

**SRI LANKA INTSITIUTE OF ADVANCE TECHNOLOGICAL EDUCATION**  
**HIGHER NATIONAL DIPLOMA IN INFORMATION TECNLOGY**



**Project Report**  
**Web Application For Beauty Saloon**

Name of student	Registration Number	Signature
M.T.T.M.Sewwandi	ANU/IT/2021/F/11	

**HNDIT 4052**  
**Programming Individual Project**  
**Supervisor : K.H.A. Hettige**

**Academic Year : 2021**

**Year II ,Semester 02**

**Submission date - 2024 May**

## DECLARATION

We hereby declare that this Final Project report document is based on my original work and has not been submitted previously for any other Diploma in any institution. To the best of our knowledge, it does not contain any material written by another person.

Student Name : M.T.T.M.Sewwandi

Reg number :ANU/IT/2021/F/11

Signature :

Date :

Supervisor Name :

Signature :

Date :

## ACKNOWLEDGMENTS

It's a great pleasure for us to take this opportunity to deliver My worthy thanks for those who have been my guardians and supporters to successfully complete this web application. I pleasure deliver our since gravitate to supervisor Mr K.H.A Hettige for his continuous guidance and support throughout the development.

It's my duty to be thanking our head of department of HNDIT Mrs. D.M.N.K Dissanayake For her continuous guidance to this project .It's our bond to be thanking demonstrators of the department of IT, for they support and continuous supervision.

It's a great pleasure to remind our Institute, Advanced Technological Institute Anuradhapura which gave us the permission to develop this project

## ABSTRACT

This online appointment booking web application for salons offers a convenient platform for users to schedule appointments with ease. Users can register, login, and book appointments for desired dates and times. The system allows users to view booking details and manage their appointments effortlessly. Additionally, administrators have access to manage salon services and review incoming appointment requests, accepting or declining them based on availability. With a user-friendly interface and efficient functionality, this application streamlines the appointment booking process for both clients and salon staff, enhancing overall customer satisfaction and operational efficiency.

## Table of Contents

DECLARATION .....	7
ACKNOWLEDGMENTS .....	8
ABSTRACT.....	9
<b>01.INTRODUCTION TO PROJECT .....</b>	<b>12</b>
1.1 INTRODUCTION .....	12
1.2 BACKGROUND .....	13
1.3 PROBLEM DEFINITION .....	14
1.4 AIM & OBJECTIVES .....	15
1.4.1 AIM.....	15
1.4.2 OBJECTIVES .....	15
1.4 SCOPE WITH BOUNDARIES .....	16
<b>02. METHODOLOGY AND DESIGN.....</b>	<b>17</b>
2.1 REQUIREMENTS.....	17
2.1.1 SOFTWARE REQUIREMENTS .....	17
2.1.2 HARDWARE REQUIREMENTS.....	17
2.1.3 TECHNOLOGY .....	17
2.2 FUNCTIONAL REQUIREMENTS .....	18
2.2.1 USE CASE 01 .....	19
2.2.2.USE CASE 02.....	21
2.2.3 USE CASE 03.....	23
2.2.4.USE CASE 04.....	25
2.2.5 USE CASE 05.....	26
2.2.6 USE CASE 06.....	28
2.2.7 USE CASE 07.....	30
2.2.8. USE CASE 08.....	32
2.2.9 USE CASE 09.....	33
2.2.10 USE CASE 10.....	35
2.3. NON – FUNCTIONAL REQUIREMENTS.....	37
2.4 .SYSTEM DESIGN .....	38
2.4.1 INTERFACE DESIGN FOR USER MODULE .....	38
2.4.2 INTERFACE DESIGN FOR ADMIN MODULE.....	40
2.4.3 DATABASE DESIGN.....	41

<b>03. TESTING</b> .....	42
3.1. TEST CASES.....	42
<b>04. IMPLEMENTATION</b> .....	53
4.1 INSTALLATION GUIDE .....	53
4.2 USER GUIDE.....	53
4.3 BACKUP PROCEDURES/CYCLES .....	53
4.4 SECURITY PROCEDURES .....	54
<b>05 EVALUATION &amp; CONCLUSION</b> .....	55
<b>06.REFERENCES</b> .....	56

# 01.INTRODUCTION TO PROJECT

## 1.1 INTRODUCTION

The development of an online appointment booking web application for salons represents a significant milestone in leveraging modern technology to enhance the efficiency and convenience of salon services. This project integrates HTML, CSS, JavaScript, PHP, and MySQL to create a seamless platform for users to schedule appointments and for salon administrators to manage their services effectively.

At its core, this web application aims to simplify the process of booking salon appointments for users while providing salon owners with a robust management system. Through the utilization of HTML, CSS, and JavaScript, the user interface is designed to be intuitive and user-friendly, ensuring a seamless experience for clients navigating the booking process. JavaScript enhances the interactivity of the application, allowing for dynamic updates and real-time feedback during the booking process, enhancing user engagement.

On the server-side, PHP facilitates the processing of user requests and interactions, ensuring secure data transmission and efficient handling of booking transactions. MySQL serves as the backend database management system, storing user information, appointment details, and salon service data. This relational database enables effective data management, allowing administrators to access and manipulate information as needed. By leveraging these technologies, this web project not only simplifies the appointment booking process but also empowers salon owners with tools to streamline their operations and provide exceptional customer service.

## 1.2 BACKGROUND

The motivation behind developing this online appointment booking web application for salons stemmed from the recognition of a common challenge faced by both salon owners and clients: the cumbersome and time-consuming process of scheduling appointments. With the proliferation of digital solutions in various industries, it became evident that leveraging technology could significantly improve the efficiency and convenience of salon services.

This project emerged as a response to this need, aiming to create a user-friendly platform that simplifies the appointment booking process for clients while providing salon owners with effective management tools. By harnessing the power of HTML, CSS, JavaScript, PHP, and MySQL, the project seeks to streamline booking procedures, enhance user experience, and optimize salon operations. Through this initiative, the goal is to bridge the gap between traditional salon management practices and modern technological advancements, ultimately improving the overall customer satisfaction and business performance of participating salons.



## 1.3 PROBLEM DEFINITION

Traditional salon appointment booking systems often involve phone calls or walk-in appointments, leading to inefficiencies, missed opportunities, and customer frustration. Clients face challenges in securing preferred appointment times, while salon owners grapple with manual booking processes and limited visibility into their schedule. This project addresses these issues by developing an online booking platform. Clients can easily schedule appointments at their convenience, reducing wait times and improving satisfaction. Meanwhile, salon owners gain access to a centralized system for managing appointments, enhancing organization and maximizing bookings. By automating and digitizing the booking process, this project aims to overcome the limitations of traditional methods, providing a streamlined solution for both clients and salon administrators. [1]

## 1.4 AIM & OBJECTIVES

### 1.4.1 AIM

The aim of our project is to design and implement a user-friendly salon management system that addresses the inefficiencies of traditional appointment booking methods and enhances overall operational effectiveness in the beauty industry.

### 1.4.2 OBJECTIVES

- Develop an intuitive user interface for easy appointment scheduling and management.
- Implement a centralized database system to store client information and appointment records securely.
- Integrate automated reminders and notifications to reduce no-show appointments and improve customer engagement.
- Provide reporting and analytics functionalities to track salon performance and customer preferences.
- Enhance scalability and adaptability to accommodate future growth and technological advancements in the beauty industry.

## 1.4 SCOPE WITH BOUNDARIES

This project's scope encompasses the development of a web-based appointment booking system tailored specifically for salons. It includes features such as user registration, login authentication, appointment scheduling, viewing booking details, and administrative functions for managing services and appointments. However, it does not extend to advanced features like payment processing or integration with external platforms. For instance, while users can book appointments, the system does not handle financial transactions. Additionally, the scope does not cover extensive customization options for salon owners beyond basic service management. The project focuses on core functionalities essential for efficient appointment management within a salon setting.

## 02. METHODOLOGY AND DESIGN

### 2.1 REQUIREMENTS

#### 2.1.1 SOFTWARE REQUIREMENTS

<b>SOFTWARE</b>	<b>DESCRIPTION</b>
Windows Operating System	Arrange all the softwares and hardwares on the device
Any Web Browser	Testing and Debugging, Implementation of Frontend Code,
Front-End development languages: PHP,CSS,JS,HTML5	For design the content of the web application.(client-side)
Back-End development languages: PHP,MySQL	For organizing and database creations(server-side)
XAMPP Server	Create a connection between Apache and MySQL in the local host Server

#### 2.1.2 HARDWARE REQUIREMENTS

<b>HARDWARE</b>	<b>DESCRIPTION</b>
Acer E5-576-38CG Laptop Intel core i3 – 7100U CPU@ 2.20GHz RAM 4GB Storage – 500GB ,128SSD	To develop the system and installation of the software
Keyboard and mouse	

#### 2.1.3 TECHNOLOGY

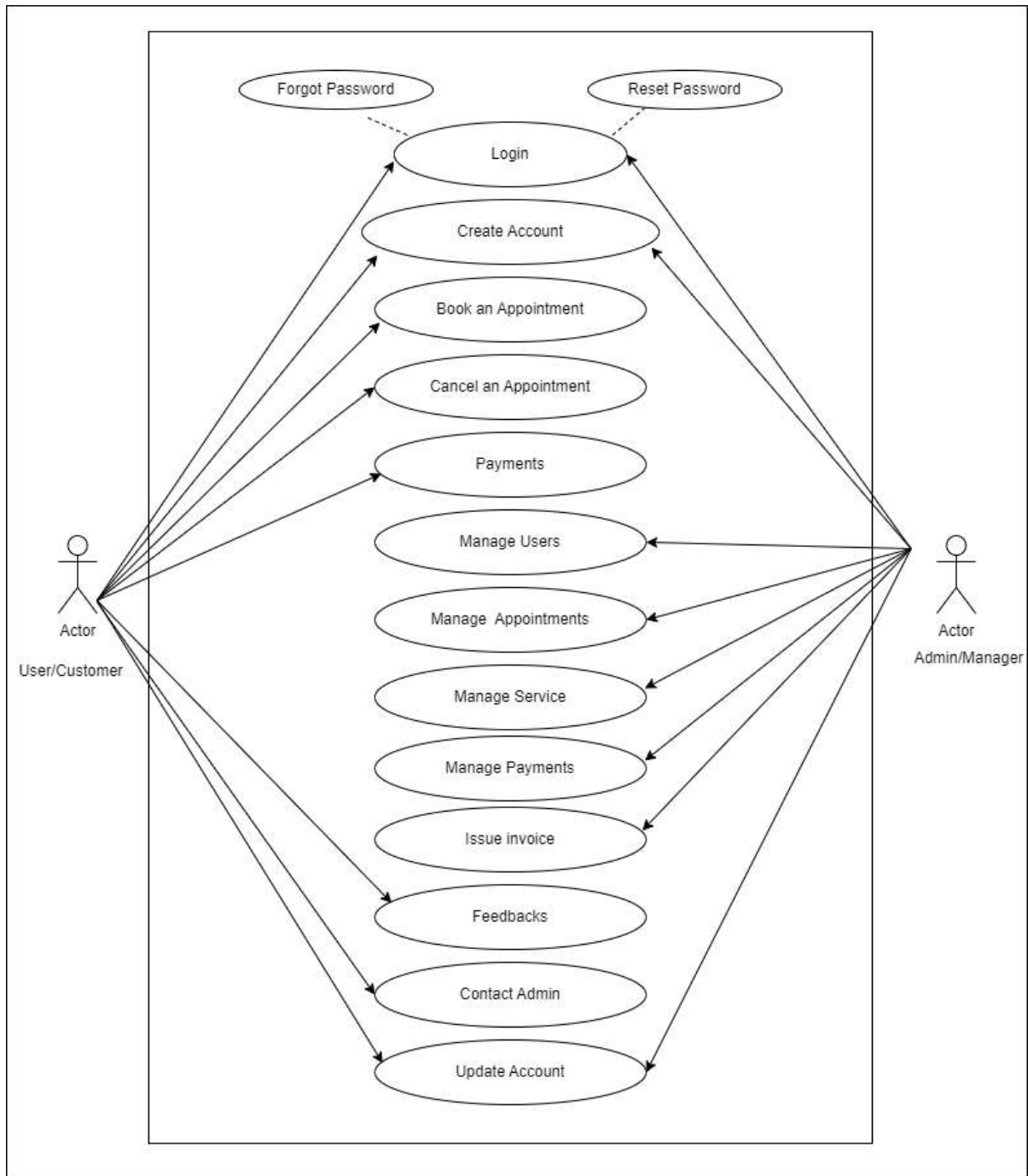
	<b>LANGUAGE</b>
Frontend	HTML,CSS,Java Script
Backend	PHP
Database	MySQL

## 2.2 FUNCTIONAL REQUIREMENTS

In the creation and development of this Salon web application, there are functional requirements to perform.

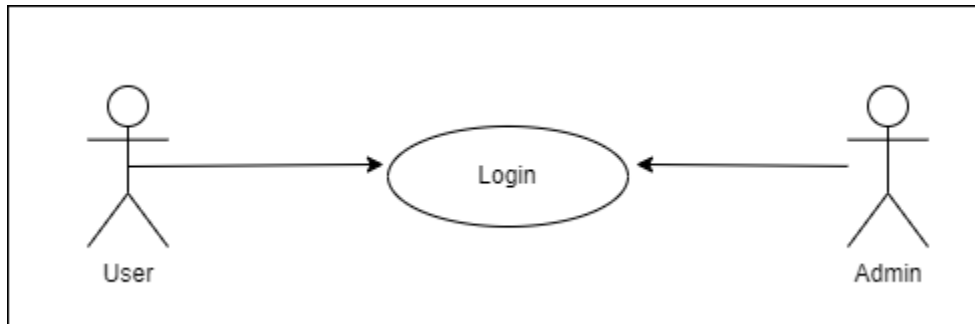
After careful analysis the system has been identified to have the following modules:

1. Administrator module
2. User module



## 2.2.1 USE CASE 01

### LOGIN

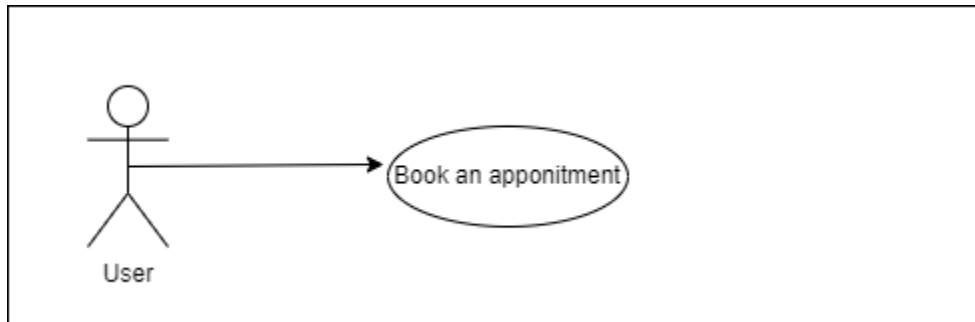


Use Case Name:	Login
Actor:	01.User 02.Admin
Overview:	Admin and User Both of actors an login the system.
Pre-condition:	01 Admin: The admin must have an admin account in the system 02 User : The user has to fill out the user registration form for self-registration to log in to the system.
Description:	After login page loaded,  01.Admin : The admin must enter the correct email address and correct password and press the login button to log in to the admin login page .  02. User : The user can log in to the system by entering the correct email address and correct password and pressing the login button.
Alternative flows:	1 <sup>st</sup> - If any actor wouldn't enter the username and password (blank) and then press the login button, they will get an error message like "Please fill out this field". 2 <sup>nd</sup> - If any actor would enter only the username only the password then press

	<p>the login button, they will get an error message like “Please fill out this field”.</p> <p>3<sup>rd</sup> - If any actor would enter the username and password then press the login button, but if that username or password is incorrect, they will get an pop up error message like “Please enter the correct username or password”.</p>
Post-conditions:	<p>*After pressing the login button each actor will successfully log into the System and they redirect to their own dashboard page and dashboard page will be loaded as per the account type.</p> <p>01 Admin: In the admin dashboard, there will be loaded the registered and the registered users’ details.The admin can check data in the user registration forms</p> <p>02.User: In the user dashboard, it will be loaded and display the user’s profile details and also the user can book to the appointment in the salon and the user can see his or her booked appointment list details.</p> <p>*Each actor can log out from the system by clicking on “Log Out”.</p>

## 2.2.2.USE CASE 02

### BOOK AN APPONTMENT



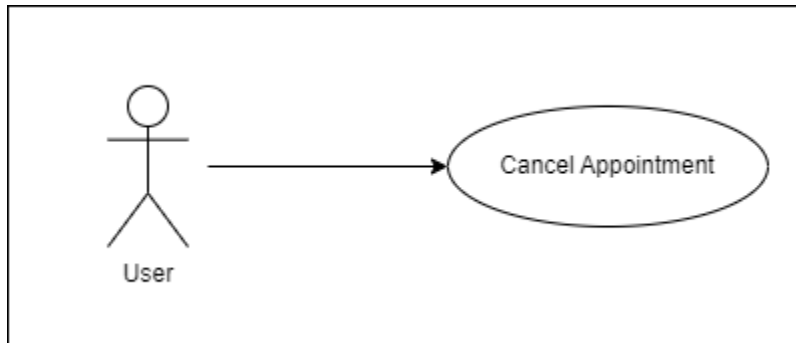
Use Case Name:	Book an Appointment
Actor:	User
Overview:	User can book the appointment
Pre-condition:	The user must log in to the system
Description:	<p>After the admin add the salon service. The user can view ,select and book the appointment that he or she selected.</p> <ul style="list-style-type: none"><li>• Access Appointment Booking: After logging in, the user navigates to the appointment booking section of the system.</li><li>• View Available Services: The user views the list of available salon services for booking.</li><li>• Select Service: From the list of available services, the user selects the desired service for which they want to book an appointment.</li><li>• Choose Appointment Slot: The user selects a suitable appointment slot from the available options corresponding to the selected service.</li><li>• Confirm Appointment: The user confirms the selected service and appointment slot for booking.</li></ul>
Alternative flows:	<ul style="list-style-type: none"><li>• Invalid Selection: If the user fails to select a service or appointment slot before confirming, they receive an error message prompting them to make a selection.</li></ul>



	<ul style="list-style-type: none"> <li>• Booking Conflict: If the selected appointment slot is already booked by another user, the system notifies the user and prompts them to choose an alternative slot.</li> </ul>
Post-conditions:	<p>Upon successful booking:</p> <ul style="list-style-type: none"> <li>• The appointment is added to the user's appointment list.</li> <li>• The user can view the booked appointment details in their dashboard.</li> <li>• The system may send a confirmation notification to the user.</li> <li>• The user has the option to log out from the system by clicking on the "Log Out" button.</li> </ul>

### 2.2.3 USE CASE 03

#### CANCEL AN APPOINTMENT

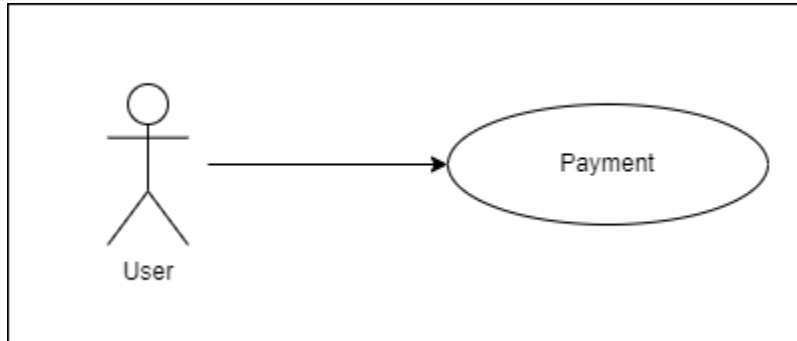


Use Case Name:	Cancel the Appointment
Actor:	User
Overview:	User can cancel the user's booked appointment
Pre-condition:	The user must be logged in to the system and have a booked appointment that they wish to cancel.
Description:	<p>After user booking an appointment if user need to cancel it then user can cancel appointment</p> <ul style="list-style-type: none"><li>• Access Appointment Management: After logging in, the user navigates to the appointment management section of the system.</li><li>• View Booked Appointments: The user views their list of booked appointments.</li><li>• Select Appointment to Cancel: From the list of booked appointments, the user selects the appointment they wish to cancel.</li><li>• Initiate Cancellation: The user initiates the cancellation process for the selected appointment.</li></ul>
Alternative flows:	<ul style="list-style-type: none"><li>• Late Cancellation: If the appointment cancellation request is made close to the scheduled appointment time, the system may prompt the user with a confirmation message informing them of any cancellation fees or penalties.</li><li>• Confirmation Prompt: After initiating the cancellation process, the system may prompt the user with a confirmation dialog to ensure they indeed want to cancel the appointment.</li></ul>

	<ul style="list-style-type: none"> <li>• Confirm Cancellation: The user confirms the cancellation of the selected appointment</li> </ul>
Post-conditions:	<p>If user cancel appointment there is no appointment under the user .</p> <p>Upon successful cancellation:</p> <ul style="list-style-type: none"> <li>• The appointment is removed from the user's booked appointments list.</li> <li>• The slot for the canceled appointment becomes available for booking by other users.</li> <li>• The system may send a notification to the user confirming the cancellation.</li> <li>• If any cancellation fees apply, the system may process refunds if applicable.</li> <li>• The user has the option to view their updated appointment list or perform other actions in their dashboard.</li> </ul>

## 2.2.4.USE CASE 04

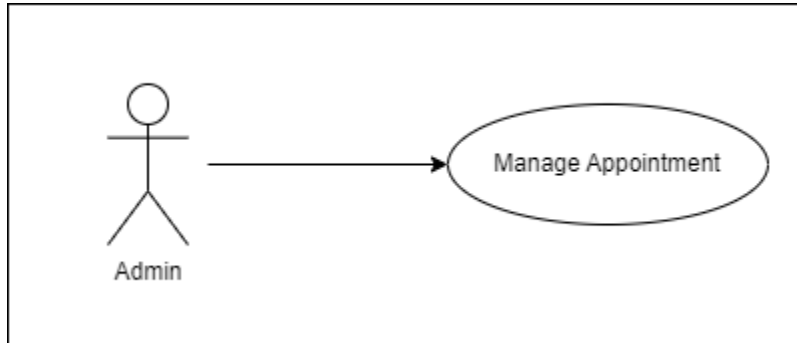
### PAYMENT



Use Case Name:	Payment
Actor:	User
Overview:	User can pay the payment fee for their booked appointments
Pre-condition:	User must log in to system and must need to book appointment before they pay the fee.
Description:	After the book user;s appointment then user can pay the fee.
Alternative flows:	If something wrong with payment get a error message unsuccessful payment.
Post-conditions:	<div>After user pay the payment and submit payment confirm button user can book their appoinment successfully<ul style="list-style-type: none"><li>• The appointment is marked as paid in the system.</li><li>• The user receives a payment confirmation notification.</li><li>• The system may update the user's appointment status to reflect the payment.</li><li>• If applicable, the system may send a receipt</li><li>• The user has the option to view their updated appointment details or perform other actions in their dashboard.</li></ul></div>

## 2.2.5 USE CASE 05

### MANAGE APPOINTMENT

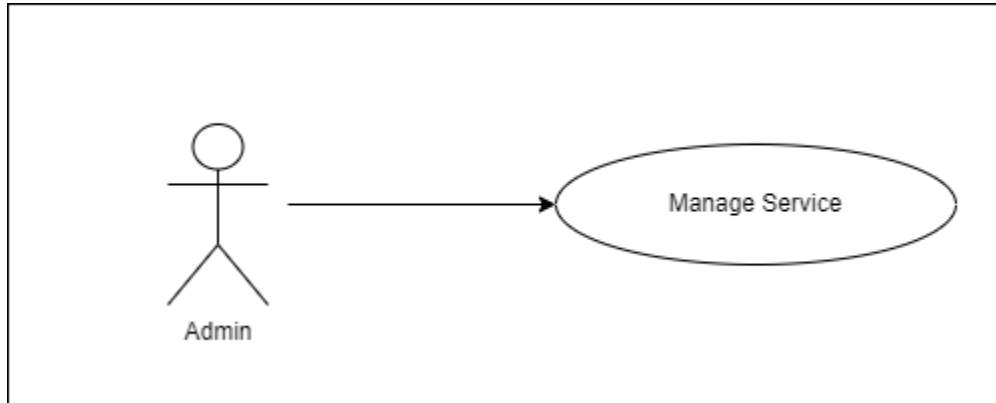


Use Case Name:	Manage Appointment
Actor:	Admin
Overview:	Admin can select or reject the appointment the user book.
Pre-condition:	Admin must log in to the system before manage the appointments.
Description:	<ul style="list-style-type: none"><li>• After admin log in the admin dashbord admin can view the user booking appointments and manage it.Admin can select or reject the appointment ,remart and update the status of appointment user can view it</li><li>• Access Appointment Management: After logging in, the admin navigates to the appointment management section of the system.</li><li>• View Appointments: The admin views the list of appointments scheduled within the system.</li><li>• Search and Filter Appointments: The admin may search for specific appointments by date, time, user name, or other relevant criteria. They can also filter appointments based on various parameters such as status (booked, canceled, completed, etc.).</li><li>• Modify Appointments: The admin has the ability to modify existing appointments, including changing the appointment time, service, or user details if necessary.</li></ul>

	<ul style="list-style-type: none"> <li>• Cancel Appointments: If required, the admin can cancel appointments on behalf of users. This could be due to unforeseen circumstances, conflicts, or other reasons.</li> </ul>
Alternative flows:	<ul style="list-style-type: none"> <li>• Appointment Conflict: If the admin attempts to schedule a new appointment that conflicts with an existing appointment, the system notifies them and prompts for resolution (e.g., reschedule or cancel conflicting appointment).</li> <li>• Appointment Rescheduling: If a user requests to reschedule an appointment, the admin can facilitate the process by modifying the appointment details accordingly.</li> <li>• Generate Reports: The admin may generate reports summarizing appointment statistics, such as total appointments booked, canceled, or completed within a specific timeframe.</li> </ul>
Post-conditions:	<ul style="list-style-type: none"> <li>• After manage appointment admin can do his other manage salon work or log out in system.</li> <li>• The appointment details are updated in the system according to the admin's actions.</li> <li>• Users may receive notifications regarding any changes made to their appointments.</li> <li>• The admin has the option to view updated appointment records or perform further management tasks as needed.</li> </ul>

## 2.2.6 USE CASE 06

### MANAGE SERVICE



Use Case Name:	Manage Services
Actor:	Admin
Overview:	Admin can manage the services in the salon .
Pre-condition:	Admin must log in to the system before admin manage the services.
Description:	<p>After admin log in to system admin can manage saloon services .Admin can add new services,update current services and delete current services.</p> <ul style="list-style-type: none"><li>• Access Service Management: After logging in, the admin navigates to the service management section of the system.</li><li>• Add New Service: The admin has the ability to add a new service to the list of available salon services. This includes providing details such as service name, description, duration, and price.</li><li>• Edit Existing Service: If necessary, the admin can edit the details of existing services. This may involve updating service descriptions, durations, or prices to reflect changes in offerings or business policies.</li><li>• Remove Service: If a service is no longer offered or needs to be removed for any reason, the admin can initiate the removal</li></ul>

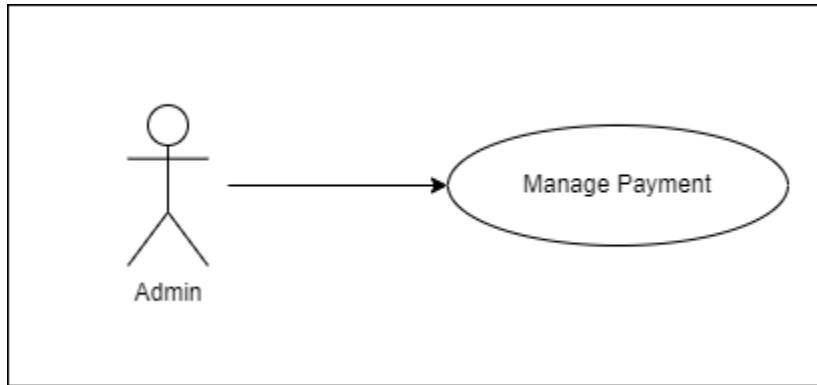
	process. This involves selecting the service to be removed and confirming the action.
Alternative flows:	Confirmation : Before removing a service, the system may prompt the admin with a confirmation dialog to ensure they intend to proceed with the removal.
Post-conditions:	<ul style="list-style-type: none"> <li>• The list of available salon services is updated in the system according to the admin's actions.</li> <li>• Users may see the updated list of services when booking appointments.</li> <li>• If a service is removed, any appointments associated with that service may need to be adjusted or canceled, and affected users may be notified.</li> <li>• The admin has the option to view the updated service list or perform further management tasks as needed.</li> </ul>

[2]



## 2.2.7 USE CASE 07

### MANAGE PAYMENT

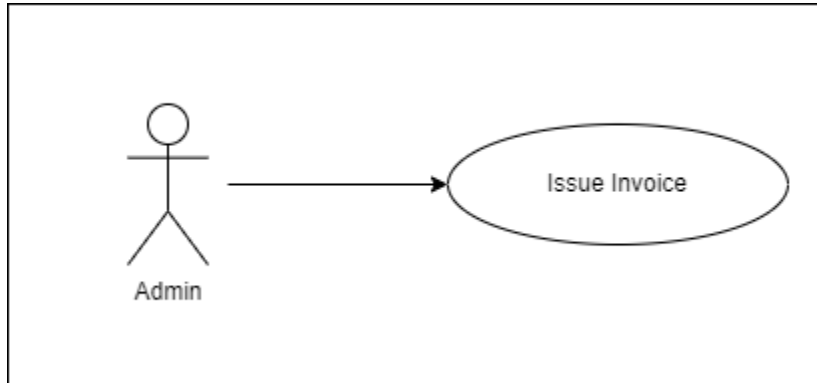


Use Case Name:	Manage payment
Actor:	Admin
Overview:	Admin can mark the payments of users.
Pre-condition:	Admin must log in to the system before manage payments
Description:	<p>After admin log in his dashboard admin can view the user's payment details and can mark it.</p> <ul style="list-style-type: none"><li>• Access Invoice Generation: After logging in, the user navigates to the invoice generation section of the system.</li><li>• View Payment History: The user views their payment history to identify the payment for which they wish to generate an invoice.</li><li>• Select Payment: From the payment history, the user selects the specific payment transaction for which they require an invoice.</li><li>• Initiate Invoice Generation: The user initiates the invoice generation process for the selected payment.</li></ul>
Alternative flows:	<ul style="list-style-type: none"><li>• If the user wishes to customize the invoice with additional details (e.g., billing address, name), the system may provide options for inputting such information.</li></ul>
Post-conditions:	<ul style="list-style-type: none"><li>• After marked payments admin can issue the invoice of payments for the users.</li><li>• The invoice is generated and made available for download or viewing within the system.</li><li>• The user receives confirmation of successful invoice generation.</li></ul>

	<ul style="list-style-type: none"><li>• The invoice may include details such as the service name, service date, payment amount, and any additional information provided by the user.</li><li>• The user has the option to download, print, or email the invoice for their records or further processing as needed.</li></ul>
--	--

## 2.2.8. USE CASE 08

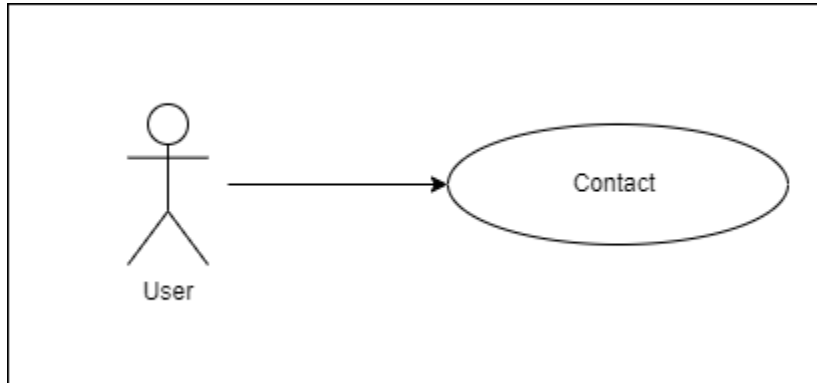
### ISSUE INVOICE



Use Case Name:	Issue Invoice
Actor:	Admin
Overview:	Admin can issue invoice for the users payments
Pre-condition:	Admin must log in to his dashboard before issue invoice.
Description:	After marked the users payment admin can issue the invoice to user and user can view it .
Alternative flows:	If something wrong issueing invoice admin get a error message.
Post-conditions:	<div>After issue invoice admin can do his other manage salon work or log out in system.</div> <ul style="list-style-type: none"><li>• The invoice is generated and made available for download or viewing within the system.</li><li>• The admin receives confirmation of successful invoice issuance.</li><li>• The invoice includes details such as the service name, service date, payment amount, and any additional information provided.</li><li>• The invoice may be sent to the user via email or made available for download from the user's account.</li><li>• The admin has the option to view, download, or print the issued invoice for record-keeping purposes or further processing as needed.</li></ul>

## 2.2.9 USE CASE 09

### CONTACT

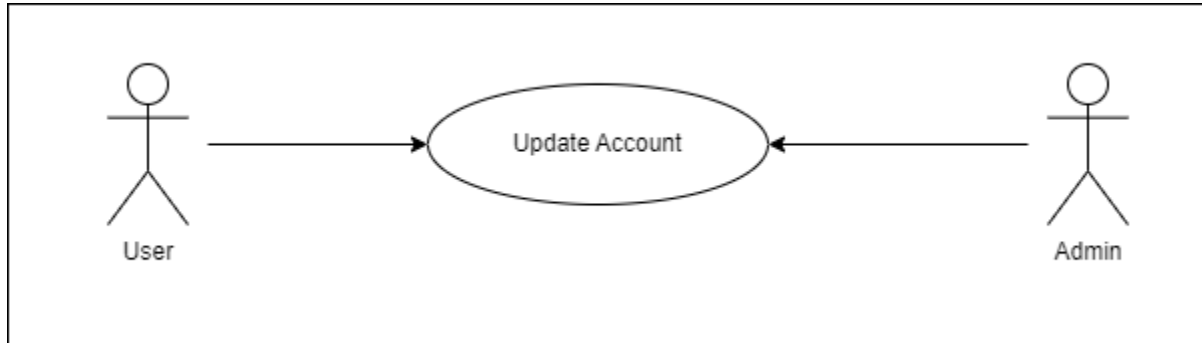


Use Case Name:	Contact
Actor:	User
Overview:	User can contact the admin for if user having problem or something went wrong.
Pre-condition:	<p>User must need to login system before user contact the admin</p> <ul style="list-style-type: none"><li>• Access Messaging Interface: After logging in, the user navigates to the messaging section of the system.</li><li>• Select Admin Contact: The user selects the option to contact the admin from the list of available contacts.</li><li>• Compose Message: The user composes their message to the admin, including the subject and content of the message.</li><li>• Attach Files (Optional): If necessary, the user may attach files or documents to the message to provide additional context or information.</li></ul>
Description:	After login user's account if user have some problem or need to inform something about the service user can contact admin and send a message to admin and admin can review it.
Alternative flows:	If something error system sent a error message to the user.
Post-conditions:	<ul style="list-style-type: none"><li>• After contact admin user can log out the system</li><li>• The message is delivered to the admin's messaging inbox.</li><li>• The admin receives a notification of the new message.</li></ul>

	<ul style="list-style-type: none"><li>• The user may receive a confirmation message indicating that the message has been successfully sent.</li><li>• The user may continue to engage in further communication with the admin if needed.</li><li>• The admin can view and respond to the user's message through the messaging interface.</li></ul>
--	--

## 2.2.10 USE CASE 10

### UPDATE ACCOUNT



Use Case Name:	Update the account
Actor:	01.Admin 02.User
Overview:	Admin and user both of the actors can update their account details .
Pre-condition:	01.Admin : Admin must need to login his admin account before update his account .  02.User : User must need to login his admin account before update his/her account
Description:	01.Admin : After admin log in the his admin dashboard admin can change their name,email or password and reset his account details.  02.User : After User log in the his admin dashboard admin can change their name,email or password and reset his account details.
Alternative flows:	<ul style="list-style-type: none"><li>• Password Change: If updating the password, the system may require the user or admin to enter their current password for security purposes before proceeding with the change.</li><li>• Verification: In some cases, the system may require additional verification steps (e.g., email verification) before allowing certain account updates.</li><li>• Review Changes: The user or admin reviews the changes made to their account information to ensure accuracy and completeness.</li><li>• Save Changes: After confirming the updates, the user or admin saves the changes to their account.</li></ul>
Post-conditions:	01.Admin : After update the admin details admin can save it and admin can do his other manage salon work or log out in system.

	<p>02.User : After upadate the user details admin can save it and admin can do his booking or other needed works or log out in system.</p> <ul style="list-style-type: none"><li>• The updated account information is reflected in the system.</li><li>• If applicable, the user or admin may receive a confirmation message indicating that the changes have been successfully saved.</li><li>• The user or admin can continue using the system with the updated account information.</li></ul>
--	--

## 2.3. NON – FUNCTIONAL REQUIREMENTS

- **Security.**

- The system must be secure from unauthorized access and also this system must be assured of all data inside the system or the system will not be protected against malware attacks.
- A user must remember his or her username or password before logging into this system.
- Another user cannot log into someone else's profile.

- **Hashing Algorithm Implementation:**

All sensitive information such as passwords stored in the salon booking website system database must be securely hashed using a strong cryptographic hashing algorithm. Examples of secure hashing algorithms include MD 5 or bcrypt.

- **Password Storage:**

When a user registers or updates their password, the system hashes the password using the chosen hashing algorithm before storing it in the database. For example, if a user chooses the password "password123", it would be hashed into a string of characters like "3fde34b812e134423c9582d95692d83a".

- **Authentication Process:**

When a user logs in, the system hashes the provided password using the same algorithm and compares it to the hashed password stored in the database. If the hashes match, the user is authenticated and granted access to the system.

- **Protection Against Password Retrieval:**

Hashing ensures that even if the database is compromised, passwords cannot be retrieved in their original form. This provides an additional layer of security, protecting user accounts in the event of a data breach.

Ex:

In the salon booking website system, when a user registers a new account or updates their password, the system securely hashes the password using the MD5 hashing algorithm along with a randomly generated salt. For example, if a user sets their password to "password123", the system generates a unique salt, say "f1e2d3c4", and hashes the combination of the password and salt. The resulting hashed password, along with the salt, is then stored in the database.

Password: password123

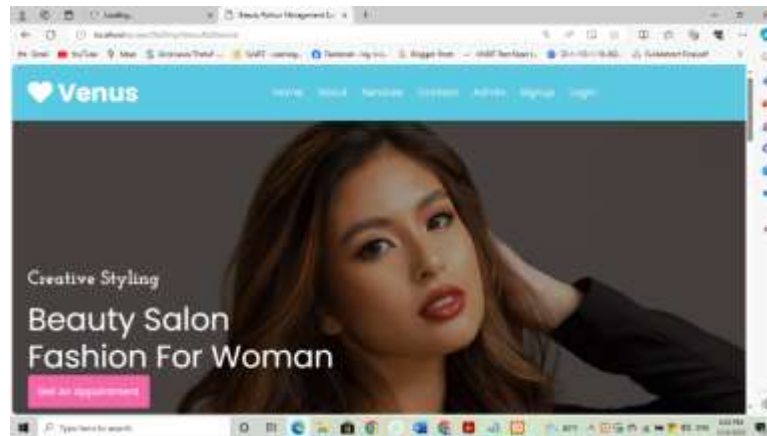
Hashed Password: 2d2abef19a10a6ff80169b051f0027925f33e5a68e7d7d62285dd72e9a3400d3:f1e2d3c4



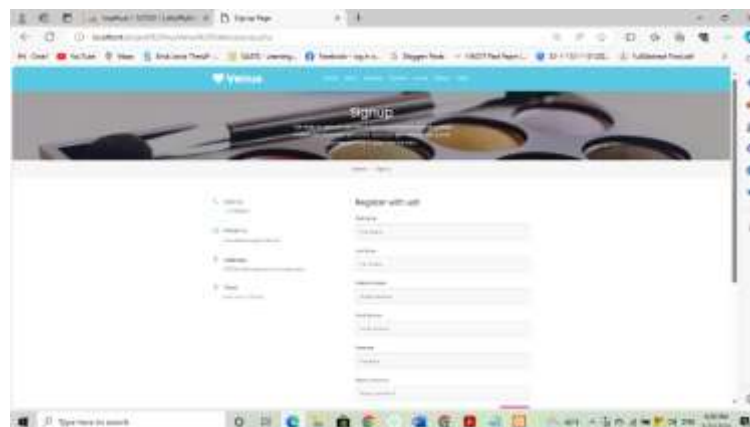
## 2.4 .SYSTEM DESIGN

### 2.4.1 INTERFACE DESIGN FOR USER MODULE

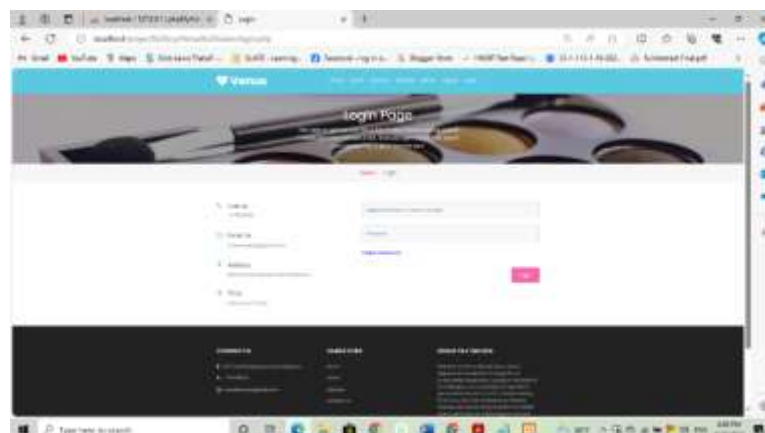
Home page



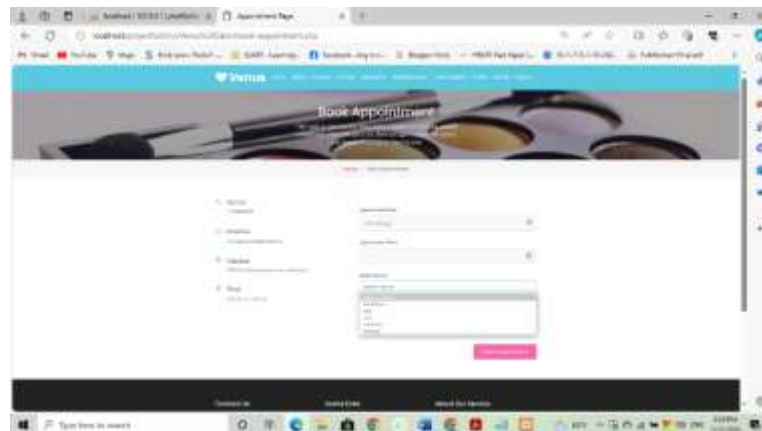
Register page



Login page

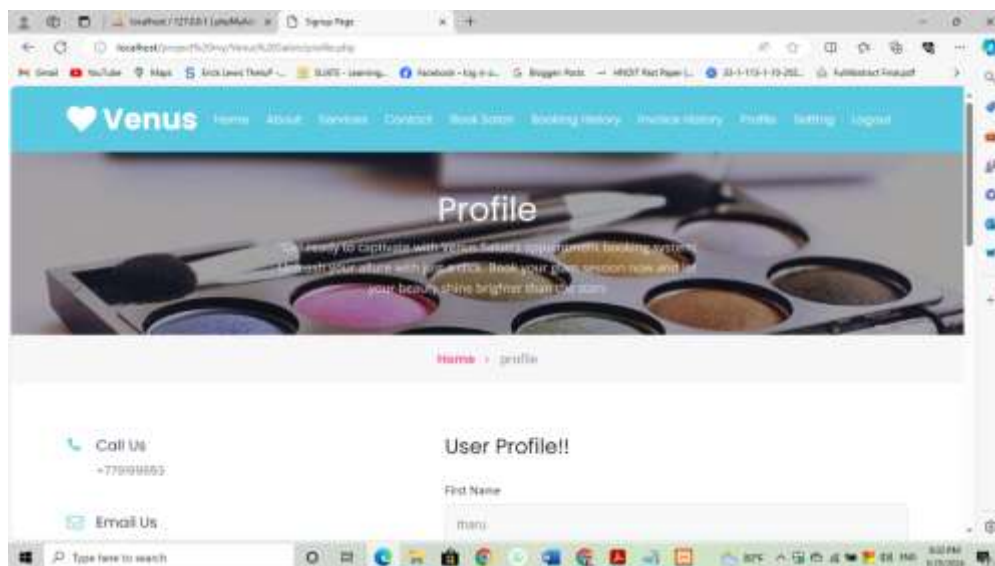


## Book Appointment page



A screenshot of a web browser displaying the 'Book Appointment' page for 'Venus'. The page features a blue header with the 'Venus' logo and navigation links. Below the header is a large image of a makeup palette. The main content area contains a form with the following fields: 'Service' (a dropdown menu), 'Appointment Date' (a date picker), 'Appointment Time' (a time picker), and 'Address' (a text input field). A pink 'Book Appointment' button is located at the bottom right of the form. The browser's address bar shows the URL 'localhost:127.0.0.1:3000/appointment.html'.

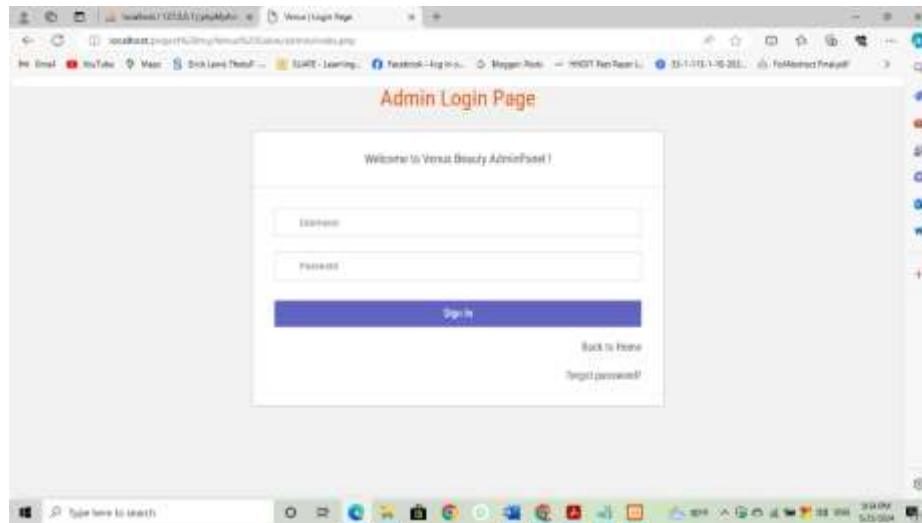
## User account



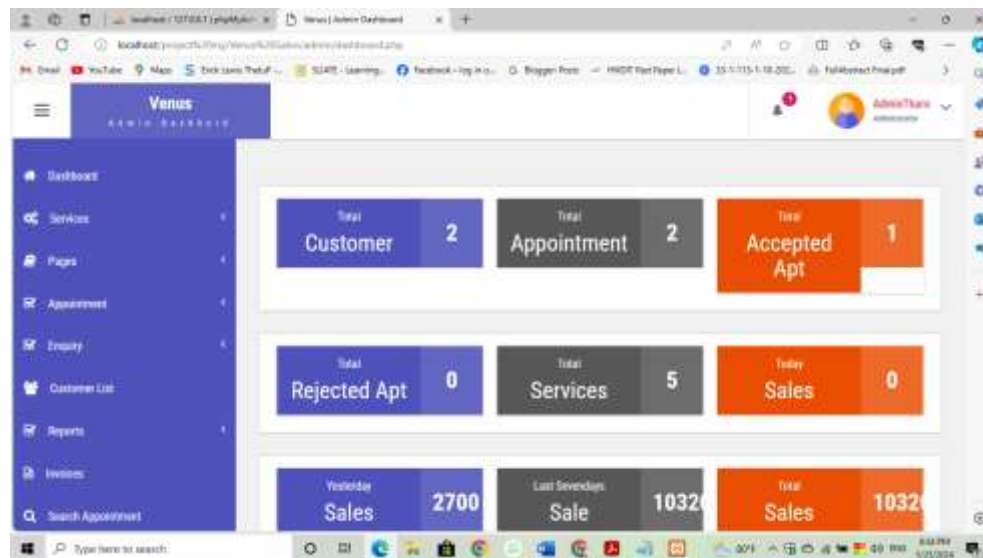
A screenshot of a web browser displaying the 'User Profile' page for 'Venus'. The page features a blue header with the 'Venus' logo and navigation links. Below the header is a large image of a makeup palette. The main content area contains a form with the following fields: 'First Name' (a text input field) and 'Last Name' (a text input field). A pink 'Update Profile' button is located at the bottom right of the form. The browser's address bar shows the URL 'localhost:127.0.0.1:3000/user/profile.html'.

## 2.4.2 INTERFACE DESIGN FOR ADMIN MODULE

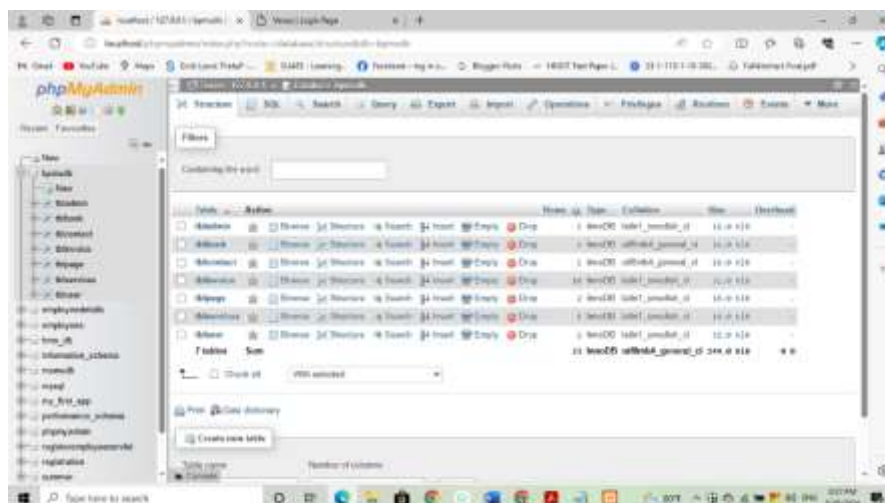
### Admin Login



### Admin dashboard



## 2.4.3 DATABASE DESIGN



## 03. TESTING

### 3.1. TEST CASES

Test Case ID	01	Test Case Description	Test the Register Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	

#### QA Tester's Log

Tester's Name	Tharushi Sewwandi	Date Tested	13-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	
3	
4	

S #	Test Data
1	User Name = Tharu Sewwandi
2	Pass = 123
3	Email = <a href="mailto:t@gmail.com">t@gmail.com</a>
4	T.P = 783678399

<b>Test Scenario</b>	Create new account on entering username , T.P Number valid email and password, then user can register salon system and create new account
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username,Email,T.P Number & Password	User can be entered	As Expected	Pass
3	Click Submit	User can create account successfully.	As Expecte	Pass

Test Case ID	02	Test Case Description	Test the Login Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	Windows 10

QA Tester’s Log

Tester's Name	Tharushi Sewwandi	Date Tested	15-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	
3	
4	

S #	Test Data
1	User Name = Tharu
2	Pass = 123
3	UserName= asha
4	

**Test Scenario**      Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password	User can be entered	As Expected	Pass
3	Click Submit	User is logged in dashboard	As Expected	Pass
4	If enter invalid details	Invalid login	Login fail	Pass

Test Case ID	03	Test Case Description	Test the Admin Salon System		
Created By	ANU/11	Reviewed By	K.D.A.Hettige	Version	

**QA Tester's Log**

Tester's Name	Tharushi Sewwandi	Date Tested	13-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	
3	
4	

S #	Test Data
1	Username for admin = admin
2	Password = test@123
3	
4	

**Test Scenario**      Verify on entering valid admin username and password, the admin can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password	Admin can be entered	As Expected	Pass
3	Click Submit	Log in to the admin dashboard.	As Expected	Pass
4				

Test Case ID	04	Test Case Description	Test the Book appointment Salon System		
Created By	ANU/11	Reviewed By	K.D.A.Hettige	Version	

QA Tester's Log

Tester's Name	Tharushi Sewwandi	Date Tested	13-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Login user account
3	
4	

S #	Test Data
1	Username = tharu
2	Appointment date & Time
3	Need service
4	

Test Scenario	Book the appointment and submit it
---------------	------------------------------------

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Email Number & Password	User can be entered	As Expected	Pass
3	Click Submit	Log in to the dashboard.	As Expected	Pass
4	Open appointment booking page	Book appointment	As Expected	Pass



Test Case ID	05	Test Case Description	Test the cancel appointment Salon System		
Created By	ANU/11	Reviewed By	K.D.A.Hettige	Version	

QA Tester’s Log

Tester's Name	Tharushi Sewwandi	Date Tested	13-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Login user account
3	Book appointment
4	cancel

S #	Test Data
1	Username = tharu
2	Appointment date & Time
3	Need service
4	

Test Scenario

Book the appointment and submit it

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Email Number & Password	User can be entered	As Expected	Pass
3	Click Submit	Log in to the dashboard.	As Expected	Pass
4	Open appointment booking page	Book appointment	As Expected	Pass
5	Click cancel appointment	Cancel the booked appointment	As Expected	

Test Case ID	06	Test Case Description	Test the Payment Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	

QA Tester’s Log

Tester's Name	Tharushi Sewwandi	Date Tested	15-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Login Account
3	Book appointment
4	View invoice

S #	Test Data
1	Download invoice
2	
3	
4	

Test Scenario

Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Log in account	User can be entered	As Expected	Pass
3	Book appointment	User need to book appointment	As Expected	Pass
4	Wait for accept appointment after accept it can download /view invoice	View cost of the service	As Expected	Pass

Test Case ID	07	Test Case Description	Test the Manage users in Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	

**QA Tester’s Log**

Tester's Name	Tharushi Sewwandi	Date Tested	15-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Log the admin dashboard
3	
4	

S #	Test Data
1	Log admin account
2	
3	
4	

**Test Scenario**

Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password of admin account	User can be entered	As Expected	Pass
3	Click Submit	User is logged in dashboard	As Expected	Pass
4	Manage the customer	Can see the customer name and details	As Expected	Pass

Test Case ID	08	Test Case Description	Test the manage appointment Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	

QA Tester’s Log

Tester's Name	Tharushi Sewwandi	Date Tested	15-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Log admin account
3	
4	

S #	Test Data
1	Booked appointment
2	
3	
4	

Test Scenario

Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password admin account	admin can be entered	As Expected	Pass
3	Click Submit	admin is logged in dashboard	As Expected	Pass
4	Accept or reject the appointment	Admin can accept or reject the customer requesting booking	As Expected	Pass

Test Case ID	09	Test Case Description	Test the Manage services in Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	

**QA Tester's Log**

Tester's Name	Tharushi Sewwandi	Date Tested	15-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Log in to the admin account
3	
4	

S #	Test Data
1	
2	
3	
4	

**Test Scenario**    Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password	Admin can be entered	As Expected	Pass
3	Click Submit	Admin is logged in dashboard	As Expected	Pass
4	Add or remove service	Admin can add new service or remove or update services	As Expected	Pass

<b>Test Case ID</b>	10	<b>Test Case Description</b>	Test the payment manage Salon System		
<b>Created By</b>	ANU11	<b>Reviewed By</b>	K.D.A.Hettige	<b>Version</b>	

#### QA Tester's Log

<b>Tester's Name</b>	Tharushi Sewwandi	<b>Date Tested</b>	15-May-2024	<b>Test Case (Pass/Fail/Not Executed)</b>	Pass
----------------------	-------------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Access to Web application
2	Log in admin account
3	
4	

S #	Test Data
1	Payment details
2	
3	
4	

#### Test Scenario

Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password	Admin can be entered	As Expected	Pass
3	Click Submit	Admin is logged in dashboard	As Expected	Pass
4	Manage payments	View bookings and create the invoices for that and sent it to customer	As Expected	Pass

Test Case ID	11	Test Case Description	Test the contact us in Salon System		
Created By	ANU11	Reviewed By	K.D.A.Hettige	Version	

QA Tester’s Log

Tester's Name	Tharushi Sewwandi	Date Tested	15-May-2024	Test Case (Pass/Fail/Not Executed)	Pass
---------------	-------------------	-------------	-------------	------------------------------------	------

S #	Prerequisites:
1	Access to Web application
2	Log in user account
3	
4	

S #	Test Data
1	User Name = user name
2	E <a href="mailto:t@gmail.com">mail=t@gmail.com</a>
3	Message = grate service
4	

Test Scenario

Verify on entering valid username and password, the user can login

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Open application	Site should open	As Expected	Pass
2	Enter Username & Password	User can be entered	As Expected	Pass
3	Click Submit	User is logged in dashboard	As Expected	Pass
4	Load contact us page	User can give feedback about service	As Expected	Pass

## 04. IMPLEMENTATION

### 4.1 INSTALLATION GUIDE

- Download and install Sublime Text for code editing.
- Install XAMPP to set up a local server environment, including Apache, MySQL, and PHP.
- Configure XAMPP and start Apache and MySQL services.
- Download and install phpMyAdmin for MySQL database management.
- Clone or download the project files from the repository.
- Place the project files in the "htdocs" directory of XAMPP.
- Import the MySQL database using phpMyAdmin.
- Update database connection settings in the PHP files if necessary.
- Access the web application by navigating to "http://localhost/my\_project" in a web browser.

### 4.2 USER GUIDE

- Registration: Users can create an account by providing required details.
- Login: Existing users can log in using their credentials.
- Booking Appointment: Users can select a date and time from available slots and book an appointment for salon services.
- Viewing Booking Details: Users can view their booked appointments along with details like date, time, and selected services.
- Admin Management: Admins can log in, manage salon services, view appointment requests, and accept or decline bookings.

### 4.3 BACKUP PROCEDURES/CYCLES

- Perform regular backups of the MySQL database using phpMyAdmin.
- Schedule backups on a weekly basis, ensuring that all critical data is preserved.
- Store backups in a secure location, either locally or on cloud storage.
- Test backup restoration procedures periodically to verify data integrity and availability.
- Document backup procedures and store them in a secure location for reference.



## 4.4 SECURITY PROCEDURES

- Enforce secure password policies for user accounts to prevent unauthorized access.
- Sanitize user inputs to prevent SQL injection and other forms of attacks.
- Apply role-based access control to restrict access to sensitive functionalities.
- Regularly update software components (e.g., PHP, Apache, MySQL) to patch security vulnerabilities.
- Monitor server logs for suspicious activities and implement intrusion detection mechanisms.
- Educate users and administrators about security best practices, such as avoiding phishing scams and using strong passwords. [3]

## 05 EVALUATION & CONCLUSION

The salon appointment booking web application has undergone a comprehensive evaluation across various dimensions, yielding valuable insights into its performance and potential for future development. Firstly, in terms of the degree of objectives met, the project has largely succeeded in its primary goals. Users can register, login, and book appointments seamlessly, while administrators can efficiently manage salon services and appointment schedules. However, the absence of integrated payment functionality remains a notable objective yet to be fully realized.

Secondly, the application demonstrates commendable usability, accessibility, reliability, and friendliness. Its intuitive interface facilitates easy navigation and booking for users of diverse backgrounds and abilities. The system's reliability is upheld by stable performance and quick response times, contributing to a positive user experience. While efforts were made to ensure accessibility, ongoing refinements may further enhance inclusivity for all users.

Thirdly, user response has been generally positive, with clients appreciating the convenience and efficiency offered by the platform. Feedback indicates satisfaction with the booking process, although there is a desire for additional features such as integrated payment options to streamline transactions further. Nonetheless, the increase in appointment bookings and positive engagement metrics underscore the platform's value to users. [4]

Despite its strengths, the project is not without limitations and drawbacks. The absence of integrated payment functionality presents a notable barrier to seamless transaction processing within the platform. Additionally, while efforts were made to ensure accessibility, further improvements may be necessary to address the needs of users with disabilities fully.

Looking ahead, several opportunities for future modifications, improvements, and extensions exist. Integration of a payment gateway API stands out as a key enhancement to streamline transaction processing and enhance user satisfaction. Furthermore, personalized features such as appointment reminders and notifications could further enhance the user experience and foster client loyalty. Ongoing usability testing and accessibility audits will be essential to identify areas for refinement and ensure the platform's continued success in meeting the evolving needs of users.

Moving forward, integrating a payment gateway API stands as a key improvement to the system. This enhancement would allow users to make secure online payments, improving the overall user experience and streamlining the booking process. Additionally, further usability testing and accessibility audits could identify areas for refinement, ensuring a seamless experience for all users. Furthermore, exploring options for mobile optimization and implementing additional features such as appointment reminders or loyalty programs could further enhance the application's value proposition.s

## 06.REFERENCES

- [1] "slideshare," [Online]. Available: <https://www.slideshare.net/slideshow/clien-trak-spa-and-salon-software/3508092>. [Accessed 25 05 2024].
- [2] Adarsh Kumar Gulshan, Amir Khan, "Salon Management System," *IJCRT*, vol. 9, no. 5 may 2021, p. 6, 2021.
- [3] W. Silva, "Salon Management System," University of Colombo School of Computing, Colombo, 2018.
- [4] "Leprechaun," Leprechaun Saloon Software,, [Online]. Available: <https://leprechaun-software.blogspot.com/>. [Accessed 25 05 2024].