

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



A Mini Project Report on

ONLINE FOOD ORDERING SYSTEM

*Submitted in partial fulfillment of the requirements as a part of the DBMS Lab for the V Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi*

Submitted by

RACHIT TAWANI

1RN18IS082

MEHUL GUPTA

1RN18IS066

Under the Guidance of

Faculty Incharge

DR. S Sathish K

**Professor & Head
Dept. of ISE, RNSIT**

Lab Incharge

Mr. R Rajkumar

**Asst. Professor
Dept. of ISE, RNSIT**



ESTD: 2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

**Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098**

2020 – 2021

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



ESTD:2001

An Institute with a Difference

CERTIFICATE

This is to certify that the Mini project report entitled **ONLINE FOOD ORDERING SYSTEM** has been successfully completed by **RACHIT TAWANI** bearing USN **1RN18IS082** and **MEHUL GUPTA** bearing USN **1RN18IS066**, presently **V semester** student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree **Bachelor of Engineering in Information Science and Engineering** under **Visvesvaraya Technological University, Belagavi** during academic year 2020 – 2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

DR. S SATHISH KUMAR
Faculty Incharge

Mr. R Rajkumar
Lab Incharge

Dr. S SATHISH KUMAR
Professor and HOD

External Viva

Name of the Examiners

1. _____

2. _____

Signature with date

DECLARATION

We, RACHIT TAWANI [USN: 1RN18IS082] and MEHUL GUPTA [USN:1RN18IS066] student of V Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Mini project entitled *Online Food Ordering System* has been carried out by us and submitted in partial fulfillment of the requirements for the *V Semester degree of Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi* during academic year 2020-20201.

Place: Bengaluru

Date:

RACHIT TAWANI

(1RN18IS082)

MEHUL GUPTA

(1RN18IS066)

ABSTRACT

Our Project Online Food Ordering System includes registration and login facilities to user who wishes to place an order from our Web Application after the authentication using email and password given at the time of registration. After user logs in, our home page opens from where user can choose from different products and can add to cart. After finalizing products user need to give address, payment details to place the order and can view previous order.

The mini project has been placed to be having a view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. The database connectivity is planned using the “SQL Connection” methodology. The interactive web pages have been designed using HTML along with CSS and Bootstrap for designing them.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this Mini project work.

We would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

We would like to express our thanks to our Principal **Dr. M K Venkatesha** for his support and inspired us towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. S SATHISH KUMAR** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our Project work.

We would like to express our profound and cordial gratitude to our Lab Incharge Mr. **R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance, constructive comments and continuous encouragement throughout the Project work.

We would like to express our profound and cordial gratitude to our Faculty Incharge **DR. S SATHISH KUMAR** Assistant Professor, Department of Information Science and Engineering for his/her valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped us to carry out the project work.

And lastly, We would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in carrying out this Project work.

RACHIT TAWANI

USN: 1RN18IS082

MEHUL GUPTA

USN:1RN18IS066

TABLE OF CONTENTS

CERTIFICATE	ii
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGMENT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	vii
ABBREVIATIONS	viii
1. INTRODUCTION	1
1.1 Background	1
1.2 Introduction to Online Food Ordering System	1
2. E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM	2
2.1 Description of ER Diagram	2
2.2 Description of Relational Schema Diagram	4
3. SYSTEM DESIGN	6
3.1 Table Description	6
3.2 Stored Procedure	8
3.3 Triggers	9
4. IMPLEMENTATION	10
4.1 Front-end Development	10
4.2 Back-end Development	11
4.3 Discussion of code Segment	13
4.4 Discussion of Results	17
4.5 Application of project	26
5. CONCLUSION AND FUTURE ENHANCEMENT	27
5.1 Conclusion	27
5.2 Future Enhancement	27
REFERENCES	28

LIST OF FIGURES

Figure. No.	Descriptions	Page
Figure. 2.1	E-R Diagram for Online Food Ordering System	03
Figure. 2.2	Relational Schema - Online Food Ordering System	04
Figure. 4.1	Landing Page	17
Figure. 4.2	Sign up Pgae	18
Figure. 4.3	Signup Page	19
Figure. 4.4	Login Page	22
Figure. 4.5	Login Page	23
Figure. 4.6	Home Page	23
Figure. 4.7	Modal Page	24
Figure. 4.8	Cart page	24
Figure. 4.9	Purchase page	25
Figure. 4.10	Purchase page	25
Figure. 4.11	Order page	26
Figure. 4.12	Order history	26

ABBREVIATIONS

CSS	-	Cascading style sheets
DBMS	-	Database Management System
ER	-	Entity Relationship
HTML	-	Hypertext Markup Language
HTTP	-	Hypertext Transfer Protocol
JS	-	JavaScript
PHP	-	PHP Hypertext Preprocessor
SQL	-	Structured Query Language

Chapter 1

INTRODUCTION

1.1 Background

A **database** is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

The database management system (DBMS) is the software that interacts with end users, applications, the database itself to capture and analyze the data and provides facilities to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.2 Introduction to Online Food Ordering System

This mini project Online Food Ordering (OFD) System is designed for users to have luxury of having their favorite food delivered to their door steps and managing their orders. Users can sign-up and login in the system and can have a look to all the products in the home page and can add multiple products with different quantity to the cart from where user can place the order by filling out the basic details and after that order confirmation will be shown where user can see their order history as well.

These services are provided in an efficient, cost effective manner with a goal of reducing time and maximizing the ease of ordering food. In Online Food Ordering System user has access to see all the products, price and description and accordingly can to cart and order.

Chapter 2

E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM

2.1 Description of ER Diagram

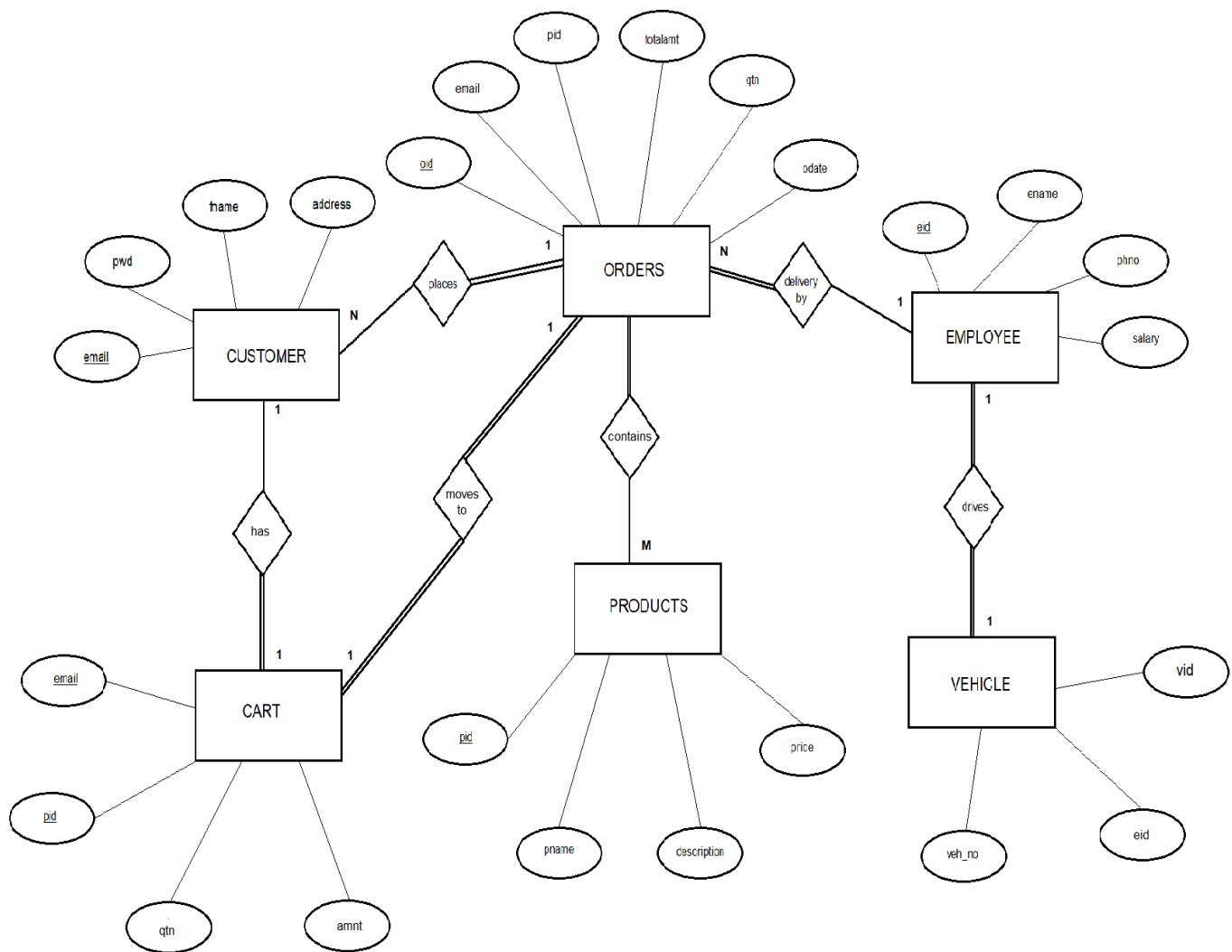


Figure 2.1: E-R Diagram for Online Food Ordering System

Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.

2.1.1 Components of Online Food Ordering System, E-R Diagram

Entity types like **CUSTOMER** and **PRODUCTS** are in rectangular boxes.

1. Relationships like **CONATAINS** and **HAS** are in diamond boxes, attached to entity types with straight lines.
2. Attributes are shown in ovals like **Name** and **Email**, each attached by a straight line to entity or relationship type.
3. Key attributes like **Email** and **Product_id** are underlined.

2.1.2 E-R Diagram Relationships Description

1. **CUSTOMER:ORDERS** is of cardinality N:1 as Many customer can place orders at a time and therefore connected by **PLACES** relationship. There is total participation from ORDERS as all the orders must have a customer.
2. **CUSTOMER:CART** is of cardinality 1:1 as one customer can have its own one cart and therefore connected by **HAS** relationship. There is total participation from CART as cart has customer, but partial participation from CUSTOMER as a customer may exist without a cart
3. **CART:ORDERS** is of cardinality ration 1:1 as one cart items can place only one order and hence connected by **MOVES TO** relationship. There is total participation from CART and ORDERS.
4. **ORDERS:PRODUCTS** is of cardinality N:M as one order can have multiple products and one product can be there in multiple orders and therefore connected by **CONTAINS** relationship. There is total participation from orders as all the orders must have some products.
5. **ORDERS:EMPLOYEE** is of cardinality N:1 as one employee can be assigned to deliver many orders at a time and many orders can be delivered by one employee and therefore connected by **DELIVERED BY** relationship. There is total participation from ORDERS as all orders must be serviced by some employee.
6. **EMPLOYEE:VEHCHILE** is of cardinality of 1:1 as one employee is assigned with one vehicle and there is total participation from both employee and vehicle.

2.2 Description of Relational Schema Diagram

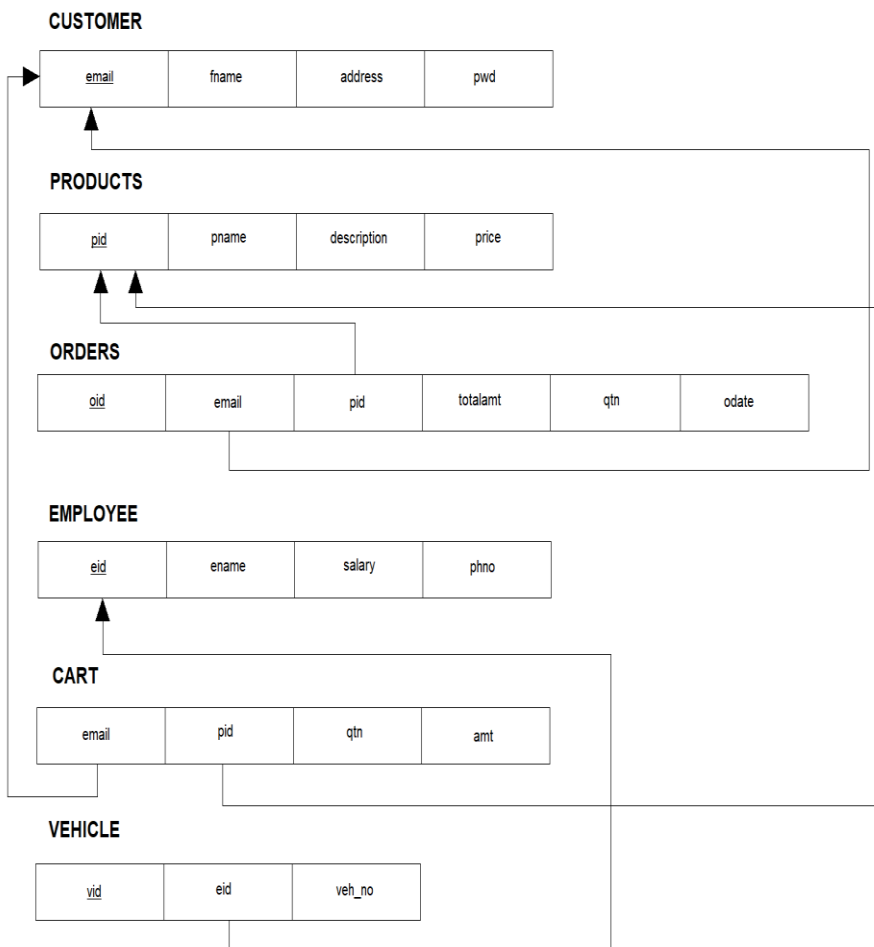


Figure 2.2 Relational Schema Diagram for Online Food Ordering System

2.2.1 General Constraints

1. **NULL Constraint:** Attributes that are under NOT NULL constraints have to be filled compulsorily. Almost all the attributes in the project are under NOT NULL constraint.
2. **Entity Integrity Constraint:** This constraint makes sure that no primary key can have a NULL value assigned to it. The primary keys involved in the project include:
 - Email
 - Product_ID
 - Employee_ID
3. **Referential Integrity Constraints:** A table in the back end of the project may have references pointing to an attribute in another table. For example: PRODUCT_ID in the ORDER table refers to PRODUCT_ID in PRODUCT TABLE table. The various tables are also linked with multiple foreign keys which are all set to cascade any update or delete operation on the attribute in the main table. The various Foreign Key attributes are:
 - Employee_ID
 - Email
 - Product_ID

2.2.2 Schema Description

The above Figure.2.2 shows the relational schema of Online Food Ordering System. It has the following entities and normalization of tables is taken in consideration.

1. **CUSTOMER:** This table contains the details of customers like email, name, password, address, phno.
2. **PRODUCTS:** This table contains details about products and has attributes product_id , name, description and price.
3. **ORDERS:** This table contains details of order and columns, order_id, email referring to customer table, product_id referring to product table.
4. **CART:** This table consists the details of cart of customer and has columns email, product_id, qtn and amount.
5. **EMPLOYEE:** This table consists details of all the Employees and has employee_id , ename, phno, salary.
6. **VEHCILE:** This table consists of list of all the vehicles serviced by each employee. It has the following columns v_id, veh_no, employee_id.

Chapter 3

SYSTEM DESIGN

3.1 Table Description

CUSTOMER

Server: MySQL:3306 » Database: marwadi_foods » Table: customer									
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	email	varchar(25)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	fname	varchar(25)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	address	varchar(40)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 4	pwd	varchar(25)	latin1_swedish_ci		Yes	NULL			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Fulltext](#)
[Fulltext](#)

Table 3.1 Customer

CART

Server: MySQL:3306 » Database: marwadi_foods » Table: cart									
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	email	varchar(225)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 2	pid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 3	qtn	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/> 4	amnt	int(11)			Yes	NULL			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Fulltext](#)
[Fulltext](#)

Table 3.2 Cart

PRODUCT

Server: MySQL:3306 » Database: marwadi_foods » Table: products									
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	pid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	pname	varchar(25)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	description	varchar(100)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 4	price	int(11)			Yes	NULL			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
[Change](#)
[Drop](#)
[Primary](#)
[Unique](#)
[Index](#)
[Fulltext](#)
[Fulltext](#)

Table 3.3 Products

ORDER1

Server: MySQL 3306 » Database: marwadi_foods » Table: order1

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Triggers](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	oid1	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	email	varchar(225)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	pid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 4	qtn	int(11)			No	None			Change Drop More
<input type="checkbox"/> 5	amt	int(11)			No	None			Change Drop More
<input type="checkbox"/> 6	odate	datetime			No	None			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
 Change
 Drop
 Primary
 Unique
 Index
 Fulltext
 Fulltext

Table 3.4 Order1

ORDERS

Server: MySQL 3306 » Database: marwadi_foods » Table: orders

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Triggers](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	oid	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	email	varchar(25)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/> 3	pid	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/> 4	qtn	int(11)			No	None			Change Drop More
<input type="checkbox"/> 5	totalamt	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/> 6	odate	datetime			No	None			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
 Change
 Drop
 Primary
 Unique
 Index
 Fulltext
 Fulltext

Table 3.5 Orders

EMPLOYEE

Server: MySQL 3306 » Database: marwadi_foods » Table: employee

[Browse](#)
[Structure](#)
[SQL](#)
[Search](#)
[Insert](#)
[Export](#)
[Import](#)
[Privileges](#)
[Operations](#)
[Triggers](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	eid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	ename	varchar(25)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	phno	varchar(10)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 4	salary	int(11)			Yes	NULL			Change Drop More

☐ Check all
 With selected:
 [Browse](#)
 Change
 Drop
 Primary
 Unique
 Index
 Fulltext
 Fulltext

Table 3.6 Employee

VEHICLE

Server: MySQL 3306 » Database: marwadi_foods » Table: vehicle									
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers									
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	vid	int(11)			No	None			Change Drop More
<input type="checkbox"/> 2	eid	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/> 3	veh_no	int(11)			Yes	NULL			Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext									

Table 3.7 Vehicle

3.2 Stored Procedures

One stored procedure is used to add products with specified quantity to the cart from the home page.

- DELIMITER \$\$

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `getproduct`(IN `mail`
VARCHAR(225), IN `id` INT(11), IN `qtn` INT(11), IN `amt` INT(11))
BEGIN
    INSERT INTO cart VALUES(mail,id,qtn,amt);
END $$
DELIMITER ;
```

Second stored procedure is used to remove products from cart if user wants to remove some of products.

- DELIMITER \$\$

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `deleteproduct`(IN `id`
INT(11))
BEGIN
    DELETE FROM cart WHERE pid=id;
END $$
DELIMITER ;
```


3.3 Trigger

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

In this project a trigger is added to insert products and delete products in the order table as we insert or remove products from cart table.

- ```
CREATE TRIGGER `insertlog` AFTER INSERT ON `cart`
FOR EACH ROW INSERT INTO orders VALUES(null, new.email, new.pid,
new.qtn, new.amnt, NOW())
```
- ```
CREATE TRIGGER `deletelog` BEFORE DELETE ON `cart`  
FOR EACH ROW DELETE FROM orders  
WHERE pid=old.pid
```

Chapter 4

IMPLEMENTATION

4.1 Front-end Development

The front-end is built using a combination of technologies such as Hypertext Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS). Front-end developers design and construct the user experience elements on the web page or app including buttons, menus, pages, links, graphics and more.

4.1.1 Hypertext Markup Language

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. HTML is the standard markup language for creating Web pages. It stands for Hyper Text Markup Language. It describes the structure of a Web page. It consists of a series of elements. It elements tell the browser how to display the content. It elements are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page.

4.1.2 Cascading style sheets

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications. Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. CSS solved that issue. CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file. CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

4.1.3 JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly. Advantages are: **Less server interaction, immediate feedback to the visitors, increased interactivity and richer interfaces.**

4.2 Back-end Development

Backend is server side of the website. It stores and arranges data, and also makes sure everything on the client-side of the website works fine. It is the part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend.

4.2.1 Backend scripting language - PHP Hypertext Preprocessor

PHP is used as the server side scripting language. PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. It is compatible with all servers used today. It is easy to use and runs efficiently on the server side. It can run on various platforms like windows, Linux, UNIX, Mac OS-X etc. and since it is a scripting language, it comes with predefined functions which makes it easy to implement any logic necessary.

4.2.2 Web Server – Apache

Apache is an open-source and free web server software that powers around 46% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation. It allows website owners to serve content on the web — hence the name “web server”. Although we call Apache a web server, it is not a physical server, but rather software that runs on a server. Its job is to establish a connection between a server and the browsers of website visitors (Firefox, Google Chrome, Safari, etc.) while delivering files back and forth between them (client-server structure). Apache is a cross-platform software, therefore it works on both UNIX and Windows servers.

4.2.3 Database – MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed and supported by MySQL AB, which is a Swedish company. It is released under an open-source license. So you have nothing to pay to use it. It is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. It uses a standard form of the well-known SQL data language. It works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc. It works very quickly and works well even with large data sets. It is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). It is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

4.3 Discussion of Code Segment

This section talks about the important code sections and modules that are implemented in the Online Food Ordering System. These modules add logic to the complete system, and make it function the way it is supposed to. It also talks about the integration between the front end HTML code and the back end MySQL database.

4.3.1 Login Module

```
<?php
$conn = new mysqli("localhost", "root", "", "marwadi_foods");

// Check connection
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}
$requestType = $_SERVER['REQUEST_METHOD'];
if ($requestType == 'POST') {
    $email = $_POST['email'];
    $pwd = $_POST['pwd'];
    $sql = "SELECT pwd FROM customer WHERE email = '$email'";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();
    if ($result->num_rows > 0 and $row["pwd"]==$pwd) {
        header("Location:home.php");
        exit;
    }
?>
    <script>
        alert("Successfully Logged-in!");
    </script>
<?php
}
else{
    ?>
    <script>
        alert("Wrong Credentials");
    </script>
<?php
}
?>
```

4.3.2 Sign-up module

```
<?php
$conn = new mysqli("localhost", "root", "", "marwadi_foods");

// Check connection
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}

$requestType = $_SERVER['REQUEST_METHOD'];
if ($requestType == 'POST') {
    $email = $_POST['email'];
    $fname = $_POST['name'];
    $addr = $_POST['addr'];
    $pwd = $_POST['pwd'];
    $query = "INSERT INTO customer VALUES('" . $email . "', '" . $fname . "', '" . $addr . "', '" . $pwd . "')";
    $res = $conn->query($query);
    if ($res) {
        header("Location:home.php");
    }
    ?>
    <script>
        alert("Successfully Signed-Up!");
    </script>
}
else{
    ?>
    <script>
        alert("User already exist!");
    </script>
}
?>
```

4.3.3 Managing cart module

```
<?php
session_start();
$conn = new mysqli("localhost", "root", "", "marwadi_foods");
if(isset($_POST['Add_To_Cart']))
{
    $sql="CALL `getproduct`('".$_POST["Email"]."', '".$_POST["pid"]."', '".$_POST["Quantity"]."', '".$_POST["Price"]*$_POST["Quantity"]."')";
    $res=$conn->query($sql);
}
if($_SERVER["REQUEST_METHOD"]=="POST")
{
    if(isset($_POST['Add_To_Cart']))
    {
        if(isset($_SESSION['cart']))
        {
            $myItems=array_column($_SESSION['cart'],'Item_name');
            if(in_array($_POST['Item_name'],$myItems))
            {
                echo"<script>
                    alert('Item Already Added');
                    window.location.href='home.php';
                </script>";
            }
            else
            {
                $count=count($_SESSION['cart']);
                $_SESSION['cart'][$count]=array('Pid'=>$_POST['pid'],'Item_name'=>$_POST['Item_name'],'Price'=>$_POST['Price'],'Quantity'=>$_POST['Quantity']);
                print_r($_SESSION['cart']);
                echo"<script>
                    window.location.href='home.php';
                </script>";
            }
        }
        else
        {
            $_SESSION['cart'][0]=array('Pid'=>$_POST['pid'],'Item_name'=>$_POST['Item_name'],'Price'=>$_POST['Price'],'Quantity'=>$_POST['Quantity']);
            echo"<script>
                window.location.href='home.php';
            </script>";
        }
    }
}
```

```
if(isset($_POST['Remove_Item']))
{
    foreach ($_SESSION['cart'] as $key =>$value)
    {
        if($value['Item_name']==$_POST['Item_name'])
        {
            $sql="CALL `deleteproduct`('".$value['Pid']."'");
            $res=$conn->query($sql);
            unset($_SESSION['cart'][$key]);
            $_SESSION['cart']=array_values($_SESSION['cart']);
            echo"<script>
                window.location.href='mycart.php';
            </script>";
        }
    }
}
?>
```


4.3.4 Order module

```
<?php
session_start();
$conn = new mysqli("localhost", "root", "", "marwadi_foods");

// Check connection
if ($conn->connect_errno) {
    echo "Failed to connect to MySQL: " . $conn->connect_error;
    exit();
}

$employee = rand(1001,1005);
$sql = "SELECT * FROM EMPLOYEE WHERE eid='$employee'";
$result = $conn->query($sql);
$row = $result->fetch_assoc();

$sql1 = "SELECT * FROM vehicle where eid='$employee'";
$result1 = $conn->query($sql1);
$row1 = $result1->fetch_assoc();

$requestType = $_SERVER['REQUEST_METHOD'];
if ($requestType == 'POST') {
    $sql2="INSERT INTO order1 SELECT * FROM orders";
    $result2 = $conn->query($sql2);
}

??
```

4.4 Discussion of Results



Figure 4.1 Index Page

The above Figure 4.1 is the snapshot of the landing page with the tag line and login and signup buttons for user, if user already has an account he can directly login else he can sign up.



Figure 4.2 Signup Page

The above Figure 4.2 is the snapshot of the signup page with the signup form. User can sign up by filling up all the necessary details.



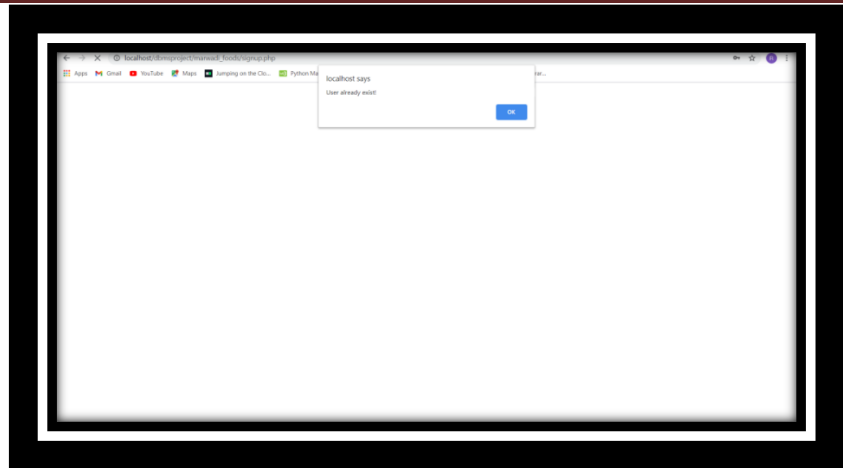


Figure 4.3 Signup Page

The above Figure 4.3 is the snapshot of the alert with a message that user already exists and hence user can directly login to the system.



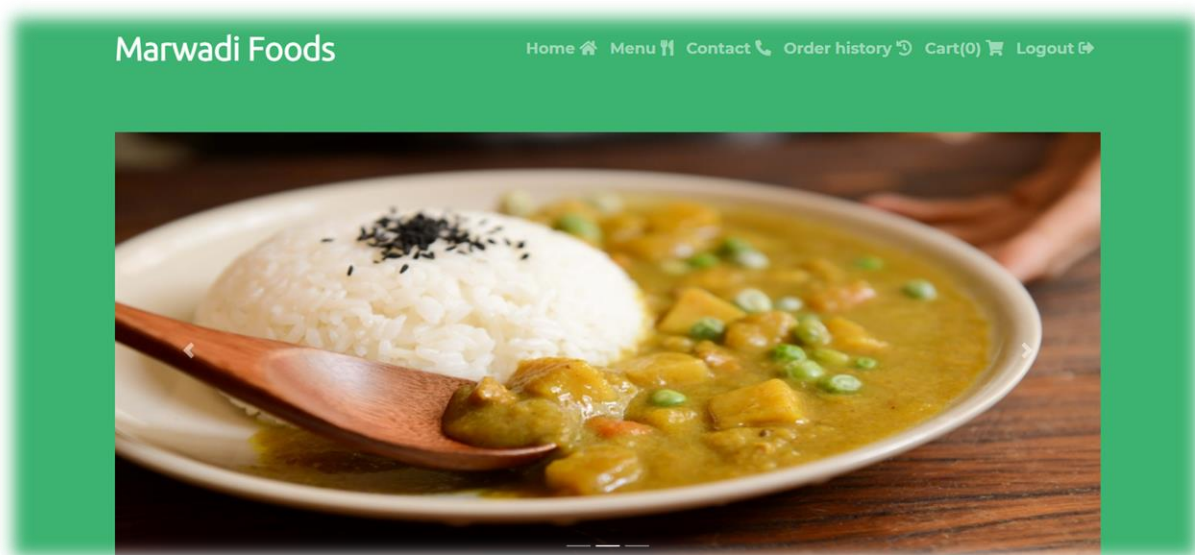
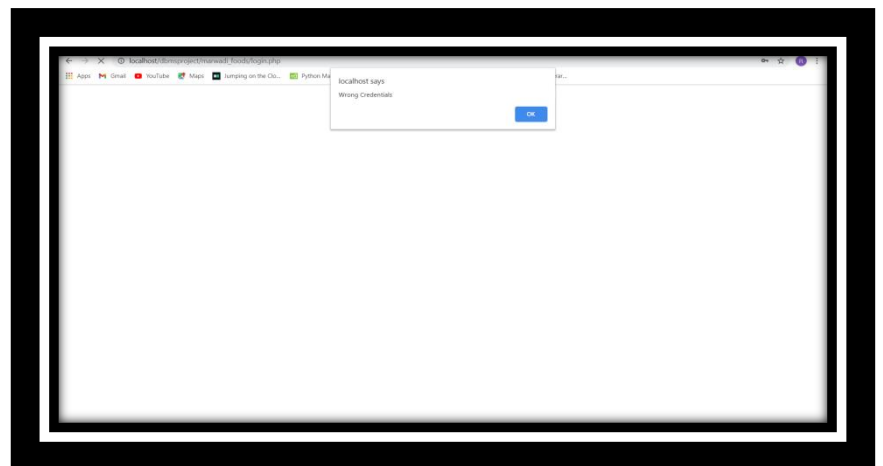
Figure 4.4 Login Page

The above Figure 4.4 is the snapshot of login page where user can enter their email id and password to login.



Figure 4.5 Login Page

The above Figure 4.5 is the snapshot of the alert when the email and its corresponding password doesn't match as entered by user.



**Safety**

We take safety of our customers very seriously hence maintaining the high standards of safety.

**Taste**

Our food taste soo good, you lick your fingers.

**Guaranteed 30 min delivery.**

We don't like to make our customer wait. We Guarantee to deliver your food within 30 mins.

Menu

Menu

**Pizza**

Price - ₹200

Flattened bread dough with toppings.

Quantity

Add to Cart

**Burger**

Price - ₹150

Sandwich consisting of patties, lettuce and sauces.

Quantity

Add to Cart

**Chilli Paneer**

Price - ₹100

Cubes of fried paneer tossed in soy and chilli sauce.

Quantity

Add to Cart

**Dal Baati**

Price - ₹250

Dal-combination of five lentils and Baati-small wheat bread balls.

Quantity

Add to Cart

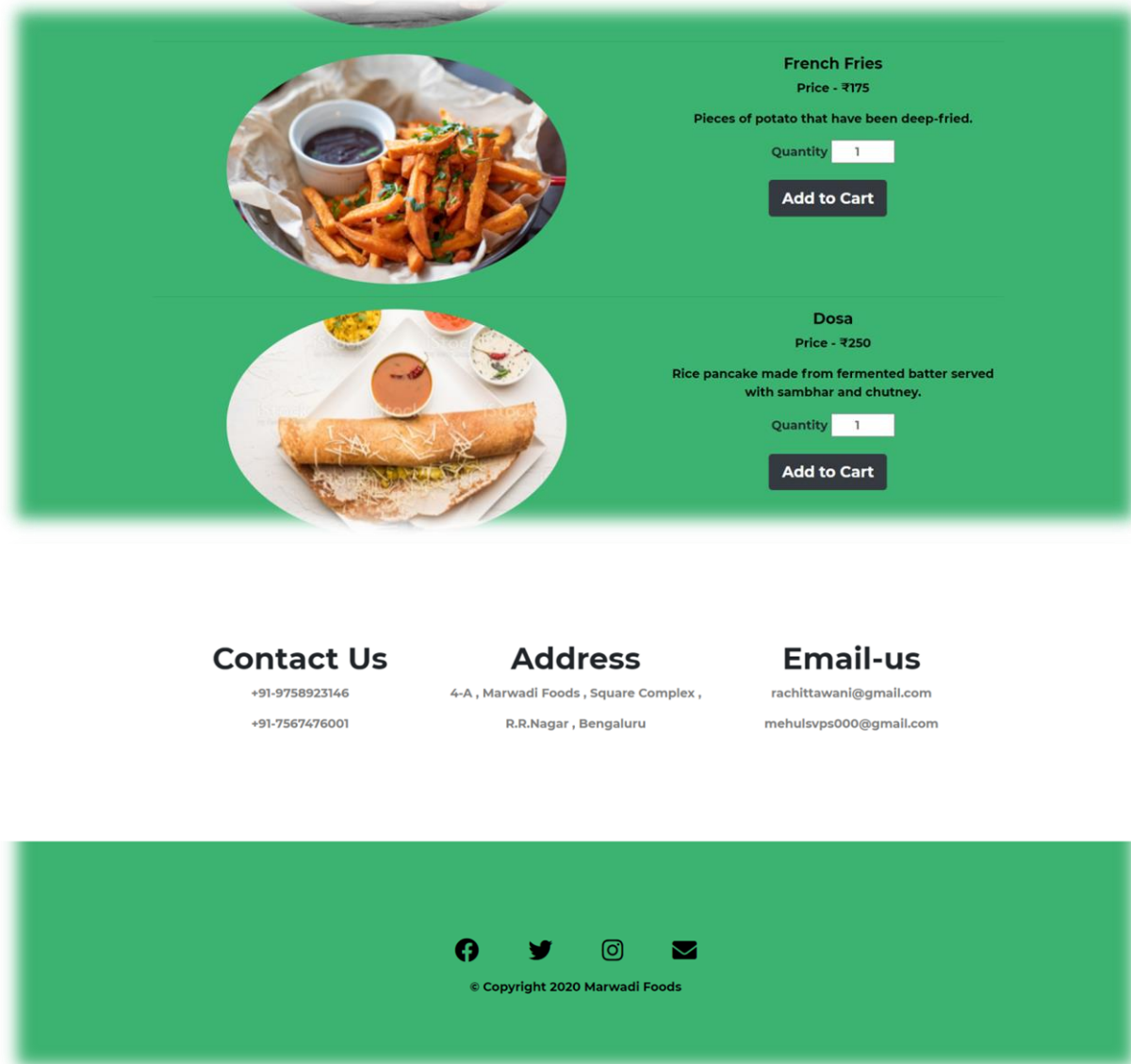


Figure 4.6 Home Page

The above Figure 4.6 is the snapshot of the homepage where user can go through all the products available and can add them to cart with specified quantity.

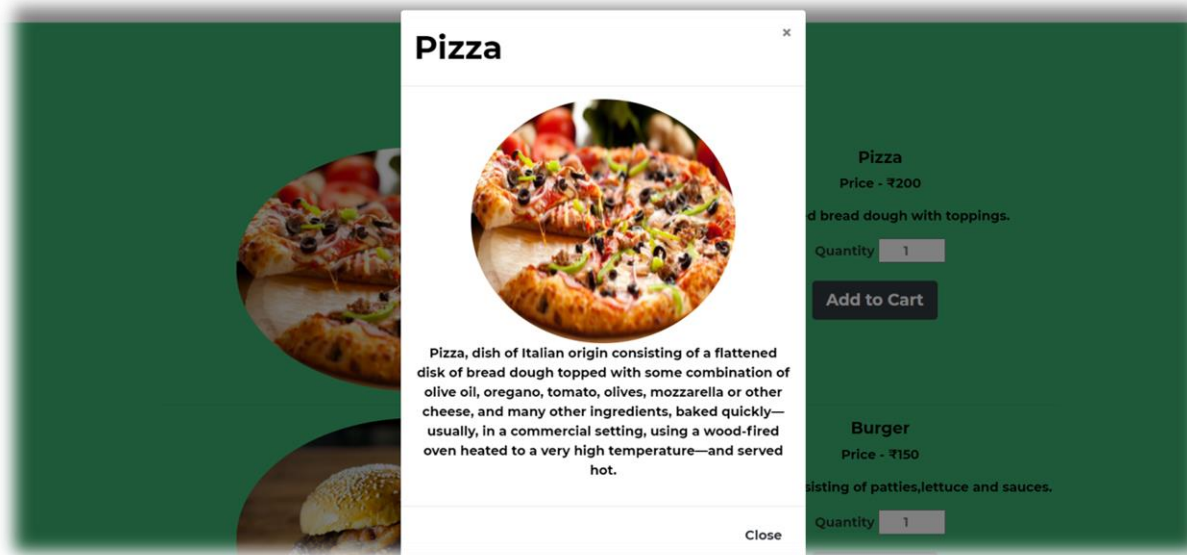


Figure 4.7 Modal Page

The above Figure 4.1 is the snapshot of the modal of a product which gives more details about the product description.

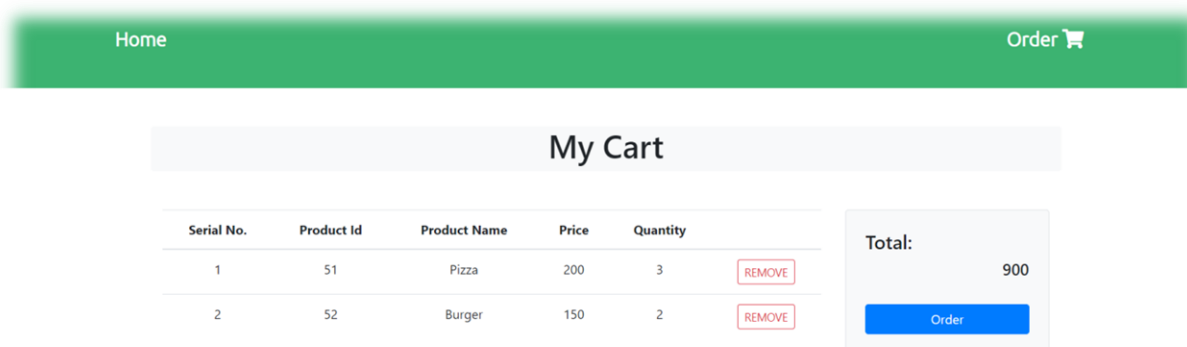


Figure 4.8 Cart Page

The above Figure 4.1 is the snapshot of the cart page where user can see the total amount to be paid and user can also remove certain product if user wishes to.

The form is titled 'Purchase Page' and is divided into two main sections: 'Billing Address' and 'Payment'. The 'Billing Address' section includes fields for 'Full Name', 'Email', 'Address', 'City', 'State', and 'Pin Code'. A tooltip message 'Please fill out this field.' is visible over the 'Email' field. The 'Payment' section includes a 'Accepted Cards' section with logos for Visa, Mastercard, and American Express. Below this are fields for 'Name on Card', 'Credit Card Number', 'Expiry Month', 'Expiry Year', and 'CVV'. A 'Back to Cart' link with a shopping cart icon is located in the top right corner of the form.

Figure 4.9 Purchase Page

The above Figure 4.1 is the snapshot of the purchase page where users is asked to enter delivery address and payment details to place an order.

This form is a filled-in version of the one in Figure 4.9. The 'Billing Address' section contains the following data: 'Full Name' is 'Rachit Tawani', 'Email' is 'rachit@gmail.com', 'Address' is 'RNSIT college', 'City' is 'Bengaluru', 'State' is 'Karnataka', and 'Pin Code' is '560098'. The 'Payment' section contains: 'Accepted Cards' with logos for Visa, Mastercard, and American Express, 'Name on Card' is 'Rachit', 'Credit Card Number' is '1234-5678-9078-4321', 'Expiry Month' is '2', 'Expiry Year' is '22', and 'CVV' is '123'. A checkbox labeled 'Shipping address same as billing' is checked. A green button labeled 'Continue to checkout' is at the bottom.

Figure 4.10 Purchase Page

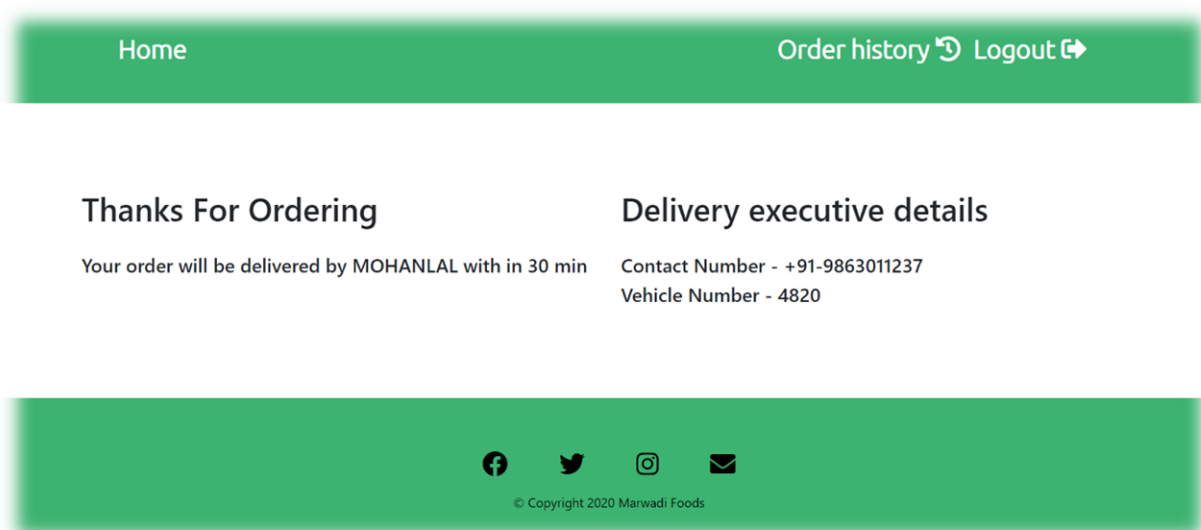


Figure 4.11 Order Page

The above Figure 4.11 is the snapshot of the order page in which order confirmation details are showed along with the delivery executive details like contact number and vehicle number.

Order History				
Order Id	Product Id	Quantity	Price	Date & Time
51	51	3	600	2020-12-23 12:51:10
52	52	2	300	2020-12-23 12:51:15

Figure 4.12 Order History Page

The above Figure 4.12 is the snapshot of the order history page having details of all the orders placed by the user sorted according to the dates of order placed.

4.5 Application of project

These types of Online Food Ordering System are examples of flourish technology has had on the world. These help user to order food from their laptops, mobiles phones or tablets. They are easy to use and get the job done in very few steps. The traditional methods of having food at home includes physically going to restaurant and long wait. As they are deployed on the web, they can be multi lingual which would support diversity and cause these systems to spread worldwide.

Major applications of Online Food Ordering System are:

1. All the types of different food are available in one place.
2. Easy access to food at user's doorstep.
3. A database of all the year's activity is created for use if needed later.

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

This project was an attempt to make the structure and working of an Online Food Ordering System simpler and user friendly. This mini project was an attempt to make it similar to the real-world implementation. In this scenario, all the work of the Online Food Ordering System was achieved in a constructive manner. Given the right guidance and support its applications and availability can be enhanced.

5.2 Future Enhancements

1. Email and message verifications could be sent to the users on successfully placing an order.
2. Hosting it on an online web server.
3. Addition of admin login to add more products directly from the front-end.
4. Improving the order history page of a user.

Chapter 6

REFERENCES

- [1] Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGRAW HILL, 3rd Edition.
- [2] Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.
- [3] www.stackoverflow.com
- [4] www.mysql.com/doc
- [5] <https://docs.microsoft.com>
- [6] <https://www.geeksforgeeks.org>
- [7] <https://www.youtube.com>