**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

BELGAUM – 590014



A DBMS(17CSL58) PROJECT ON

“BEAUTY PARLOUR MANAGEMENT SYSTEM”

***Submitted in partial fulfillment of the requirement of 5th semester***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**BY**

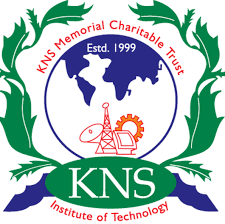
**C H TEJASHWINI 1KN17CS024**

**MERCY RABECAL P 1KN17CS052**

***UNDER THE GUIDANCE OF***

**Mr. Shaik Mohammed Ghouse, Mr. Mohammed Ziaulla,**

Asst. Prof, Dept of CSE Asst. Prof, Dept of CSE

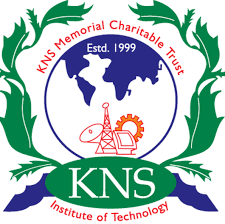
****

**K.N.S INSTITUTE OF TECHNOLOGY**

**Thirumenahalli, Yelahanka, Bangalore – 560064**

**K.N.S INSTITUTE OF TECHNOLOGY**

**Department Of Computer Science and Engineering**

****

**CERTIFICATE**

Certified that the technical project work entitled **“BEAUTY PARLOUR MANAGEMENT SYSTEM”**  is a bonafide work carried out by **C H TEJASHWINI (1KN17CS024) and MERCY RABECAL P (1KN17CS052)** in partial fulfillment of for the award of bachelor of Engineering In Information Science and Engineering of Visvesvaraya Technological University, Belgaum during the academic year 2019-2020.

It is certified that all correction/suggestions indicated for internal assessments have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in aspects of project work prescribed for the Bachelor Of Engineering degree.

………………………….. …………………………… …..…….……………………

**INTERNAL GUIDE INTERNAL GUIDE HOD**

Mr. Mohammed Ziaulla Mr. Shaik Mohammed Ghouse Mr.Mohamed Shakir

Asst Prof, Dept of CSE Asst Prof, Dept of CSE Dept of CSE

KNSIT Bangalore. KNSIT Bangalore. KNSIT Bangalore.

**External Viva:**

**Name of the Examiner** **Signature with Date**

1. **1.**

**2. 2.**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mentioning the people who made it possible. If there is a driving force that kept use going on in doing this project, it is the constant support of our guide **Mr. Mohammed Ziaulla and Mr. Shaik Mohammed Ghouse.** We present our sincere and heartiest thanks to him, for giving us a patienthearing and clearing our doubts.

We would like to express our profound thanks to our HOD **Mr. Mohamed Shakir** for his support and encouragement towards us at every stage in successful completion of our project.

We would like to express our gratitude to **Dr. S.M Prakash, Principal , KNSIT Bangalore,** for providing us congenial environment to work in.

We place our gratitude to our parents who always encouraged us to pursue our interest, who have led us onto the right path. We are undoubted for all the love and affection they bestowed upon us. Many are responsible for the knowledge and experience we have gained during our project and throughout the course. With profound sense of gratitude and regards we acknowledge for support extended by our friends for their support in completion of this project.

Finally, we would also like to thank all those who knowingly or unknowingly helped to us all throughout our project.

**C H TEJASHWINI 1KN17CS024**

**MERCY RABECAL P 1KN17CS052**

**ABSTRACT**

Beauty parlour management system is a web-based salon management application with appointment scheduling functionality. It connects clients, salons, and stylists in an online community allowing users to browse salons and stylists, and book or cancel appointments. Users can also write and read reviews of salons and particular stylists. Salons can specify the stylists that work at their salons, as well as the services they offer. Salons can also book appointments for customers, and can view and print schedules in convenient formats.

It serves a similar need in the restaurant industry, but nothing quite like this existed to bring clients and beauty salons together online. SalonBook.com fills this void in a way that is on-demand, easy to use, and effective for users and salon managers.

My project will use MySQL and PHP to back the interface with strong database functionality. For appointment scheduling, This Project will integrate Web Calendar as a backend database for appointments as well as a front-end scheduling interface. Web Calendar is a PHP-based calendaring application that can be a stand-alone program or integrated into other applications. This project will target the major web browsers as the initial platform for the Beta version.

The final deliverable will be a functioning web application that can handle all specified use cases. Some of the major use cases include user account registration, login/logout, appointment scheduling, schedule viewing and printing, adding stylists and services to a salon account, adding schedules to stylist account, accumulating points in a client account, writing and reading reviews for specified salons and/or services, and creating temporal salon promotions that users can browse and filter.

**CONTENTS**

**1. INTRODUCTION 1**

**2. SIGNIFICANCE OF PROJECT DEVELOPMENT 2**

**3. SCOPE 2**

**4. EXISTING SYSTEM 3**

**5. PROPOSED SYSTEM 3**

**6. PROBLEMS IN EXISTING SYSTEM 3**

**7. HARDWARE AND SOFTWARE SPECIFICATION 4**

**8. TOOLS AND PLATFORM 5**

**8.1.1 ABOUT FRONT END (PHP,HTML,CSS,JS) 5**

**8.1.2 ABOUT BACKEND 9**

**8.2 PLATFORM (MYSQL) 10**

**9. FEASIBILITY STUDY 11**

**9.2 NEEDS FOR FEASIBILITY STUDY 11**

**9.3 ECONOMIC FEASIBILITY 11**

**9.4 TECHNICAL FEASIBILITY 12**

**9.5 BEHAVIOURAL FEASIBILITY 12**

**10. CONTEXT LEVEL DIAGRAM 13**

**11. ENTITY-RELATIONSHIP DIAGRAM 15**

**12. EVENT TABLE 16**

**13. SYSTEM CHART 17**

**14. STRUCTURE CHART 18**

**15. USE CASE DIAGRAM 19**

**16. DATAFLOW DIAGRAM 20**

**17. PACKAGE DIAGRAM 23**

**18. PERT CHART 24**

**19. GANTT CHART 24**

**20. SECURITY TESTING OF PROJECT 25**

**20.1.1 WHITE BOX TESTING 25**

**20.1.2 BLACK BOX TESTING 25**

**20.1.3 ALPHA TESTING 26**

**20.1.4 BETA TESTING 26**

**20.1.5 UNIT TESTING 26**

**20.1.6 INTEGRATION TESTING 26**

**20.1.7 VALIDATION TESTING 26**

**20.1.8 SYSTEM TESTING 26**

**20.2 IMPLEMENTATION AND SOFTWARE**

**SPECIFICATION TESTING 27**

**20.2.1 DETAILED DESIGN OF IMPLEMENTATION 27**

**20.2.2 TECHNICAL DESIGN 27**

**20.2.3 TEST SPECIFICATIONS AND PLANNING 27**

**20.2.4 PROGRAMMING AND TESTING 27**

**20.2.5 USER TRAINING 27**

**20.2.6 ACCEPTANCE TEST 27**

**20.2.7 INSTALLATION PHASE 27**

**20.2.8 SYSTEM INSTALLATION 27**

**20.2.9 REVIEW PHASE 28**

**20.2.10 DEVELOP RECAP 28**

**20.2.11 POST IMPLEMENTATION REVIEW 28**

**20.3 STEPS IN UNIT TESTING 28**

**20.4 STEPS IN SYSTEM TESTING 29**

**20.5 SPECIAL CATEGORY TEST 29**

**21. SYSTEM ANALYSIS 30**

**22. RESULT 31**

**23. CONTRIBUTION 31**

**24. FUTURE DIRECTION 32**

**25. SNAPSHOTS 34**

**26. REFERNCES 39**

**1. INTRODUCTION**

Beauty is the gift of god and transmitted from one to the other generation. The beauty parlour shop is the need of every age of men women and children famous poet Keats defined the word beauty as : " A thing of beauty is a joy for ever: Beauty parlour is a very important shop to make the people good looking by application of cosmetics treatment of hair and nourishment of skin by various methods . Life style is fast changing in the modern era and the women have become more conscious about their make up the status of women has improved a lot and is improving further. This has resulted in their life style changing. Also their economic independence encourages them to resort this type of services.

Beauty parlour is a service oriented establishment in which women receive treatment to increase their beauty. Skin care, facial make up, hair nourishment and model hair cutting are the most important nourishing activities of a beauty parlour. The use of machinery, ayurvedic formulations and approved synthetic chemicals and medicines vary from person to person as per need. In present project the provision of different beauty therapy are present with use of limited number of machinery.

From the ages past people have used sandal wood oil turmeric powder and milk etc. For the treatment of skin but now people have become more educated they need proper and professional treatment of skin hairs nails and teeth. Every man women and child want to be beautiful. A good beauty parlour is required in every city town and other places. A beauty parlour with all modern facilities and services at reasonable charges more visit can be expected from each individuals due to the scarcity of experience and qualified beautician there is immense marketability for this venture.

In the 21st century where people are busy with their works, they cannot afford to waste time by waiting for their time to come in the parlour. So there was a need to provide a solution to such problems of the customer who were the major need of the business. Hence, we created an online solution where customer can book an appointment with the beautician to meet his/her beauty needs with worrying about time constraints and choose the time of his choice rather than waiting in the parlour for his/her turn to get beautified by the magical hands of the beautician.

**2. SIGNIFICANCE OF PROJECT DEVELOPMENT**

Being a salon client today is inconvenient. You have to remember that you need to make an appointment, then hope you remembered during business hours, and finally scramble to find the phone number and take time out of your busy day to make the call. And never mind finding user reviews of local salons and stylists; tracking those down somewhere on the web is a headache that most don’t even attempt. This is the experience of countless salon-goers every day, even as we move well into the twenty first century. There is a strong, well-articulated need for a “one-stop-shop” online hub that connects clients to salons, putting all the information users need in one convenient place, on demand.

Equally surprising, many salons today are still using paper and pencil systems to schedule appointments and manage customers. These systems are grossly inefficient at sharing and syncing information, and are prone to costly human error. This project promises to end the dependence on paper-pencil systems, and put salons in control with an easy to use interface. This project will be more than just an online scheduler; it will be a salon management tool that will allow salons to manage stylists and services, promote sales to customers, and track customer satisfaction.

* Provide a Single environment from where an administrator can interact with the beauty shop system resources within the beauty centre network.
* Provides Info regarding the beauty parlour no of services running on the beauty centre system.
* To provide an upgrade support for the beauty shop system resources.
* Provides interface to interact with the beauty shop system parameters like network settings, OS Settings, Services, and Process etc.
* Provides means of reporting the beauty parlour system setting to a person through email. There by an administrator can know the status of the systems throughout the network.

**3. SCOPE**

Current scope of the beauty parlour project is to beauty parlour management system project in php is to check the state of the beauty parlour services and drivers, state of the beauty parlour processes, state of drivers the skin care, cosmetic treatments and hair salon application doesn’t provide any method to work with these aspects of the beauty parlour management.

**4. EXISTING SYSTEM**

In present system you have to call the salon to fix an appointment. After taking an appointment you have to remember the date of the appointment. User is also not able to find the best salon in their locality. He can find out the services of any salon only after taking their services. But in proposed system you can check review online and find out who is giving best services. Use can also check that which salon gives good customer satisfaction.

In current system salon take appointment on register. They manage customer record on register. And it is very difficult to find out old appointment details in this system. Making report for the salon business is also very tiresome task. This system is prone to costly human error. Beauty parlour management system allow salon to manage stylists and services, promote sales to customers, and track customer satisfaction.

**5. PROPOSED SYSTEM**

* Beauty parlour management system is developed to automate all the salon activity. Using this system user can manage their salon and also use this system to book appointment of his customer online.
* User can also check the top rated salons in his locality using this system online. User can get the address of the top rated salon in his locality by clicking on the name of the salon.
* 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. Customer can login into the system by giving valid username and password.
* After login he can check his upcoming appointments. Salon admin can also how many appointment they have to complete on particular date. User can also search any salon by entering its name in the search box.

**6. PROBLEMS IN EXISTING SYSTEM**

* The current system is provided by Microsoft Corporation, but requires a person to know the methodology to interact with the beauty shop system configuration.
* Not a single interface to monitor all the free download project report events. Don’t notify to the Spa salon user in case any event generated errors.
* System resources cannot be freed frequently based on the beauty centre parameters.
* Administrator is not notified regarding any failure by the skin health system.
* The current system doesn’t allow the remote access of the beauty parlour resources of one system with other·
* Three different Categories are provided to know the current status of the beauty parlour services, drivers etc.
* The single application can be used to configure multiple versions of operating systems which is possible by allowing the beauty parlour application settings to beauty parlour management system project in java get configures and show appropriate panel for the beauty shop administrator.
* Updating of components can be done by accessing the beauty parlour components of the beauty parlour system as well as the skin care, cosmetic treatments and hair salon resources available on the beauty centre websites.
* Maintain the beauty centre information in the beauty centre system registry so that the settings can be read and written directly from the skin care registry.

**7. HARDWARE & SOFTWARE SPECIFICATIONS**

* **Hardware Requirements:**
* **Processor:** Intel i3 or higher
* **Ram:** 4GB or Higher
* **Storage:** 1 TB or Higher
* **Software Requirements:**
* **Operating System:** Windows 7/8/10
* **Software Tool:** XAMPP Server
* **Database & Language Used:** MySQL, PHP
* **User Interface:** HTML,JQUERY,AJAX,JAVASCRIPT

**8. TOOLS AND PLATFORM**

**8.1 PLATFORMS**

**8.1.1 ABOUT FRONT END:**

**8.1.1.1 PHP:-**

**Introduction**

PHP is now officially known as “**PHP: Hypertext Preprocessor**”. It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities is endless. The output from PHP code is combined with the HTML in the script and the result sent to the user’s web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because the entire browser sees is HTML.

PHP’s support for Apache and MySQL further increases its popularity. Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it. The combination of Apache, MySQL and PHP is all but unbeatable.

That doesn’t mean that PHP cannot work in other environments or with other tools. In fact, PHP supports an extensive list of databases and web-servers. While in the mid-1990s it was ok to build sites, even relatively large sites, with hundreds of individual hard-coded HTML pages, today’s webmasters are making the most of the power of databases to manage their content more effectively and to personalize their sites according to individual user preferences.

**8.1.1.2 HTML**

**HTML** or **Hyper Text Markup Language** is the standard markup language used to create web pages.

HTML was created in 1991 by Tim Berners-Lee at CERN in Switzerland. It was designed to allow scientists to display and share their research.

HTML is written in the form of HTML elements consisting of *tags* enclosed in angle **) HTML**

**Images - The <img> Tag and the src Attribute**

In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only, and has no closing tag.

To display an image on a page, you need to use the src attribute. src stands for "source". The value of the src attribute is the URL of the image you want to display.

**Syntax for defining an image:**

<img src="*url*" alt="*some text*">

**b) HTML FORMS**

HTML forms are used to pass data to a server.

|  |
| --- |
| The <form> tag is used to create an HTML form:  <form> . *input elements* . </form> |

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

**c)Image tag (<img>) :**

To add an image to an HTML document, we just need to include an <IMG> tag with a

reference to the desired image. The <IMG> tag is an empty element i.e. it doesn’t require a

closing tag and we can use it to include from small icons to large images.

**Syntax: <imgsrc=”URL” alt=”alternative text”>**

**8.1.1.3 HTML 5**

HTML5 will be the new standard for HTML. The previous version of HTML, HTML 4.01,

came in 1999. The web has changed a lot since then. HTML5 is still a work in progress.

However, the major browsers support many of the new HTML5 elements and APIs.

HTML5 is cooperation between the World Wide Web Consortium (W3C) and the Web

Hypertext Application Technology Working Group (WHATWG).

WHATWG was working with web forms and applications, and W3C was working with

XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

Some rules for HTML5 were established:

a) New features should be based on HTML, CSS, DOM, and JavaScript

b) Reduce the need for external plug-ins (like Flash)

c) Better error handling

d) More markup to replace scripting

e) HTML5 should be device independent

f) The development process should be visible to the public

**8.1.1.4 CSS**

**CSS tutorial** or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

1. CSS stands for Cascading Style Sheet.
2. CSS is used to design HTML tags.
3. CSS is a widely used language on the web.
4. HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table less web design).

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified.

With plain HTML you define the colors and sizes of text and tables throughout your pages. If

you want to change a certain element you will therefore have to work your way through the

document and change it. With CSS you define the colors and sizes in "styles". Then as you

write your documents you refer to the styles. Therefore: if you change a certain style it will

change the look of your entire site. Another big advantage is that CSS offers much more

detailed attributes than plain HTML for defining the look and feel of your site.

**8.1.1.5 JAVASCRIPT**

**JavaScript** (**JS**) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.

**8.1.2 About Back End:**

In a previous blog, we talked about how web programmers are concerned with launching websites, updates, and maintenance, among other things. All of that works to support the front-end of the website. The back-end has three parts to it: server, application, and database.

To better explain how all of this works, let’s use the example of a customer trying to purchase a plane ticket using a website. Everything that the customer sees on the webpage is the front-end, as we have explained before, but once that customer enters all of his or her information, such as their name, billing address, destination, etc, the web application stores the information in a database that was created previously on the server in which the website is calling for information.

The web application creates, deletes, changes, renames, etc items in the database. For example, when a customer purchases a ticket, that creates an item in the database, but when they have a change in their order or they wish to cancel, the item in the database is changed.

.In short, when a customer wants to buy a ticket, the backend operation is the web application communicating with the server to make a change in a database stored on said server. Technologies like PHP, Ruby, Python, and others are the ones backend programmers use to make this communication work smoothly, allowing the customer to purchase his or her ticket with ease

**8.2 PLATFORM**

**8.2.1 MY SQL:**

The database has become an integral part of almost every human's life. Without it, many things we do would become very tedious, perhaps impossible tasks. Banks, universities, and libraries are three examples of organizations that depend heavily on some sort of database system. On the Internet, search engines, online shopping, and even the website naming convention would be impossible without the use of a database. A database that is implemented and interfaced on a computer is often termed a database server.  
 One of the fastest SQL (Structured Query Language) database servers currently on the market is the MySQL server, developed by T.c.X. DataKonsultAB. MySQL, available for download at www.mysql.com, offers the database programmer with an array of options and capabilities rarely seen in other database servers. MySQL is free of charge for those wishing to use it for private and commercial use. Those wishing to develop applications specifically using MySQL should consult MySQL's licensing section, as there is charge for licensing the product.

brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as Java Script which affect the behavior of HTML web pages.

HTML is descriptive markup language. Library of various markup languages is defined in various browsers.

**9. FEASIBILITY STUDY**

A feasibility study is undertaken to determine to the possibility or probability of either improving the existing system or developing a completely new system. It helps to obtain the overview of the problem and to get a rough assessment of whether other feasible solution exists.

**9.2 NEEDS FOR FEASIBILITY STUDY:**

The feasibility study is needed for following things:-

* Answer the questions whether a new system is to be installed or not?
* Determine the potential of the existing system.
* Improve the existing system.
* Know what should be embedded in the new system.
* Define the problems and objectives involved.
* Avoid costly repairs at later stage when system is implemented.
* Avoid crash implementation of the new system.
* Avoid the ‘Hardware approach’ i.e. getting a computer first and then deciding how to use it.

The Feasibility study is divided in to three parts:-

* TECHNICAL FEASIBILITY
* ECONOMIC FEASIBILITY
* OPERATIONAL FEASIBILITY

**9.3 ECONOMIC FEASIBILITY**

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and Implement the system.

This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases.

In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving lot of time.

**9.4 TECHNICAL FEASIBILITY**

Technical feasibility centers on the existing manual system of the test management process and to what extent it can support the system.

According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities.

The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

**9.5 BEHAVIOURAL FEASIBILITY**

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

**10. CONTEXT LEVEL DIAGRAM**

A System Context Diagram (SCD) in [software engineering](http://en.wikipedia.org/wiki/Software_engineering) and [systems engineering](http://en.wikipedia.org/wiki/Systems_engineering) is a [diagram](http://en.wikipedia.org/wiki/Diagram) that represents the actors outside a system that could interact with that [system](http://en.wikipedia.org/wiki/System). This diagram is the highest level view of a [system](http://en.wikipedia.org/wiki/System), similar to [Block diagram](http://en.wikipedia.org/wiki/Block_diagram), showing a, possibly [software](http://en.wikipedia.org/wiki/Software_system)-based, system as a whole and its [inputs](http://en.wikipedia.org/wiki/Input/output) and [outputs](http://en.wikipedia.org/wiki/Output) from/to external factors.

Account Checking

Cosmetic Shop Management System

Customer

Admin

(User)

Supplier

Staff

(User)

Request for Item

Supply of Item

Login

Customer Record,

Create bills

Request for Item

Supply of Item

Login

Man. Login Details

Bill

Bill

Payment

Payment

**Context Level Diagram for Cosmetic Shop Management System**

**Choice of Product**

**Customer**

**Order**

**Suggestions**

**Inquiry**

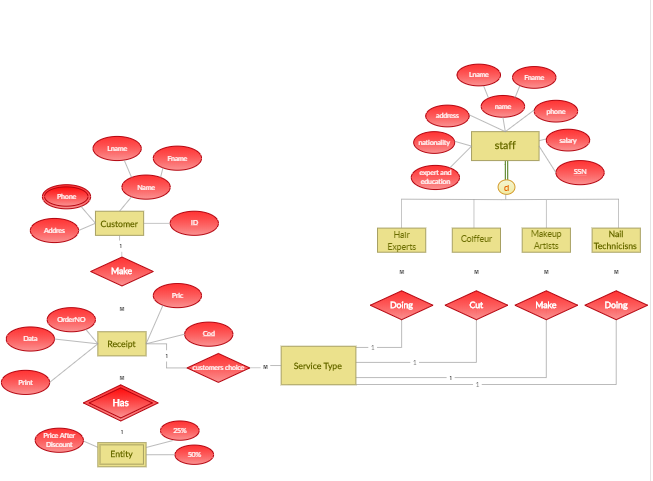
**Bill**

**Products**

**Supplier**

**11. Entity-Relationship Diagram**

* This document is an entity-relationship diagram, or “ERD,” for a system to manage Inventory Management System.
* An ERD is a model that identifies the concepts or entities that exist in a system and the relationships between those entities.
* An ERD is often used as a way to visualize a relational database: each entity represents a database table, and the relationship lines represent the keys in one table that point to specific records in related tables.
* ERD may also be more abstract, not necessarily capturing every table needed within a database, but serving to diagram the major concepts and relationships.
* This ERD is of the latter type, intended to present an abstract, theoretical view of the major entities and relationships needed for management of electronic resources.
* It may assist the database design process for an e-resource management system, but does not identify every table that would be necessary for an electronic resource management database.



**12. Event Table**

The event table is a table of data that is typically written to the [log file](http://www.neurobs.com/pres_docs/html/09_data_reporting/01_logfiles/index.html) for each [scenario](http://www.neurobs.com/pres_docs/html/03_getting_started/04_scenarios/index.html) and also appears in the [Analysis window](http://www.neurobs.com/pres_docs/html/09_data_reporting/02_analysis_features/02_using_the_analyzer.htm). The event table contains timing information about specific events that occur during the scenario.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Activity** | **Response** | **Destination** |
| Customer Registration | Ask for Item | Customer | Create new Bill | Customer registered | Sells Person |
| Sells Person check availability of Stock | Stock Inquiry | Customer | Look up Stock availability | Stock details | Customer |
| Customer select Item | Enquiry about Item | Sells Person | selecting | Item selected | Customer |
| Sells Person gives Item to Customer | Check the Price | Staff | Look up in register | Item given | Customer |
| Produce Bill report | Quantity inquiry | Sells Person | Checking the Stock | Bill report | Customer |
| Customer Checks data | Ask for Bill | Sells  Person | checking | Data checked | Accountant |
| Summary of Bill report | End of day | Sells Person | Produce Bill summary report | Bill summary report | Admin (Owner) |

**13. SYSTEM CHART**

A flowchart that emphasizes the component operations which make up a system.

**BILL**

**GENERATES**

**BILL**

**CUSTOMER**

**EMPLOYEE**

**DEBIT / CREDIT**

**CUSTOMERDATABASE**

**EMPLOYEEDATABASE**

**DEBIT / CREDIT DATABASE**

**PRODUCT / SUPPLIER DATABASE**

**CUSTOMER**

**EMPLOYEE**

**SHOP ACCOUNT**

**Cosmetic Shop Management**

**PRODUCTS**

**PRODUCTS**

**BILL DATABASE**

**14. STRUCTURE CHART**

A **Structure Chart** (SC) in [software engineering](http://en.wikipedia.org/wiki/Software_engineering) and [organizational theory](http://en.wikipedia.org/wiki/Organizational_theory) is a [chart](http://en.wikipedia.org/wiki/Chart), which shows the breakdown of the [configuration system](http://en.wikipedia.org/wiki/Configuration_system) to the lowest manageable levels.

This chart is used in [structured programming](http://en.wikipedia.org/wiki/Structured_programming) to arrange the program modules in a tree structure. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between the modules.

**Cosmetic Shop Management System**

**Bill**

**Customer**

**Peripherals**

**Supplier / Order**

**Reports**

Customer details

Bill details

Order

Details

Peripherals details

**15. USECASE DIAGRAM**

A use case diagram in the [Unified Modeling Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of behavioral diagram defined by and created from a [Use-case analysis](http://en.wikipedia.org/wiki/Use-case_analysis). Its purpose is to present a graphical overview of the functionality provided by a system in terms of [actors](http://en.wikipedia.org/wiki/Actor_(UML)), their goals (represented as [use cases](http://en.wikipedia.org/wiki/Use_case)), and any dependencies between those use cases.

Admin

Staff

Customer

**16. DATAFLOW DIAGRAM**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system). DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of [data processing](http://en.wikipedia.org/wiki/Data_processing) (structured design).

CUSTOMER DETAILS

**Customer**

**Customer**

**Details**

**1.2**

**UPDATE**

**1.3**

**DELETE**

**1.1**

**ADD**

**1.4**

**SEARCH**

**CUSTOMER**

**DETAILS**

**1.0**

**Customer**

**Customer Information**

**Customer**

**1.1**

**ADD**

**Customer**

**Customer**

**1.3**

**DELETE**

**Customer**

**Customer**

**1.2**

**UPDATE**

**Customer**

**Customer**

**1.4**

**SEARCH**

**Customer**

**ACCOUNT**

**1.1**

**ADD**

**ACCOUNT**

BILL ORDER DETAILS

**Order Information**

**Order**

**Order**

**Details**

**2.2**

**UPDATE**

**2.3**

**DELETE**

**2.1**

**ADD**

**2.4**

**SEARCH**

**Order**

**DETAILS**

**2.0**

**Bill Order**

DEALER / PRODUCT DETAILS

**Product**

**Information**

**PRODUCT**

**e**

**3.2**

**UPDATE**

**3.3**

**DELETE**

**3.1**

**ADD**

**3.4**

**SEARCH**

**PRODUCT**

**DETAILS**

**3.0**

**Dealer Order**

**3.5**

**EDIT**

**Product**

**Details**

EMPLOYEE DETAILS

**Employee**

**7.2**

**UPDATE**

**7.3**

**DELETE**

**7.1**

**ADD**

**7.4**

**SEARCH**

**Employee DETAILS**

**7.0**

**Employee**

**7.5**

**EDIT**

**Employee**

**Details**

**Employee Informa**tio**n**

**17.**

**17. PACKAGE DIAGRAM**

**Package diagram** is **UML** [structure diagram](https://www.uml-diagrams.org/uml-25-diagrams.html#structure-diagram) which shows [packages](https://www.uml-diagrams.org/package-diagrams.html#package) and dependencies between the packages.

Log Management

Subsystem

Admin Management

Subsystem

Customer Management

Subsystem

Dealer Management

Subsystem

Report Management

Subsystem

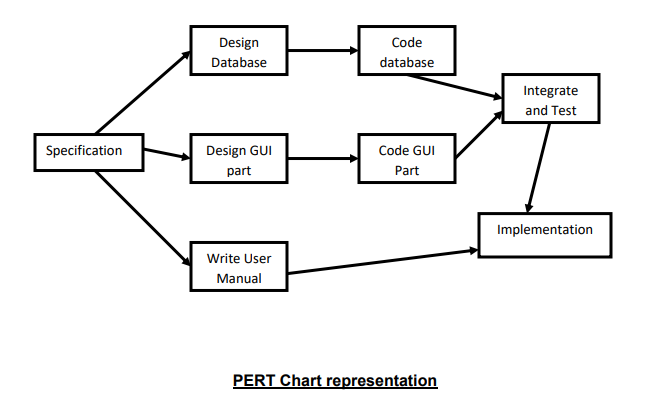
Employee Management

Subsystem

Cosmetic Shop Management System

**18. PERT CHART (PROGRAM EVALUATION REVIEW TECHNIQUE)**

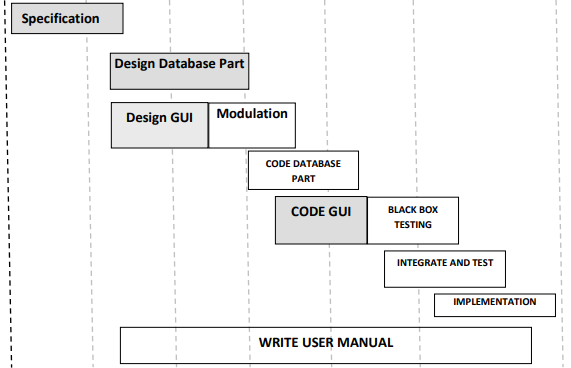
PERT chart is organized for events, activities or tasks. It is a scheduling device that shows graphically the order of the tasks to be performed. It enables the calculation of the critical path. The time and cost associated along a path is calculated and the path requires the greatest amount of elapsed time in critical path.



**19. GANTT CHART**

It is also known as Bar chart is used exclusively for scheduling purpose. It is a project controlling technique. It is used for scheduling. Budgeting and resourcing planning. A Gantt is a bar chart with each bar representing activity. The bars are drawn against a time line. The length of time planned for the activity. The Gantt chart in the figure shows the Gray parts is slack time that is the latest by which a task has been finished.

|  |  |  |
| --- | --- | --- |
| **AUGUST** | **SEPTEMBER** | **OCTOBER** |



**20. SECURITY TESTING OF THE PROJECT**

Testing is vital for the success of any software. no system design is ever perfect. Testing is also carried in two phases. first phase is during the software engineering that is during the module creation. second phase is after the completion of software. This is system testing which verifies that the whole set of programs hanged together.

**20.1.1 WHITE BOX TESTING**

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. All logical parts of the software checked once. Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing tests all the independent part within a module a logical decisions on their true and the false side are exercised , all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

**20.1.2 BLACK BOX TESTING**

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

**20.1.3 ALPHA TESTING**

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

**20.1.4 BETA TESTING**

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to thedevelopers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

**20.1.5 UNIT TESTING**

Each module is considered independently. It focuses on each unit of software as implemented in the source code. It is white box testing.

**20.1.6 INTEGRATION TESTING**

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. Modules are integrated by using the top down approach.

**20.1.7 VALIDATION TESTING**

Validation testing was performed to ensure that all the functional and performance requirements are met.

**20.1.8 SYSTEM TESTING**

It is executing programs to check logical changes made in it with intention of finding errors. a system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

**20.2 IMPLEMENTATION AND SOFTWARE SPECIFICATION**

**TESTINGS**

**20.2.1 DETAILED DESIGN OF IMPLEMENTATION**

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

**20.2.2 TECHNICAL DESIGN**

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

**20.2.3 TEST SPECIFICATIONS AND PLANNING**

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

**20.2.4 PROGRAMMING AND TESTING**

This activity encompasses actual development, writing, and testing of program units or modules.

**20.2.5 USER TRAINING**

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

**20.2.6 ACCEPTANCE TEST**

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

**20.2.7 INSTALLATION PHASE**

In this phase the new computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

**20.2.8 SYSTEM INSTALLATION**

The process of starting the actual use of a system and training user personnel in its operation.

**20.2.9 REVIEW PHASE**

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized Transystem in terms of benefits and savings projected at the start of the project.

**20.2.10 DEVELOPMENT RECAP**

A review of a project immediately after completion to find successes and potential problems in future work.

**20.2.11 POST-IMPLEMENTATION REVIEW**

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

**20.3 STEPS IN UNIT TESTING**

The steps involved during Unit testing are as follows:

a. Preparation of the test cases.

b. Preparation of the possible test data with all the validation checks.

c. Complete code review of the module.

d. Actual testing done manually.

e. Modifications done for the errors found during testing.

f. Prepared the test result scripts.

The unit testing done included the testing of the following items:

1. Functionality of the entire module/forms.

2. Validations for user input.

3. Checking of the Coding standards to be maintained during coding.

4. Testing the module with all the possible test data.

5. Testing of the functionality involving all type of calculations etc.

6. Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, We integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

**20.4 STEPS IN SYSTEM TESTING**

• Integration of all the modules/forms in the system.

• Preparation of the test cases.

• Preparation of the possible test data with all the validation checks.

• Actual testing done manually.

• Recording of all the reproduced errors.

• Modifications done for the errors found during testing.

• Prepared the test result scripts after rectification of the errors.

The System Testing done included the testing of the following items:

1. Functionality of the entire system as a whole.

2. User Interface of the system.

3. Testing the dependent modules together with all the possible test data scripts.

4. Verification and Validation testing.

5. Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

**20.5 SPECIAL CATEGORY TEST**

There are other six tests, which fall under special category. They are described below:

**• Peak Load Test:** It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.

**• Storage Testing:** It determines the capacity of the system to store transaction data on a disk or in other files.

**• Performance Time Testing:** it determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.

**• Recovery Testing:** This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.

**• Procedure Testing:** It determines the clarity of documentation on operation and uses of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.

**• Human Factors Testing:** It determines how users will use the system when processing data or preparing reports.

**21. SYSTEM ANALYSIS**

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the System to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

**22. RESULT**

The result of this project was a successful high-fidelity prototype of a Salon management and scheduling application. I successfully implemented all of the above mentioned use cases, and achieved clean interfaces that further enhance the user experience.

The many-to-one database tables successfully represent a robust set of scenarios and relationships that are critical to salon management. WebCalendar was integrated well to provide salons with full calendaring functionality. The aesthetic customization of WebCalendar makes it undifferentiated from this application.

For users, scheduling was the heart of the application. And indeed, users can currently select a salon, stylist, date, and service to make an appointment. They can then view or cancel appointments, and rate salons on past appointments. Additionally, I achieved the goal of making this app a one-stop hub for beauty parlour related information, including Salon search and contact information, peer reviews of nearby salons and stylists, and easy viewing of relevant promotions that salons are advertising. Together, all of this functionality gives users an unmatched value proposition for booking salon appointments. Users also accumulate points after attending and rating an appointment, which, upon launch of the site, would be redeemable for salon credit. This is an additional source of incentive to use this application.

For more results, see video screen captures of user and salon interfaces in action.

**23. CONTRIBUTION**

The contributions of this project to my knowledge base are extensive. Before starting, I had no previous experience with web development or any of the tools used for this application.

Database Design is an invaluable aspect of the process that I mastered in the early stages of the project. Building many-to-one relationships and robust databases were required in developing this app. Familiarizing myself with SQL and database languages is also very useful going forward.

I also take away the combined understanding of PHP, MYSQL, and HTML, and how they come together to create a rich, database-backed web application.

Finally, web applications can become more than the sum of their parts with the successful integration of open source tools. Under limited time and resources, there is only so much one person can develop. However, by identifying appropriate open source projects and integrating them seamlessly, an application can become much more extensive and functional. Integrating WebCalendar, and widgets like the JavaScript date picker are perfect examples of this. If integrated poorly, these tools are obvious and detract from the user experience. Fortunately, I was able to integrate these tools well and package more than the sum of its parts.

In terms of contributions to the field, this prototype stands as a proof of concept for the idea of web-based appointment scheduling of any kind. Doctors’ offices, dentist offices, salons, and many other industries are in great need of online, on demand scheduling. This project proves that it is very possible to build this functionality with the above-mentioned tools at very low cost. All tools used were open source.

**24. FUTURE DIRECTION**

Like I mentioned earlier, full integration of open source tools is critical in creating a seamless overall application. There are some ways in which WebCalendar was not integrated which I would like to pursue given more time.

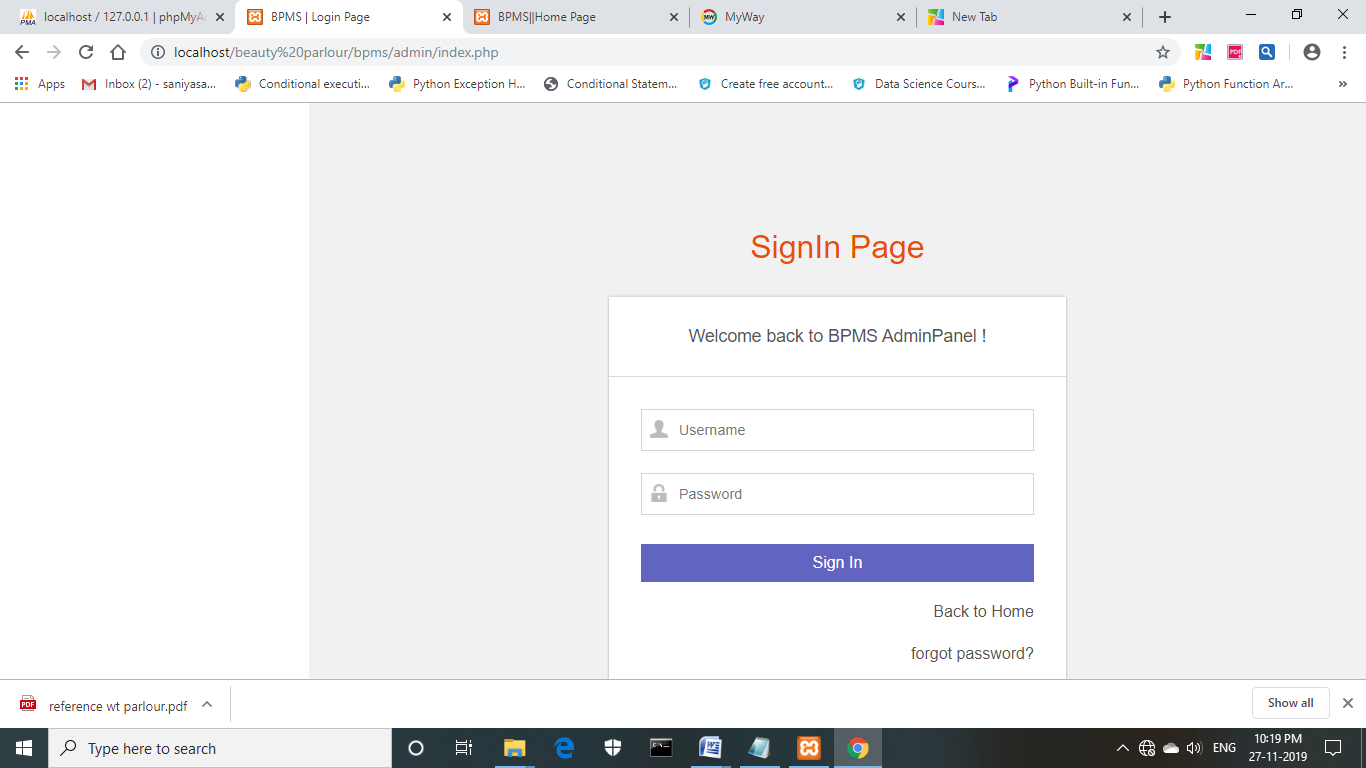
The first is to automate WebCalendar account creation and login when those tasks are completed for this application. This would require inserting a new Salon user into the appropriate WebCalendar database tables when a salon creates a user account. Of course, this would also ensure that the WebCalendar account information matches their account information. The same for login would be an important improvement. As it exists now, salons have to log in to the WebCalendar application additionally within the Beauty parlour management system when they navigate to it via the Dashboard. Separate sign in is only required once per session though, as the login is stored throughout a session even as they go back and forth from WebCalendar to this application. Even so, automated account creation and login are important integration features that would be next in line, given more time.

Also, a constraint of WebCalendar was that it only allowed a user to have one calendar. Ideally, a salon would have one calendar for each stylist. This could be achieved using an automated WebCalendar account creation script with a well-defined prefix naming system. Salons would be oblivious to what is actually going on behind the scenes, and simply be able to navigate to calendars that are tied to each stylist. When a salon would add a stylist, another WebCalendar account (and corresponding calendar) would be created, using a prefix of the stylist name and salon name. This would dramatically increase the power of the application as a schedule visualization and distribution tool. Availability would be more apparent, and schedules could be printed for each stylist at the start of the day.

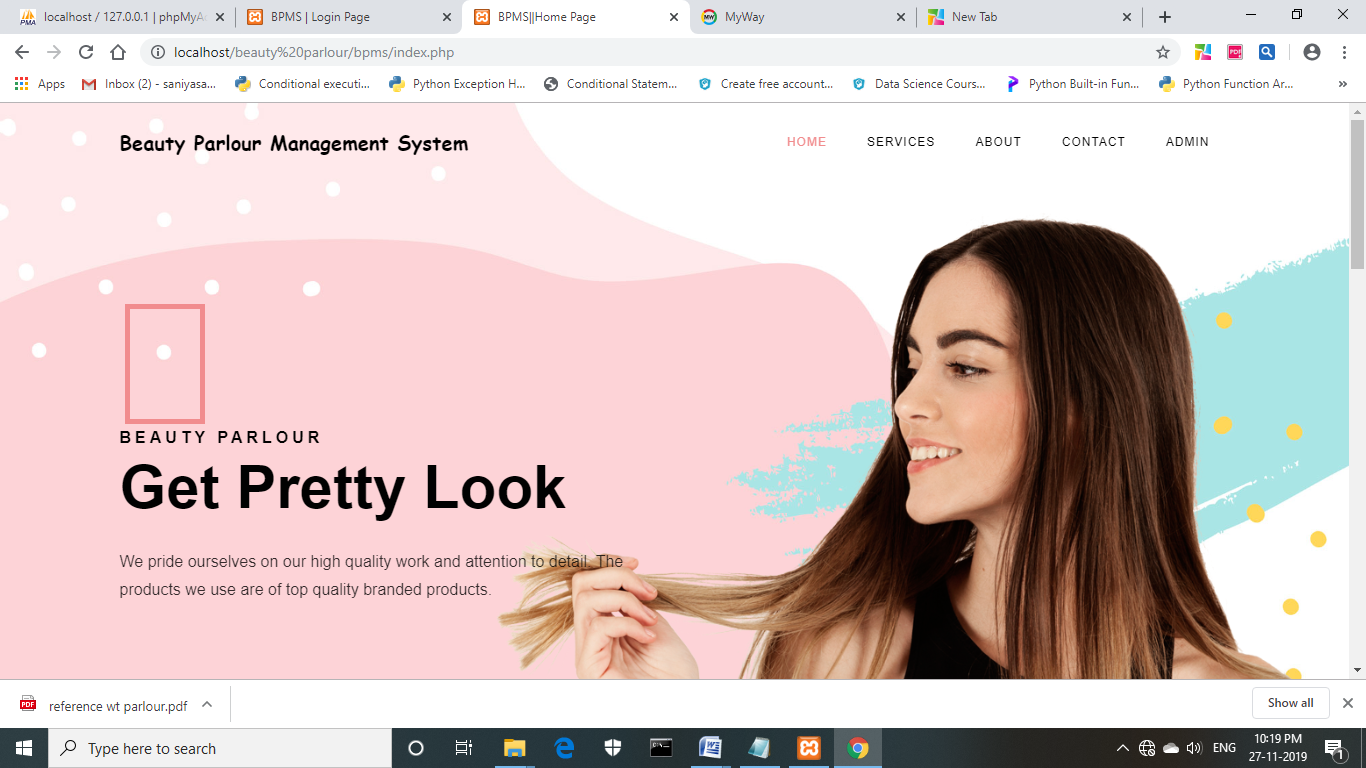
Additionally, I would like to host the prototype online and run usability tests with surveys to gauge the effectiveness of the interface and functionality. This is a key step to perfecting the front-end interface and the functionality of the tools.

Finally, I would like to explore more dynamic, Web 2.0 tools like JavaScript, jQuery, and OpenLaszlo in more depth with the hope of making the interface richer for the user. Over the course of this project, I was able to get a small taste of these web development tools and their capabilities. They would allow me to give my application a more desktop software-feel on the web, with features like animation and drag-and drop. Unfortunately, I did not have enough time to fully explore them, and so that would be an addition improvement to consider going forward.

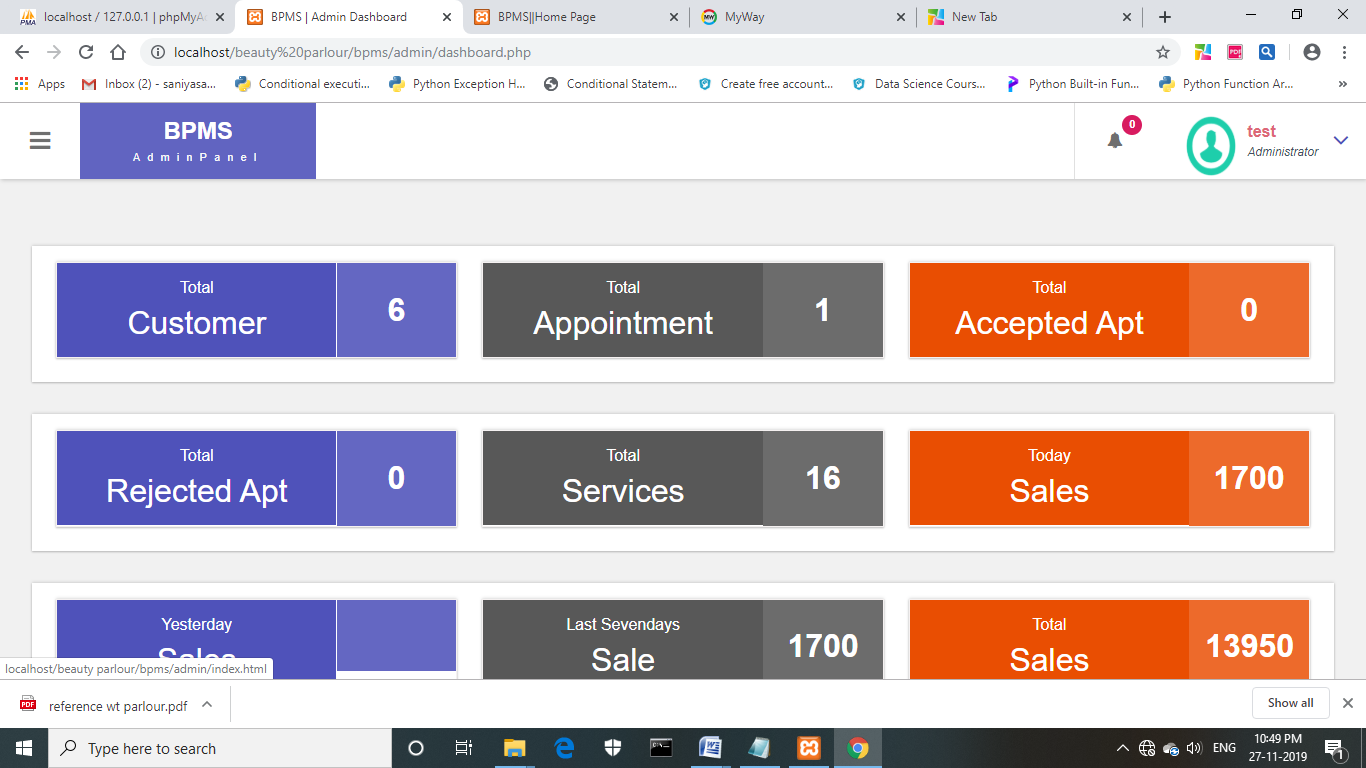
**25. SNAPSHOTS**

****

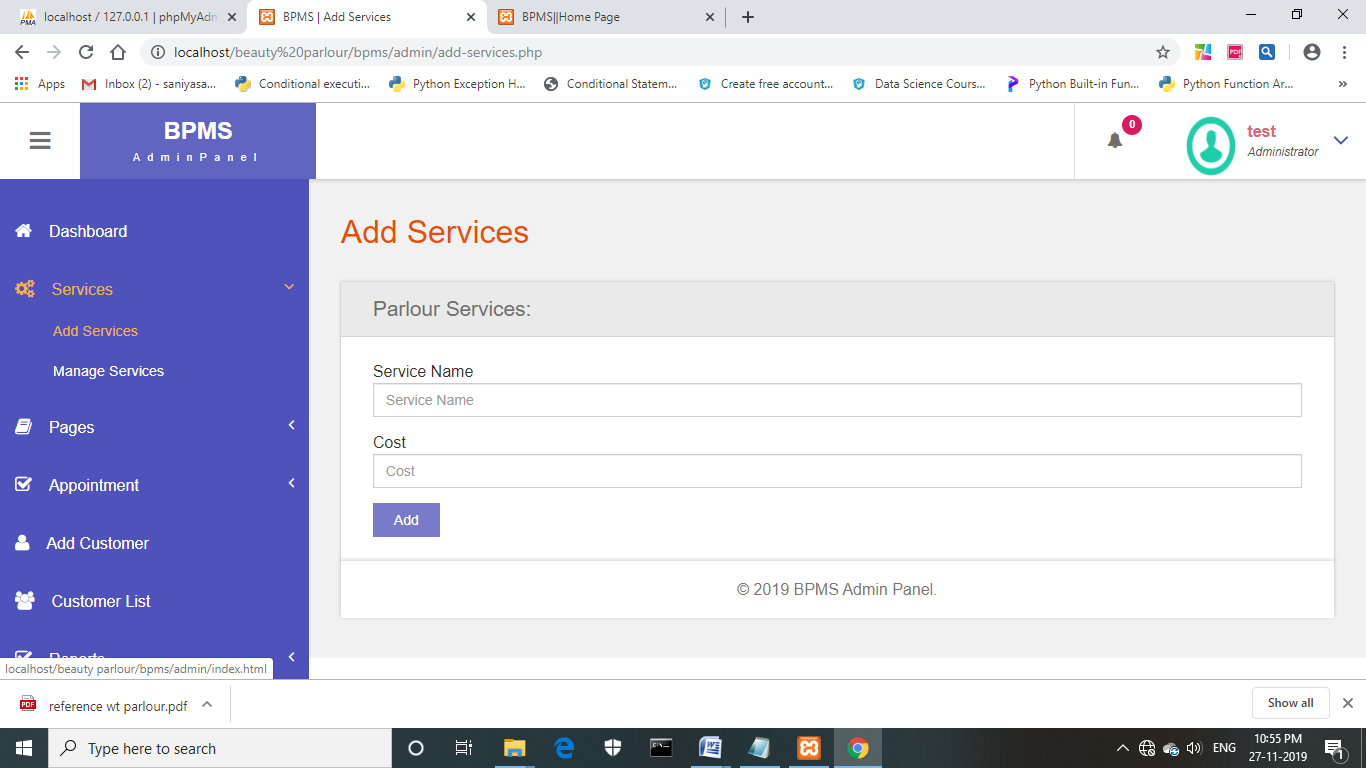
**FIG : OUTPUT WINDOW FOR ADMIN LOGIN PAGE**

****

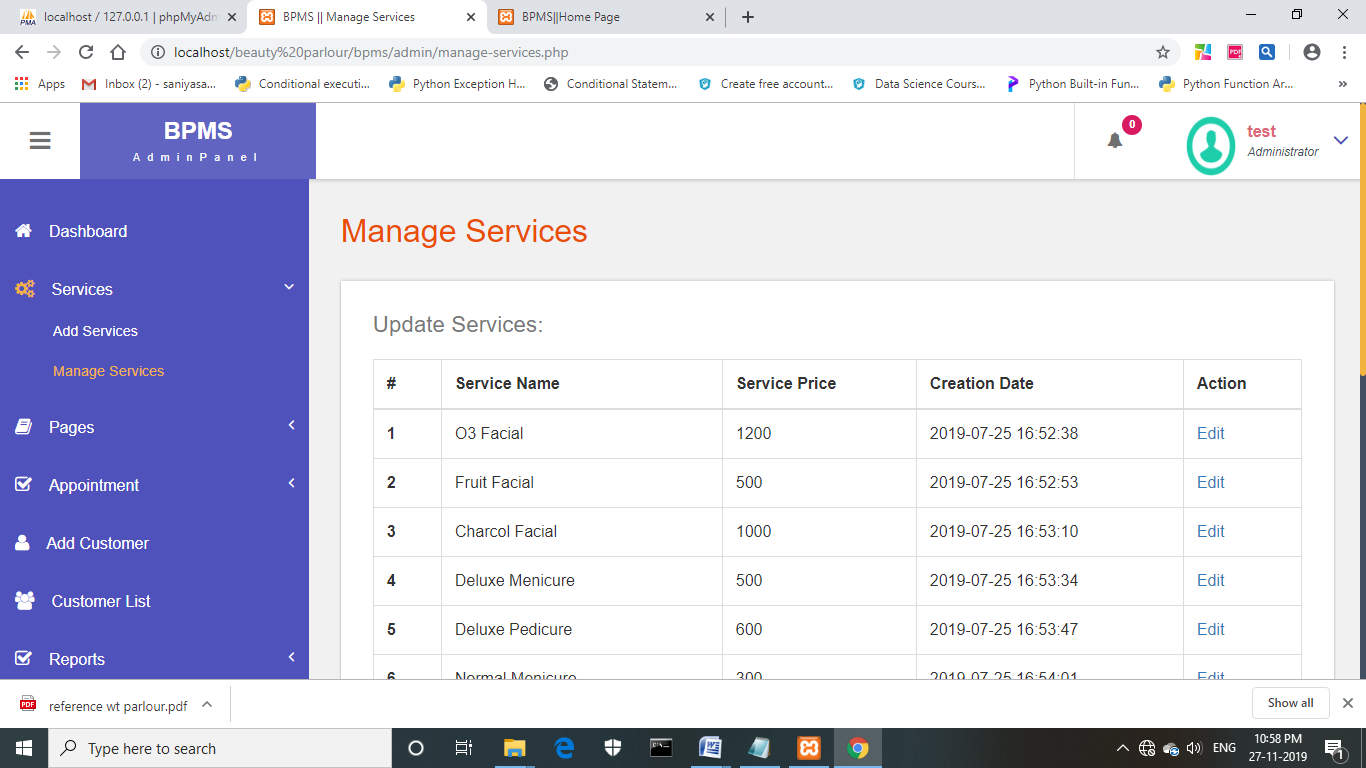
**FIG : OUTPUT WINDOW FOR CUSTOMER LOGIN**

****

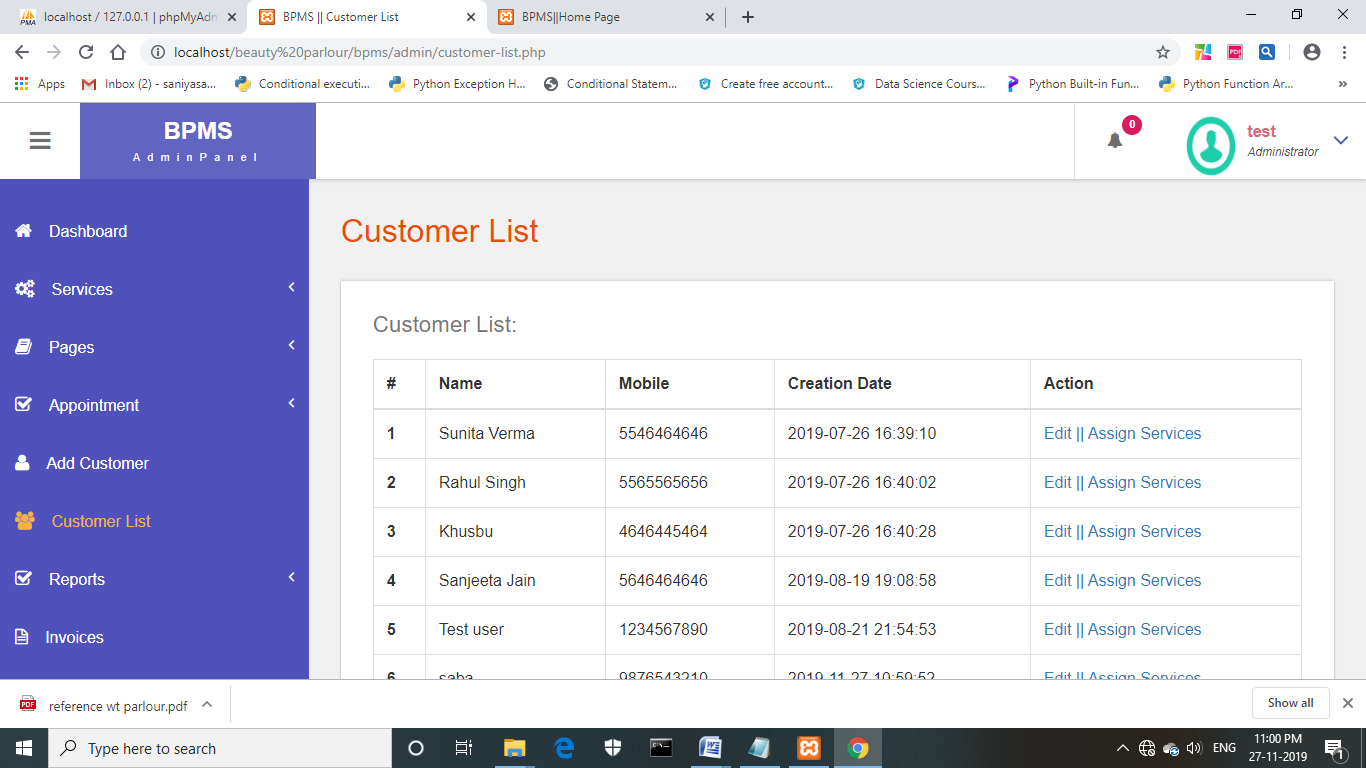
**FIG : OUTPUT WINDOW FOR ADMIN HOMEPAGE**

****

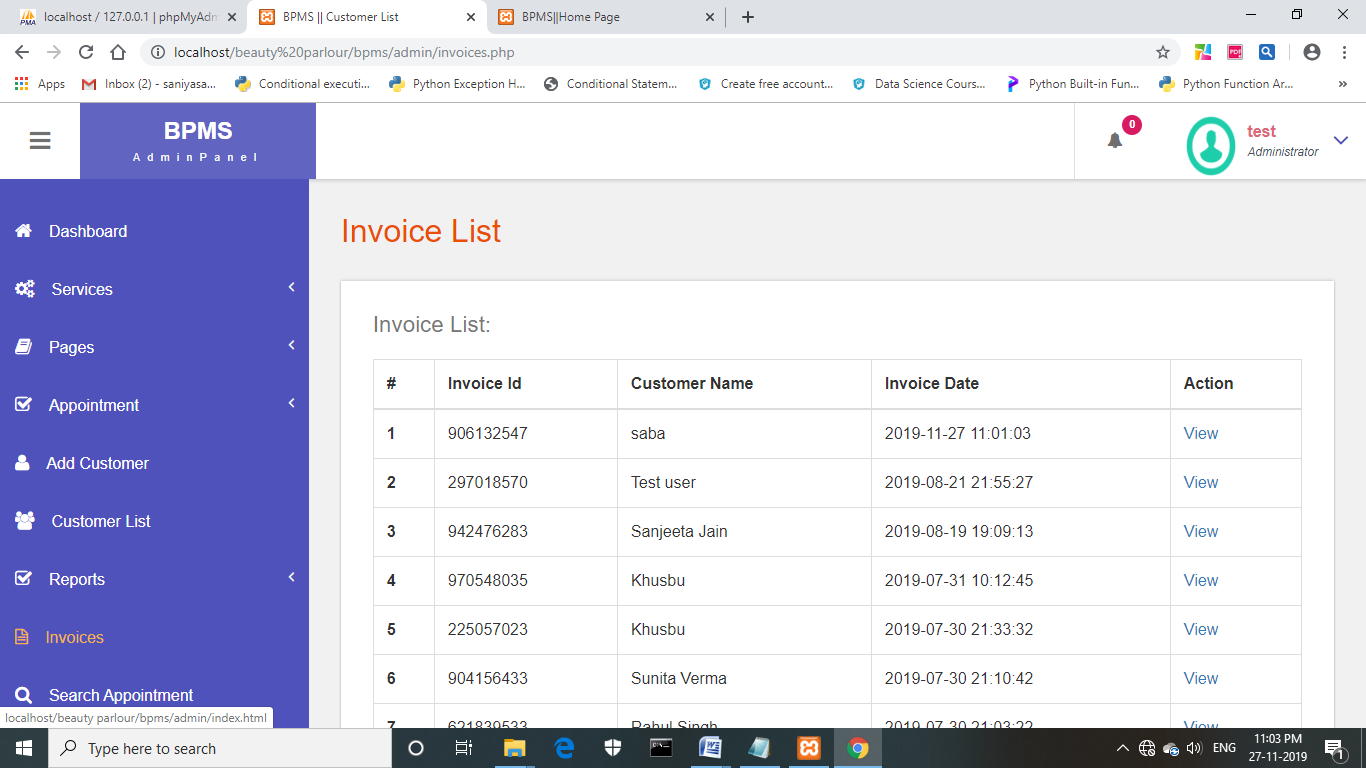
**FIG : OUTPUT WINDOW FOR ADDING SERVICES**

****

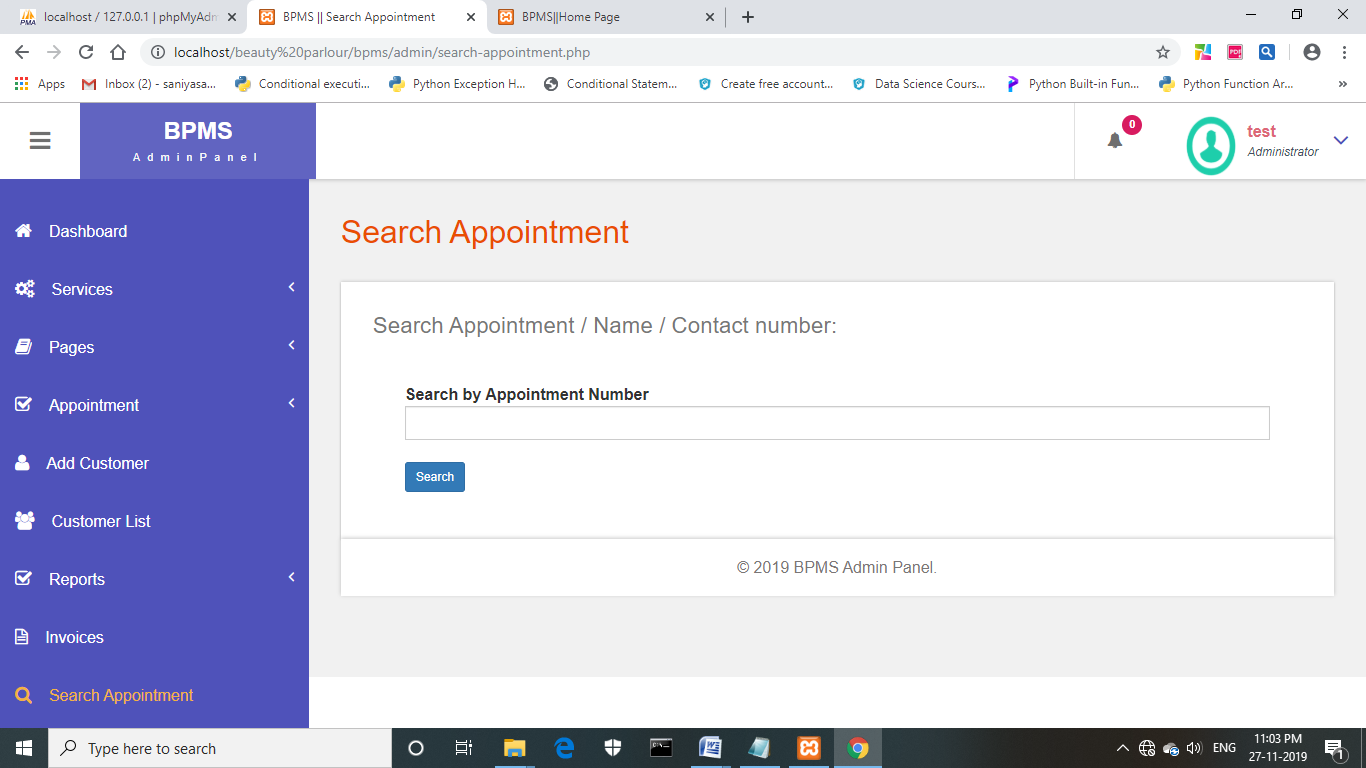
**FIG : OUTPUT WINDOW TO MANAGE SERVICES**

****

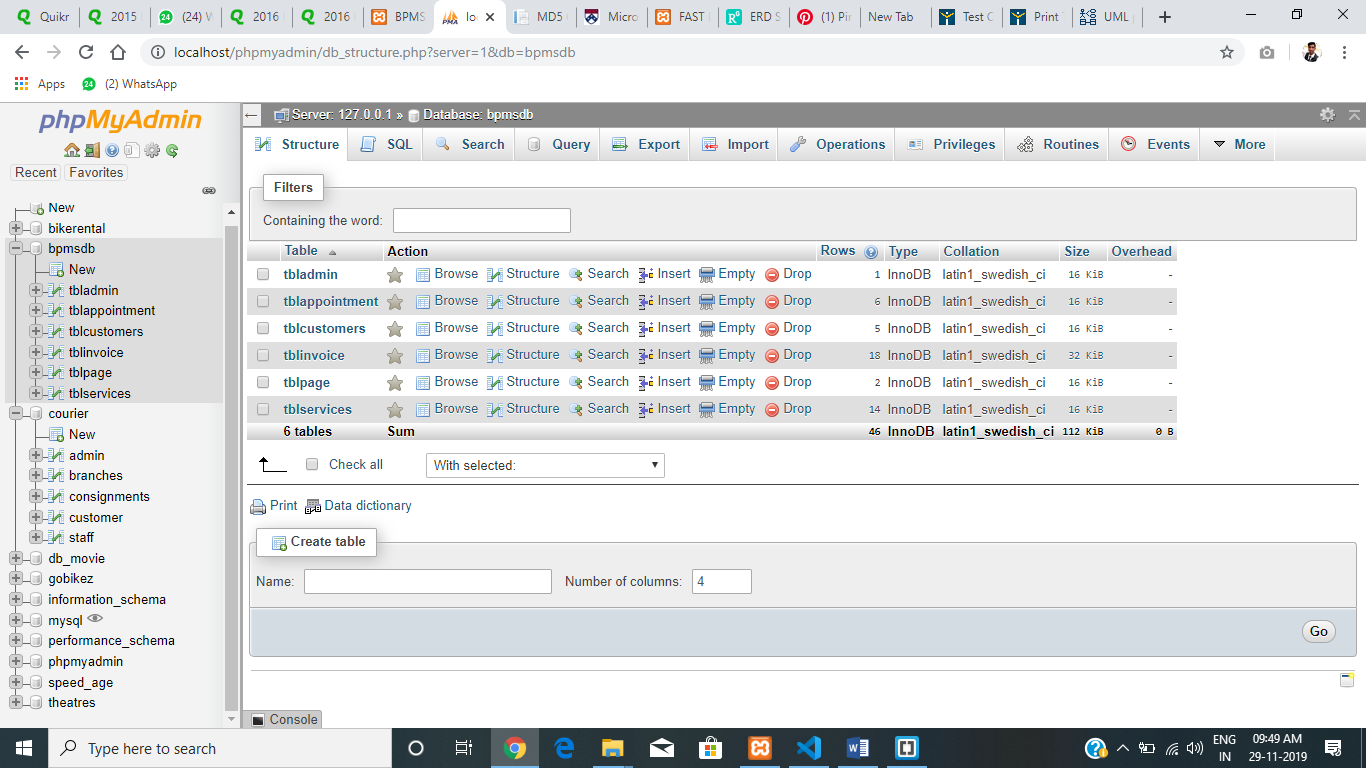
**FIG : OUTPUT WINDOW FOR LIST OF CUSTOMERS**

****

**FIG : OUTPUT WINDOW FOR THE INVOICE LIST**

****

**FIG : OUTPUT WINDOW FOR SEARCHING THE APPOINTMENT**



**FIG : TABLES**

#### 26. REFERENCES

* PHP And MySQL Web Development

By: Like Welling

PLATINUM EDITION

* PHP Solutions: Dynamic Web Design Made Easy

By: David

SECOND EDITION

* PHP For The Web: Visual Quick Start Guide

By: Larry Ullman

FOURTH EDITION

**Web references**

* www.Google.com
* www.Codeguru.com
* [www.Saveearth.org](http://www.saveearth.org/)
* [www.w3schools.com](http://www.w3schools.com)
* [www.ionic.com](http://www.ionic.com)
* [www.stackoverflow.com](http://www.stackoverflow.com)
* <https://spectrum-a6a15.firebaseapp.com/>