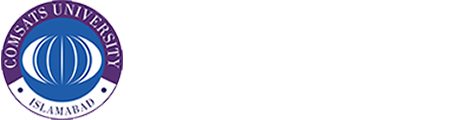
**COMSATS University, Islamabad**

***Title****: Cozy Corner Eatery*

***Course****: Web Technology*

***Project***



**Submitted by:** Shameen Shahid (FA19-BCS-074)

**BCS-5B**

**Submitted to**: Sir Waseem Abbas Hashmi

**Date**: Jan 18th, 2022

**Department of Computer Science**

**Islamabad Campus**

***Code:***

***Admin Code:***

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <meta http-equiv="X-UA-Compatible" content="IE=edge">
6. <meta name="viewport" content="width=device-width, initial-scale=1.0">
7. <title>Admin login</title>
8. </head>
9. <style>
10. \*{
11. margin:0;
12. padding:0;
13. box-sizing:border-box;
14. font-family: 'Popins',sans-serif;
15. }
16. section{
17. position:relative;
18. width:100%;
19. height:100vh;
20. display:flex;
21. }
22. section .imgBx{
23. position:relative;
24. width:50%;
25. height:100%;
26. }
27. section .imgBx:before{
28. content:'';
29. position:absolute;
30. top:0;
31. left:0;
32. width:100%;
33. height:100%;
34. background:linear-gradient(225deg,#e91e63,#03a9f4);
35. z-index:1;
36. mix-blend-mode: screen;
37. }
38. section  .imgBx img{
39. position:absolute;
40. top:0;
41. left:0;
42. width:100%;
43. height:100%;
44. object-fit:cover;
45. }
46. section .contentBx{
47. display:flex;
48. justify-content:center;
49. align-items:center;
50. width:50%;
51. height:100%;
52. }
53. section .contentBx .formBx{
54. width: 50%;
55. }
56. section .contentBx .formBx h2{
57. color:#607d8b;
58. font-weight:600;
59. font-size:1.5em;
60. text-transform: uppercase;
61. margin-bottom:20px;
62. border-bottom: 4px solid #ff4584;
63. display:inline-block;
64. letter-spacing:1px;
66. }
67. section .contentBx.formBx .inputBx{
68. margin-bottom:20px;
69. }
70. section .contentBx .formBx  .inputBx span{
71. font-size:16px;
72. margin-bottom: 5px;
73. display: inline-block;
74. color:#607d8b;
75. font-weight: 300;
76. font-size: 16px;
77. letter-spacing: 1px;
78. }
79. section .contentBx .formBx  .inputBx input{
80. width:100%;
81. padding:10px 20px;
82. outline:none;
83. font-weight: 400;
84. border:1px solid #607d8b;
85. font-size: 16px;
86. letter-spacing: 1px;
87. color:#607d8b;
88. background:transparent;
89. border-radius:30px;
91. }
92. section .contentBx .formBx  .inputBx input[type="submit"]
93. {
94. padding:10px;
95. background: #ff4584;
96. color:#fff;
97. outline:none;
98. border:none;
99. font-weight:500;
100. cursor:pointer;
101. }
102. section .contentBx .formBx  .inputBx input[type="submit"]:hover{
103. background:#f53677;
104. }
105. section .contentBx .formBx .remember
106. {
107. margin-bottom: 10px;
108. color: #607d8b;
109. font-weight: 400;
110. font-size:14px;
111. }
112. #me{
113. font-family: 'Popins';
114. color: white;
115. }
116. #me
117. {
118. width:100%;
119. padding:10px 20px;
120. outline:none;
121. font-weight: 400;
122. border:1px solid #607d8b;
123. font-size: 16px;
124. letter-spacing: 1px;
125. background:transparent;
126. border-radius:30px;
127. padding:10px;
128. background: #ff4584;
129. outline:none;
130. border:none;
131. font-weight:500;
132. cursor:pointer;
133. }
134. #me{
135. background:#f53677;
136. }
137. @media (max-width: 768px)
138. {
139. section .imgBx{
140. position:absolute;
141. top:0;
142. left:0;
143. width:100%;
144. height:100%;
145. }
146. section .contentBx{
147. display:flex;
148. justify-content: center;
149. align-items:Center;
150. width:100%;
151. height:100%;
152. z-index:1;
153. }
154. section .contentBx .formBx{
155. width:100%;
156. padding:40px;
157. background:rgb(255 255 255 /0.9);
158. margin:50px;
159. }
160. section .contentBx.formBx h3{
161. color:#607d8b;
162. text-align: center;
163. margin: 30px 0 10px;
164. font-weight:500;
165. }
166. }
167. </style>
169. <body>
170. <section>
171. <div class="imgBx">
172. <img src="assets/images/adminrest.jpg">
173. </div>
174. <div class="contentBx">
175. <div class="formBx">
176. <h2>Admin Login</h2>
177. <form action="/admin" method="POST">
178. @csrf
179. <div class="inputBx">
180. <span>Username</span>
181. <input type="text" name="username" required="required"/><span id="uname"></span><br/>
182. </div>
183. <div class="inputBx">
184. <span>Password</span>
185. <input type="password" name="password" required="required"/><span id="password" ></span><br/>
186. </div>
187. <br>
188. <div class="remember">
189. <label><input type="checkbox" name="">Remember me</label>
190. </div>
191. <div class="inputBx">
192. <input type="submit" value="Login">
193. </div>
194. <br>
195. <br>
196. <div class="inputBx">
197. <button id="me" class="button" type="button"><a href="home" id="col">Home</a></button>
198. </div>
200. </form>
201. </div>
203. </div>
204. </section>
205. <?php
207. ?>
209. </body>
210. </html>
212. ***Products:***
213. <!DOCTYPE html>
214. <html lang="en">
215. <head>
216. <meta charset="UTF-8">
217. <meta http-equiv="X-UA-Compatible" content="IE=edge">
218. <meta name="viewport" content="width=
219. , initial-scale=1.0">
220. <title>Dishes</title>
221. </head>
222. <style>
223. \*{
224. margin:0;
225. padding: 0;
226. box-sizing:border-box;
227. font-family: 'Popins',sans-serif;
228. }
229. body{
230. display:flex;
231. height:100vh;
232. justify-content: center;
233. align-items:center;
234. background:linear-gradient(135deg, #71b7e6,#9b59b6);
235. }
236. .container{
237. max-width: 700px;
238. width:100%;
239. background:#fff;
240. padding:25px 30px;
241. border-radius:5px;
242. }
243. .container .title{
244. font-size:25px;
245. font-weight: 500;
246. position: relative;
247. }
248. .container .title::before{
249. content:'';
250. position: absolute;
251. left:0;
252. bottom:0;
253. height:3px;
254. width:30px;
255. background: #9b59b6;
257. }
258. .container form .user-details{
259. display: flex;
260. flex-wrap: wrap;
261. justify-content: space-between;
262. }
263. form .user-details .input-box{
264. margin: 20px 0 12px 0;
265. width: calc(100% / 2 - 20px);
266. }
267. .user-details .input-box .details{
268. display: block;
269. font-weight:  500;
270. margin-bottom:5px;
271. }
272. .user-details .input-box input{
273. height:45px;
274. width:100%;
275. outline:none;
276. border-radius: 5px;
277. border: 1px solid #ccc;
278. padding-left:15px;
279. font-size:16px;
280. border-bottom-width: 2px;
281. transition: all 0.3s ease;
282. }
283. .user-details .input-box input:focus,
284. .user-details .input-box input:valid{
285. border-color:#9b59b6;
286. }
287. form .button{
288. height: 45px;
289. margin: 45px 0;
290. }
291. form .button input{
292. height: 100%;
293. width: 100%;
294. outline: none;
295. color: #fff;
296. border:none;
297. font-size: 18px;
298. font-weight: 500;
299. border-radius: 5px;
300. letter-spacing: 1px;
301. background: linear-gradient(135deg ,#71b7e6,#9b59b6)
302. }
303. form .button input:hover{
304. background: linear-gradient(-135deg ,#71b7e6,#9b59b6)
305. }
306. @media (max-width: 584px)
307. {
308. .container{
309. max-width: 100%;
310. }
311. form .user-details .input-box{
312. margin-bottom: 15px;
313. width:100%;
314. }
315. .container form .user-details{
316. max-height: 300px;
317. overflow-y: scroll;
318. }
319. .user-details::-webkit-scrollbar{
320. width:8;
322. }
323. }
324. </style>
325. <body>
326. <div class="container">
327. <div class="title">Place order</div>
328. <form action="/addproduct" method="post">
329. @csrf
330. <div class="user-details">
331. <div class="input-box">
332. <span class="details">Dish name</span>
333. <input type="text" name="dish" palceholder="Enter dish name" required> <span id="dish"></span>
334. </div>
335. <div class="input-box">
336. <span class="details">Dish type</span>
337. <input type="text" name="dishtype" palceholder="Enter dish type" required><span id="dishtype"></span>
338. </div>
339. <div class="input-box">
340. <span class="details">Quantity</span>
341. <input type="number" name="Quantity" palceholder="Enter price" required><span id="quantity"></span>
342. </div>
343. <div class="input-box">
344. <span class="details">Special Instructions</span>
345. <input type="text" name="special" palceholder="Enter any special instruction" required><span id="special"></span>
346. </div>
347. </div>
348. <div class="button">
349. <input type="submit"value="Submit">
350. </div>
351. </form>
352. </div>
354. </body>
355. </html>

***Controllers:***

1. <?php
3. namespace App\Http\Controllers;
4. use Request;
5. use Illuminate\Support\Facades\DB;
7. class Homecontroller extends Controller{
9. public function index()
10. {
11. return view("home");
12. }
13. public function dashboard()
14. {
15. return view("admindashboard");
16. }
18. public function login()
19. {
20. return view("login");
21. }
22. public function registeration()
23. {
24. return view("registeration");
25. }
26. public function adminlogin()
27. {
28. return view("admin");
29. }
30. function register()
31. {
32. return view("registeration");
33. }
34. public function addproduct()
35. {
36. return view("addproduct");
37. }
38. public function updatedish()
39. {
40. return view("updateproduct");
41. }
42. public function deletedish()
43. {
44. return view("deleteproduct");
45. }
46. public function updatepass()
47. {
48. return view("updatepass");
49. }
50. public function menu()
51. {
52. return view("menu");
53. }
54. public function profile()
55. {
56. return view( "profile");
57. }
58. public function updatepassworduser()
59. {
60. return view("updatepassword");
61. }
62. public function updateemailuser()
63. {
64. return view("updateemail");
65. }
66. public function updateaddressuser()
67. {
68. return view("updateaddress");
69. }
70. public function order()
71. {
72. return view("addtocart");
73. }
75. function match()
76. {
77. $uname=Request::input('username');
78. $password=Request::input('password');
79. $login=DB:: select('select password from admin where username = ?',[$uname]);
80. if(count ($login) > 0)
81. {
82. foreach($login as $auth)
83. {
84. if(($auth ->password)==$password)
85. {
86. return redirect('admindashboard');
87. }
88. else{
89. $error='Password or user id does not match';
90. if (empty($error)){}
91. else{echo '<script>alert("Password or id does not match")</script>'; }
92. return redirect('admin')->with('error',$error);
93. }
94. }
95. }
96. }
97. function store()
98. {
99. $firstname= Request::input('firstname');
100. $lastname= Request::input('lastname');
101. $username= Request::input('username');
102. $address= Request::input('address');
103. $email= Request::input('email');
104. $password= Request::input('password');
106. DB::insert('insert into customers(firstname,lastname,username,address,email,password) values(?,?,?,?,?,?)',[$firstname, $lastname,$username,$address,$email,$password]);
108. return redirect('login');
110. }
111. function adding()
112. {
113. $dish= Request::input('dish');
114. $dishtype=Request::input('dishtype');
115. $price= Request::input('price');
116. $chef= Request::input('chef');
118. DB::insert('insert into products(dish\_name,dish\_type,price,chef) values(?,?,?,?)',[$dish,$dishtype,$price,$chef]);
120. return redirect('admin');
122. }
123. function reviews()
124. {
125. $name= Request::input('name');
126. $email=Request::input('email');
127. $message= Request::input('message');
129. DB::insert('insert into reviews(name,email,message) values(?,?,?)',[$name,$email,$message]);
131. return redirect('profile');
133. }
134. function makereservation()
135. {
136. $name= Request::input('name');
137. $email=Request::input('email');
138. $phone= Request::input('phone');
139. $number\_guests=Request::input('number\_guests');
140. $date=Request::input('date');
141. $time=Request::input('time');
142. $message=Request::input('message');
144. DB::insert('insert into tablereserve(name,email,phone,guests,datee,time,message) values(?,?,?,?,?,?,?)',[$name,$email,$phone, $number\_guests,$date,$time,$message]);
146. return redirect('home');
148. }
150. public function updatedishrecord(){
151. $dish = Request::input('dish');
152. $price = Request::input('price');
153. DB::update('update products set  price=? where dish\_name= ?', [$price, $dish]);
154. return redirect('admindashboard');
155. }
156. public function deletedishrecord(){
157. $dish = Request::input('dish');
158. $dishtype = Request::input('dishtype');
159. DB::delete('delete from products  where dish\_name= ?,dish\_type=?', [$dish, $dishtype]);
160. return redirect('admindashboard');
161. }
163. public function updatepassword(){
164. $password = Request::input('password');
165. $new\_password = Request::input('new\_password');
166. DB::update('update admin set  password=? where password= ?', [$new\_password, $password]);
167. return redirect('admin');
168. }
170. public function updateuserpassword(){
171. $password = Request::input('password');
172. $new\_password = Request::input('new\_password');
173. DB::update('update customers set  password=? where password= ?', [$new\_password, $password]);
174. return redirect('updatepassword');
175. }
177. public function updateuseremail(){
178. $name=Request::input('name');
179. $email = Request::input('email');
180. $new\_email = Request::input('new\_email');
181. $login=DB:: select('select username from customers where username = ?',[$name]);
182. $login2=DB:: select('select email from customers where email = ?',[$email]);
183. if(count ($login) > 0)
184. {
185. foreach($login as $auth)
186. {
187. foreach($login2 as $auth2)
188. {
190. if(($auth ->username)==$name)
191. {
193. if(($auth2 ->email)==$email)
194. {
195. DB::update('update customers set  email=? where email= ?', [$new\_email, $email]);
196. return redirect("profile");
197. }
198. else
199. {
200. $error='Password or user id does not match';
201. if (empty($error)){}
202. else{echo '<script>alert("Password or id does not match")</script>'; }
203. return redirect('admin')->with('error',$error);
204. }
205. }
206. else{
207. $error='Password or user id does not match';
208. if (empty($error)){}
209. else{echo '<script>alert("Password or id does not match")</script>'; }
210. return redirect('admin')->with('error',$error);
211. }
212. }
213. }
214. }
215. }
217. public function updateuseraddress(){
218. $username=Request::input('username');
219. $address = Request::input('address');
220. $new\_address = Request::input('new\_address');
221. $login=DB:: select('select username from customers where username = ?',[$username]);
222. $login2=DB:: select('select address from customers where address = ?',[$address]);
223. if(count ($login) > 0)
224. {
225. foreach($login as $auth)
226. {
227. foreach($login2 as $auth2)
228. {
230. if(($auth ->username)==$username)
231. {
233. if(($auth2 ->address)==$address)
234. {
235. DB::update('update customers set  address=? where address= ?', [$new\_address, $address]);
236. return redirect("profile");
237. }
238. else{
239. $error='Password or user id does not match';
240. if (empty($error)){}
241. else{echo '<script>alert("Password or id does not match")</script>'; }
242. return redirect('admin')->with('error',$error);
243. }
244. }
245. else{
246. $error='Password or user id does not match';
247. if (empty($error)){}
248. else{echo '<script>alert("Password or id does not match")</script>'; }
249. return redirect('admin')->with('error',$error);
250. }
251. }
252. }
253. }
254. }

257. public function addtocart(){
258. $dish=Request::input('dish');
259. $dishtype = Request::input('dishtype');
260. $Quantity = Request::input('Quantity');
261. $special=Request::input('special');
262. $login=DB:: select('select dish from products where dish\_name = ?',[$dish]);
263. $login2=DB:: select('select price from products where dish\_name = ?',[$dish]);
264. if(count ($login) > 0)
265. {
266. foreach($login as $auth)
267. {
268. if(($auth ->dish\_name)==$dish)
270. {
271. DB::insert('insert into order(dishname,dishtype,Quantity,special\_instruction) values(?,?,?,?)',[$dish,$dishtype,$Quantity, $special]);
272. return redirect('profile');
273. }
274. else{
275. $error='Password or user id does not match';
276. if (empty($error)){}
277. else{echo '<script>alert("Password or id does not match")</script>'; }
278. return redirect('admin')->with('error',$error);
279. }
280. }
281. }
282. }

285. }

**Screenshot:*Home Page:A picture containing website

Description automatically generated***

***Table Reservation:Graphical user interface, website

Description automatically generated***

***User login:***

***A picture containing diagram

Description automatically generated***

***Admin Dashboard:***

***Graphical user interface, application

Description automatically generated***

***User Profile:***

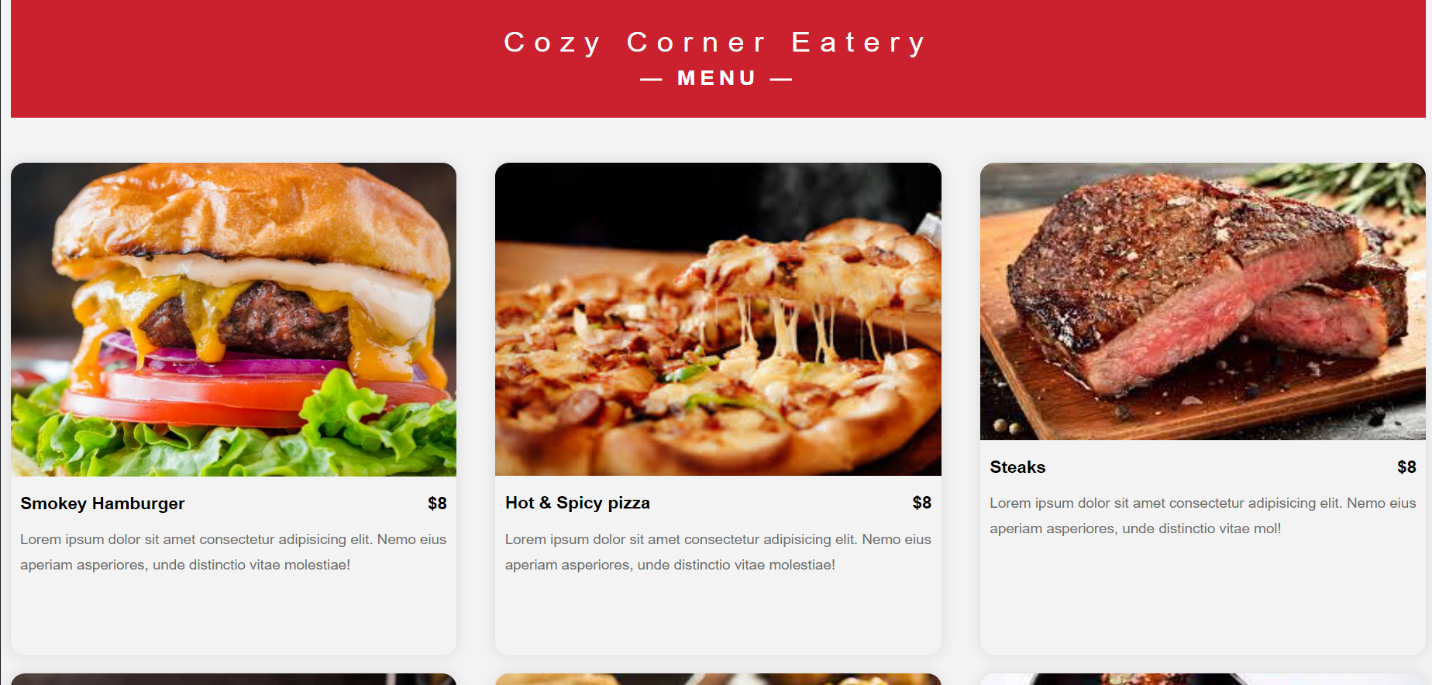
***A screen shot of a video game

Description automatically generated with low confidence***

***Review:Graphical user interface, website

Description automatically generated***

***Menu:***

******