<div align="right">CHAPTER 1</div>

INTRODUCTION

Online food ordering is the process of ordering food from a website. The product can be either ready-to-eat food (e.g., direct from a certified home-kitchen, restaurant) or food that has not been specially prepared for direction consumption (e.g., vegetables direct from a farm/garden, frozen meats. etc).The aim of developing Online Food Ordering system project is to replace the traditional way of taking orders with computerized system.

1.1 Motivation for this system?

The main motivation of this system is to manage the details of item category, food, delivery address, order, and shopping cart. It manages all the information about item category, customer, shopping cart, item category. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose is to build and application program to reduce the managing the item category, food customers. It tracks all he delivery address ordered.

THE DATA STORED USING THIS ARE :-

1) LOGIN DATA

2) USER SIGN UP

3) NAMES AND DETAILS OF THE PERSON

4) VIEW FOOD DETAILS

5) STORING THE DATA

6) SIGN OUT

1.2 EXISTING SYSTEM

Online Food Ordering System uses MYSQL Server as backend so there is not any chance of data loss or data security. A customer can choose to have the food delivered or for pick-up. The process consists of a customer choosing the restaurant of their choice, scanning the menu items, choosing an item, and finally choosing for pick-up or delivery. Payment is then administered by paying with a credit card or debit card through the app or website or in cash at the restaurant when going to pickup. The website and app inform the customer of the food quality, duration of food preparation, and when the food is ready for pick-up or the amount of time it will take for delivery

1.3 PROPOSED SYSTEM

So the system proposed by me is same as the Existing system but the few helpful or useful

changes done are it will once take to the login page, if not signed up you can sign up with the

button below and once you sign up you can login in with the credentials you have given and

once you login you will reach to the home page which will contain DASH BOARD, NEW

VISITOR, MANAGE VISITORS, VISITOR BETWEEN DATE. There u can see the persons
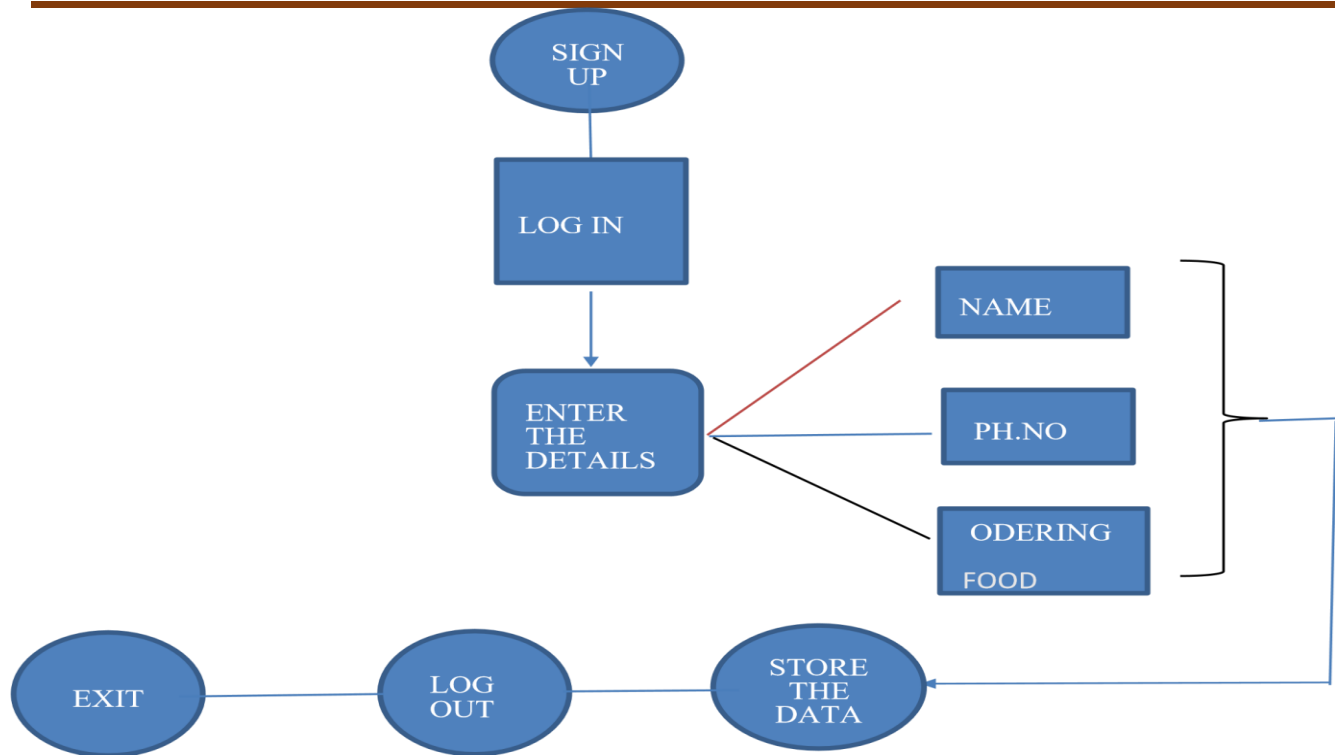
who visited and you can see the details of the person.

Fig.1.1 Project Flowchart

1.4 ADVANTAGES AND DISADVANTAGES

ADVANTAGES ARE :-

1. Embracing Paperless

2. Reduced Waiting Times

3. Increased Security

4. Maintain food order database

5. Record Food orders details

6. We can see the details of person when ever we want.

7. This software is the better and easy way of doing the front office work

CHAPTER 2

REQUIREMENTS

Components Required –

1. HARDWARE COMPONENTS ARE :

- o   RAM – 1GB or Above

- o   Hard Disk – Minimum 20GB free space

- o   Processor – Pentium 4 (1.6GHZ) or Higher

2. SOFTWARE COMPONENTS ARE :

- o   Operating system: Windows DXP/ 2000/ Vista/0/ 7/ 8/ 10AS

- o   Front End: XAMPP [PHP ADMIN]

- o   Back End: MYSQL DATABASE

- o   Language: XAMPP server, HTML,PHP.

- o   Database: MYSQL PHP MYADMIN

CHAPTER 3

IMPLEMENTATION

3.1 HTML (Hyper Text Markup Language)

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination   of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure   of web pages.

HTML is a markup language that is used by the browser to manipulate text, images, and other content to display it in the required format.

Characteristics of HTML:

Easy to understand: It is the easiest language you can say, very easy to grasp this   language and easy to develop.

Flexibility: This language is so much flexible that you can create whatever you   want, a flexible way to design web pages along   with the text.

Linkable: You can make linkable text like users can connect from one page to   another page or website through these characteristics.

Limitless features: You   can add videos, gifs, pictures or sound anything you   want that will make the website more attractive and understandable.

Support: You can use this   language to display the documents on any platform   like Windows, Linux or Mac.

3.1(a) TOPICS OF HTML :

O   Basics

O   HTML 5

O   Tags

O   Attributes

O   Misc

3.1(b) BASICS:

O   Attributes

O   Paragraphs

O   Tables

O   Labels

O   Lists

O   Links

O   Images

O   Layouts

O   Text Format

3.1(c) MOST COMMONLY USED TAGS:

HTML TAG:-

   It is the root of the html document which is used to specify that the document is html.

SYNTAX:-   <html> STATEMENT….. </html>

HEAD TAG :-

Head tag is used to contain all the head element in the html file. It contains the title, style, meta etc tag.

SYNTAX :-   <head> STATEMENT…..</head>

BODY TAG:

It is used to define the body of html document. It contains image, tables, lists, etc.

TITLE TAG:

It is used to define the title of html document

HEADING TAG:

It is used to define the heading   of html document

PARAGRAPH TAG:

It is used to define paragraph content in html document.

BOLD TAG:

It is used to specify bold   content in html document

ITALIC TAG:

It is used to write   the content in italic format

ORDERED LIST TAG:

It is used to list the content in a particular order.

UNORDERED LIST TAG:

It is used to list the content without order

FONT TAG:

It is used to specify the font size, font colour and font-family in html document.

LINE BREAK TAG:

It is used to break the line

IMAGE TAG:

It is used to add image element in html document

3.2 CSS (CASCADING SYSTEM SHEETS):

Cascading Style Sheets, fondly   referred to as CSS, is a simply designed language intended to simplify the process of making web   pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables   you to do this independent of the HTML that makes up each web   page.

There are three types of CSS which are given below:

• Inline CSS

• Internal or Embedded CSS

• External CSS

Inline CSS:

Inline CSS contains the CSS property in the body   section attached with element is known as   inline CSS. This kind of style is specified within an HTML tag using the style attribute

Internal or Embedded CSS:

This can be used when a single HTML document must be styled uniquely the CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.

External CSS:

External CSS contains separate CSS file   which contains only style property with the help of tag attributes (For example class, id, heading,   etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using **LINK** tag. This means that for each element, style can be set   only once and that will be applied across web pages.

3.2(a) PROPERTIES OF CSS:

Inline CSS has the highest priority, then comes Internal/Embedded followed by External   CSS which has the least priority. Multiple style sheets can be defined on one page. If for an
 HTML tag, styles are defined in multiple style sheets then the below order will be followed.

  O   As Inline   has the highest priority, any styles that are defined in the internal and   external style sheets are overridden by Inline styles .

  O   Internal or Embedded stands 2nd in the priority list and overrides the styles in the external style sheet .

  O   External style sheets have the last priority. If there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML tags.

3.2(b) FUNCTIONS OF CSS:

CSS functions are used as a value for various CSS properties. For example, you can use   the rgb() function to provide a color value (such as color: rgb(255,0,215)), or the attr() function to retrieve the value of an HTML   attribute. Many functions are used in CSS transforms. For example, the rotate() function can be used to rotate an element, the scale() function can be used to change the size of an element, and the translate() function can be used to move an element .

There are also other handy functions, such as circle() to clip an element to a circle or to create circle for text to flow around, and the calc() function that can be used to provide a calculated value for a property .

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.

3.3 JAVASCRIPT IN PHP:

JavaScript is the client side scripting language and PHP is the server side scripting language. JavaScript is used as client side to check and verify client details and PHP is server side used to interact with database. In PHP, HTML is used as a string in the code.

3.3(a) JSON PHP (JAVASCRIPT OBJECT NOTATION):

JSON stands for the JavaScript Object Notation. It is used to exchanging and storing the data   from the web-server. JSON uses the object notation of JavaScript. JavaScript objects can be   converted into the JSON and receive JSON format text into the JavaScript objects. Converting   the JavaScript object into the JSON format is done by the given function:

JS 0 N.stringify(ob ject)

Exchanging the data from server, PHP as server language used. The JSON.parse() function is used to get the data from php 0r from the any 0ther server. For the receiving the data from the server

few AJAX statements t0 check whether the sever is ready to respond the data from the   server 0r not. If those conditions are fulfilled then the data from the php file can be received.

3.3(b) PASSING JAVASCRIPT VARAIABLES TO PHP:

JavaScript is the client side and PHP is the server side script language. The way to pass a JavaScript variable to PHP is through a request.

Method 1: This example uses form element and GET/P0ST method to pass JavaScript variables to PHP. The form of contents can be accessed through the GET and P0ST actions in PHP. When the f0rm is submitted, the client sends the f0rm data in the form of a URL such as:

```
<!DOCTYPE html>
<html>

<head>
        <title>
                Passing JavaScript variables to PHP
    </title>
</head>

<body>
        <h1 style="color:green;">
                GeeksforGeeks
    </h1>


                <form method="get" name="form" action="destination.php">
                        <input type="text" placeholder="Enter Data" name="data">
```

```
<input type="submit" value="Submit">                </form>
</body> </html>
```

3.4 PHP (HYPERTEXT PREPROCESSOR)

The term PHP is an acronym for PHP- HYPERTEXT PREP0CESS0R. PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files   and HTML5 codes can also be written in a PHP file. The thing that differentiates PHP with client-   side language like HTML UP is, PHP codes are executed on the server whereas HTML codes are   directly rendered on the browser.

3.4(a) FUNCTIONS OF PHP:

A function is a block of code written in a program to perform some specific task. We can relate functions in programs to employees in a 0ffice in real life for a better understanding of how   functions work. Suppose the boss wants his employee to calculate the annual budget. So how   will this process complete   The employee will take information about the statics from the boss,   performs calculations and calculate the budget and shows the result to his boss. Functions
works in a similar manner. They take information as parameter, executes a block of statements   or perform operations on this parameters and returns the result

PHP provides us with two major types 0f functions:

• **Built-in functions** :

 PHP provides us with huge collection of built-in library functions. These   functions are already coded and stored in form of functions. To use those we just   need to call them as per our requirement like, var dump, fopem(), printr(), gettype () and so on.

• **User Defined Functions** :

Apart from the built-in functions, PHP allows us to create our own customised

functions called the user-defined functions.

Using this we can create 0ur own packages of code and use it wherever

necessary by simply calling it.

3.4(b) CREATING A FUNCTION:

While creating a user defined function we need to keep few things in mind:

• Any name ending with an open and closed parenthesis is a function.

• A function name always begins with the keyword functions.

• To call a function we just need to write its name followed by the parenthesis

• A function name cannot start with a number. It can start with an alphabet or underscore.

• A function name is not case-sensitive.

SYNTAX:
```
        Function Function_ name() {
                Executable code
        }
```

3.4(c) ARGUMENTS:

The information or variable, within the function's parenthesis, are called parameters. These are   used to hold the values executable during runtime. A user is free to take in as many parameters   as he wants, separated with a comma 0perator. These parameters are used to accept inputs   during runtime. While passing the values like during a function call, they are called Arguments.

An argument is a value passed to a function and a parameter is used to hold th0se arguments.   In common term, both parameter and argument mean the same. We need to keep in mind that   for every parameter we need to pass its corresponding argument.

3.5 MYSQL DATABASE:

MYSOL database is being operated threw XAMPP server, basically the apache. Here we have to set up the mysql database by adding or creating a new table and then have to code all the SQL commands in the server or we can comment all the commands in NOTEPAD or NOTEPAD++ etc and then we can import the file into the serve and run it. XAMPP is a cross-platform web server used to develop and test programs on a local server. It is developed and managed by Apache Friends and is open-source. It has an Apache HTTP Server, MariaDB, and interpreter for 11 different programming languages like Perl and PHP. XAMPP Stands for cross-platform, Apache, MySQL, PHP, and Perl.

- It allows you to build a website on a local web server on your computer.
- Stores data locally

   In this article, we are going to perform Database operations like Create, Insert, Update, Delete data from the database created in XAMPP localhost server. We are also going to create a table and then start performing database operations. Following are the list of database operations with their respective syntax.

PROCEDURE:
Follow the below steps to perform database operations on XAMPP:

- Start XAMPP Server
- Create Database and Create Table
- Perform Database Operations
- Verify Resultant Table

3.6 BOOTSTRAP:

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites. Nowadays, the websites are perfect for all the browsers (IE,   Firefox and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers – Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Why Bootstrap?

• Faster and Easier Web-Development.

• It creates Platform-independent web-pages.

• It creates Responsive Web-pages.

• It designed to be responsive to mobile devices too.

• It is Free! Available on www.getbootstrap.com

CHAPTER 4

CODE IMPLEMENTATION

ALL THE PHP ANF HTML FILES:

- ✟ ADD PROFILE

- ✟ MANAGE PROFILE

- ✟ PASSWORD

- ✟ CHANGE PASSWORD

- ✟ ENTER PRODUCTS

- ✟ ADD IMAGES

- ✟ INVOICE

- ✟ INDEX

- ✟ LOG OUT

- ✟ VIEW INVOICE

- ✟ SIGN UP

- ✟ SIGN IN

- ✟ BW DATE

- ✟ DETAILS OF THE PERSON

INDEX FILE:

```php
<?php
session_start();
?>
<html>

 <head>
  <title> Home | Le Cafe' </title>
 </head>

 <link rel="stylesheet" type = "text/css" href ="css/bootstrap.min.css">

 <link rel="stylesheet" type = "text/css" href ="css/index.css">

 <body>

  <button onclick="topFunction()" id="myBtn" title="Go to top">
   <span class="glyphicon glyphicon-chevron-up"></span>
  </button>
  <script                       type="text/javascript">
window.onscroll = function()
   {
     scrollFunction()
   };

   function scrollFunction(){
     if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {
document.getElementById("myBtn").style.display = "block";
     } else {
       document.getElementById("myBtn").style.display = "none";
     }
   }

   function        topFunction() {       document.body.scrollTop = 0;
     document.documentElement.scrollTop = 0;
   }

 </script>
```

```
    <nav      class="navbar  navbar-inverse        navbar-fixed-top      navigation-clean-search"
role="navigation">
    <div class="container">
      <div class="navbar-header">
        <button      type="button"      class="navbar-toggle      collapsed"      data-toggle="collapse"
datatarget="#myNavbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="index.php">Le Cafe'</a>
      </div>

      <div class="collapse navbar-collapse " id="myNavbar">
        <ul class="nav navbar-nav">
          <li class="active" ><a href="index.php">Home</a></li>
          <li><a href="aboutus.php">About</a></li>
          <li><a href="contactus.php">Contact Us</a></li>

        </ul>

<?php
if(isset($_SESSION['login_user1'])){
 ?>

        <ul class="nav navbar-nav navbar-right">
        <li><a href="#"><span class="glyphicon glyphicon-user"></span> Welcome <?php echo
$_SESSION['login_user1']; ?> </a></li>
        <li><a href="myrestaurant.php">MANAGER CONTROL PANEL</a></li>
        <li><a href="logout_m.php"><span class="glyphicon glyphicon-log-out"></span> Log Out
</a></li>          </ul> <?php
}
else if (isset($_SESSION['login_user2'])) {  ?>
        <ul class="nav navbar-nav navbar-right">
        <li><a  href="#"><span  class="glyphicon  glyphicon-user"></span>  Welcome  <?php  echo
$_SESSION['login_user2']; ?> </a></li>
```

```
<li><a  href="foodlist.php"><span  class="glyphicon  glyphicon-cutlery"></span> Food  Zone
</a></li>
       <li><a href="cart.php"><span class="glyphicon glyphicon-shopping-cart"></span> Cart
        (<?php
        if(isset($_SESSION["cart"])){
$count = count($_SESSION["cart"]);
        echo "$count";
        }          else
echo "0";
?>)
       </a></li>
       <li><a href="logout_u.php"><span class="glyphicon glyphicon-log-out"></span> Log Out
</a></li>          </ul>
 <?php
}
else {

  ?>

<ul class="nav navbar-nav navbar-right">
       <li><a href="#" class="dropdown-toggle active" data-toggle="dropdown" role="button" aria-
haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-user"></span>
Sign Up <span class="caret"></span> </a>
         <ul class="dropdown-menu">
        <li> <a href="customersignup.php"> User Sign-up</a></li>
        <li> <a href="managersignup.php"> Manager Sign-up</a></li>

       </ul>
       </li>

       <li><a href="#" class="dropdown-toggle active" data-toggle="dropdown" role="button" aria-
haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-log-in"></span>
Login <span class="caret"></span></a>
        <ul class="dropdown-menu">
        <li> <a href="customerlogin.php"> User Login</a></li>
        <li> <a href="managerlogin.php"> Manager Login</a></li>

       </ul>
       </li>
```

```
        </ul>

<?php
} ?>
    </div>

     </div>
   </nav>

   <div class="wide">
            <div class="col-xs-5 line"><hr></div>
      <div class="col-xs-2 logo"><img src="images/LogoImage.jpg"></div>
      <div class="col-xs-5 line"><hr></div>
      <div class="tagline">Good Food is Good Mood</div>
   </div>
   <br>
   <div class="orderblock">
   <h2>Feeling Hungry?</h2>
   <center><a class="btn btn-success btn-lg" href="customerlogin.php" role="button" > Order Now
</a></center>
   </div>
</body>
</html>
```

 SIGN UP PAGE:

```
<?php  session_start();

$error='';

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

if (empty($_POST['username']) || empty($_POST['password'])) {
$error = "Username or Password is invalid";

}

else

{
```

```php
// Define $username and $password

$username=$_POST['username'];

$password=$_POST['password']; require

'connection.php';

$conn = Connect();


// SQL query to fetch information of registerd users and finds user match.

$query = "SELECT username, password FROM MANAGER WHERE username=? AND password=?

LIMIT 1";


$stmt = $conn->prepare($query);

$stmt -> bind_param("ss", $username, $password);

$stmt -> execute();

$stmt -> bind_result($username, $password);

$stmt -> store_result();


if ($stmt->fetch())

{

  $_SESSION['login_user1']=$username;  //        Initializing        Session                    header("location:

myrestaurant.php"); // Redirecting To Other Page

} else {

$error = "Username or Password is invalid";

}

mysqli_close($conn); // Closing Connection

}

}
```

CUSTOMER LOGIN:

```php
   <?php
include('login_u.php');


if(isset($_SESSION['login_user2'])){
header("location: foodlist.php");
}
?>
```

```html
<!DOCTYPE html>
<html>

 <head>
   <title> Guest Login | Le Cafe' </title>
 </head>

 <link rel="stylesheet" type = "text/css" href ="css/managerlogin.css">
 <link rel="stylesheet" type = "text/css" href ="css/bootstrap.min.css">
 <script type="text/javascript" src="js/jquery.min.js"></script>
 <script type="text/javascript" src="js/bootstrap.min.js"></script>

 <body>


  <button onclick="topFunction()" id="myBtn" title="Go to top">
    <span class="glyphicon glyphicon-chevron-up"></span>
  </button>


  <script type="text/javascript">
```

```
window.onscroll = function()

{

 scrollFunction()

};


function scrollFunction(){

  if (document.body.scrollTop > 20 || document.documentElement.scrollTop > 20) {

    document.getElementById("myBtn").style.display = "block";

  } else {

    document.getElementById("myBtn").style.display = "none";

  }

}


function topFunction() {

  document.body.scrollTop = 0;

  document.documentElement.scrollTop = 0;

 }

</script>


<nav class="navbar navbar-inverse navbar-fixed-top navigation-clean-search" role="navigation">

 <div class="container">

  <div class="navbar-header">

   <button    type="button"    class="navbar-toggle    collapsed"    data-toggle="collapse"    data-target="#myNavbar">

     <span class="sr-only">Toggle navigation</span>

     <span class="icon-bar"></span>

     <span class="icon-bar"></span>

     <span class="icon-bar"></span>

   </button>
```

```
        <a class="navbar-brand" href="index.php">Le Cafe'</a>
    </div>


    <div class="collapse navbar-collapse " id="myNavbar">
     <ul class="nav navbar-nav">
       <li ><a href="index.php">Home</a></li>
       <li><a href="aboutus.php">About</a></li>
       <li><a href="contactus.php">Contact Us</a></li>
     </ul>


     <ul class="nav navbar-nav navbar-right">
       <li><a  href="#"  class="dropdown-toggle  active"  data-toggle="dropdown"  role="button"  aria-
haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-user"></span> Sign Up <span
class="caret"></span> </a>
         <ul class="dropdown-menu">
          <li> <a href="customersignup.php"> User Sign-up</a></li>
          <li> <a href="managersignup.php"> Manager Sign-up</a></li>


       </ul>
       </li>


       <li><a  href="#"  class="dropdown-toggle  active"  data-toggle="dropdown"  role="button"  aria-
haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-log-in"></span> Login <span
class="caret"></span></a>
         <ul class="dropdown-menu">
         <li> <a href="customerlogin.php"> User Login</a></li>
         <li> <a href="managerlogin.php"> Manager Login</a></li>


       </ul>
```

```
        </li>
      </ul>
    </div>


    </div>
  </nav>


  <div class="container">
  <div class="jumbotron">
   <h1>Hi Guest,<br> Welcome to <span class="edit"> Le Cafe' </span></h1>
   <br>
  <p>Kindly LOGIN to continue.</p>
   </div>
   </div>


  <div class="container" style="margin-top: 4%; margin-bottom: 2%;">
    <div class="col-md-5 col-md-offset-4">
      <label style="margin-left: 5px;color: red;"><span> <?php echo $error;  ?> </span></label>
    <div class="panel panel-primary">
     <div class="panel-heading"> Login </div>
     <div class="panel-body">


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


     <div class="row">
      <div class="form-group col-xs-12">
       <label for="username"><span class="text-danger" style="margin-right: 5px;">*</span> Username:
</label>
        <div class="input-group">
```

```
        <input        class="form-control"        id="username"        type="text"        name="username"
placeholder="Username" required="" autofocus="">

        <span class="input-group-btn">

        <label        class="btn        btn-primary"><span        class="glyphicon        glyphicon-user"        aria-
hidden="true"></label>

    </span>

     </span>

    </div>

   </div>

  </div>


   <div class="row">

    <div class="form-group col-xs-12">

     <label for="password"><span class="text-danger" style="margin-right: 5px;">*</span> Password:
</label>

     <div class="input-group">

      <input        class="form-control"        id="password"        type="password"        name="password"
placeholder="Password" required="">

       <span class="input-group-btn">

        <label        class="btn        btn-primary"><span        class="glyphicon        glyphicon-lock"        aria-
hidden="true"></span></label>

     </span>


     </div>

    </div>

   </div>


   <div class="row">

    <div class="form-group col-xs-4">
```
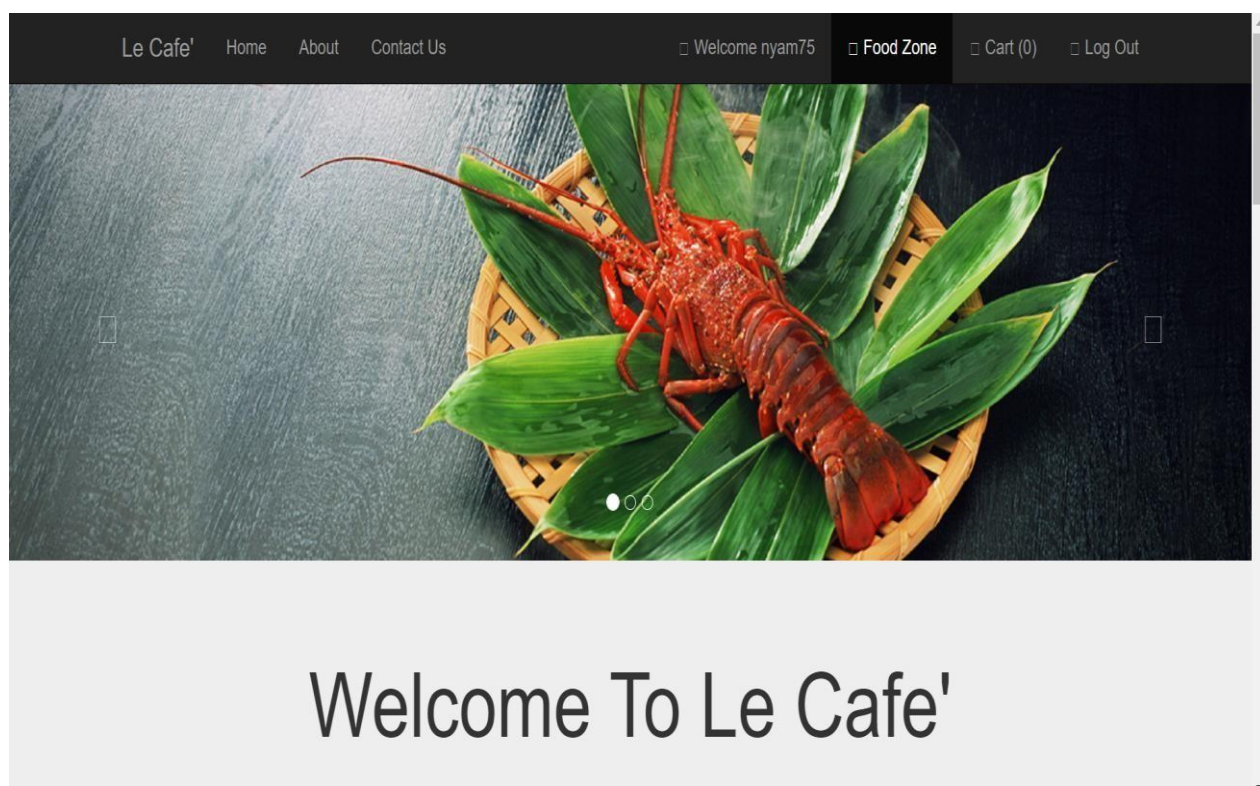
```html
            <button class="btn btn-primary" name="submit" type="submit" value=" Login ">Submit</button>
        </div>


    </div>
    <label style="margin-left: 5px;">or</label> <br>
    <label style="margin-left: 5px;"><a href="customersignup.php">Create a new account.</a></label>


    </form>
    </div>
    </div>
    </div>
    </div
    </body>
</html>
```
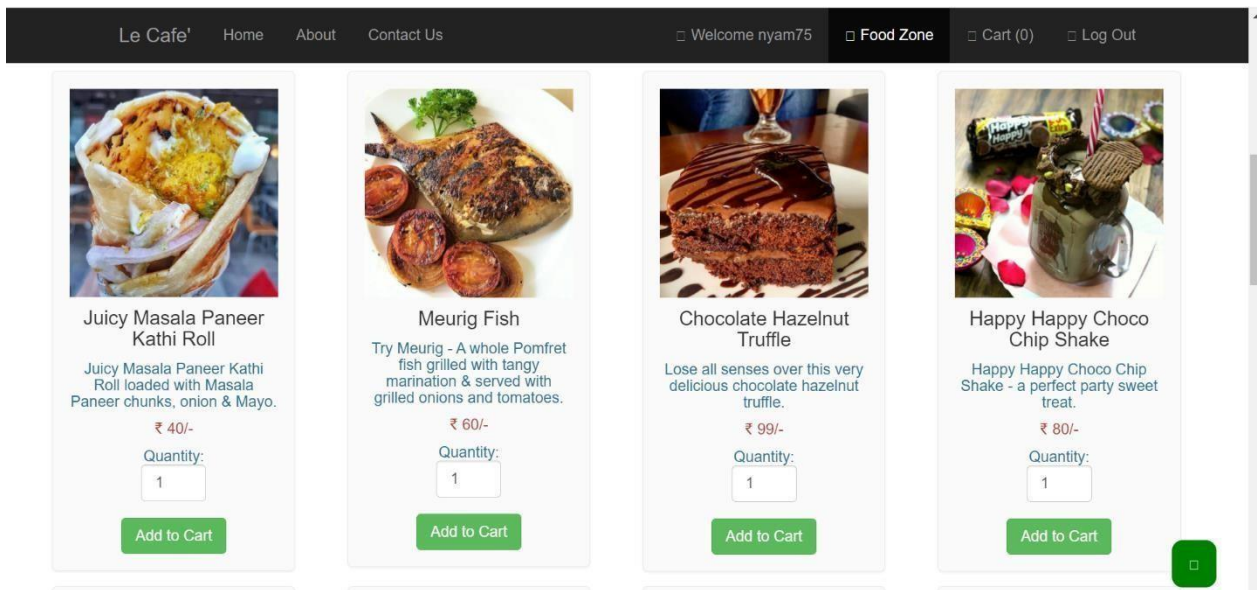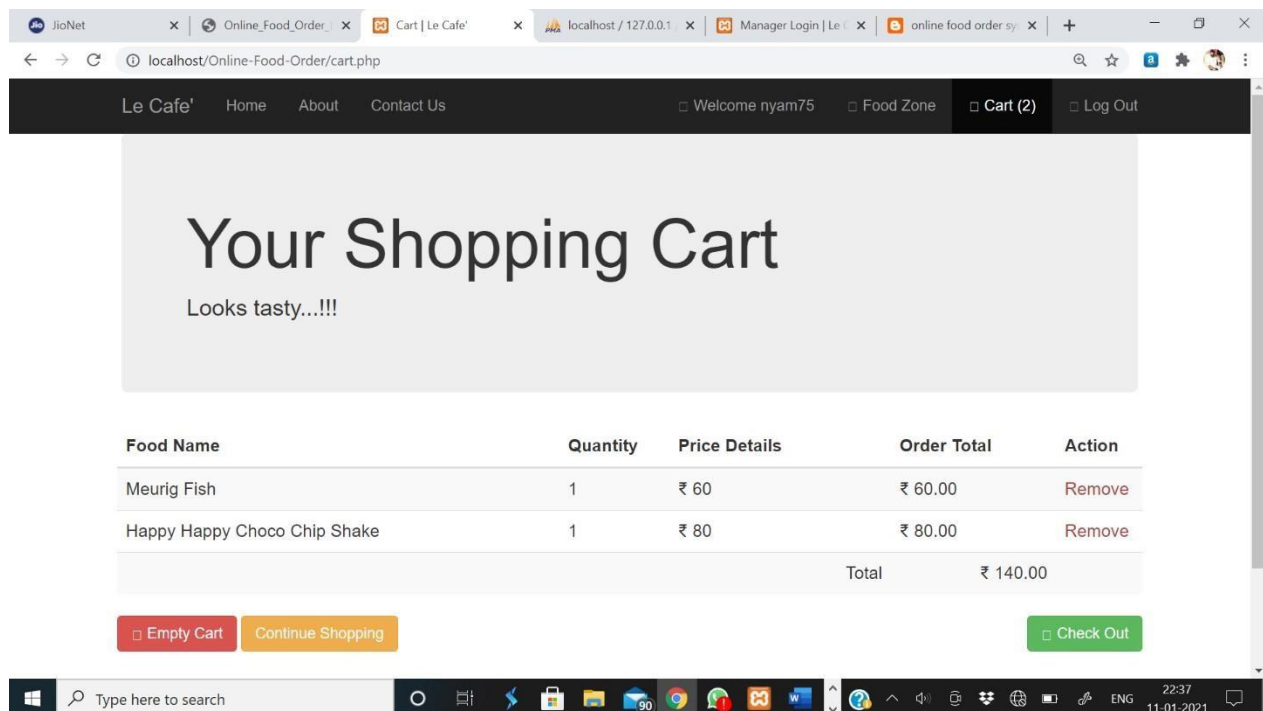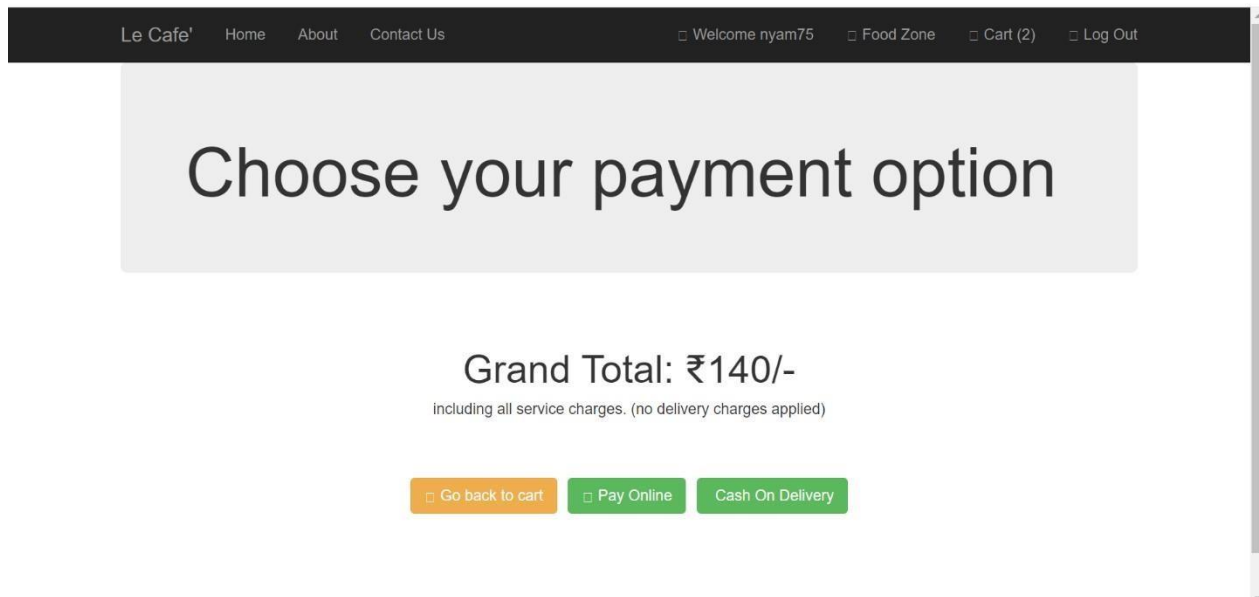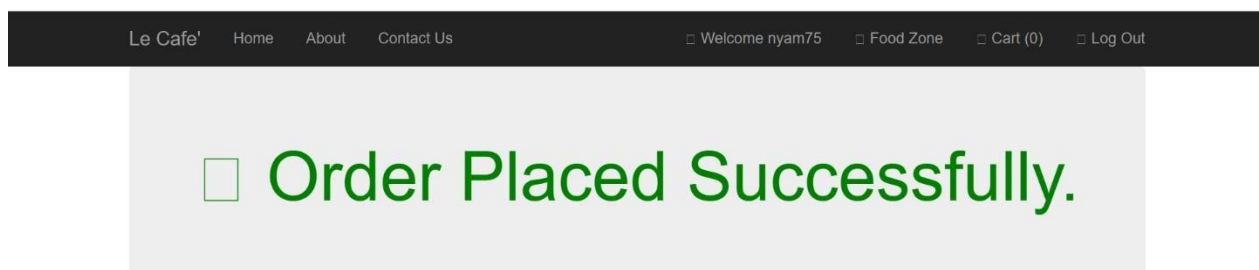
OUTPUT:

Output:1-Home page

Output:2-Food Zone For Ordering Food

Output:3-Food Items To book

Output:4-Food Items In The Cart

Output:-Choosing Payment Option

Output:5-Order Placed Successfully

CHAPTER 5

CONCLUSION

Therefore, conclusion of the proposed system is based on user's need and is user centered. The system is developed in considering all issues related to all user which are included in this system. Wide range of people can use this if they know how to operate android smart phone. Various issues related to Mess/Tiffin Service will be solved by providing them a full fledged system. Thus, implementation of Online Food Ordering system is done to help and solve one of the important problems of people. Based on the result of this research, it can be concluded: It helps customer in making order easily; It gives information needed in making order to customer. The Food website application made for restaurant and mess can help restaurant and mess in receiving orders and modifying its data and it is also made for admin so that it helps admin in controlling all the Food system.

CHAPTER 6

BIBLIOGRAPHY

1. https://code-projects.org/online-food-order-system-in-php-with-source-code/

2. https://csveda.com/mini-projects/food-ordering-system-using-php-and-mysql/

3. https://itsourcecode.com/free-projects/php-project/online-food-ordering-system-source-code/

4. https://www.phpjabbers.com/food-delivery-script/