A
PROJECT REPORT ON

Online Hair Salon System reduce waiting time

By

SHAH YASH (CE-144) (19CEUEG051)
VAGHANI YASH (CE-170) (19CEUOS078)
SAKHIYA VIDUR (CE-136) (19CEUOG101)

B.Tech CE Semester-VI
Subject: System Design Practice

Guided by:
Prof.Pandav K. Patel
Assistant Professor
Dept. of Comp. Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University

CERTIFICATE

This is to certify that the practical / term work carried out in the subject of

System Design Practice and recorded in this journal is the bonafide

work of

SHAH YASH (CE-144) (19CEUEG051)
VAGHANI YASH (CE-170) (19CEUOS078)
SAKHIYA VIDUR (CE-136) (19CEUOG101)

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2021-2022**.

Prof. Pandav K. Patel
Assistant Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia,
Head,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Contents

Abstract

There is lot of Platform for the restaurant industry, but nothing quite like this existed to bring clients and beauty salons together online hair salon system fills this void in a way that is on-demand, easy to use, and effective for users and salon managers.

Nowadays time is more costly than money so waiting for hours and hours to get hair cutting is not affordable. That's why our application reduces waiting time by showing number in queue and approximate time to customer. Customer can find best salon as per rating of salon and also give rating to salon for improve salon credit.

Owners can improve number of customers by publishing salon in our application. Owner can easily manage customers, stylist and services.

Introduction

- **About Project:**

Online hair salon system (reduced waiting time) is a web-based salon management application with booking scheduling functionality. It connects clients, salons, and stylists in an online community allowing users to browse salons and stylists, and book appointment. Users can also give and read rating of salons. Salon Owners can specify the stylists that work at their salons and the services provided by salon.

Our application includes Some of the major use cases user account registration, login/logout, List of nearby salon range of 50km, booking of seat, view Orders history, rate completed orders at customer side. Queueing customer, manage stylists, services and salon details at Owner side. Inspection and verification of salon at admin side.

- **Technology:**

My project uses MongoDB Atlas and Express-js to back the interface with strong database functionality and React Framework as frontend. Online hair salon system integrates Cloudinary as a cloud storage for Dynamic image upload, google map picker for view map and Nodemailer for mailing.This project will target the major web browsers as the initial platform.

- **Tools:**
  - Visual Studio Code (editor)
  - MongoDb compass
  - Github

SOFTWARE REQUIREMENTS SPECIFICATION (SRS):

Online Hair Salon System (reduce wait time)

R 1: User side

Description: User can see only nearby salon list if user want to see full details and wants to book seat then user must sign in to system.

R 1.1: Authentication

Description: If user is new to the system then First he/she must sign up to the system otherwise user can directly login and then enter to the system.

R 1.1.1: Sign Up

Input: User details.
Process: Validate details.
Output: Redirect to OTP verification page.

R 1.1.2: Sign In

Input: User details.
Process: Validate details.
Output: Redirect to Home page.

R 1.1.3: Log Out

Input: User Selection.
Output: Redirect to Home page.

R 1.2: Salon Details

Description: as per user selection system shows salon details view

R 1.2.1: Hair Services

Input: User selection.
Output: list the all-hair services.

R 1.2.2: About

Input: User selection.
Output: Show Salon and Owner details.

R 1.2.3: Employee Details

Input: User selection.
Output: Show Employee Details of salon.

R 1.2.4: Book Seat

Input: User Selection.
Process Set: Show summary Page
Process: User Confirmation.
Output: Mail Sent to user email, Confirmation message show.

R 1.3: Orders History
Description: it will show list and status of all orders.

R 1.3.1: Rating
Description: Only Completed Orders are Rated by User.

Input: User selection.
Process: add rated star to salon.
Output: Confirmation Message.

R 1.4: List Near By Salon
Description: Display all salon near by user's current location
Input: User's location.
Process: Find distance user's current location to salon's location.
Output: List of nearby salons.

R 2: Owner Side
Description: Owner First have to sign up to system and then register for Salon, admin review the salon and then after approval by admin Owner enter in to Owner Home page.

R 2.1: Authentication
Description: If user is new to the system then First he/she must sign up to the system otherwise user can directly login and then enter to the system.

R 2.1.1: Sign Up
Input: User details.
Process: Validate details.
Output: Redirect to Salon Register.

R 2.1.2: Sign In

Input: User details.
Process: Validate details.
Output:

1) if salon not registered then redirect to salon registration.
2) if salon registered and under verification then redirect to verification status.
3) otherwise redirect to owner home page.

R 2.1.3: Register Salon

Input: User details.
Process: Validate details.
Output: Redirect to Verification status Page.

R 2.1.4: Log Out

Input: User Selection.
Output: Redirect to Home page.

R 2.2: Manage Customer

Description: it shows list of customers in queue.

R 2.2.1 Complete Order

Input: User selection.
Output: Customer Remove from list.

R 2.2.2 Cancel Order

Input: User selection.
Output: Customer Remove from list.

R 2.3: Manage Employee

Description: it shows list of Employee.

R 2.3.1 Add Employee

Input: Detail.
Process: it checks the capacity of salon.
Output: Employee Added to list.

R 2.3.2 Edit Employee

Input: Employee Details.
Output: list updated.

R 2.3.3 Delete Employee

Input: User selection.
Output: Employee removed from list.

R 2.4: Manage Hair Services

Description: it shows list of Hair Services as per category.

R 2.4.1 Add Service

Input: Details.
Output: Service Added to list.

R 2.4.2 Edit Service

Input: Details.
Output: list updated.

R 2.4.3 Delete Employee

Input: User selection.
Output: Service removed from list.

R 3: Admin Side

Description: Admin First have to sign into system then he/she will be redirect to page which shows the list of requested salon registration.

R 3.1: Salon Details

Input: User selection.
Output: show Details of salon.

R 3.2: Approve Salon Registration

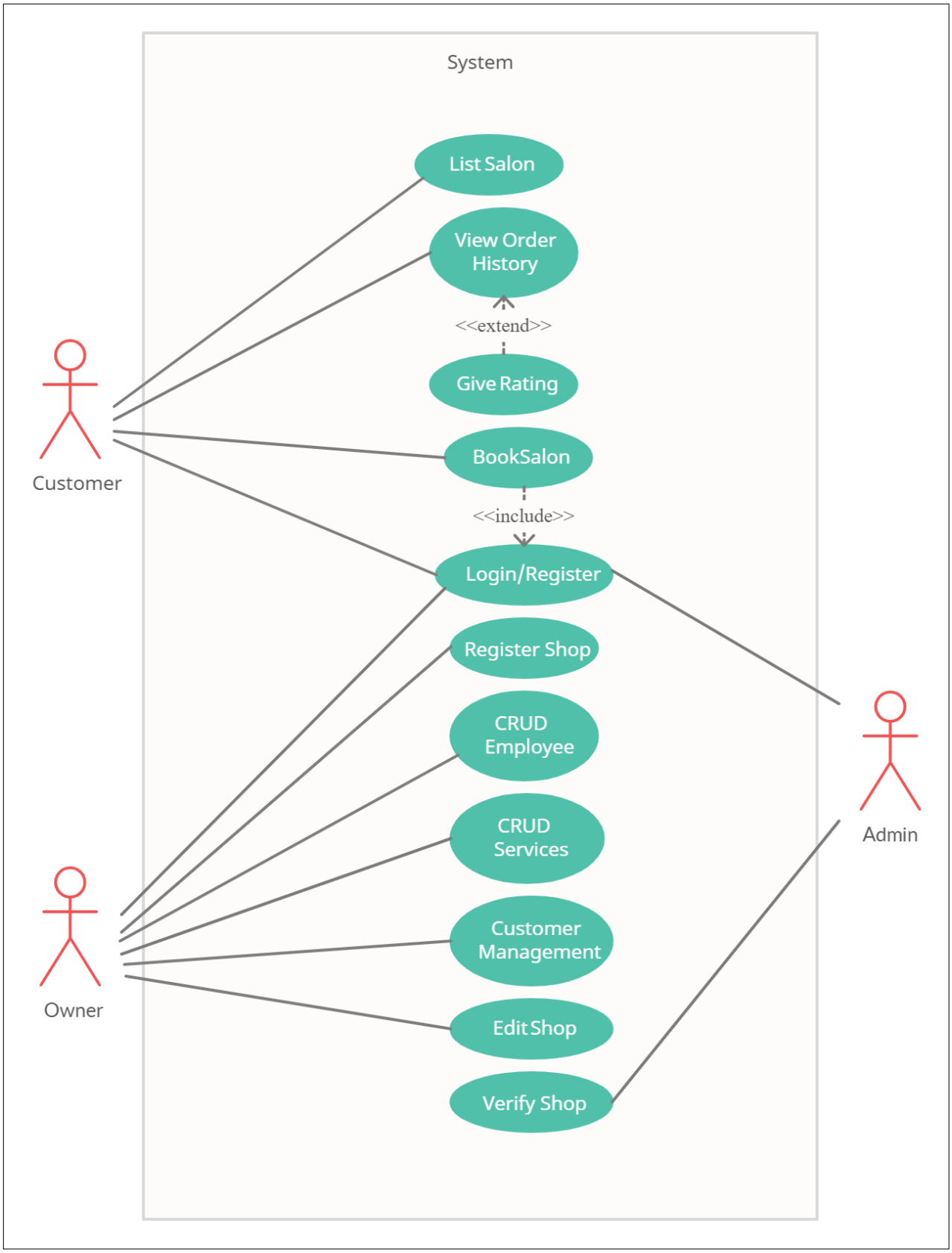
Input: User selection.
Process: Salon Registration is approved.
Output: salon removed from list.

R 3.3: Reject Salon Registration

Input: User selection.
Process: Salon Registration is Rejected.
Output: salon removed from list.

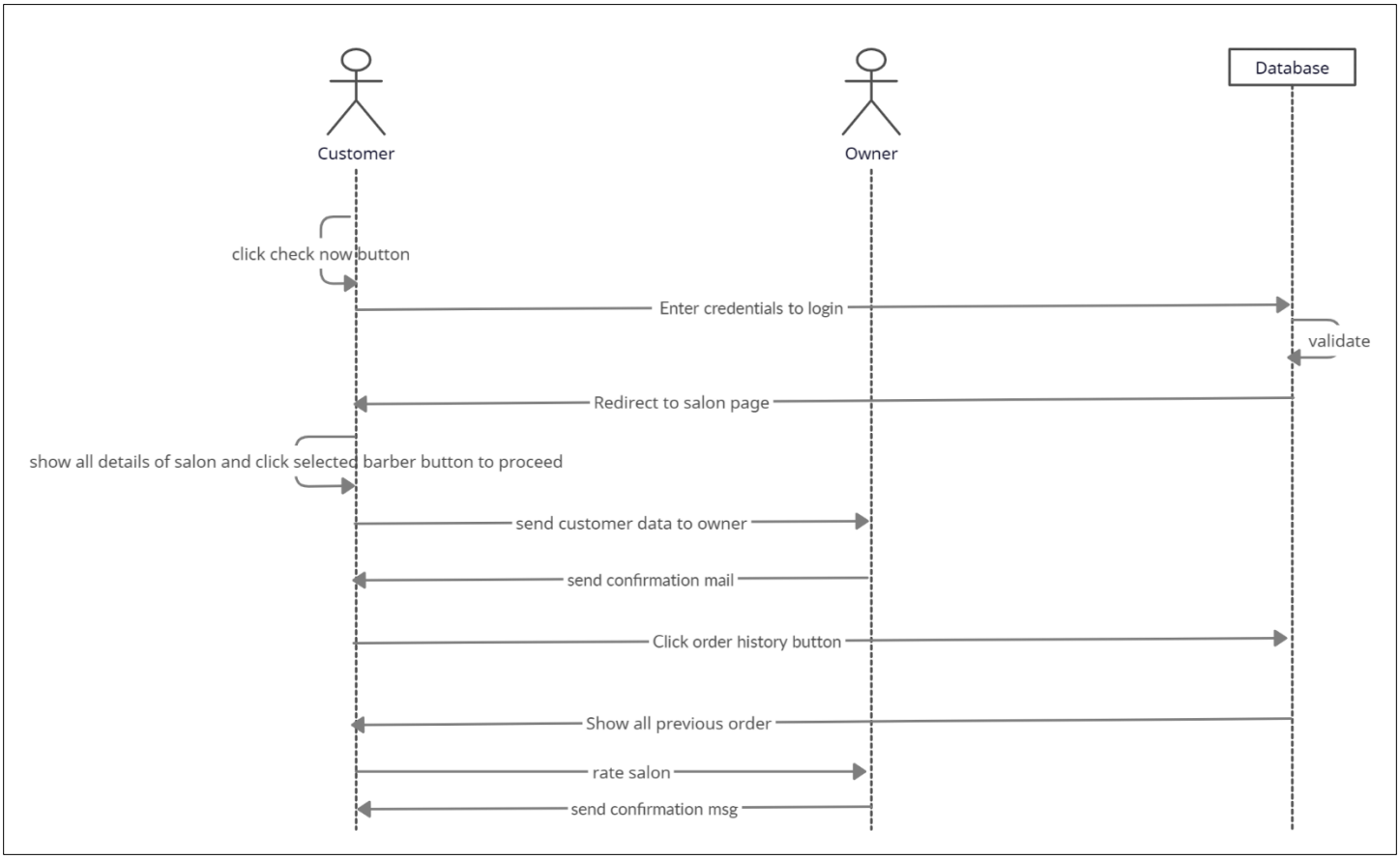
DESIGN

Use Case Diagram

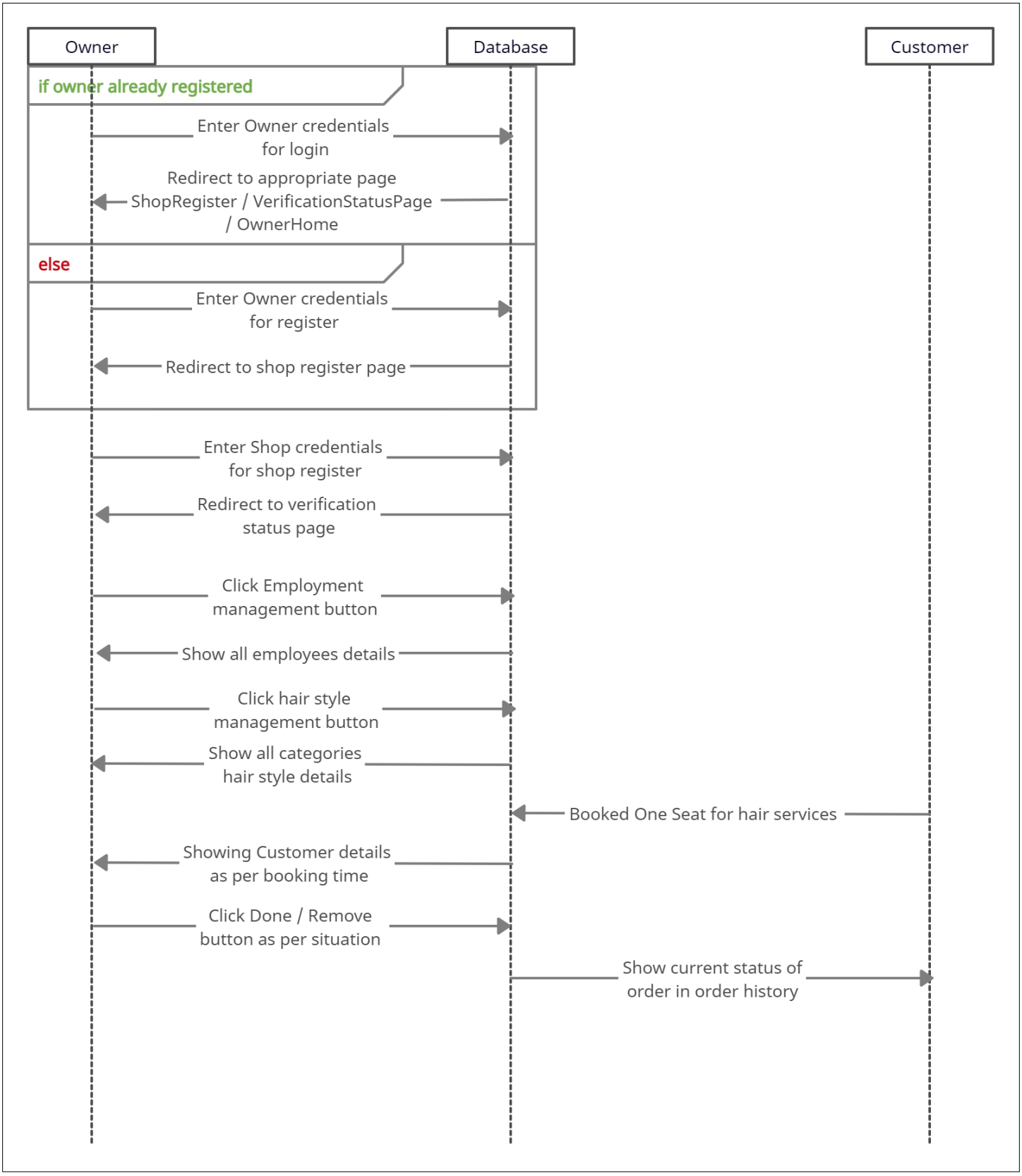


Sequence Diagram

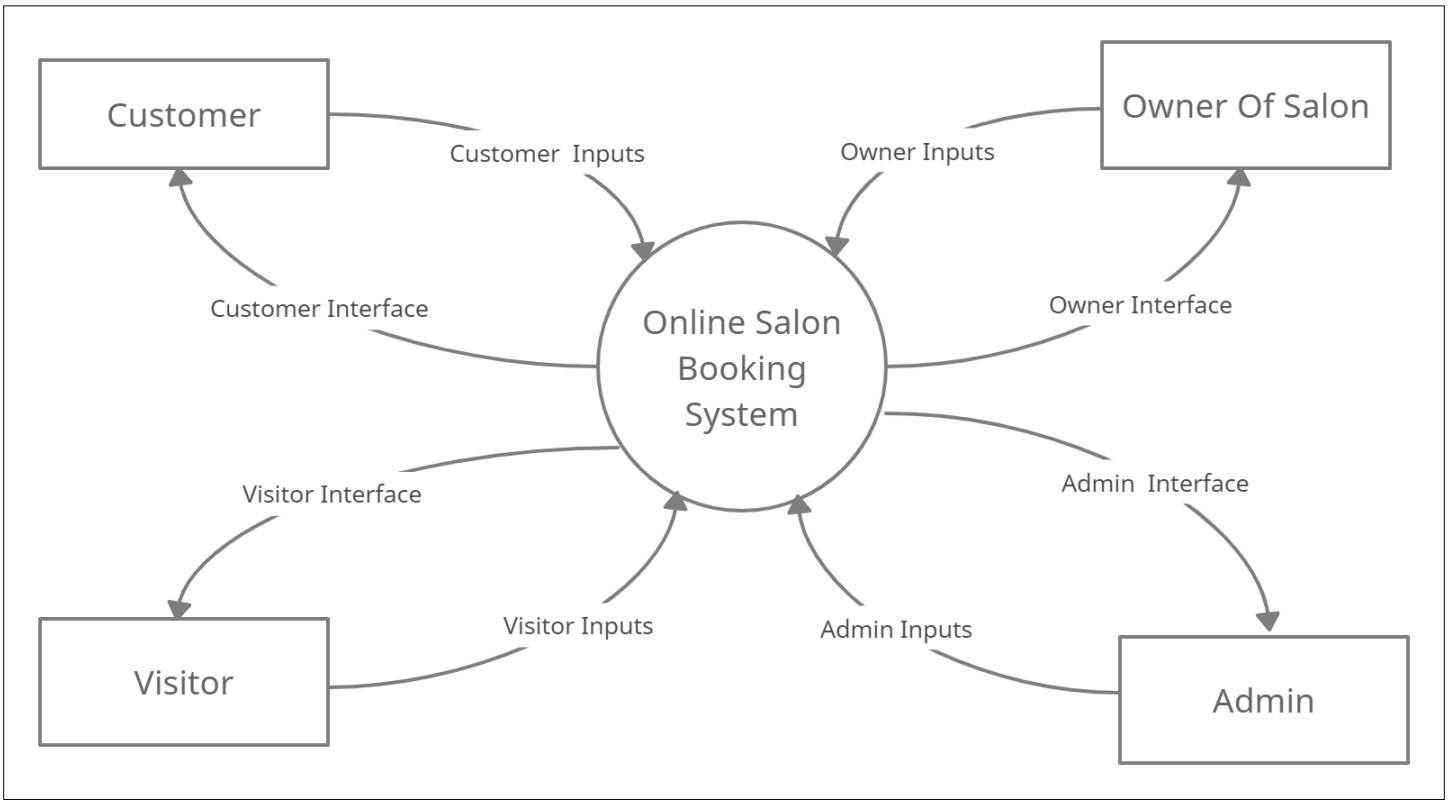
1) Seat Booking



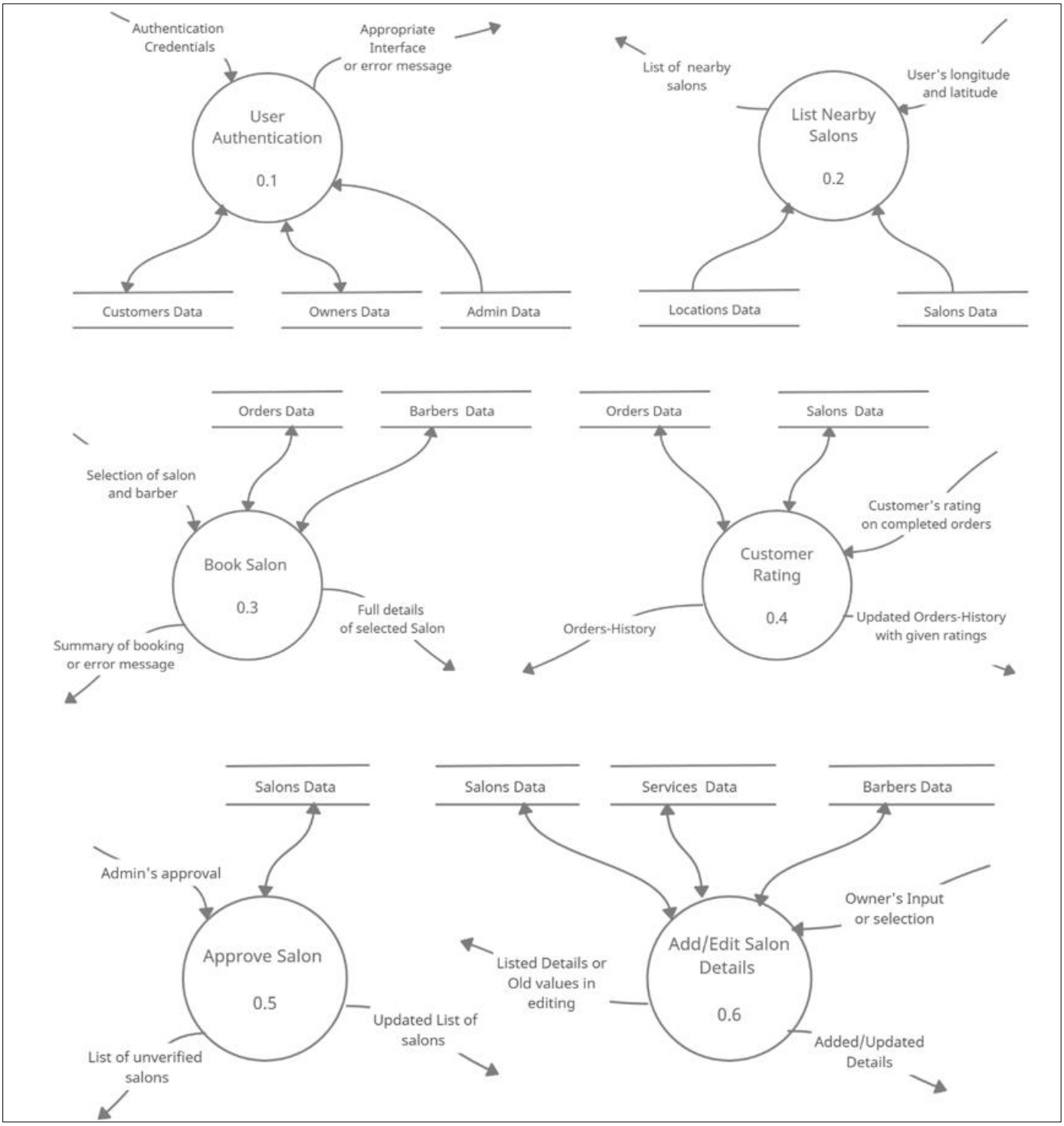
2) Sequence Of Owner Interface



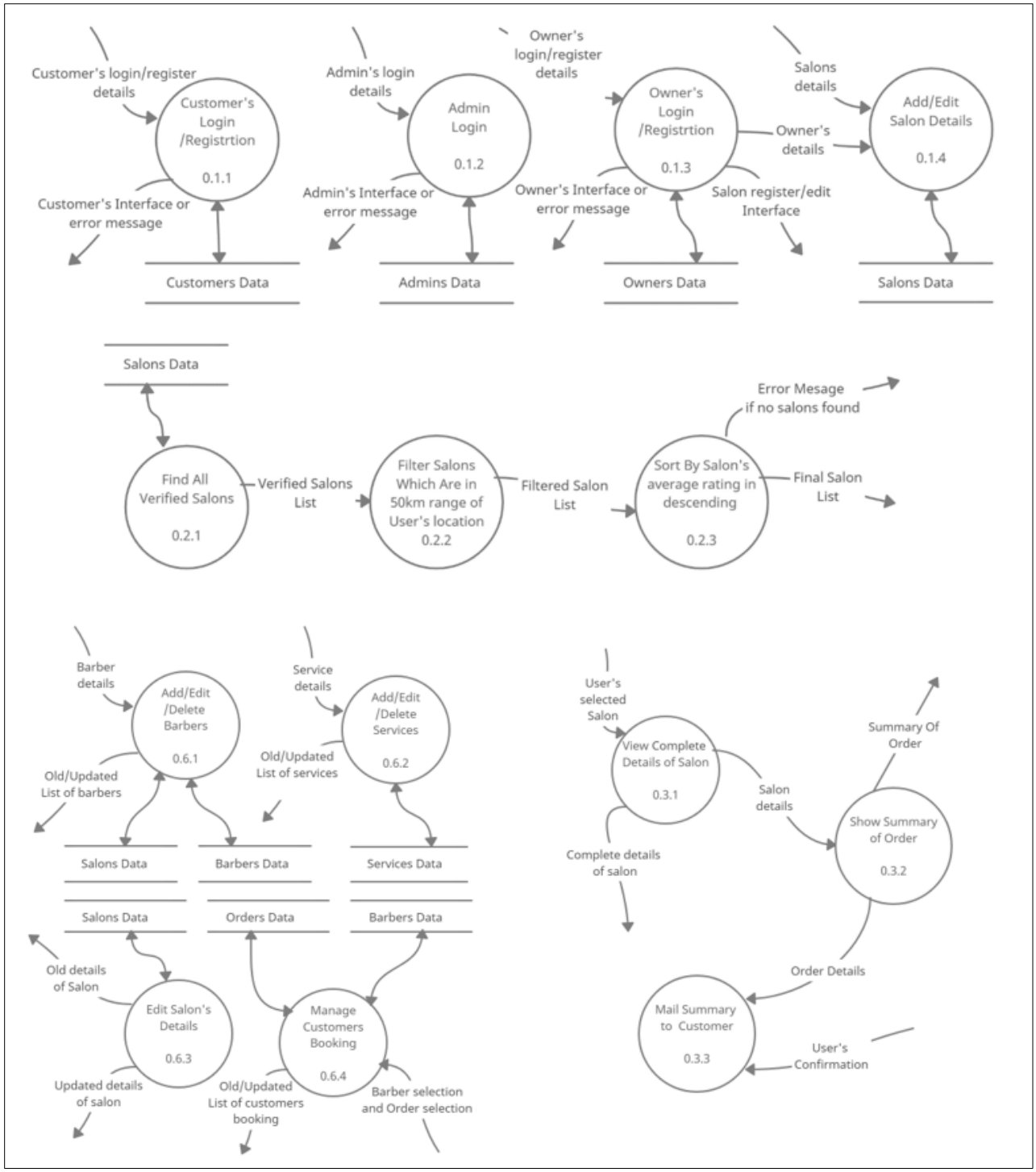
Data-Flow Diagram



Level - 0

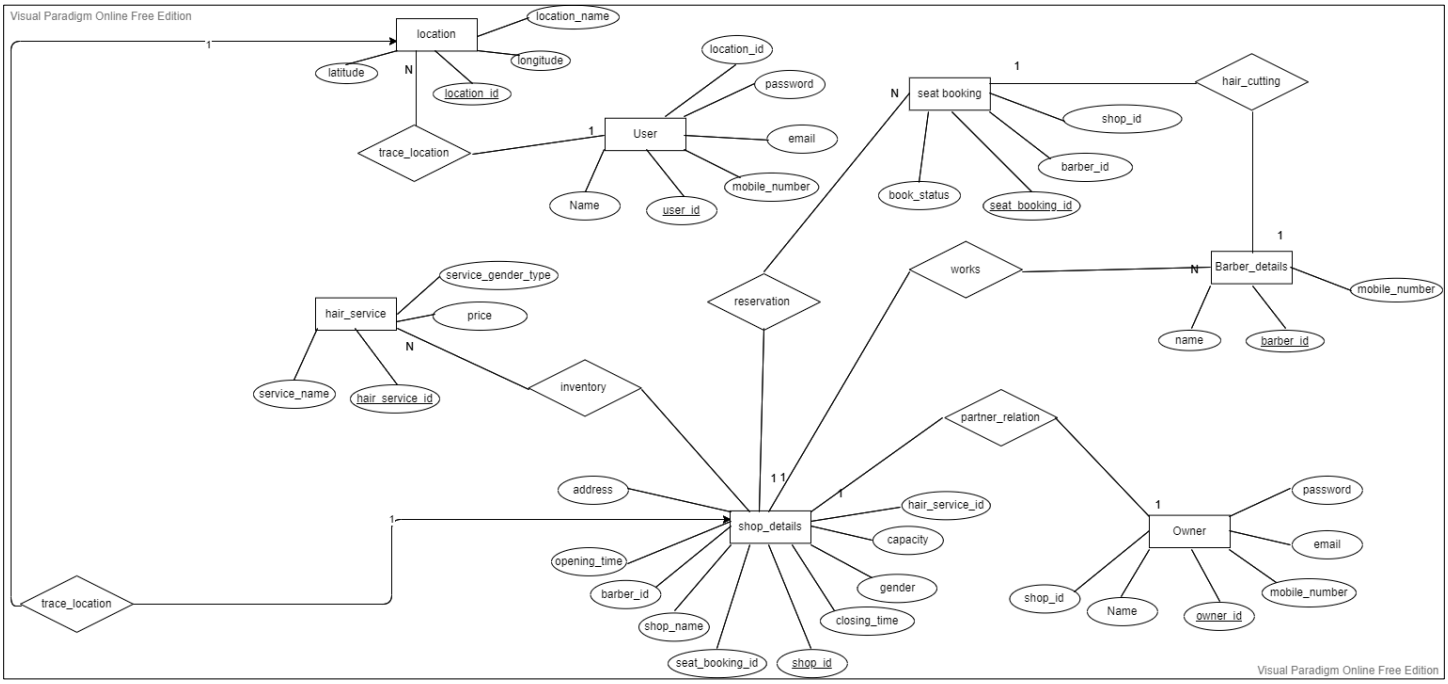

Authentication Credentials
Appropriate Interface or error message
User Authentication
0.1
Customers Data
Owners Data
Admin Data
List of nearby salons
User's longitude and latitude
List Nearby Salons
0.2
Locations Data
Salons Data
Orders Data
Barbers Data
Selection of salon and barber
Book Salon
0.3
Full details of selected Salon
Summary of booking or error message
Orders Data
Salons Data
Customer's rating on completed orders
Customer Rating
0.4
Orders-History
Updated Orders-History with given ratings
Salons Data
Salons Data
Services Data
Barbers Data
Admin's approval
Approve Salon
0.5
Updated List of salons
List of unverified salons
Owner's Input or selection
Add/Edit Salon Details
0.6
Listed Details or Old values in editing
Added/Updated Details

Level – 1


Customer's login/register details
Customer's Login /Registrtion
0.1.1
Customer's Interface or error message
Customers Data
Admin's login details
Admin Login
0.1.2
Admin's Interface or error message
Admins Data
Owner's login/register details
Owner's Login /Registrtion
0.1.3
Owner's Interface or error message
Owners Data
Salons details
Owner's details
Add/Edit Salon Details
0.1.4
Salon register/edit Interface
Salons Data
Salons Data
Find All Verified Salons
0.2.1
Verified Salons List
Filter Salons Which Are in 50km range of User's location
0.2.2
Filtered Salon List
Sort By Salon's average rating in descending
0.2.3
Error Mesage if no salons found
Final Salon List
Barber details
Add/Edit /Delete Barbers
0.6.1
Old/Updated List of barbers
Salons Data
Barbers Data
Service details
Add/Edit /Delete Services
0.6.2
Old/Updated List of services
Services Data
Salons Data
Orders Data
Barbers Data
Old details of Salon
Edit Salon's Details
0.6.3
Updated details of salon
Manage Customers Booking
0.6.4
Old/Updated List of customers booking
Barber selection and Order selection
User's selected Salon
View Complete Details of Salon
0.3.1
Complete details of salon
Salon details
Show Summary of Order
0.3.2
Summary Of Order
Order Details
Mail Summary to Customer
0.3.3
User's Confirmation

Level - 2

ER Diagram


Visual Paradigm Online Free Edition
location
location_name
latitude
longitude
location_id
trace_location
User
location_id
password
email
mobile_number
Name
user_id
seat booking
shop_id
barber_id
book_status
seat_booking_id
hair_cutting
Barber_details
mobile_number
name
barber_id
works
reservation
hair_service
service_gender_type
price
service_name
hair_service_id
inventory
partner_relation
Owner
password
email
mobile_number
shop_id
Name
owner_id
shop_details
address
hair_service_id
capacity
opening_time
barber_id
shop_name
seat_booking_id
shop_id
gender
closing_time
trace_location
Visual Paradigm Online Free Edition

Data Dictionary

Tables

Collections

CREATE COLLECTION

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
admins	1	74.0 B	74.0 B	2	41.0 KB	
barbers	25	139.0 B	3.5 KB	2	73.7 KB	
customerorders	45	148.0 B	6.7 KB	1	36.9 KB	
customers	7	126.4 B	885.0 B	2	73.7 KB	
hairservices	36	110.3 B	4.0 KB	1	36.9 KB	
locations	39	68.0 B	2.7 KB	1	36.9 KB	
owners	28	133.9 B	3.7 KB	2	73.7 KB	
shops	20	447.8 B	9.0 KB	1	36.9 KB	

Admin:

```
const adminSchema = new mongoose.Schema(
    {
        email: { type: String, required: true, unique: true },
        password: { type:String, required: true }
    }
)
```

Barber:

```
const barberSchema = new mongoose.Schema(
    {
        name: { type: String, required: true },
        mobile_num:{type: Number, required: true},
        email: { type: String, required: true, unique: true },
        customer_turn_number : {type:Number,default:0},
        customer_ids :[{type: mongoose.Schema.Types.ObjectId,ref:Customer}]
    }
)
```

Customer:

```
const customerSchema = new mongoose.Schema(
    {
        name: { type: String, required: true },
        mobile_num:{type: Number, required: true},
        email: { type: String, required: true, unique: true },
        password: { type:String, required: true }
    }
)
```

CustomerOrders:

```
const CustomerOrderSchema = new mongoose.Schema(
    {
        customer_id :  {type: mongoose.Schema.Types.ObjectId, ref: Customer},
        shop_id : {type: mongoose.Schema.Types.ObjectId, ref: Shop},
        barber_id : {type: mongoose.Schema.Types.ObjectId, ref: Barber},
        status : {type: String, default: "waiting"},
        rating : {type:Number,default:0},
        date:{type:Date,required:true}
    }
)
```

HairService:

```
const hairServiceSchema = new mongoose.Schema(
    {
        service_name: { type: String, required: true },
        price:{type: Number, required: true},
        gender_type: { type: Boolean, required: true },//t : male,f:female
        category : {type:String , required:true},
    }
)
```

Location:

```
const locationSchema = new mongoose.Schema(
    {
        longitude :{type: Number, required: true },
        latitude:{type: Number, required: true }
    }
)
```

Owner:

```
const ownerSchema = new mongoose.Schema(
    {
        name: { type: String, required: true },
        mobile_num:{type: Number, required: true},
        email: { type: String, required: true, unique: true },
        password: { type:String, required: true },
        shopRegisterFlag:{type:Boolean,required:true , default:false}
    }
)
```

Shop(salon):

```
const shopSchema = new mongoose.Schema(
    {
        shop_name: { type: String, required: true },
        address: { type: String, required: true },
        opening_time:{ type: String, required: true },
        closing_time:{ type: String, required: true },
        salon_gender_type : { type: String, required: true } ,
        capacity_seats : { type: Number, required: true },
        verified : {type:String,required:true,default:"pending"},
        number_of_rating : {type:Number,default:0},
        avg_rating : {type:Number,default:0},
        location_id:{type: mongoose.Schema.Types.ObjectId,ref:Location},
        barber_ids:[{type: mongoose.Schema.Types.ObjectId,ref:Barber}],
        owner_id : {type: mongoose.Schema.Types.ObjectId, ref: Owner},
        hair_service_id :  [{ type:  mongoose.Schema.Types.ObjectId, ref: Service }],
        images_pub_ids : [] //public ids of images stored on cloudinary

    }
)
```

Implementation Details

- **Modules created and brief description of each module**

  This Project consists of 4 major modules.

  1) User Module
  2) List Nearby Salon Module
  3) Rating Module
  4) Salon Details Management Module

  Each module consists of several methods to implement the required functionality Implementation is done using MERN.

  1) User Module

  There are three types of Users: Admin, Owner, Customer. This module is the base for authentication and authorization to ensure the security aspect of the user.

  2) List Nearby Salon Module

  In this module it finds salons in range of 50km as per user's current location by finding distance between salons and user's location (longitude, latitude) and sort salon list by average rating.

  3) Rating Module

  In this module customer can give rating to their completed orders which is also reflected in salon's average rating.

  4) Salon Details Management Module

  In this module Owner can update salon's details (i.e. address, photos etc..). Owner can manage all details of Customer appointments, services and Employees.

➢ Function prototypes which implement major functionality

1) List salon

```
function distance(lat1, lon1, lat2, lon2, unit) {
    var radlat1 = Math.PI * lat1 / 180
    var radlat2 = Math.PI * lat2 / 180
    var theta = lon1 - lon2
    var radtheta = Math.PI * theta / 180
    var dist = Math.sin(radlat1) * Math.sin(radlat2) +
            Math.cos(radlat1) * Math.cos(radlat2) * Math.cos(radtheta);
    if (dist > 1) {
        dist = 1;
    }
    dist = Math.acos(dist)
    dist = dist * 180 / Math.PI
    dist = dist * 60 * 1.1515
    //K for KM
    if (unit == "K") { dist = dist * 1.609344 }
    return dist;
}
```

```
export const listShops = async (req, res) => {
    try {
        const { lon, lat } = req.body;
        const shops = await Shop.find({verified:"Accept"});
        const shopList = [];

        for(let i=0;i<shops.length;i++){
            const loc = await Location.findById(shops[i].location_id);
            if(loc && distance(lat, lon, loc.latitude,loc.longitude, "K")<50) {
                shopList.push(shops[i]);
            }
        }
        shopList.sort((a, b)=> {return b.avg_rating-a.avg_rating;});
        //generating prefix link for images
        const prefix_link = process.env.BEGIN_LINK + process.env.CLOUDINARY_NAME + process.env.SUB_FOLDER_PATH;

        res.json({ stat: true, shops: shopList,prefix_link:prefix_link, message: "Shop list." });
    }
    catch (err) {
        res.json({ wentWrong: true, message: "Something went wrong !" });
        console.log(err.message);
    }
}
```

2) Rating Module

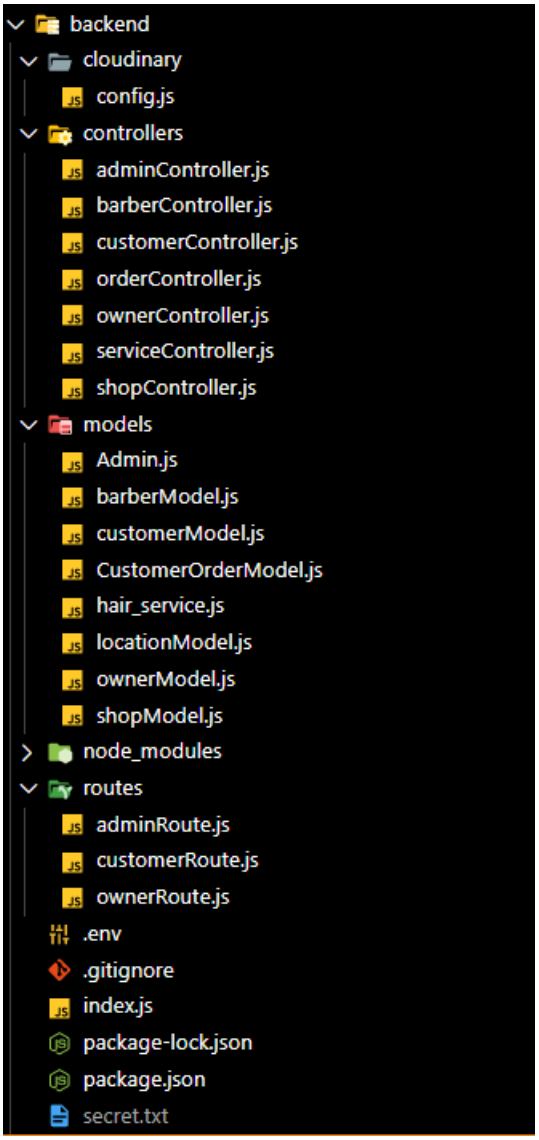
```
export const updateRating = async (req, res) => {
    try {
        const { order_id,new_rating} = req.body;
        console.log(order_id,new_rating)
        const order = await CustomerOrder.findById(order_id);
        console.log(order)

        const shop = await Shop.findById(order.shop_id);
        console.log(shop)

        const avgRating  = shop.avg_rating;
        const NumberOfRating = shop.number_of_rating;
        const newRating = (avgRating*NumberOfRating/(NumberOfRating + 1)) + (new_rating/(NumberOfRating + 1));

        await shop.set({number_of_rating : NumberOfRating + 1,avg_rating : newRating});
        await shop.save();

        await order.set({rating : new_rating});
        await order.save();

        res.json({ stat: true, message: "Thank you for your valuable rating." });
    }
    catch (err) {
        res.json({ wentWrong: true, message: "Something went wrong !" });
        console.log(err.message);
    }
}
```

3)Edit Shop

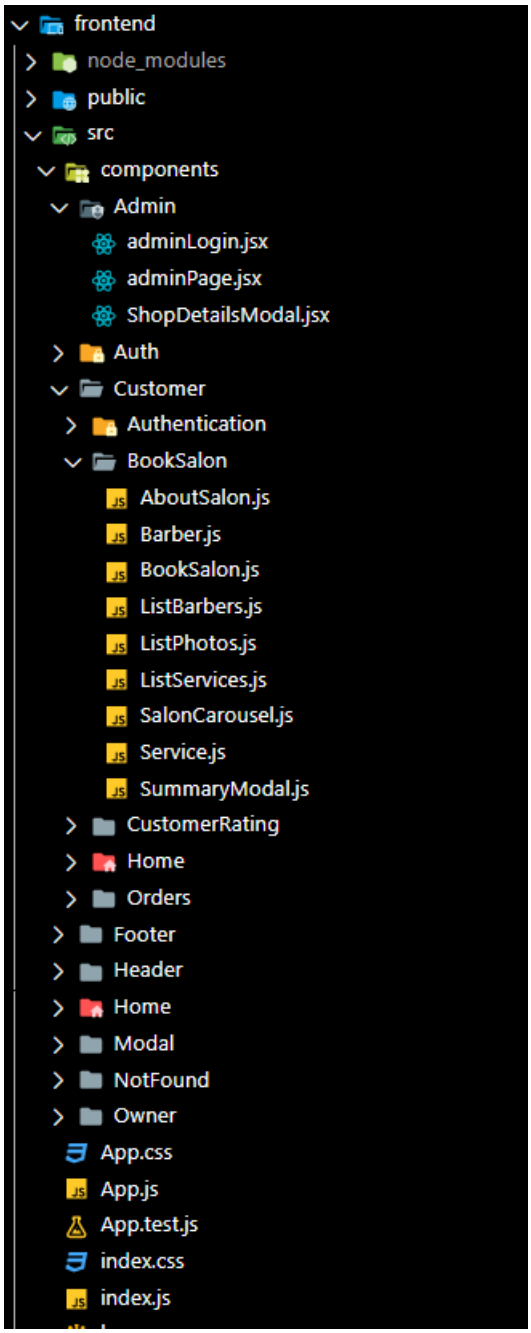
```
export const editShop = async (req, res) => {
    try {
        const {shop_id, shop_name, address, opening_time, closing_time, salon_gender_type,
            capacity_seats, longitude, latitude, loc_id,index_images_to_delete, images_to_add } = req.body;
        const shop = await Shop.findById(shop_id);

        //updating location
        const upd_location_obj = await Location.findByIdAndUpdate(loc_id,{ longitude: longitude, latitude: latitude });

        //deleting selected images
        let images_ids_array = shop.images_pub_ids;

        //sorting in descending
        index_images_to_delete.sort( (a, b)=> {return b-a;});
        for(let i=0; i<index_images_to_delete.length; i++)
        {
            let pub_id = images_ids_array.splice(index_images_to_delete[i],1)[0];
            await cloudinary.uploader.destroy(pub_id);
        }

        // uploading new images
        for(let i=0;i<images_to_add.length; i++)
        {
            const uploadResponse = await cloudinary.uploader.upload(images_to_add[i], {
                upload_preset: 'ml_default',
            });
            images_ids_array.push(uploadResponse.public_id);
        }
        let status= "";
        if(shop.verified=="Reject")
        {
            status="pending";
        }
        else
        {
            status = shop.verified;
        }
        await shop.set({shop_name:shop_name, address:address, opening_time:opening_time, closing_time:closing_time, salon_gender_type:salon_gender_type,
            verified:status, capacity_seats:capacity_seats,images_pub_ids:images_ids_array});
        await shop.save();
        res.json({ stat: true,shop:shop ,message: "Shop details updated successfully!." });
    }
    catch (err) {
        res.json({ wentWrong: true, message: "Something went wrong !" });
        console.log(err.message);
    }
}
export const getlocation = async (req,res)=>{
```

➢ Project Structure

Backend:



Frontend:

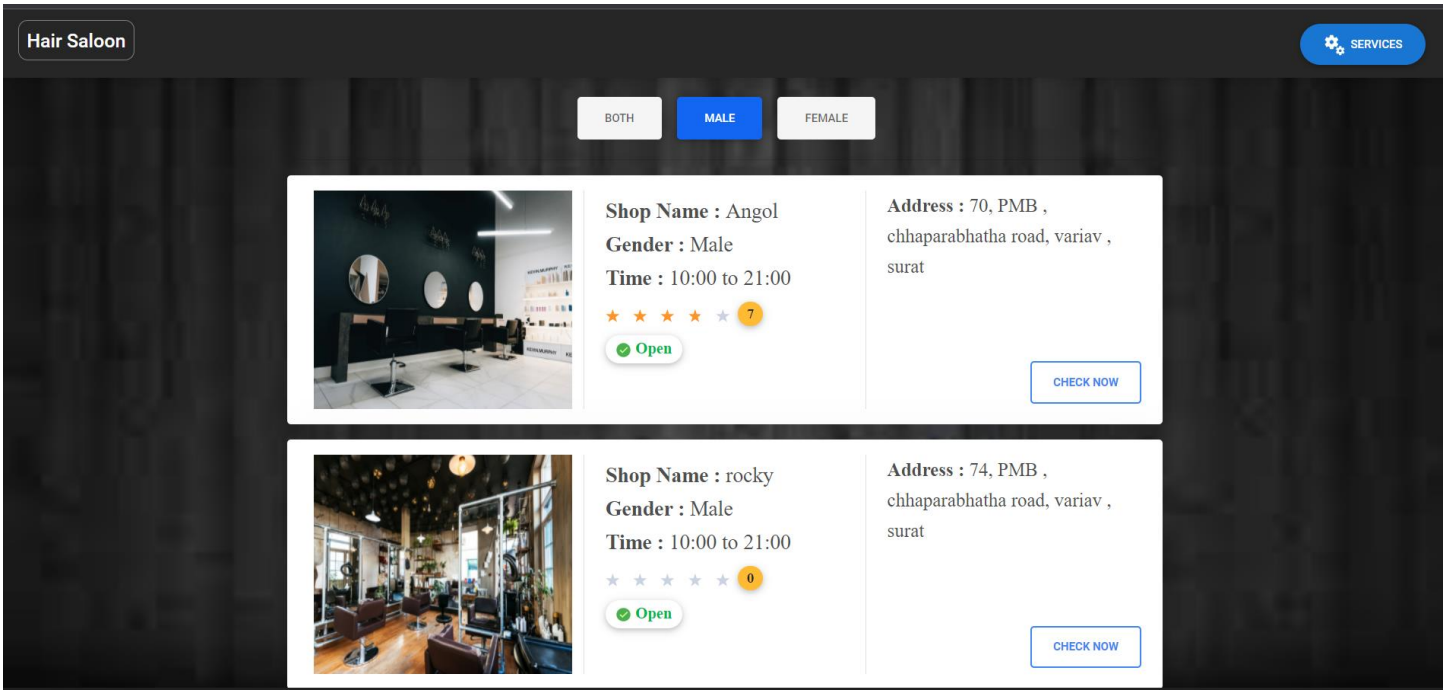


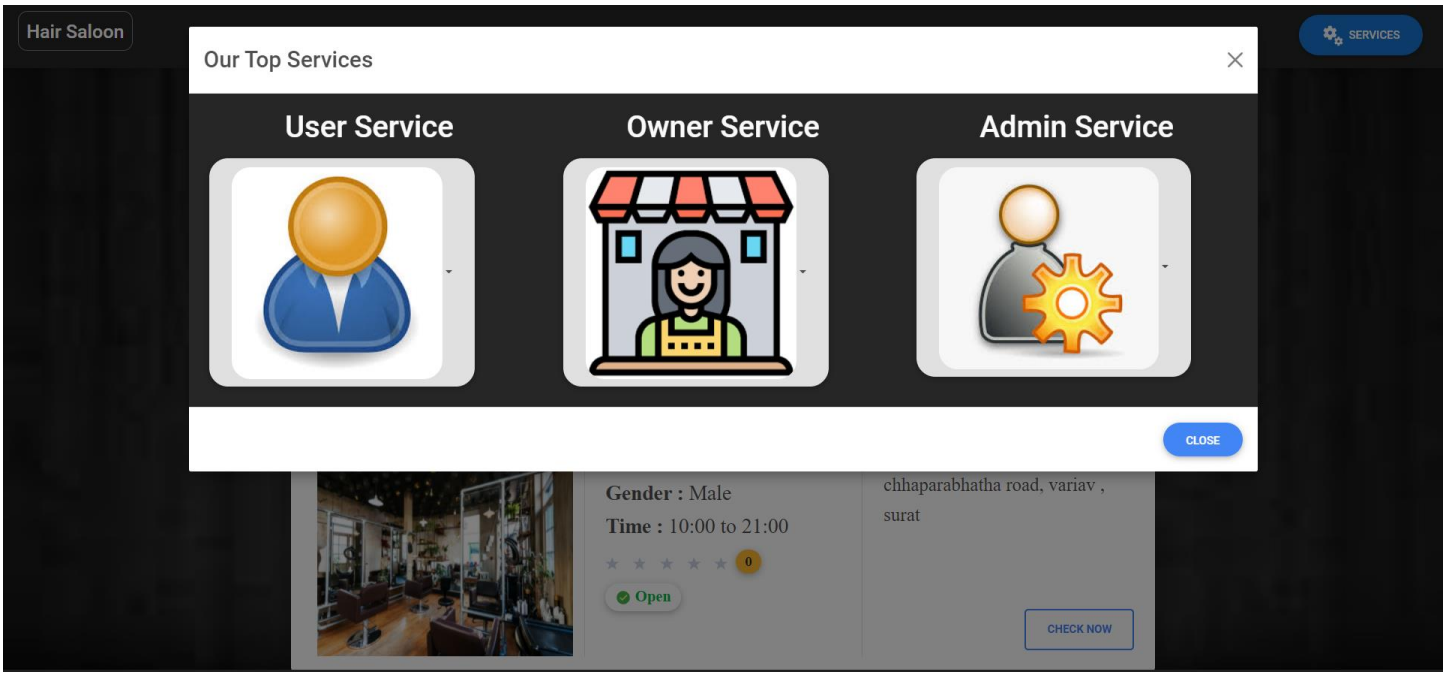
Testing

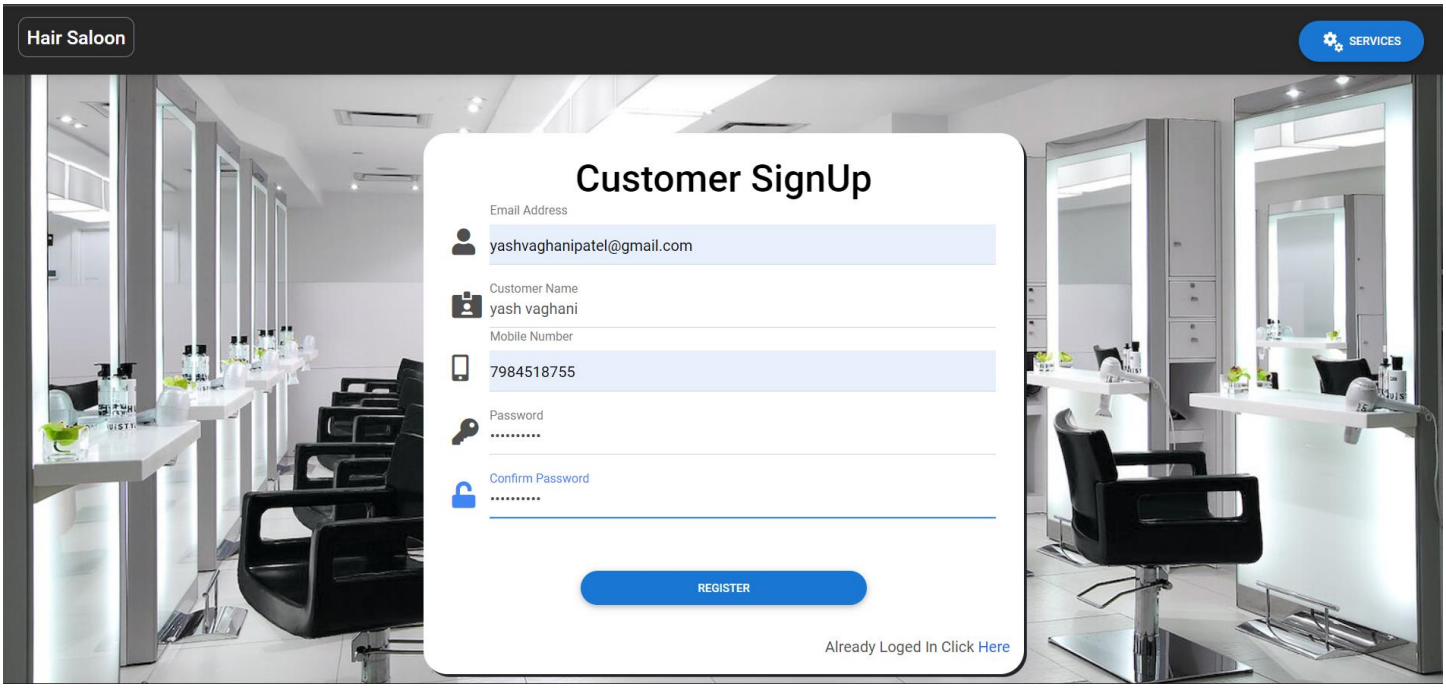
Test Cases

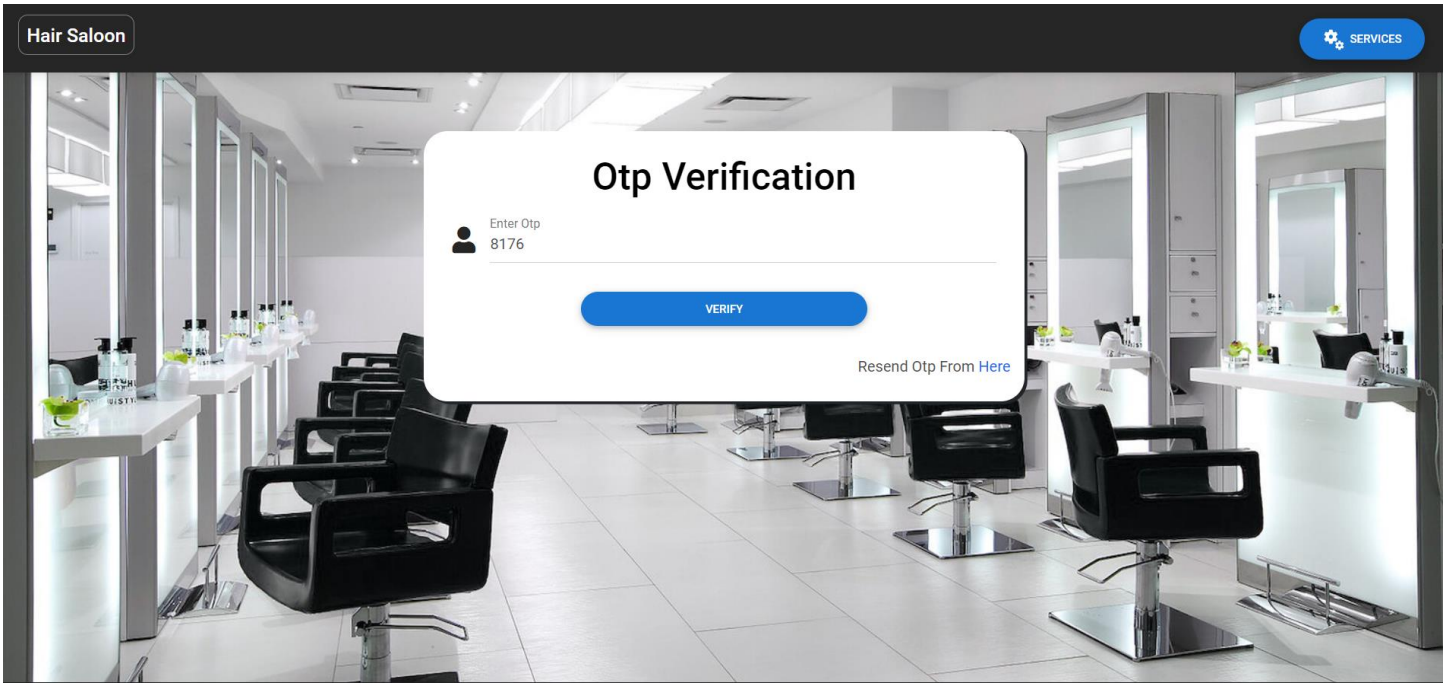
Test Case Id	SRS Id	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC_01	R 1.1 R 2.1	Authentication of customer and owner	Credential	Success or error message	Success or error message	Pass
TC_02	R 1.2	Salon Detail	Selection of salon	Full detail of salon	Full detail of salon	Pass
TC_03	R 1.2.4	Book seat	Stylist selection	Summary mail and message	Summary mail and message	Pass
TC_04	R 1.3.1	Rating	Customer's rating	Confirmation message	Confirmation message	Pass
TC_05	R 1.4	List Salon	Customer's location	List of nearby salons	List of nearby salons	Pass
TC_06	R 2.2	Manage Customer appointments	Selection of stylist and customer	Confirmation message	Confirmation message	Pass
TC_07	R 2.3	Manage Employees	Employee Details or selection	Confirmation message	Confirmation message	Pass
TC_08	R 2.4	Manage Services	Service Details or selection	Confirmation message	Confirmation message	Pass
TC_09	R 3.1	Inspections Of Salon Details	Selection of Salon	Salon Details	Salon Details	Pass
TC_10	R 3.2	Approve Salon	Selection of salon	Confirmation message	Confirmation message	Pass
TC_11	R 3.3	Reject Salon	Selection of salon	Confirmation message	Confirmation message	Pass

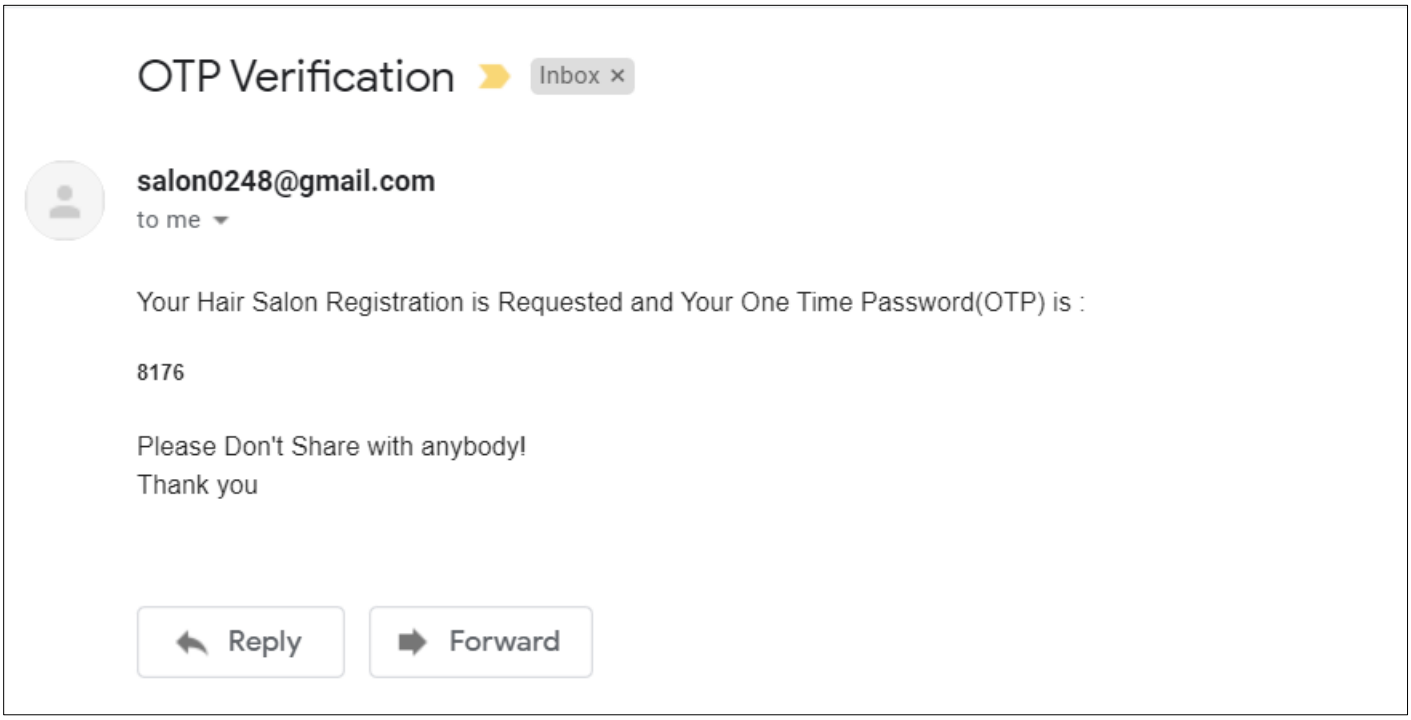
screenshots

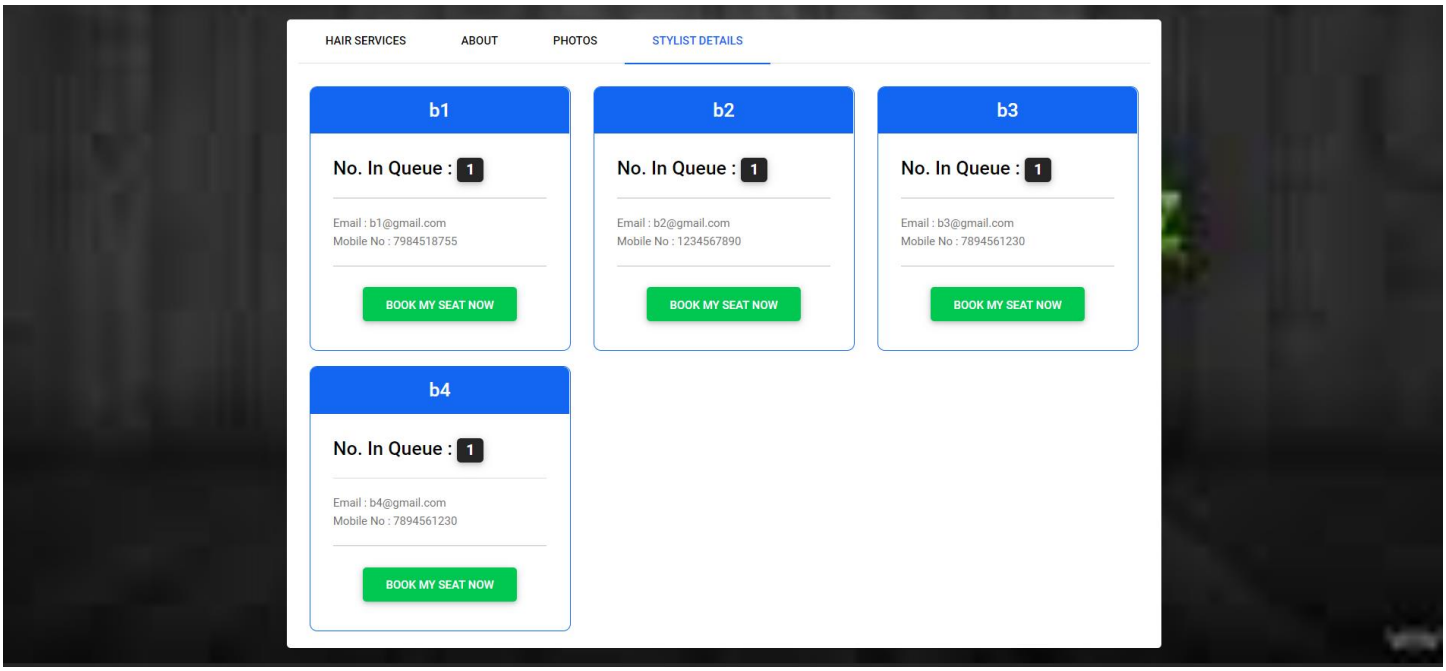


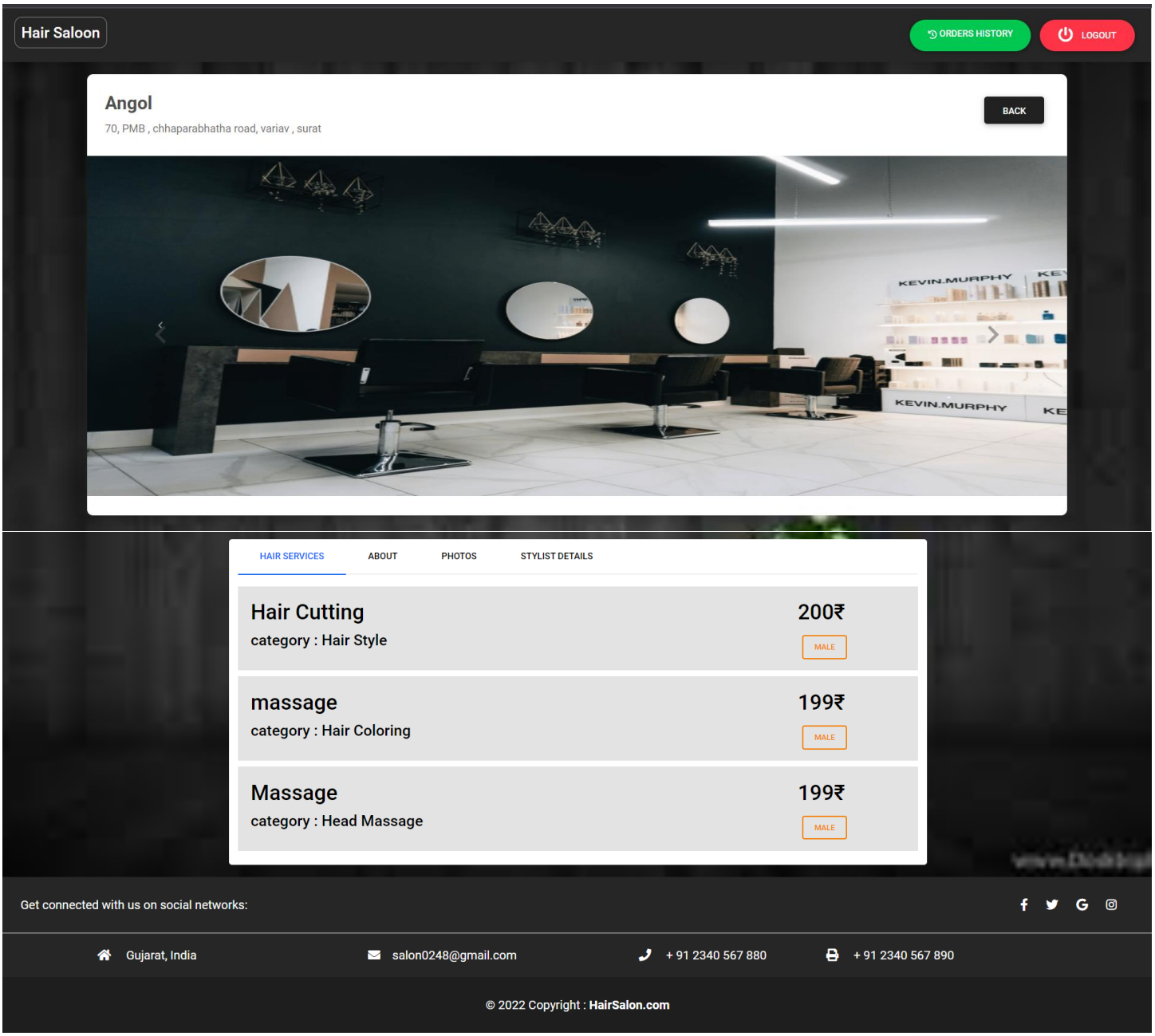



Hair Saloon
SERVICES
Customer SignUp
Email Address
yashvaghanipatel@gmail.com
Customer Name
yash vaghani
Mobile Number
7984518755
Password
Confirm Password
REGISTER
Already Loged In Click Here

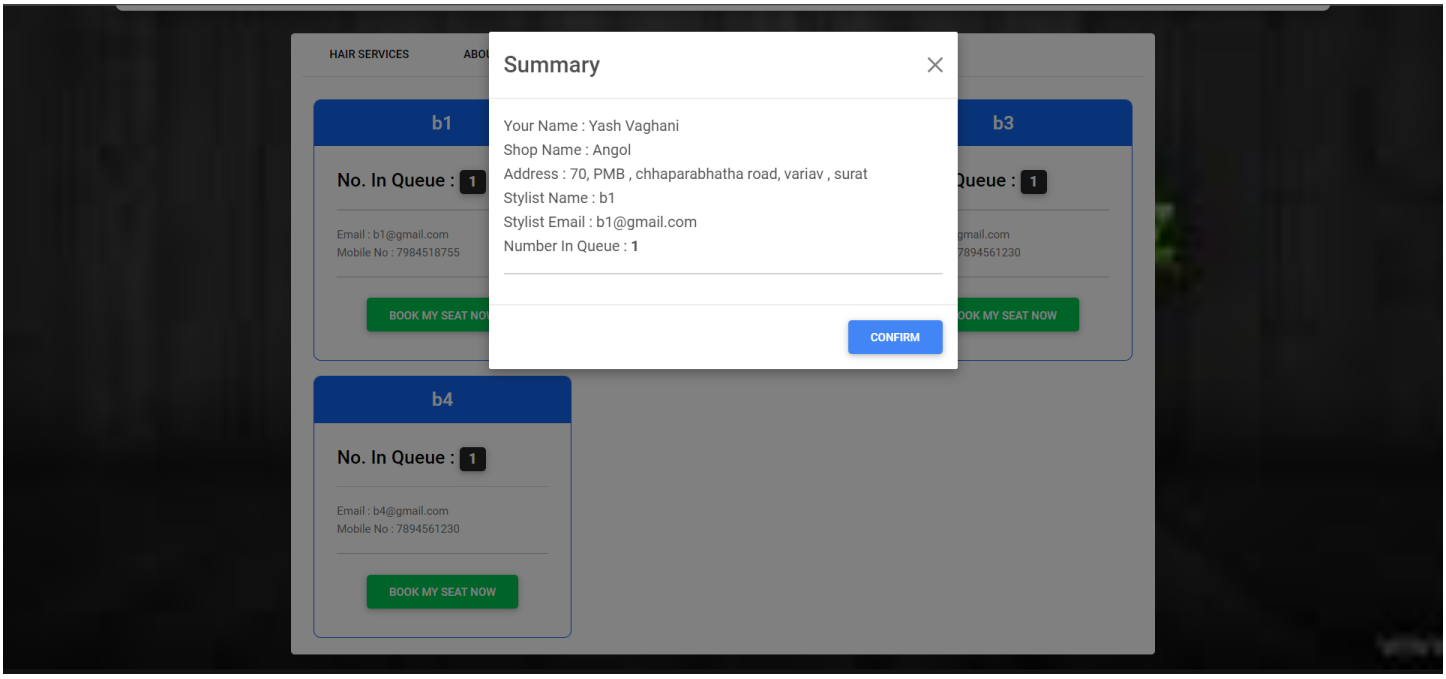

Hair Saloon
SERVICES
Otp Verification
Enter Otp
8176
VERIFY
Resend Otp From Here

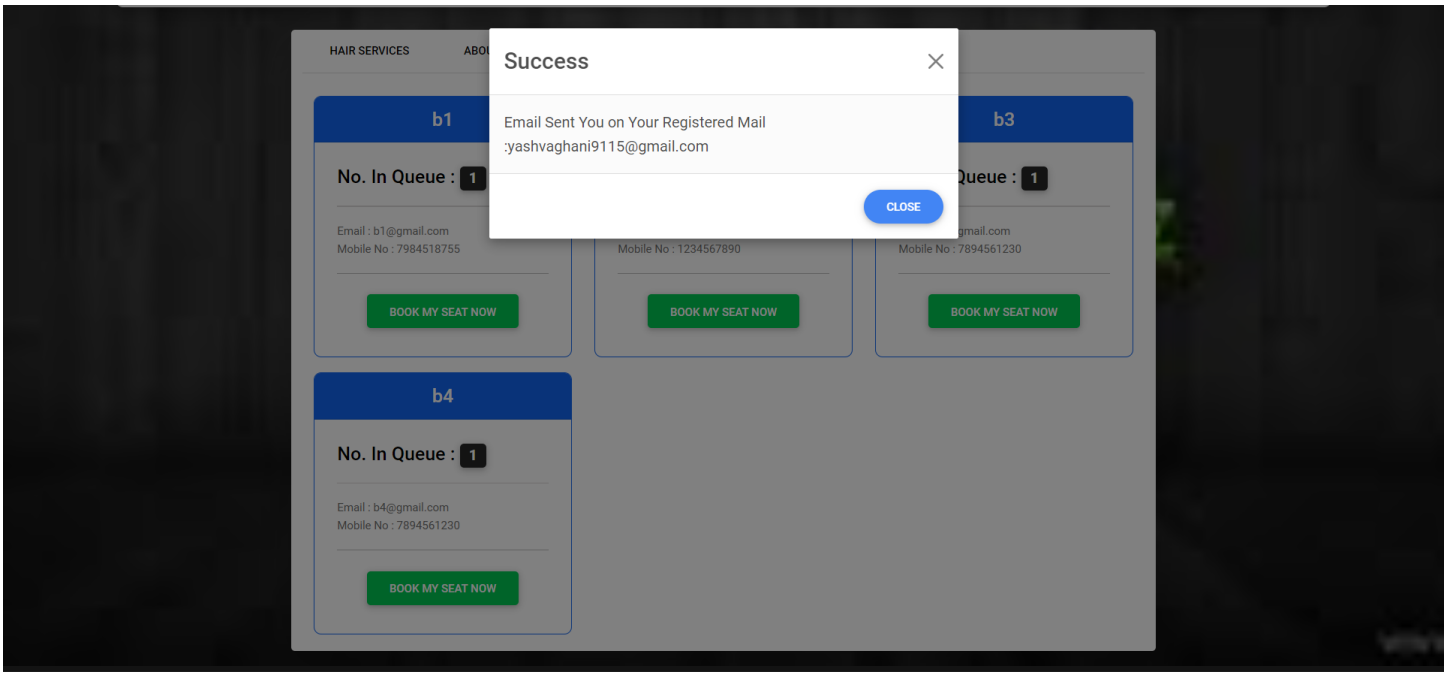

OTP Verification
Inbox ×
salon0248@gmail.com
to me
Your Hair Salon Registration is Requested and Your One Time Password(OTP) is :
8176
Please Don't Share with anybody!
Thank you
Reply
Forward

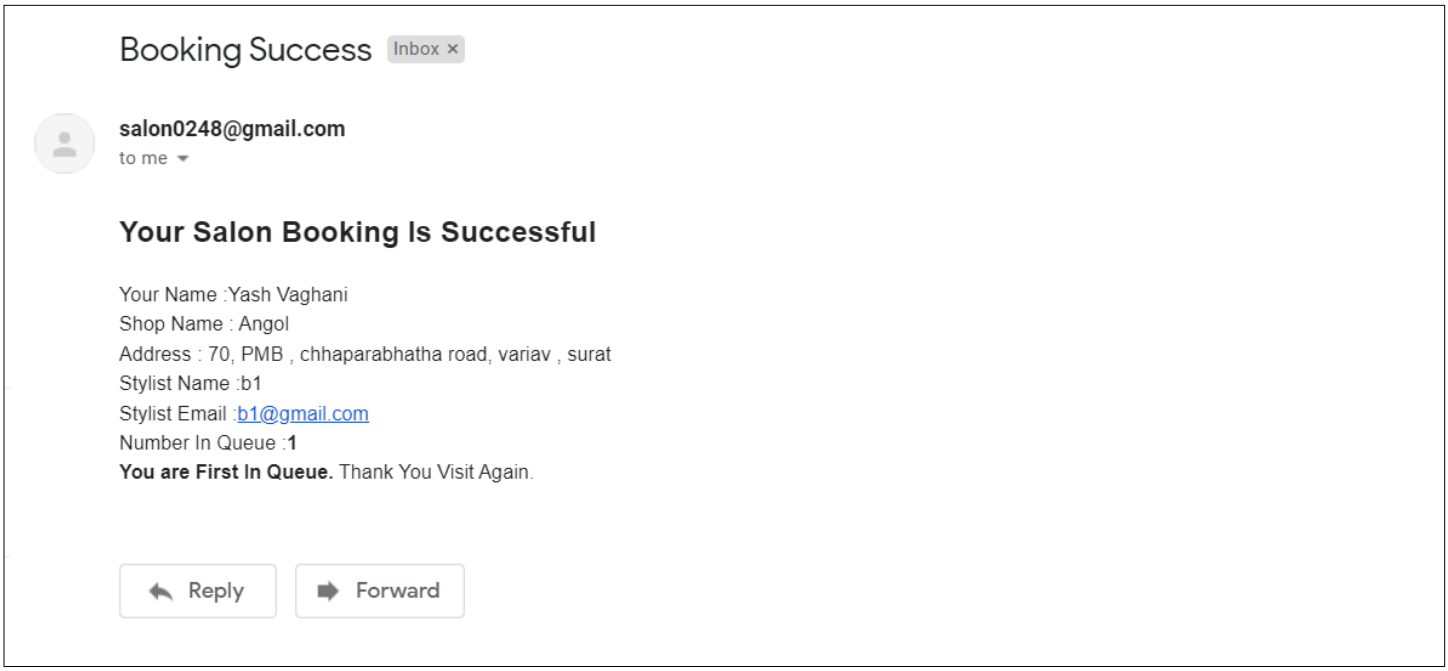

HAIR SERVICES
ABOUT
PHOTOS
STYLIST DETAILS
b1
No. In Queue : 1
Email : b1@gmail.com
Mobile No : 7984518755
BOOK MY SEAT NOW
b2
No. In Queue : 1
Email : b2@gmail.com
Mobile No : 1234567890
BOOK MY SEAT NOW
b3
No. In Queue : 1
Email : b3@gmail.com
Mobile No : 7894561230
BOOK MY SEAT NOW
b4
No. In Queue : 1
Email : b4@gmail.com
Mobile No : 7894561230
BOOK MY SEAT NOW


Hair Saloon
ORDERS HISTORY
LOGOUT
Angol
70, PMB , chhaparabhatha road, variav , surat
BACK
KEVIN.MURPHY
HAIR SERVICES
ABOUT
PHOTOS
STYLIST DETAILS
Hair Cutting
category : Hair Style
200₹
MALE
massage
category : Hair Coloring
199₹
MALE
Massage
category : Head Massage
199₹
MALE
Get connected with us on social networks:
Gujarat, India
salon0248@gmail.com
+ 91 2340 567 880
+ 91 2340 567 890
© 2022 Copyright : HairSalon.com

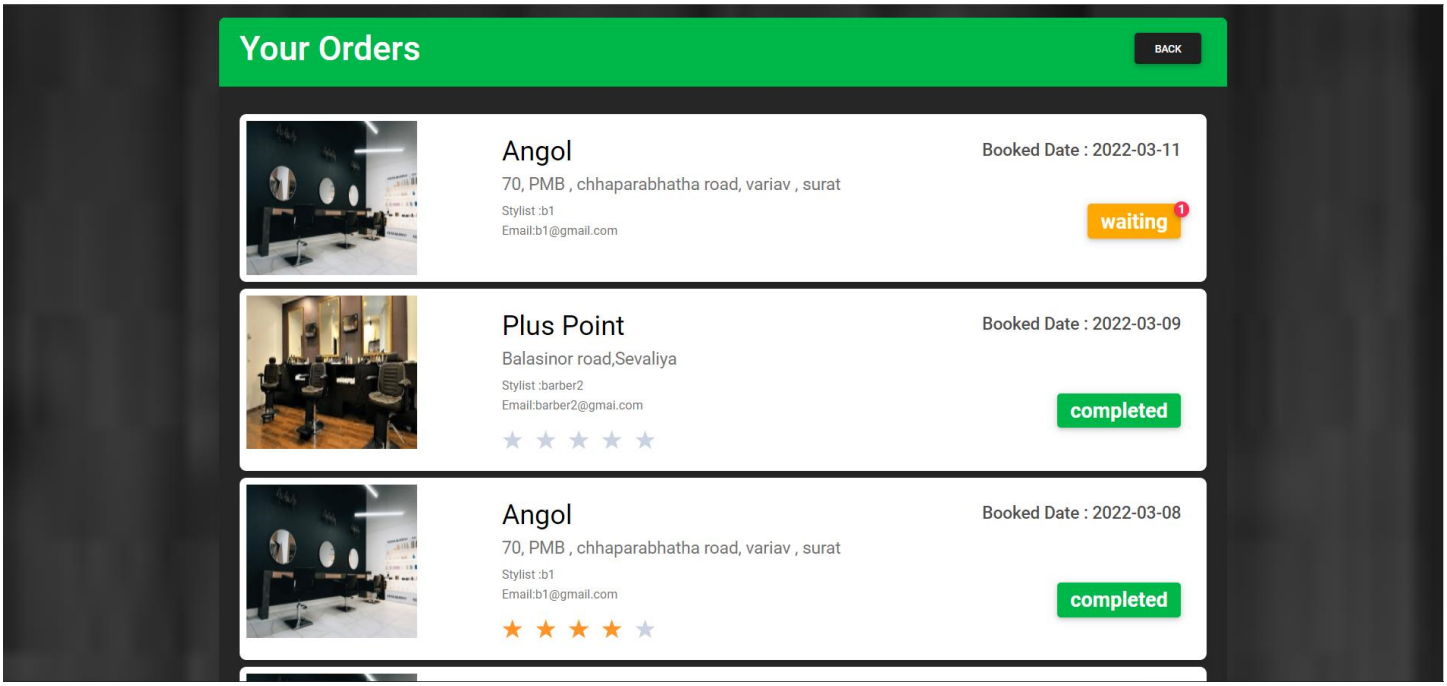
After selecting stylist



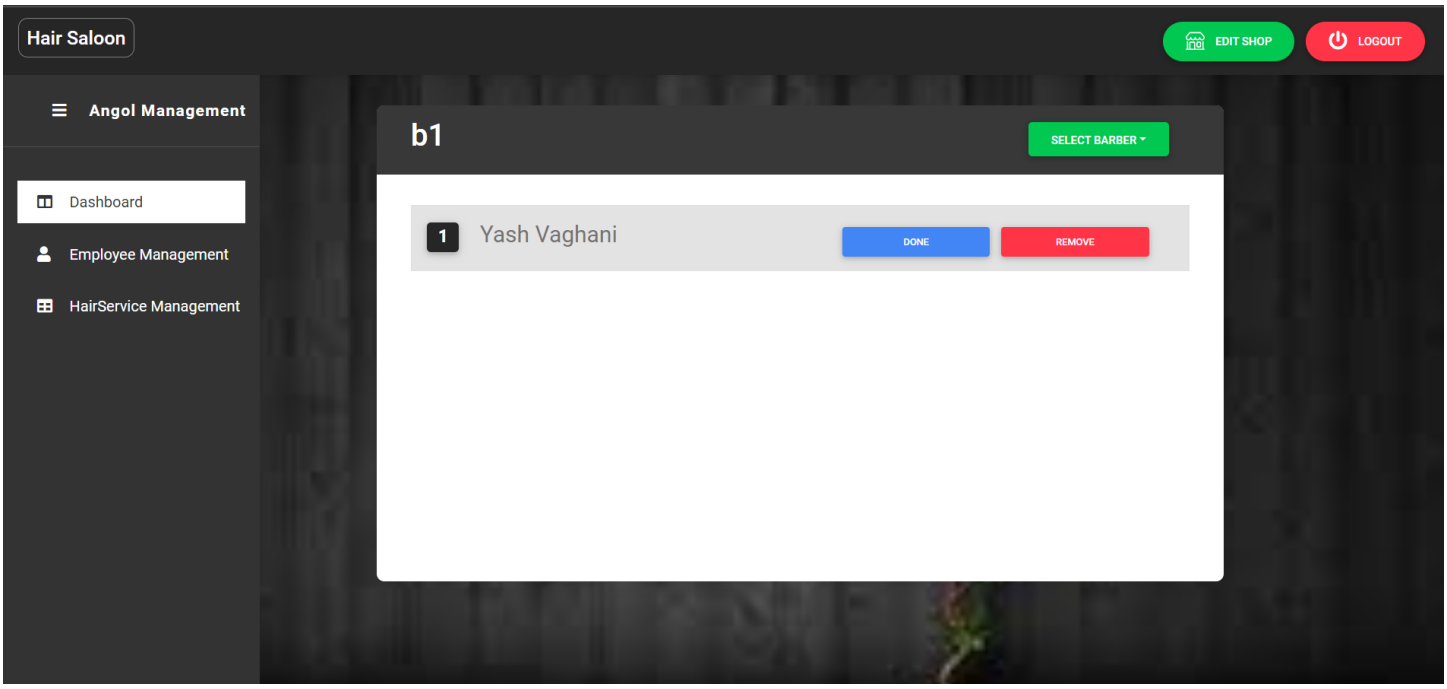


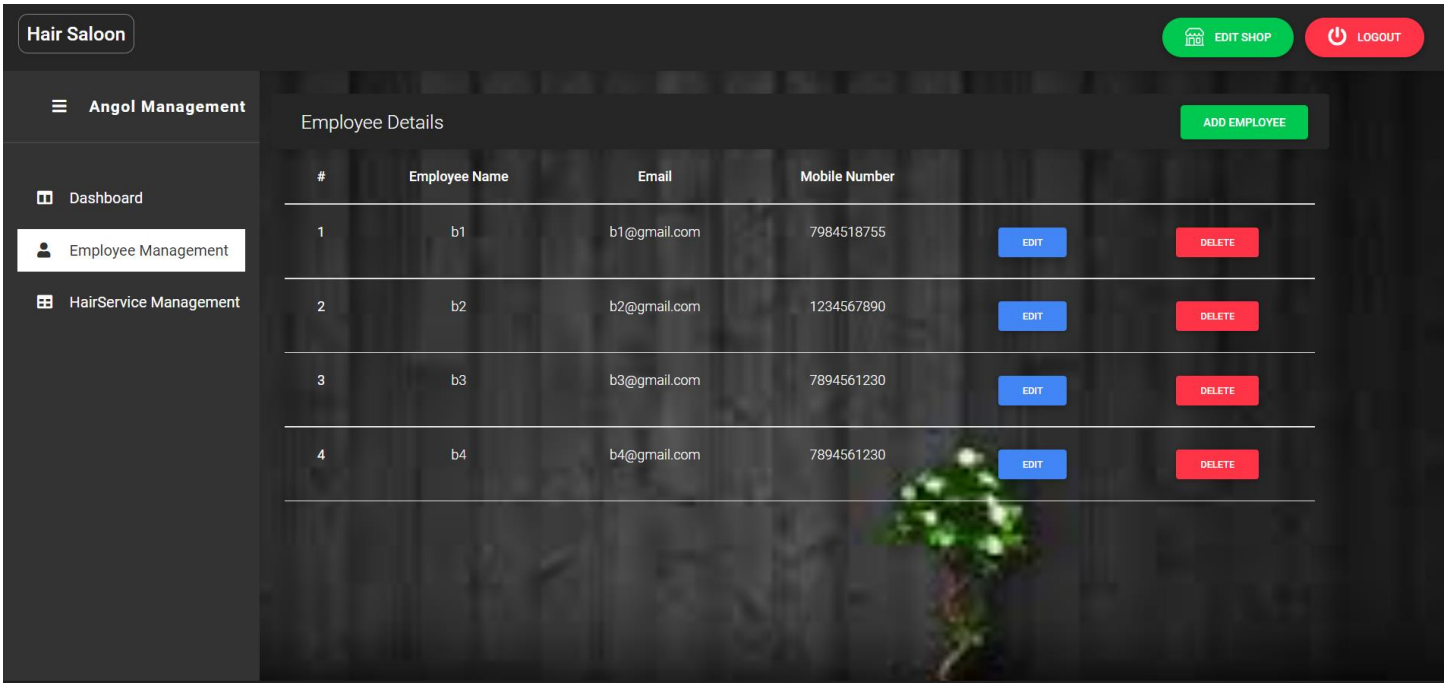


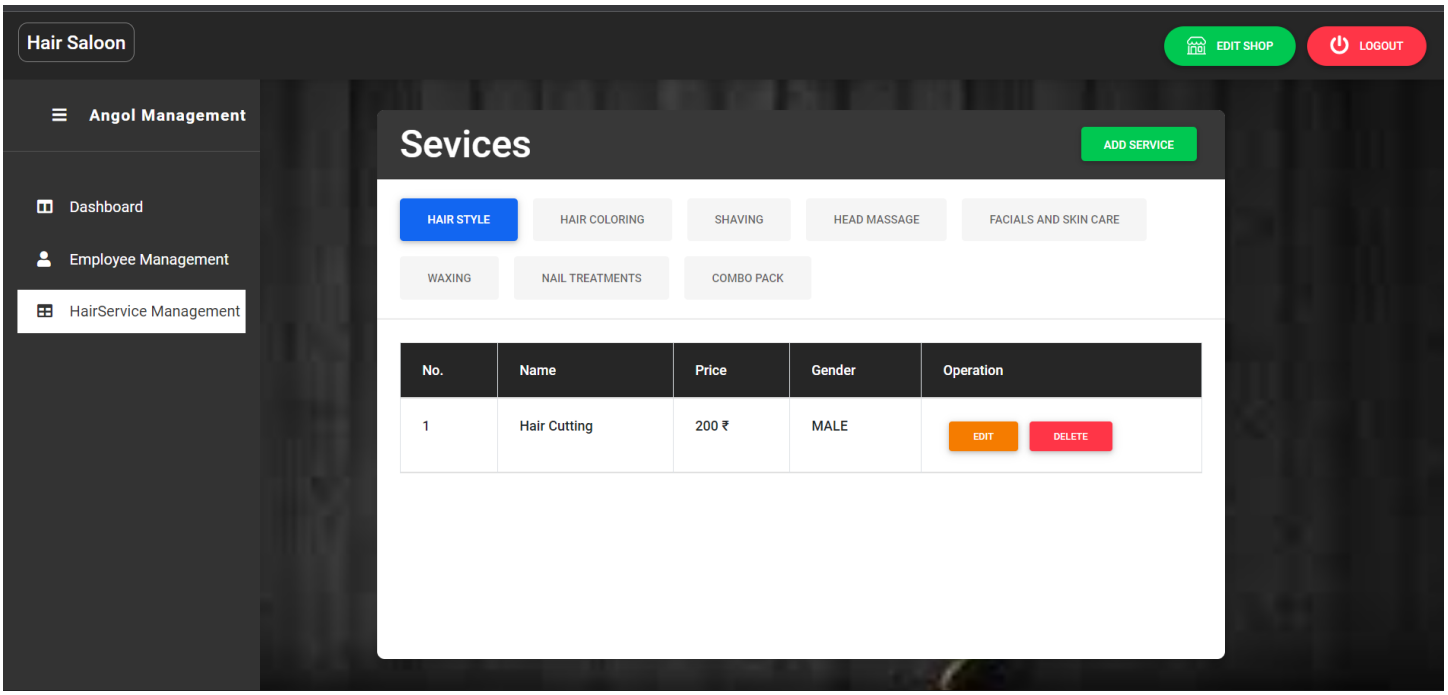
Order History:

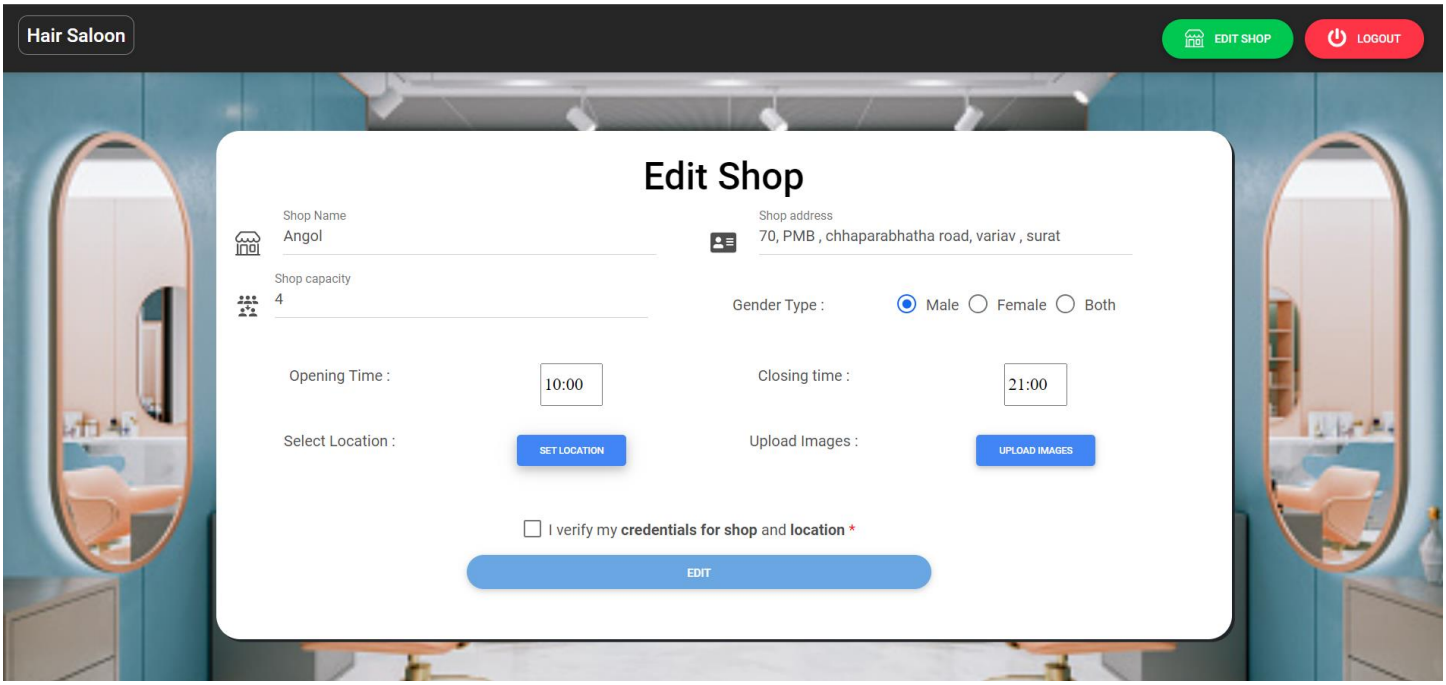


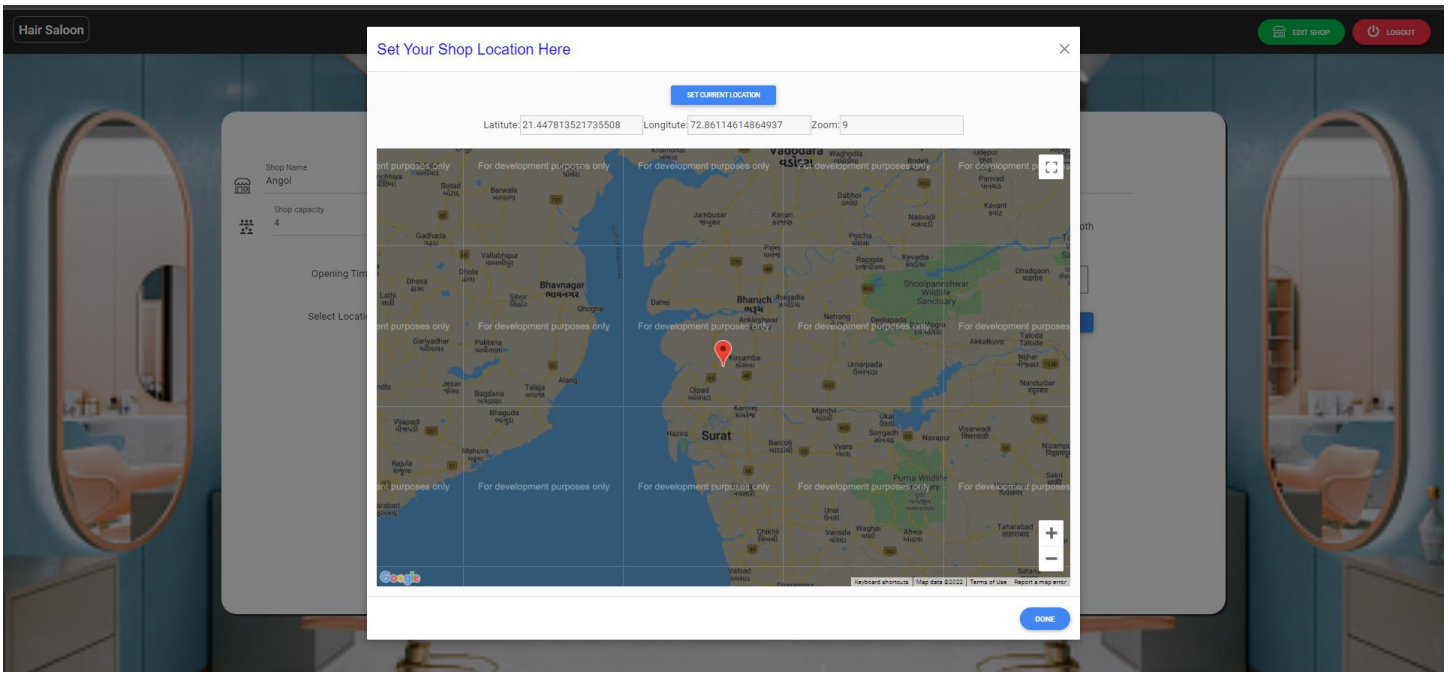
Owner Interface:

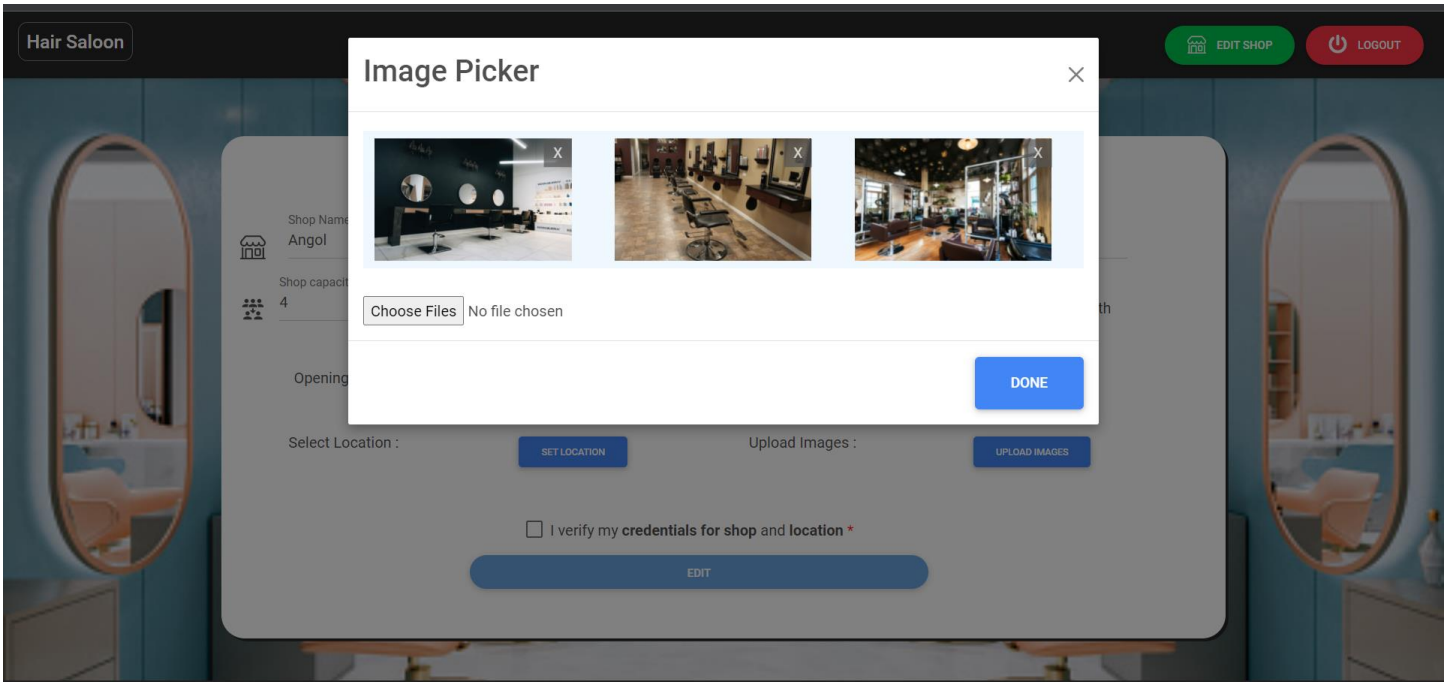




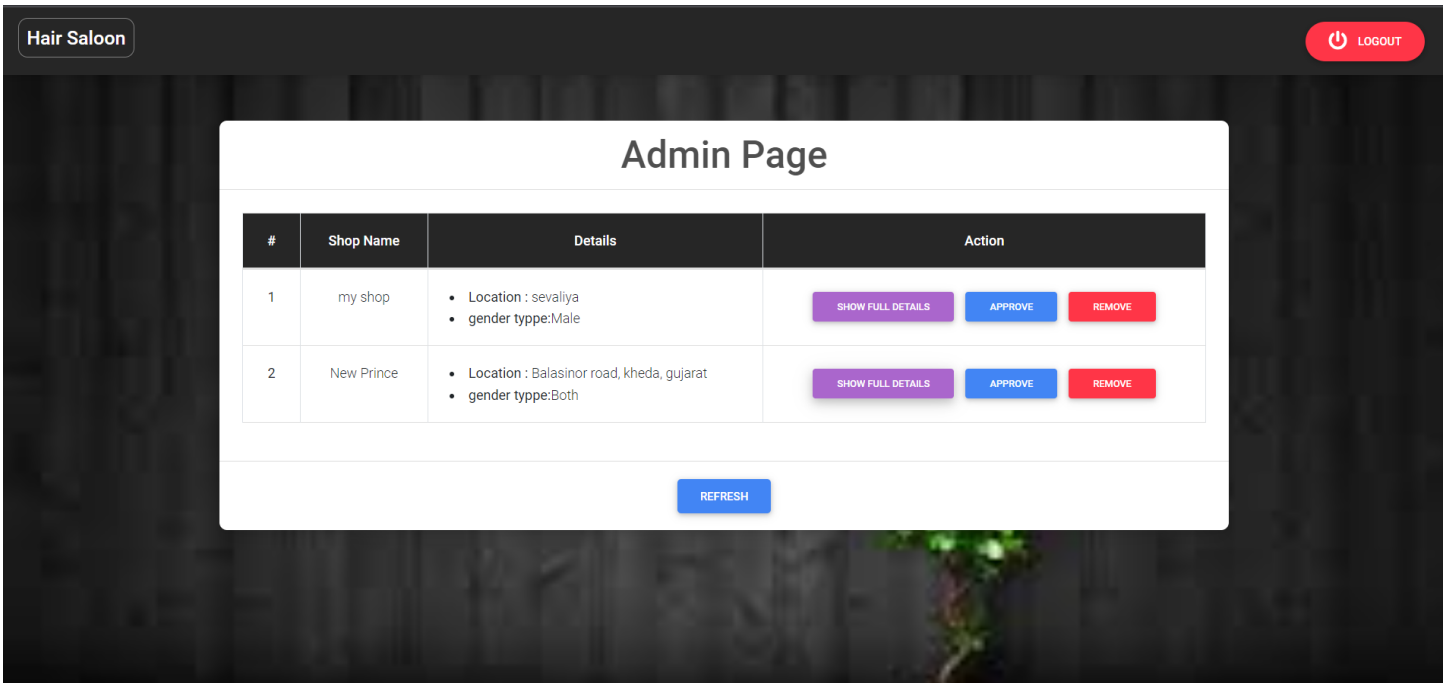

Hair Saloon
EDIT SHOP
LOGOUT
Angol Management
Dashboard
Employee Management
HairService Management
Sevices
ADD SERVICE
HAIR STYLE
HAIR COLORING
SHAVING
HEAD MASSAGE
FACIALS AND SKIN CARE
WAXING
NAIL TREATMENTS
COMBO PACK
No.
Name
Price
Gender
Operation
1
Hair Cutting
200 ₹
MALE
EDIT
DELETE

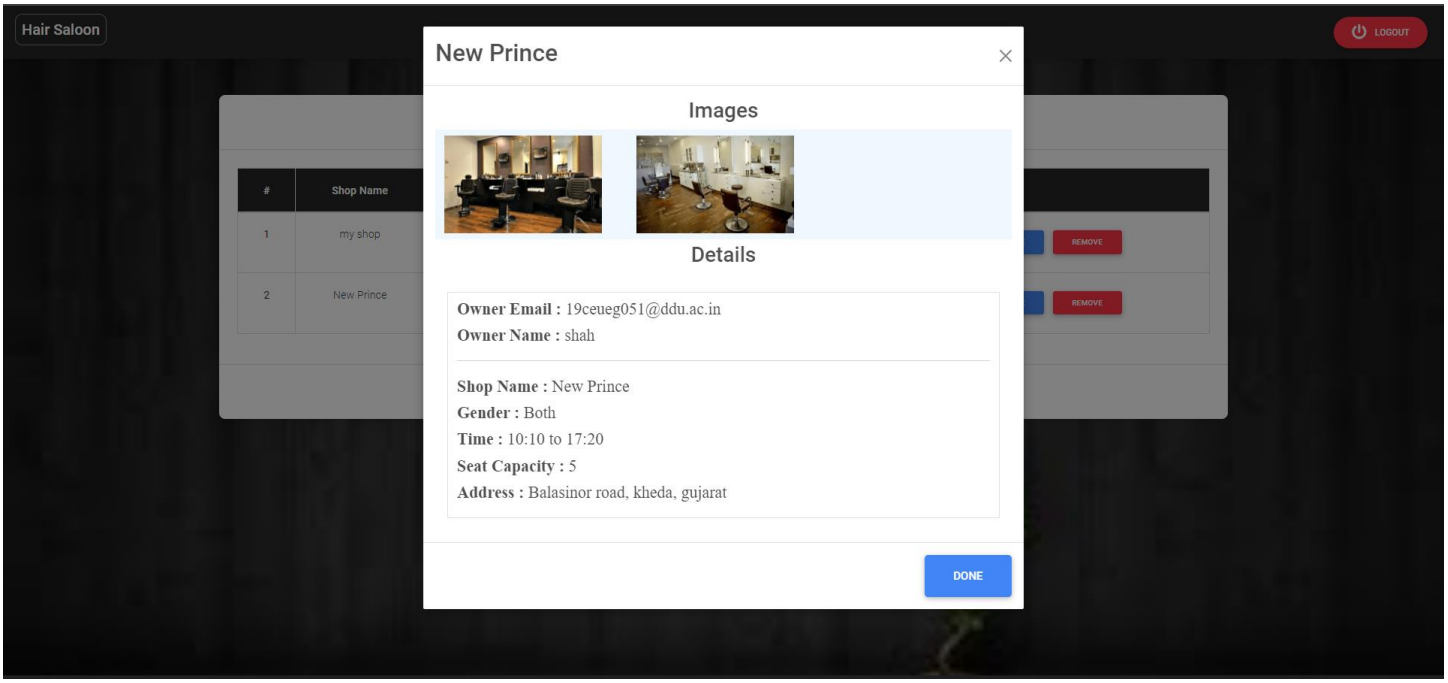

Hair Saloon
EDIT SHOP
LOGOUT
Edit Shop
Shop Name
Angol
Shop address
70, PMB , chhaparabhatha road, variav , surat
Shop capacity
4
Gender Type :
Male
Female
Both
Opening Time :
10:00
Closing time :
21:00
Select Location :
SET LOCATION
Upload Images :
UPLOAD IMAGES
I verify my credentials for shop and location *
EDIT


Hair Saloon
EDIT SHOP
LOGOUT
Set Your Shop Location Here
SET CURRENT LOCATION
Latitute: 21.447813521735508
Longitute: 72.86114614864937
Zoom: 9
DONE


Hair Saloon
EDIT SHOP
LOGOUT
Image Picker
Choose Files
No file chosen
DONE
Shop Name
Angol
Shop capacity
4
Opening
Select Location :
SET LOCATION
Upload Images :
UPLOAD IMAGES
I verify my credentials for shop and location *
EDIT

Admin Interface:





Conclusion

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

- Customer Registration / login (with all validation)
- List nearby salon
- Book appointment
- Order history
- Rate order
- Owner Registration / login (with all validation)
- Manage Services/Employee/Customer
- Edit salon
- Admin login
- Inspect and verify salon

Limitation and Future Enhancement

- In our system to list near by salon we have compared location of every salon instead of that we can improve it by only comparing specific range of longitude and latitude.
- In our system we can also include payment module.
- We can also display statistics of rating.

Reference / Bibliography

Following links and websites were referred during the development of this project:

- Stackoverflow
- Npmjs
- React
- MDBReact
- Cloudinary