/API) such as [ISAM](https://en.wikipedia.org/wiki/ISAM) or [VSAM](https://en.wikipedia.org/wiki/VSAM). Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify *how* to reach a record, e.g. with or without an [index](https://en.wikipedia.org/wiki/Database_index).

**3.1.2 MySql**

MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, and "[SQL](https://en.wikipedia.org/wiki/SQL)", the abbreviation for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language).

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)). In 2010, when Oracle acquired Sun, Wideness [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB](https://en.wikipedia.org/wiki/MariaDB).

MySQL is a component of the [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) [web application](https://en.wikipedia.org/wiki/Web_application) [software stack](https://en.wikipedia.org/wiki/Software_stack)MySQL is used by many database-driven web applications, including [Drupal](https://en.wikipedia.org/wiki/Drupal), [Joomla](https://en.wikipedia.org/wiki/Joomla), [phpBB](https://en.wikipedia.org/wiki/PhpBB), and [WordPress](https://en.wikipedia.org/wiki/WordPress).

**3.1.3 MongoDb**

MongoDB is an [open source](https://whatis.techtarget.com/definition/open-source) database management system (DBMS) that uses a document-oriented database model which supports various forms of data. It is one of numerous nonrelational [database](https://searchsqlserver.techtarget.com/definition/database) technologies which arose in the mid-2000s under

the [NoSQL](https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL) banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using [tables](https://whatis.techtarget.com/definition/table) and [rows](https://searchoracle.techtarget.com/definition/row) as in [relational databases](https://searchdatamanagement.techtarget.com/definition/relational-database), the MongoDB architecture is made up of collections and documents.

**3.1.4 DBMS**

The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. By data, we mean known facts that can be recorded and that have embedded meaning. A datum is a unit of data. Meaningful data combined to form information. Hence, information is interpreted data - data provided with semantics.

**3.2 Relational algebraic queries**

1) Find the user who has given ratings more than 5.

2) Finding the most costly food available on the website.

3) Finding the delivery guy with maximum salary.

4) Finding the total number of full time delivery guys.

5) Finding the total sum of money that is spent on the delivery guys each month.

**3.3 Queries designed using SQL commands**

* **Simple Queries**

1. Find the restaurants for which a user has placed order.

select rid,foodi\_tems,price

from website

where uid=1001;

2. Find the restaurant that has taken a particular order given a food item**.**

select \* from restaurant;

select distinct(w.rid),r.rname

from website w, restaurant r

where foodi\_tems like 'IdliVada' and w.rid=r.rid;

3. Find the delivery boy who has taken a particular order.

select d.Did,dname,dphone

from delivery\_boy d, orders o

where o.did=d.did and oid=1;

4. Track the delivery boy who has taken orders.

select da.Did,dname,dest\_address

from destination da, delivery\_boy d

where da.did=d.did and d.did=1;

5. List the feedbacks given to restaurants.

select rname, ratings

from feedback f ,restaurant r

where uid = 1001 and f.rid=r.rid ;

* **Nested Queries**

1. Find the vegetarian restaurant with rating greater than 6.

select r.rname,r.rid

from restaurant r

where r.rid=any(select rid from feedback where ratings>6) and typ="V";

2. Find the non vegetarian restaurant with rating less than 5.

select r.rname ,r.rid

from restaurant r

where r.rid=any(select rid from feedback where ratings<5) and typ="N";

3. Find the salary of delivery boy whose salary is greater than part time workers.

select did,dname

from delivery\_boy

where salary>some

(select salary

from delivery\_boy

where joptype="P");

4. Find who ordered both vegetarian and non vegetarian.

select u.uname from users u

where u.uid in

(select w.uid from website w,restaurant r

where

u.uid = w.uid and w.rid = r.rid and r.typ = 'v'or r.typ = 'n');

5. List those customers who have not rated any restaurants more than 5.

select u.uid,u.uname from users u

where u.uid not in

(select f.uid from feedback f where f.uid=u.uid and ratings>5);

* **Set operations**
* **Union**

1. List all users who ordered either veg or non veg food.

select typ,w.uid, u.uname,foodi\_tems

from restaurant r, website w, users u

where w.rid=r.rid and typ="V"

union

select typ,w.uid, u.uname,foodi\_tems

from restaurant r, website w, users u

where w.rid=r.rid and typ="N";

2. List of all delivery boys of either part time or full time.

select joptype,did,dname,dphone,salary from delivery\_boy

group by joptype,did

having joptype="F"

union

select \* from delivery\_boy

group by joptype,did

having joptype="P";

* **Intersection**

1. List all users ordered both veg non veg.

select u.uid, uname

from website w, users u

where

exists

(select w.rid,typ

from restaurant r, website w, users u

where w.rid=r.rid and typ="N");

2. Details of both part time as well as full time delivery boy.

select joptype,did,dname,dphone,salary from delivery\_boy

where joptype

in

("P","F");

* **Group by**

1. Group orders by a users.

select uid,rid,foodi\_tems,price

from website

group by uid,sid;

2. Group restaurants according to type Veg or Non veg.

select typ, rname,rid, otime,ctime

from restaurant

group by typ,rid;

* **Having**

1. Group delivery boys according to their working mode (Part-time/Full-time).

having slary>5000

select joptype,did,dname,dphone,salary

from delivery\_boy

group by joptype,did

having salary>5000

order by joptype;

2. List restaurants having closing time >9.

select typ,rid,rname,ctime

from restaurant

group by typ,rid

having ctime>9;

* **Exists - Not Exists**

1. To print the restaurant id and its name which has atleast one order. and the price is less than 150

select rid,rname

from restaurant

where

exists

(select rid

from website

where restaurant.rid=website.rid and price<150 );

2. List all restaurants not in Bangalore.

select \* from restaurant

where not exists

(select rid from restaurant where address like "%bangalore" );

* **Aggregate function**

1. Find the average salary of a delivery boy.

select avg(salary)

from delivery\_boy;

2. List total number of orders by an user.

select count(sid)

from website

where uid=1003;

3. Find the delivery boy with max salary.

select did,max(salary)

from delivery\_boy;

4. Find the delivery boy with min salary.

select did,min(salary)

from delivery\_boy;

5. Find the total amount paid by an user on orders.

select sum(price)

from website

where uid=1001;

* **BETWEEN**

1. Users with age between 30 and 40.

select uname

from users

where age between 30 and 40;

2. Restaurant with ratings between 5 and 8.

select r.rname,f.rid

from restaurant r,feedback f

where r.rid=f.rid and f.ratings between 5 and 8;

3. Food items with price between 250 and 400.

select foodi\_tems,price

from website

where price between 250 and 400;

* **LIKE**

1. To find the age of users with 3 as the start number.

select uname,age

from users

where age like '3\_';

2. To find food items with price starting with 2.

select foodi\_tems,price

from website

where price like '2\_\_';

3. To find the delivery boy name which starts with g.

select dname,joptype

from delivery\_boy

where dname like 'G\_\_\_\_\_\_';

* **Correlated Queries**

1. Which delivery boy hasn't delivered yet.

select d.did from delivery\_boy d

where not exists (select dd.did from destination dd where d.did=dd.did);

* **VIEW**

1. Creating view for restuarent name which has rating greater than 5.

create view view1 as

select r.rname,f.ratings

from restaurant r join feedback f

on r.rid=f.rid and f.ratings>5;

2. Creating view for delivery boy and customer they served.

create view view2 as

select d.dname,e.dest\_customer

from delivery\_boy d join destination e

on d.did=e.did;

2. Select name and salary of part time delivery boy with greatest salary.

select dname,salary

from delivery\_boy

where salary>(select max(salary) from delivery\_boy where joptype="F") and joptype="P";

* **Triggers**

1. Create a trigger to increase the salary of the delivery boy by 10 percent.

create trigger sal2

before insert on delivery\_boy

for each row

set new.salary=new.salary+0.1\*(new.salary);

**4 Conclusion**

The main objective of the application is to help Computer Science students understands the basics of Java, JavaScript and HTML. The following results have been achieved after completing the system and relate back to the system’s objective.

• Should allow Computer Science students to browse through the code and application: This can be achieved when students are able to run and install the application. When they run the application, they can browse through the implementation of different objects.

• Should allow users to browse through different product categories: This is achieved through an easy to use graphical interface menu options.

• Should allow users to save items to the cart and view detailed information about the order: The users can add any number of items to the cart from any of the available food categories by simply clicking the Add to Cart button for each item. Once item is added to the cart, user is presented with detailed order to review or continue shopping.

• Should allow the user to Check out the item(s): This is achieved using the “Proceed to checkout button” in the cart initially and then “Checkout” button at last step after “review Order” step. Button is disabled when there are no items in the cart.

• Should allow the user to process the payment: This is achieved when user selects “Processed to Checkout” button and fill up the Payment information details.

• Should allow the user to see Success message after placing an order: This is achieved when user successfully places an order. The user is given the order conformation number along with success message.

**5 References**

1. Ramez Elmasri, Shamkanth B. Navathe, 6th edition, Database Systems.

2. <http://getbootstrap.com/>

3. https://www.youtube.com/watch?v=oepmLGQP1m4&list=PLUoqTnNH2Xz\_BUrjcahKWDh

4. <http://www.javazoom.net/jzservlets/uploadbean/uploadbean.html>

5. https://javabrains.io/

6. http://www.java2s.com/Tutorial/Java/0360\_\_JSP/JSPDummyShoppingCart.htm

7. <https://docs.oracle.com/cd/E24628_01/server.121/e41484.pdf>