

MINI PROJECT REPORT ON CANTEEN MANAGEMENT SYSTEM

UNDER THE GUIDENCE OF

Mr. Akash Chaudhary

Technical Trainer

T&D dept.

GLA University, Mathura.

Under taken by

PRAKHAR AGARWAL 181500468

HARSH AGARWAL 181500246

CONTENT

- **Abstract**
- **Objective**
- **Introduction**
- **Problem**
- **Implementation**
- **Progress**
- **Work Remaining**
- **DFD**
- **Screenshots**
- **References**

Abstract

The project is entitled to “Online Canteen Management System”.

The project is the outcome of our summer training and the technical skills we have learned in the college on “Full Stack Web Development”. The Full Stack Web Development comprises of two parts Frontend and Backend in Frontend we have used Handlebars and CSS whereas in Backend we have used Node JS and Mongo DB. Using this an online platform has been created for the canteen so that all the orders and the delivery of the meals can be done smoothly.

The main objective of the project is to make the process of the canteen easier and effective. It will save time of the people who waits in the long queues for the food. The people gathering that takes place in large numbers in the canteen will also get reduced.

An online platform has been made successfully, now an individual or a person can make an account on our platform and can order anything from the canteen from the menu and can also pay for the meal via. Online mode and can take away the order from the canteen smoothly. Money can also get added into the accounts wallet and will keep modifying when the payment is made.

With the help of this project every person can order their meal easily and we can prevent a large horde of people gathering and huge queues and rush in the canteen, proper social distancing can be made in this time of Covid-19 and proper norms can be followed in the canteen.

Objective

1. The main objective of the project is to make the process of the canteen easier and effective.
2. It will save time of the people who waits in the long queues for the food.
3. The people gathering that takes place in large numbers in the canteen will also get reduced.

Introduction

As the title suggests Canteen Management System which means how can we manage a canteen in a more effective way as canteens are very crowded in colleges and offices during lunch hours and break time, horde of people gather at one place and we need to consider the Covid-19 era in which these type of situations can be harmful so for this we created a canteen management system where students can place the orders by a web application, the payments can be done online, they just need to go to the canteen to collect the order, the balance can be updated online using any payment method.

Problem

The problem is that during the lunch hours of the office, schools and colleges and when the time is at peak there is a large rush in the canteens and the people rush to get the meal which they want and large queues forms in front of the counter and there is no place to sit and eat your lunch peacefully and moreover no social distancing is followed in the canteen and as we know that how much social distancing is important in this time of Covid-19.

So that is why we have created an online canteen platform where an individual can place any order from the menu via. Online mode and can also pay for it at the same time from the online wallet and then the individual can collect their order from the take away window. This action saves time of the person prevents rush in the canteen and more importantly proper norms of Covid-19 can be followed in the canteen and each and everyone can get the meal they want.

Implementation

The project uses NodeJS as a working methodology. Various libraries of NodeJS has been used in this project to make this project working more effective, efficient and apprehensive. The project works on Node JS along with other technologies where each tech used in this project has a different purpose. Some techs give us to perform the functions and operations in effective and minimized manner, whereas some tech helps us to perform the operations in the backend without being observed or shown to the user in the front end framework the CSS helps us to make our project more attractive and interactive so that each function can be visualized easily.

The main platform on which the project works is Node JS. The libraries that are included in the project are ‘express’, ‘mongoose’, ‘path’, ‘express-handlebars’, ‘body-parser’, ‘passport’, ‘express-file upload’, ‘express-session’, ‘connect-flash’.

Frontend and Backend

Frontend: -

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

Front End of the project lies on Handlebars, CSS AND JavaScript.

- **Handlebars:** - Handlebars is a simple templating language. It uses a template and an input object to generate HTML or other text formats. Handlebars templates look like regular text with embedded Handlebars expressions. With Handlebars, you can separate the generation of HTML from the rest of your JavaScript and write cleaner code.
- **CSS:** - CSS is cascading style sheet which is used to give designer look to HTML using the external file.
- **Java script:** - Java script is used for client-side scripting which can help in using validation on the website and many more other functions.

Backend: -

Back-end Development refers to the server-side development. It focuses on databases, scripting, website architecture. It contains behind-the-scene activities that occur when performing any action on a website. It can be an account login or making a purchase from an online store. Code written by back-end developers helps browsers to communicate with database information.

Back End of the project lies on NodeJS.

Node.JS: - It is an open-source, cross platform, backend JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Progress

The project is going as follows: -

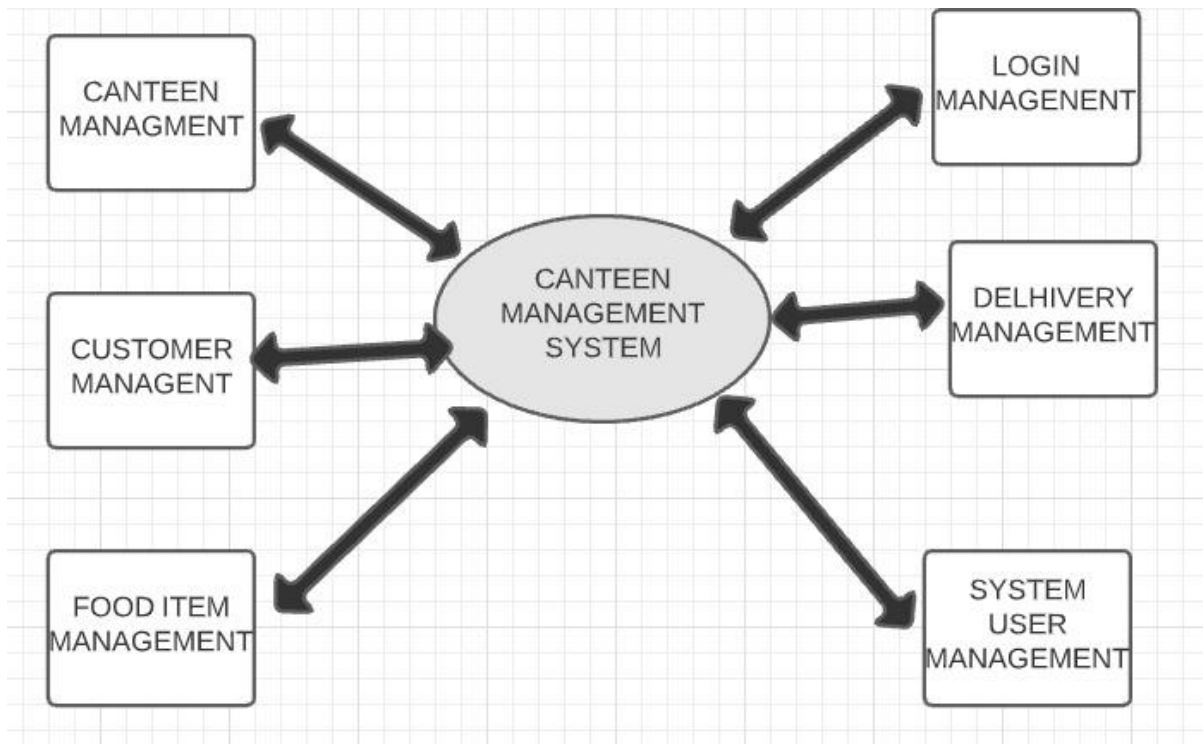
- Sign-Up page has been made successfully.
- The sign-up page is the place where you can create an account by filling the personal details and setting the credentials of the account.
- Sign-In page has been made successfully.
- The sign-in or the log-in page is the place where you can log-in into your account to see the menu to order something.
- Admin page has been made successfully.
- The admin portal controls all the working of the online canteen system all the orders or meals are approved by the admin through this portal.
- Home page has been made successfully.
- The home page consists of two options sign-in and sign-up where the user can create an account or login into the existing account.

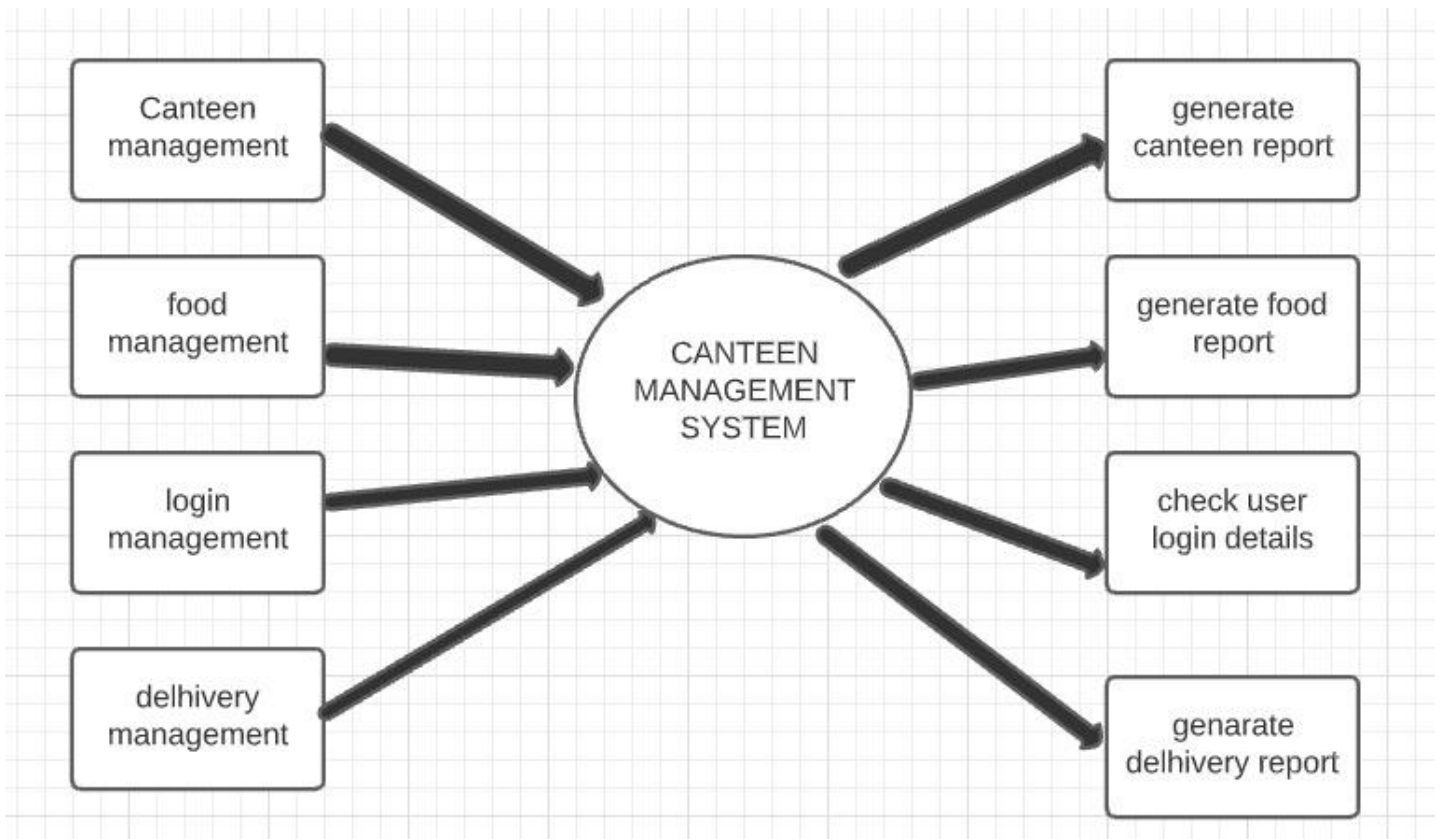
Work Remaining

- **Customer wallet:** - In this page the customer can add money from any bank and the meal price will get cut from the wallet only while placing an order.
- **Placing order:** - With the help of this page the customer can select any dish from the menu and can place an order for it.
- **Pre-book order of the customer:** - With the help of this page a customer can pre-book a order and can receive that order later.
- **Subscription of a customer:** - There are two types of customer premium and non-premium customers.
- **Admin menu updation:** -Admin is responsible for updating the menu if any new dish comes in the course.
- **Delivery/Approval:** - The admin is responsible for approving the order of the customer and get it delivered to the customer
- **Admin Dashboard:** - In this page the admin can see the list and manes of all the customers who have account on the online platform and can see their order history.

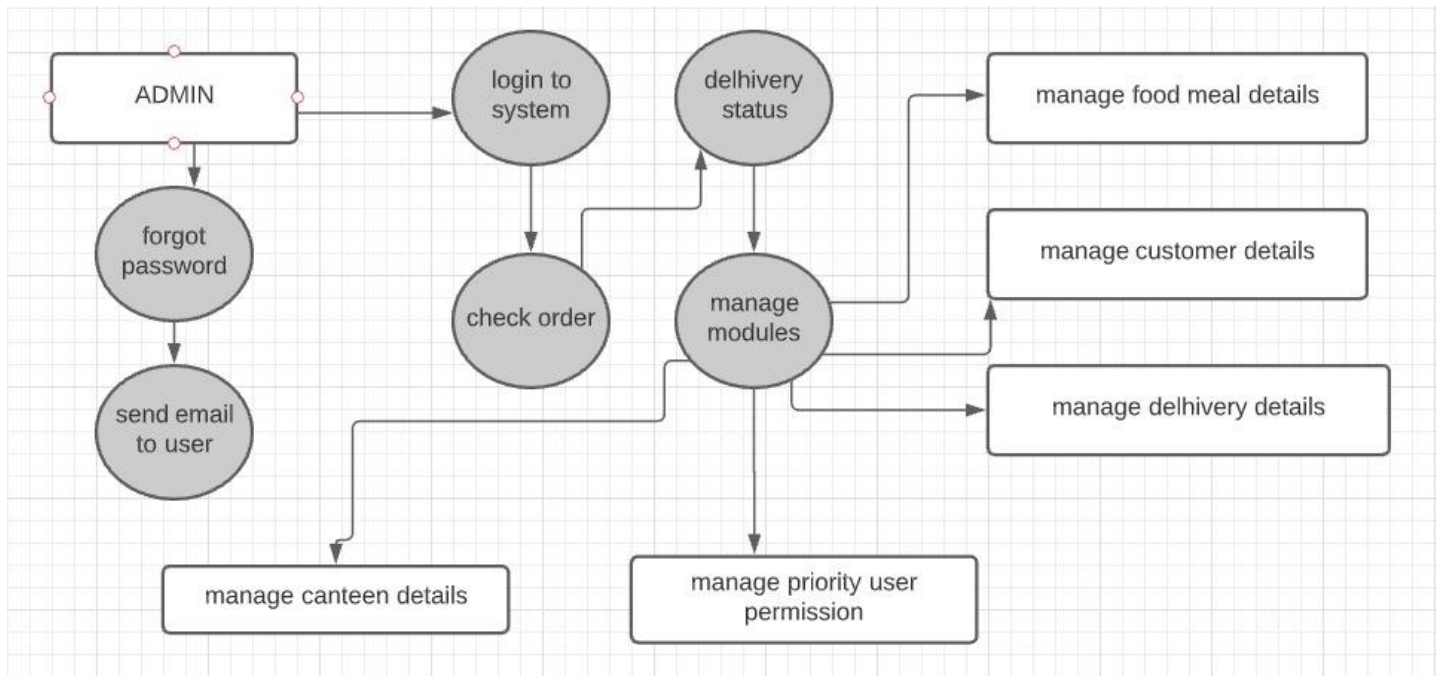
DFD:

0 Level DFD: -



1st Level DFD: -

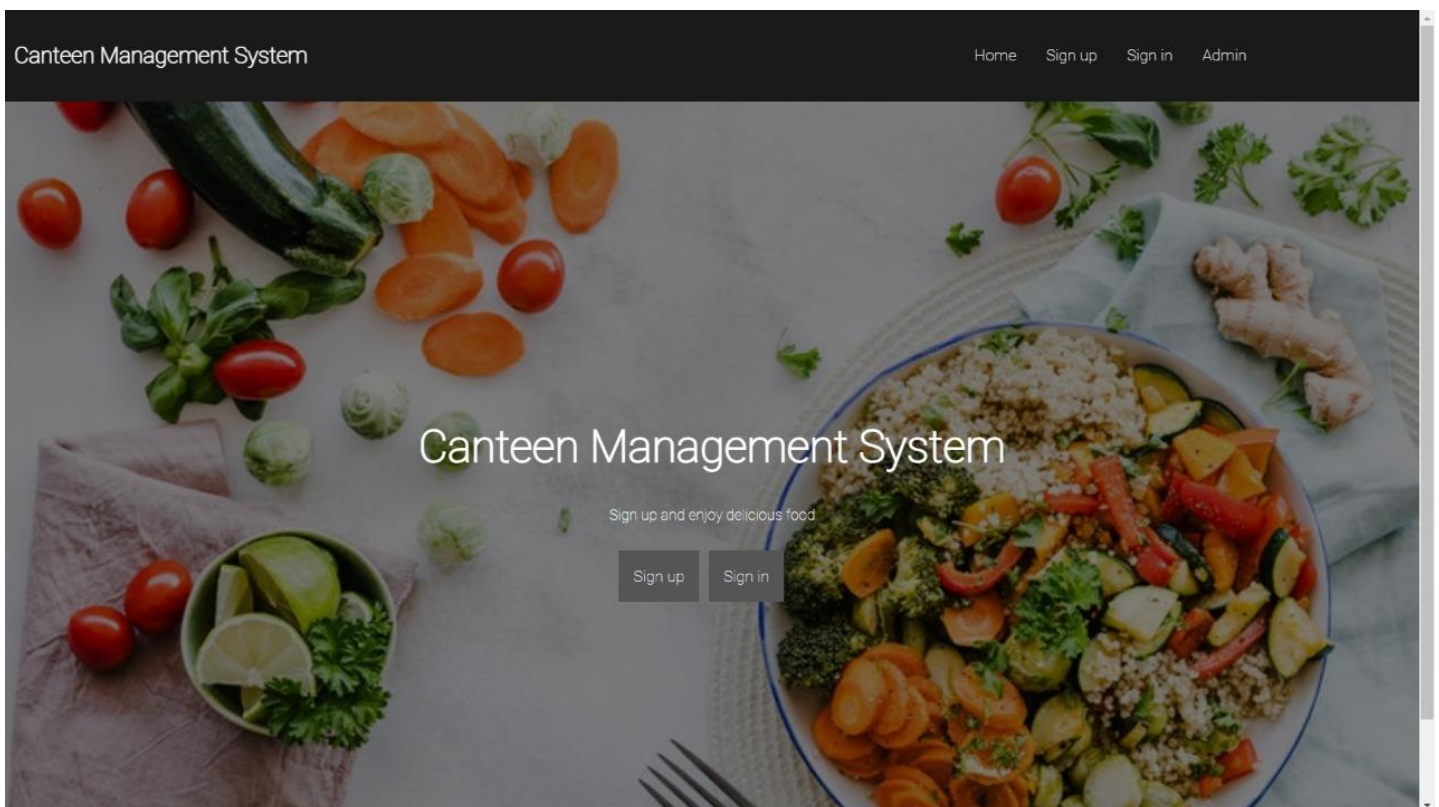
2nd Level DFD :-

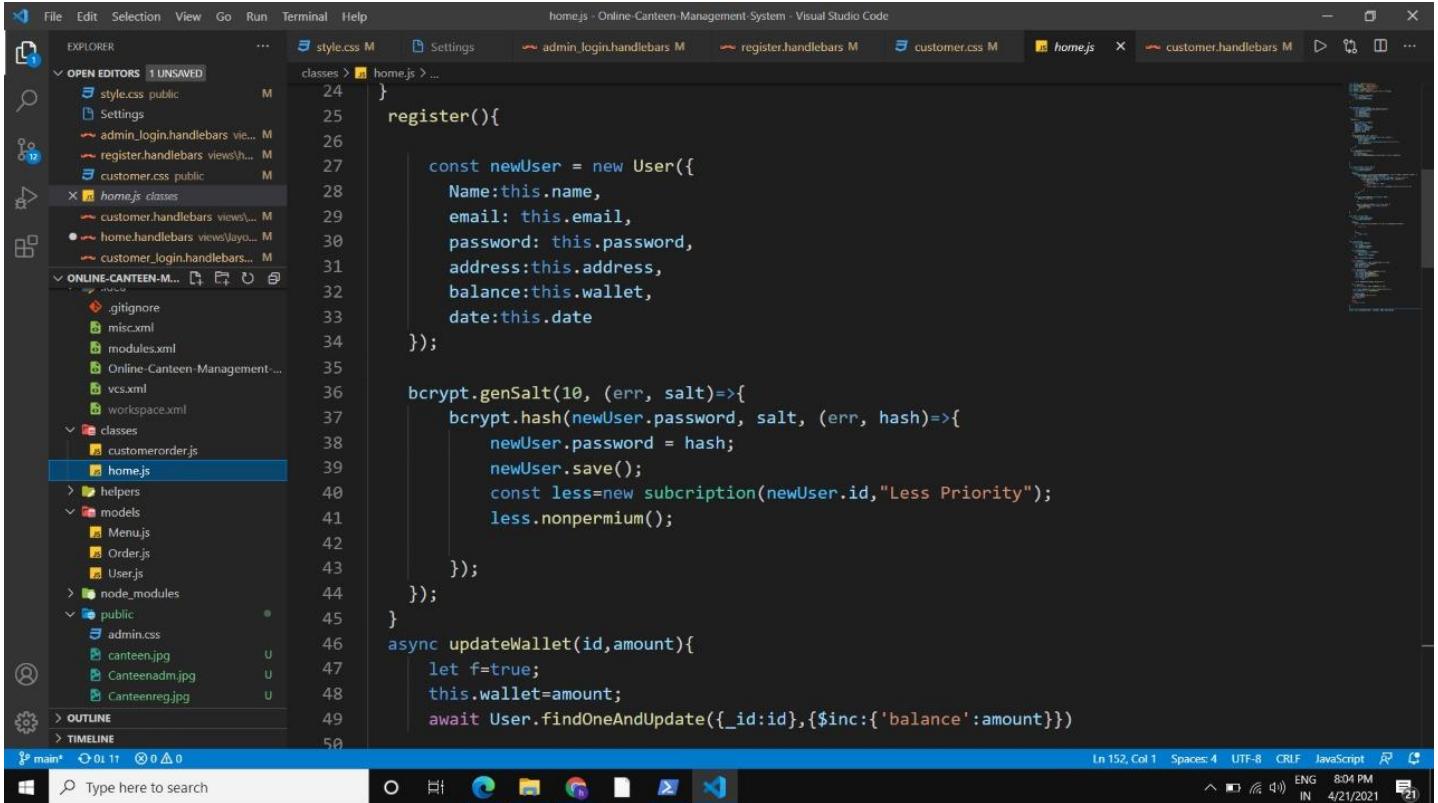


Implementation of the project work: -

Screenshots

Home.JS

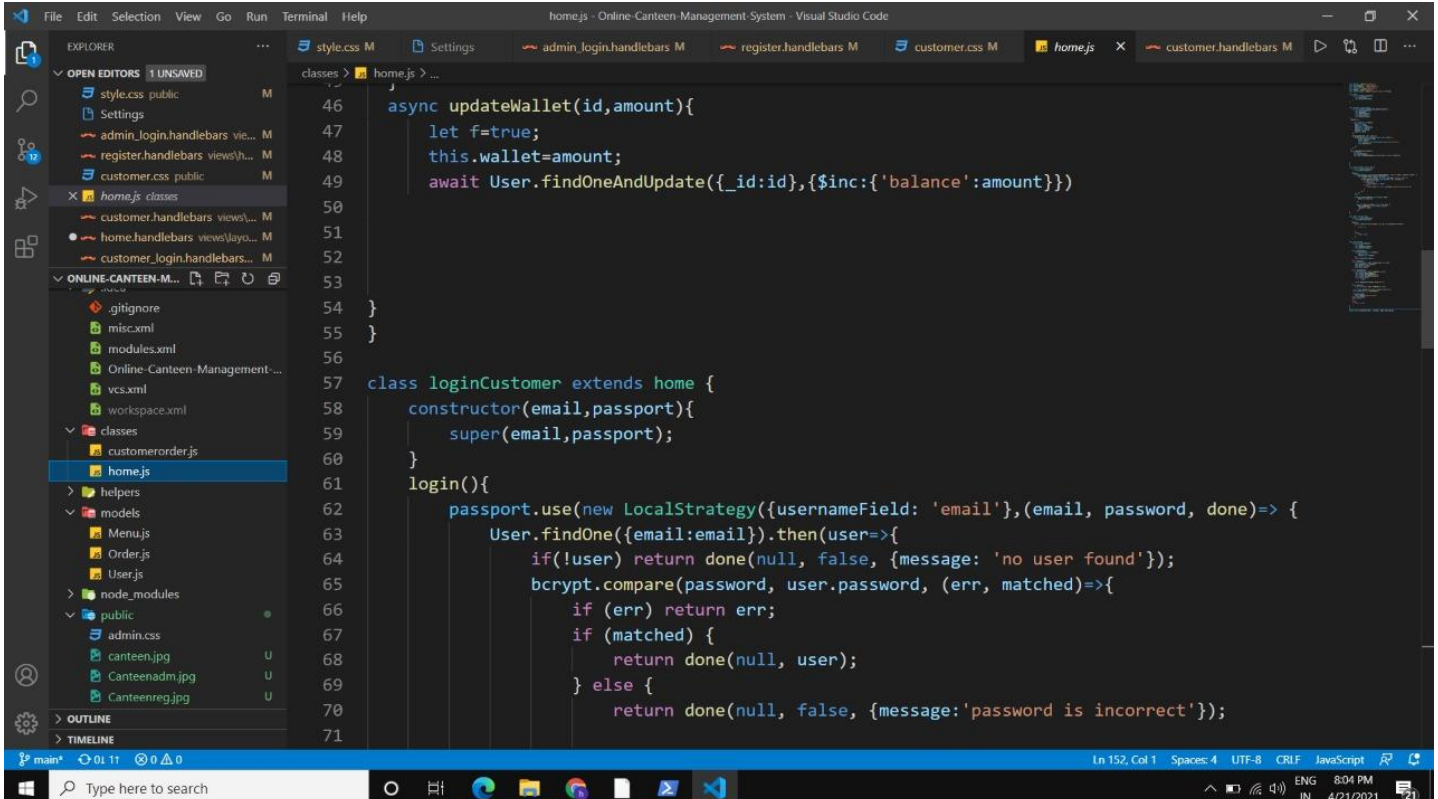




```

24 }
25
26 register(){
27
28   const newUser = new User({
29     Name: this.name,
30     email: this.email,
31     password: this.password,
32     address: this.address,
33     balance: this.wallet,
34     date: this.date
35   });
36
37   bcrypt.genSalt(10, (err, salt) => {
38     bcrypt.hash(newUser.password, salt, (err, hash) => {
39       newUser.password = hash;
40       newUser.save();
41       const less = new subscription(newUser.id, "Less Priority");
42       less.nonpermium();
43     });
44   });
45 }
46
47 async updateWallet(id, amount) {
48   let f = true;
49   this.wallet = amount;
50   await User.findOneAndUpdate({ _id: id }, { $inc: { 'balance': amount } })

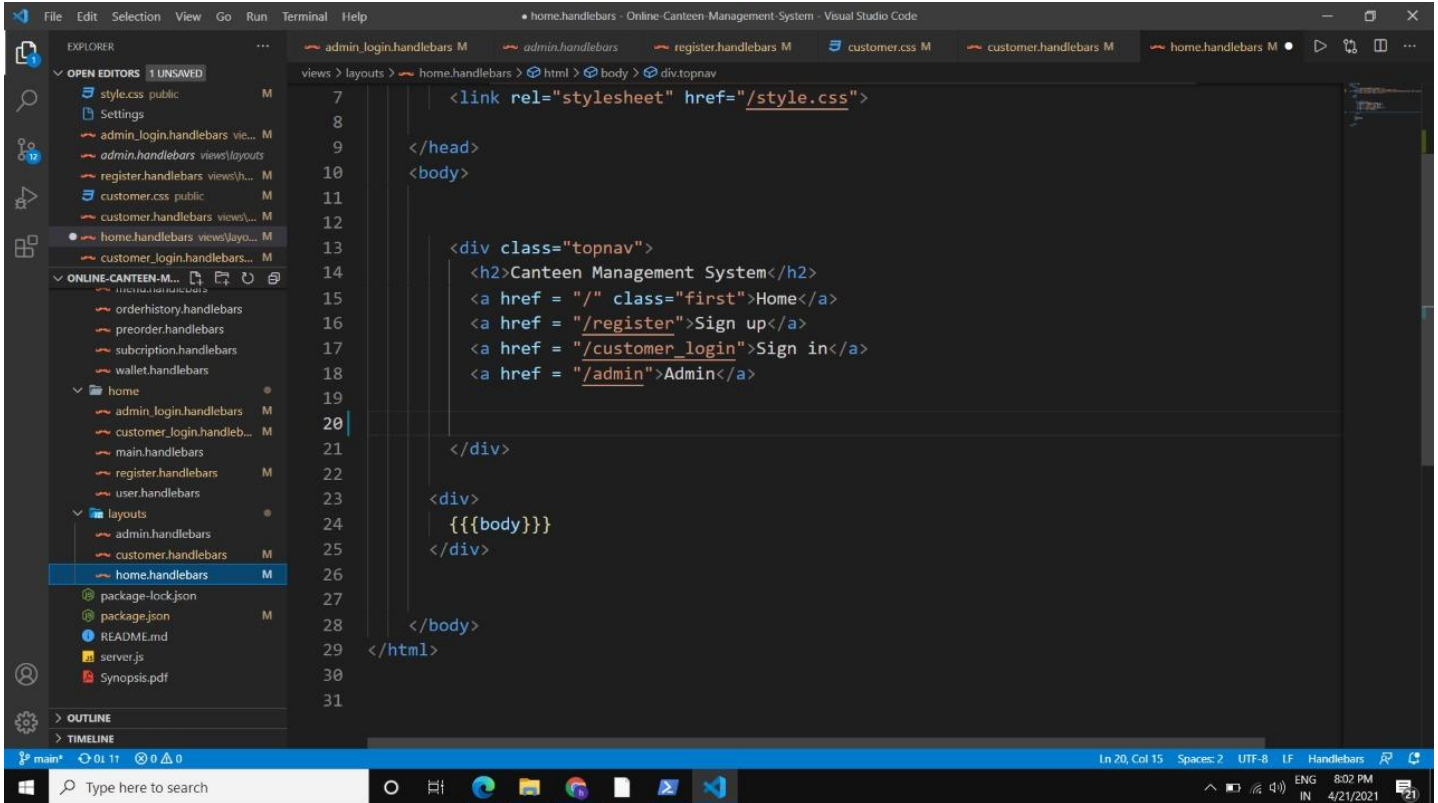
```



```

46
47 async updateWallet(id, amount) {
48   let f = true;
49   this.wallet = amount;
50   await User.findOneAndUpdate({ _id: id }, { $inc: { 'balance': amount } })
51 }
52
53
54 }
55
56
57 class loginCustomer extends home {
58   constructor(email, passport) {
59     super(email, passport);
60   }
61   login() {
62     passport.use(new LocalStrategy({ usernameField: 'email' }, (email, password, done) => {
63       User.findOne({ email: email }).then(user => {
64         if (!user) return done(null, false, { message: 'no user found' });
65         bcrypt.compare(password, user.password, (err, matched) => {
66           if (err) return err;
67           if (matched) {
68             return done(null, user);
69           } else {
70             return done(null, false, { message: 'password is incorrect' });
71           }
72         });
73       });
74     });
75   }
76 }

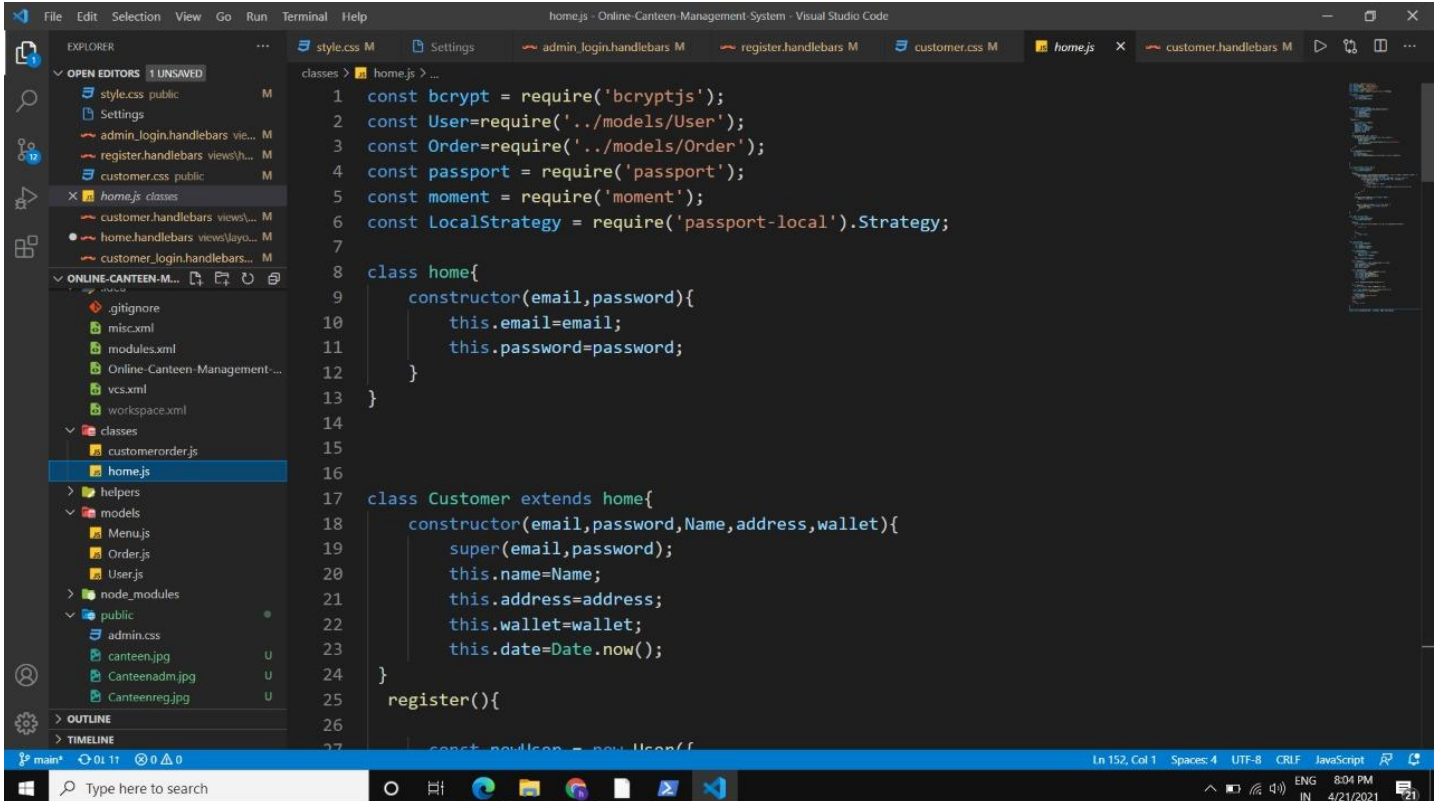
```



```

7      <link rel="stylesheet" href="/style.css">
8
9    </head>
10   <body>
11
12     <div class="topnav">
13       <h2>Canteen Management System</h2>
14       <a href = "/" class="first">Home</a>
15       <a href = "/register">Sign up</a>
16       <a href = "/customer_login">Sign in</a>
17       <a href = "/admin">Admin</a>
18     </div>
19
20     <div>
21       {{{body}}}
22     </div>
23
24   </body>
25 </html>

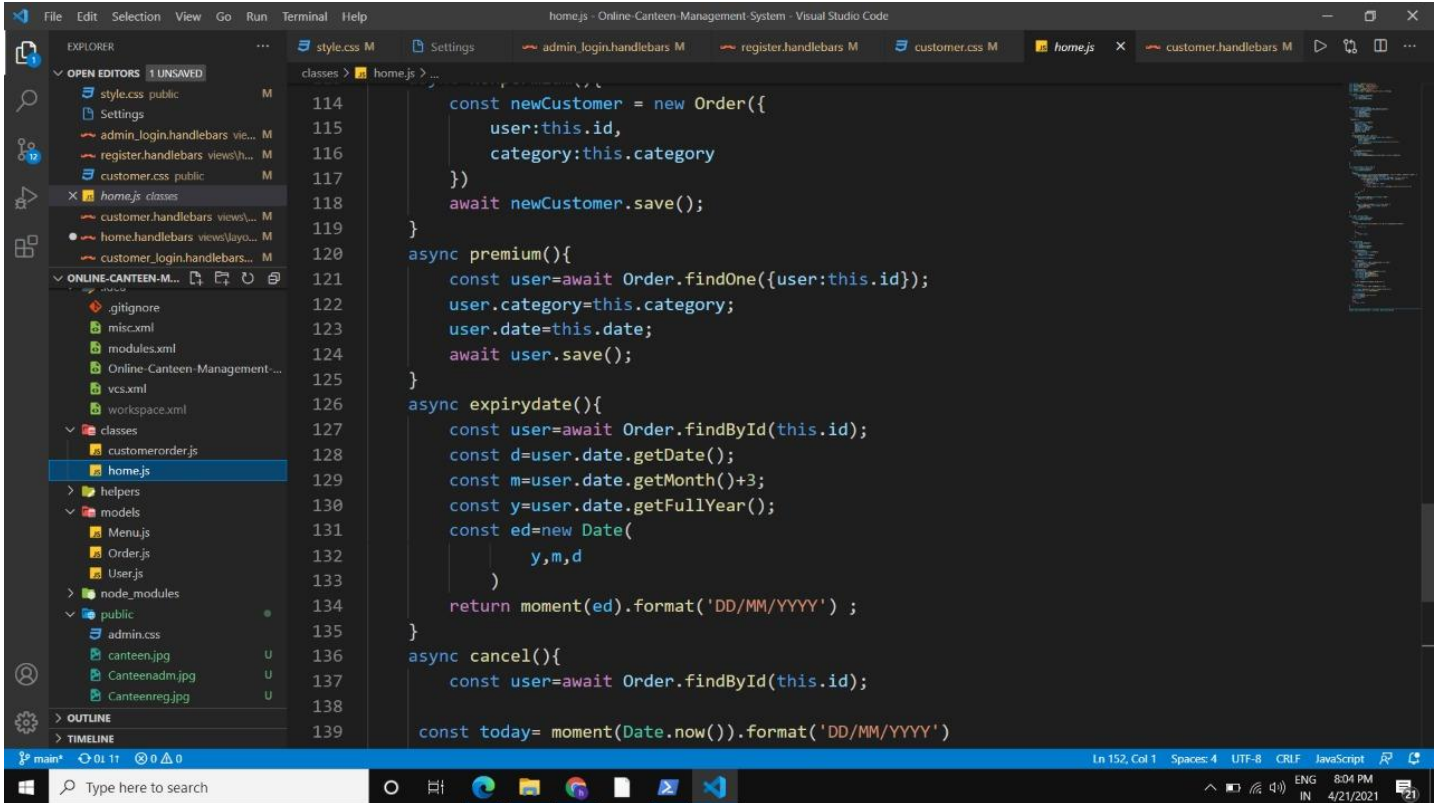
```



```

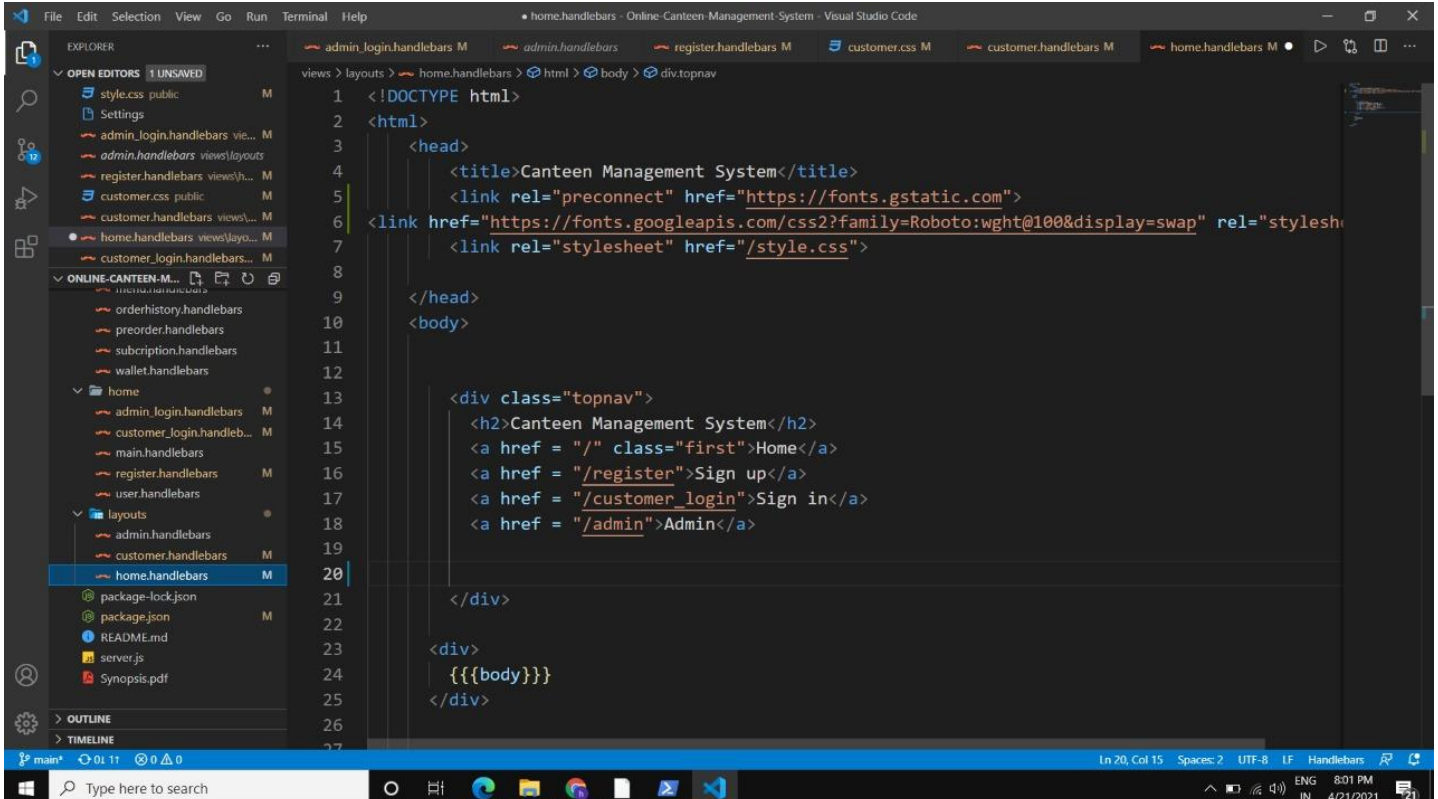
1  const bcrypt = require('bcryptjs');
2  const User=require('../models/User');
3  const Order=require('../models/Order');
4  const passport = require('passport');
5  const moment = require('moment');
6  const LocalStrategy = require('passport-local').Strategy;
7
8  class home{
9    constructor(email,password){
10      this.email=email;
11      this.password=password;
12    }
13  }
14
15  class Customer extends home{
16    constructor(email,password,Name,address,wallet){
17      super(email,password);
18      this.name=Name;
19      this.address=address;
20      this.wallet=wallet;
21      this.date=Date.now();
22    }
23    register(){
24      const newUser = new User({

```

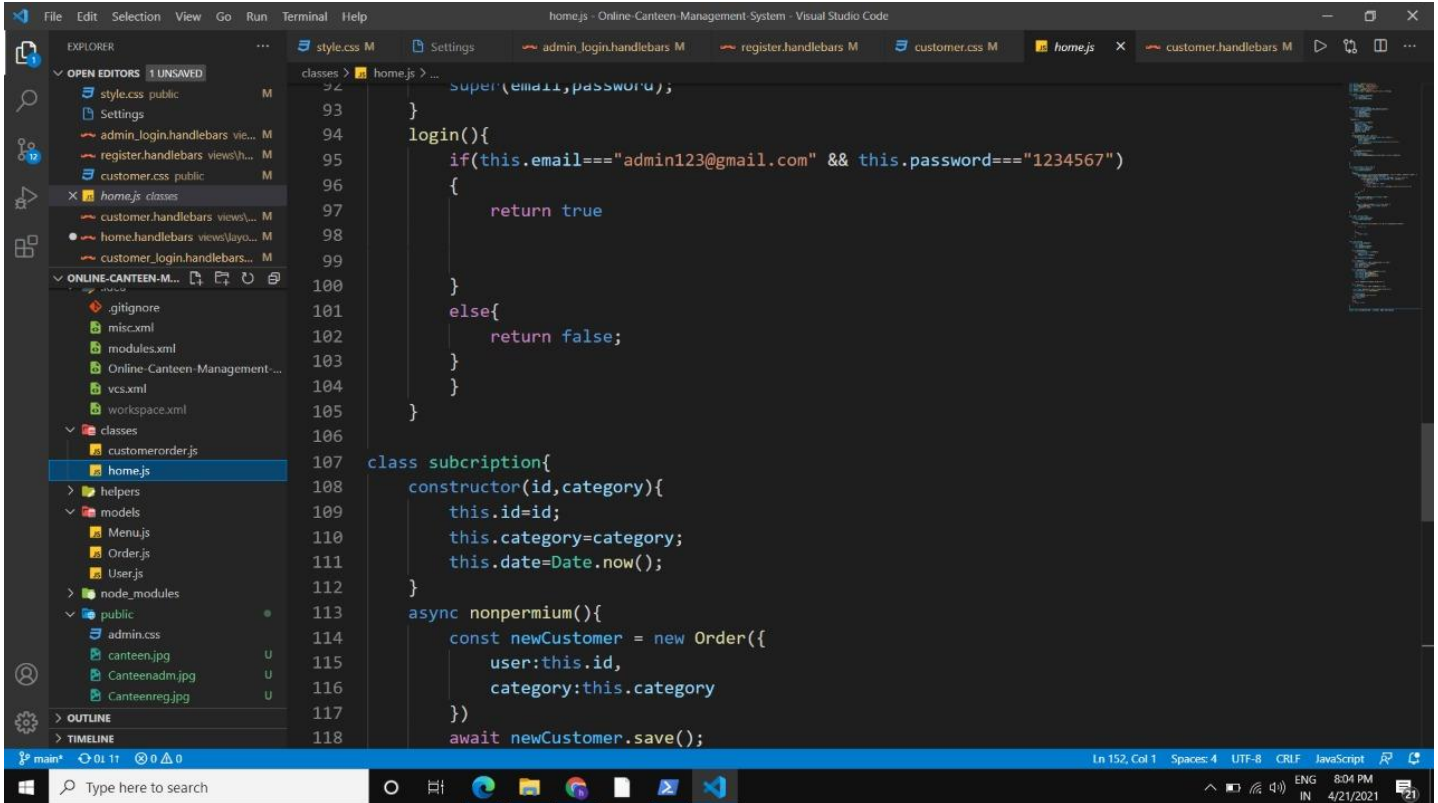
```

114     const newCustomer = new Order({
115         user:this.id,
116         category:this.category
117     })
118     await newCustomer.save();
119 }
120 async premium(){
121     const user=await Order.findOne({user:this.id});
122     user.category=this.category;
123     user.date=this.date;
124     await user.save();
125 }
126 async expirydate(){
127     const user=await Order.findById(this.id);
128     const d=user.date.getDate();
129     const m=user.date.getMonth()+3;
130     const y=user.date.getFullYear();
131     const ed=new Date(
132         y,m,d
133     )
134     return moment(ed).format('DD/MM/YYYY') ;
135 }
136 async cancel(){
137     const user=await Order.findById(this.id);
138     const today= moment(Date.now()).format('DD/MM/YYYY')
139 }
  
```



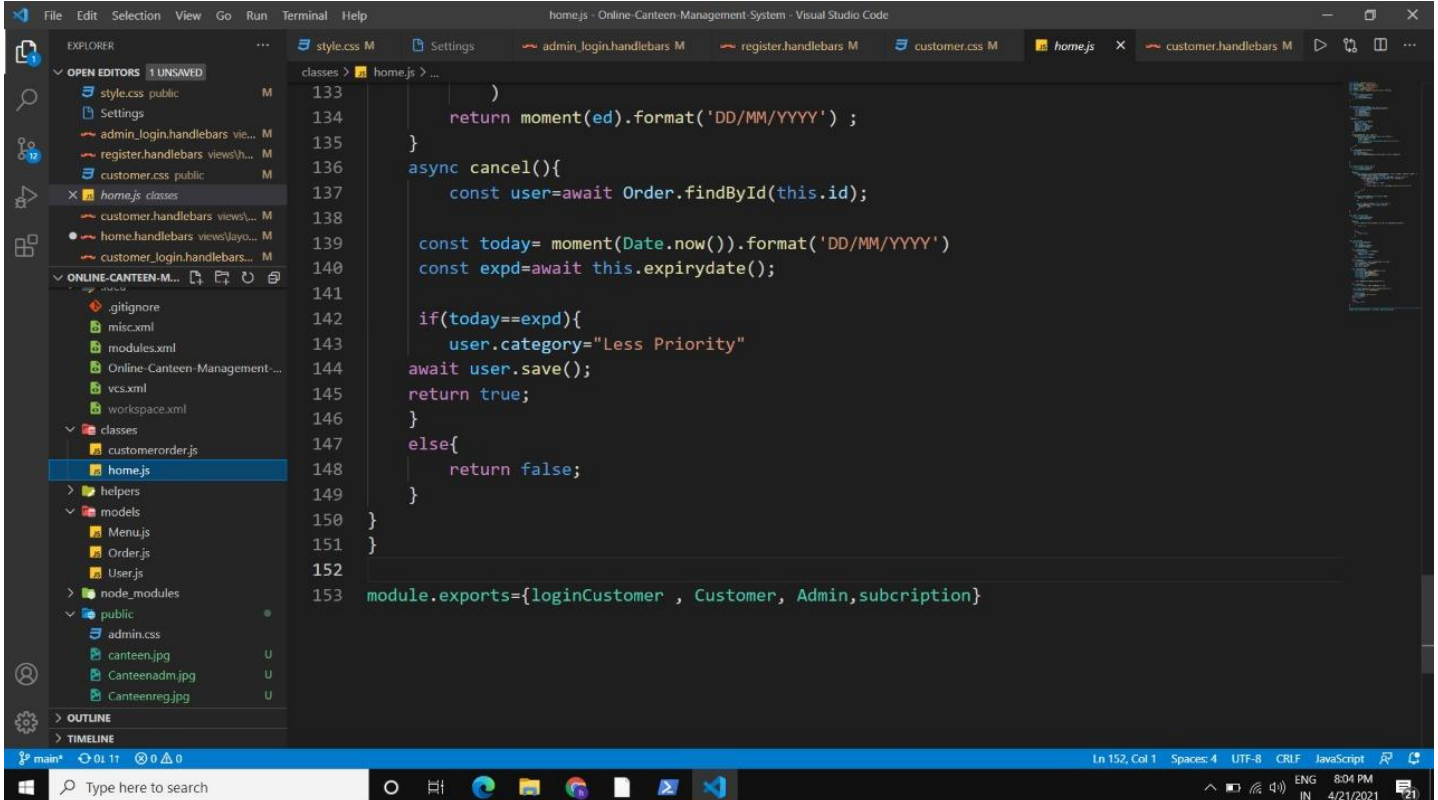
```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <title>Canteen Management System</title>
5         <link rel="preconnect" href="https://fonts.gstatic.com">
6         <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@100&display=swap" rel="stylesheet">
7         <link rel="stylesheet" href="/style.css">
8     </head>
9     <body>
10
11         <div class="topnav">
12             <h2>Canteen Management System</h2>
13             <a href = "/" class="first">Home</a>
14             <a href = "/register">Sign up</a>
15             <a href = "/customer_login">Sign in</a>
16             <a href = "/admin">Admin</a>
17         </div>
18
19         <div>
20             {{{body}}}
21         </div>
22     </body>
23 </html>
  
```



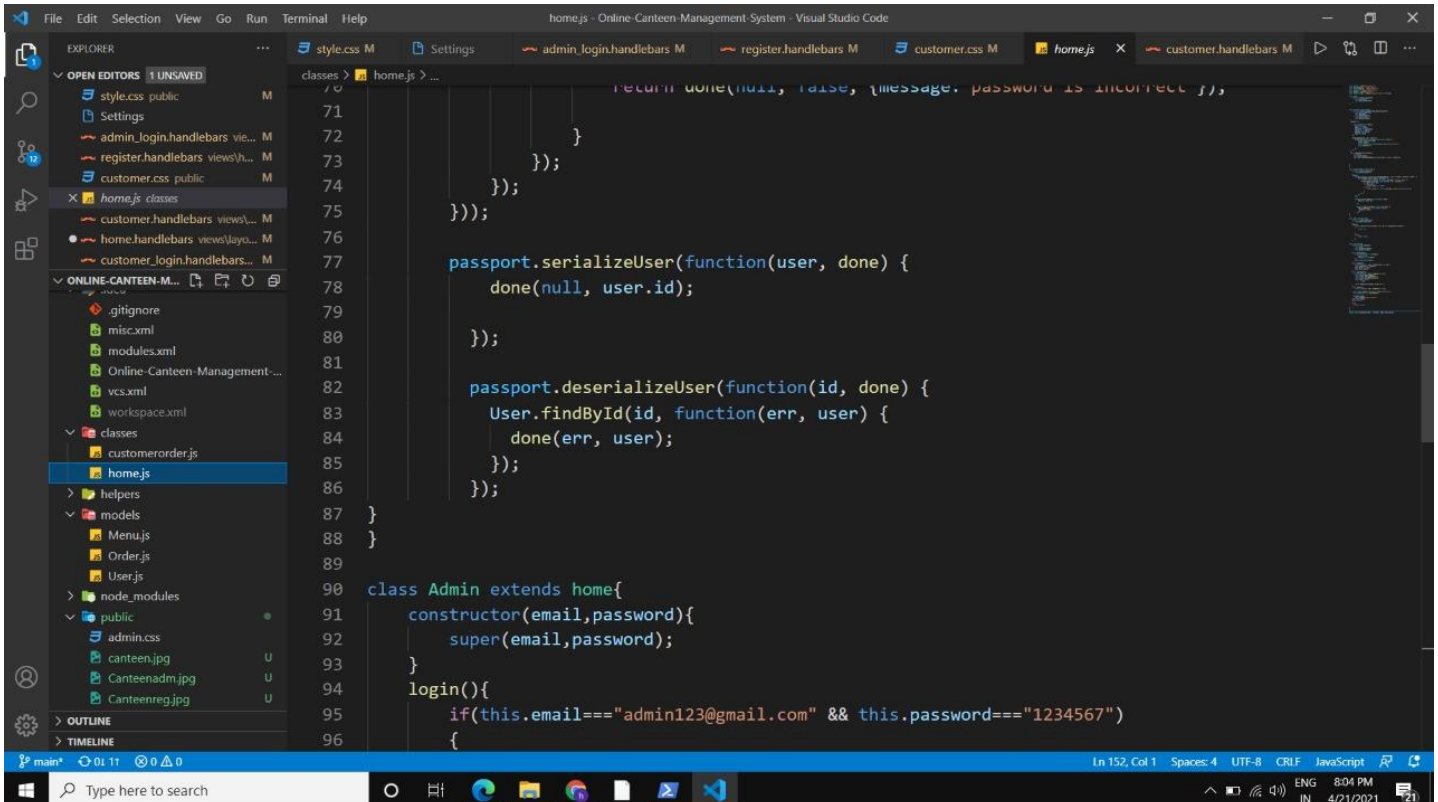
```

100 super(email,password);
101 }
102 login(){
103   if(this.email=="admin123@gmail.com" && this.password=="1234567")
104   {
105     return true
106   }
107   else{
108     return false;
109   }
110 }
111 class subscription{
112   constructor(id,category){
113     this.id=id;
114     this.category=category;
115     this.date=Date.now();
116   }
117   async nonpermium(){
118     const newCustomer = new Order({
119       user:this.id,
120       category:this.category
121     })
122     await newCustomer.save();
  
```



```

133   )
134   return moment(ed).format('DD/MM/YYYY') ;
135 }
136 async cancel(){
137   const user=await Order.findById(this.id);
138
139   const today= moment(Date.now()).format('DD/MM/YYYY')
140   const expd=await this.expirydate();
141
142   if(today==expd){
143     user.category="Less Priority"
144     await user.save();
145     return true;
146   }
147   else{
148     return false;
149   }
150 }
151 }
152
153 module.exports={loginCustomer , Customer, Admin,subscription}
  
```



```

71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
return done(null, false, {message: 'password is incorrect'});
}
});
});
});
passport.serializeUser(function(user, done) {
  done(null, user.id);
});
passport.deserializeUser(function(id, done) {
  User.findById(id, function(err, user) {
    done(err, user);
  });
});
});
}
}
class Admin extends home{
  constructor(email,password){
    super(email,password);
  }
  login(){
    if(this.email=="admin123@gmail.com" && this.password=="1234567")
    {

```


Admin login.handlebars:-

Canteen Management System Home Sign up Sign in Admin

Admin Login

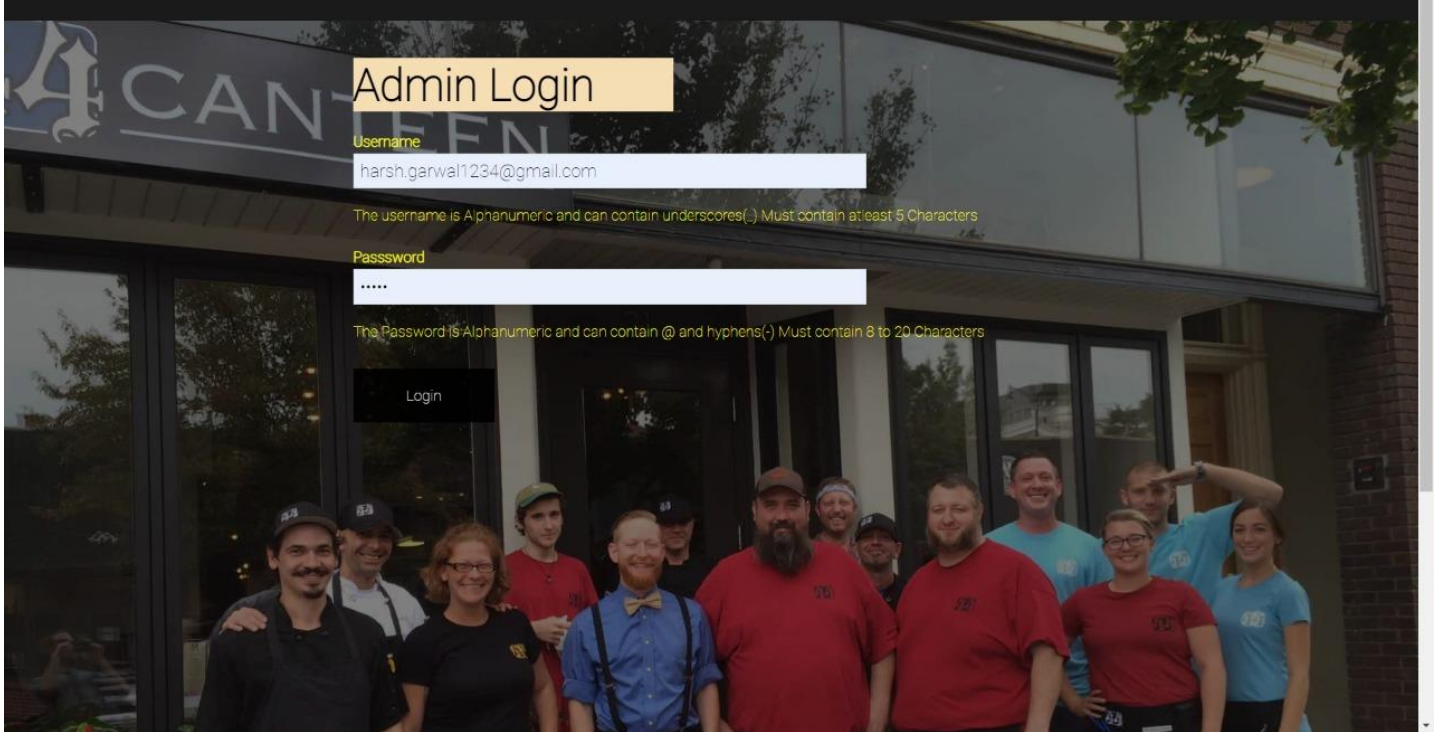
Username
harsh.garwal1234@gmail.com

The username is Alphanumeric and can contain underscores(.) Must contain atleast 5 Characters

Password
.....

The Password is Alphanumeric and can contain @ and hyphens(-) Must contain 8 to 20 Characters

Login



```

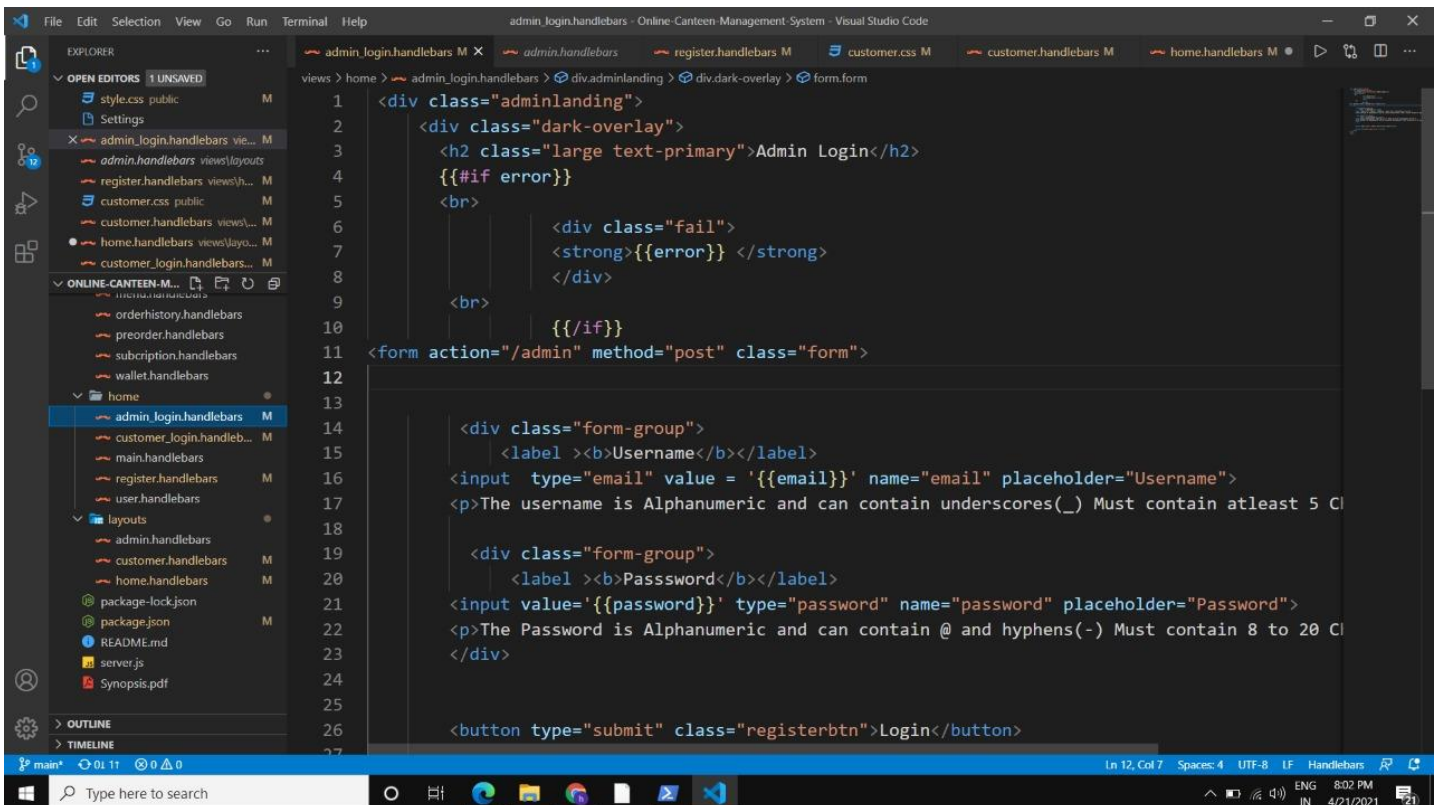
File Edit Selection View Go Run Terminal Help
admin_login.handlebars - Online-Canteen-Management-System - Visual Studio Code

EXPLORER
OPEN EDITORS 1 UNSAVED
style.css public M
Settings
admin_login.handlebars vie... M
admin.handlebars views/layouts M
register.handlebars views/h... M
customer.css public M
customer.handlebars views/... M
home.handlebars views/layo... M
customer_login.handlebars... M
ONLINE-CANTEEN-M... M
orderhistory.handlebars
preorder.handlebars
subscription.handlebars
wallet.handlebars
home
admin_login.handlebars M
customer_login.handleb... M
main.handlebars
register.handlebars M
user.handlebars
layouts
admin.handlebars M
customer.handlebars M
home.handlebars M
package-lock.json
package.json M
README.md
server.js
Synopsis.pdf
OUTLINE
TIMELINE

views > home > admin_login.handlebars > @div.adminlanding > @div.dark-overlay > @form.form
21 <input value= {{password}} type= password name= password placeholder= password >
22 <p>The Password is Alphanumeric and can contain @ and hyphens(-) Must contain 8 to 20 Characters</p>
23 </div>
24
25
26 <button type="submit" class="registerbtn">Login</button>
27
28 <script src="admin_login.js.js"></script>
29 </form>
30 </div>
31 </div>
32
33

```

main* 01:11 0 0 0 In 12, Col 7 Spaces: 4 UTF-8 LF Handlebars ENG 8:02 PM 4/21/2021

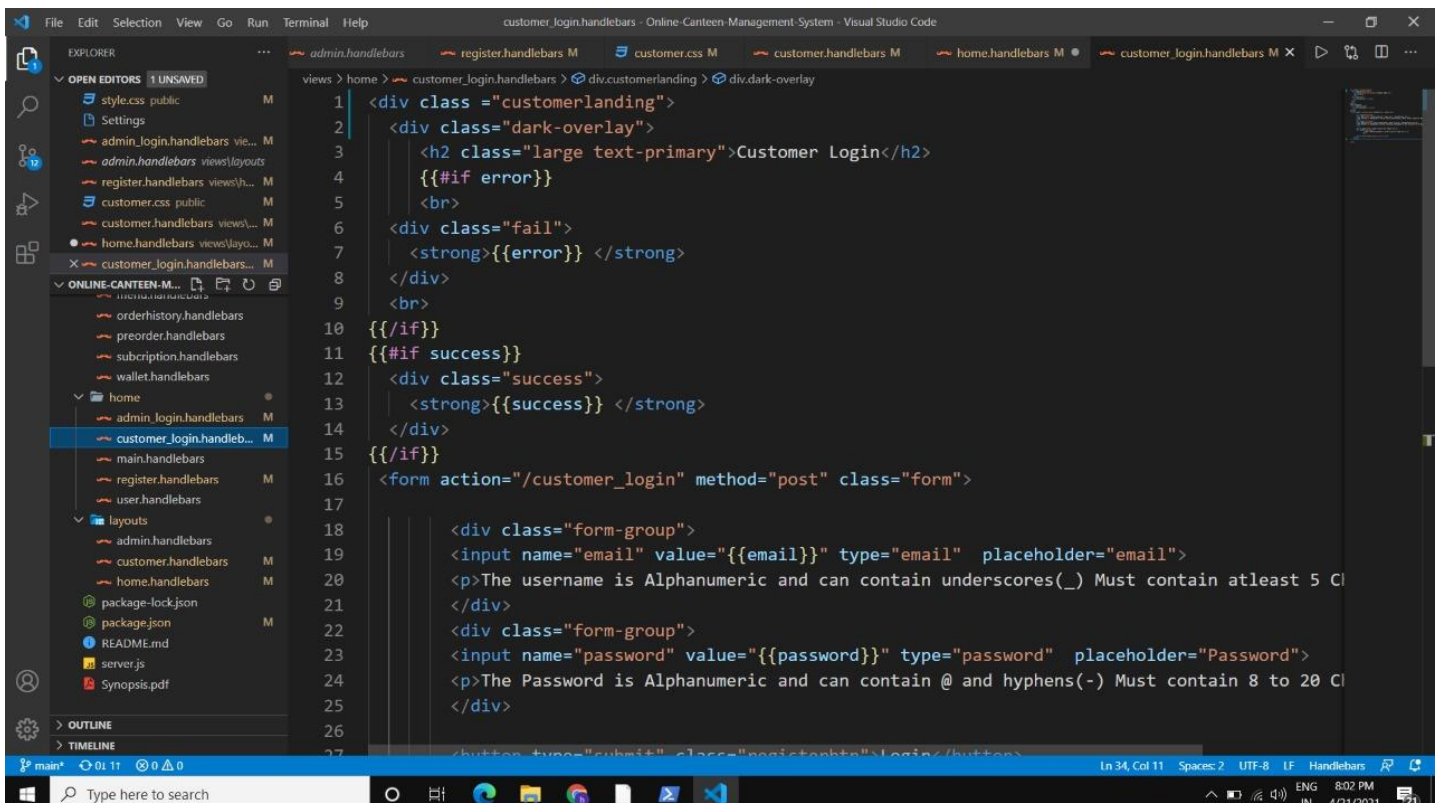
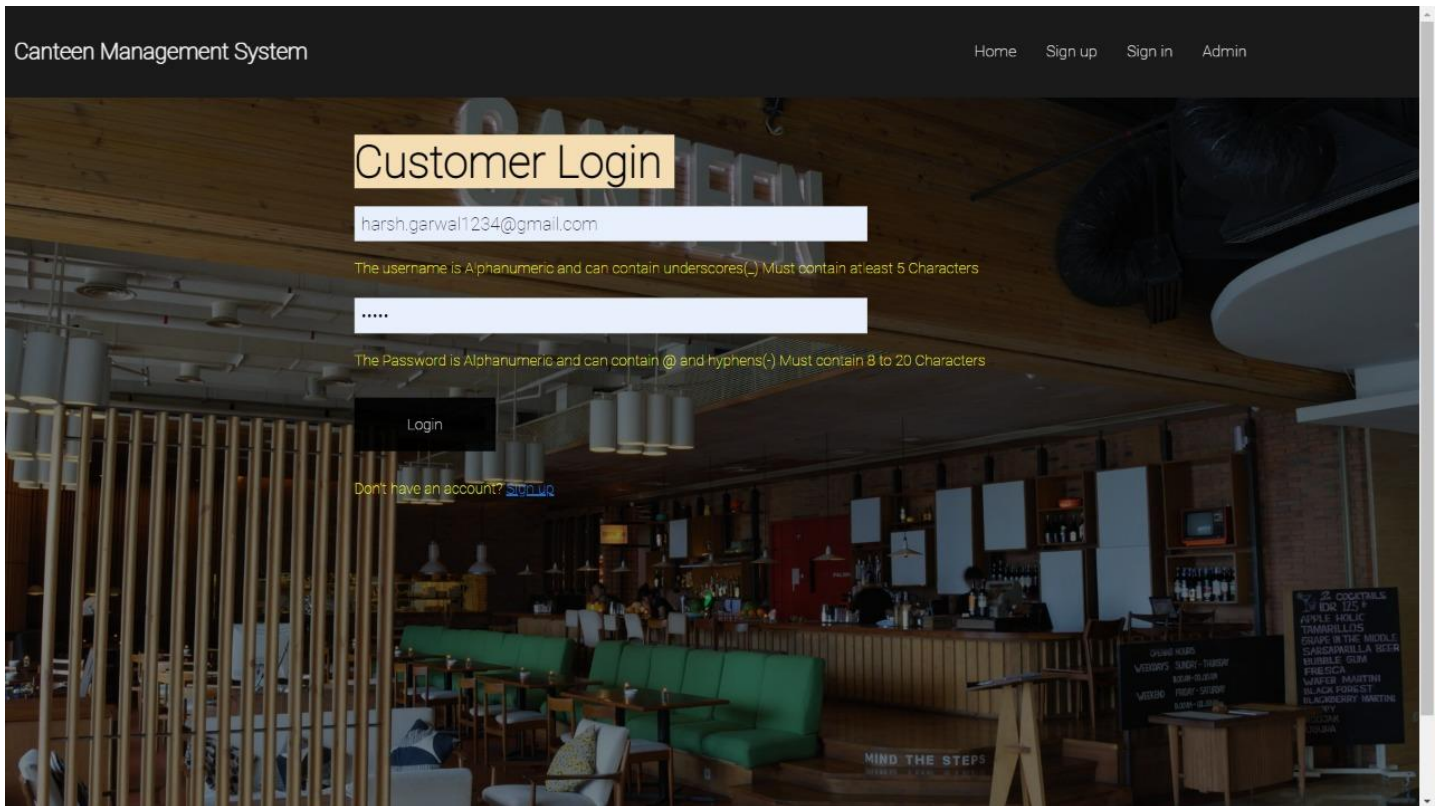


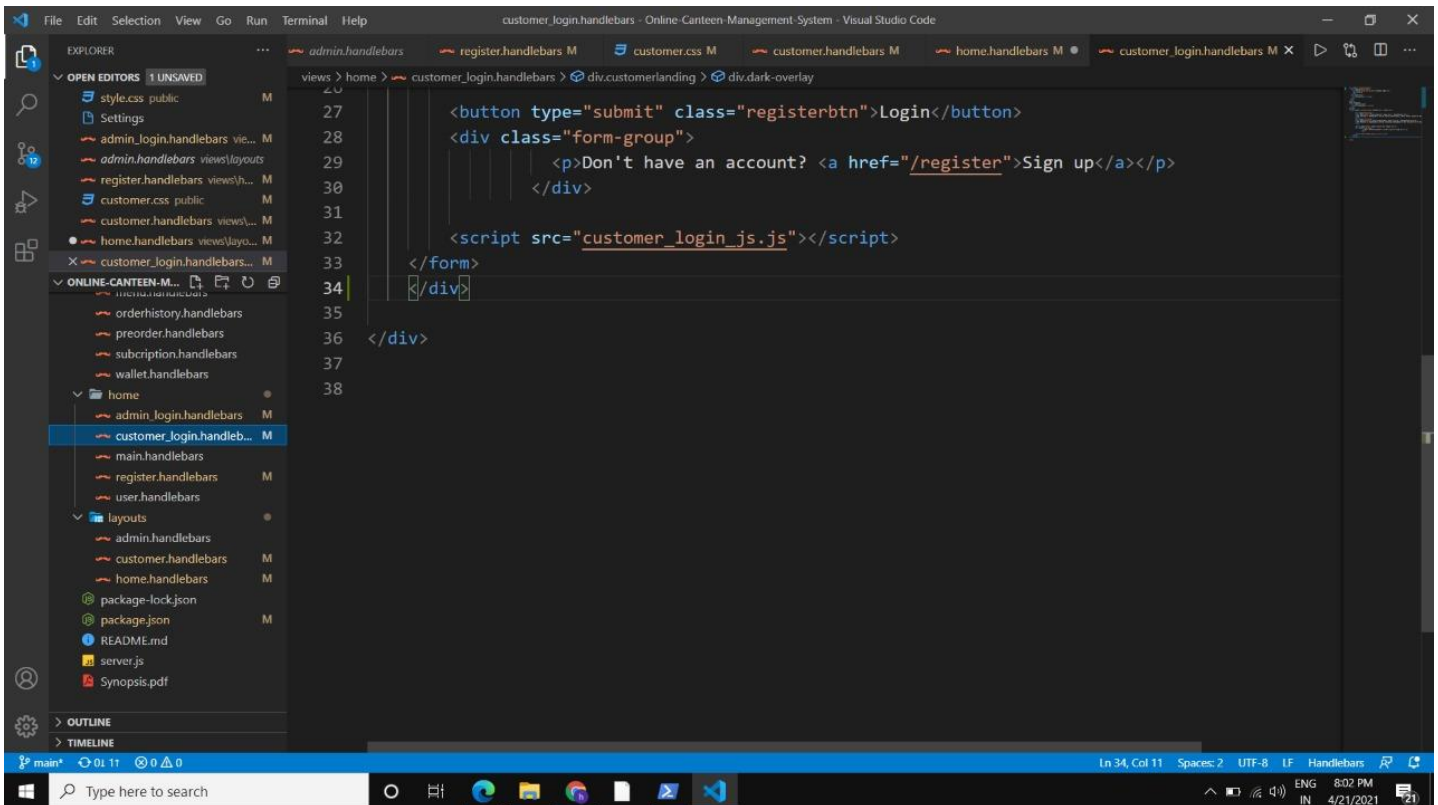
```

1 <div class="adminlanding">
2   <div class="dark-overlay">
3     <h2 class="large text-primary">Admin Login</h2>
4     {{#if error}}
5     <br>
6       <div class="fail">
7         <strong>{{error}}</strong>
8       </div>
9     <br>
10    {{/if}}
11 </div>
12 <form action="/admin" method="post" class="form">
13
14   <div class="form-group">
15     <label><b>Username</b></label>
16     <input type="email" value="{{email}}" name="email" placeholder="Username">
17     <p>The username is Alphanumeric and can contain underscores(_) Must contain atleast 5 C
18
19   <div class="form-group">
20     <label><b>Passsword</b></label>
21     <input value="{{password}}" type="password" name="password" placeholder="Password">
22     <p>The Password is Alphanumeric and can contain @ and hyphens(-) Must contain 8 to 20 C
23   </div>
24
25   <button type="submit" class="registerbtn">Login</button>
26
27

```

Customer login.handlebars:-





```
27 <button type="submit" class="registerbtn">Login</button>
28 <div class="form-group">
29   <p>Don't have an account? <a href="/register">Sign up</a></p>
30 </div>
31
32 <script src="customer_login_js.js"></script>
33 </form>
34 </div>
35
36 </div>
37
38
```

Register.handlebars:-

Canteen Management System

Home Sign up Sign in Admin

Already have an account? [Sign up.](#)

Register

Please fill in this form to create an account.

Name
Enter Full name

Email
harsh.garwal1234@gmail.com

Address
Enter Address

Password
.....

Confirm Password
Repeat Password

Register

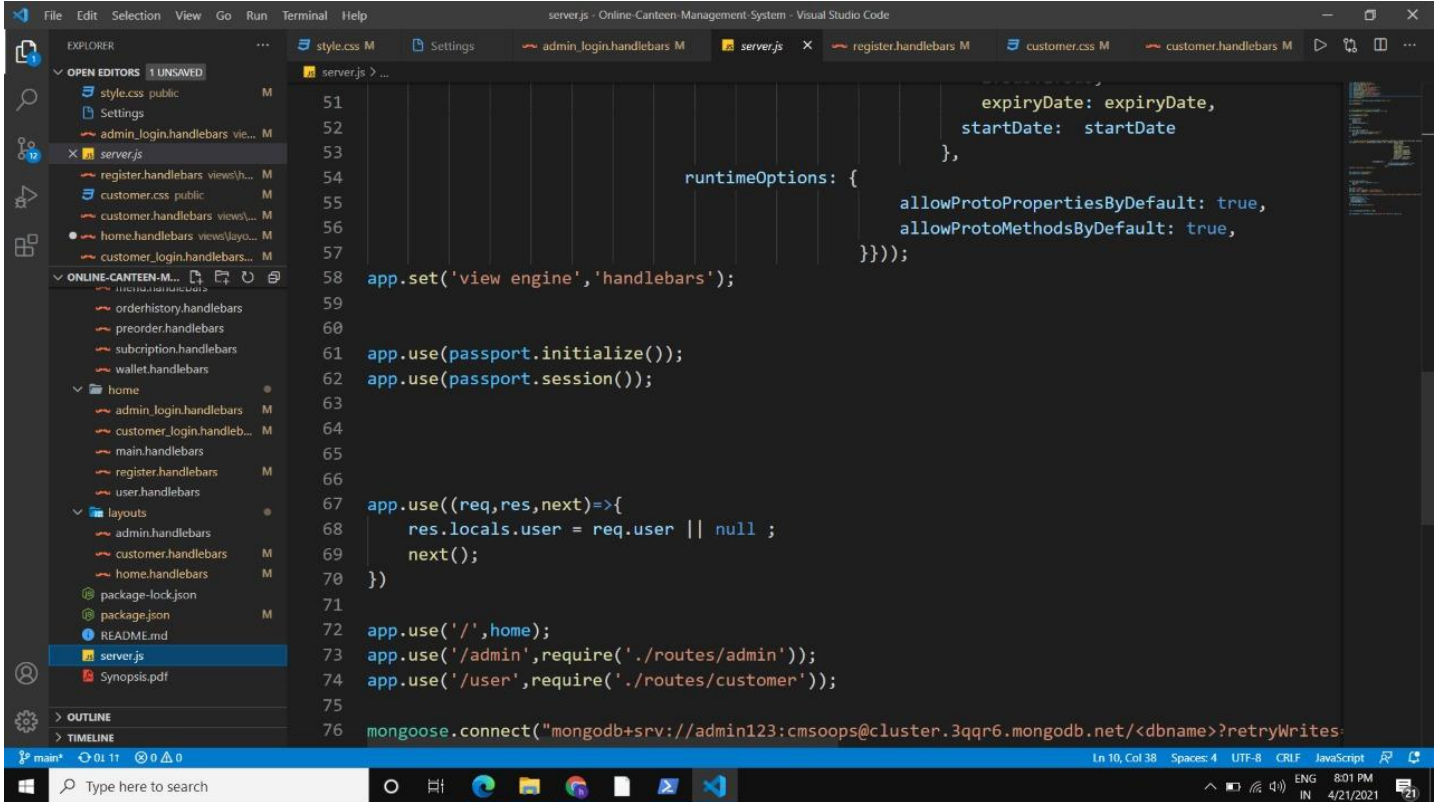
```

register.handlebars - Online-Canteen-Management-System - Visual Studio Code
views > home > register.handlebars > div.registerlanding > div.dark-overlay > div > br
26 | <input name="address" value="{{address}}" type="text" placeholder="Enter Address" />
27 | <div class="form-group">
28 | <label ><b>Password</b></label>
29 | <input name="password" value="{{password}}" type="password" placeholder="Enter Password" />
30 | <div class="form-group">
31 | <label ><b>Confirm Password</b></label>
32 | <input type="password" placeholder="Repeat Password" name="confirmpassword" id="confirmpassword" />
33 |
34 |
35 |
36 | <button type="submit" class="registerbtn">Register</button>
37 | <br>
38 | <br>
39 | </div>
40 | </form>
41 | </div>
42 |
43 | <div class="container signin">
44 | <p>Already have an account? <a href="/customer_login">Sign in</a>.</p>
45 | </div>
46 | </div>
47 |
48 | </div>

```

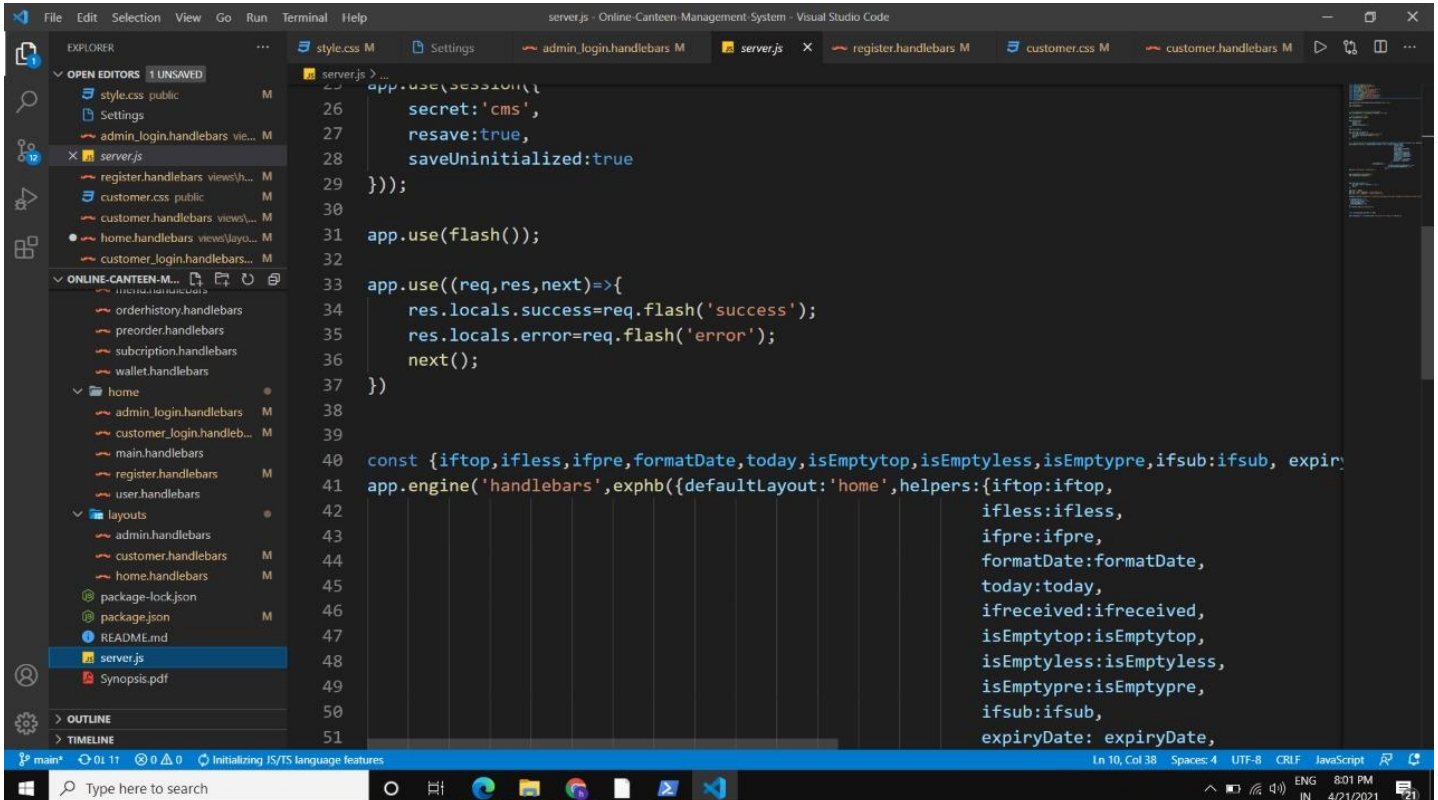



Server.JS: -



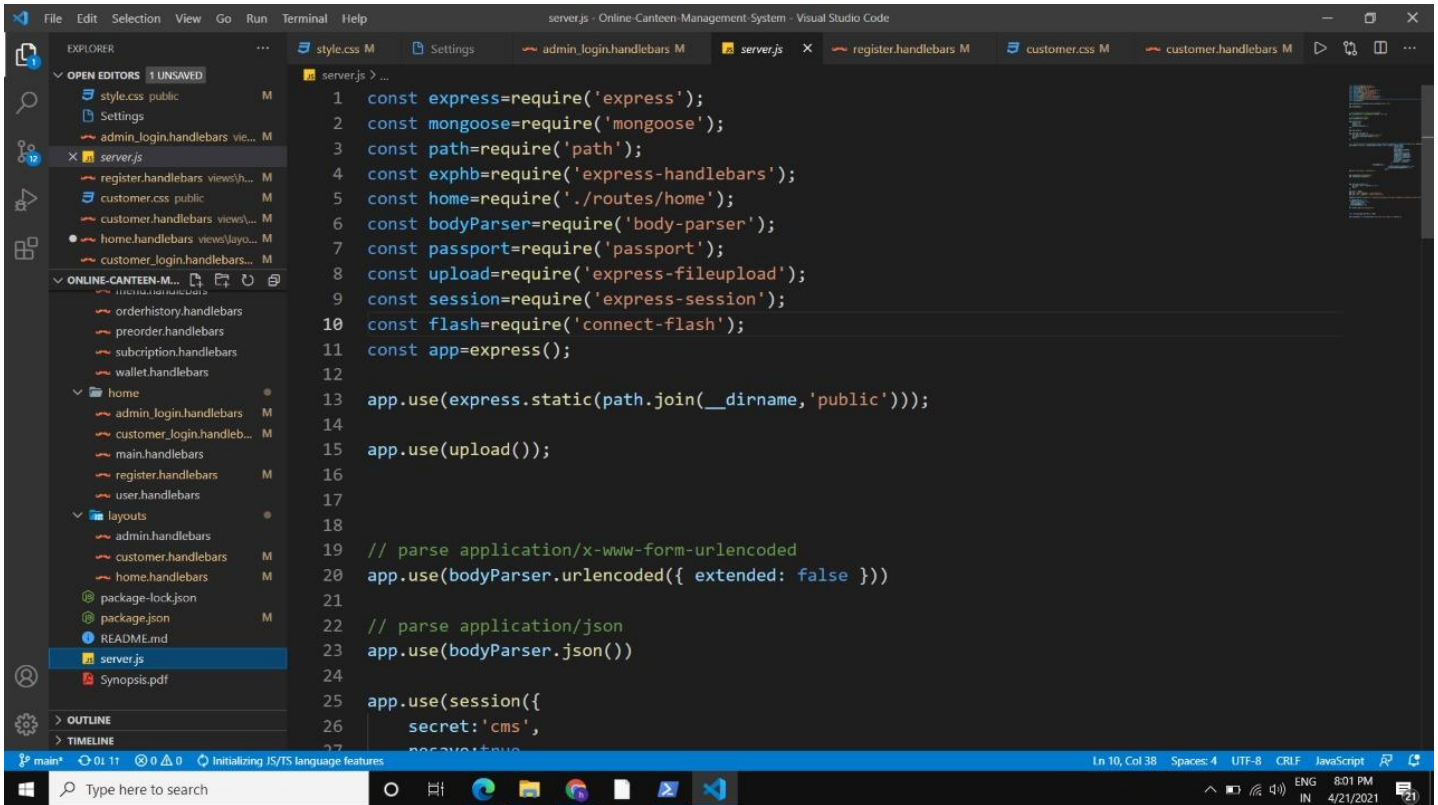
```

51
52
53
54
55
56
57
58 app.set('view engine','handlebars');
59
60
61 app.use(passport.initialize());
62 app.use(passport.session());
63
64
65
66
67 app.use((req,res,next)=>{
68     res.locals.user = req.user || null ;
69     next();
70 })
71
72 app.use('/',home);
73 app.use('/admin',require('./routes/admin'));
74 app.use('/user',require('./routes/customer'));
75
76 mongoose.connect("mongodb+srv://admin123:cmsoops@cluster.3qqr6.mongodb.net/<dbname>?retryWrites=
  
```



```

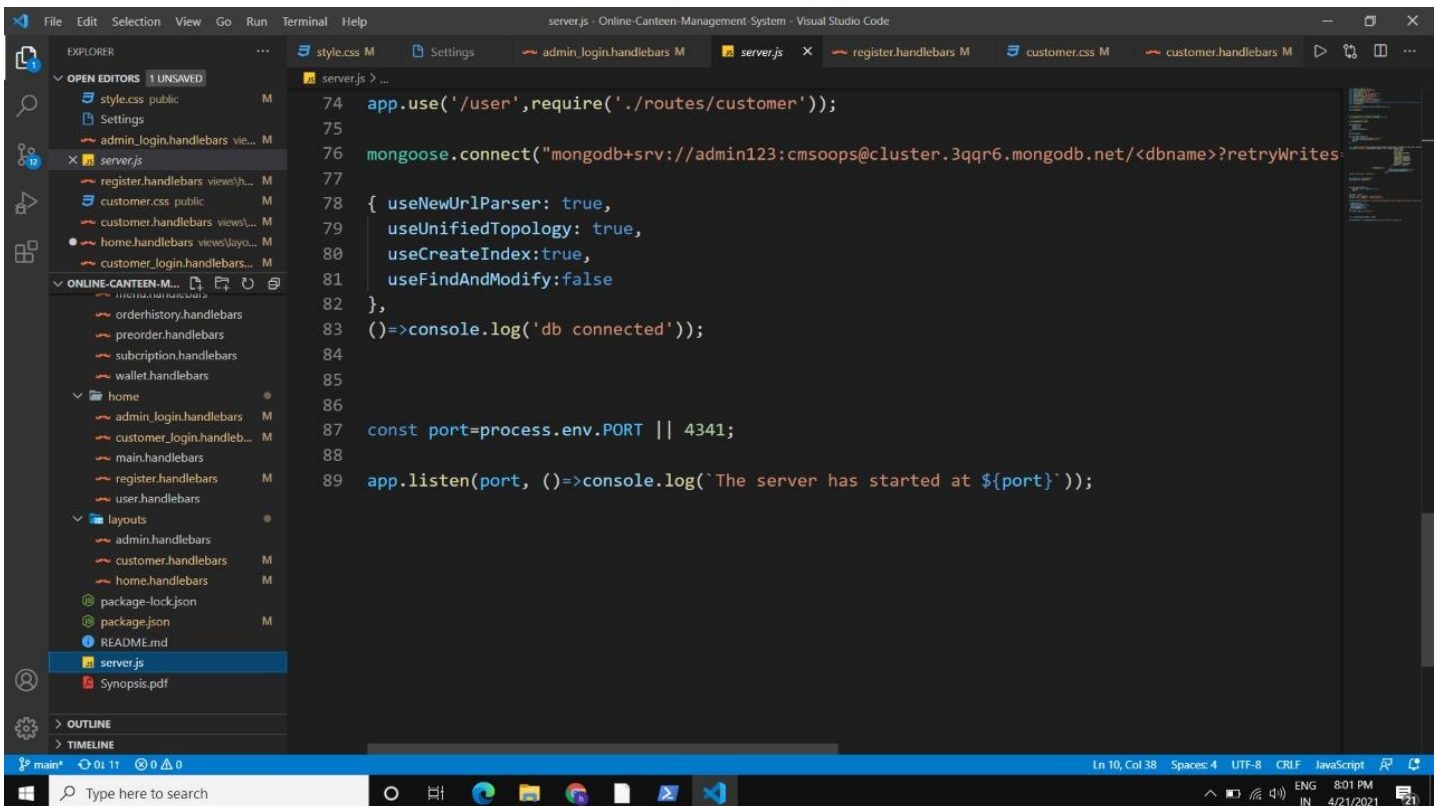
25 app.use(session({
26     secret:'cms',
27     resave:true,
28     saveUninitialized:true
29 }));
30
31 app.use(flash());
32
33 app.use((req,res,next)=>{
34     res.locals.success=req.flash('success');
35     res.locals.error=req.flash('error');
36     next();
37 })
38
39
40 const {iftop,ifless,ifpre,formatDate,today,isEmptytop,isEmptyless,isEmptypre,ifsub:ifsub, expiryDate:expiryDate,
41 app.engine('handlebars',exphb({defaultLayout:'home',helpers:{iftop:iftop,
42     ifless:ifless,
43     ifpre:ifpre,
44     formatDate:formatDate,
45     today:today,
46     ifreceived:ifreceived,
47     isEmptytop:isEmptytop,
48     isEmptyless:isEmptyless,
49     isEmptypre:isEmptypre,
50     ifsub:ifsub,
51     expiryDate:expiryDate,
  
```



```

1  const express=require('express');
2  const mongoose=require('mongoose');
3  const path=require('path');
4  const exphb=require('express-handlebars');
5  const home=require('./routes/home');
6  const bodyParser=require('body-parser');
7  const passport=require('passport');
8  const upload=require('express-fileupload');
9  const session=require('express-session');
10 const flash=require('connect-flash');
11 const app=express();
12
13 app.use(express.static(path.join(__dirname,'public')));
14
15 app.use(upload());
16
17
18
19 // parse application/x-www-form-urlencoded
20 app.use(bodyParser.urlencoded({ extended: false }))
21
22 // parse application/json
23 app.use(bodyParser.json())
24
25 app.use(session({
26   secret:'cms',
27   resave:true,
28   saveUninitialized:true
29 }));
30
31 app.use(flash());
32
33 // passport session
34 app.use(passport.initialize());
35 app.use(passport.session());
36
37 // home routes
38 app.use(home);
39
40 // register route
41 app.use(register);
42
43 // login route
44 app.use(login);
45
46 // subscription route
47 app.use(subscription);
48
49 // order history route
50 app.use(orderhistory);
51
52 // pre order route
53 app.use(preorder);
54
55 // wallet route
56 app.use(wallet);
57
58 // user route
59 app.use(user);
60
61 // admin route
62 app.use(admin);
63
64 // customer route
65 app.use(customer);
66
67 // home route
68 app.use(home);
69
70 // layouts
71 app.use(layouts);
72
73 // package-lock.json
74
75 // package.json
76
77 // README.md
78
79 // server.js
80
81 // Synopsis.pdf
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

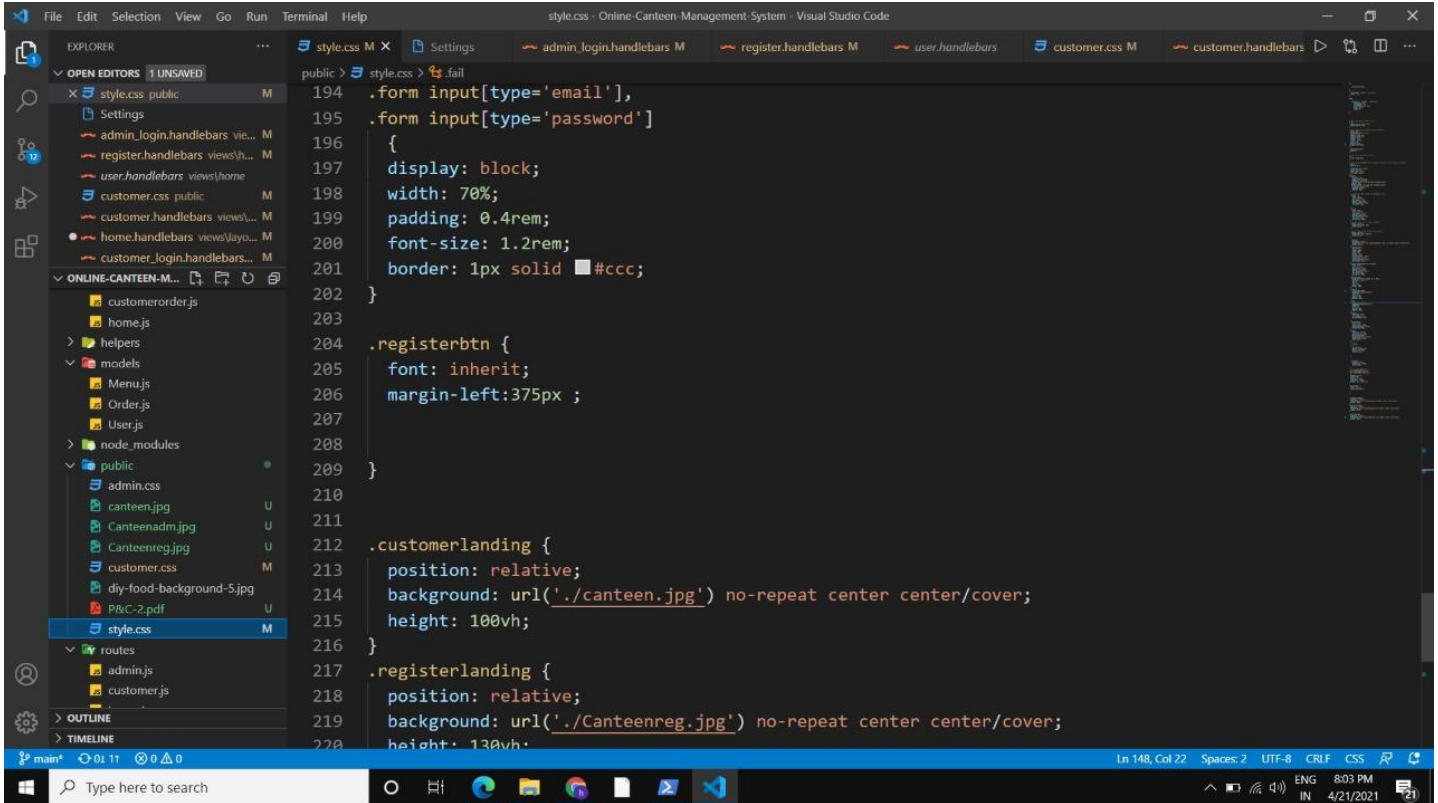


```

74 app.use('/user',require('./routes/customer'));
75
76 mongoose.connect("mongodb+srv://admin123:cmsoops@cluster.3qqr6.mongodb.net/<dbname>?retryWrites=true&ssl=true&sslValidate=false");
77
78 { useNewUrlParser: true,
79   useUnifiedTopology: true,
80   useCreateIndex:true,
81   useFindAndModify:false
82 },
83 ()=>console.log('db connected'));
84
85
86
87 const port=process.env.PORT || 4341;
88
89 app.listen(port, ()=>console.log(`The server has started at ${port}`));
90
91
92
93
94
95
96
97
98
99
100

```

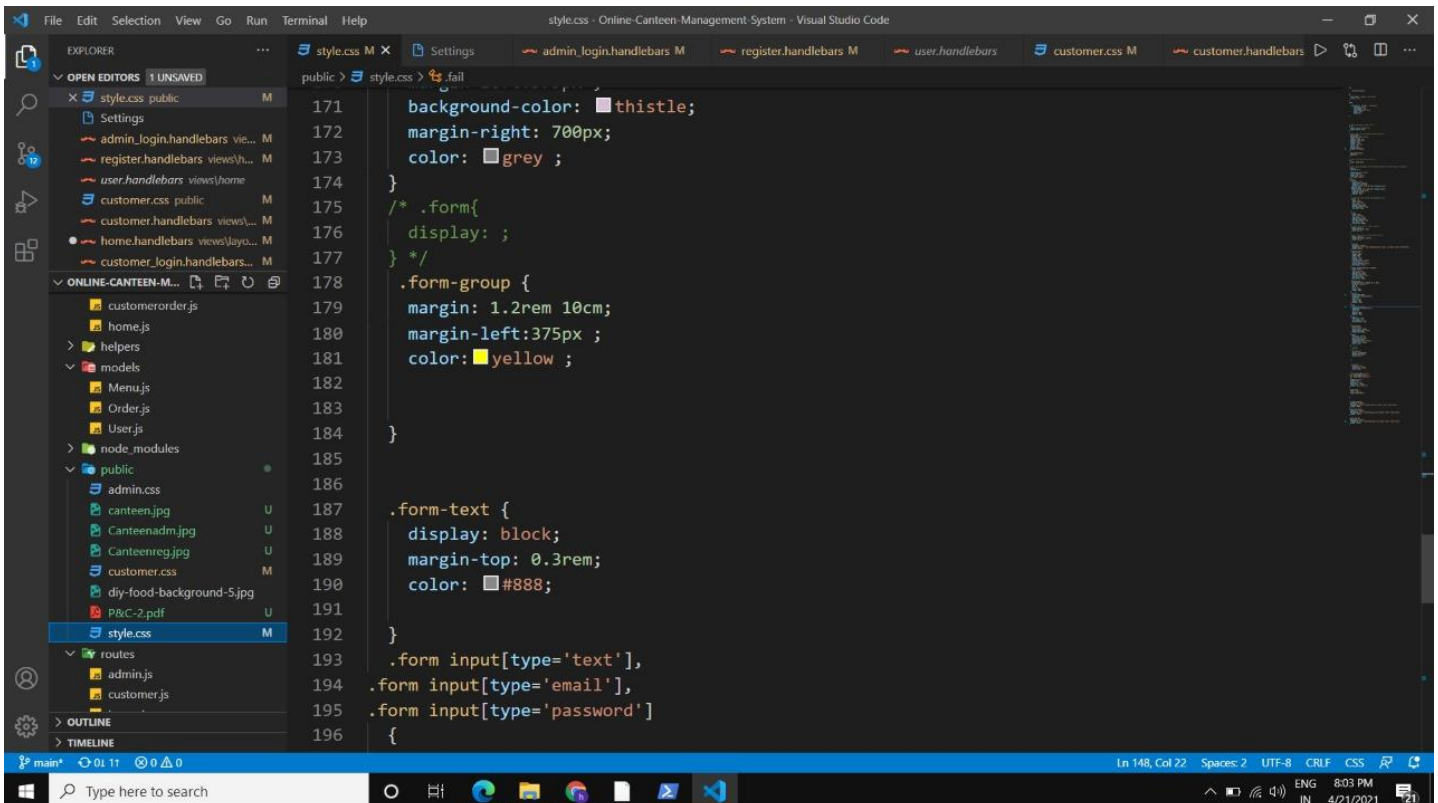

Style.css: -



```

194 .form input[type='email'],
195 .form input[type='password']
196 {
197     display: block;
198     width: 70%;
199     padding: 0.4rem;
200     font-size: 1.2rem;
201     border: 1px solid #ccc;
202 }
203
204 .registerbtn {
205     font: inherit;
206     margin-left: 375px;
207 }
208
209
210
211
212 .customerlanding {
213     position: relative;
214     background: url('./canteen.jpg') no-repeat center center/cover;
215     height: 100vh;
216 }
217
218 .registerlanding {
219     position: relative;
220     background: url('./Canteenreg.jpg') no-repeat center center/cover;
221     height: 130vh;

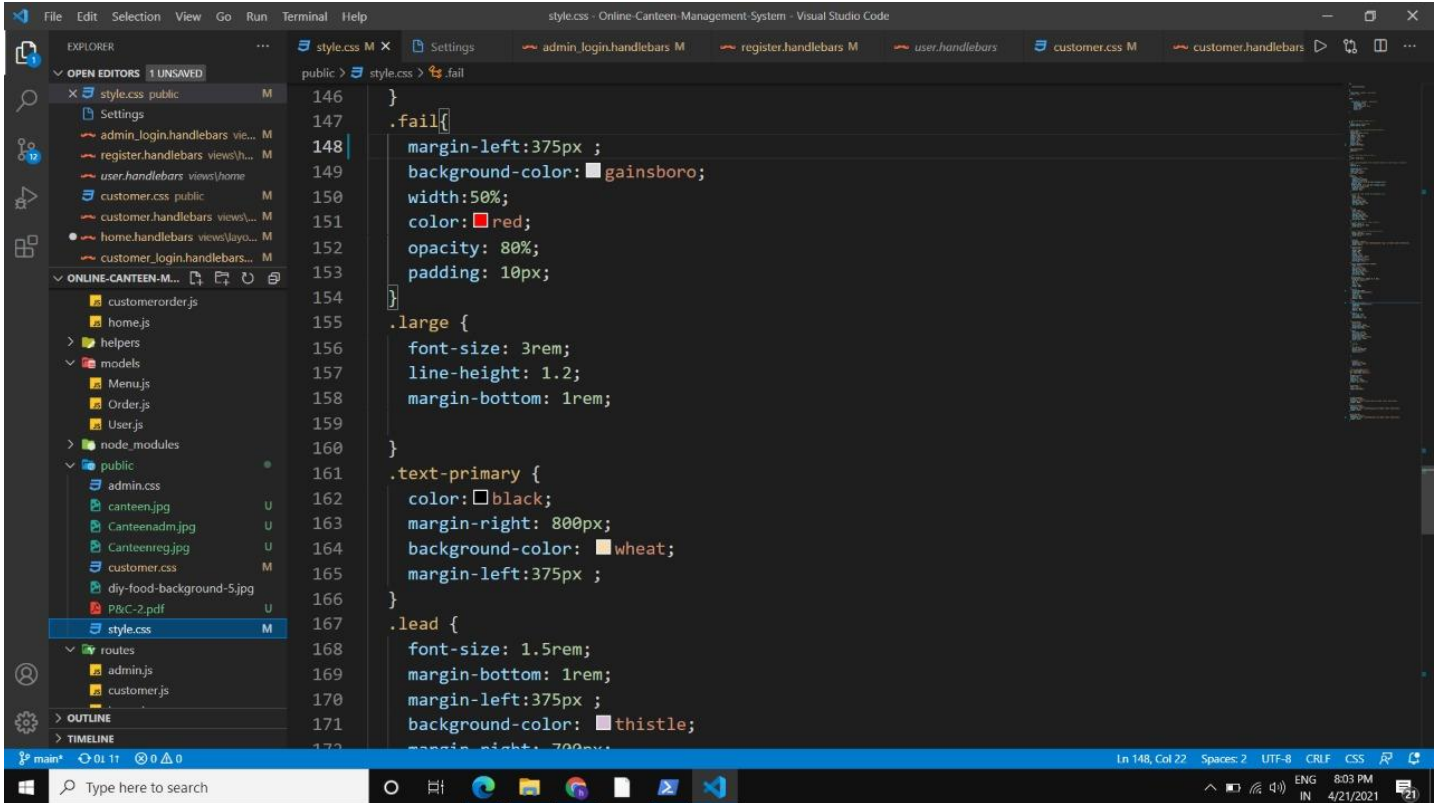
```



```

171 background-color: #thistle;
172 margin-right: 700px;
173 color: #grey;
174 }
175
176 /* .form{
177     display: ;
178 } */
179
180 .form-group {
181     margin: 1.2rem 10cm;
182     margin-left: 375px;
183     color: #yellow;
184 }
185
186
187
188
189
190
191
192
193
194 .form-text {
195     display: block;
196     margin-top: 0.3rem;
197     color: #888;
198 }
199
200 .form input[type='text'],
201 .form input[type='email'],
202 .form input[type='password']
203 {

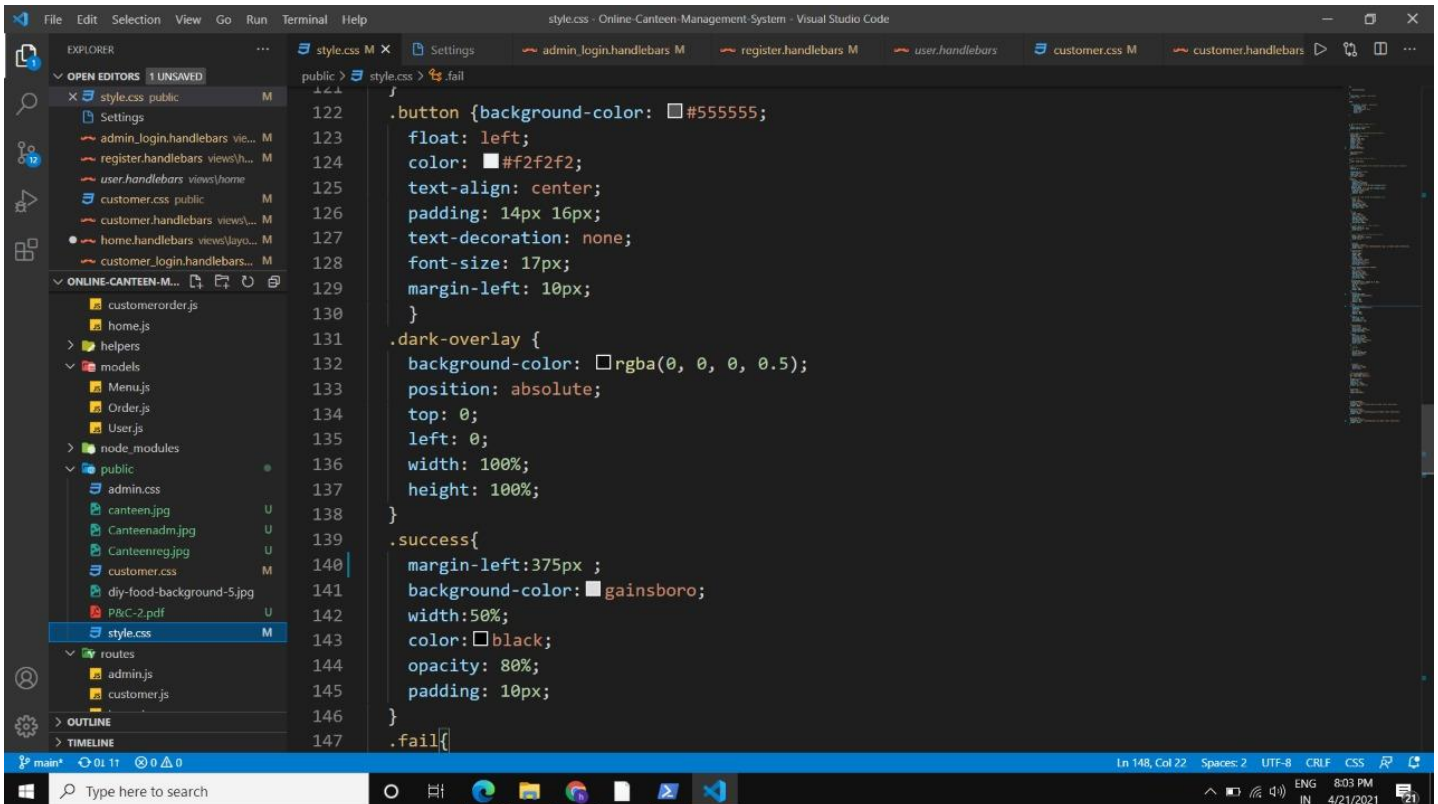
```



```

146 }
147
148 .fail{
149     margin-left:375px ;
150     background-color: #gainsboro;
151     width:50%;
152     color: #red;
153     opacity: 80%;
154     padding: 10px;
155 }
156
157 .large {
158     font-size: 3rem;
159     line-height: 1.2;
160     margin-bottom: 1rem;
161 }
162
163 .text-primary {
164     color: #black;
165     margin-right: 800px;
166     background-color: #wheat;
167     margin-left:375px ;
168 }
169
170 .lead {
171     font-size: 1.5rem;
172     margin-bottom: 1rem;
173     margin-left:375px ;
174     background-color: #thistle;
175     margin-right: 700px;

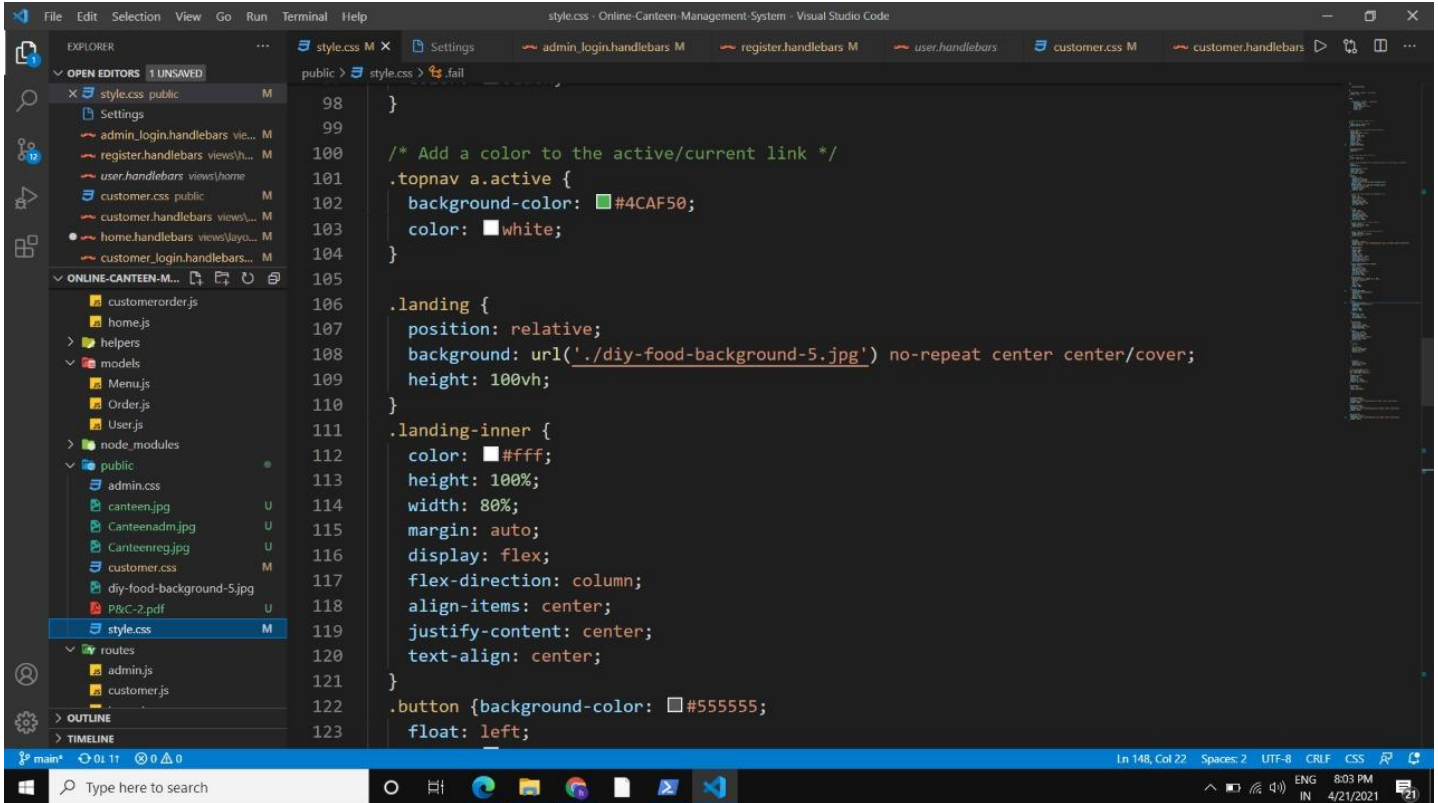
```



```

122 }
123
124 .button {background-color: #555555;
125     float: left;
126     color: #f2f2f2;
127     text-align: center;
128     padding: 14px 16px;
129     text-decoration: none;
130     font-size: 17px;
131     margin-left: 10px;
132 }
133
134 .dark-overlay {
135     background-color: rgba(0, 0, 0, 0.5);
136     position: absolute;
137     top: 0;
138     left: 0;
139     width: 100%;
140     height: 100%;
141 }
142
143 .success{
144     margin-left:375px ;
145     background-color: #gainsboro;
146     width:50%;
147     color: #black;
148     opacity: 80%;
149     padding: 10px;
150 }
151
152 .fail{

```



```

style.css - Online-Canteen-Management-System - Visual Studio Code
public > style.css > .fail
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123

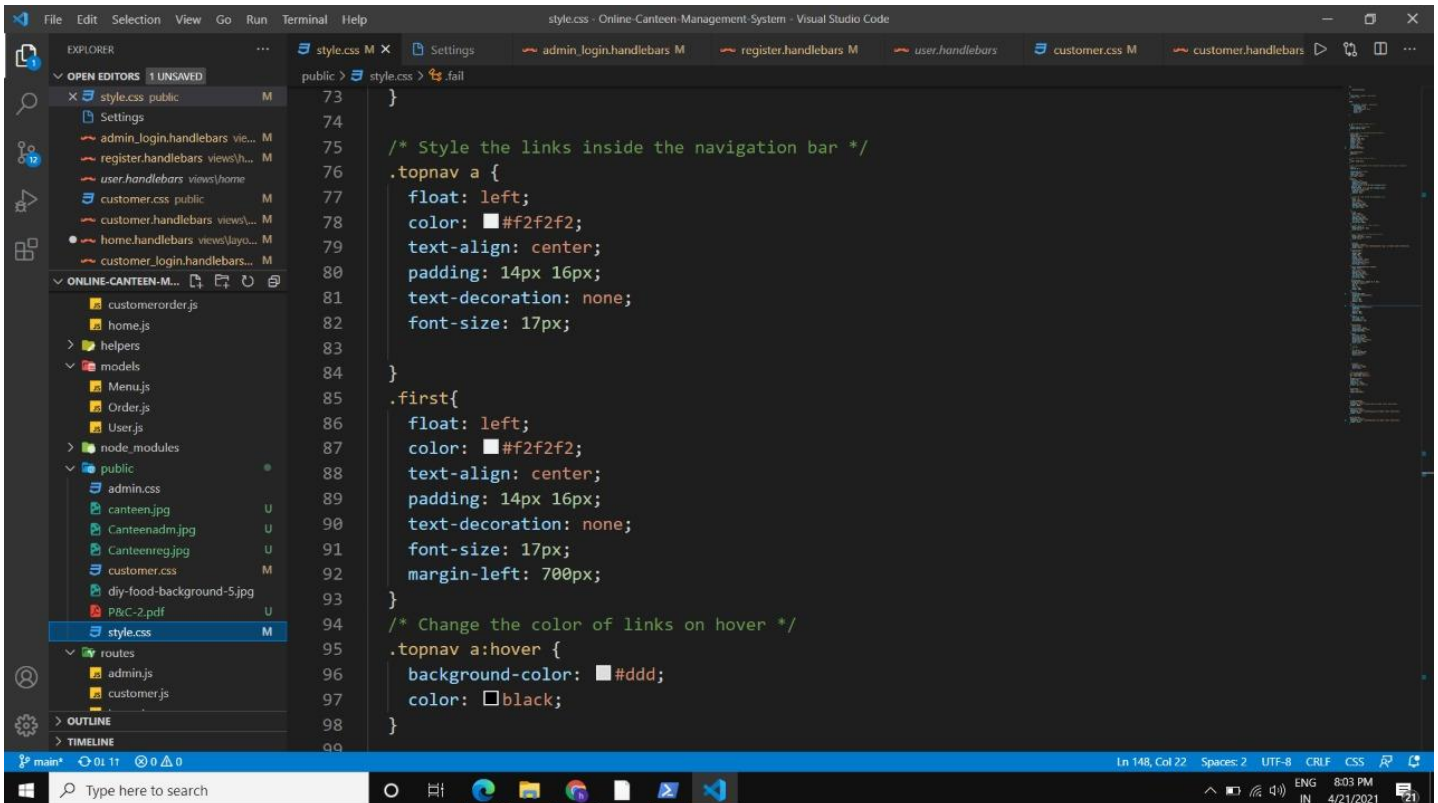
/* Add a color to the active/current link */
.topnav a.active {
  background-color: #4CAF50;
  color: white;
}

.landing {
  position: relative;
  background: url('./diy-food-background-5.jpg') no-repeat center center/cover;
  height: 100vh;
}

.landing-inner {
  color: #fff;
  height: 100%;
  width: 80%;
  margin: auto;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  text-align: center;
}

button {background-color: #555555;
float: left;

```



```

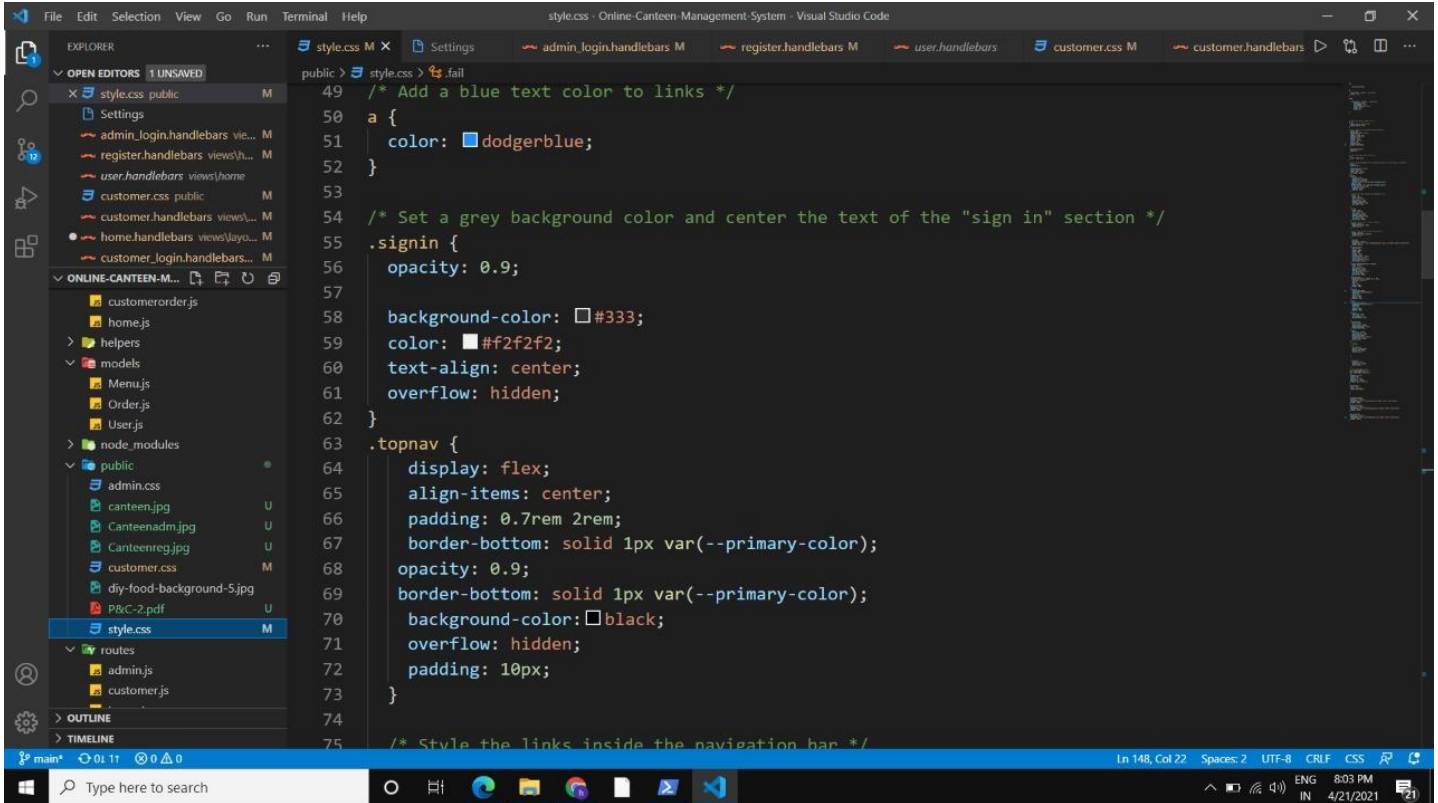
style.css - Online-Canteen-Management-System - Visual Studio Code
public > style.css > .fail
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

.first{
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
  margin-left: 700px;
}

/* Change the color of links on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

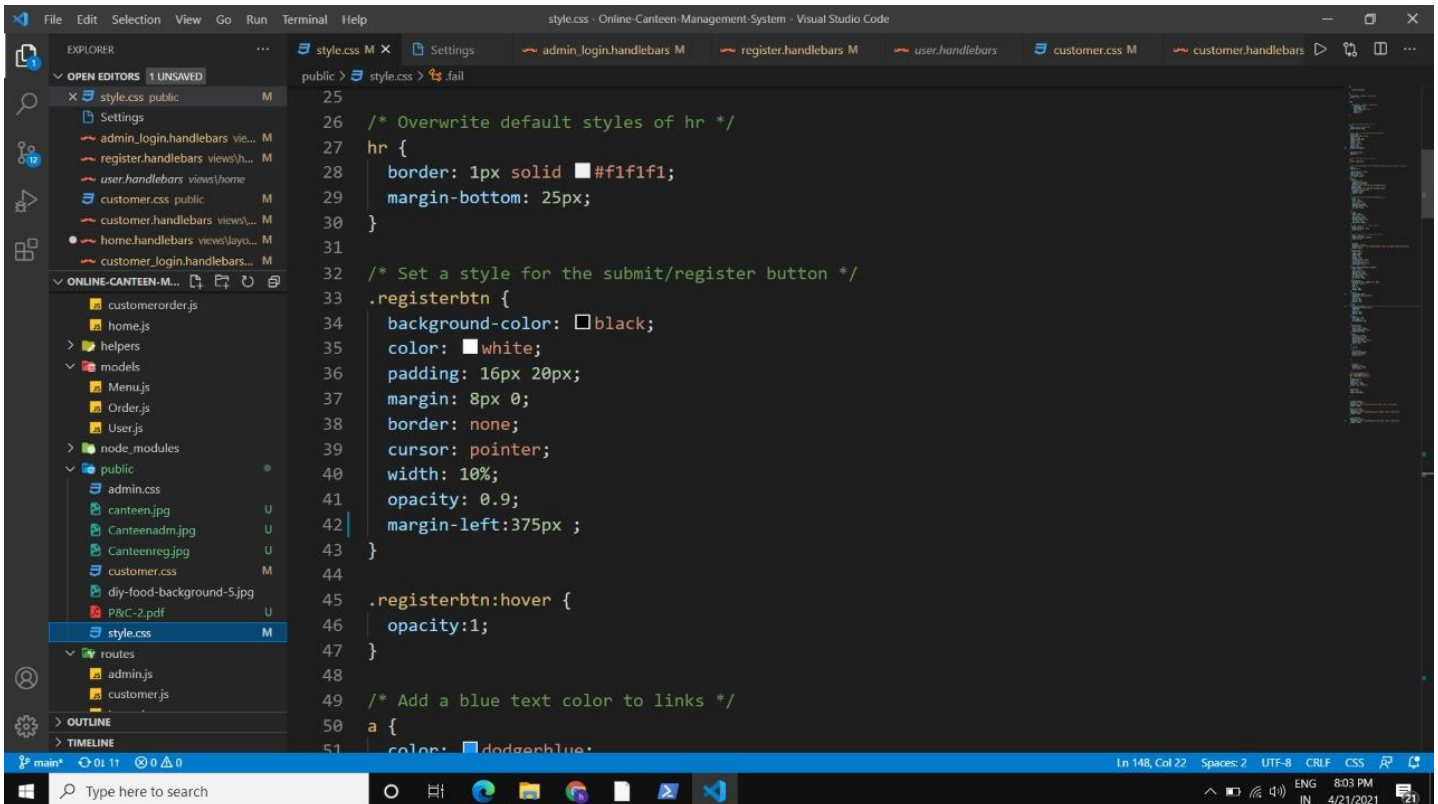
```

```

public > style.css > .fail
49 /* Add a blue text color to links */
50 a {
51   color: dodgerblue;
52 }
53
54 /* Set a grey background color and center the text of the "sign in" section */
55 .signin {
56   opacity: 0.9;
57
58   background-color: #333;
59   color: #f2f2f2;
60   text-align: center;
61   overflow: hidden;
62 }
63 .topnav {
64   display: flex;
65   align-items: center;
66   padding: 0.7rem 2rem;
67   border-bottom: solid 1px var(--primary-color);
68   opacity: 0.9;
69   border-bottom: solid 1px var(--primary-color);
70   background-color: black;
71   overflow: hidden;
72   padding: 10px;
73 }
74
75 /* Style the links inside the navigation bar */

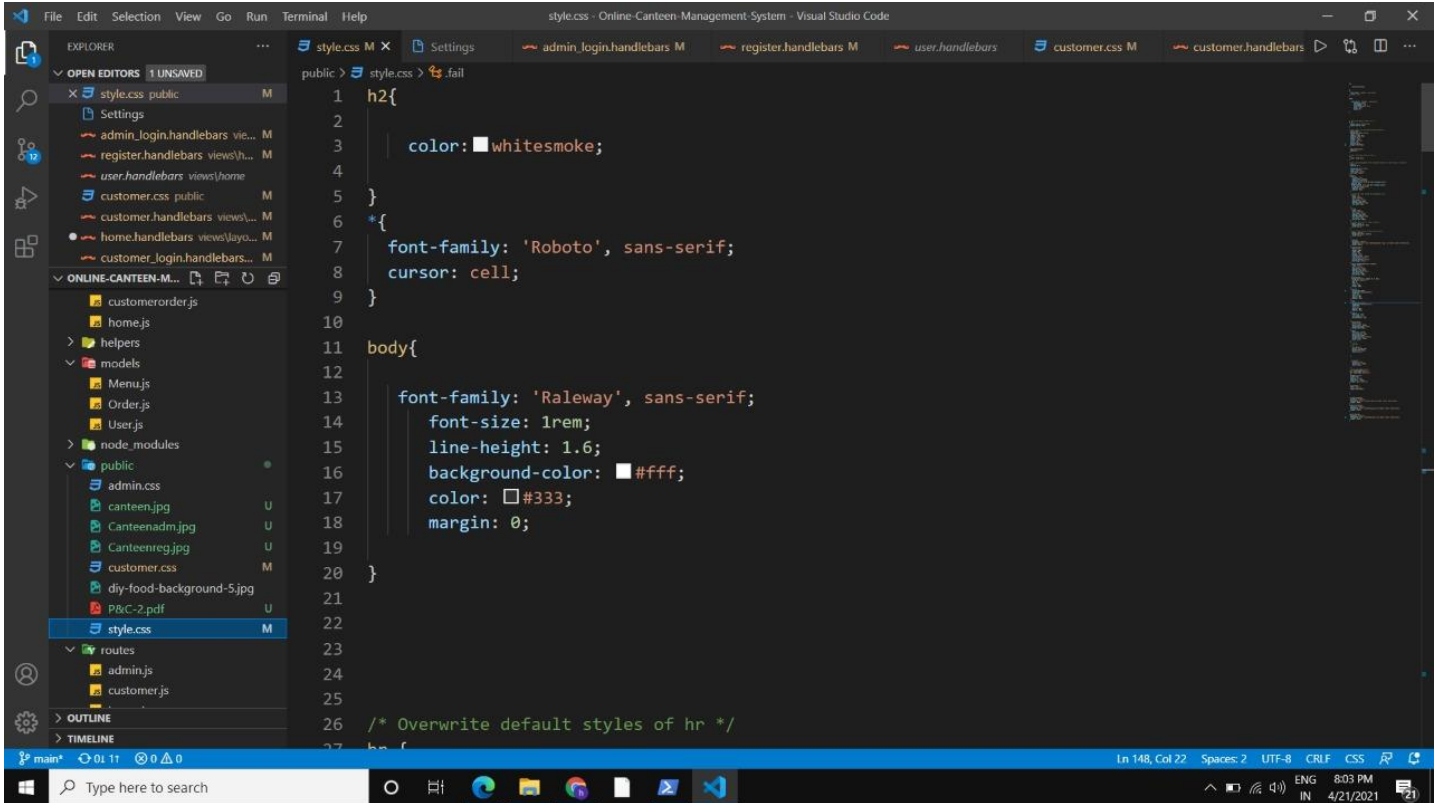
```



```

public > style.css > .fail
25
26 /* Overwrite default styles of hr */
27 hr {
28   border: 1px solid #f1f1f1;
29   margin-bottom: 25px;
30 }
31
32 /* Set a style for the submit/register button */
33 .registerbtn {
34   background-color: black;
35   color: white;
36   padding: 16px 20px;
37   margin: 8px 0;
38   border: none;
39   cursor: pointer;
40   width: 10%;
41   opacity: 0.9;
42   margin-left: 375px;
43 }
44
45 .registerbtn:hover {
46   opacity: 1;
47 }
48
49 /* Add a blue text color to links */
50 a {
51   color: dodgerblue;

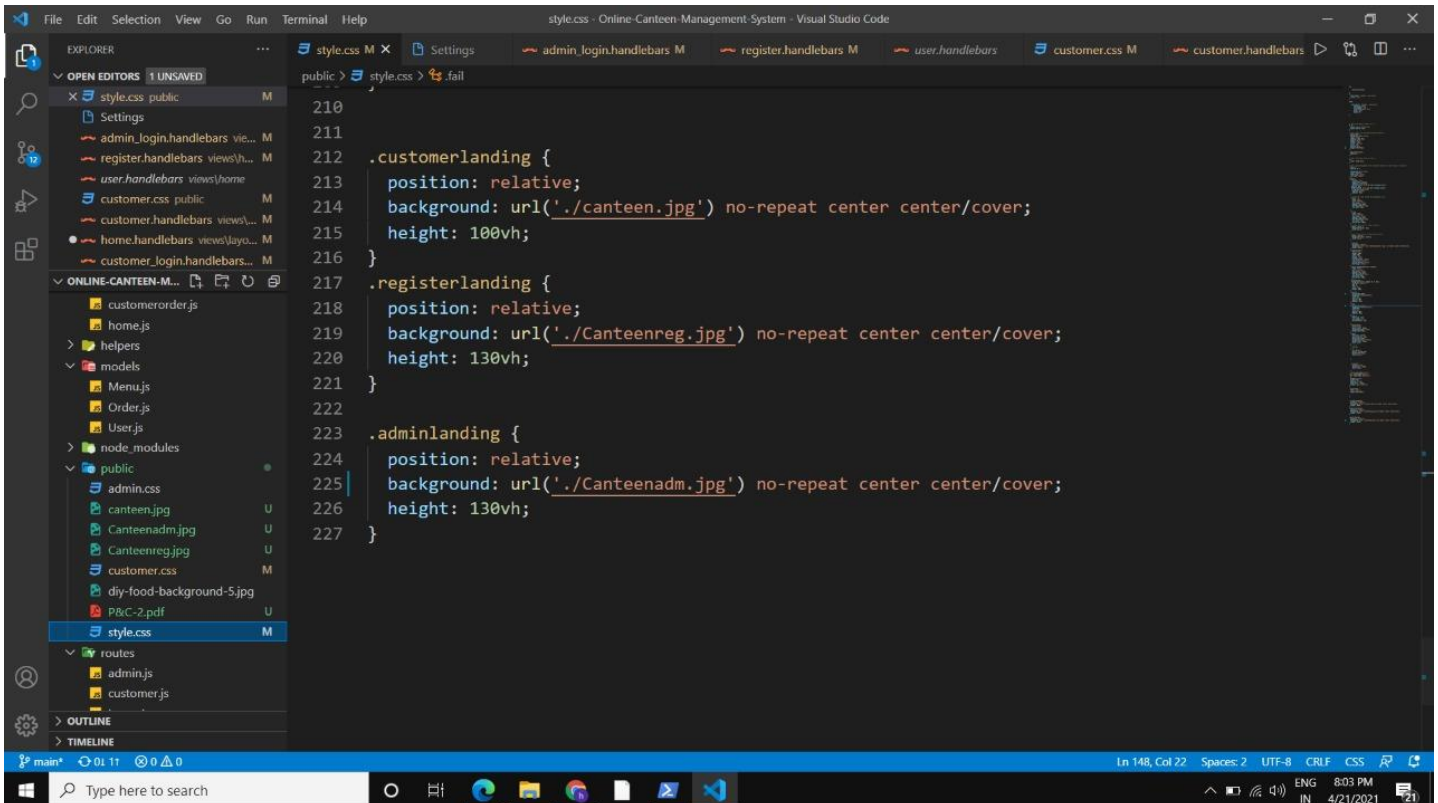
```



```

1  h2{
2
3      color: whitesmoke;
4
5  }
6  *{
7      font-family: 'Roboto', sans-serif;
8      cursor: cell;
9  }
10
11 body{
12
13     font-family: 'Raleway', sans-serif;
14     font-size: 1rem;
15     line-height: 1.6;
16     background-color: #fff;
17     color: #333;
18     margin: 0;
19
20 }
21
22
23
24
25
26 /* Overwrite default styles of hr */
27

```

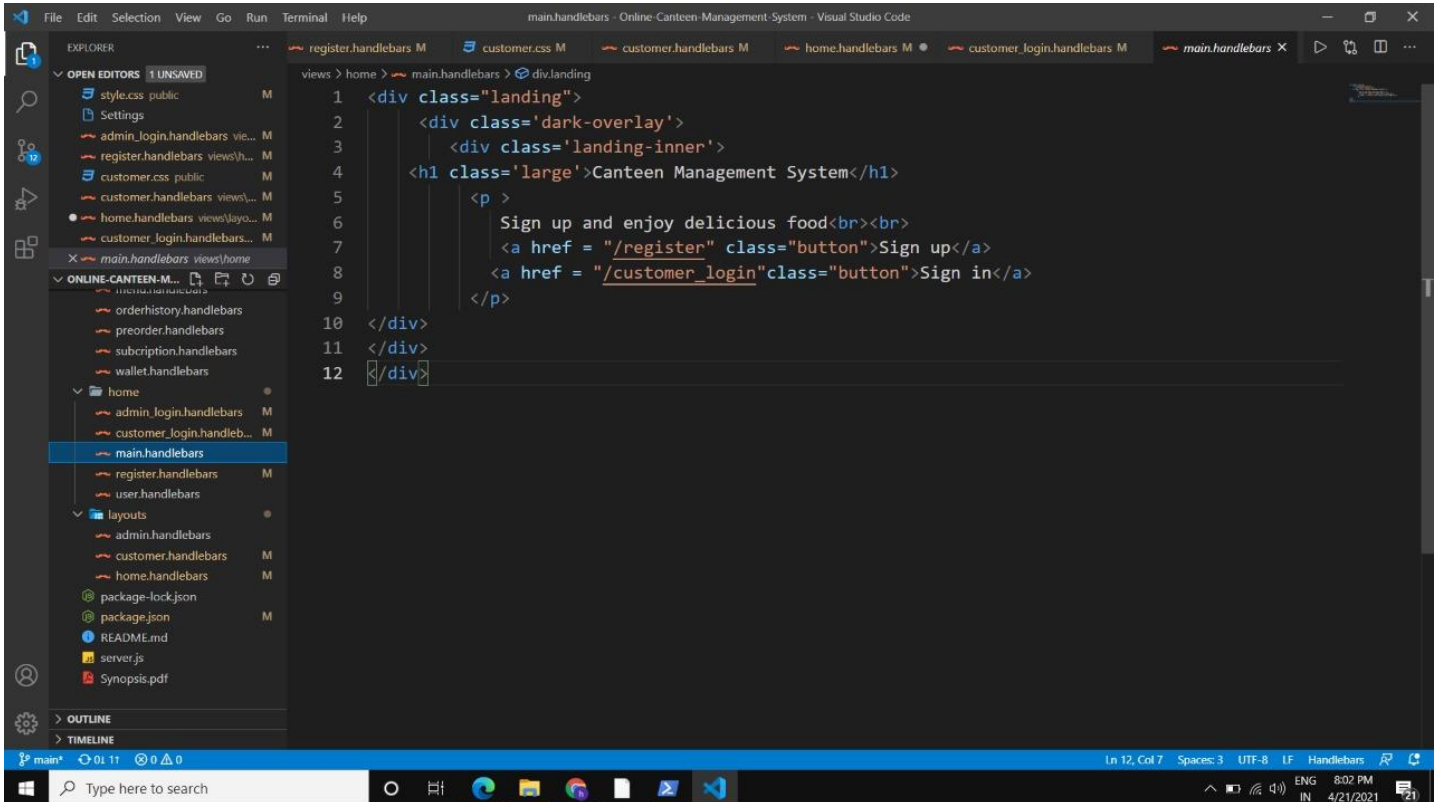


```

210
211
212 .customerlanding {
213     position: relative;
214     background: url('../canteen.jpg') no-repeat center center/cover;
215     height: 100vh;
216 }
217
218 .registerlanding {
219     position: relative;
220     background: url('../Canteenreg.jpg') no-repeat center center/cover;
221     height: 130vh;
222 }
223
224 .adminlanding {
225     position: relative;
226     background: url('../Canteenadm.jpg') no-repeat center center/cover;
227     height: 130vh;
228 }
229

```


Main.handlebars:-



```
1 <div class="landing">
2   <div class='dark-overlay'>
3     <div class='landing-inner'>
4       <h1 class='large'>Canteen Management System</h1>
5       <p>
6         Sign up and enjoy delicious food<br><br>
7         <a href = "/register" class="button">Sign up</a>
8         <a href = "/customer_login" class="button">Sign in</a>
9       </p>
10    </div>
11  </div>
12 </div>
```

References

- <https://www.w3schools.com/>
- <https://www.wikipedia.org/>
- <https://stackoverflow.com/>
- <https://www.javatpoint.com/>
- [YouTube](#)