**PROJECT SYNOPSIS**

**ON**

**“PIZZA-HUT MANAGEMENT SYSTEM”**

Project work submitted as partial fulfillment of the course for the Award of the degree

**BACHELOR OF COMPUTER APPLICATION**

**OF**

**BANGALORE UNIVERSITY**

****

SUBMITTED BY

**ABHISHEK M : 181NSB7002**

Under the Guidance

Of

**(Prof.ShivaKumar.T)**



**GT INSTITUTE OF MANAGEMENT STUDIES & RESEARCH**

Sunkadakatte, Bangalore-560091

**GT INSTITUTE OF MANAGEMENT STUDIES & RESEARCH**

**Sunkadakatte, Bangalore-560091**

**Certificate**

This is to Certify that ABHISHEK M (181NSB7002) BCA 6TH Semesterin “GT Institute of Management Studies & Research” has satisfactorily completed her project entitled “PIZZA-HUT MANAGEMENT SYSTEM” as a fulfillment of requirements for requirements for the award of the degree in Bachelor of Computer Application awarded by Bangalore University for the academic year 2020-2021.



Signature of the Guide Signature of the Principal

**ACKNOWLEDGEMENT**

It is not exaggeration to say that the emergence of this book was outcome of observing project. It not only involved number of days of efforts but also extensive consultation, case study, analysis, programming and much more. In fact, this project work enjoyed the creation and innovative support from the concerned guide.

I take this opportunity to express our sincere thanks to Prof. ShivaKumar.T, for providing necessary facilities and guidance or the project.

My sincere thanks to our professor in BCA Department, G.T Institute of Management studies& Research, for there valuable suggestions, spontaneous encouragement and constant help in improving the presentation of this project.

I also like to thank our Principal Dr. Prof Suvarna K, My parents and all our classmates, who have contributed directly or in directly in our efforts to complete this project successfully.

With Gratitude

**Abhishek M [181NSB7002]**

**Table of Contents**

|  |  |
| --- | --- |
| **NAME OF THE CHAPTERS** |  |
| **1. Introduction**   * 1. Project Abstract   2. Project Introduction   3. Scope of Project   4. Existing System   5. Proposed System   **2. Software Requirement And Specification(SRS)**  2.1 Software Requirement  2.2 Hardware Requirement  **3. Feasibility Study**  3.1 Operational Feasibility  3.2 Technical Feasibility  3.3 Economical Feasibility  **4. Software Development Life Cycle (SDLC)**  4.1 Waterfall Model  **5. Table Design(Data Structure)**  **6. System Design**  **7. Data Flow Diagram (DFD) And ER Diagram**  **8. Input/output Screenshots**  8.1 Index Page  8.2 Category Page  8.3 Menu Page  8.4 Contact Page  8.5 Order Page  8.6 Login Page  8.7 Dashboard Page  8.8 Manage Admin Page  8.9 Add Admin Page  8.10 Manage Category Page  8.11 Update Category Page  8.12 Manage Food Page  8.13 Update Food Page  8.14 Add Food Page  8.15 Manage Order Page  8.16 Update Order Page    **9. Sample Coding**  9.1 Index Page  9.2 Category Page  9.3 Menu Page  9.4 Contact Page  9.5 Order Page  9.6 Admin Login Page  9.7 Main Page  9.8 Manage Admin Page  9.9 Add Admin Page  9.10 Update Password  9.11 Update Admin Page  9.12 Manage Category Page  9.13 Update Category Page  9.14 Manage Food Page  9.15 Manage Order Page  **10. Testing and Results**  10.1 Software Testing  10.2 White Box Testing  10.3 Black Box Testing  10.4 System Testing  10.4.1 Code testing  10.4.2 Unit Testing  10.4.3 System Testing  10.4.4 Integration Testing  10.4.5 Acceptance Testing  10.4.6 Specification Testing  10.4.7 Performance Time Testing  **11. Implementation**  **12. Future Enhancement**  **13. References** |  |

**Chapter No: 1**

**Pizza-Hut Management System**

**PROJECT ABSTRACT**:

The purpose of Pizza-Hut Management System is to automate the existing manual systems by the help of computerized equipments and full-fledged computer software fulfilling their requirements, so basically the project describes how to manage for good performance and better services for the clients.

**INTRODUCTION**:

The “Pizza-Hut Management System” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for particular need of the company to carry out operations in a smooth and efficient manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by all this it proves it is user-friendly. Pizza-Hut Management System as described above can lead to error free and secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

**SCOPE**

In “Pizza-Hut Management System” we have designed website for both user and for management. For user we have designed website with animation and responsive website for better user experience and for management we have designed for better productivity and we can control the user website contents from admin page to keep updating and changing the food and categories. Admin website is secured with login and encrypted password.

**THE EXISTING SYSTEM**

The existing process making the order through mobile. Customer can request all this done through as manually that means though physical interaction through phone. In this System data has not maintain as centralized. Due this the management unable to get the up to date information regarding pizzas or customer request.

**PROPOSED SYSTEM**

The proposed project is a single centralized framework that would integrate various order management processes such as orders details, maintenance etc. The proposed web based [pizza](https://www.bartleby.com/topics/pizza) management system helps in streamlining and easing the process of managing pizza order. In an organized manner thereby reducing the administration costs associated with manual way of managing orders.

**Chapter No: 2**

**SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

**HARDWARE REQUIREMENTS**

|  |  |
| --- | --- |
| Processor | Intel(R) Dual Core(TM) ,more |
| RAM | 4GB |
| Hard disk | 80GB or more |
| Keyboard | Normal or Multimedia |
| Mouse | Compatible mouse |
| Monitor | 15” CRT, or LCD monitor |

**SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| Operating System | Microsoft windows 7 and above versions |
| Front End Tool | Visual Code |
| Back End Tool | MYSQL Server , PHP |
| Additional Software | XAMPP SERVER for Database |
| Browser | Google Chrome |

**Chapter No: 3**

**FEASIBILITY STUDY**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it’s worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

**3.1 OPERATIONAL FEASIBILITY**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

**3.2 TECHNICAL FEASIBILITY**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures.

**3.3 ECONOMICAL FEASIBILITY**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible.

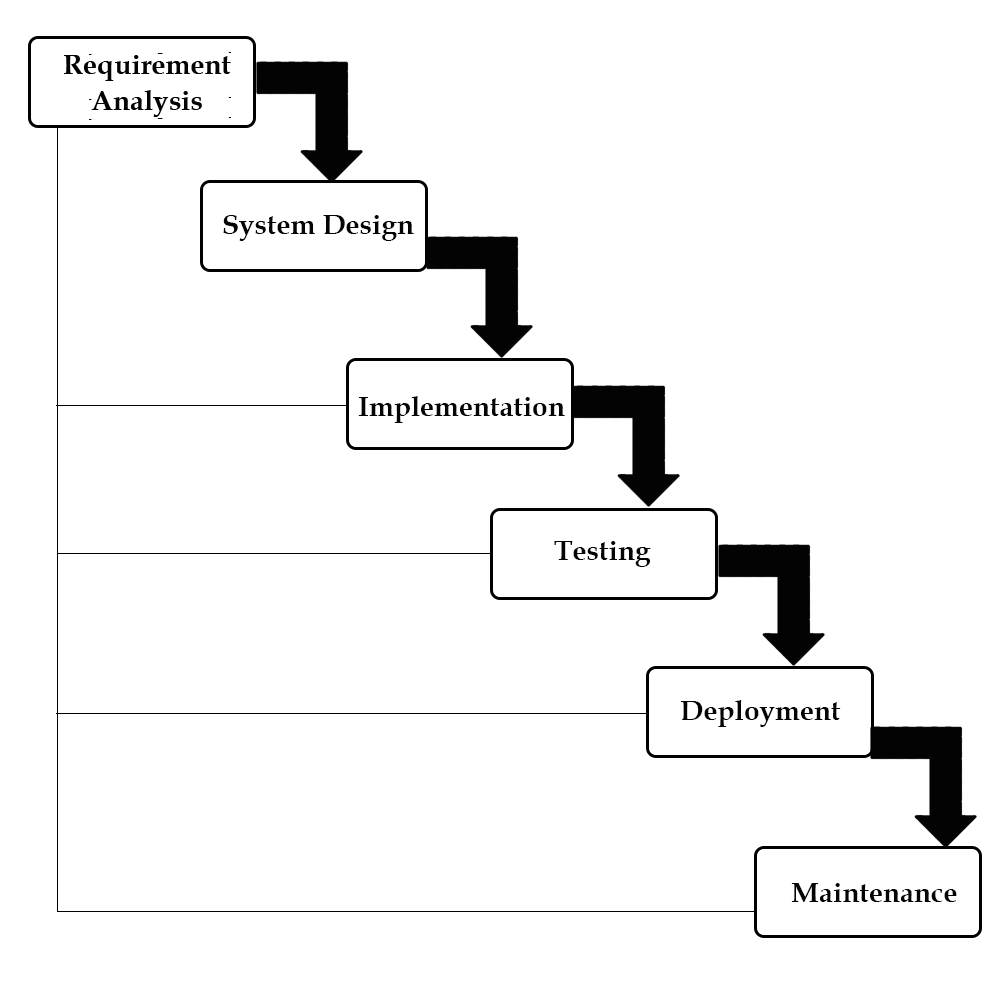
**Chapter No: 4**

**Software Development Life Cycle Model**

**4.1 WATERFALL MODEL**

The waterfall model was selected as the SDLC model due to the following reasons:

* Requirements were very well documented, clear and fixed.
* Technology was adequately understood.
* Simple and easy to understand and use.
* There were no ambiguous requirements.
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Clearly defined stages.
* Well understood milestones. Easy to arrange tasks.



**Chapter No: 5**

**Table Design (MYSQL)**

**5.1 MySQL    **

MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). It is one part of the very popular LAMP platform consisting of Linux, Apache, My SQL, and PHP. Currently My SQL is owned by Oracle. My SQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows, or Mac OS. Wikipedia and YouTube use My SQL. These sites manage millions of queries each day. My SQL comes in two versions: My SQL server system and My SQL embedded system.

RDBMS TERMINOLOGY

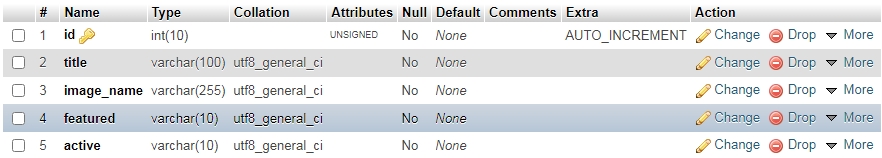
Before we proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:**A database is a collection of tables, with related data.
* **Table:**A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* **Column:**One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:**A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy:**Storing data twice, redundantly to make the system faster.
* **Primary Key:**A primary key is unique. A key value cannot occur twice in one table. With a key, you can find at most one row.
* **Foreign Key:**A foreign key is the linking pin between two tables.
* **Compound Key:**A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:**An index in a database resembles an index at the back of a book.
* **Referential Integrity:**Referential Integrity makes sure that a foreign key value always points to an existing row.

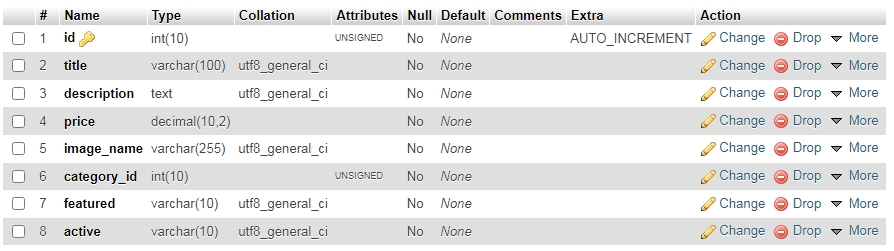
**Tbl\_admin**



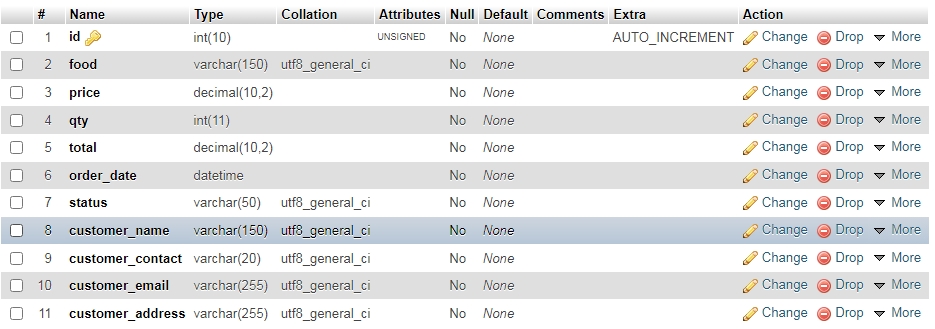
**Tbl\_category**

****

**Tbl\_food**



**Tbl\_Order**



**Chapter No: 6**

**System Design Details**

In this Section we will do Analysis of Technologies to use for implementing the project.

**6.1 : FRONT END**

**6.1.1 HTML    **

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**6.1.2 Css    **

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium.The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

**6.2 : BACK END**

**6.2.1 PHP    **

PHP is a server side scripting language that is used to develop Static websites or Dynamic websites or Web applications. PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages. PHP scripts can only be interpreted on a server that has PHP installed. The client computers accessing the PHP scripts require a web browser only. A PHP file contains PHP tags and ends with the extension ".php".

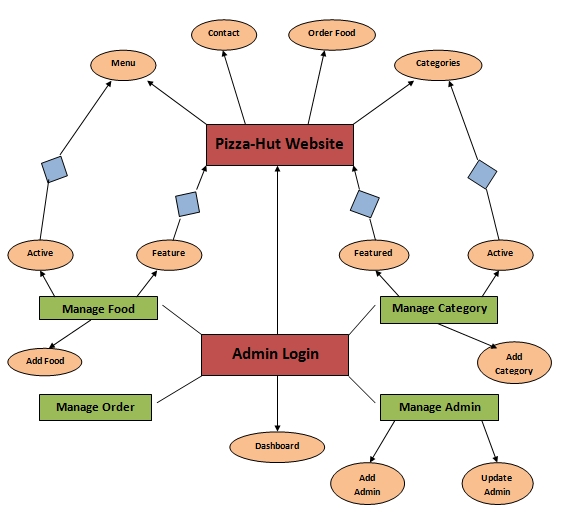
The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file. The thing that differentiates PHP with client-side language like HTML is, PHP codes are executed on the server whereas HTML codes are directly rendered on the browser.

PHP: Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994.PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data. PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control.

**Chapter No: 7**

**Data Flow Diagram (DFD) And ER Diagram**

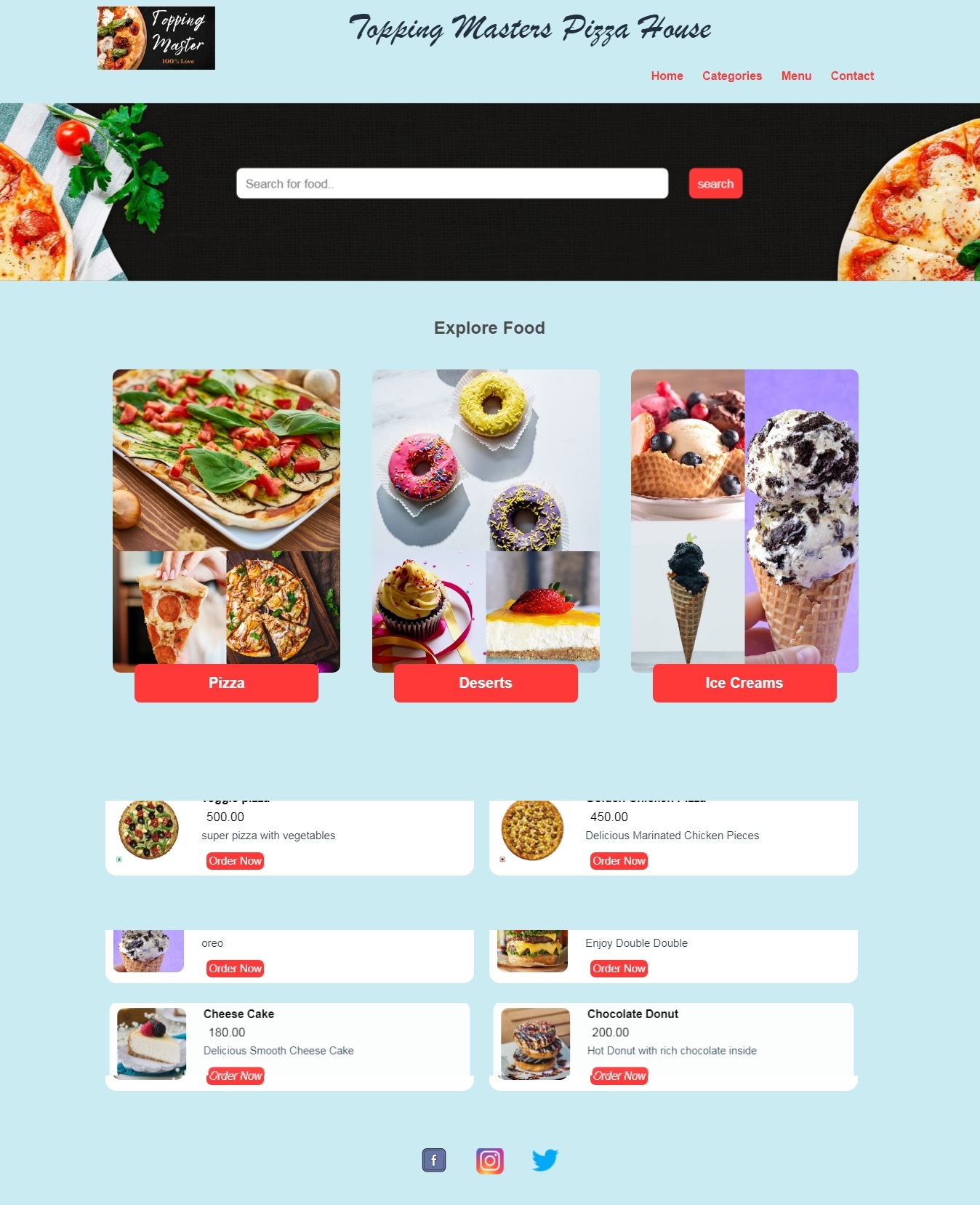
**7.1 E-R Diagram**



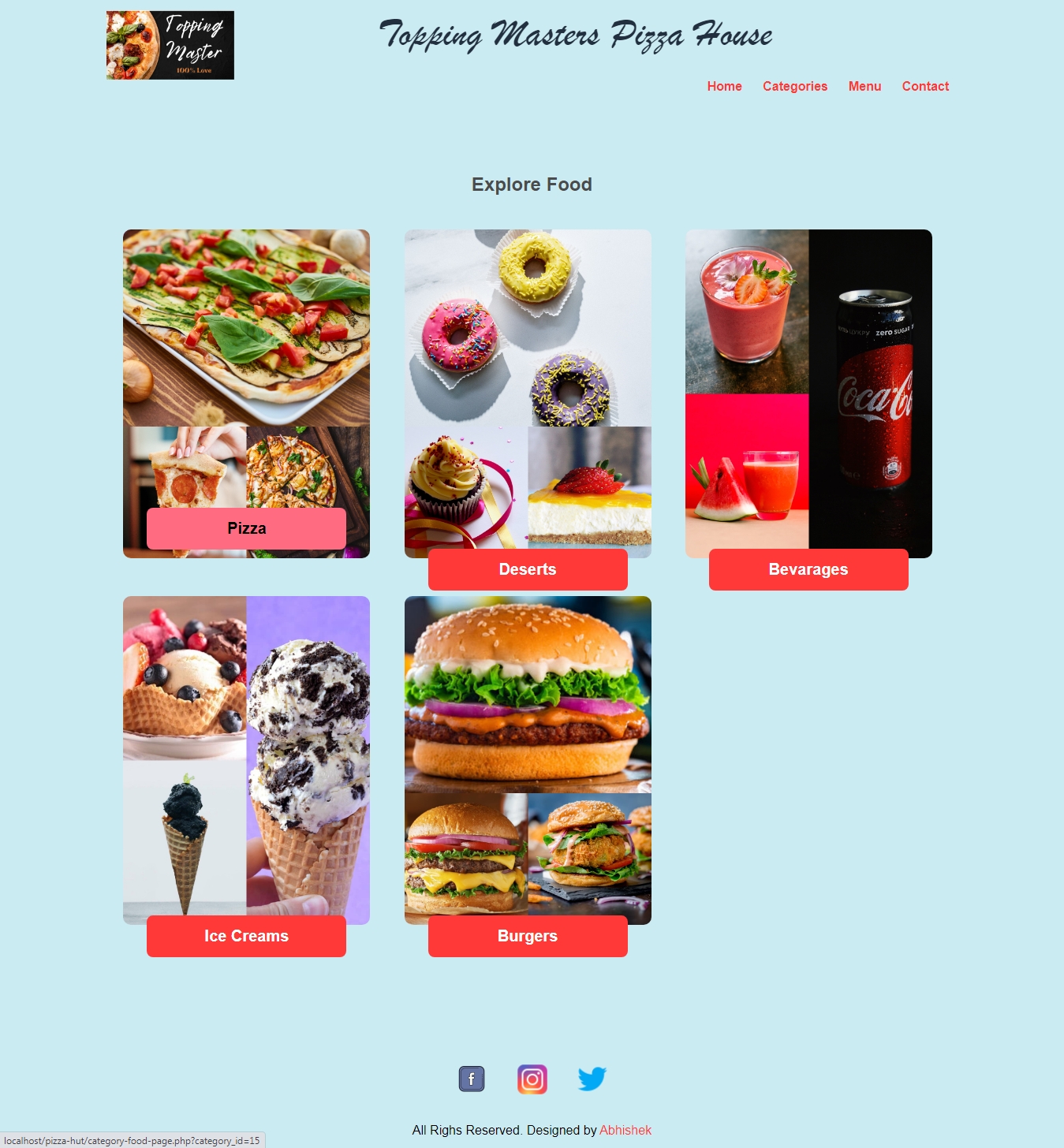
**Chapter No: 8**

**Screen Shots**

**8.1 Main Page**



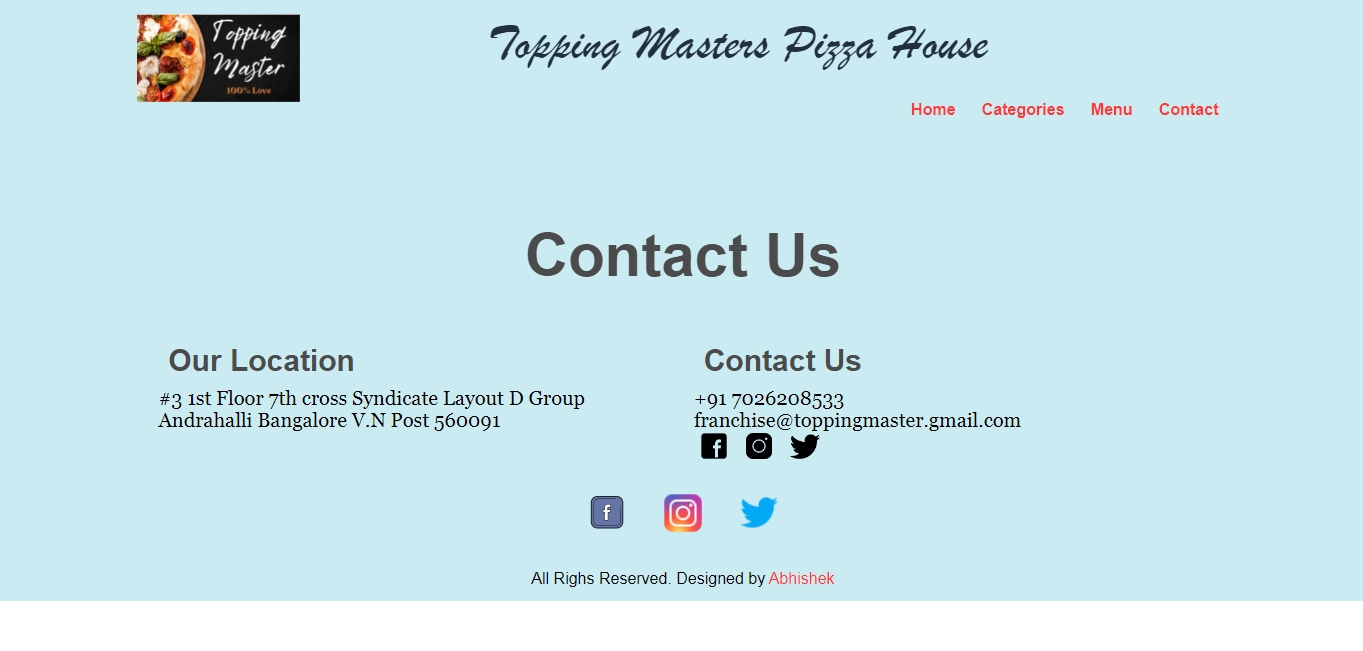
**8.2 Category Page**



**8.3 Menu Page**

****

**8.4 Contact Page**

****

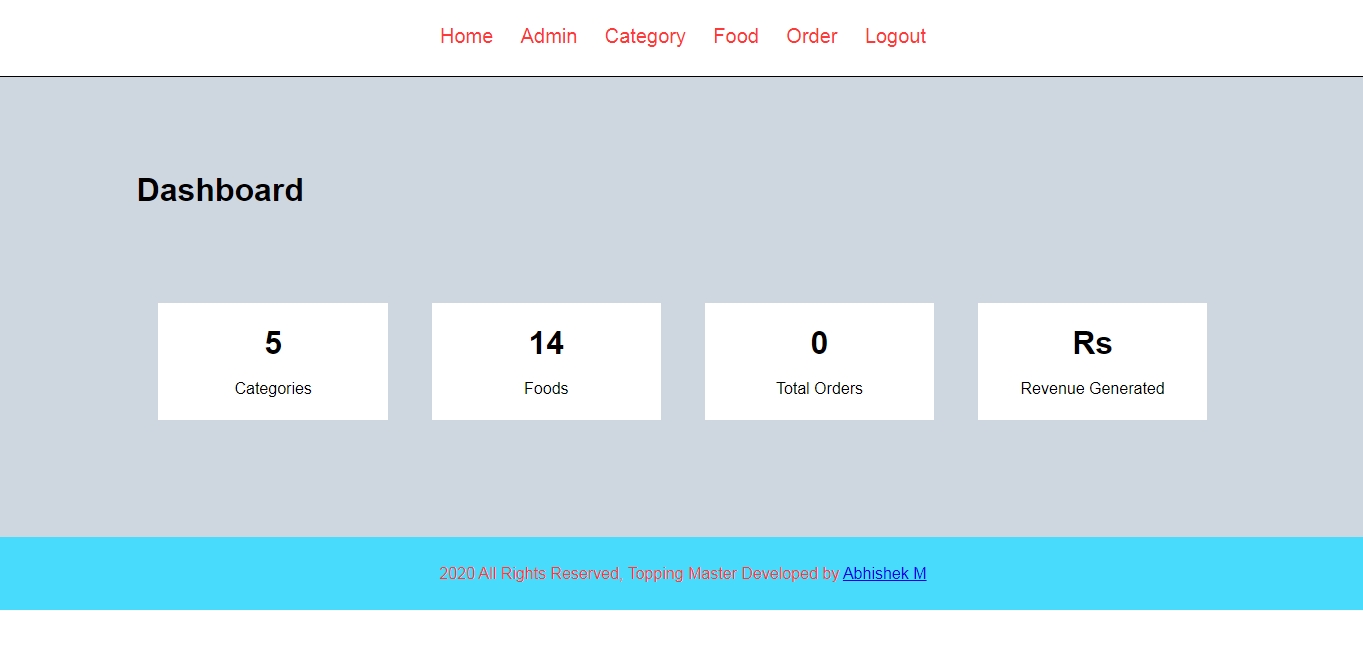
**8.5 Order Page**

****

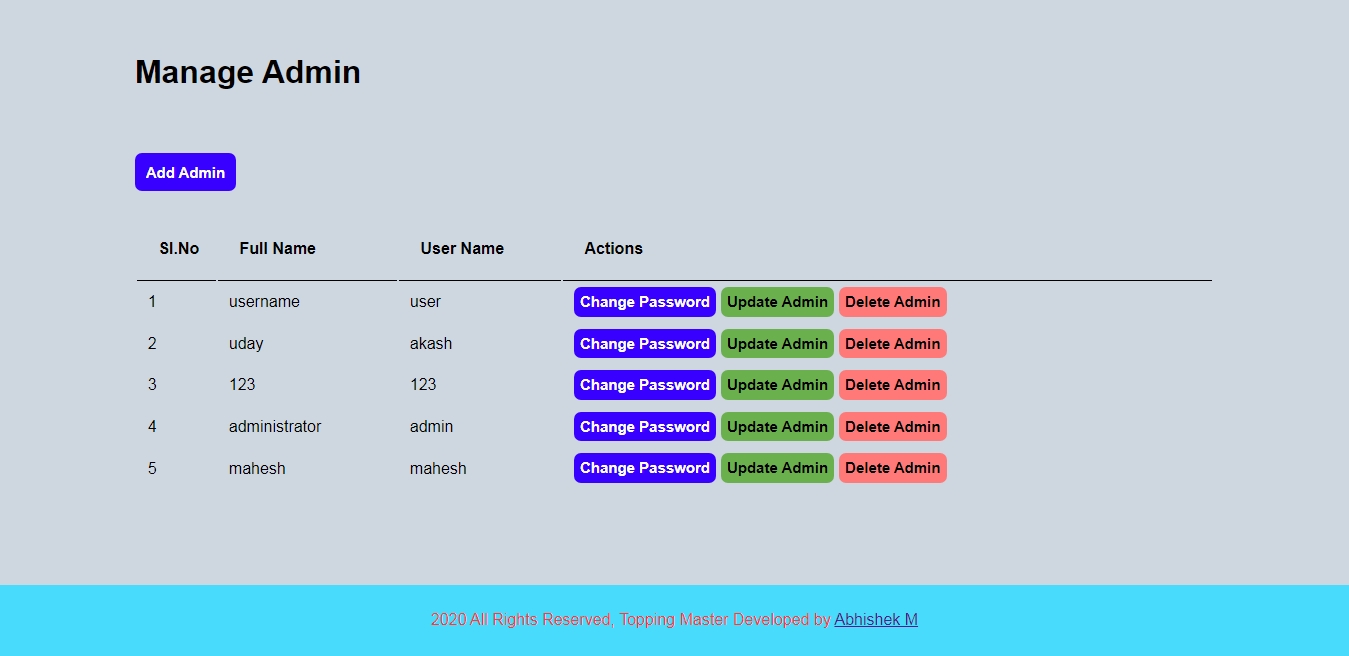
**8.6 Login Page**

****

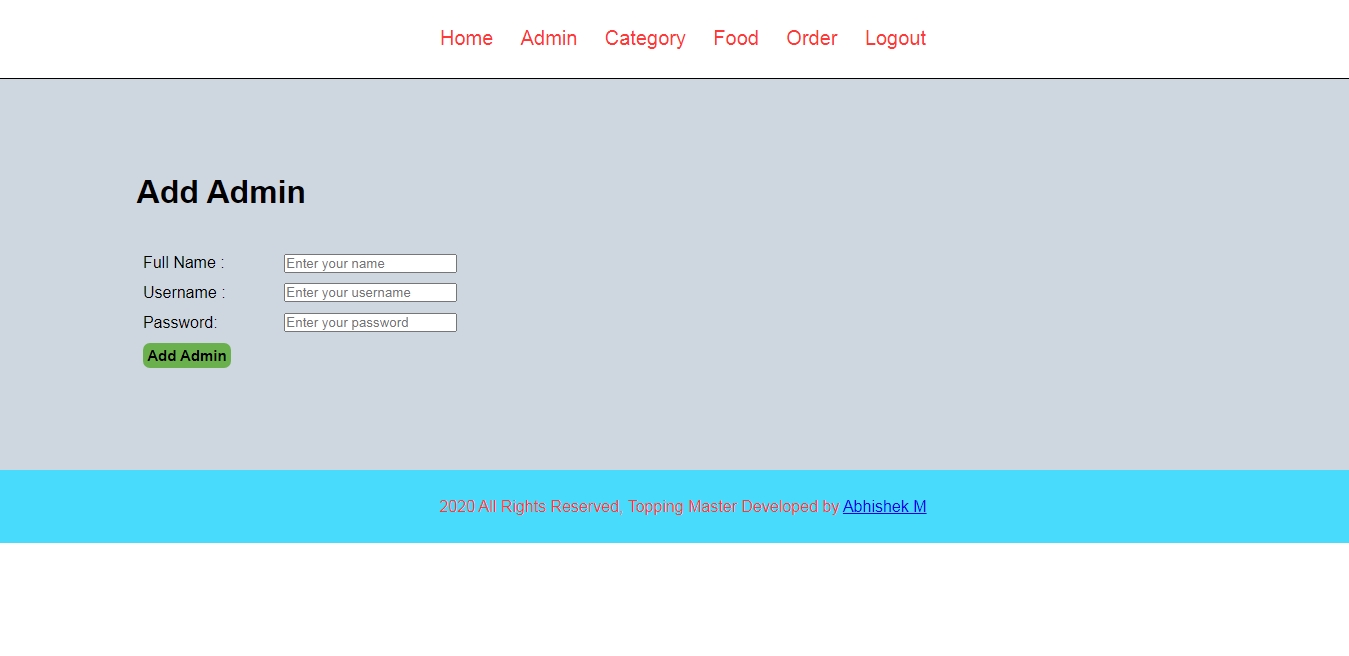
**8.7 Dashboard Page**

****

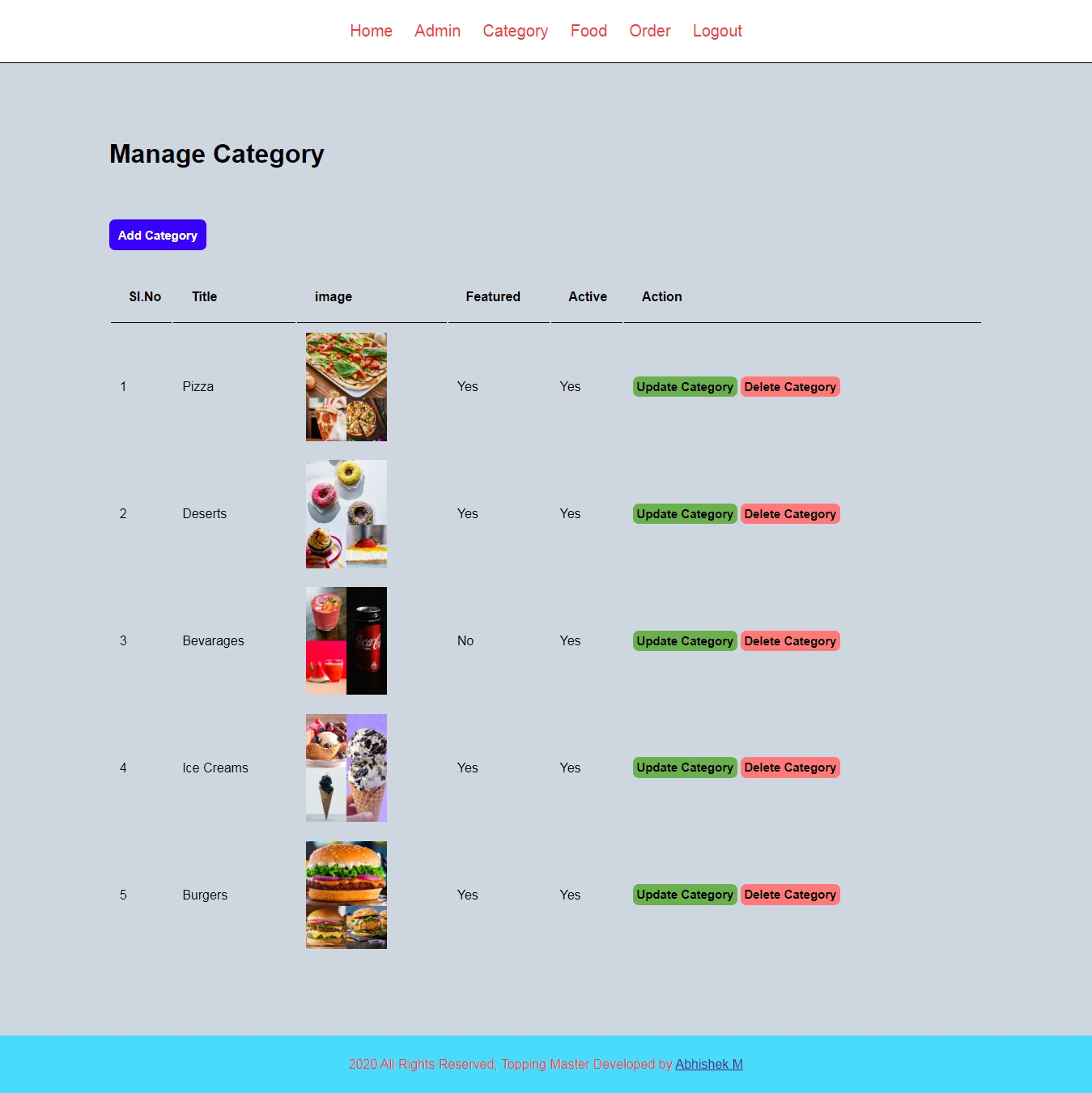
**8.8 Admin Page**

****

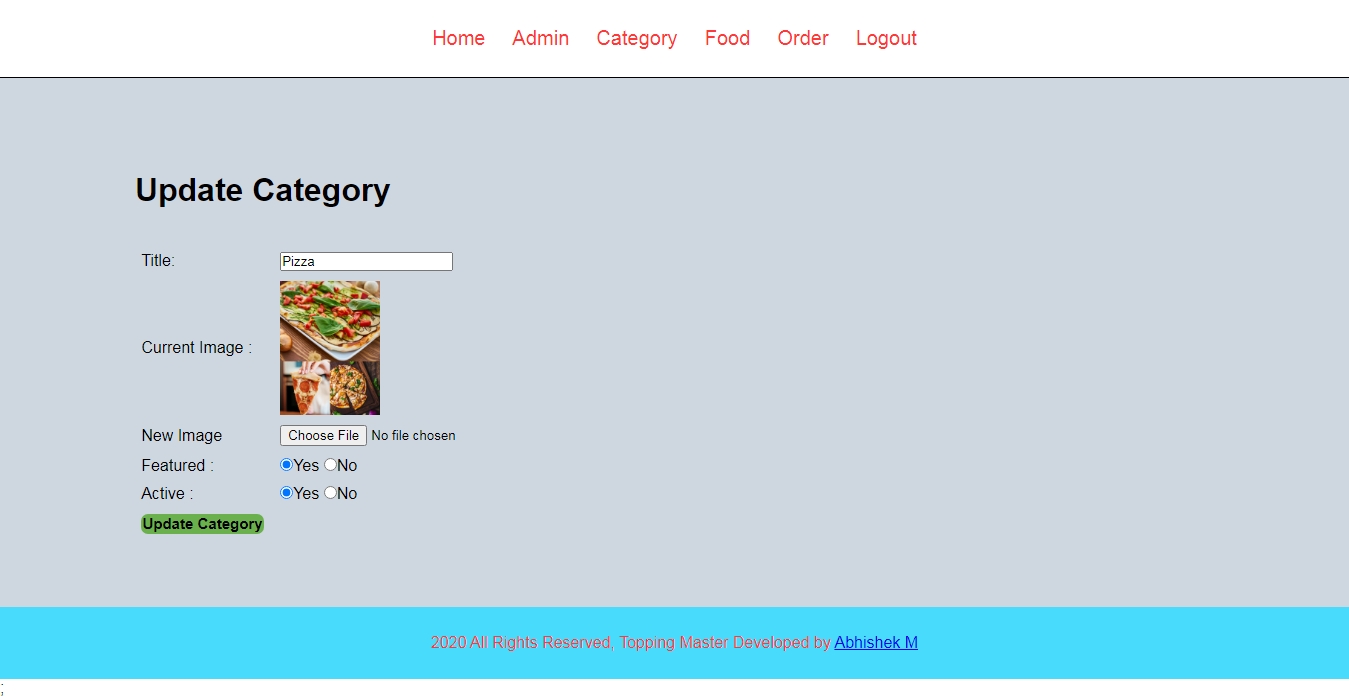
**8.9 Add Admin Page**

****

**8.10 Category Page**

****

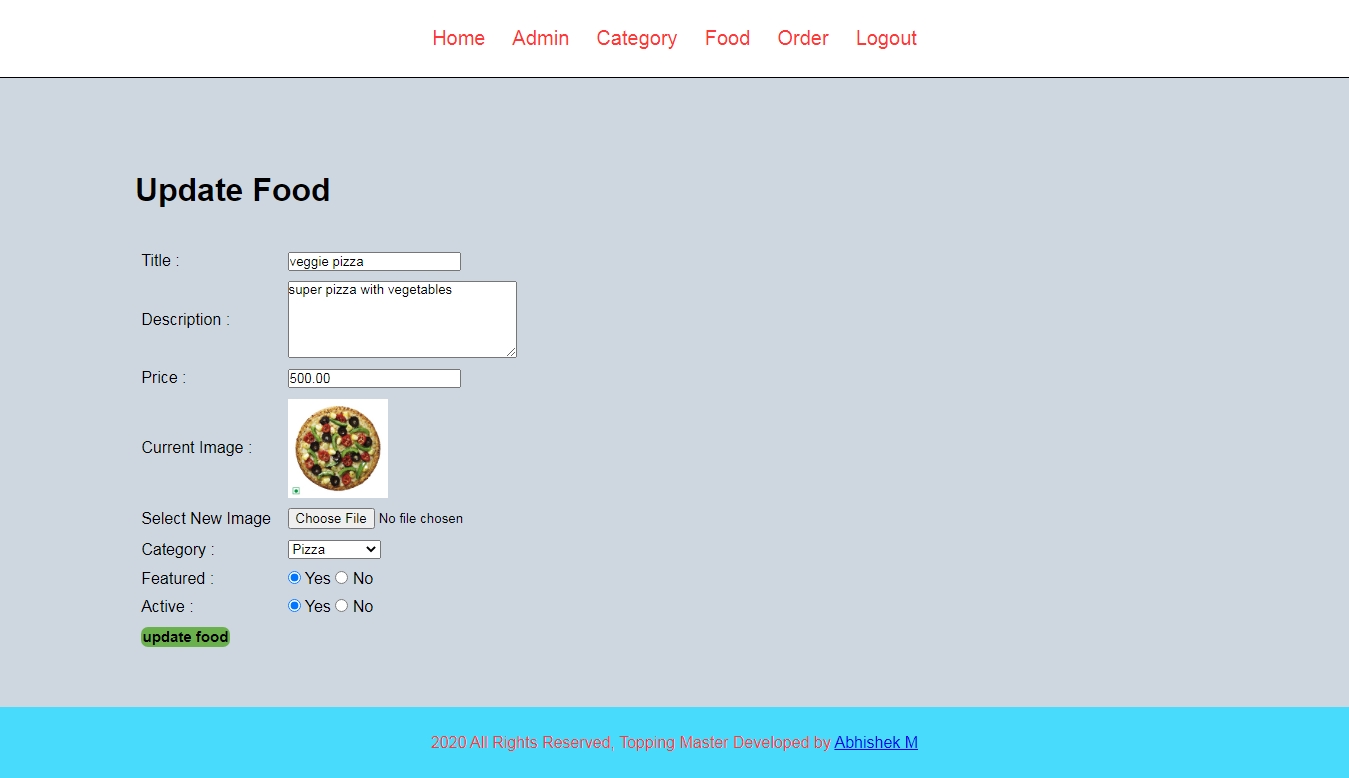
**8.11 Update Category Page**

****

**8.12 Manage Food Page**

****

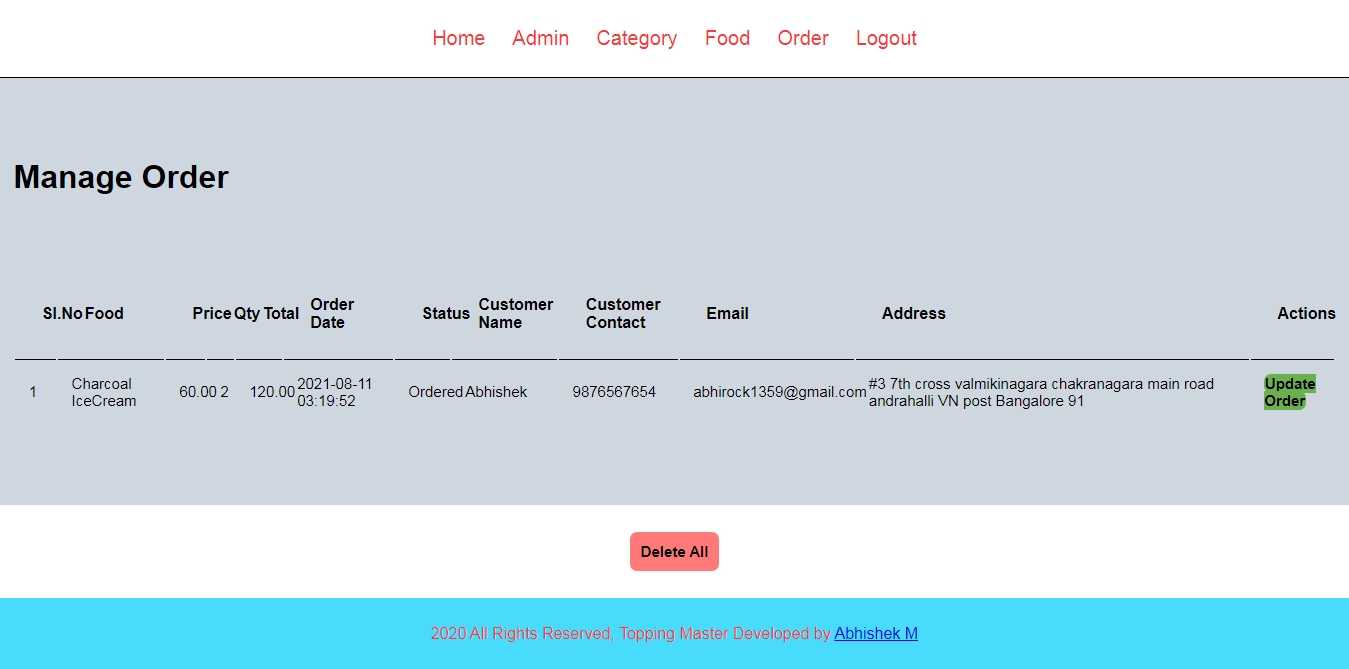
**8.13 Update Food Page**

****

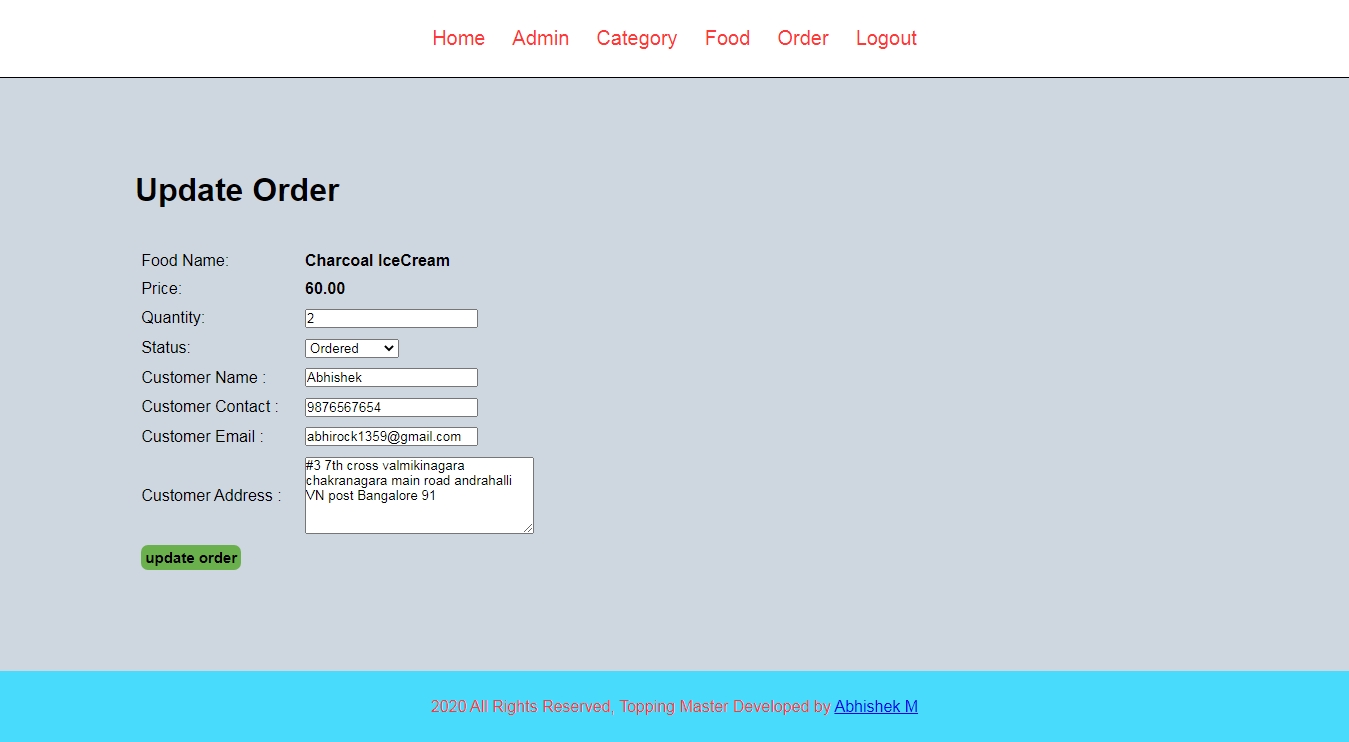
**8.14 Add Food Page**

****

**8.15 Manage Order Page**

****

**8.16 Update Order Page**

****

**Chapter No: 9**

**Sample Coding**

**9.1 Index Page**

<?php include('partials-front/menu.php'); ?>

<!--Food Search Starts here-->

<section class="food-search text-center">

<div class="container" data-aos="fade-up">

<form action="food-search-page.html" method="POST">

<input type="search" name="search" placeholder="Search for food.." required>

<input type="submit" name="submit" value="search" class="btn btn-primary">

</form>

</div>

</section>

<!--Food Search Ends here-->

<?php

if(isset($\_SESSION['order']))

{

echo $\_SESSION['order'];

unset($\_SESSION['order']);

}

?>

<!--Categories Starts here-->

<section class="categories">

<div class="container">

<h2 class="text-center" data-aos="fade-up" data-aos-duration="1000">Explore Food</h2>

<?php

// create query to display categories from database

$sql = "SELECT \* FROM tbl\_category WHERE active = 'Yes' AND featured = 'Yes' LIMIT 3";

// execute query

$res = mysqli\_query($conn, $sql);

// count rows to check whether category is available or not

$count = mysqli\_num\_rows($res);

if($count>0)

{

// category available

while($row = mysqli\_fetch\_assoc($res))

{

// get the values title, image\_name, id

$id = $row['id'];

$title = $row['title'];

$image\_name = $row['image\_name'];

?>

<a href="<?php echo SITEURL; ?>category-food-page.php?category\_id=<?php echo $id; ?>">

<div class="box-3 float-container">

<!-- check whether image is availale or not -->

<?php

if($image\_name=="")

{

// display message

echo "<div class = 'error'>Image not available</div>";

}

else

{

// image Available

?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $image\_name ?>" alt="pizza image" class="img-responsive img-curve">

<?php

}

?>

<h3 class="float-text float-text-img"> <?php echo $title; ?></h3>

</div>

</a>

<?php

}

}

else

{

// category not available

echo "<div class = 'error'>Category Not Added</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!--Categories Ends here-->

<!--Food Menu Starts here-->

<section class="food-menu">

<div class="container">

<h2 class="text-center" data-aos="fade-up">Today's Menu</h2>

<?php

// getting foods from database that are active and features

// sql query

$sql2 = "SELECT \* FROM tbl\_food WHERE active='Yes' AND featured = 'Yes' LIMIT 6";

// execute query

$res2 = mysqli\_query($conn, $sql2);

// count rows

$count2 = mysqli\_num\_rows($res2);

// check whether food available or not

if($count2>0)

{

// food available

while($row = mysqli\_fetch\_assoc($res2))

{

// get all values

$id = $row['id'];

$title = $row['title'];

$price = $row['price'];

$description = $row['description'];

$image\_name = $row['image\_name'];

?>

<div class="food-menu-box" data-aos="zoom-in">

<div class="food-menu-img">

<?php

// check whether image available or not

if($image\_name == "")

{

// image not availabel

echo "<div class='error'>Image not available</div>";

}

else

{

// image available

?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name ?>" alt="pizz image" class="img-responsive img-curve" width="100px" height="100px">

<?php

}

?>

</div>

<div class="food-menu-desc">

<h4><?php echo $title; ?></h4>

<p class="food-price"><?php echo $price; ?></p>

<p class="food-details"><?php echo $description; ?></p>

<br>

<a href="<?php echo SITEURL; ?>order-page.php?food\_id=<?php echo $id; ?>" class="btn btn-primary"> Order Now</a>

</div>

</div>

<?php

}

}

else

{

// food not available

echo "<div class='error'>Food Not Available</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!--Food Menu Ends here-->

<!-- pop section starts here -->

<div class="popup">

<div class="contentbox">

<div class="close"></div>

<div class="imgbx">

<img src="images/getvaccine1.png">

</div>

<div class="content">

<div>

<h2> Get Vaccinated </h2>

<h3> Get Covid19 Vaccine and wear mask when you are outside and maintain social distance from one another.</h3>

</div>

</div>

</div>

</div>

<!-- automatic show popup after 2 seconds -->

<script>

const popup = document.querySelector('.popup');

const close = document.querySelector('.close');

window.onload = function(){

setTimeout(function(){

popup.style.display = "block";

// add time delay

},2000)

}

close.addEventListener('click', () => {

popup.style.display = "none";

} )

</script>

<!-- pop section ends here -->

<?php include('partials-front/footer.php'); ?>

**9.2 Category Page**

<?php include('partials-front/menu.php'); ?>

<!--Categories Starts here-->

<section class="categories">

<div class="container">

<h2 class="text-center" >Explore Food</h2>

<?php

// display all category that are active

$sql = "SELECT \* FROM tbl\_category WHERE active = 'Yes'";

$res = mysqli\_query($conn, $sql);

// count rows

$count = mysqli\_num\_rows($res);

// check whether category available

if($count>0)

{

// category available

while($row = mysqli\_fetch\_assoc($res))

{

// get values

$id = $row['id'];

$title = $row['title'];

$image\_name = $row['image\_name'];

?>

<a href="<?php echo SITEURL; ?>category-food-page.php?category\_id=<?php echo $id; ?>">

<div class="box-3 float-container">

<?php

if($image\_name =="")

{

// image not available

echo "<div class='error'>Image not available</div>";

}

else

{

// image Available

?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $image\_name ?>" alt="pizza image" class="img-responsive img-curve">

<?php

}

?>

<h3 class="float-text float-text-img"> <?php echo $title; ?></h3>

</div>

</a>

<?php

}

}

else

{

// category not available

echo "<div class='error'>Category Not Found</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!--Categories Ends here-->

<?php include('partials-front/footer.php'); ?>

**9.3 Menu Page**

<?php include('partials-front/menu.php'); ?>

<!-- fOOD sEARCH Section Starts Here -->

<section class="food-search text-center">

<div class="container">

<form action="<?php echo SITEURL; ?>food-search-page.php" method="POST">

<input type="search" name="search" placeholder="Search for Food.." required>

<input type="submit" name="submit" value="Search" class="btn btn-primary">

</form>

</div>

</section>

<!-- fOOD sEARCH Section Ends Here -->

<!--Food Menu Starts here-->

<section class="food-menu">

<div class="container">

<h2 class="text-center">All Menu</h2>

<?php

// display food which are active

$sql = "SELECT \* FROM tbl\_food WHERE active='Yes'";

// execute the query

$res = mysqli\_query($conn, $sql);

// count rows

$count = mysqli\_num\_rows($res);

//check whether food available

if($count>0)

{

// food availbale

while($row = mysqli\_fetch\_assoc($res))

{

$id = $row['id'];

$title = $row['title'];

$description = $row['description'];

$price = $row['price'];

$image\_name = $row['image\_name'];

?>

<div class="food-menu-box " data-aos="zoom-in">

<div class="food-menu-img">

<?php

// check whether image is available

if($image\_name=="")

{

// image not available

echo "<div class='error'>Image not found</div>";

}

else

{

// image available

?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name; ?>" alt="pizz image" class="img-responsive img-curve" width="100px" height="100px">

<?php

}

?>

</div>

<div class="food-menu-desc">

<h4><?php echo $title; ?></h4>

<p class="food-price"><?php echo $price; ?></p>

<p class="food-details"><?php echo $description; ?></p>

<br>

<a href="<?php echo SITEURL; ?>order-page.php?food\_id=<?php echo $id; ?>" class="btn btn-primary"> Order Now</a>

</div>

</div>

<?php

}

}

else

{

// food not available

echo "<div class='error'>Food not Found</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!--Food Menu Ends here-->

<?php include('partials-front/footer.php'); ?>

**9.4 Contact Page**

<?php include('partials-front/menu.php'); ?>

<section class="categories">

<div class="container">

<h1 class="text-center"> Contact Us </h1>

<div class="contact-box">

<h2>Our Location</h2>

#3 1st Floor 7th cross Syndicate Layout D Group Andrahalli Bangalore V.N Post 560091

<br>

<br>

</div>

<div class="contact-box">

<h2>Contact Us</h2>

+91 7026208533 <br>

franchise@toppingmaster.gmail.com

<section class="social">

<ul>

<li>

<a href ="#"><img src="https://img.icons8.com/ios-filled/50/000000/facebook--v1.png" width="30px" height="30px"></a>

</li>

<li>

<a href ="#"><img src="https://img.icons8.com/ios-filled/50/000000/instagram-new.png" width="30px" height="30px"></a>

</li>

<li>

<a href ="#"><img src="https://img.icons8.com/ios-filled/50/000000/twitter.png" width="30px" height="30px"></a>

</li>

</ul>

</div>

</section>

</div>

</div>

</section>

<?php include('partials-front/footer.php'); ?>

**9.5 Order Page**

<?php

ob\_start();

?>

<?php include('partials-front/menu.php') ?>

<?php

// check whether food id is set or not

if(isset($\_GET['food\_id']))

{

// get food id and details of selected food

$food\_id = $\_GET['food\_id'];

// get the details

$sql = "SELECT \* FROM tbl\_food WHERE id=$food\_id";

// execute query

$res = mysqli\_query($conn, $sql);

// count rows

$count = mysqli\_num\_rows($res);

// check whether data is available

if($count==1)

{

// we have data

// get the data from database

$row = mysqli\_fetch\_assoc($res);

$title = $row['title'];

$price = $row['price'];

$image\_name = $row['image\_name'];

}

else

{

// Food not available

// redirect to home page

header('location:'.SITEURL);

}

}

else

{

// redirect to homepage

header('location:'.SITEURL);

}

?>

<!-- Order Section Starts Here -->

<section class = "order-background">

<div class="container">

<h2 class="text-center text-white" style="font-size: 35px;">Fill this form to confirm your order.</h2>

<form action="" method="POST" class="order" >

<fieldset>

<legend style="color: white;">Selected Food</legend>

<div class="food-menu-img food-menu-box">

<?php

// check whether the image is availbalr or not

if($image\_name == "")

{

// image not availble

echo "<div class='error'>Image not Available</div>";

}

else

{

// image is availbale

?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name; ?>" class="img-responsive img-curve" width="100px" height = "100px">

<?php

}

?>

</div>

<div class="food-menu-desc">

<h3><?php echo $title; ?></h3>

<input type="hidden" name="food" value="<?php echo $title; ?>">

<p class="food-price"><?php echo $price; ?></p>

<input type="hidden" name="price" value="<?php echo $price; ?>">

<div class="order-label">Quantity</div>

<input type="number" name="qty" class="input-responsive" value="1" required>

</div>

</fieldset>

<br>

<br>

<fieldset>

<legend>Delivery Details</legend>

<div class="order-label ">Full Name</div>

<input type="text" name="full\_name" placeholder="E.g. Abhishek" class="input-responsive" required>

<div class="order-label">Phone Number</div>

<input type="tel" name="contact" placeholder="E.g. 8846xxxxxx" class="input-responsive" required>

<div class="order-label">Email</div>

<input type="email" name="email" placeholder="E.g. hi@gmail.com" class="input-responsive" required>

<div class="order-label">Address</div>

<textarea name="address" rows="10" placeholder="E.g. Street, City, Country" class="input-responsive" required></textarea>

<input type="submit" name="submit" value="Confirm Order" class="btn btn-primary">

</fieldset>

</form>

<?php

// check whether submit button is clicked or not

if(isset($\_POST['submit']))

{

// get all the details

$food = $\_POST['food'];

$price = $\_POST['price'];

$qty = $\_POST['qty'];

$total = $price \* $qty;

$order\_date = date("Y-m-d h:i:sa"); //order date

$status = "Ordered"; //ordered, ondelivery, delivered

$customer\_name = $\_POST['full\_name'];

$customer\_contact = $\_POST['contact'];

$customer\_email = $\_POST['email'];

$customer\_address = $\_POST['address'];

// save the order in database

$sql2 = "INSERT INTO tbl\_order SET

food = '$food',

price = $price,

qty = $qty,

total = $total,

order\_date = '$order\_date',

status = '$status',

customer\_name = '$customer\_name',

customer\_contact = '$customer\_contact',

customer\_email = '$customer\_email',

customer\_address = '$customer\_address'

";

// echo $sql2;die();

// execute query

$res2 = mysqli\_query($conn, $sql2);

// check whether query executed or not

if($res2==true)

{

//Query executed and order saved

$\_SESSION['order'] = "<div class='success text-center'>Food Ordered Successfully</div>";

header('location:'.SITEURL);

}

else

{

// failed to order food

$\_SESSION['order'] = "<div class='error text-center'>Food Ordered Successfully</div>";

header('location:'.SITEURL);

ob\_enf\_fluch();

}

}

?>

</div>

</section>

<!-- Order Section Ends Here -->

<?php include('partials-front/footer.php'); ?>

**9.6 Admin Login Page**

<?php include('../config/constants.php') ?>

<html>

<head>

<title>Login - Food Order System</title>

<link rel="stylesheet" href="../css/owner.css">

</head>

<body class="bgimage">

<div class="login">

<h1 class="text-center">Login</h1>

<br><br>

<?php

if(isset($\_SESSION['login']))

{

echo $\_SESSION['login'];

unset($\_SESSION['login']);

}

?>

<?php

if(isset($\_SESSION['no-login-message']))

{

echo $\_SESSION['no-login-message'];

unset($\_SESSION['no-login-message']);

}

?>

<br><br>

<!-- Login Form starts here -->

<form action="" method="POST" class="text-center">

<p class="style">Username:</p><br>

<input type="text" name="username" placeholder="Enter Username" class="style"><br><br>

<p class="style">Password:</p><br>

<input type="password" name="password" placeholder="Enter Password" class="style" ><br><br>

<input type="submit" name="submit" value="Login" class = "btn btn-primary"><br><br>

</form>

<!-- Login Form ends here -->

<p class="text-center">Created by <a href="#">Abhishek</a></p>

</div>

</body>

</html>

<?php

// check whether the submit button is clicked or not

if(isset($\_POST['submit']))

{

// procced for Login

// 1. Get the data from Login Form

$username=$\_POST['username'];

$password=md5($\_POST['password']);

// 2. SQL to check whether the username and password exists or not

$sql="SELECT \* FROM tbl\_admin WHERE username='$username' AND password='$password'";

// 3. Execute Query

$res = mysqli\_query($conn, $sql);

// 4. count rows to check whether the user exists or not

$count = mysqli\_num\_rows($res);

if($count==1)

{

// user availbale

$\_SESSION['login'] = "<div class = 'success'>Login success.</div> ";

$\_SESSION['user'] = $username;//To check whether the user is logedin or not and logout will use it

// Redirect to Manage-admin Page

header('location:'.SITEURL.'admin/main.php' );

}

else

{

// User not available

$\_SESSION['login'] = "<div class = 'error text-center'>Username Or Password is Wrong.</div> ";

// Redirect to Manage-admin Page

header('location:'.SITEURL.'admin/login.php' );

}

}

?>

**9.7 Main Page**

<?php include('partial/menu.php');?>

<!-- Main Content Starts -->

<div class="main-content">

<div class="wrapper">

<h1>Dashboard</h1>

<br><br>

<?php

if(isset($\_SESSION['login']))

{

echo $\_SESSION['login'];

unset($\_SESSION['login']);

}

?>

<br><br>

<div class="col-4 text-center">

<?php

$sql = "SELECT \* FROM tbl\_category";

// execute query

$res = mysqli\_query($conn, $sql);

// count rows

$count = mysqli\_num\_rows($res);

?>

<h1><?php echo $count; ?></h1>

<br>

Categories

</div>

<div class="col-4 text-center">

<?php

$sql2 = "SELECT \* FROM tbl\_food";

// execute query

$res2 = mysqli\_query($conn, $sql2);

// count rows

$count2 = mysqli\_num\_rows($res2);

?>

<h1><?php echo $count2; ?></h1>

<br>

Foods

</div>

<div class="col-4 text-center">

<?php

$sql3 = "SELECT \* FROM tbl\_order";

// execute query

$res3 = mysqli\_query($conn, $sql3);

// count rows

$count3 = mysqli\_num\_rows($res3);

?>

<h1><?php echo $count3; ?></h1>

<br>

Total Orders

</div>

<div class="col-4 text-center">

<?php

// create sql for query to get total revenue

// aggregate function in sql

$sql4 = "SELECT SUM(total) AS Total FROM tbl\_order WHERE status='Delivered'";

// execute the query

$res4 = mysqli\_query($conn, $sql4);

// get the values

$row4 = mysqli\_fetch\_assoc($res4);

// get the total revenue

$total\_revenue = $row4['Total'];

?>

<h1> Rs <?php echo $total\_revenue; ?></h1>

<br>

Revenue Generated

</div>

<div class="clear-fix"></div>

</div>

</div>

<!-- Main Content Ends -->

<?php include('partial/footer.php');?>

**9.8 Manage Admin Page**

<?php include('partial/menu.php'); ?>

<div class="main-content">

<div class="wrapper">

<h1>Update Admin</h1>

<br><br>

<?php

// 1. Get the id of selelcted admin

$id=$\_GET['id'];

// 2. create tyhe sql query to get the details

$sql="SELECT \* FROM tbl\_admin WHERE id=$id";

// Ecxecute query

$res=mysqli\_query($conn, $sql);

// check wheather the query is executed or not

if($res==true)

{

// check whether the data is available or not

$count=mysqli\_num\_rows($res);

// check whether we have admin data or not

if($count==1)

{

// get details

// echo "admin available"

$row=mysqli\_fetch\_assoc($res);

$full\_name=$row['full\_name'];

$username=$row['username'];

}

else

{

// Redirect to manage admin page

header('location:'.SITEURL.'admin/managae-admin.php');

}

}

?>

<form action="" method="POST">

<table class="tbl-30">

<tr>

<td>Full Name: </td>

<td>

<input type="text" name="full\_name" value="<?php echo $full\_name; ?>">

</td>

</tr>

<tr>

<td>Username: </td>

<td>

<input type="text" name="username" value="<?php echo $username; ?>">

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="id" value="<?php echo $id ?>">

<input type="submit" name="submit" value="Update Admin" class="btn btn-secondary">

</td>

</tr>

</table>

</div>

</div>

<?php

// check wheather the submit button is clicked or not

if(isset($\_POST['submit']))

{

//echo "button clicked";

// get all the alues from form to update

$id = $\_POST['id'];

$full\_name = $\_POST['full\_name'];

$username = $\_POST['username'];

// create a sql query to object admin

$sql="UPDATE tbl\_admin SET

full\_name = '$full\_name',

username = '$username'

WHERE id='$id'

";

// Execute the query

$res = mysqli\_query($conn, $sql);

// check whether the query is executed or not

if($res == true)

{

// Query executed and admin updated

$\_SESSION['update'] = "<div class = 'success'> Admin Updated successfully</div>";

// Redirect to manage admin page

header('location:'.SITEURL.'admin/manage-admin.php');

}

else

{

// failed to update

$\_SESSION['update'] = "<div class = 'error'> Failed to update Admin</div>";

// Redirect to manage admin page

header('location:'.SITEURL.'admin/manage-admin.php');

}

}

?>

<?php include('partial/footer.php'); ?>

**9.9 Add Admin Page**

<?php include('partial/menu.php');?>

<div class="main-content">

<div class="wrapper">

<h1> Add Admin </h1>

<br><br>

<?php

if(isset($\_SESSION['add'])) //checking whether the session is set or not

{

echo $\_SESSION['add']; //displaying the session message

unset($\_SESSION['add']); // Removing session message

}

?>

<form action="" method="POST">

<table class="tbl-30">

<tr>

<td>Full Name : </td>

<td><input type="text" name="full\_name" placeholder="Enter your name"></td>

</tr>

<tr>

<td>Username : </td>

<td><input type="text" name="username" placeholder="Enter your username"></td>

</tr>

<tr>

<td>Password: </td>

<td><input type="password" name="password" placeholder="Enter your password"></td>

</tr>

<tr>

<td colspan="2">

<input type="submit" name="submit" value="Add Admin" class="btn btn-secondary">

</td>

</tr>

</table>

</form>

</div>

</div>

<?php include('partial/footer.php');?>

<!-- for saving in database -->

<?php

// Process the value from form and save in database

// check wheather the button is clicked or not

if(isset($\_POST['submit']))

{

// button clicked

// echo "button clicked";

//1. get data fro form

$full\_name = $\_POST ['full\_name'];

$username = $\_POST['username'];

$password = md5($\_POST['password']); //encrypting password

//2. sql query to save the data to database

$sql = "INSERT INTO tbl\_admin SET

full\_name = '$full\_name',

username= '$username',

password= '$password'

";

//3. executing query and saving data in database

$res = mysqli\_query($conn, $sql) or die(mysqli\_error());

// 4. Check wheather the data is inserted or not and display appropriate message

if($res==TRUE)

{

// data inserted

//echo "data inserted";

// creating session variable to display message

$\_SESSION['add']="<div class='success'>Admin Added Successfully</div>";

// redirecting page to manage admin page

header("location:".SITEURL. 'admin/manage-admin.php');

}

else

{

// data not inserted

//echo "data not inserted";

$\_SESSION['add']="Failed to Add Admin";

// redirecting page to manage admin page

header("location:".SITEURL. 'admin/add-admin.php');

}

}

?>

**9.10 Update password**

<?php include('partial/menu.php');?>

<div class="main-content">

<div class="wrapper">

<h1>Change Password</h1>

<br><br>

<?php

if(isset($\_GET['id']))

{

$id=$\_GET['id'];

}

?>

<form action="" method="POST">

<table class = "tbl-30">

<tr>

<td>Old Password: </td>

<td>

<input type="password" name = "current\_password" placeholder="current pasword">

</td>

</tr>

<tr>

<td>New Password:</td>

<td>

<input type="password" name="new\_password"placeholder="new password">

</td>

</tr>

<tr>

<td>Confirm Password:</td>

<td>

<input type="password" name="confirm\_password" placeholder="confirm password">

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="id" value="<?php echo $id ?>">

<input type="submit" name="submit" value="Change Password" class="btn btn-secondary">

</td>

</tr>

</table>

</form>

</div>

</div>

<?php

// check wheather the submit button is clicked or not

if(isset($\_POST['submit']))

{

// echo "clicked";

// 1. Get the data from from

$id=$\_POST['id'];

$current\_password = md5($\_POST['current\_password']);

$new\_password = md5($\_POST['new\_password']);

$confirm\_password = md5($\_POST['confirm\_password']);

// 2. Check whether the user with current ID and current password exists or not

$sql = "SELECT \* FROM tbl\_admin WHERE id=$id AND password='$current\_password'";

// execute the query

$res = mysqli\_query($conn, $sql);

if($res==true)

{

// check whether data is availbale or not

$count=mysqli\_num\_rows($res);

if($count==1)

{

// user exists and password can be changed

// echo "User Found";

// chech wheather the new password and confirm password match or not

if($new\_password == $confirm\_password)

{

// update the password

$sql2 = "UPDATE tbl\_admin SET

password='$new\_password'

WHERE id=$id

";

// Execute the query

$res2 = mysqli\_query($conn, $sql2);

// check whether query is executed or not

if($res2==true)

{

// display success message

// redirect to manage admin page with success message

$\_SESSION['change-pwd'] = "<div class= 'success'>Password changed successfully. </div>";

header('location:'.SITEURL.'admin/manage-admin.php');

}

else

{

// display error message

// redirect to manage admin page with failed message

$\_SESSION['change-pwd'] = "<div class= 'error'>Failed to change Password. </div>";

header('location:'.SITEURL.'admin/manage-admin.php');

}

}

else

{

// redirect to manage admin page with error

$\_SESSION['pwd-not-match'] = "<div class= 'error'>pwd did not match. </div>";

header('location:'.SITEURL.'admin/manage-admin.php');

}

}

else

{

// User does not exists set message and redirect

$\_SESSION['user-not-found'] = "<div class= 'error'>User Not Found. </div>";

header('location:'.SITEURL.'admin/manage-admin.php');

}

}

// 3. Check whether the new password and confirm password match or not

// 4. Change Password if all above is true

}

?>

<?php include('partial/footer.php');?>

**9.11 Update Admin Page**

<?php include('partial/menu.php'); ?>

<div class="main-content">

<div class="wrapper">

<h1>Update Admin</h1>

<br><br>

<?php

// 1. Get the id of selelcted admin

$id=$\_GET['id'];

// 2. create tyhe sql query to get the details

$sql="SELECT \* FROM tbl\_admin WHERE id=$id";

// Ecxecute query

$res=mysqli\_query($conn, $sql);

// check wheather the query is executed or not

if($res==true)

{

// check whether the data is available or not

$count=mysqli\_num\_rows($res);

// check whether we have admin data or not

if($count==1)

{

// get details

// echo "admin available"

$row=mysqli\_fetch\_assoc($res);

$full\_name=$row['full\_name'];

$username=$row['username'];

}

else

{

// Redirect to manage admin page

header('location:'.SITEURL.'admin/managae-admin.php');

}

}

?>

<form action="" method="POST">

<table class="tbl-30">

<tr>

<td>Full Name: </td>

<td>

<input type="text" name="full\_name" value="<?php echo $full\_name; ?>">

</td>

</tr>

<tr>

<td>Username: </td>

<td>

<input type="text" name="username" value="<?php echo $username; ?>">

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="id" value="<?php echo $id ?>">

<input type="submit" name="submit" value="Update Admin" class="btn btn-secondary">

</td>

</tr>

</table>

</div>

</div>

<?php

// check wheather the submit button is clicked or not

if(isset($\_POST['submit']))

{

//echo "button clicked";

// get all the alues from form to update

$id = $\_POST['id'];

$full\_name = $\_POST['full\_name'];

$username = $\_POST['username'];

// create a sql query to object admin

$sql="UPDATE tbl\_admin SET

full\_name = '$full\_name',

username = '$username'

WHERE id='$id'

";

// Execute the query

$res = mysqli\_query($conn, $sql);

// check whether the query is executed or not

if($res == true)

{

// Query executed and admin updated

$\_SESSION['update'] = "<div class = 'success'> Admin Updated successfully</div>";

// Redirect to manage admin page

header('location:'.SITEURL.'admin/manage-admin.php');

}

else

{

// failed to update

$\_SESSION['update'] = "<div class = 'error'> Failed to update Admin</div>";

// Redirect to manage admin page

header('location:'.SITEURL.'admin/manage-admin.php');

}

}

?>

<?php include('partial/footer.php'); ?>

**9.12 Manage Category Page**

<?php include('partial/menu.php');?>

<!-- Main Content Starts -->

<div class="main-content">

<div class="wrapper">

<h1>Manage Category</h1>

<br><br>

<?php

if(isset($\_SESSION['add']))

{

echo $\_SESSION['add'];

unset($\_SESSION['add']);

}

if(isset($\_SESSION['remove']))

{

echo $\_SESSION['remove'];

unset($\_SESSION['remove']);

}

if(isset($\_SESSION['delete']))

{

echo $\_SESSION['delete'];

unset($\_SESSION['delete']);

}

if(isset($\_SESSION['no-category-found']))

{

echo $\_SESSION['no-category-found'];

unset($\_SESSION['no-category-found']);

}

if(isset($\_SESSION['update']))

{

echo $\_SESSION['update'];

unset($\_SESSION['update']);

}

if(isset($\_SESSION['upload']))

{

echo $\_SESSION['upload'];

unset($\_SESSION['upload']);

}

if(isset($\_SESSION['failed-remove']))

{

echo $\_SESSION['failed-remove'];

unset($\_SESSION['failed-remove']);

}

?>

<br><br>

<!-- button to add admin -->

<a href="<?php echo SITEURL; ?>admin/add-category.php" class="btn btn-primary">Add Category</a>

<br><br><br>

<table class="tbl-full">

<tr>

<th> Sl.No</th>

<th>Title</th>

<th>image</th>

<th>Featured</th>

<th>Active</th>

<th>Action</th>

</tr>

<?php

// Query to get all category

$sql="SELECT \* FROM tbl\_category";

// execute query

$res=mysqli\_query($conn, $sql);

// count rows

$count = mysqli\_num\_rows($res);

// Create serial number variable

$sn=1;

// check whether we have database or not

if($count>0)

{

// we have data in database

// get the data and display

while($row = mysqli\_fetch\_assoc($res))

{

$id = $row['id'];

$title = $row['title'];

$image\_name = $row['image\_name'];

$featured = $row['featured'];

$active = $row['active'];

?>

<tr>

<td><?php echo $sn++; ?></td>

<td><?php echo $title; ?></td>

<td>

<?php

// check whether the image name is avilbale or not

if($image\_name!="")

{

// display image

?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $image\_name; ?>" width="100px">

<?php

}

else

{

// display error message

echo "<div class='error'>Image not Found</div>";

}

?>

</td>

<td><?php echo $featured; ?></td>

<td><?php echo $active; ?></td>

<td>

<a href="<?php echo SITEURL; ?>admin/update-category.php?id=<?php echo $id; ?>" class="btn btn-secondary">Update Category</a>

<a href="<?php echo SITEURL; ?>admin/delete-category.php?id=<?php echo $id; ?>&image\_name=<?php echo $image\_name; ?>" class="btn btn-danger">Delete Category</a>

</td>

</tr>

<?php

}

}

else

{

// no data in database

// display no data

?>

<tr>

<td colspan="6"><div class="error">No Category Added</div></td>

</tr>

<?php

}

?>

</table>

</div>

</div>

<!-- Main Content Ends -->

<?php include('partial/footer.php');?>

**9.13 Update Category Page**

<?php include('partial/menu.php');?>

<div class="main-content">

<div class="wrapper">

<h1>Update Category</h1>

<br><br>

<?php

// check whether the id is set or not

if(isset($\_GET['id']))

{

// get id and all details

$id = $\_GET['id'];

// crete sql id to get all details

$sql = "SELECT \* FROM tbl\_category WHERE id=$id";

// Execute the query

$res = mysqli\_query($conn,$sql);

// count the rows to check whether the id is valid or not

$count = mysqli\_num\_rows($res);

if($count==1)

{

// get all data

$row = mysqli\_fetch\_assoc($res);

$title = $row['title'];

$current\_image = $row['image\_name'];

$featured = $row['featured'];

$active = $row['active'];

}

else

{

// redirect to manage categoey page with message'

$\_SESSION['no-category-found'] = "<div class='error'>Category Not Found</div>";

header('location:'.SITEURL.'admin/manage-category.php');

}

}

else

{

// redirect to manage category

header('location:'.SITEURL.'admin/manage-category.php');

}

?>

<form action="" method="POST" enctype="multipart/form-data">

<table class="tbl-30">

<tr>

<td>Title:</td>

<td>

<input type="text" name="title" value="<?php echo $title ?>" >

</td>

</tr>

<tr>

<td>Current Image : </td>

<td>

<?php

if($current\_image !="")

{

// Display the image

?>

<img src="<?php echo SITEURL;?>images/category/<?php echo $current\_image;?> " width="100px" >

<?php

}

else

{

// display message

echo "<div class='error'>Image not added</div>";

}

?>

</td>

</tr>

<tr>

<td>New Image</td>

<td>

<input type="file" name="image" id="">

</td>

</tr>

<tr>

<td>Featured : </td>

<td>

<input <?php if($featured=="Yes"){echo "checked";} ?> type="radio" name="featured" value="Yes">Yes

<input <?php if($featured=="No"){echo "checked";} ?> type="radio" name="featured" value="No">No

</td>

</tr>

<tr>

<td>Active : </td>

<td>

<input <?php if($active=="Yes"){echo "checked";} ?> type="radio" name="active" value="Yes">Yes

<input <?php if($active=="No"){echo "checked";} ?> type="radio" name="active" value="No">No

</td>

</tr>

<tr>

<td>

<input type="hidden" name="current\_image" value="<?php echo $current\_image; ?>">

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Update Category" class="btn btn-secondary">

</td>

</tr>

</table>

</form>

<?php

// check if submit is pressed

if(isset($\_POST['submit']))

{

// echo "clicked";

// 1. Get all the values from form

$id = $\_POST['id'];

$title=$\_POST['title'];

$current\_image = $\_POST['current\_image'];

$featured = $\_POST['featured'];

$active = $\_POST['active'];

// 2. Updating new image if selected

// check whether the image is selected or not

if(isset($\_FILES['image']['name']))

{

// get the image details

$image\_name = $\_FILES['image']['name'];

// check whether the image is availbale are not

if($image\_name !="")

{

// image available

// A. upload the new image

// Auto Rename our image

// get the extension of our image(jpg, png etc)

$ext = end(explode('.',$image\_name));

// Rename the image

$image\_name="Food\_Category\_".rand(000, 999).'.'.$ext;

$source\_path=$\_FILES['image']['tmp\_name'];

$destination\_path="../images/category/".$image\_name;

// finally upload the image

$upload = move\_uploaded\_file($source\_path , $destination\_path);

// check whether the image is uploaded or not

// and if image is not uploded then we will stop process and redirect with error message

if($upload==false)

{

// Set Message

$\_SESSION['upload'] = "<div class='error'>Failed to upload Image.<div>";

// redirect to add category page

header('location:'.SITEURL.'admin/manage-category.php');

// Stop Process

die();

}

//B. Remove the current image if available

if($current\_image!="")

{

$remove\_path = "../images/category/".$current\_image;

$remove = unlink($remove\_path);

// Check whether the image is removed or not

// if fialed to remove then display message and stop process

if($remove==false)

{

// failed to remove image

$\_SESSION['failed-remove'] = "<div class='error'>Failed to remove current image</div>";

header('location:'.SITEURL.'admin/manage-ategory.php');

}

}

}

else

{

$image\_name = $current\_image;

}

}

else

{

$image\_name = $current\_image;

}

// 3. Update to database

$sql2 = "UPDATE tbl\_category SET

title = '$title',

image\_name='$image\_name',

featured = '$featured',

active = '$active'

WHERE id = $id

";

// Execute the query

$res2= mysqli\_query($conn, $sql2);

// 4. Redirect to manage category page

// Check wether query executed or not

if($res2==true)

{

// category updated

$\_SESSION['update'] = "<div class='success'>Category Updated Successfully</div>";

header('location:'.SITEURL.'admin/manage-category.php');

}

else

{

// failed to update category

$\_SESSION['update'] = "<div class='error'> Failed to update Category</div>";

header('location:'.SITEURL.'admin/manage-category.php');

}

}

?>

</div>

</div>

<?php include('partial/footer.php')?>;

**9.14 Manage Food**

<?php include('partial/menu.php');?>

<!-- Main Content Starts -->

<div class="main-content">

<div class="wrapper">

<h1>Manage Food</h1>

<br><br>

<?php

if(isset($\_SESSION['add']))

{

echo $\_SESSION['add'];

unset($\_SESSION['add']);

}

if(isset($\_SESSION['delete']))

{

echo $\_SESSION['delete'];

unset($\_SESSION['delete']);

}

if(isset($\_SESSION['upload']))

{

echo $\_SESSION['upload'];

unset($\_SESSION['upload']);

}

if(isset($\_SESSION['unauthorize']))

{

echo $\_SESSION['unauthorize'];

unset($\_SESSION['unauthorize']);

}

if(isset($\_SESSION['update']))

{

echo $\_SESSION['update'];

unset($\_SESSION['update']);

}

?>

<br><br>

<!-- button to add admin -->

<a href="<?php echo SITEURL; ?>admin/add-food.php " class="btn btn-primary">Add Food</a>

<br><br><br>

<table class="tbl-full">

<tr>

<th> Sl.No</th>

<th>Title</th>

<th>Price</th>

<th>Image</th>

<th>Featured</th>

<th>Active</th>

<th>Actions</th>

</tr>

<?php

// create sql query to get all the food details

$sql = "SELECT \* FROM tbl\_food";

// execute the query

$res = mysqli\_query($conn, $sql);

// count rows to check whether there are food or not

$count=mysqli\_num\_rows($res);

// create number variable for sl number

$sn=1;

if($count>0)

{

// we have food in database

// Get the food from database and display

while($row=mysqli\_fetch\_assoc($res))

{

// get the value from individual columns

$id = $row['id'];

$title = $row['title'];

$price = $row['price'];

$image\_name = $row['image\_name'];

$featured = $row['featured'];

$active = $row['active'];

?>

<tr>

<td><?php echo $sn++; ?></td>

<td><?php echo $title; ?></td>

<td><?php echo $price; ?></td>

<td>

<?php

// check whether we have image or not

if($image\_name=="")

{

// we do not have image

echo "<div class='error'>Image Not Added</div>";

}

else

{

// we have image

?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name; ?>" width = "100px">

<?php

}

?>

</td>

<td><?php echo $featured; ?></td>

<td><?php echo $active; ?></td>

<td>

<a href="<?php echo SITEURL; ?>admin/update-food.php?id=<?php echo $id;?>" class="btn btn-secondary">Update Food</a>

<a href="<?php echo SITEURL; ?>admin/delete-food.php?id=<?php echo $id;?>&image\_name=<?php echo $image\_name; ?>" class="btn btn-danger">Delete Food</a>

</td>

</tr>

<?php

}

}

else

{

// food not added to database

echo "<tr> <td colspan='7' class = 'error'>Food not added yet</td></tr>";

}

?>

</table>

</div>

</div>

<!-- Main Content Ends -->

<?php include('partial/footer.php');?>

**9.15 Manage Order**

<?php

ob\_start();

?>

<?php include('partial/menu.php');?>

<!-- Main Content Starts -->

<div class="main-content">

<div class="wrapper-order">

<h1>Manage Order</h1>

<br><br>

<?php

if(isset($\_SESSION['update']))

{

echo $\_SESSION['update'];

unset($\_SESSION['update']);

}

?>

<?php

if(isset($\_SESSION['delete']))

{

echo $\_SESSION['delete'];

unset($\_SESSION['delete']);

}

?>

<br><br>

<!-- button to add admin -->

<table class="tbl-full">

<tr >

<th> Sl.No</th>

<th>Food</th>

<th>Price</th>

<th>Qty</th>

<th>Total</th>

<th>Order Date</th>

<th>Status</th>

<th>Customer Name</th>

<th>Customer Contact</th>

<th>Email</th>

<th>Address</th>

<th>Actions</th>

</tr>

<?php

// get all the details from database

$sql = "SELECT \* FROM tbl\_order ORDER BY id DESC" ;

// Execute query

$res = mysqli\_query($conn, $sql);

// count number of rows

$count = mysqli\_num\_rows($res);

$sn = 1;

// check data is available

if($count>0)

{

// data available

while($row = mysqli\_fetch\_assoc($res))

{

// Get all the order details

$id = $row['id'];

$food = $row['food'];

$price = $row['price'];

$qty = $row['qty'];

$total = $row['total'];

$order\_date = $row['order\_date'];

$status = $row['status'];

$customer\_name = $row['customer\_name'];

$customer\_contact = $row['customer\_contact'];

$customer\_email = $row['customer\_email'];

$customer\_address = $row['customer\_address'];

?>

<tr class = "small">

<td><?php echo $sn++; ?></td>

<td><?php echo $food; ?></td>

<td><?php echo $price; ?></td>

<td><?php echo $qty; ?></td>

<td><?php echo $total; ?></td>

<td><?php echo $order\_date; ?></td>

<td>

<?php

if($status=="Ordered")

{

echo "<label>$status</label>";

}

elseif($status=="On Delivery")

{

echo "<label style='color:orange;'>$status</label>";

}

elseif($status=="Delivered")

{

echo "<label style='color:green;'>$status</label>";

}

elseif($status=="Cancelled")

{

echo "<label style='color:red;'>$status</label>";

}

?>

</td>

<td><?php echo $customer\_name; ?></td>

<td><?php echo $customer\_contact; ?></td>

<td><?php echo $customer\_email; ?></td>

<td><?php echo $customer\_address; ?></td>

<td>

<a href="<?php echo SITEURL; ?>admin/update-order.php?id=<?php echo $id; ?>" class="btn btn-secondary">Update Order</a>

</td>

</tr>

<?php

}

}

else

{

// data not available

echo "<tr><td colspan='12' class = 'error'> Order not available </td></tr>";

}

?>

</table>

</div>

</div>

<!-- Main Content Ends -->

<div class="main-content, wrapper , text-center">

<form action="" method="POST" class="order" >

<input type="submit" name="Delete" value="Delete All" class="btn btn-danger">

</form>

</div>

<?php

// check whether delete button is clicked or not

if(isset($\_POST['Delete']))

{

// echo "pressed";

$sql2 = "DELETE FROM tbl\_order";

// execute query

$res2 = mysqli\_query($conn, $sql2);

if($res2 == TRUE)

{

//Query executed and updated

$\_SESSION['delete'] = "<div class='success'>Order deleted Successfully</div>";

header('location:'.SITEURL.'admin/manage-order.php');

}

else

{

// failed to update

$\_SESSION['delete'] = "<div class='error'>Order delete failed</div>";

header('location:'.SITEURL.'admin/manage-order.php');

ob\_enf\_fluch();

}

}

?>

<?php include('partial/footer.php');?>

**Chapter No: 10**

**TESTING**

**10.1 SOFTWARE TESTING**

As the coding is completed according to the requirement, we have to test the quality of the software. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Although testing is to uncover the errors in the software but is also demonstrates that software functions appear to be working as per the specification, those performance requirements appear top have been met. In addition, data collected as testing is conducted provide a good indication of software reliability and some indications of software quality as a whole. To assure the software quality we conduct both White Box Testing and Black Box Testing.

**10.2 WHITE BOX TESTING:**

White Box Testing is a test case design method that uses the control structure of the procedural designs to derive test cases. As we are using a non procedural language, there is very small scope for the white Box Testing. Whenever it is necessary, there the control structures are tested and successfully passed all the control structures with a very minimum error.

**10.3 BLACK BOX TESTING:**

Black Box Testing focuses on the functional requirement of the software. It enables to derive sets of input conditions that will fully exercise all functional requirements for a program.

The Black Box Testing finds almost all errors. If finds some interface errors and errors in accessing the database and some performance errors. In Black Box Testing we use mainly two techniques: Equivalence Partitioning the Boundary Volume Analysis Technique.

**Equivalence Partitioning:**

In this method we divide input domain of program into classes of data from which test cases are derived. An Equivalence class represents a set of valid or invalid or a set of related values or a Boolean condition. The equivalence for these is:

Input condition requires specific, value-specific or non-specific two classes.

* Input condition requires a range- in the range or out of range two classes.
* Input condition specifies a member of a set- belongs to a set or not belongs to the set two classes.
* Input condition is Boolean- valid or invalid Boolean condition two classes.

By these types of equivalent classes, we can test for many cases.

**Boundary Values Analysis:**

Number of errors usually occurs at the boundaries of the input domain generally. In this technique a selection of test cases is exercised using boundary values i.e. around boundaries.

By the above two techniques, we eliminated almost all errors from the software and checked for numerous test values for each and every input value. The results were satisfactory.

**10.4 SYSTEM TESTING**

System testing is designated to uncover weakness that was not detected in the earlier tests. The total system is tested for recovery and fallback after various major failures to ensure that no data are lost. An acceptance test is done to validity and reliability of the system. The philosophy behind the testing is to find error in project. There are many test cases designed with this in mind. The flow of testing is as follows:

**10.4.1 Code Testing:**

Specification testing is done to check if the program does with it should do and how it should behave under various condition or combinations and submitted for processing in the system and it s checked if any overlaps occur during the processing.

This strategy examines the logic of the program. Here only syntax of the code is tested. In code testing syntax error are corrected, to ensure that the code is perfect.

**10.4.2 Unit Testing:**

The first level of testing is called unit testing. Here different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test oracles.

Unit testing comprises a set of tests preformed by an individual programmer prior to integration of the units into a large system. A program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first IB the modules to locate errors. These errors are verified and corrected so that the unit perfectly fits to the project.

**10.4.3 System Testing:**

The next level of testing is system testing and acceptance testing. This testing is done to check if the system has met its requirements and to find the external behavior of the system.

System testing involves two kinds of activities:

* Integration testing
* Acceptance testing.

**10.4.4 Integration Testing:**

The next level of testing is called the Integration Testing. In this many tested modules are combined into subsystems, which were then tested.

Test case data is prepared to check the control flow of all the modules and to exhaust all possible inputs to the program. Situations like treating the modules when there is no data entered in the test box is also tested.

This testing strategy dictates the order in which modules must be available, and exerts strong influence on the order in which the modules must be written. Debugged and unit tested. In integration testing, all the modules/units on which unit testing is performed are integrated together and tested.

**10.4.5 Acceptance Testing:**

This testing is performed finally by user to demonstrate that the implemented system satisfies its requirements. The user gives various inputs to get required outputs.

**10.4.6 Specification Testing:**

Specification testing is done to check if the program does what it should do and how it should behave under various conditions or combination and submitted for processing in the system and it is checked if any overlaps occur during the processing.

**10.4.7 Performance Time Testing:**

Performance time testing is done to determine how long it takes to accept and respond i.e., the total time for processing when it has to handle quite a large number of records. It is essential to check the exception speed of the system, which runs well with only a handful of test transactions. Such systems might be slow when fully loaded. So testing is done by providing large number of data for processing. A system testing is designed to uncover weaknesses that were not detected in the earlier tests.

The total system is tested for recovery and fallback after various major failures to ensure that no data are lost during an emergency. An acceptance test is done to ensure the user about the validity and reliability of the system.

**Chapter No: 11**

**Implementation**

**Chapter No: 12**

**Scope for Improvement and Future Enhancement**

**Chapter No: 13**

**References**

**BIBLIOGRAPHY**

* https://www.tutorialspoint.com/index.htm
* https://www.javatpoint.com
* https://www.w3schools.com
* https://html.com