MINI PROJECT REPORT

On

QR-BASED
CAFETERIA AUTOMATION SYSTEM

Submitted by

SHUBHAM GUPTA
(171500333)
PRASHANT TIWARI
(171500232)
LALIT VARSHNEY
(171500174)

Department of Computer Engineering & Applications
Institute of Engineering & Technology

GLA University
Mathura- 281406, INDIA
2021

Department of Computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha, Mathura – 281406

Declaration

I hereby declare that the work which is being presented in the Mini Project "**QR-BASED CAFETERIA AUTOMATION SYSTEM**"**,** in fulfilment of the requirements for Summer Training viva voce, is an authentic record of my own work carried under the supervision of **Mr Pankaj Kapoor.**

Name of Candidates:

Shubham Gupta(171500333)

Prashant Tiwari(171500232)

Lalit Varshney(171500174)

Course: B.Tech (CSE)

Year: 3rd Year

Semester: VI Semester

Acknowledgement

I thank the almighty for giving me the courage and perseverance in completing the project.

This project itself is acknowledgement for all those people who have given me their heartfelt cooperation in making this project a grand success. I extend my sincere thanks to Mr Pankaj Kapoor, Associate Professor at GLA University for providing valuable guidance at every stage of this project work. I am profoundly grateful towards the unmatched services rendered by him. Lat but not Least, I would like to express to deep sense of gratitude and earnest thanks to my dear parents for their moral support and heartful cooperation in doing main project.

Abstract

The main objective of QR-Based Cafeteria Automation system is to automate the existing manual system with the help of advance computerized software so that valuable data can be stored for longer period with easy accessing and manipulation of the same.

The registered user can access the account with valid credentials. User can surf the food items according to categories.

This website will provide the list of different menu list with different categories. User can select any item from canteen and can order for it by using wallet Payment. It will provide fast services to their college students, Staffs etc.

Contents

Chapter 1

Introduction

1.1 Motivation

The main objective of online food ordering system is to automate the existing manual system with the help of advance computerized software so, that valuable data can be stored for longer period with easy accessing and manipulation of the same. Usually People have to go to canteen and order the foods and they have to wait in queue for a long time to get the orders. But with the help of this you just have to follow a very simple process to order your stuffs. And you need not to wait in the long queue.

1.2 Overview

The main aim of this project QR-Based Cafeteria automation system is to provide fast services to their college students, Staffs etc. Usually People have to go to canteen and order the foods and they have to wait in queue for a long time to get the orders. But with the help of this you just have to follow a very simple process to order your stuffs. And you need not to wait in the long queue. The main advantage of an online ordering system is that it greatly simplifies the ordering process for both the customer and the canteen. When the customer visits the ordering web page, they are presented with an interactive and up-to-date menu, complete with all available options and adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the customer can review the details at any time before checking out. This provides instant visual confirmation of what was selected. This system also greatly lightens the load on the cafe's end, as the entire process of taking orders is automated. Once an order is placed on the web page, it is entered into the database and then

retrieved, in pretty much real-time, by a web-based application on the canteen's end.

Time Saving: Our QR-Based Cafeteria automation system is developed with a primary aim of Saving Time. The customer can order the food and it is also efficient for canteen workers because this system takes lesser time as compared to phone based or manually based system.

No Complication: Major complication part for Cafeteria automation system is adding a product or managing products section. In our Cafeteria automation system, no complicated part is involved in managing sections; we have taken immense care in this section and nullified all complications which make this system accurate and unique.

Cost Effective: It's cheaper. You don't have to purchase multiple copies of software to install on multiple computers. Multiple copies often require you to pay multiple licensing fees, but since you aren't actually purchasing any software with an online system, that's not a concern.

Security: Online systems are just as secure. Most online Cafeteria automation system programs allow you to create multiple user accounts with various levels of access. Your data is stored on secure, protected servers that feature firewalls and other online security programs.

1.3 Objective

QR-Based Cafeteria Automation System is the system where customers order their food and receive food in the canteen without any delay as they can directly go and collect what they ordered without waiting for a turn or waiting time. This system aims to accelerate customer orders and customer order system used by employees to accept customer order.

The purpose of the system is to develop a simple Cafeteria Automation System and implement it, which later will be used for a web-based application.

The proposed "QR-Based Cafeteria Automation System" is economically feasible because:

1. The system requires very fewer time factors as compared to manual system.
2. The system will provide fast and efficient automated environment instead of slow and error prone manual system, thus reducing both time and manpower spent in running the system.
3. The system will have GUI interface and very less user training is required to learn it.

1.4 How it benefits a cafeteria

QR-Based Cafeteria automation system saves the cafe's time by avoiding the food orders over the phone that has to be done manually. By making the ordering process fully automated, it increases the cost effectiveness and productivity of the canteen with a less manpower. In addition, it keeps you one step ahead of your competitors who don't serve online.

QR-Based Cafeteria automation system even helps a casual dining canteen and provides an additional revenue source. It allows a canteen owner to easily update the online menu, food items etc and helps to stay in touch with the customers by offering discounts and targeted promotions. With a simplified management process, online ordering system makes it easy to handle multiple Cafe's from centralized application

1.5 How it benefits a customer

Now a day's people are getting busy with their work, making the canteen online even saves customers time undoubtedly. Is allow them to order food online by

creating a flexible ordering platform and serve them in time. The customer can select the food items from the online menu on the canteen website and can order accordingly as per their interest. In fact, they can place order the food orders within their budget by adding or removing the food items as per the cost variations.

This QR-Based Cafeteria automation system is useful for the Cafe as well customer because Cafeteria automation system save the time of the Cafe's workers as well of the customers and as saves the manpower.

Chapter 2

Technology Used

2.1 HTML

Hypertext Mark-up 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.

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. So basically, we used HTML to design our website in which we different element and different attribute so that it's looks attractive so some of the elements are listed here:

1. <html></html>
2. <head></head>
3. <body></body>

And we follow the basic syntax approach to design our page that is :

```
<html><head>
   <title> College Management System</title>
   <link rel=" stylesheet" href=" ../CSS/style.css">
```

</head><body> …… …… …….. </body></html>

Cafeteria Software Requirement

2.2 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 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 also has rules for alternate formatting if the content is accessed on a mobile device. The CSS specifications are maintained by the World Wide Web Consortium (W3C). CSS provide the HTML element styling so that element looks attractive so we different type of properties in CSS to style the element some of the examples are listed below and basic syntax follow:

```
.class-name or #id-name
{    width: 100%;
   height: 80px;
   position: absolute;
   background-color: #1f1f1f;
   top: 0;
   opacity: 0.9;
   border-bottom: 3px solid white;
   z-index: 8;
   border-top-right-radius: 10px;
   border-bottom-right-radius: 10px;
```

```
border-top: 1px solid white;
border-right: 3px solid white;}
```


2.3 JavaScript

JS, is a high-level, interpreted scripting language. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM. Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets. Java script used for validation purpose so in our project we use Java Script to define the major functionality for specific element. When we use Java Script in our project, we use the <script></script> tag to define the function of an element.

2.4 BootStrap 4

Bootstrap 4 is the newest version of Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Bootstrap 4 is completely free to download and use!. Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for creating common user interface components

like forms, buttons, navigations, dropdowns, alerts, modals, tabs, accordions, carousels, tooltips, and so on.

Bootstrap gives you ability to create flexible and responsive web layouts with much less efforts.

Bootstrap was originally created by a designer and a developer at Twitter in mid-2010. Before being an open-sourced framework, Bootstrap was known as Twitter Blueprint.

2.4 jQuery

jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript.

jQuery is a fast and concise JavaScript library created by John Resig in 2006. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for Rapid Web Development.

Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.

All jQuery methods are inside a document ready event to prevent any jQuery code from running before the document is finished loading (is ready). It is friendly, which is to say it provides helpful ways to avoid conflicts with other JavaScript libraries.

Basic syntax for any jQuery function is:

```
$(selector).action()
```

1. A $ sign is to define/access jQuery

2. A (selector) is to “query (or find)” HTML elements in html page
3. A jQuery action() is the action to be performed on the selected element(s)

Example:

```
$(document).ready(function(){
    $("button").click(function(){
        $(".gfg").hide(); };) });
```

Cafeteria Software Requirement

2.5 PHP

PHP: Hypertext Pre-processor (or simply PHP) is a general-purpose programming language originally designed for web development. PHP originally stood for Personal Home Page. 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. PHP used as backend programming as well as to connect to the database so PHP code will start in <?php?> tag in which all the PHP code done.

Syntax:

```
<? php

        ……….

                ……….

?>
```

Chapter 3

Software Requirement Analysis

3.1 Problem Statement

The main objective of QR-Based Cafeteria Automation system is to automate the existing manual system with the help of advance computerized software so that valuable data can be stored for longer period with easy accessing and manipulation of the same. Canteen management system is to provide fast services to their college students, Staffs etc. Usually People have to go to canteen and order the foods and they have to wait in queue for a long time to get the orders. But with the help of this you just have to follow a very simple process to order your stuffs. And you need not to wait in the long queue.

3.2 Modules and their Functionalities

3.2.1 Frond End(Main Website)

Through this front end customer can login and can perform the operations for which they are authorized respectively.

Functionalities provided to the customers are defined below.

1. Register: Student need to register first with basic registration details and need to create a valid login id and password.
2. Login: Student need to login using their valid login credentials in order to access the web application.
3. View Items: All the food items will be displayed to the student at once with description and cost.

4 Add to Cart: Single or multiple food item can be added to cart by selecting quantity.

5 Order and Pay: Order can be placed of selected food items by using a dummy card

6 Checkout: You can checkout with the items that you have selected and at your table the order will be served.

3.3 Software Requirement

3.3.1 Brackets

Brackets is a source code editor with a primary focus on web development. Created by Adobe Systems, it is free and open-source software licensed under the MIT License, and is currently maintained on GitHub by Adobe and other open-source developers. It is written in JavaScript, HTML and CSS. Brackets is cross-platform, available for macOS, Windows, and most Linux distributions. The main purpose of brackets is its live HTML, CSS and JavaScript editing functionality Brackets Quick edit enables inline editing of CSS, Color Property, and JavaScript elements for developers. This built-in feature can be applied to multiple functions or properties simultaneously and all updates are applied directly to the file

associated with the changed elements. Live Preview, this feature also pushes code edits instantly to the browser to present an updated webpage as the developers modify the code.

3.3.2 WAMP

It Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). It is often used for web development and internal testing, but may also be used to serve live websites.

The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet.

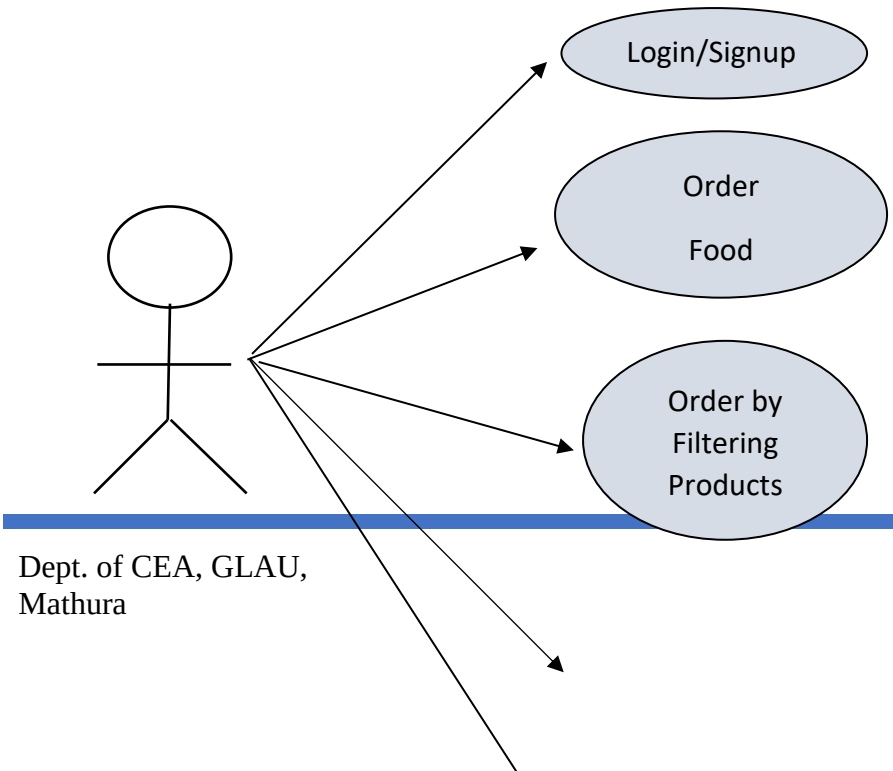
WAMP also includes MySQL and PHP, which are two of the most common technologies used for creating dynamic websites. MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.

Chapter 4

Software Design

4.1. Use case Diagram

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different **use** cases in which the user is involved.



Customer

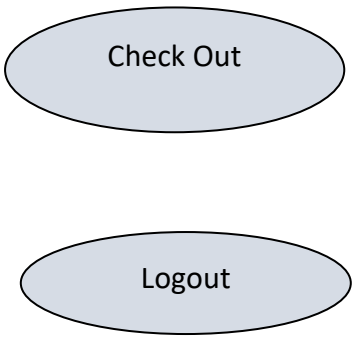


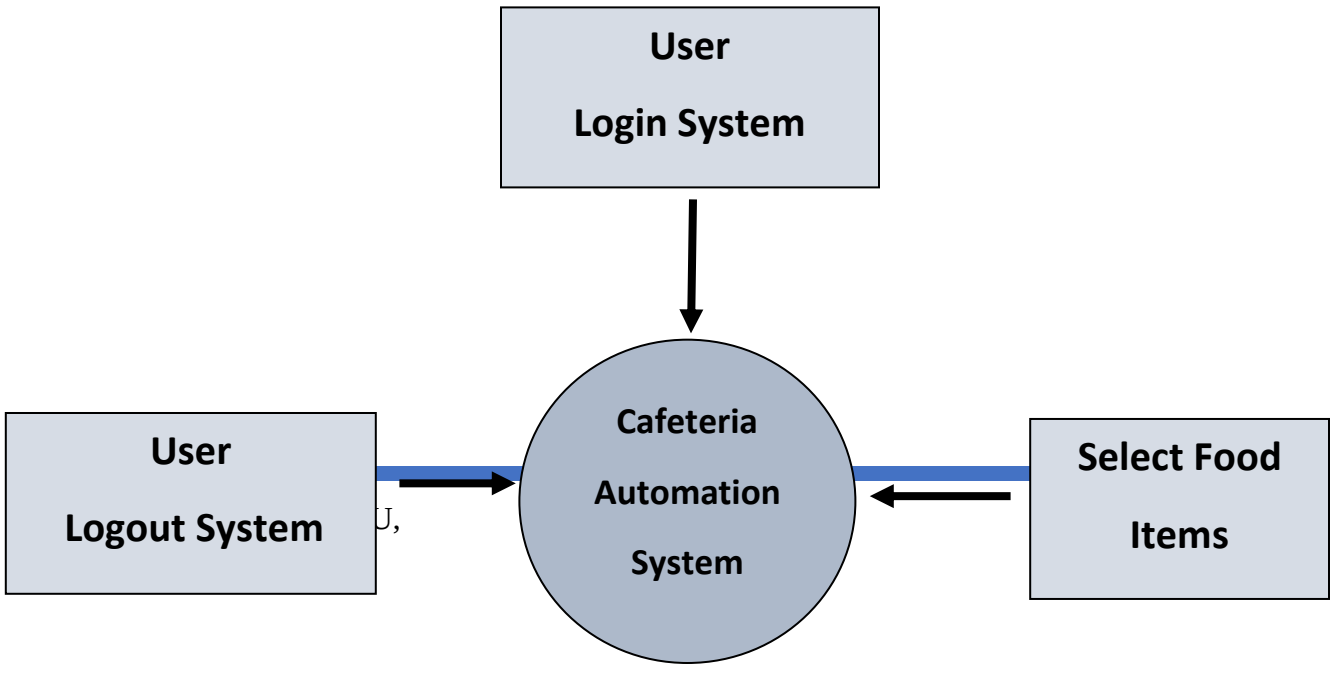
Fig 4.1- Use Case Diagram

4.2. Dataflow Diagrams

Data Flow diagrams show the flow of data from external entities into the system,

and from one process to another within the system.

1. Level -0 Diagram: The level 0 diagram provides a conceptual view of the process and its surrounding input, output and data stores. It is called context level Data flow diagram also.



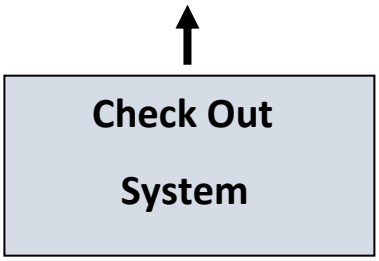


Fig. 4.2 Level Zero Dataflow Diagram (DFD)

4.3 Structure of Databases

4.3.1 Category Table Structure

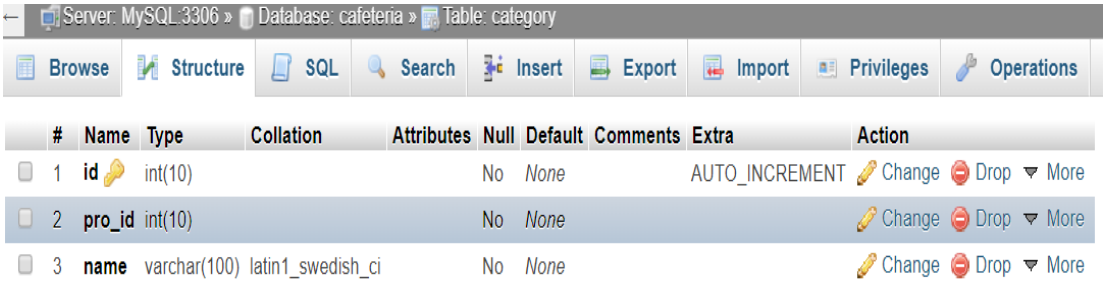


Fig 4.3- Category Table Structure

4.3.2 User Table Structure

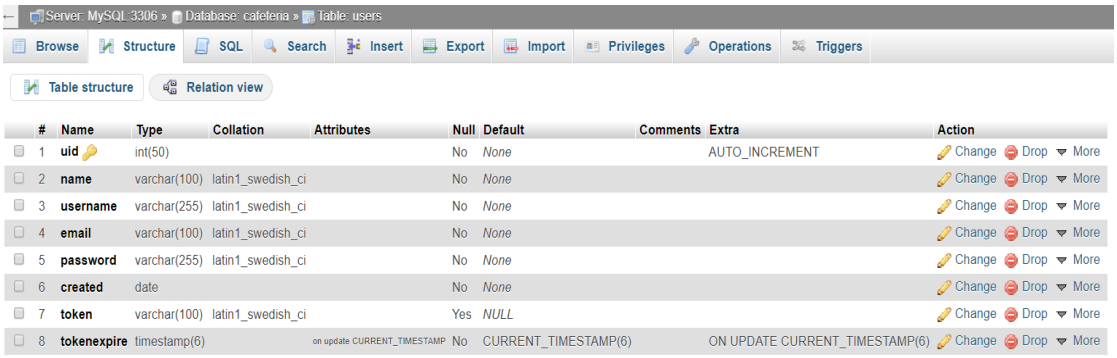
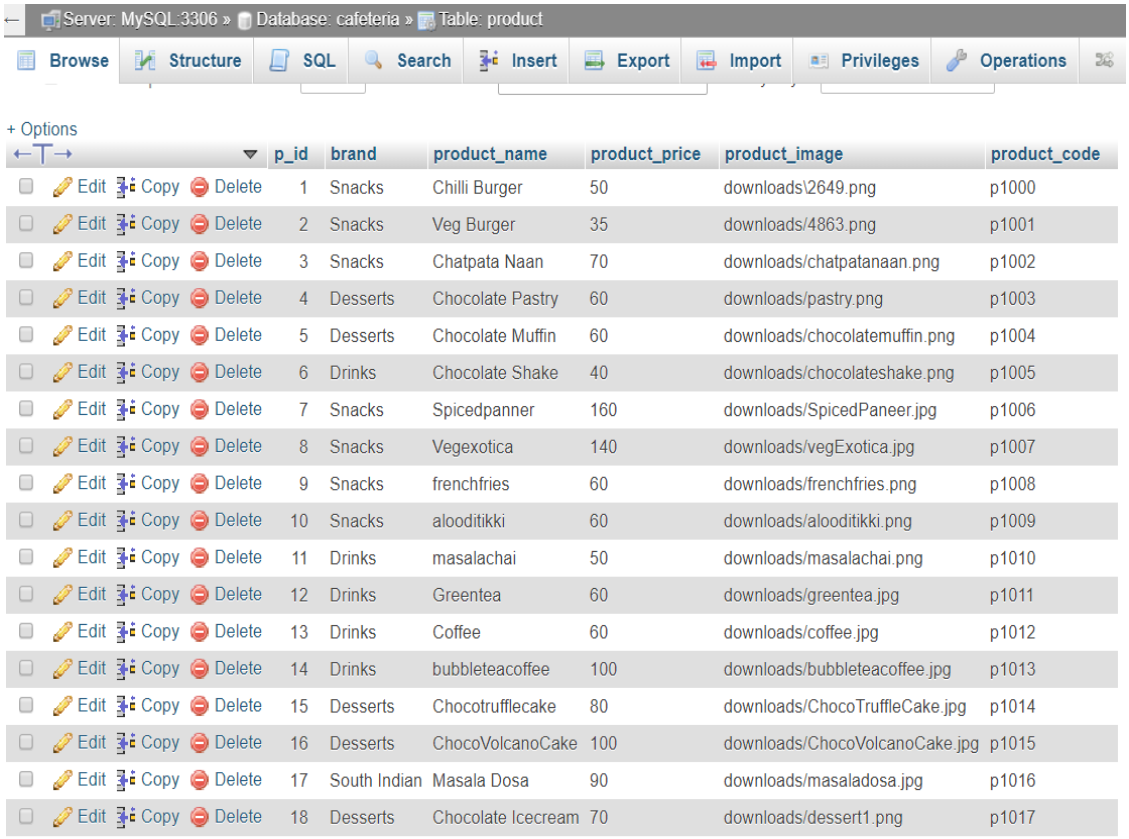


Fig 4.4- User Table Structure

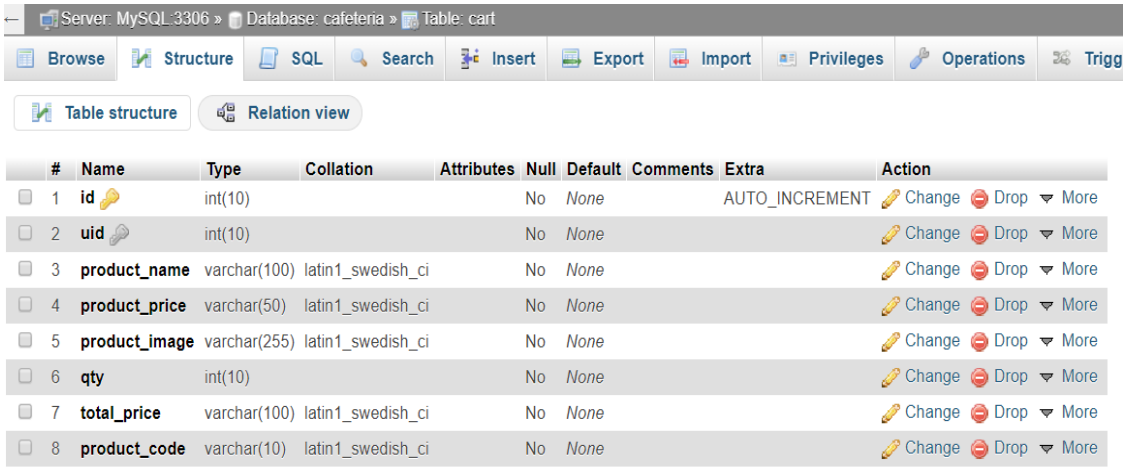
4.3.3 Product Table Structure



	p_id	brand	product_name	product_price	product_image	product_code
Edit Copy Delete	1	Snacks	Chilli Burger	50	downloads\2649.png	p1000
Edit Copy Delete	2	Snacks	Veg Burger	35	downloads/4863.png	p1001
Edit Copy Delete	3	Snacks	Chatpata Naan	70	downloads/chatpatanaan.png	p1002
Edit Copy Delete	4	Desserts	Chocolate Pastry	60	downloads/pastry.png	p1003
Edit Copy Delete	5	Desserts	Chocolate Muffin	60	downloads/chocolatemuffin.png	p1004
Edit Copy Delete	6	Drinks	Chocolate Shake	40	downloads/chocolateshake.png	p1005
Edit Copy Delete	7	Snacks	Spicedpanner	160	downloads/SpicedPaneer.jpg	p1006
Edit Copy Delete	8	Snacks	Vegexotica	140	downloads/vegExotica.jpg	p1007
Edit Copy Delete	9	Snacks	frenchfries	60	downloads/frenchfries.png	p1008
Edit Copy Delete	10	Snacks	alooditikki	60	downloads/alooditikki.png	p1009
Edit Copy Delete	11	Drinks	masalachai	50	downloads/masalachai.png	p1010
Edit Copy Delete	12	Drinks	Greentea	60	downloads/greentea.jpg	p1011
Edit Copy Delete	13	Drinks	Coffee	60	downloads/coffee.jpg	p1012
Edit Copy Delete	14	Drinks	bubbleteacoffee	100	downloads/bubbleteacoffee.jpg	p1013
Edit Copy Delete	15	Desserts	Chocotrufflecake	80	downloads/ChocoTruffleCake.jpg	p1014
Edit Copy Delete	16	Desserts	ChocoVolcanoCake	100	downloads/ChocoVolcanoCake.jpg	p1015
Edit Copy Delete	17	South Indian	Masala Dosa	90	downloads/masaladosa.jpg	p1016
Edit Copy Delete	18	Desserts	Chocolate Icecream	70	downloads/dessert1.png	p1017

Fig 4.5- Product table Structure

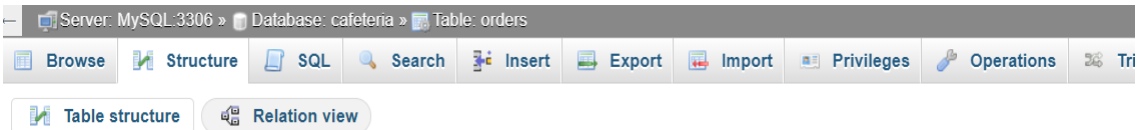
4.3.4 Cart Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	uid	int(10)			No	None			Change Drop More
3	product_name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	product_price	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
5	product_image	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
6	qty	int(10)			No	None			Change Drop More
7	total_price	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
8	product_code	varchar(10)	latin1_swedish_ci		No	None			Change Drop More

Fig 4.6- Cart Table Structure

4.3.5 Orders Table Structure



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
3	email	varchar(100)	latin1_swedish_ci		No	None			Change Drop More
4	phone	varchar(20)	latin1_swedish_ci		No	None			Change Drop More
5	address	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
6	pmode	varchar(50)	latin1_swedish_ci		No	None			Change Drop More
7	products	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
8	amount_paid	varchar(100)	latin1_swedish_ci		No	None			Change Drop More

Fig 4.7- Order Table Structure

Chapter 5

Implementation

5.1 Storing the files in Main Directory

First of all, We have to store all the files of our project in the htdocs folder of the wamp server so that wamp server can easily fetch the files on a localhost otherwise the localhost will not be able to detect the files. Files include images, php files, html files, css files, bootstrap files etc.

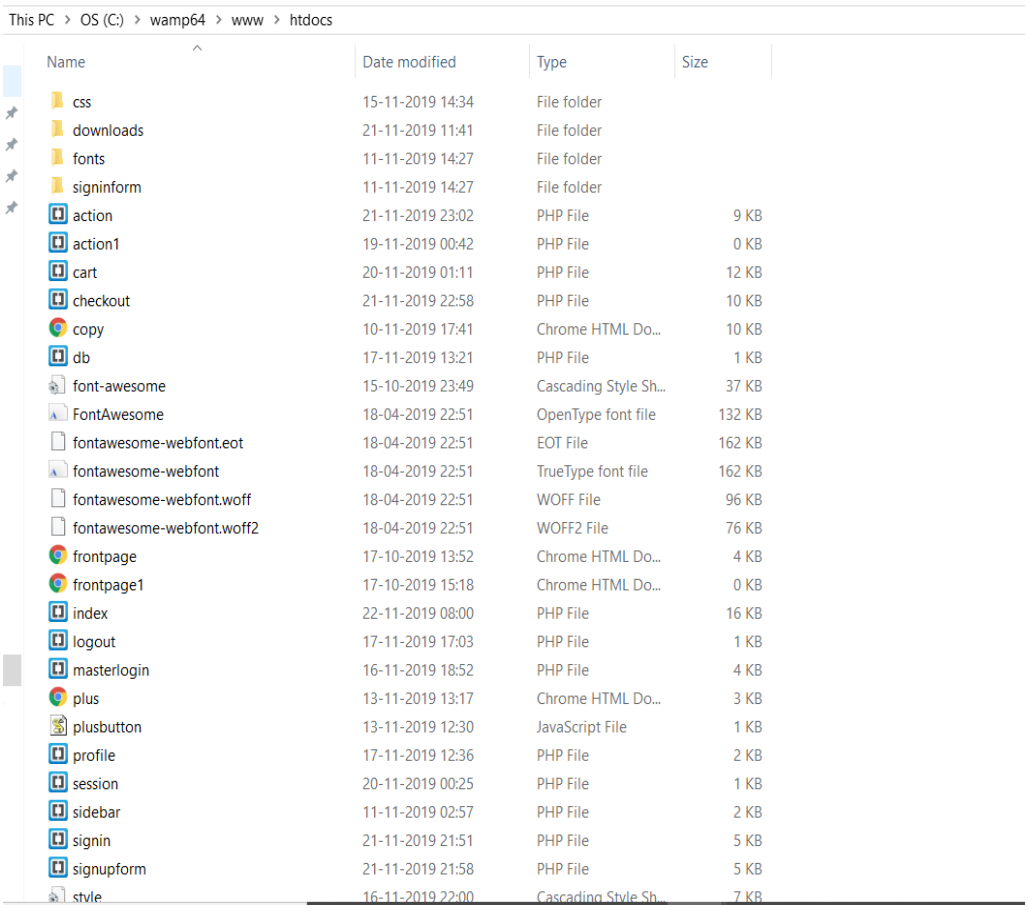


Fig 5.1 – Storing the files in main directory

5.1.1 QR Code

5.2 Login Page

To design login page we used html and css for frontend and PHP for backend. Here are the layout and code for login.php through this login page any user can login.

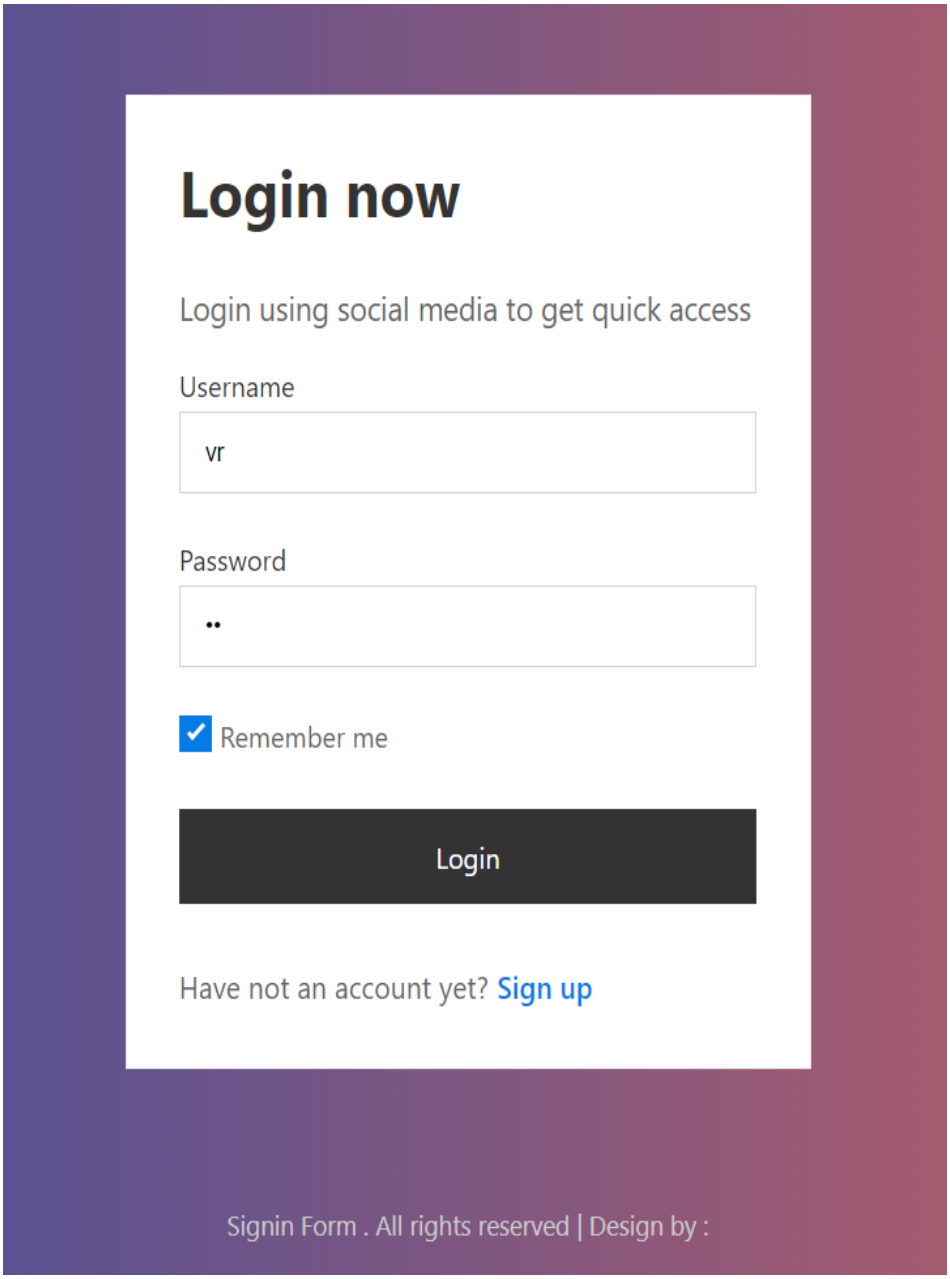
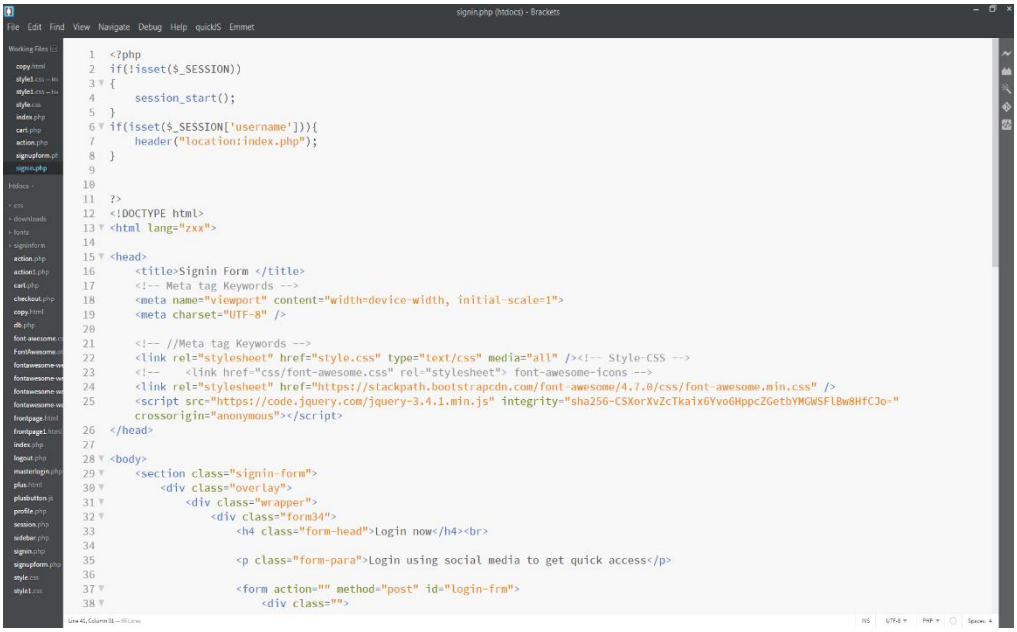
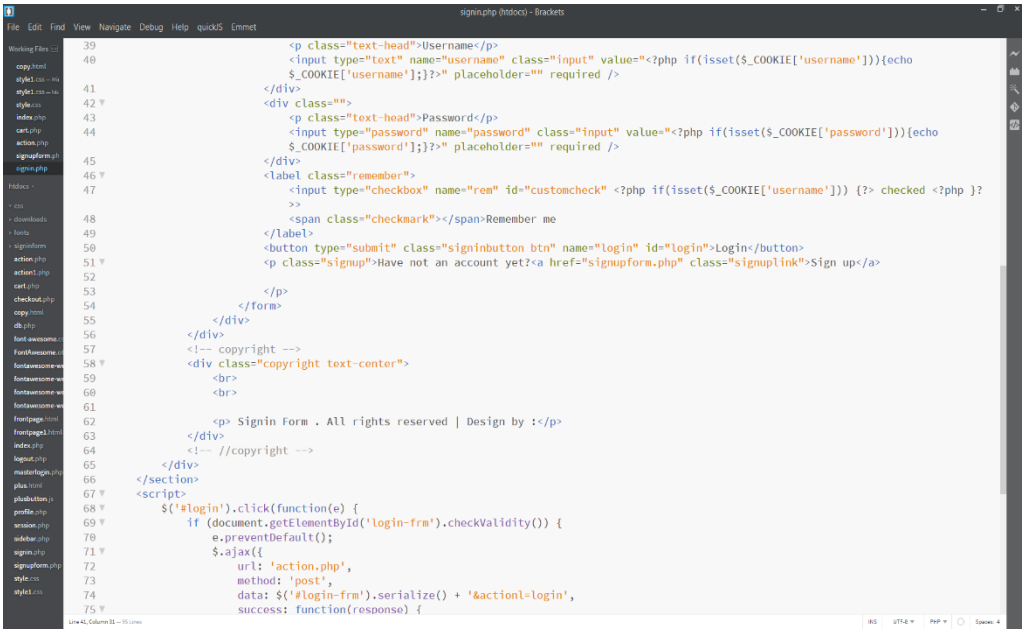


Fig 5.2 – login Page



```php
<?php
if(!isset($_SESSION))
{
    session_start();
}
if(isset($_SESSION['username'])){
    header("location:index.php");
}


?>
<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>Signin Form </title>
    <!-- Meta tag Keywords -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta charset="UTF-8" />

    <!-- //Meta tag Keywords -->
    <link rel="stylesheet" href="style.css" type="text/css" media="all" /><!-- Style-CSS -->
    <!--    <link href="css/font-awesome.css" rel="stylesheet"> font-awesome-icons -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" />
    <script src="https://code.jquery.com/jquery-3.4.1.min.js" integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
    crossorigin="anonymous"></script>
</head>

<body>
    <section class="signin-form">
        <div class="overlay">
            <div class="wrapper">
                <div class="form34">
                    <h4 class="form-head">Login now</h4><br>

                    <p class="form-para">Login using social media to get quick access</p>

                    <form action="" method="post" id="login-frm">
                        <div class="">
```

Fig 5.3 – Coding of login Page



```php
                            <p class="text-head">Username</p>
                            <input type="text" name="username" class="input" value="<?php if(isset($_COOKIE['username'])){echo
                            $_COOKIE['username'];}?>" placeholder="" required />
                        </div>
                        <div class="">
                            <p class="text-head">Password</p>
                            <input type="password" name="password" class="input" value="<?php if(isset($_COOKIE['password'])){echo
                            $_COOKIE['password'];}?>" placeholder="" required />
                        </div>
                        <label class="remember">
                            <input type="checkbox" name="rem" id="customcheck" <?php if(isset($_COOKIE['username'])) {?> checked <?php }?
                            >
                            <span class="checkmark"></span>Remember me
                        </label>
                        <button type="submit" class="signinbutton btn" name="login" id="login">Login</button>
                        <p class="signup">Have not an account yet?<a href="signupform.php" class="signuplink">Sign up</a>

                        </p>
                    </form>
                </div>
            </div>
            <!-- copyright -->
            <div class="copyright text-center">
                <br>
                <br>

                <p> Signin Form . All rights reserved | Design by :</p>
            </div>
            <!-- //copyright -->
        </div>
    </section>
    <script>
        $('#login').click(function(e) {
            if (document.getElementById('login-frm').checkValidity()) {
                e.preventDefault();
                $.ajax({
                    url: 'action.php',
                    method: 'post',
                    data: $('#login-frm').serialize() + '&action=login',
                    success: function(response) {
```

Fig 5.4 – Coding of login Page

5.3 Signup Page

To design Signup page we used html and css for frontend and PHP for backend. Here are the layout and code for signup.php through this login page any user can signup.

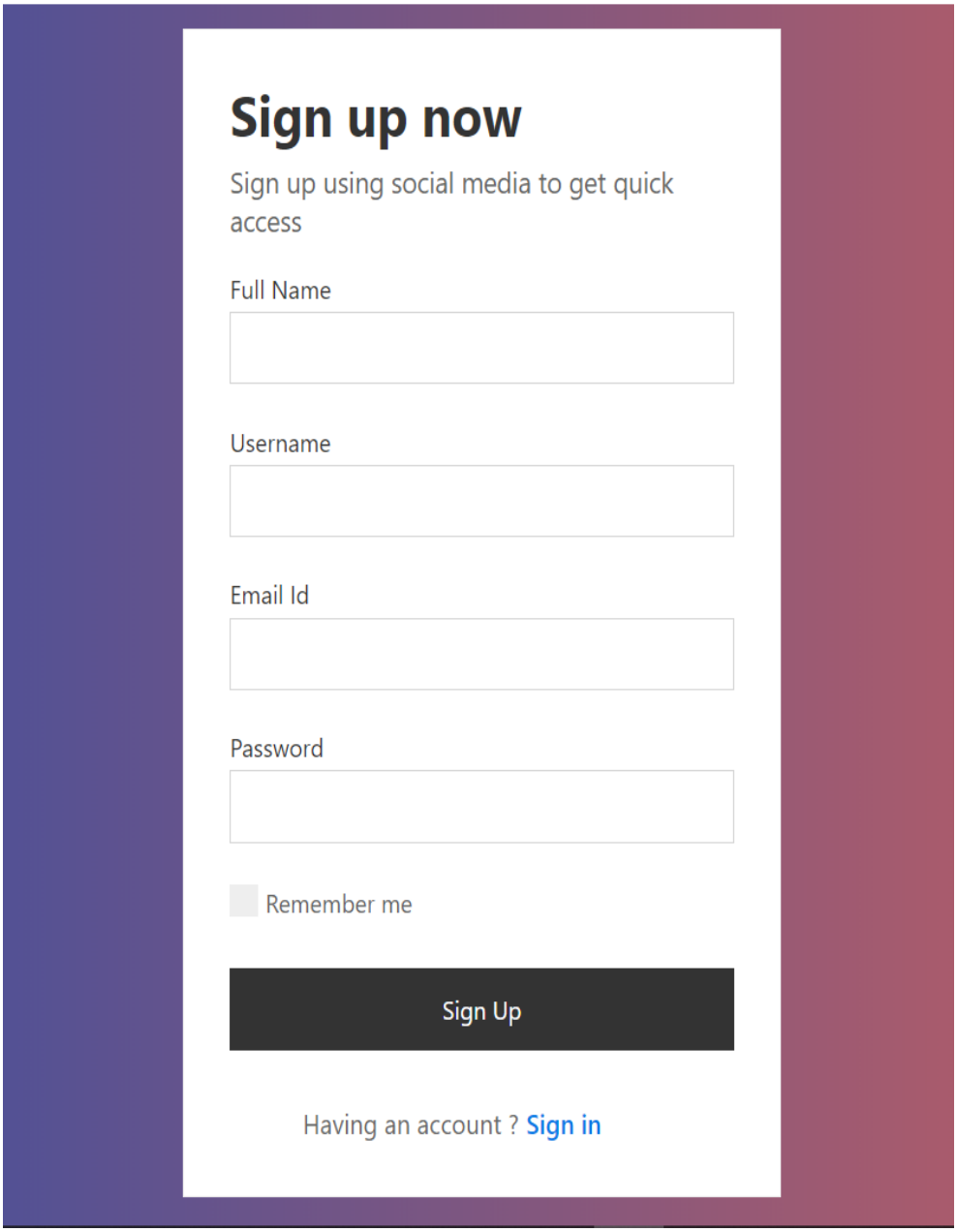
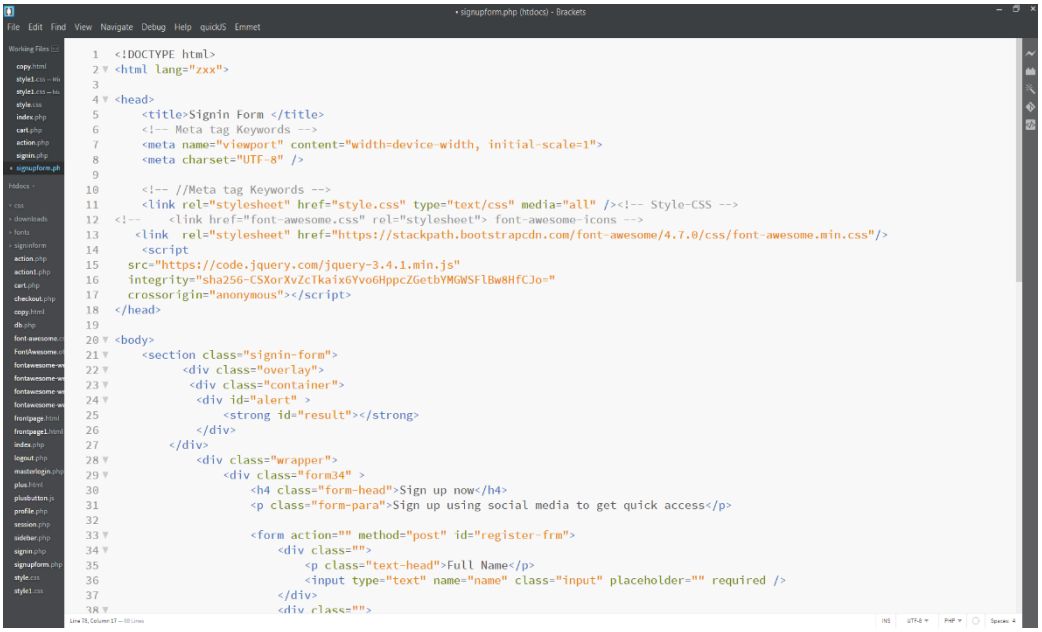
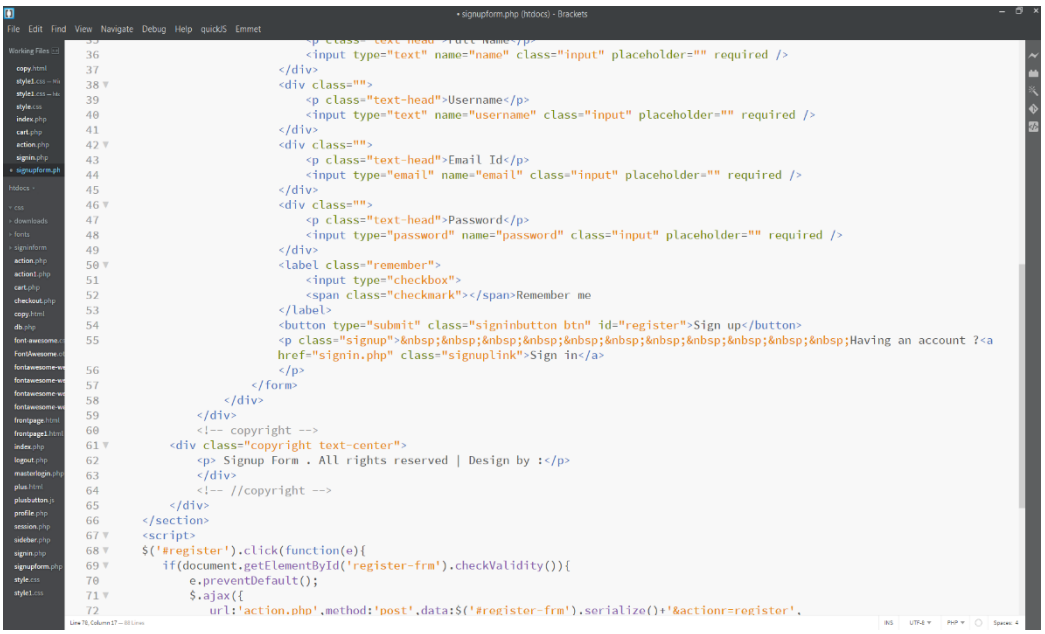


Fig 5.3 – Signup Page



```
<!DOCTYPE html>
<html lang="zxx">

<head>
    <title>Signin Form </title>
    <!-- Meta tag Keywords -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta charset="UTF-8" />

    <!-- //Meta tag Keywords -->
    <link rel="stylesheet" href="style.css" type="text/css" media="all" /><!-- Style-CSS -->
<!--    <link href="font-awesome.css" rel="stylesheet"> font-awesome-icons -->
   <link  rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"/>
    <script
  src="https://code.jquery.com/jquery-3.4.1.min.js"
  integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
  crossorigin="anonymous"></script>
</head>

<body>
    <section class="signin-form">
          <div class="overlay">
           <div class="container">
            <div id="alert" >
                <strong id="result"></strong>
            </div>
        </div>
            <div class="wrapper">
                <div class="form34" >
                    <h4 class="form-head">Sign up now</h4>
                    <p class="form-para">Sign up using social media to get quick access</p>

                    <form action="" method="post" id="register-frm">
                        <div class="">
                            <p class="text-head">Full Name</p>
                            <input type="text" name="name" class="input" placeholder="" required />
                        </div>
                        <div class="">
```

Fig 5.4 – Signup Page



```
                            <input type="text" name="name" class="input" placeholder="" required />
                        </div>
                        <div class="">
                            <p class="text-head">Username</p>
                            <input type="text" name="username" class="input" placeholder="" required />
                        </div>
                        <div class="">
                            <p class="text-head">Email Id</p>
                            <input type="email" name="email" class="input" placeholder="" required />
                        </div>
                        <div class="">
                            <p class="text-head">Password</p>
                            <input type="password" name="password" class="input" placeholder="" required />
                        </div>
                        <label class="remember">
                            <input type="checkbox">
                            <span class="checkmark"></span>Remember me
                        </label>
                        <button type="submit" class="signinbutton btn" id="register">Sign up</button>
                        <p class="signup">           Having an account ?<a
                        href="signin.php" class="signuplink">Sign in</a>
                        </p>
                    </form>
                </div>
            </div>
            <!-- copyright -->
        <div class="copyright text-center">
            <p> Signup Form . All rights reserved | Design by :</p>
            </div>
            <!-- //copyright -->
        </div>
    </section>
    <script>
    $('#register').click(function(e){
       if(document.getElementById('register-frm').checkValidity()){
           e.preventDefault();
           $.ajax({
             url:'action.php',method:'post',data:$('#register-frm').serialize()+'&action=register',
```

Fig 5.5 – Signup Page

5.4 Main Page

It is a responsive page of our website which contains all the functionalities that a general food website should have in it.

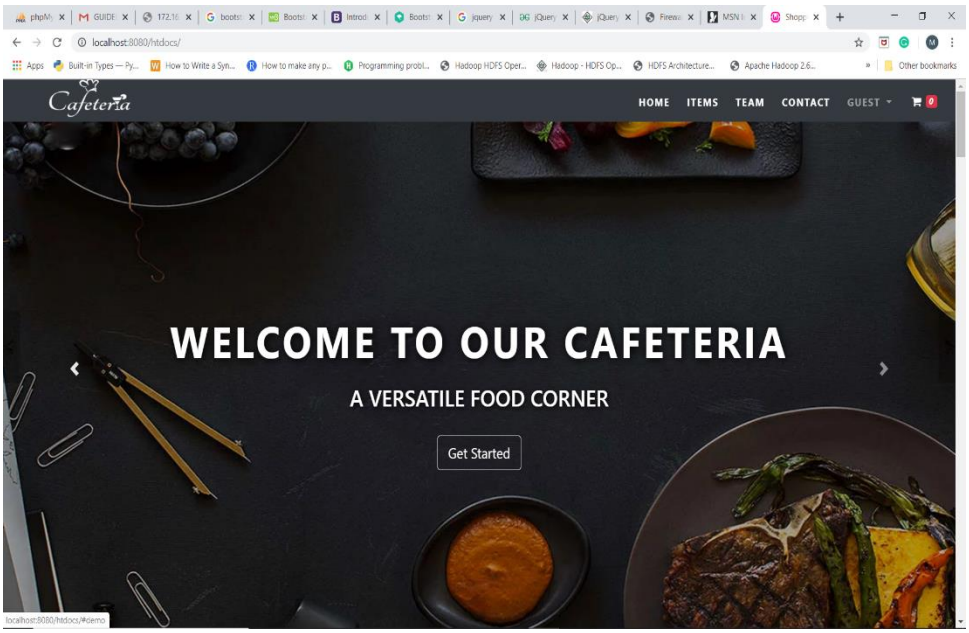


Fig 5.5 – Main Page

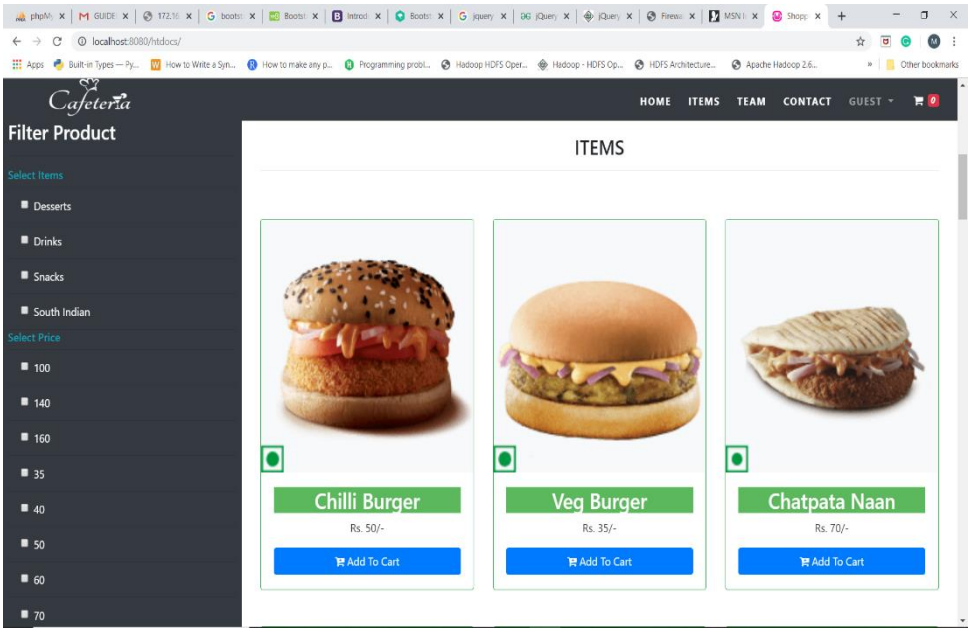


Fig 5.6 – Main Page

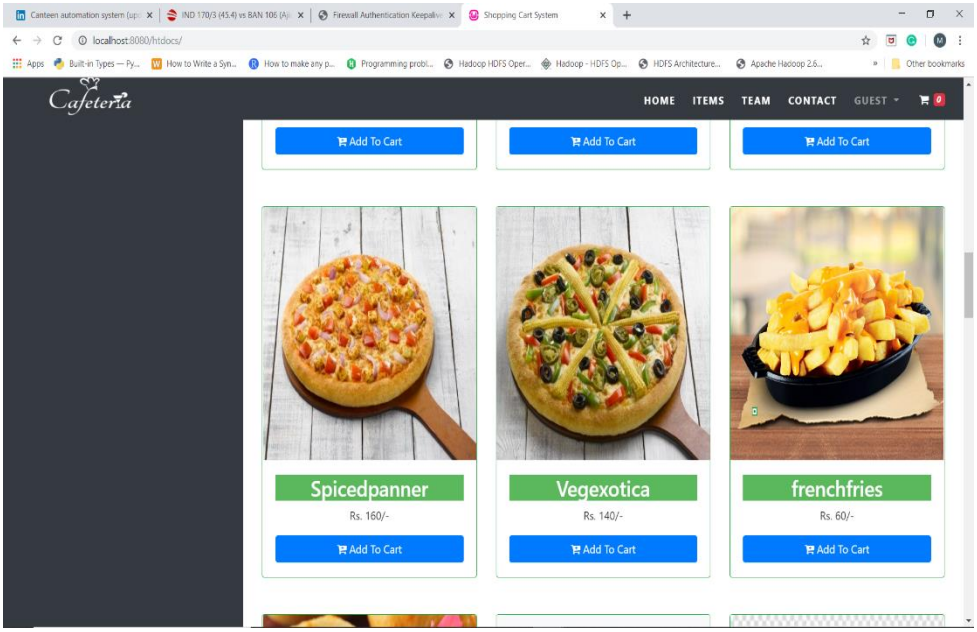


Fig 5.7 – Main Page

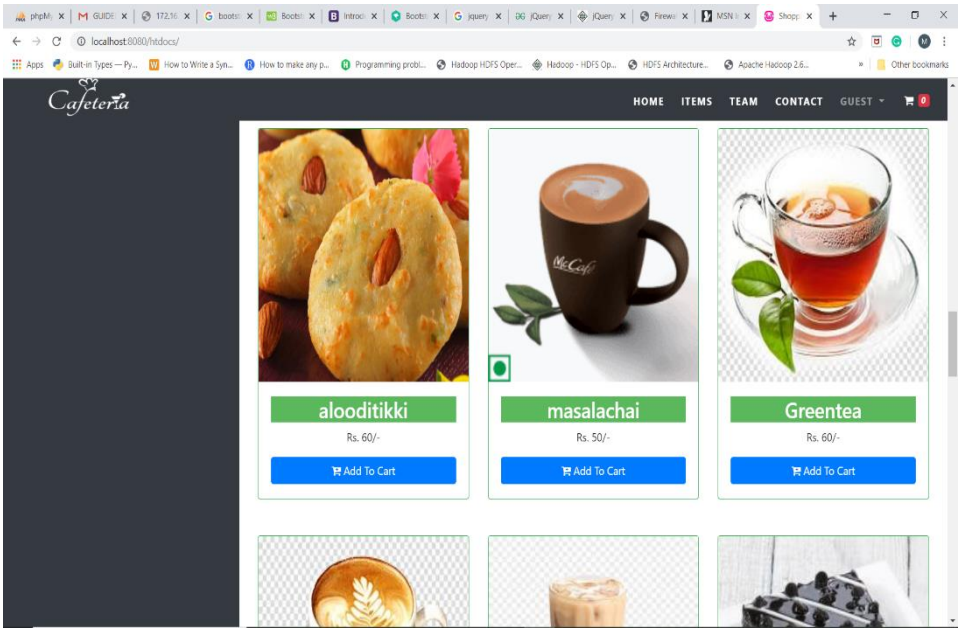


Fig 5.7– Main Page

5.5 Checkout Page

In this page, the items which has been selected by the respective user will be shown the cart.

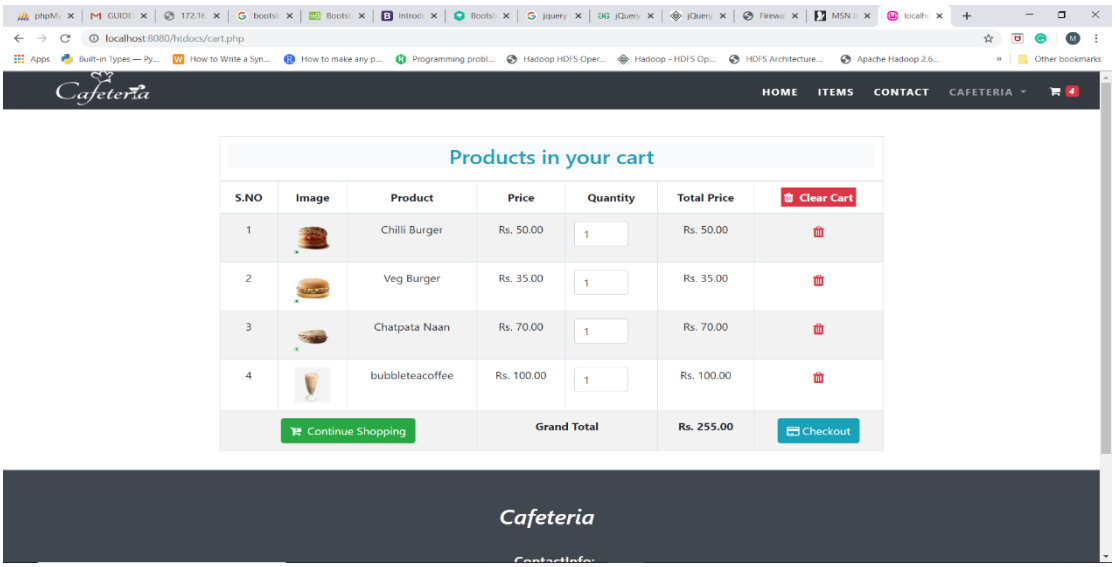


Fig 5.10– Checkout Page

5.6 Enter details Page

The user will have to enter the personal details and Table Number to proceed further. This data will get stored in cart table and will be displayed in greeting page.

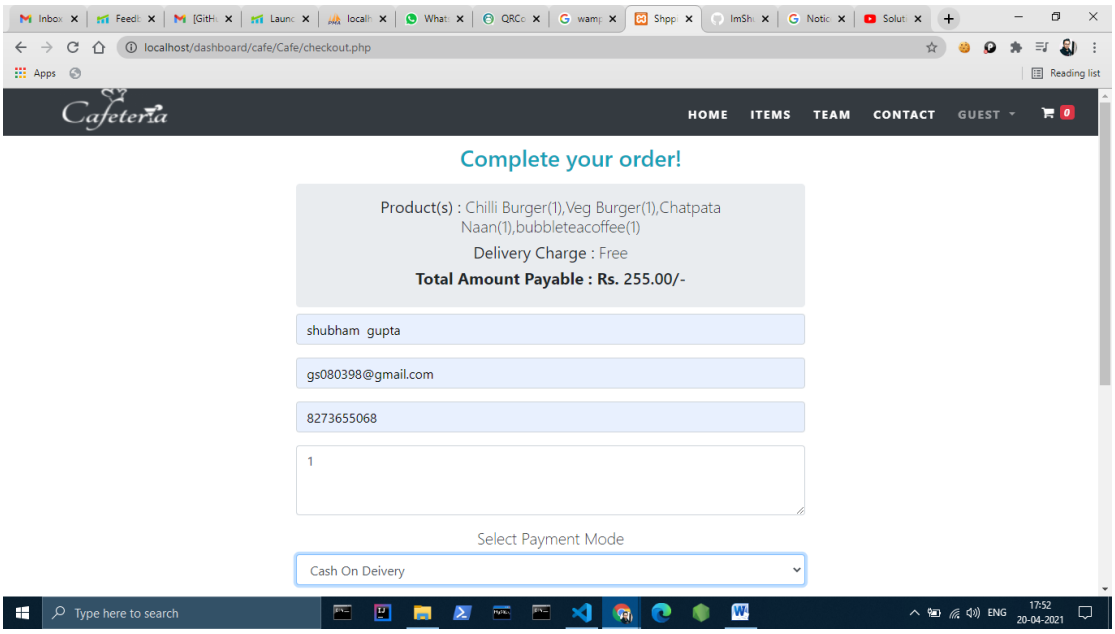


Fig 5.10- Place Order Page

5.7 Greeting Page

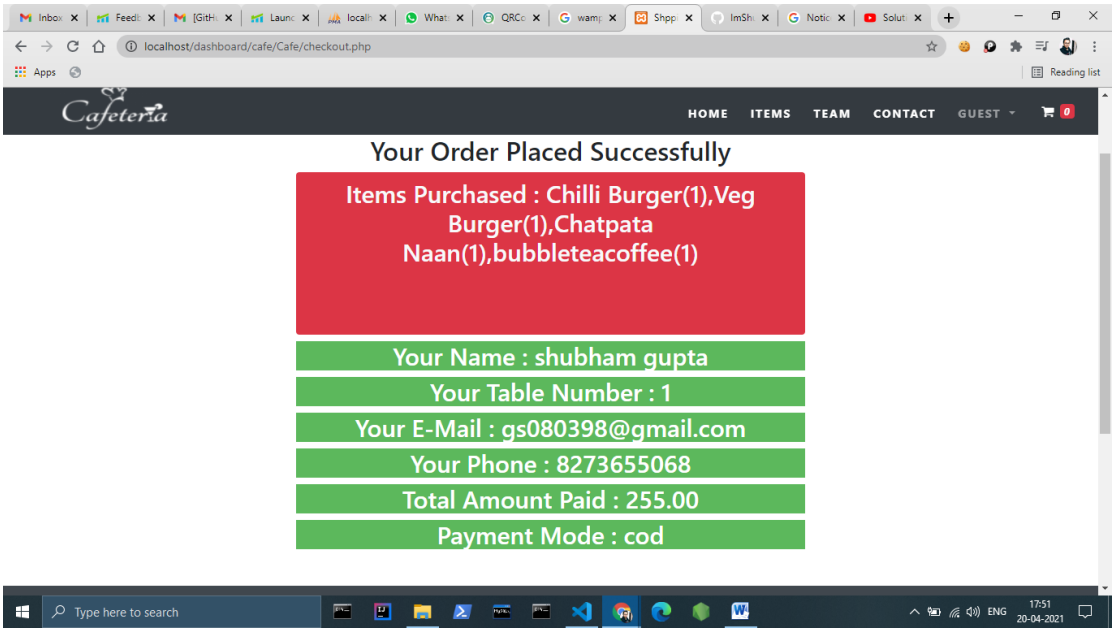


Fig 5.11-Greeting Page

CHAPTER 6

Validation

6.1 Signup Page Validation

```
<script>
   $('#register').click(function(e){
      if(document.getElementById('register-frm').checkValidity()){
         e.preventDefault();
         $.ajax({
           url:'action.php',method:'post',data:$('#register-
frm').serialize()+'&actionr=register',
            success:function(response){
               alert(response);
//                $("#alert").show();
//                $('#result').html(response);
//
            }

         });
      }
    return true;
   });

   </script>
```

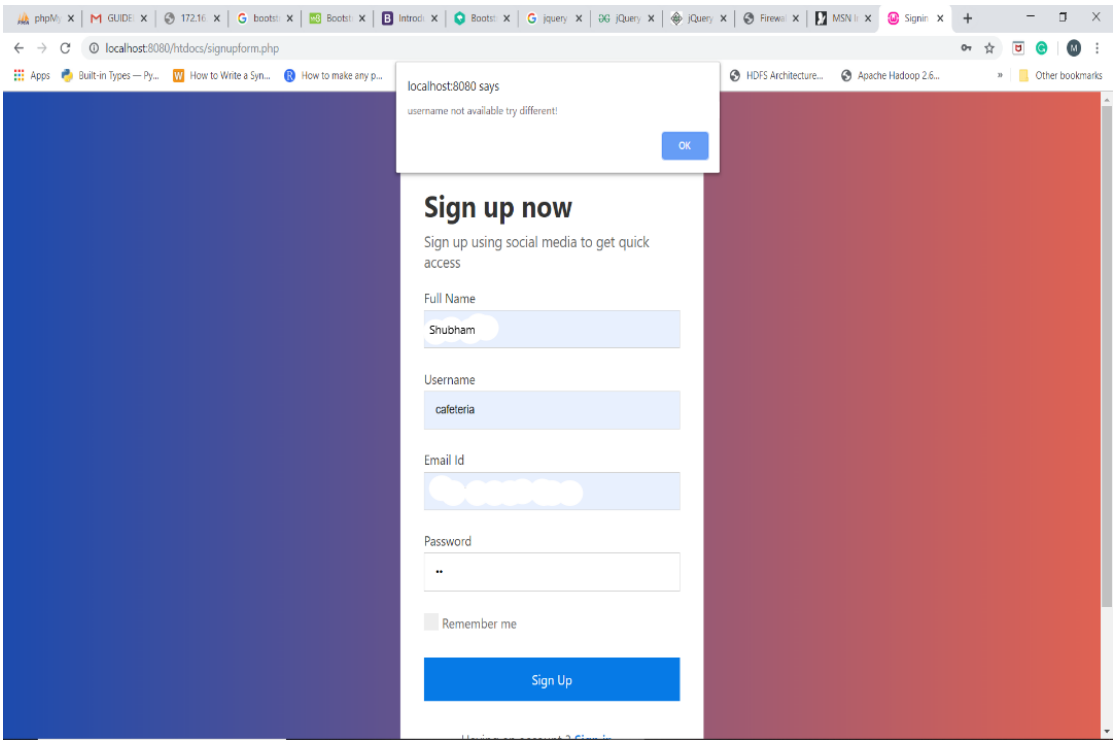


Fig 6.1- Username error message

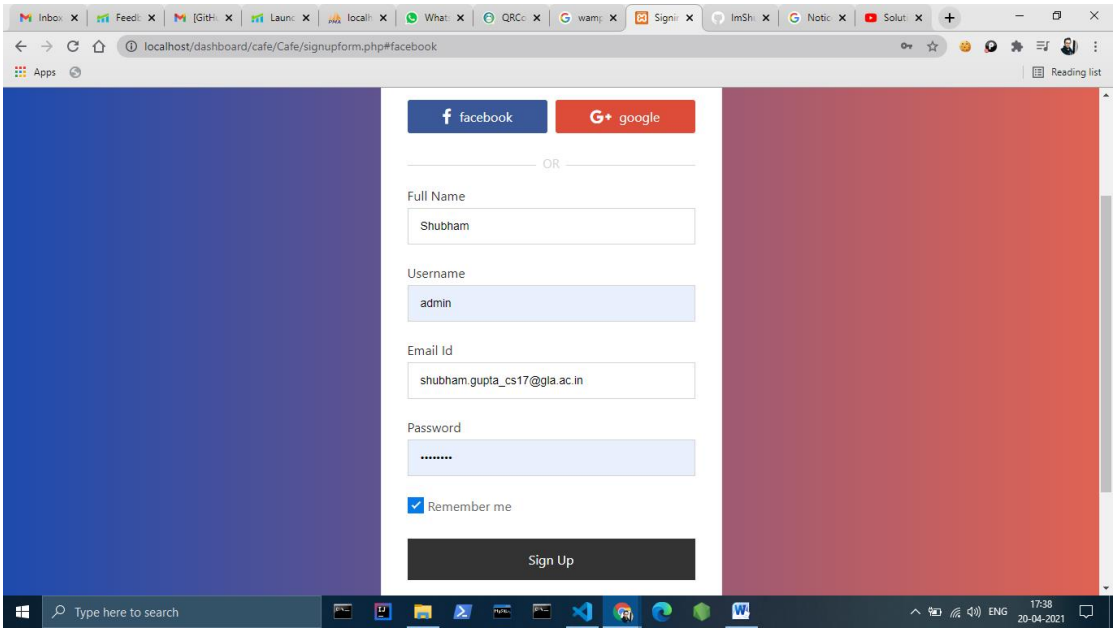


Fig 6.2- Email error message

6.2 Signin Page Validation

```
<script>
      $('#login').click(function(e) {
         if  (document.getElementById('login-frm').checkValidity())
{
            e.preventDefault();
            $.ajax({
               url: 'action.php',
               method: 'post',
               data: $('#login-frm').serialize() + '&actionl=login',
               success: function(response) {
                  if (response === "ok") {
                     window.location = 'index.php';
                  } else {
                     alert(response);
                     $("#alert").show();
                     $('#result').html(response);

                  }
               }

            });
         }
         return true;
      });

   </script>
```

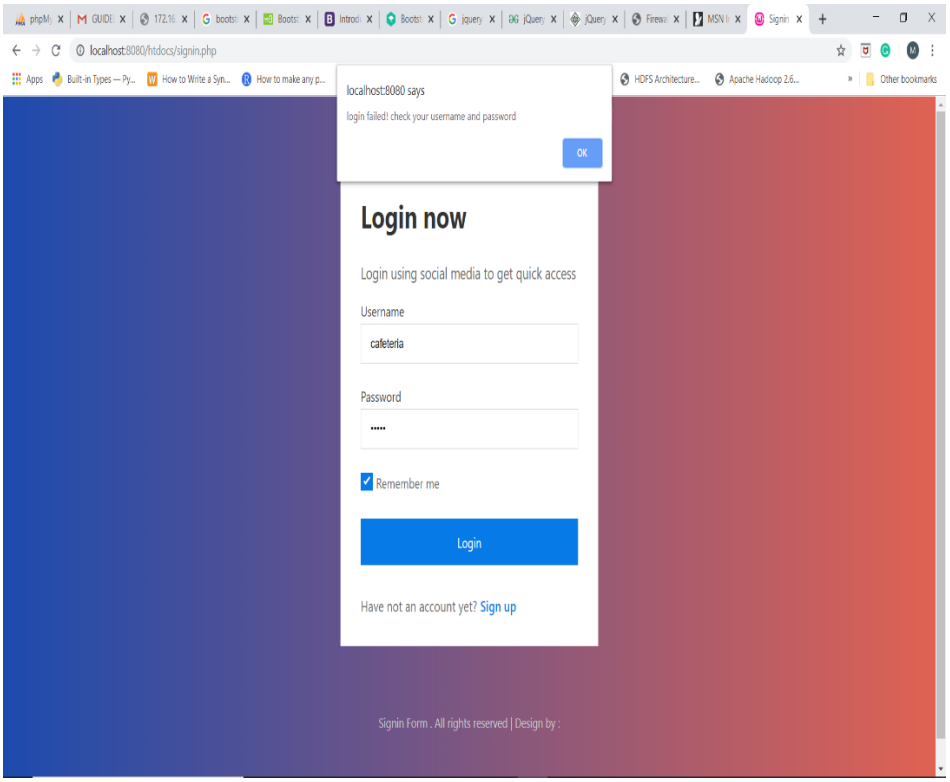


Fig 6.2- Login failed message

CHAPTER 7

Conclusion

After the completion of project, customer can view the overall menu of our cafeteria but cannot order any item without registration. New customer can create a new account by going on sign up page. After registration, he can login and now can add any item in any quantity in the shopping cart. He can also apply different filters like price, category of food items etc. Quantity of items can be increased or decreased in the final cart page. Even he can delete any item or can clear overall cart items in one go by clicking the option clear cart. He can place the order only through cash on delivery mode. After placing the order he can view the summary with an order id which is the table number of that particular customer.

CHAPTER 8

Future Scope

This project is not fully completed yet. QR-BASED Cafeteria Automation System has more than this. If we continue this project in future then we can add following functionalities in it.

1. Admin Panel which can update the items like changing the price, quantity etc, make any item unavailable if not present, can see the order of customers.
2. Payment Gateway which will enable the people to pay the amount through debit cards, credit cards or net banking etc.
3. Discount coupons can be generated for the customers so that more number of customers can visit and order food items on our website.

CHAPTER 9

References/Bibliography

This project is the output of work of our team member and guidance of our mentor. But there are some website which helped us a lot. They are :

1. https://www.youtube.com/watch?v=5YgscpAC0gE
2. https://www.youtube.com/watch?v=cRoBoztHXr0
3. https://www.w3schools.com/bootstrap4/
4. https://www.w3schools.com/html/
5. https://www.w3schools.com/css/
6. https://www.w3schools.com/js/
7. https://www.w3schools.com/jquery