CHAPTER 1

ABOUT COMPANY

As part of the course in Computer Science and Engineering degree, prescribed by Visvesvaraya Technological University, an internship is undertaken. The details of thecompany that provided the internship are given.

1.1 About the Company

Company name: Saunshi Software Services (LLP)

Established: 2006

It is a digital branding and web development company based In hubli,.

As a company it assures time bound service to our customers and support them in developing winning strategies for their business through innovative and latest technologies at very affordable cost.

They provide seamless service by controlling cost and maximizing profits at the same time by keeping the customers happy .Saunshi software services helps you put processes in place no matter how small the business that can reduce service time and even give full patient/ customer history in a click. Saunshi software make applications that are truly personalized like using messaging features for reminders, thank you messages and many more. Pain points and challenges are isolated by working with Industry experts .Saunshi software services will start saving from the first month itself and will cover its cost in the first three months at same time enabling you to grow and improve your bottom line considerably.

Aim of the company

To provide branding services, social media marketing, web design and web development

1.2 Domains handled by the company

Company in the midst of a significant transformation regarding the way we produce products thanks to the digitization of manufacturing.. Company under various department the technologies of Industry 4.0 include Integrated System, IOT, Data Management, Cloud

Computing, Cyber security, Artificial Intelligence, Additive Manufacturing, Augmented Reality.

The company also deals with inventory and billing that is complete automation of stock management and invoicing process with high level of accuracy and speed. It offers real time update of inventory which will help in maintain optimal inventory.

1.3 Objectives

The objective of the web design is to handle the entire design of a website. The software keeps track of all the information about the entire website. The system contains database where all the information will be stored safely.

1.3.1 To gain skills and knowledge

This internship provided us essential skills and knowledge one requires in the field of web designing. The crucial tools used during the tenure helped us in gaining knowledge about programming languages.

1.3.2 To get field work experience

By taking this training we enhanced our knowledge in Web designing and got insight in how the websites are designed using HTML and CSS.

1.3.3 To enhance our communication skills

By interacting with my trainee and classmates I got to learn a lot. It helped me to enhance my communicative skills and represent my work with confidence. It boosted my confidence to design more webpages and create some great designs just for fun.

1.3.4 To link theory with practice

First we learned the theory aspect and then we put that into practice. By doing the practical work, our concept got clearer and it was easy to code into HTML once we got familiar with it. By putting our theoretical knowledge into practical, coding became more fun.

1.3.5 To build a network

By learning how to code in HTML and CSS it is easy to build and design our own websites with all the changes we want. We also got to know about how websites are designed initially and the logic behind that.

CHAPTER 02

TASKS PERFORMED

The weekly tasks performed have been explained in detail below which gives the overview of several concepts undertaken during the internship.
The proposed system will provide manage day to day salon process easily.

Manage branch details :- Salon owner can manage his salons in one location. Each branches have separate login and separate branch account. Salon owner and SBP admin can generate reports with all possible areas according branches and as all in one branches. Also can manage branch properties add employee / add service etc.

Manage service details :- Every salon has offer veracity of services related to beauty culture. Salon has to maintain a service list with the price and service details. It's also help full for salon employees to deal with customers. Especially when creating bill. To full fill this manage service module will be added to the system.

Manage bookings by salon :- Salon's customers can directly call to salon and salon admin should create booking for that customer with specific service and time. Additionally booking can assign specific employee.

Manage invoice :- Customers had to pay full billed amount when the service received. In this case the salon front desk should be able to generate bill for customers. And the payment details for the bill should be recorded in to the system.

Generate reports :- System reports are most important for owner to get valuable decisions and prediction in management. Many types of reports are allowed in system. Daily summary of payment details, summary of booking, monthly summary of them, and most requested reports types by salon owner, and so on.

Manage customer details :- Their customers are mostly regular basis. So they are planning to keep historical data to evaluate who got service and product from their salons. For this they are need to keep customers details against the issued reports historical data.

Manage product details :- Each day salon expends lot of products and equipment in their activities. So it should be manage to prevent run out stocks. General reorder points for those items set into system and system will manage message to indicate before stock ran out.

2.1 WEEK 1: HTML, CSS and BOOTSTRAP

In the first week we learnt about basics of HTML, various HTML tags, syntax of CSS, selectors like id, class, attribute selectors and their uses, advantages and disadvantages of CSS and why to use Bootstrap and applications of bootstrap.

2.1.1 HTML

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages. Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext. As its name suggests, HTML is a Markup Language which means we use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. HTML is being widely used to format web pages with the help of different tags.

HTML Tags

HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces <Tag Name>. Except few tags, most of the tags have their corresponding closing tags. For example, <html> has its closing tag</html> and <body> tag has its closing tag </body> tag etc.

Table 2.1 HTML tags and their description

Tag	Description
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, , <p> etc.
<h1>	This tag represents the heading.
<p>	This tag represents a paragraph

2.1.2 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects. CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- CSS saves time - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.
- Pages load faster - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.

- Easy maintenance - To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.
- Superior styles to HTML - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Multiple Device Compatibility - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cellphones or for printing.
- Global web standards – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

2.1.3 BOOTSTRAP

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS, and Javascript. Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.

Why Use Bootstrap?

- Mobile first approach: Bootstrap 3 framework consists of Mobile first styles throughout the entire library instead of them in separate files.
- Browser Support: It is supported by all popular browsers.
- Easy to get started: With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
- Responsive design: Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter Bootstrap Responsive Design.
- Provides a clean and uniform solution for building an interface for developers.
- It contains beautiful and functional built-in components which are easy to customize.
- It also provides web-based customization and best of all it is an open source.

What Bootstrap Package Includes?

- Scaffolding: Bootstrap provides a basic structure with Grid System, link styles, and background.

- CSS: Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system.
- Components: Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more.
- JavaScript Plugins: Bootstrap contains over a dozen custom jQuery plugins.
- Customize: You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

2.2 WEEK 2: Agile Methodology , GitHub

2.2.1 Agile Methodology

Different types of methodologies in software engineering can significantly eliminate the time, resources, and costs needed for project realization. However, due to the vast number of these methodologies currently existing, it becomes more and more challenging to pick up the best model for each specific case.

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

The values behind the Agile Manifesto.

The four Agile Manifesto values are:

1. Individuals and interactions over processes and tools.

It is people who drive the development process and respond to business needs on the fly, so they take precedence over processes and tools. If the latter things drive development, the team becomes less responsive and able to meet customer needs.

2. Working software over comprehensive documentation.

The Agile Manifesto deemphasizes the documentation of the development process, which historically took a huge amount of time and often bogged down the team. By avoiding unimportant minutiae, Agile shifts the team’s focus from the process itself to the results of the process (actual working software).

3. Customer collaboration over contract negotiation.

In traditional project management methods, such as Waterfall, customers negotiate the product requirements in great detail before any work starts, and they're typically involved only at the beginning and at the end. In Agile, the customer becomes an important collaborator throughout the development process, ensuring their input is incorporated, and the result meets their needs along the way.

4. Responding to change over following a plan.

Because ongoing adaptation isn't built into the traditional software development process, change is an expensive headache. Agile embraces change, focusing on releasing a minimum viable product that can be evaluated and adjusted from iteration to iteration.

2.2.2 Git and GitHub

Git is a version control system which lets you track changes you make to your files over time. With Git, you can revert to various states of your files (like a time traveling machine). Yocan also make a copy of your file, make changes to that copy, and then merge these changes to the original copy.

What is GitHub?

GitHub is an online hosting service for Git repositories. Imagine working on a project at home and while you are away, maybe at a friend's place, you suddenly remember the solution to a code error that has kept you restless for days.

How to push a repository to GitHub

Step 1 – Create a GitHub account: To be able to use GitHub, you will have to create an account first. You can do that on their website.

Step 2 – Create a repository: You can click on the + symbol on the top right corner of the page then choose "New repository". Give your repo a name then scroll down and click on "Create repository".

Step 3 – Add and commit file(s): Before we "add" and "commit" our files, you need to understand the stages of a file being tracked by Git.

Committed state: A file is in the committed state when all the changes made to the file have been saved in the local repo. Files in the committed stage are files ready to be pushed to the remote repo (on GitHub).

Modified state: A file in the modified state has some changes made to it but it's not yet saved. This means that the state of the file has been altered from its previous state in the committed state.

Staged state: A file in the staged state means it is ready to be committed. In this state, all necessary changes have been made so the next step is to move the file to the commit state.

How to add files in Git

When we first initialized our project, the file was not being tracked by Git. To do that, we use this command git add . The period or dot that comes after add means all the files that exist in the repository. If you want to add a specific file, maybe one named about.txt, you use git add about.txt. Now our file is in the staged state. You will not get a response after this command, but to know what state your file is in, you can run the git status command.

How to commit files in Git

The next state for a file after the staged state is the committed state. To commit our file, we use the git commit -m "first commit" command. The first part of the command git commit tells Git that all the files staged are ready to be committed so it is time to take a snapshot. The second part -m "first commit" is the commit message. -m is shorthand for message while the text inside the parenthesis is the commit message.

Step 4 – Push the repository to GitHub: After you create the repo, you should be redirected to a page that tells you how to create a repo locally or push an existing one.

2.3 WEEK 3: JavaScript, PHP

2.3.1 JavaSript

JavaScript is a programming language initially designed to interact with elements of web pages. In web browsers, JavaScript consists of three main parts:

- ECMAScript provides the core functionality.

- The Document Object Model (DOM) provides interfaces for interacting with elements on web pages
- The Browser Object Model (BOM) provides the browser API for interacting with the web browser.

JavaScript allows you to add interactivity to a web page. Typically, you use JavaScript with HTML and CSS to enhance a web page's functionality, such as validating forms, creating interactive maps, and displaying animated charts.When a web page is loaded, i.e., after HTML and CSS have been downloaded, the JavaScript engine in the web browser executes the JavaScript code. The JavaScript code then modifies the HTML and CSS to update the user interface dynamically. The JavaScript engine is a program that executes JavaScript code. In the beginning, JavaScript engines were implemented as interpreters.

Client-side vs. Server-side JavaScript

When JavaScript is used on a web page, it is executed in web browsers. In this case, JavaScript works as a client-side language. JavaScript can run on both web browsers and servers. A popular JavaScript server-side environment is Node.js. Unlike client-side JavaScript, server-side JavaScript executes on the server that allows you to access databases, file systems, etc.

Attribute	Description
Offline	Triggers when the document goes offline
Onabort	Triggers on an abort event
onafterprint	Triggers after the document is printed
onbeforeonload	Triggers before the document loads
onbeforeprint	Triggers before the document is printed

onblur	Triggers when the window loses focus
oncanplay	Triggers when media can start play, but might has to stop for buffering
oncanplaythrough	Triggers when media can be played to the end, without stopping for buffering
onchange	Triggers when an element changes
onclick	Triggers on a mouse click
oncontextmenu	Triggers when a context menu is triggered
ondblclick	Triggers on a mouse double-click
ondrag	Triggers when an element is dragged
ondragend	Triggers at the end of a drag operation
ondragenter	Triggers when an element has been dragged to a valid drop target
ondragleave	Triggers when an element is being dragged over a valid drop target

Table 2.3 JavaScript events and their description

Applications of JavaScript:

- Web Development: Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.
- Web Applications: With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming

Interfaces(APIs) that provide extra power to the code. The Electron and React is helpful in this department.

- Server Applications: With the help of Node.js, JavaScript made its way from client to server and node.js is the most powerful on the server-side.
- Games: Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.
- Smartwatches: JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the internet for its functioning.

Limitations of JavaScript:

- Security risks: JavaScript can be used to fetch data using AJAX or by manipulating tags that load data such as <img>, <object>, <script>. These attacks are called cross site script attacks. They inject JS that is not the part of the site into the visitor's browser thus fetching the details.
- Performance: JavaScript does not provide the same level of performance as offered by many traditional languages as a complex program written in JavaScript would be comparatively slow. But as JavaScript is used to perform simple tasks in a browser, so performance is not considered a big restriction in its use.
- Complexity: To master a scripting language, programmers must have a thorough knowledge of all the programming concepts, core language objects, client and server-side objects otherwise it would be difficult for them to write advanced scripts using JavaScript.

2.3.2 PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module

on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

Advantages of php

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them. The other uses of PHP are:

- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- You add, delete, modify elements within your database thru PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

2.4 WEEK 4: Designing the website(Frontend)

The goal of front-end development is to ensure the best possible user experience which includes efficiency, speed, and smooth functionality. In fact, a front-end developer acts as a mediator between a designer and a back-end developer.

Moreover, the tools and methodologies used for front-end development are continuously evolving and the developers need to constantly upgrade their skills. In this article, we will answer the question of why the quality of the front-end is critical and how the front-end development process should look like. This might help you choose the right front-end team.

By combining the UI, development, and analytics, businesses can deliver better experience from their customers' satisfaction standpoint:

- Performance

  Slow applications frustrate users and make them search for alternatives. A faster response leads to higher conversion rates.

- Business Intent

  The app should convey the message the company wants to communicate. The design, graphics, and content should provide answers to users' questions and encourage them to use the app more.

- Navigation

Intuitive navigation ensures that users find what they are looking for. Users need to be able to see clearly what their next steps can be and how to do them.

- Visitor Retention

  Well-designed interfaces lead to higher conversion rates. Sluggish apps with poor design may alienate users, and they will never come back.

Nevertheless, any front-end development process includes a few basic steps

Step 1. Requirement Gatherin

As in other spheres of software development, the front-end development process begins with collecting and defining business requirements. Gathering requirements for the project is the most important part. During this stage, the product owner describes the expectations of the project:

- what the goal of the project is;
- who the target audience is;
- how they will use the product.

It's important for the development team to understand the needs of the client because this information is critical for addressing the product owner's requests.After the client provides requirements for the product, the project managers and the development team start analyzing the requirements. This step is critical for minimizing the chance of a mismatch between customer expectations and the final result.

After project functionality and features are drafted in the software requirements specification document, the development team proceeds to the next stage.

Step 2. Creating Prototype

Once the requirements are collected, the developers create a prototype that is presented to the client for evaluation. A prototype is an early version of the future application. It demonstrates a basic idea of what the app will look like and how it will work.

After receiving client feedback, the development team modifies the prototype and presents it to the client again for evaluation. Once the client approves the prototype, it is used as a requirement for building the actual application.

Step 3. Development

At the development stage, all the requirements from the previous phases are transformed into the actual system. Developers define the most relevant front-end frameworks, tools, and the best development practices to ensure that the product is developed in the most efficient way. They purchase and install the respective software and hardware and create the actual code on the basis of given specifications. At this stage the frond-developer:

- develops user-facing features;
- builds reusable code for future use;

- ensures the technical feasibility of UI/UX design;
- optimizes application for maximum speed and scalability;
- assures that all user input is validated before submitting to the back-end;
- collaborates with other team members and the client.

When the development process is over, the client makes a final review.

Step 4. QA and Testing

Testing is normally carried out by Quality Assurance specialists. Once the development process is complete, the app is deployed in the testing environment and the QA specialists start testing the functionality of the app. Testing is performed to ensure that the app works according to the specified requirements.

You may wonder how this step is related to front-end development. The thing is that once the QA team finds some bugs they report on the problems to developers. The developers fix the bug and send the app back to the QA team for retesting. This procedure continues until the app is stable and works flawlessly.

Step 5. Maintenance and Support

The development doesn't end with the app rollout. Bugs may appear at any time so the development team should constantly monitor the app performance to ensure it functions properly. The main challenge here is to ensure that code enhancement doesn't cause any other malfunctions.

2.5 WEEK 5: MySQL

What is a Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds. Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory, but data fetching and writing would not be so fast and easy with those type of systems. Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as Foreign Keys.

MySQL – Introduction

- Compound Key: A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
- Index: An index in a database resembles an index at the back of a book.
- Referential Integrity: Referential Integrity makes sure that a foreign key value always points to an existing row.

MySQL Database

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). MySQL is customizable.

Administrative MySQL Command

Here is the list of the important MySQL commands, which you will use time to time to work with MySQL database:

- USE Databasename: This will be used to select a database in the MySQL workarea.
- SHOW DATABASES: Lists out the databases that are accessible by the MySQL DBMS.
- SHOW TABLES: Shows the tables in the database once a database has been selected with the use command.
- SHOW COLUMNS FROM tablename: Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.

- SHOW INDEX FROM tablename: Presents the details of all indexes on the table, including the PRIMARY KEY.
- SHOW TABLE STATUS LIKE tablename\G: Reports details of the MySQL DBMS performance and statistics.

2.6 WEEK 6: Making web application responsive

Responsive web design refers to a design strategy that creates websites that work well for mobile, tablet, and desktop devices. Websites without responsive design risk alienating a significant number of users.

How to create a Responsive Website

1. Set Appropriate Responsive Breakpoints

In responsive design, a breakpoint is the “point” at which a website’s content and design will adapt in a certain way in order to provide the best possible user experience.

Every website is accessed via devices with different screen sizes and resolutions. The software has to render perfectly across each screen size. Content or images cannot be distorted, cut out, or obscured.

To allow this, developers have to use responsive breakpoints, sometimes called CSS breakpoints or media query breakpoints. These are points defined in the code. Website content responds to these points and adjusts itself to the screen size to display the accurate layout.

Use breakpoints for the most commonly used device resolutions used across mobile, desktop, and tablet. These would be:

1920×1080 (9.61%)

1366×768 (7.87%)

360×640 (4.36%)

414×896 (4.34%)

1536×864 (4.11%)

2. Start with a Fluid Grid

Previously, websites were based on pixel measurements. Now, however, they are built on what is called a fluid grid.Basically, a fluid grid positions and sets web elements on a site in proportion to the screen size it is displayed on. Instead of making things in a single, specific size set in pixels, elements on a fluid grid will respond and resize to fit the size of the screen.A fluid grid is divided into columns; heights and widths are scaled, not set to fixed dimensions. The proportions of text and elements depend on the screen size.

3. Take touchscreens into consideration

When wondering how to make a website responsive, think of touchscreens. Most mobile devices (phones and tablets) are now equipped with touchscreens. Some laptops are also catching up, offering touchscreen along with the keyboard functions.

Naturally, a responsive website will have to calibrate itself for being accessed via touchscreens. For example, let's say there is a drop-down menu on the homepage.On desktop view, each menu item must be large enough so that it can be pressed with a fingertip of a touchscreen.On mobile screens, smaller elements like buttons should also be easier to detect and select.

Responsive Image Code breakdown:

- Setting max-width allows the image to adjust its size based on its container width.
- picture, source, and img tags are combined so that only one image is rendered, and that it fits best on the user's device.
- source is used to reference a WebP image that can be used by browsers supporting it. A second source tag references a PNG file of the same image for browsers without WebP support.WebP is an image format with advanced compression for web-based images.
- srcset notifies the browser about which image should be displayed, depending on the particular device's screen resolution.
- loading="lazy" attribute / value pair: Implements native lazy loading.

4. Define Typography

Generally, web developers define font sizes with pixels. These work on static websites, but responsive websites need a responsive font. Font size must change with respect to parent

container width. This is necessary to make typography adjust to screen size and be easily readable on multiple devices.

In the CSS3 specification, look for the unit named rems. It is similar to the em unit but acts relative to the HTML element. Because of this, the code must reset the HTML font-size:

```
html { font-size:100%; }
```

5. Use a pre-designed theme or layout to save time

If developers and designers are wondering how to build a responsive website on an exceptionally tight deadline, they can opt for using a theme or pre-designed layout with built-in responsive properties. WordPress provides multiple options in this regard (both free and paid). All designers have to do, after picking a theme, is to decide on colour, branding, and text.

2.7 WEEK 7: Testing and debugging

At the time of development and after the outcome of any application or the software product established in any programming language, both Testing and Debugging play a vital role in finding and removing mistakes.

2.7.1 Testing

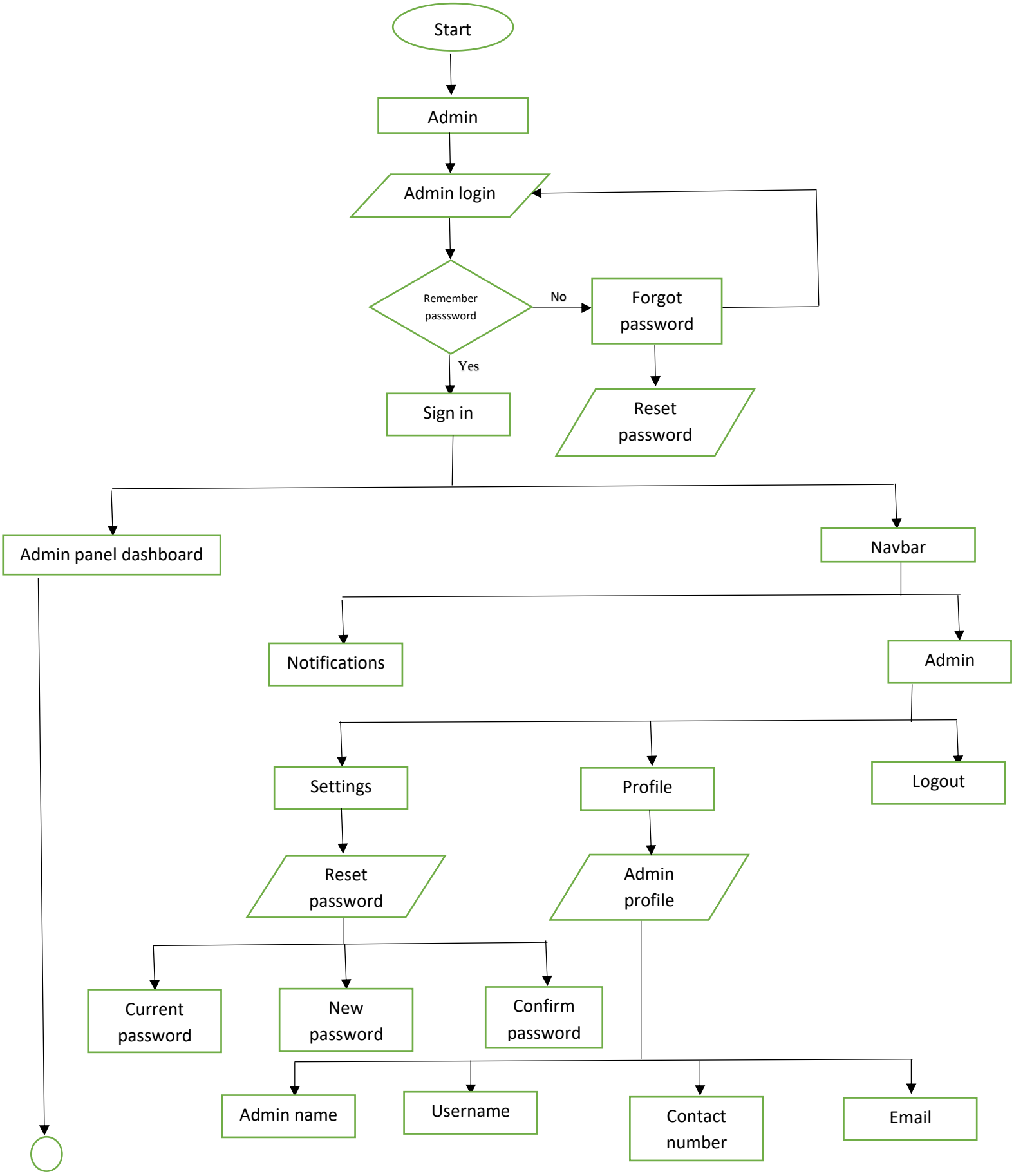
Software testing is a process of identifying defects in the software product. It is performed to validate the behavior of the software or the application compared to requirements.

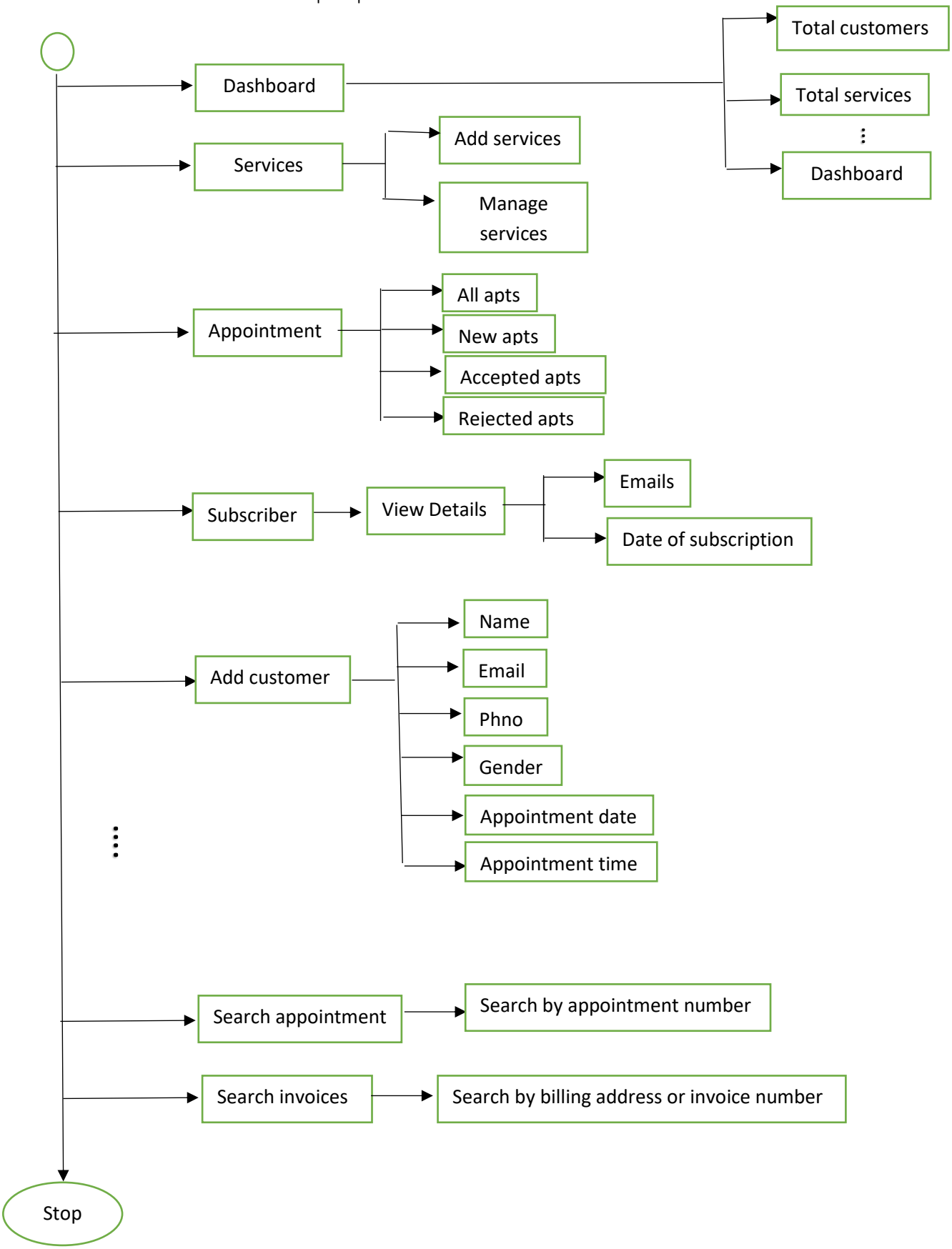
In other words, we can say that the testing is a collection of techniques to determine the accuracy of the application under the predefined specification but, it cannot identify all the defects of the software. Each software or application needs to be tested before delivering to the clients and checks whether the particular software or the application is working fine as per the given requirements.

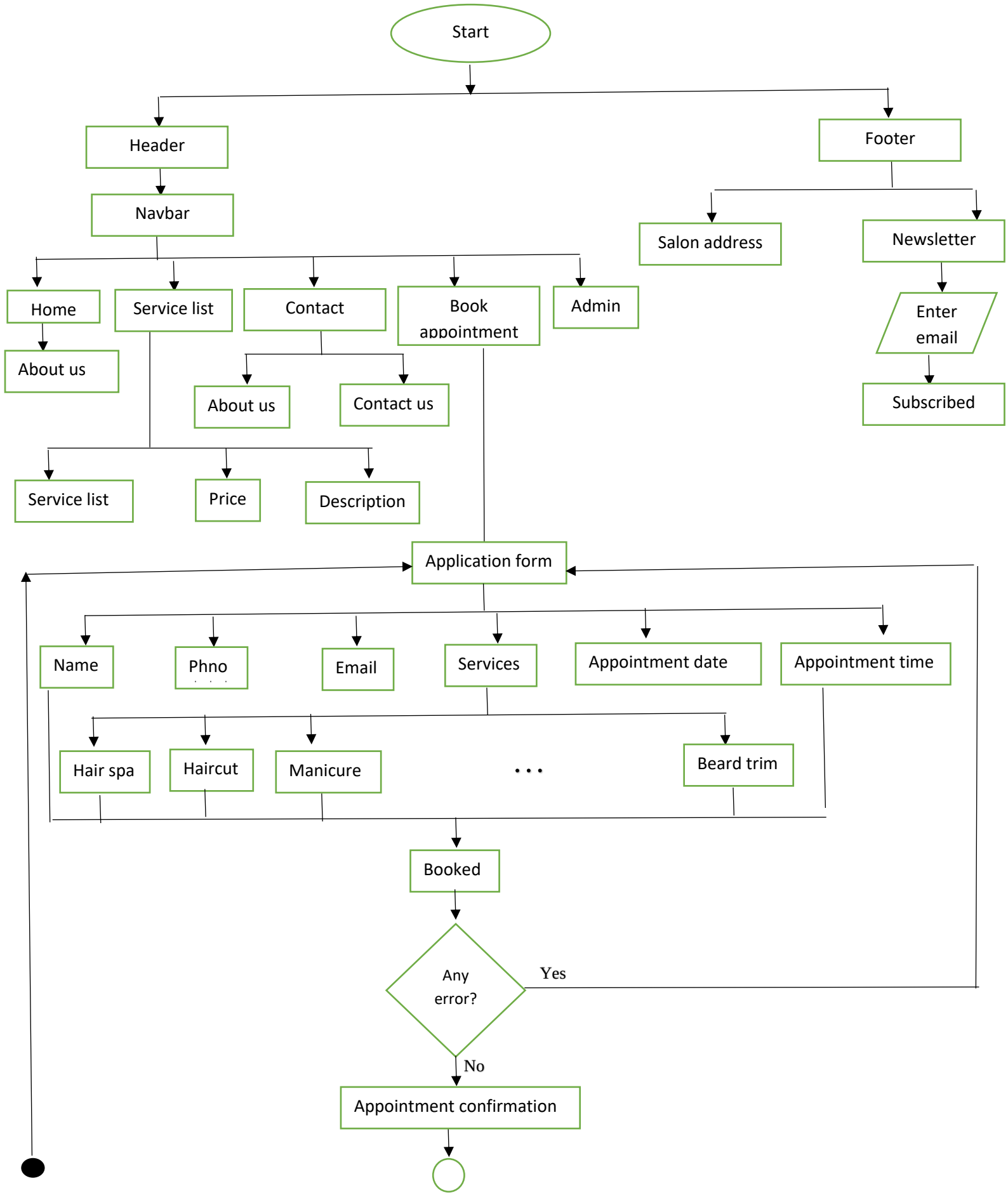
2.7.2 Debugging

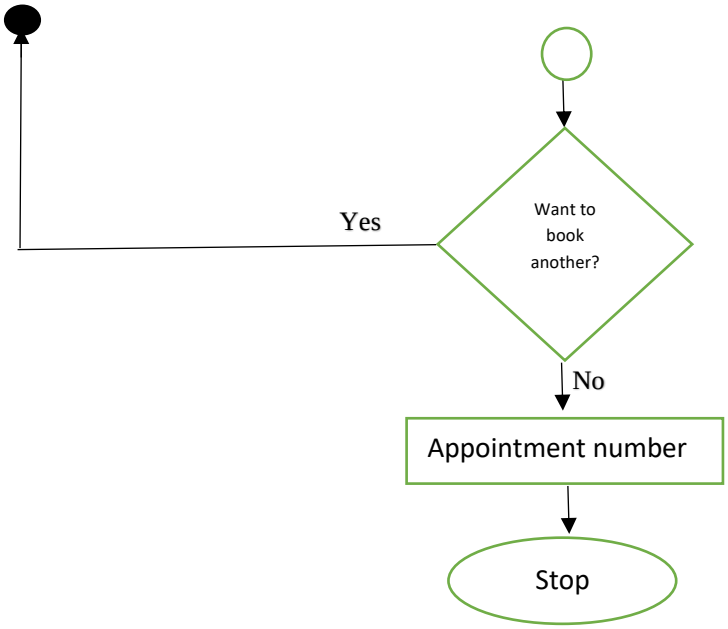
As opposed to Testing, Debugging is the action where the development team or a developer implements after receiving the test report related to the bugs in the software from the testing team.In the software development process, debugging includes detecting and modifying code errors in a software program.

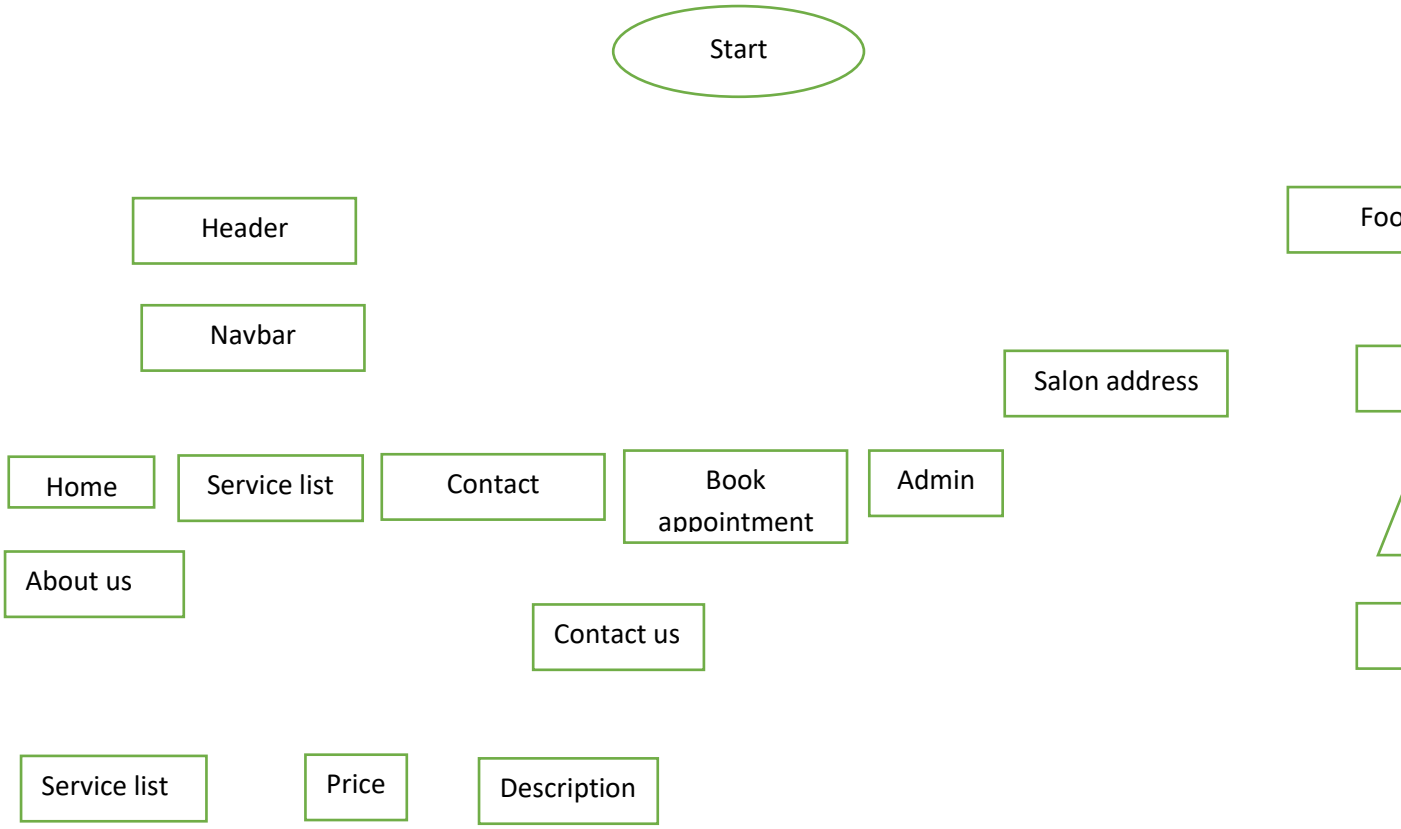
In debugging process, the developer needs to identify the reason behind the particular bug or defect, which is carried out by analyzing the coding rigorously.The developer changes the code and then rechecks whether the defect has been deleted whenever the bug or error is found.


Start
Admin
Admin login
Remember passsword
No
Forgot password
Yes
Sign in
Reset password
Admin panel dashboard
Navbar
Notifications
Admin
Settings
Profile
Logout
Reset password
Admin profile
Current password
New password
Confirm password
Admin name
Username
Contact number
Email


Dashboard
Total customers
Total services
Dashboard
Services
Add services
Manage services
Appointment
All apts
New apts
Accepted apts
Rejected apts
Subscriber
View Details
Emails
Date of subscription
Add customer
Name
Email
Phno
Gender
Appointment date
Appointment time
Search appointment
Search by appointment number
Search invoices
Search by billing address or invoice number
Stop


Start
Header
Navbar
Footer
Salon address
Newsletter
Enter email
Subscribed
Home
Service list
Contact
Book appointment
Admin
About us
About us
Contact us
Service list
Price
Description
Application form
Name
Phno
Email
Services
Appointment date
Appointment time
Hair spa
Haircut
Manicure
Beard trim
Booked
Any error?
Yes
No
Appointment confirmation


Yes
Want to book another?
No
Appointment number
Stop


Start
Header
Navbar
Salon address
Home
Service list
Contact
Book appointment
Admin
About us
Contact us
Service list
Price
Description

CHAPTER 03

IMPLEMENTATION

This Men's Salon is in MySQL, PHP, HTML and CSS. Talking about the features of this system, it contains the user and Admin section. From here, the admin can view all the records of the enrolled students and registered teachers. Also, you can edit or delete any details as per the requirements. The design of this project is pretty simple so that the user won't find any difficulties while working on it. This System in PHP helps in easy management of various records of the students. MySQL: MySql is a database, widely used for accessing querying, updating, and managing data in databases. PHP: Hypertext Preprocessor (PHP) is a technology that allows software developers to create dynamically generated web pages, in HTML, XML, or other document types, as per client request. PHP is open source software.

3.1 Week 8: Salon Web Application

During the development phase, we will iteratively implement each of the features listed during the planning phase. At this stage, there will be many setbacks and obstacle, so the team needs to constantly overcome these obstacles. Moreover, we will prioritize the most important features and need to make intelligent trade-offs between the depth of completeness of a single feature and the breadth of implementation of multiple features.

3.1.1 Working

Our project uses MySQL and PHP to back the interface with strong database functionality. To build the data tables we've used PHP My Admin, which is a web-based interface for MySQL. For the front-end development, We primarily used HTML and JavaScript widgets to style the site and create a simple, clean interface. To facilitate the process of styling the application, we used Adobe Dreamweaver. However, Dreamweaver is very limited, and the scope of my project quickly outgrew that of Dreamweaver's capabilities. Nonetheless, it was a very helpful tool early on. For appointment scheduling, we are going to integrate Web Calendar, a PHP-driven calendaring application, into my Men's Salon application. After much research, we decided to implement Web Calendar rather than Google Calendars for a number of reasons.

First, Web Calendar is installed locally and runs on MySQL. This means that all of the code running Web Calendar is in a subdirectory of my project. This gives me a great deal of control and customization of the application as we integrate it into my program. It also uses MySQL data tables that nicely sit right next to the tables storing data for Men's Salon. Thus, it is a very clean integration, and a powerful tool. Google Calendars, on the other hand, would mean outsourcing functionality to Google. Any future changes to the Google APIs would cause serious problems with my program. Also, viewing and editing Google Calendars requires user authentication as a prerequisite. This means users of my site would need Google accounts or we would personally need to be the "owner" of all the calendars. Neither of these options offered the neat, compact solution that Web Calendar provided, and therefore were not implemented.

3.1.2 Project Features and Functionality

The tab menu on the Home Page allows any visitor to view the top rated salons by four different categories, and filter these results by zip code. This feature finds all salons that meet the zip code criteria (if applicable, otherwise all salons), and for each salon, recalculates each salons average rating across the four categories every time the page is loaded. This ensures the information is always maximally updated. Then it displays the salons that have the highest average ratings by category. Users can click on the salon's name in order to get the address and phone number of the salon. When users sign in, they are directed to their dashboard, their one-stop-shop for all salon-related information and functionality. At the dashboard they are reminded about their points total and upcoming appointments. The JavaScript pull-down menus give the page a clean look while still providing an abundance of information and functionality. The dashboard also defaults to the Salon Search aspect of the application, where users can search salons by name or location. Salons have a similar dashboard that is the root point for all use cases mentioned above. An important difference, however, is the integration of Web Calendar into their management experience. Web Calendar is seamlessly embedded into the Men's Salon application, and customized to look and feel like part of the Men's Salon application. While in the Web Calendar application, salons are provided with a link back to their dashboard, as well as the footer at the bottom of the page with standard links to Home, About Us, etc

Database Design Many-to-one relationships are key to sound database design. In designing a database, inadequacies and inefficiencies that are noticed later in the development cycle can be

costly, and so a lot of time and thought went into this early on. It is tempting to create database tables like classes in object-oriented programming. This would imply that a set of promotions, or stylists belonged to a salon, and thus are fields in the “salon” table. However, this method would be quite inefficient and limiting. Each entry would have to be unnecessarily large to accommodate many potential promotions and stylists, and yet still be finitely limited. Good database design means finding these redundancies and pulling information apart into separate tables that represent relationships between different tables. It turns out that in the salon world, it is often the case that stylists work at multiple salons on different days, and sometimes offer a different subset of services at each salon. If stylists are tied to a column in a salon table, none of these dynamic relationships could be represented. However, we created a Salon-Stylist-Services table that stores a salon id, stylist id, and service id. While there may be many entries for each salon and/or stylist, there will only be one unique combination of a salon, stylist and service. Each entry represents the fact that a specific stylist offers a specific service at a specific salon. Further, when a salon removes a stylist, the stylist is not deleted from the stylist table. Rather, only the entries in the salon-stylist-service table that relate that salon to the stylist are removed. This is more natural because the stylist is not ceasing to exist; simply their relationship with that salon is. Thus, all the desired information is represented, and there is no redundancy because no combination of all three will ever be repeated. The same thing was done with promotions;

We created a separate promotions table that stores the relationship between that promotion and the salon that created it. Displaying Dynamic Data from Many-To-One Tables Many-to-one relationships are very powerful in representing data, but it does mean making a tradeoff in a different respect. Retrieving and displaying information to the end user becomes much more difficult this way. It means you are one, two, or sometimes three additional steps (database queries) away from the information you need. Here is one example of where we encountered this issue and how we overcame it: When viewing salon promotions, users can filter promotions by salon name, zip code, or city. They can also specify hair and/or beauty services they want to filter results by. The salon name, zip, and city forms allow me to query the salon table with the user’s constraints. However, this returns an array of salon database entries. If we had stored promotions as columns in the salon table, we would be done, and could simply loop through the resulting salons and display the promotions within each entry. With my design, we need to loop through each of these salons still, but then perform a database query within this loop to find promotions in the promotions table that have the salon id currently in the loop, as well as

the service constraints that the user specified. This query can also yield multiple results (promotion entries). So then we need an embedded loop that goes through each of these results and puts them in a “results” array that exists outside all loops. Once all resulting promotions are put in the “results” array, step through to the next salon in the outer loop and repeat. Clearly, this is quite a bit more difficult as a result of the many-to-one database design. However, it is a tradeoff worth making because these retrieval scripts are one-time inconveniences to solve and write; whereas the enhanced database design continually increases the power and performance of the application, especially as it grows. Delete in Batches, Insert in Batches Another algorithmic concept came into play when salons add or remove services that a stylist offers at their salon (Same also applies to beauty services offered at a salon, but slightly more simple because no stylists are involved). In this case, a salon chooses a stylist in order to edit their services. The first order of business is to properly populate the checkboxes with the services that the stylist currently offers. To do this, the script queries the many-to-one salon-stylist-service table for entries with the salon’s id that is currently logged (session variable) and the stylist id of the chosen stylist to edit (URL variable passed page to page when salon chooses stylist), and each service. If a result is found, the box is checked. The next step occurs when the salon (un)checks the desired services and submits the form by hitting the submit button. One option is to handle each check box individually; but this would be inefficient and messy. Instead, we first find all entries in the salon-stylistservices table that relate this salon to the stylist at hand, and remove them. Then, in a loop for each checkbox, we add a new entry for this salon, the stylist, and the service tied to that checkbox if it is checked. This is not only a much cleaner routine, but it also keeps all related data together in the database. This is very important for database management and upkeep as the entries grow over time.

MVC Report

The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application.

MVC Components

Following are the components of MVC −

Model

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View

The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

ASP.NET MVC Features

ASP.NET MVC provides the following features −

- Ideal for developing complex but lightweight applications.
- Provides an extensible and pluggable framework, which can be easily replaced and customized. For example, if you do not wish to use the in-built Razor or

ASPX View Engine, then you can use any other third-party view engines or even customize the existing ones.

- Utilizes the component-based design of the application by logically dividing it into Model, View, and Controller components. This enables the developers to manage the complexity of large-scale projects and work on individual components.
- MVC structure enhances the test-driven development and testability of the application, since all the components can be designed interface-based and tested using mock objects. Hence, ASP.NET MVC Framework is ideal for projects with large team of web developers.
- Supports all the existing vast ASP.NET functionalities, such as Authorization and Authentication, Master Pages, Data Binding, User Controls, Memberships, ASP.NET Routing, etc.
- Does not use the concept of View State (which is present in ASP.NET). This helps in building applications, which are lightweight and gives full control to the developers.

1. Model:

The model represents the data and the business logic of the salon management system. It is responsible for handling data storage, retrieval, and manipulation. In this case, the model would include components such as:

- Database schema and tables to store information about salons, customers, appointments, services, etc.

- Data access objects (DAOs) or data models that interact with the database and provide methods for CRUD (Create, Read, Update, Delete) operations.

- Business logic components that handle operations such as appointment scheduling, service management, customer records, etc.

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
if (strlen($_SESSION['bpmsaid'] == 0)) {
    header('location:logout.php');
}
?>
<!DOCTYPE HTML>
<html>

<head>
    <title>MSMS | Admin Dashboard</title>

    <script type="application/x-javascript">
        addEventListener("load", function() {
            setTimeout(hideURLbar, 0);
        }, false);

        function hideURLbar() {
            window.scrollTo(0, 1);
        }
    </script>
    <link href="css/bootstrap.css" rel='stylesheet' type='text/css' />
    <link href="css/style.css" rel='stylesheet' type='text/css' />
    <link href="css/font-awesome.css" rel="stylesheet">
    <script src="js/jquery-1.11.1.min.js"></script>
    <script src="js/modernizr.custom.js"></script>
    <link
href='//fonts.googleapis.com/css?family=Roboto+Condensed:400,300,300italic,400
italic,700,700italic' rel='stylesheet' type='text/css'>
    <link href="css/animate.css" rel="stylesheet" type="text/css" media="all">
    <script src="js/wow.min.js"></script>
    <script>
        new WOW().init();
    </script>
    <script src="js/Chart.js"></script>
                                <h5>Total</h5>
                            ?>
                            <div class="stats-left">
                                <h5>Total</h5>
                                <h4>Services</h4>
                            </div>
                            <div class="stats-right">
                                <label> <?php echo $totalser; ?></label>
                            </div>
                            <div class="clearfix"> </div>
                        </div>
```

```
                        <div class="col-md-4 widget states-last">
                            <?php
                            $query6 = mysqli_query($con, "select
tblinvoice.ServiceId as ServiceId, tblservices.Cost
 from tblinvoice
  join tblservices  on tblservices.ID=tblinvoice.ServiceId where
date(PostingDate)=CURDATE();");
                            while ($row = mysqli_fetch_array($query6)) {
                                $todays_sale = $row['Cost'];
                                $todysale += $todays_sale;
                            }
                            ?>
                            <div class="stats-left">
                                <h5>Today</h5>
                                <h4>Sales</h4>
                            </div>
                            <div class="stats-right">
                                <label> <?php
                                        if ($todysale == '') :
                                            echo "0";
                                        else :
                                            echo $todysale;
                                        endif;
                                        ?></label>
                            </div>
                            <div class="clearfix"> </div>
                        </div>
                        <div class="clearfix"> </div>
                    </div>

                </div>
```

2. View:

The view represents the user interface (UI) of the salon management system. It is responsible for displaying information to the users and capturing user input. The view would include components such as:

- HTML/CSS templates, JavaScript, and UI frameworks to create the visual representation of the system.

- User interfaces for tasks such as viewing salon information, managing appointments, updating customer details, etc.

```
<body>
    <?php include_once('includes/header.php'); ?>
    <div class="page-header">
        <div class="container">
            <div class="row">
                <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
                    <div class="page-caption">
                        <h2 class="page-title">Book Appointment</h2>
                        <div class="page-breadcrumb">
                            <ol class="breadcrumb">
                                <li><a href="index.php">Home</a></li>
                                <li class="active">Book Appointment</li>
                            </ol>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="content">
        <div class="container">
            <div class="row">

                <div class="col-lg-8 col-md-8 col-sm-8 col-xs-12">
                    <div class="row">
                        <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
                            <h1>Appointment Form</h1>
                            <p> Book your appointment to save salon rush.</p>
                            <form method="post">
                                <div class="row">
                                    <div class="col-md-6">
                                        <label class="control-label"
for="name">Name</label>
                                        <input type="text" class="form-
control" id="name" placeholder="Name" name="name" required="true">
                                    </div>
                                    <div class="col-md-6">
                                        <label class="control-label"
type="phone" for="phone">phone</label>
                                        <!-- <input type="text"
name="country_code" title="Error Message" pattern="[1-9]{1}[0-9]{9}"> -->
                                        <input type="text" class="form-
control" id="phone" name="phone" placeholder="Phone" required="true"
minlength="10" maxlength="10" pattern="[0-9]+">
 </div>
 <div class="col-md-6">
  <label class="control-label" type="email" for="email">email</label>
```

```
<input type="email" class="form-control" id="appointment_email"
placeholder="Email" name="email" required="true">
  </div>
  <div class="col-md-6">
   <label class="control-label" for="Subject">Services</label>
   <select name="services" id="services" required="true" class="form-control">
     <option value="">Select Services</option>
       <?php $query = mysqli_query($con, "select * from tblservices");
         while ($row = mysqli_fetch_array($query)) {
 ?>
 <option value="<?php echo $row['ServiceName']; ?>"><?php echo
$row['ServiceName']; ?></option>
  <?php } ?>
   </select>
   </div>
 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  <div class="form-group">
    <label class="control-label" for="textarea">Appointment Date</label>
     <input type="date" class="form-control appointment_date"
placeholder="Date" name="adate" id='adate' required="true">
    </div>
   </div>
   <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
    <div class="form-group">
     <label class="control-label" for="textarea">Appointment Time</label>
      <input type="time" class="form-control appointment_time"
placeholder="Time" name="atime" id='atime' required="true">
    </div>
   </div>
```

3. Controller:

The controller acts as an intermediary between the model and the view. It handles user requests, processes input, and updates the model or view accordingly. In this case, the controller would include components such as:

- Request handlers or controllers that receive user requests from the frontend and invoke appropriate actions.

- Business logic components that orchestrate operations by interacting with the model.

- APIs or endpoints to communicate with the frontend and middleware components.

```
<script src="js/classie.js"></script>
    <script>
      var menuLeft = document.getElementById( 'cbp-spmenu-s1' ),
        showLeftPush = document.getElementById( 'showLeftPush' ),
        body = document.body;

      showLeftPush.onclick = function() {
        classie.toggle( this, 'active' );
        classie.toggle( body, 'cbp-spmenu-push-toright' );
        classie.toggle( menuLeft, 'cbp-spmenu-open' );
        disableOther( 'showLeftPush' );
      };

      function disableOther( button ) {
        if( button !== 'showLeftPush' ) {
          classie.toggle( showLeftPush, 'disabled' );
        }
      }
    </script>
  <script src="js/jquery.nicescroll.js"></script>
  <script src="js/scripts.js"></script>
  <script src="js/bootstrap.js"> </script>
</body>
<?php }  ?>
```

Test Cases

Test1: Home	Remarks
Action: Home page display on click.	Success
Description: Displays Home page consisting Navbar, a button which will redirect to book appointments page, some information of "About us", the address of the salon, phone number, official email id, newsletter where the user can subscribe and get any useful information to the customers' email address.	

Test2: Service list	Remarks
Action: Service list page display on click.	Success
Description: Displays Service list page consisting Navbar, the list of services offered, their prices and the description of the services offered, another button that will redirect to book appointments page, the address of the salon, phone number, official email id, newsletter where the user can subscribe and get any useful information to the customers' email address.	

Test3: Contact us	Remarks
Action: Contact us page display on click.	Success
Description: Displays Contact us page consisting Navbar, about the salon, their vision and mission, the contact information like the address of the salon, phone number, official email id, newsletter where the user can subscribe and get any useful information to the customers' email address.	

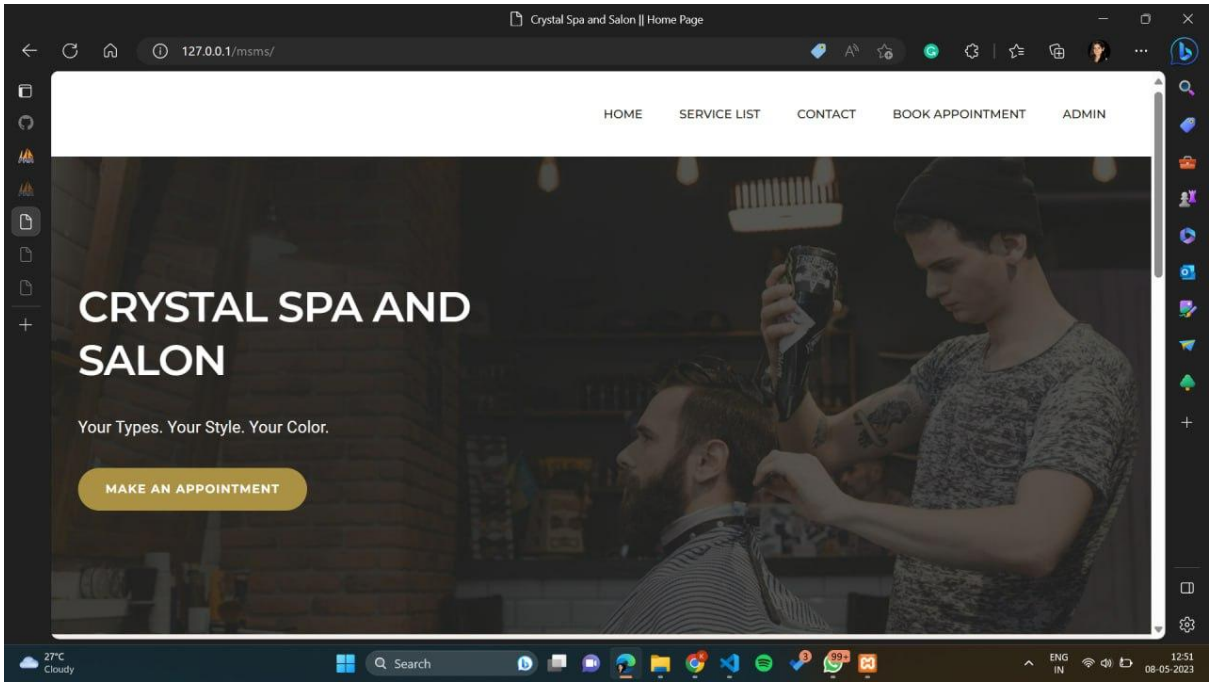
Test4: Book appointment	Remarks
Action: Book appointment page display on click.	Success
Description: Displays Book appointment page consisting Navbar, the appointment form that takes the input from the customer, which consists of Name, phone number, email address of the customer, Service name where the user has to choose one service at a time, appointment date and time of the customers' choice, the address of the salon, phone number, official email id, newsletter where the user can subscribe and get any useful information to the customers' email address.	

Test5: Book appointment	Remarks
Action: Book appointment page display on click.	Failure
Description: Displays Book appointment page consisting Navbar, the appointment form that takes the input from the customer, which consists of Name, phone number, email address of the customer, Service name where the user wants to choose more than one service at a time, appointment date and time of the customers' choice, the address of the salon, phone number, official email id, newsletter where the user can subscribe and get any useful information to the customers' email address.	

Test6: Admin login	Remarks
Action: Admin login page display on click.	Success
Description: Displays Sign in page that takes the input like username and password from the admin. There is also a forgot password feature which will redirect to the page where the password can be changed with the help of registered email and phone number. The forgot password also has a "already have an account?" feature which will again redirect to sign in page.	

Test7: Admin Dashboard	Remarks
Action: Admin Dashboard page display on click.	Success
Description: Displays the details of the booked appointments, customers, number of accepted appointments, number of rejected appointments, total number of services, sales for the day, sales of yesterday, sales for the last seven days, total sales. The admin can also add or delete services offered, edit the pages fed like about us, contact us. The admin can also view each appointment details, new appointments, reject appointments, accept appointments. It can also view the subscribers' details, can manually add the customers, view customer list, edit and add action to it, the reports between certain time constraint, sales reports (either month-wise or year-wise), the invoices generated, search a specific appointment by entering the appointment number, search the invoice number by entering the billing number.	

Test8: Admin navbar	Remarks
Action: Admin navbar page display on click.	Success
Description: Displays admin navbar page consisting of the notification page, admin settings, logout and profile of the admin. Admin settings redirects to password reset page which asks the admin for the current password for confirmation and the new password he wants to change it to. Admin profile displays admin name, username, phone number, email address.	



3.2 Snapshots

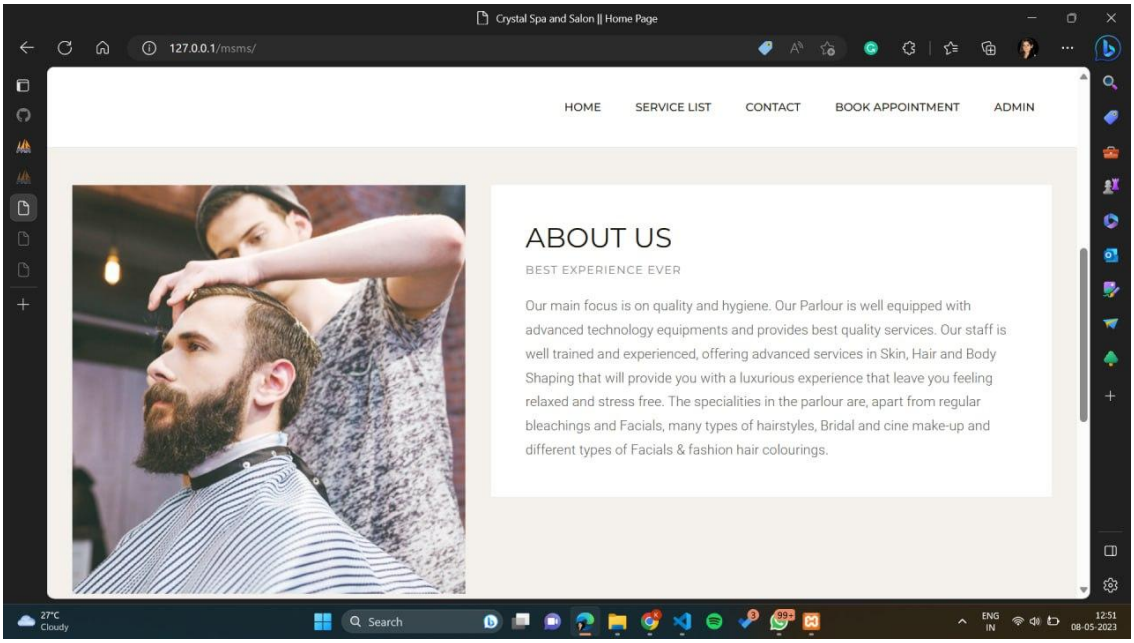


Fig 3.1 Home Page(1)

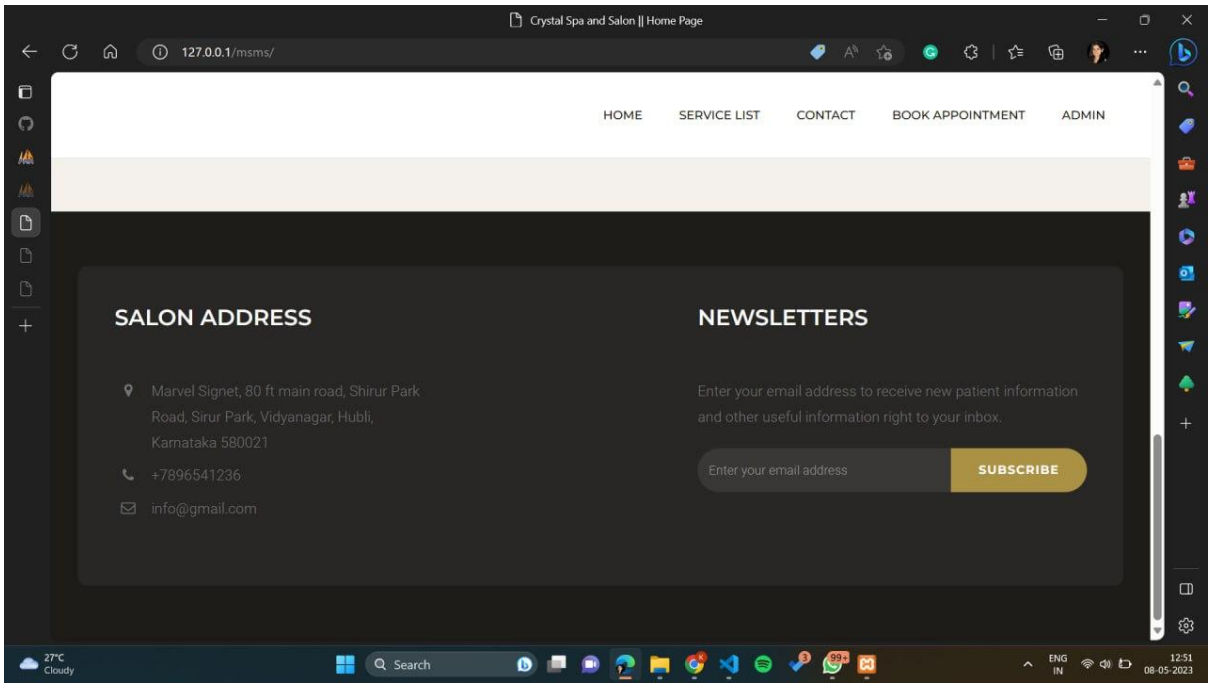


Fig 3.2 Home Page (2)

Fig 3.3 Home Page (3)

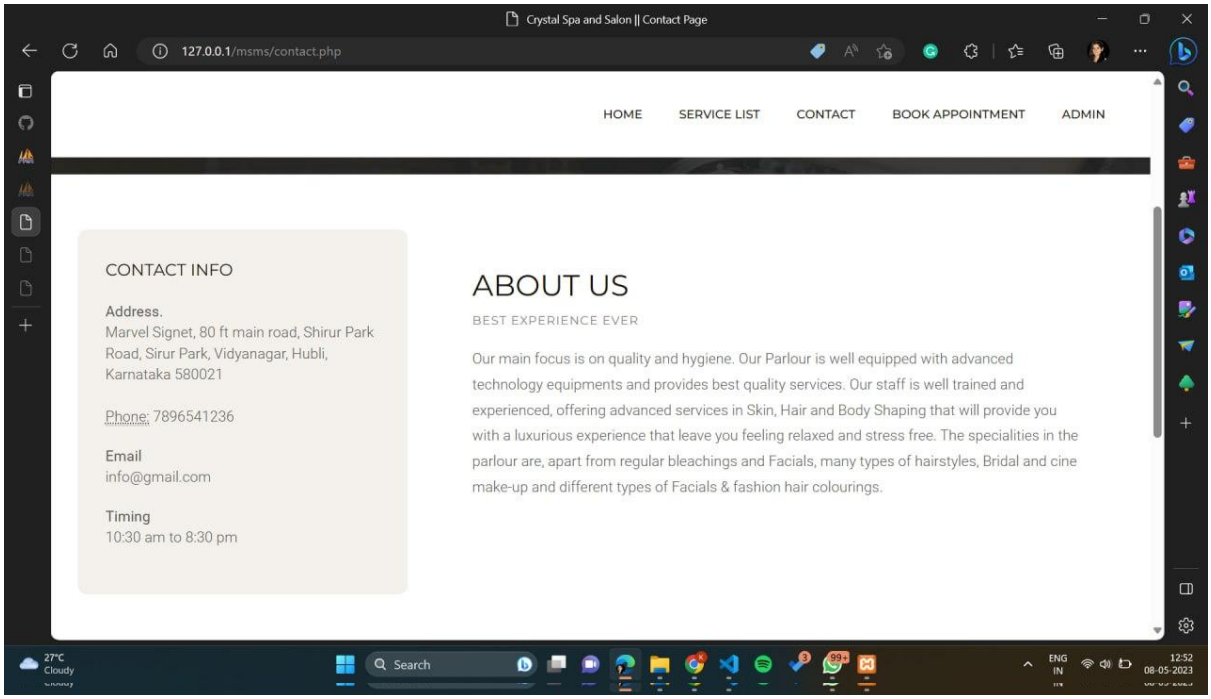


Fig 3.4 Service List

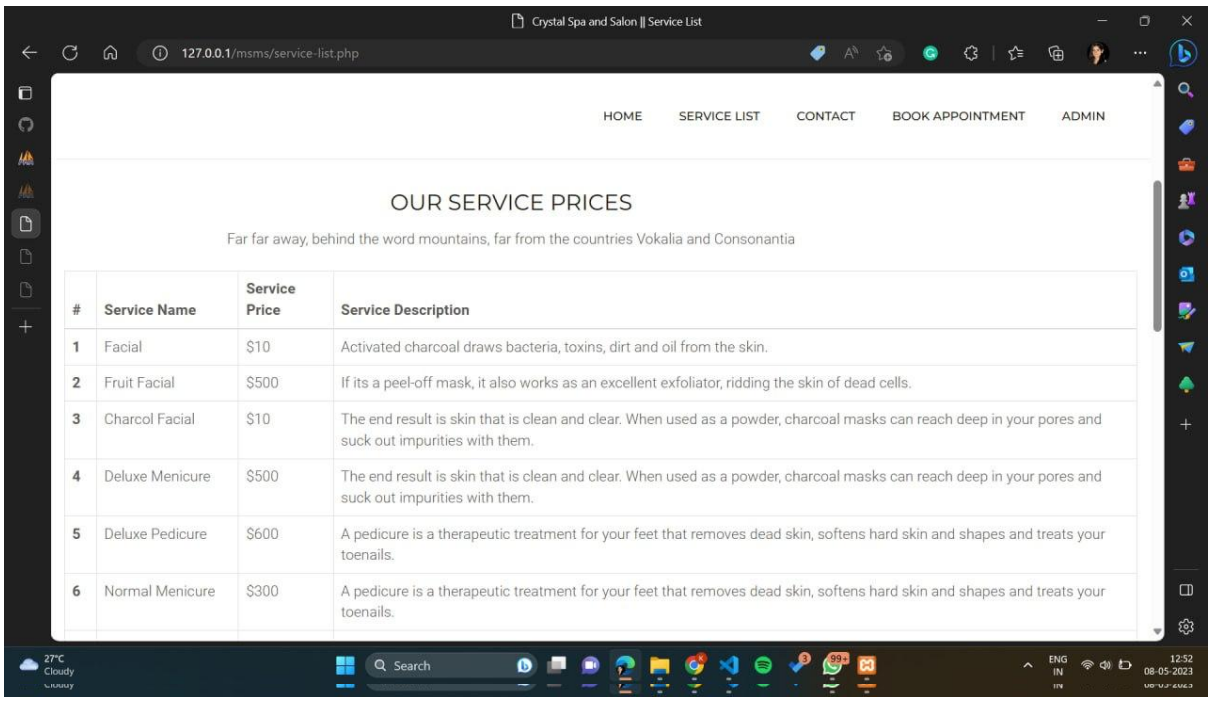


Fig 3.5 Contact

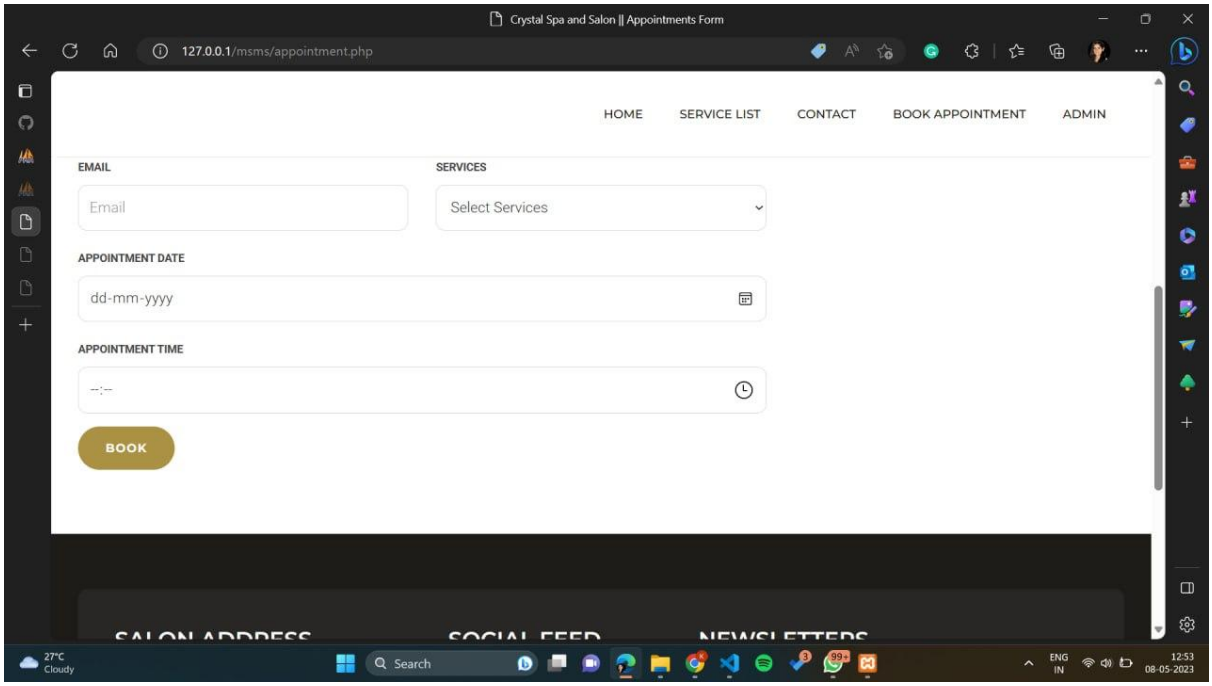


Fig 3.6 Book Appointment

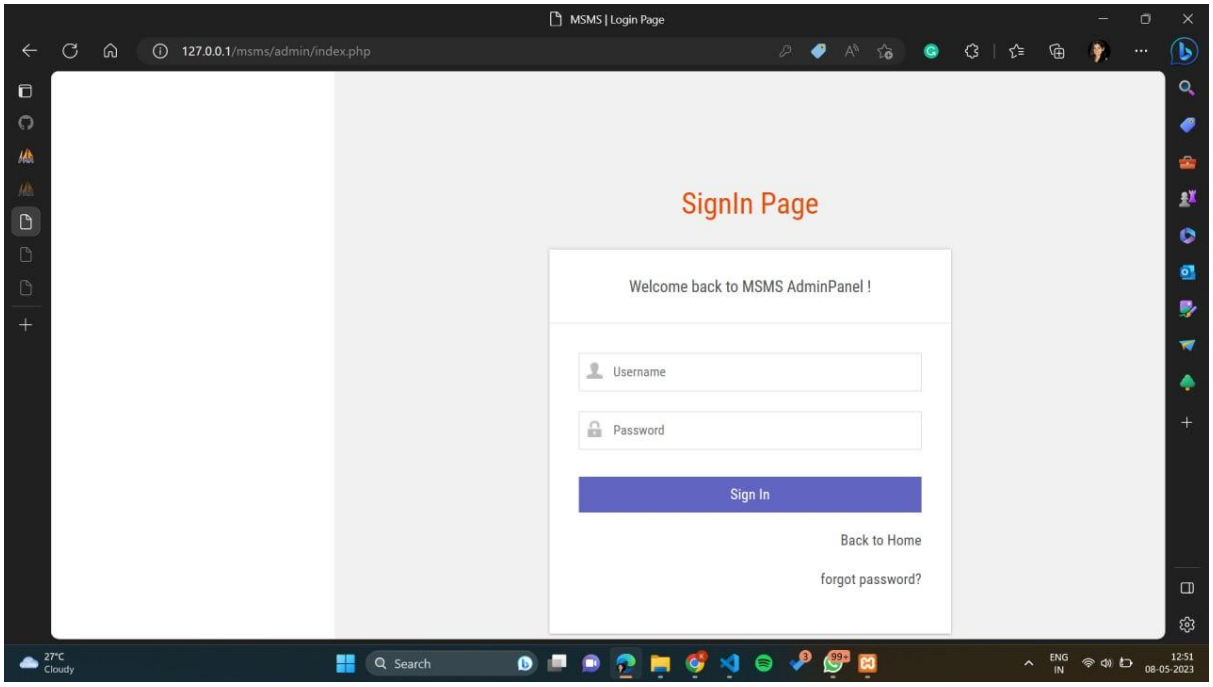


Fig 3.7 Book Appointment(1)

Fig 3.8 Admin page

CHAPTER 04

CONCLUSION AND FUTURE SCOPE

4.1 Conclusion

Working on the project was good experience. We understand the importance of the planning and designing as apart of software development but it's very difficult to complete the program for a single person. Developing the project has helped us some experience on real-time development procedures. Needed documents are generated and adequate and adequate documentation has been provided for maintenance and future enhancement. The operation required by the user to operate the system is basic knowledge of Computer and Internet only. Software is said to have attained its objective only when it meets all the requirements of the user. Developing this project helped me to gain some experience in the real-time development procedures . Hereby we conclude that the system will surely be in a valuable position in contrast to the changing business requirements and modern day cut-throat competition. The system has been tested with simple data to cover all possible options and checked for all outputs. Since the system is flexible, further modifications of this package can be easily incorporated

4.2 Future Scope

The application is yet to be released and a lot of enhancements are already thought of which are proposed to be implemented in the final version of the web-application The system is highly flexible one and is well efficient to make easy infections with the client. The key focus is given on data security, as the project is online and will be transferred in network. The speed and accuracy will be maintained in a proper way. This will be user-friendly one and can successfully overcome strict and severe validation checks. The system will be a flexible one and changes. whenever can be made easy. Using the facility and flexibility in Java and SQL the software can he developed in a neat and simple manner there by reducing the operator's work. Since, the project is developed in Java as a front-end and SQL Server as a buck-end it can be modified easily and used for a long period.

REFERENCES

[1]Wikipedia https://en.wikipedia.org/wiki/HTML

[2] Silvio Peroni “Research Articles in Simplified HTML: A Web-first format for HTML-based scholarly articles”

https://www.researchgate.net/publication/320171394_Research_Articles_in_Simplified_HTML_A_Web-first_format_for_HTML-based_scholarly_articles

[3] w3School “CSS Tutorial”

https://www.w3schools.com/css/

[4]w3School “Bootstrap 3 tutorial”

https://www.w3schools.com/bootstrap/

[5]w3School ” PHP MySQL Database”

https://www.w3schools.com/php/php_mysql_intro.asp