

MINI Project Report

2023-2024 6th semester

PROBLEM STATEMENT:

An automated system to manage accounting and service records for owners, employees, and customers

Under The Guidance of:

Guide Name: Ms. Prathimarani Jena

Designation: ASST Prof.

Team 11:

Amruthalakshmi N -1CR21AD010

Malavika -1CR21AD031

Shanvi P -1CR21AD050

Aman -1CR21AD049

Index

Offer letter.....	3
Abstract	4
Learning Objectives.....	4
Tasks Handled	5
Implementation	6
Output	7-10
Conclusion	11

From,

SHANVI (1CR21AD050)

MALAVIKA (1C21AD031)

AMAN (1CR21AD049)

AMRUTHA (1CR21AD010)

TO,

MR. BABU (PROPREITOR)

WELDONE SALONS BANGLORE

Respected sir,

We are the students of CMR Institute of Technology Bangalore from the Artificial Intelligence and Data Science department.

As a part of our 6th semester entrepreneurship project, we would like to seek this opportunity to build an accounting software for your enterprise. We have been able to collect the list of requirements for the same from your manager and staff.

We assure you that we will proceed with our project in accordance with the requirements of your enterprise.

We hereby look forward to your approval for the further proceedings of this project.

Thanks and Regards,

SHANVI

MALAVIKA

AMAN

AMRUTHA

CMRIT, Bangalore

**WELL DONE
SALON FOR MEN**
No. 158/7, 8-1
Vignesh Nagar Main Road
Bangalore - 560 037

Approved

22/5/23

Abstract:

The project presents an accounting management system tailored for the Weldone Group of Salons, featuring distinct user interfaces for owners, employees, and customers. Developed using HTML, CSS, and JavaScript, the system facilitates login and customized dashboards for each user role. Key functionalities include service selection, cost calculation, commission determination for employees, and revenue tracking for the owner. The system automates accounting tasks, enhancing accuracy and efficiency while minimizing manual errors. By addressing the specific needs of each user role, the application improves overall operational effectiveness and provides a streamlined solution for salon management.

Learning Objectives:

The primary learning objective of this project was to understand and implement a role-based accounting system using web technologies. This included:

1. Designing a responsive user interface with HTML and CSS.
2. Developing interactive features with JavaScript.
3. Managing different user roles and their specific functionalities.
4. Calculating total service costs and commissions dynamically.

Tasks Handled:

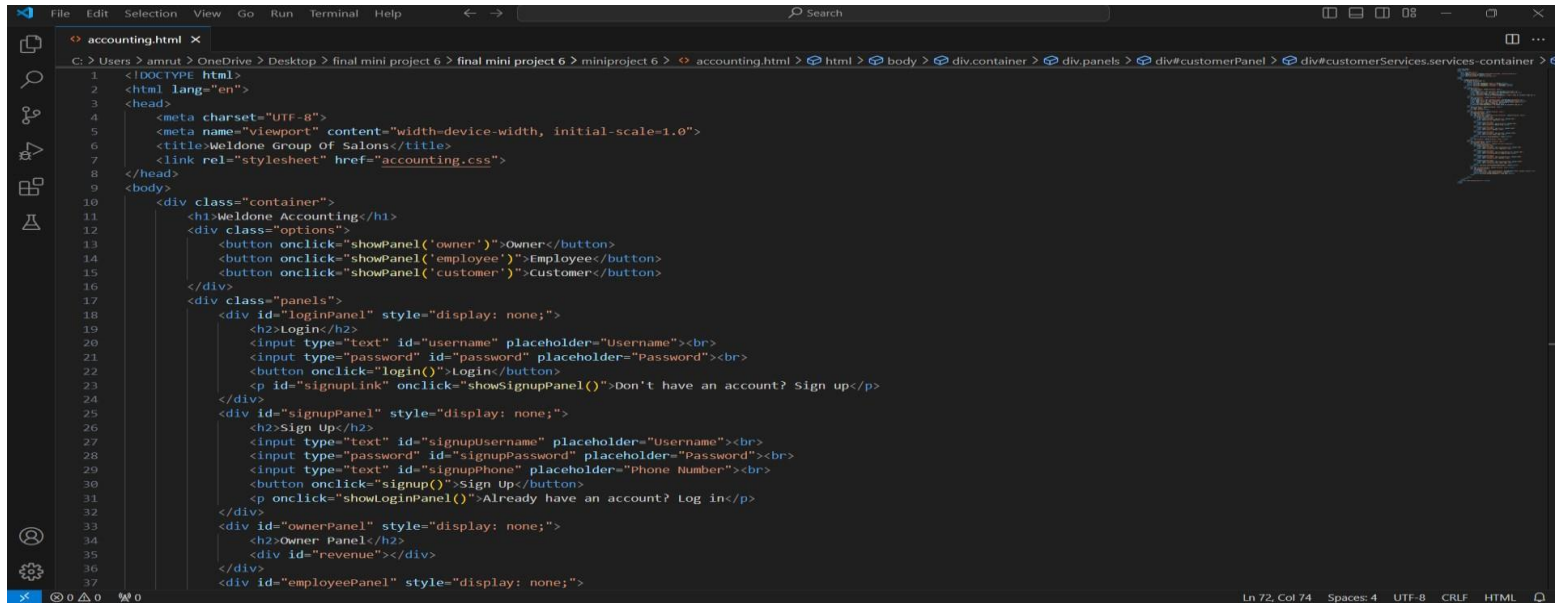
The project involved the following tasks:

- **UI/UX Design:** Creating a visually appealing and user-friendly interface for the application.
- **Role-based Panels:** Implementing login functionality and role-specific panels for owners, employees, and customers.
- **Service Selection:** Allowing employees and customers to select services and calculate total costs.
- **Commission Calculation:** Computing commissions for employees based on the selected services.
- **Revenue Tracking:** Enabling the owner to track total revenue generated by employees.

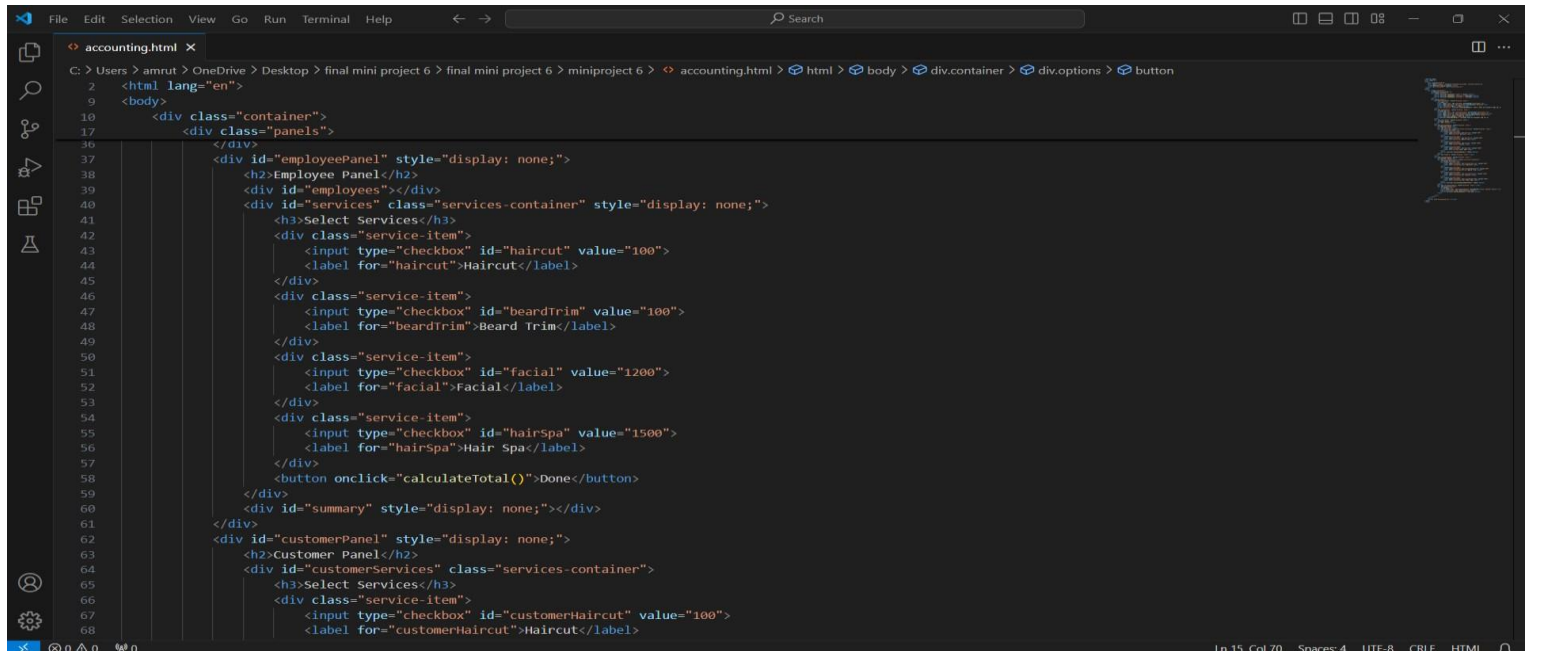
Implementation:

- **Requirements Gathering:** Identify the needs of owners, employees, and customers for the accounting system.
- **Design UI/UX:** Create wireframes and design the user interface using HTML and CSS to ensure a responsive and intuitive layout.
- **Set Up Project Structure:** Organize the project files, including HTML, CSS, and JavaScript files.
- **Develop HTML Structure:** Implement the basic HTML structure to create panels for owner, employee, and customer login and functionalities.
- **Style with CSS:** Apply styles to the HTML structure to enhance the visual appearance and responsiveness of the application.
- **Implement JavaScript Logic:** Write JavaScript functions to handle login, display role-specific panels, and manage service selection and cost calculations.
- **Add Interactivity:** Use JavaScript to make the application interactive, including showing and hiding panels, calculating totals, and displaying summaries.
- **Testing:** Conduct thorough testing to ensure all functionalities work correctly and fix any bugs or issues that arise.
- **Deployment:** Deploy the application to a web server or hosting service so it can be accessed by users.
- **Maintenance and Updates:** Monitor the application for any issues and update it with new features or improvements as needed.

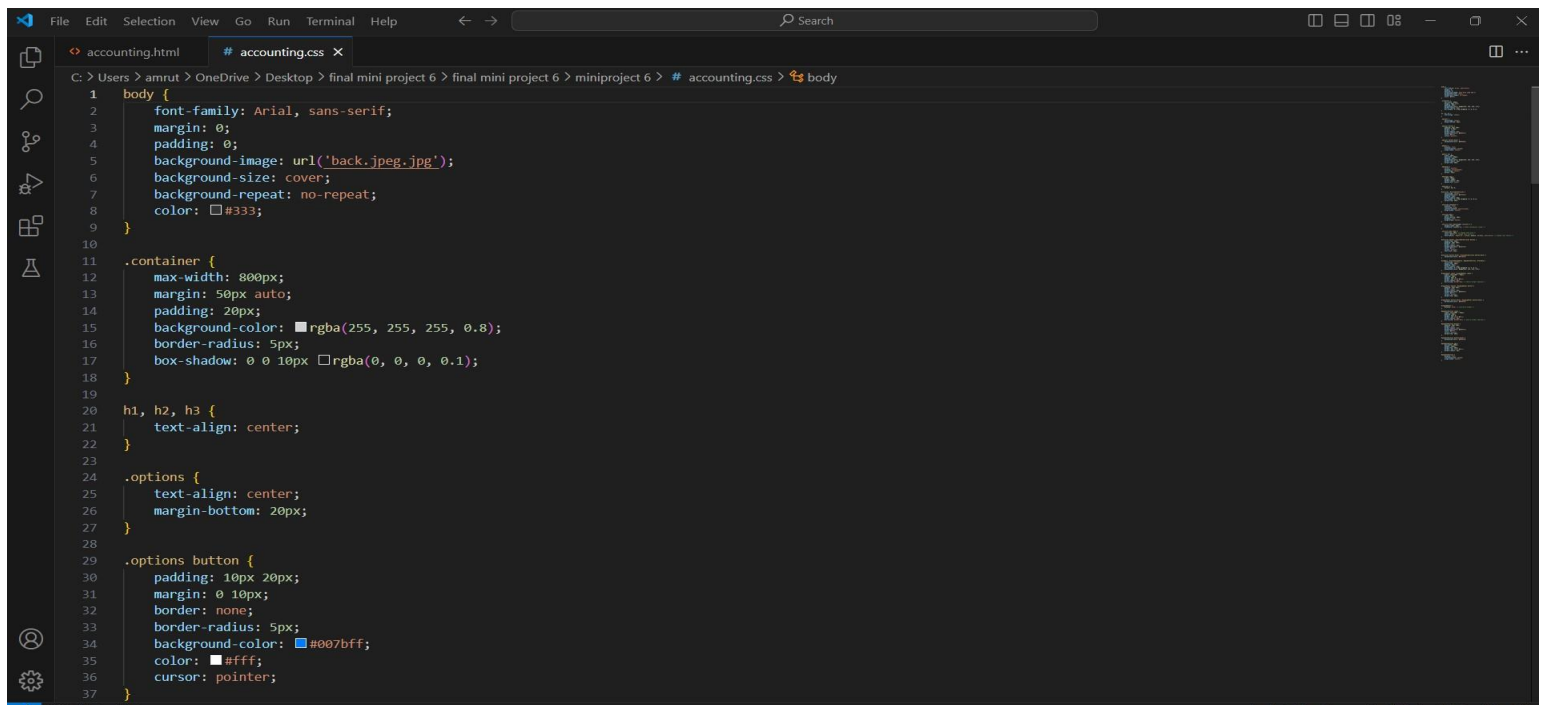
Screenshots:



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Weldone Group of Salons</title>
7   <link rel="stylesheet" href="accounting.css">
8 </head>
9 <body>
10   <div class="container">
11     <h1>Weldone Accountings</h1>
12     <div class="options">
13       <button onclick="showPanel('owner')">Owner</button>
14       <button onclick="showPanel('employee')">Employee</button>
15       <button onclick="showPanel('customer')">Customer</button>
16     </div>
17     <div class="panels">
18       <div id="loginPanel" style="display: none;">
19         <h2>Login</h2>
20         <input type="text" id="username" placeholder="Username"><br>
21         <input type="password" id="password" placeholder="Password"><br>
22         <button onclick="login()">Login</button>
23         <p id="signuLink" onclick="showSignupPanel()">Don't have an account? Sign up</p>
24       </div>
25       <div id="signupPanel" style="display: none;">
26         <h2>Sign Up</h2>
27         <input type="text" id="signupUsername" placeholder="Username"><br>
28         <input type="password" id="signupPassword" placeholder="Password"><br>
29         <input type="text" id="signupPhone" placeholder="Phone Number"><br>
30         <button onclick="signup()">Sign Up</button>
31         <p onclick="showLoginPanel()">Already have an account? Log in</p>
32       </div>
33       <div id="ownerPanel" style="display: none;">
34         <h2>Owner Panel</h2>
35         <div id="revenue"></div>
36       </div>
37       <div id="employeePanel" style="display: none;">
```

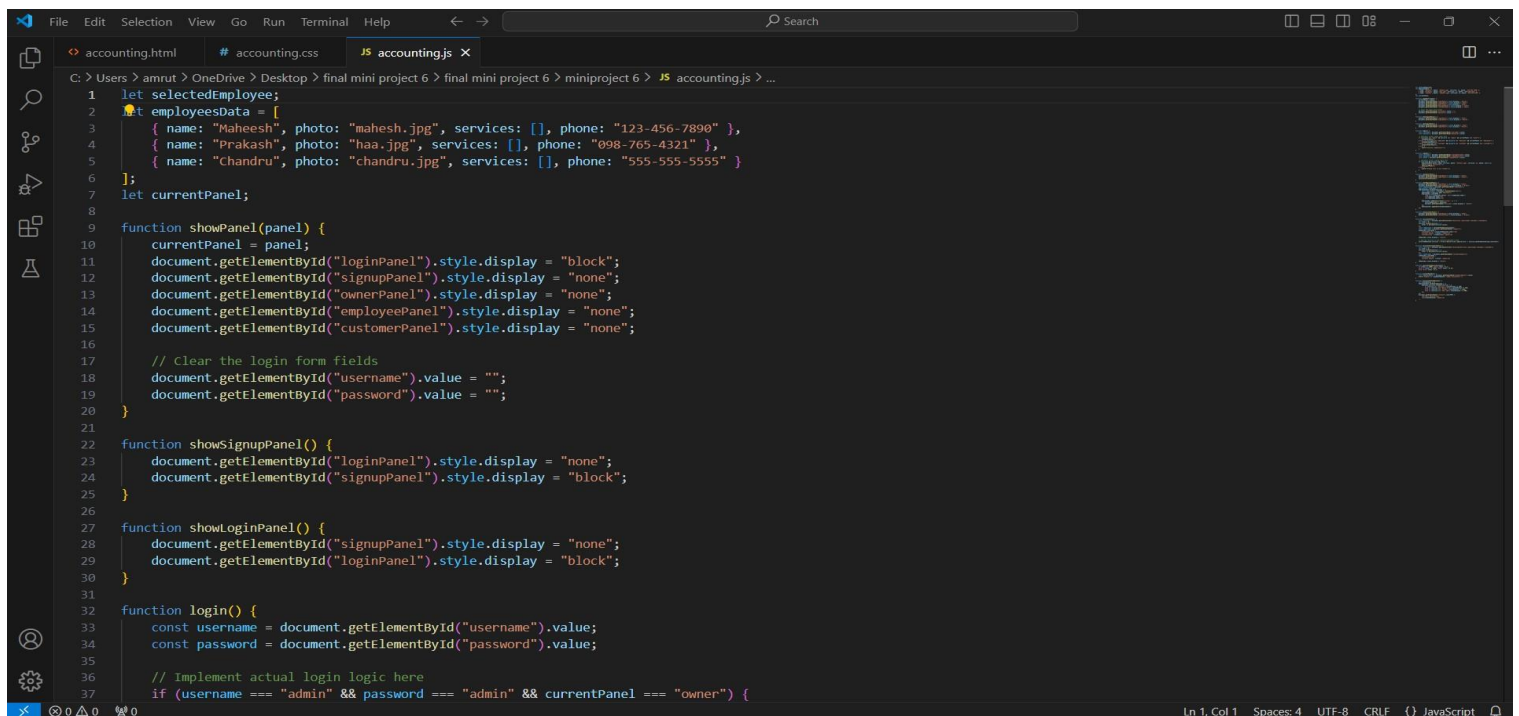


```
38   <h2>Employee Panel</h2>
39   <div id="employees"></div>
40   <div id="services" class="services-container" style="display: none;">
41     <h3>Select Services</h3>
42     <div class="service-item">
43       <input type="checkbox" id="haircut" value="100">
44       <label for="haircut">Haircut</label>
45     </div>
46     <div class="service-item">
47       <input type="checkbox" id="beardTrim" value="100">
48       <label for="beardTrim">Beard Trim</label>
49     </div>
50     <div class="service-item">
51       <input type="checkbox" id="facial" value="1200">
52       <label for="facial">Facial</label>
53     </div>
54     <div class="service-item">
55       <input type="checkbox" id="hairSpa" value="1500">
56       <label for="hairSpa">Hair Spa</label>
57     </div>
58     <button onclick="calculateTotal()">Done</button>
59   </div>
60   <div id="summary" style="display: none;"></div>
61 </div>
62 <div id="customerPanel" style="display: none;">
63   <h2>Customer Panel</h2>
64   <div id="customerServices" class="services-container">
65     <h3>Select Services</h3>
66     <div class="service-item">
67       <input type="checkbox" id="customerHaircut" value="100">
68       <label for="customerHaircut">Haircut</label>
```



The screenshot shows the Visual Studio Code editor with the file `accounting.css` open. The code defines styles for the `body`, `.container`, `h1, h2, h3`, `.options`, and `.options button` elements. The `body` style includes a background image of `back.jpeg.jpg` and a color of `#333`. The `.container` style includes a max-width of `800px`, a margin of `50px auto`, a padding of `20px`, a background color of `rgba(255, 255, 255, 0.8)`, a border radius of `5px`, and a box shadow of `0 0 10px rgba(0, 0, 0, 0.1)`. The `h1, h2, h3` style includes `text-align: center`. The `.options` style includes `text-align: center` and `margin-bottom: 20px`. The `.options button` style includes a padding of `10px 20px`, a margin of `0 10px`, a border of `none`, a border radius of `5px`, a background color of `007bff`, a color of `fff`, and a cursor of `pointer`.

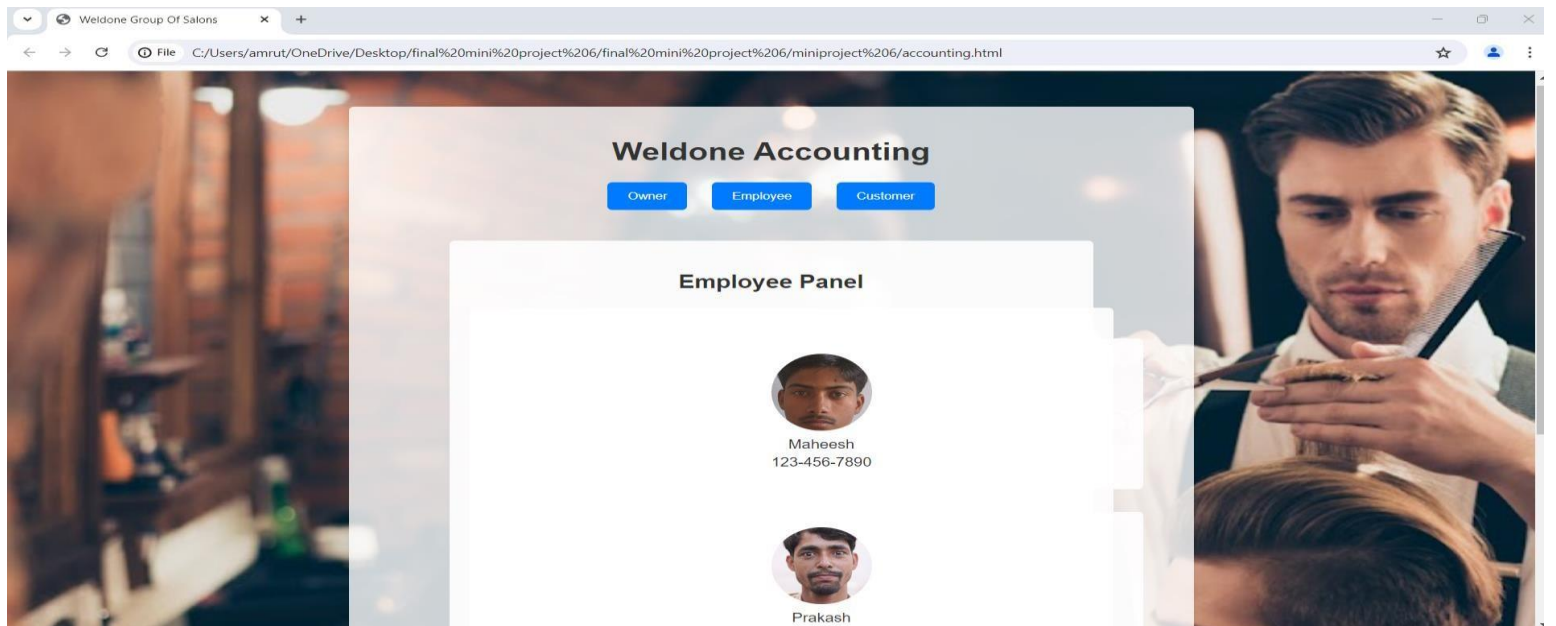
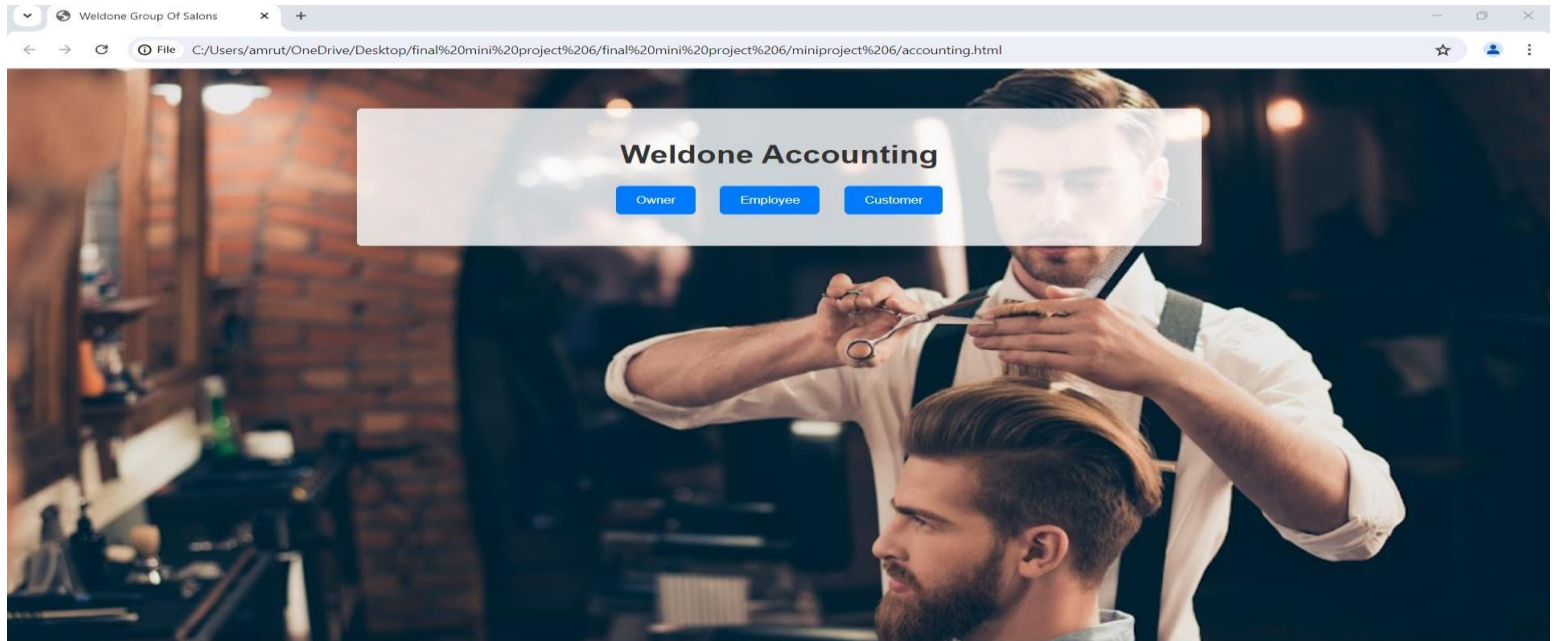
```
1 body {
2     font-family: Arial, sans-serif;
3     margin: 0;
4     padding: 0;
5     background-image: url('back.jpeg.jpg');
6     background-size: cover;
7     background-repeat: no-repeat;
8     color: #333;
9 }
10
11 .container {
12     max-width: 800px;
13     margin: 50px auto;
14     padding: 20px;
15     background-color: rgba(255, 255, 255, 0.8);
16     border-radius: 5px;
17     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
18 }
19
20 h1, h2, h3 {
21     text-align: center;
22 }
23
24 .options {
25     text-align: center;
26     margin-bottom: 20px;
27 }
28
29 .options button {
30     padding: 10px 20px;
31     margin: 0 10px;
32     border: none;
33     border-radius: 5px;
34     background-color: 007bff;
35     color: fff;
36     cursor: pointer;
37 }
```

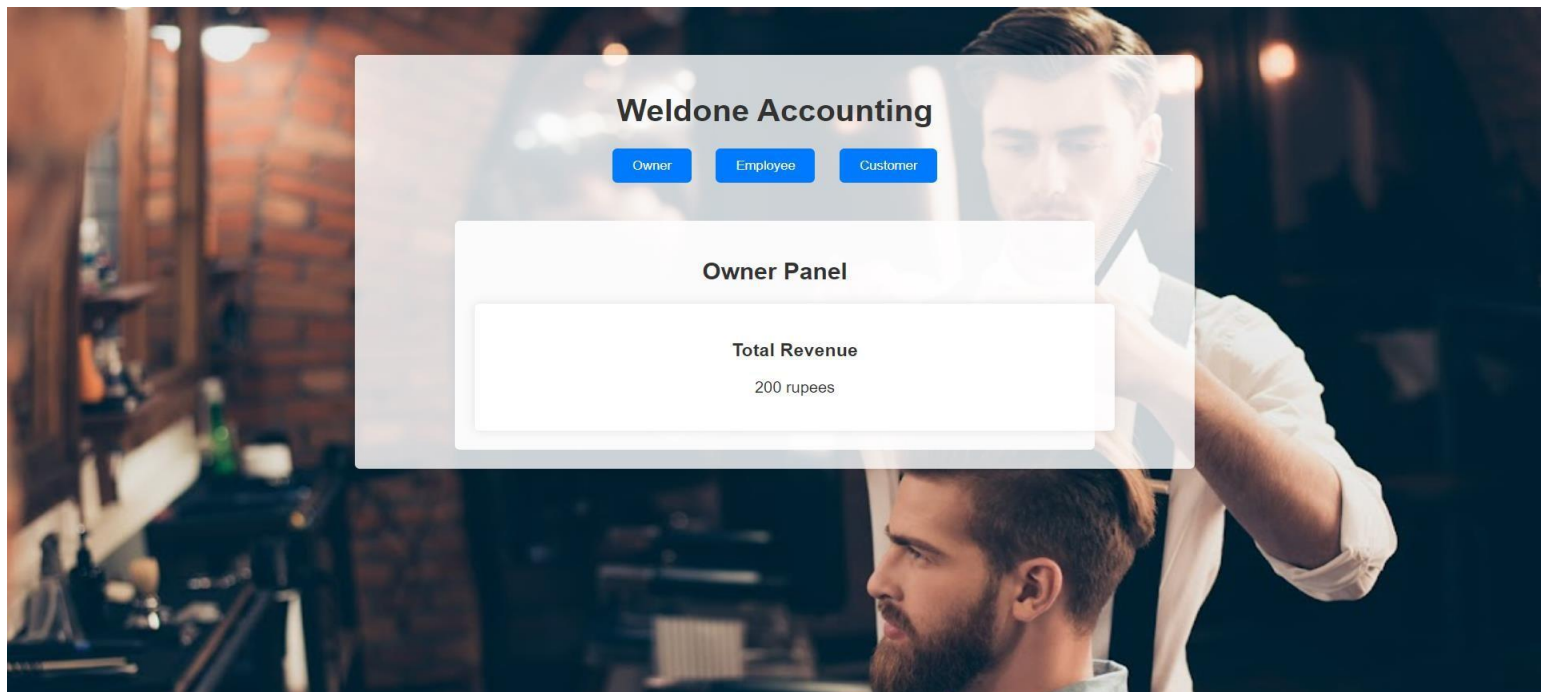
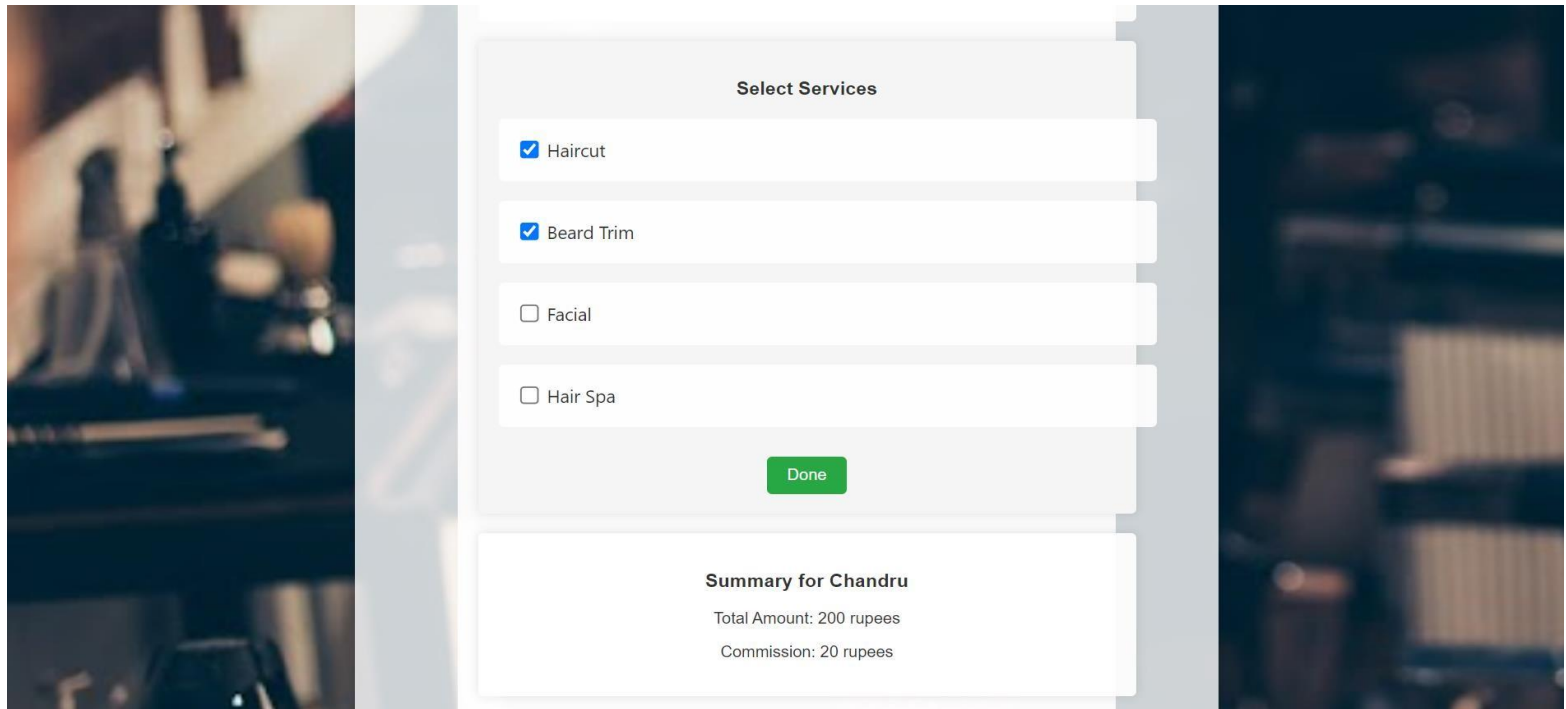


The screenshot shows the Visual Studio Code editor with the file `accounting.js` open. The code defines a `selectedEmployee` variable, an `employeesData` array, and functions for `showPanel`, `showSignupPanel`, `showLoginPanel`, and `login`. The `employeesData` array contains three objects representing employees: `Maheesh`, `Prakash`, and `Chandru`. The `showPanel` function shows the `loginPanel` and hides the others. The `showSignupPanel` function shows the `signupPanel` and hides the `loginPanel`. The `showLoginPanel` function shows the `loginPanel` and hides the `signupPanel`. The `login` function gets the `username` and `password` values and implements the login logic.

```
1 let selectedEmployee;
2 let employeesData = [
3     { name: "Maheesh", photo: "mahesh.jpg", services: [], phone: "123-456-7890" },
4     { name: "Prakash", photo: "haa.jpg", services: [], phone: "098-765-4321" },
5     { name: "Chandru", photo: "chandru.jpg", services: [], phone: "555-555-5555" }
6 ];
7 let currentPanel;
8
9 function showPanel(panel) {
10     currentPanel = panel;
11     document.getElementById("loginPanel").style.display = "block";
12     document.getElementById("signupPanel").style.display = "none";
13     document.getElementById("ownerPanel").style.display = "none";
14     document.getElementById("employeePanel").style.display = "none";
15     document.getElementById("customerPanel").style.display = "none";
16
17     // Clear the login form fields
18     document.getElementById("username").value = "";
19     document.getElementById("password").value = "";
20 }
21
22 function showSignupPanel() {
23     document.getElementById("loginPanel").style.display = "none";
24     document.getElementById("signupPanel").style.display = "block";
25 }
26
27 function showLoginPanel() {
28     document.getElementById("signupPanel").style.display = "none";
29     document.getElementById("loginPanel").style.display = "block";
30 }
31
32 function login() {
33     const username = document.getElementById("username").value;
34     const password = document.getElementById("password").value;
35
36     // Implement actual login logic here
37     if (username === "admin" && password === "admin" && currentPanel === "owner") {
```


Output:





Conclusion:

The accounting management system for the Weldone Group of Salons effectively addressed the need for a streamlined, automated solution to manage accounting and service records. By implementing distinct functionalities for owners, employees, and customers, the system enhanced accuracy and efficiency in service selection, cost calculation, commission determination, and revenue tracking. The user-friendly interface and interactive features simplified operations, reducing manual errors and improving overall productivity. This project demonstrated the successful application of web technologies to solve real-world business challenges, providing a robust and scalable solution for salon management.