A blue and white logo

Description automatically generated**JAGAN INSTITUTE OF MANAGEMENT STUDIES SECTOR – 5, ROHINI, NEW DELHI**

**(Affiliated to)**

**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY SECTOR – 16 C, DWARKA, NEW DELHI**

MINOR PROJECT FILE

**Submitted To:** **Dr. Sanjive Saxena Submitted By: SALONI SINGH**

**Professor (IT) Enrollment No.:****05414004423**

**MCA 1st Year (Shift-1)**

**1st Semester**

**DECLARATION**

This php project titled One Touch Parlor Management Website towards the completion of my course requirements for Semester I is my original work and has been carried out under the guidance of Dr. Sanjive Saxena.

The material borrowed from other sources and incorporated in the report has been duly acknowledged and referenced.

I understand that I will be held liable and accountable for my project, it is all done by me without having any copyright(s) of any of the organizations. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

#### DATE:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### SALONI SINGH (05414004423)

**CERTIFICATE**

This is to certify that the project entitled: One Touch Parlor Management Website done by Miss Saloni Singh having university enrolment number 05414004423 is an authentic work carried out by them at JAGAN INSTITUTE OF MANAGEMENT STUDIES SECTOR – 5, ROHINI

The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief. The suggestion as approved by the faculty was duly incorporated.

#### Date:

#### Dr. Sanjive Saxena

**Professor (IT)**

**ACKNOWLEDGEMENT**

We would like to express our heartfelt gratitude to JAGAN INSTITUTE OF MANAGEMENT STUDIES and Dr. Sanjive Saxena for giving us the opportunity to work under their guidance and helping us gain immense professional experience.

A sincere thanks to our mentor for giving us valuable input and ideas right from the selection of the topic for the project till its successful completion.

The successful completion of our project would not have been possible without the dedicated support from all our mentors, family, and friends.

#### Name: Saloni Singh

**Enrollment No:**

**05414004423**

## ABSTRACT

OneTouch Parlour is a well-recognized and reputed parlour among people. They have three parlours around Karol Bagh and many areas. Day by day their customer base is growing, and the current manual process is not efficient enough to provide quick and reliable service to them. Also, hard to maintain parlour internal processes. Issuing reliable reports is much needed, Because of these parlor owners willing to manage their parlors from one place and also to keep their well reputation among parlours and peoples much needed.

They have no automation system for their process. Even if they use computers and tablets they do not have a proper system to fully fill their needs. They are using a record book to track their day-by-day process in parlours. This is very inefficient and unreliable. And keeping records not accurate with relevant parlor data when calculating reports. And also salary calculation of employees is not easy.

This system will provide the functionalities of managing parlors’ details, managing day by day bookings, managing issuing many reports (SERVICES report, booking report etc.), managing service, billing, managing employee details, managing customer details and managing notification systems for day-to-day transactions. This system will help to improve the higher efficiency of the processes in the parlours. It will enable parlour staff to add data to the system without bothering about creating report formats that will increase the efficiency of parlour staff.

The project was implemented using Visual Studio as the IDE and MySQL server used as the DBMS. Coding was done using the php programming language and JavaScript.

The system will provide a user-friendly simple interface which will help them to provide reliable, quick and effective service to their customers.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **Chapter no.** | **TOPIC** | **Page no.** |
| **1.** | **INTRODUCTION** | 11 |
| 1.1 | Introduction to project | 12 |
| **2.** | **REQUIREMENT ANALYSIS** | 14 |
| 2.1 | System Requirements | 15 |
| 2.2 | Analysis | 16 |
| 2.3 | Use Case Diagram | 17 |
| **3.** | **SOFTWARE DESIGN** | 19 |
| 3.1 | Software design | 20 |
| 3.2 | DFD Level 0 | 21 |
| 3.3 | DFD Level 1 | 22 |
| 3.4 | DFD Level 2 | 23 |
| **4.** | **DATABASE DESIGN** | 24 |
| 4.1 | Database design and er relationship | 25 |
| 4.2 | ER diagram | 26 |
| **5.** | **TESTING** | 28 |
| 5.1 | Testing And Test plan | 29 |
| 5.2 | Test Strategies | 31 |
| **6.** | **ROLES AND RESPONSIBILITIES** | 36 |
| **7.** | **CONCLUSION APPENDICES** | 38 |
|  | CODING AND SCREENSHOT | 40 |
|  | BIBLIOGRAPHY | 51 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| Figure 2.1: Use Case Diagram For Current Manual System | 17 |
| Figure 3.1: DFD Level 0 | 21 |
| Figure 3.2: DFD Level 1 | 22 |
| Figure 3.3: DFD Level 2 | 23 |
| Figure 4.1: Er Diagram For Current System | 26 |
| Figure 4.2: Entity relationship tables diagram | 27 |
| Figure 5.1: Service Category View | 34 |
| Figure 5.2: Validation for required fields | 34 |
| Figure 5.3: Delete confirmation message | 35 |
| Figure 5.4: Serviced add confirmation message | 35 |

**LIST OF TABLES**

|  |  |
| --- | --- |
| Table 3.1: Use case description for System Login | 15 |
| Table 3.2: Use case description for Generate Parlour’s Reports | 16 |
| Table 5.1: Test Cases for Service | 30 |
| Table 5.2: Test Cases for appointment | 31 |
| Table 5 1: Test result for booking | 35 |

## LIST OF ACRONYMS

|  |  |
| --- | --- |
| OSMS | - Online Parlour Management System |
| RAM | - Random Access Memory |
| HDD | - Hard disk drive |
| Gb | - Gigabyte |
| RAD | - Rapid Application Development |
| OOAD | - Object Oriented Analysis and Designing |
| ORM | - Object Relational Mapper |
| DBMS | - Database Management System |
| UML | - Unified Modelling Language |
| ER | - Entity-Relationship |
| IIS | - Internet Information Services |
| IDE | - Integrated Development Environment |
| API | - Application Programming Interface |
| CSS | - Cascading Style Sheets |
| ISO | - International Organization for Standardization |
| IEC | - International Electrotechnical Commission |
| FCL | - Framework Class Library |
| CLR | - Common Language Runtime |

**CHAPTER 1: INTRODUCTION**

##### Background

Day by day modern beauty culture is getting rapidly popular among the people, especially among young women. Also they have a habit of preferring a usual one beautician. So the parlour owners are eager to provide their services at its best to the customers to make sure they remain with parlour as long as possible. In this case they are looking for the best way to manage their parlours in more efficiencies and keep customers attractive to parlours. Day by day their customer base is growing and the current manual process is not efficient enough to provide quick and reliable service to them. Managing reliable booking is much needed also generate reports, Because of these bookings directly involve in day by day parlours service

##### Motivations For The Project

* Look Well Parlour is a well-recognized and reputed group of parlours in the area.
* They have no automated parlour management system for managing their parlour process.
* Most of the employees are using smartphones and tablets, so eventually they got the basic idea about how to use web browsers and access the internet and work on it. And they can easily adapt to this online web base parlour management system with simple system introduction of how to use it.
* The system will provide user friendly simple interface which will help them to make reliable, quick and effective service

##### Scope of the Project

The proposed system will provide manage day to day parlour process easily

* Manage service details :-

Every parlour offers a veracity of services related to beauty culture. Parlour has to maintain a service list with the price and service details. It’s also helpful for parlour employees to deal with customers.

Especially when creating bills. To fully fill this manage service module will be added to the system.

* Manage bookings by parlour :-

Parlour’s customers can directly call parlour and parlour admin should create a booking for

that customer with specific service and time. Additionally booking can assign specific employees.

* Manage invoice :-

Customers had to pay the full billed amount when the service was received. In this case the parlour front desk should be able to generate bills for customers. And the payment details for the bill should be recorded into the system.

* Manage customer details :-

Their customers are mostly on a regular basis. So they are planning to keep historical data to evaluate who got service and product from their parlours. For this they need to keep customers' details against the issued reports' historical data

## Chapter 2: Requirements and Analysis

##### System Requirements

At the end of the project the system should work on the customer's environment. There are requirements in Hardware and Software to run the new system.

**Requirements for Web Hosting**

Web server requirements for hosting the application.

* XAMPP v.3.0 server
* 2Gb RAM

**End User Requirements**

Hardware requirements for this system as given below,

Computer to run web browser.

**Requirements For Software**

* Software requirements for this system is that any web browser runs on any operating system.
* Recommended Browser – Chrome
  1. **Analysis**

This section will cover the process of discovering, analysing, defining, and documenting the requirements that are related to the business.

Requirements analysis in systems engineering and software engineering encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, analysing, documenting, validating and managing software or system requirements [1].

###### Current Manual System

They have been using Microsoft Word and Excel files to keep their day to day record. They

keep records of day-to-day use items and equipment in excel files which maintain separate files in each parlour. And also each day they are creating separate word files as a summary for daily income. Once the owner visits the parlour he collects the files and does calculations to get a rough idea about the parlour income and expenses. This method is very reliable, insufficient and insecure. Even though they use computers and printers they do not have a proper system to fulfil their needs.

**Weaknesses Of Current Manual System**

When carried out further analysis of current manual system some weaknesses were identified as follows,

* No proper way to manage customers’ appointments.
* Cannot analyse their historical data to make predictions.

Very often these files get attacked by computer viruses and lost information.

* Anyone can access these files and parlour’s confidential details not protect

from unauthorised access.

* Took more time to prepare the bill.

###### Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicts the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

A diagram of a person with a person in the middle

Description automatically generated

Figure 2.1: Use Case Diagram for Parlour Admin

|  |  |
| --- | --- |
| Actor | All Employees |
| Overview | |
| Admin can login to the system | |
| Precondition | |
| Admin should have Username and Password | |
| Flow of Events | |
| Enter Login Details, Validate Login details, Login to the system | |
| Post Condition | |
| Invalid Admin will get the error message and reject login. Valid Admin will get the main window of the system | |

Table 2.2: Use case description for Admin Login

|  |  |
| --- | --- |
| Use Case | Generate Parlour’s Reports |
| Actor | Parlour Admin |
| Overview | |
| Generate Parlour’s Reports | |
| Precondition | |
| Admin should login to the system under authorised user type | |
| Flow of Events | |
| Generate parlour report using veiled parameters | |
| Post Condition | |
| Relevant Report should be shown. | |

Table 2..3: Use case description for Generate Parlour1’s Report

## Chapter 3: Software Design

* 1. **Software Design**

Software design is the process of envisioning and defining software solutions to one or more sets of problems. One of the main components of software design is the [software](https://en.wikipedia.org/wiki/Software_requirements_analysis) [requirements analysis](https://en.wikipedia.org/wiki/Software_requirements_analysis) (SRA). SRA is a part of the [software development process](https://en.wikipedia.org/wiki/Software_development_process) that lists [specifications](https://en.wikipedia.org/wiki/Specifications) used in [software engineering](https://en.wikipedia.org/wiki/Software_engineering). If the software is "semi-automated" or [user](https://en.wikipedia.org/wiki/User_centered_design) [centred](https://en.wikipedia.org/wiki/User_centered_design), software design may involve [user experience design](https://en.wikipedia.org/wiki/User_experience_design) yielding a [storyboard](https://en.wikipedia.org/wiki/Storyboard) to help determine those specifications. If the software is completely [automated](https://en.wikipedia.org/wiki/Automation) (meaning no [user](https://en.wikipedia.org/wiki/User_(computing)) or [user interface](https://en.wikipedia.org/wiki/User_interface)), a software design may be as simple as a [flow chart](https://en.wikipedia.org/wiki/Flow_chart) or text describing a planned sequence of events. There are also semi-standard methods like [Unified](https://en.wikipedia.org/wiki/Unified_Modeling_Language) [Modelling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) and [Fundamental modelling concepts](https://en.wikipedia.org/wiki/Fundamental_modeling_concepts). In either case, some [documentation](https://en.wikipedia.org/wiki/Documentation) of the plan is usually the product of the design. Furthermore, a software design may be [platform-independent](https://en.wikipedia.org/wiki/Platform-independent_model) or [platform-specific](https://en.wikipedia.org/wiki/Platform-specific_model), depending upon the availability of the technology used for the design.

Data Flow Diagram

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expanding it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

* + - Logical information flow of the system
    - Determination of physical system construction requirements
    - Simplicity of notation
    - Establishment of manual and automated systems requirements
  1. **DFD diagram Level 0**

A circle with black text

Description automatically generated

Figure 3.1: DFD Level 0

### DFD diagram Level 1

A diagram of a system

Description automatically generated

Figure 3.2: DFD Level 1

### DFD diagram Level 2

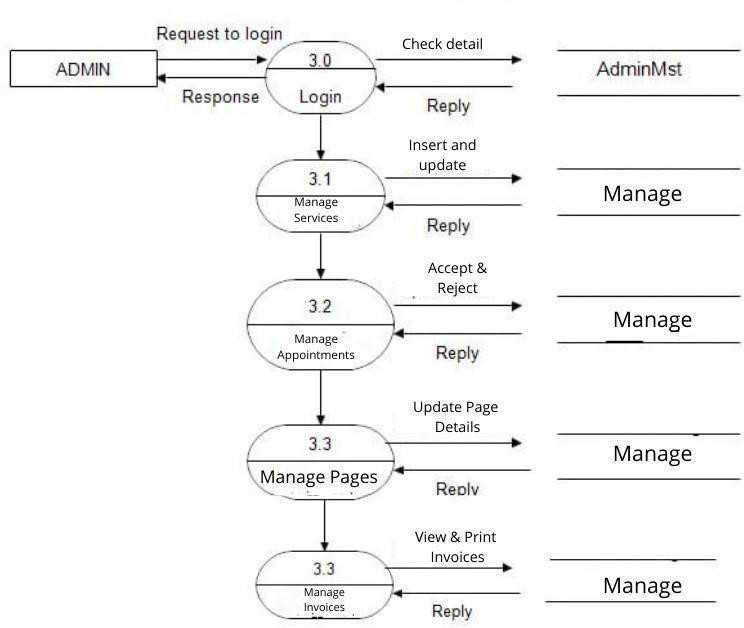


Figure 3.3: DFD Level 2

# Chapter 4: Database Design

###### Database Design And Entity Relationship Diagrams

In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them [8]. A database is a more and more important part of the information system. Database design is achieved through data modelling. Following Figure 4.1 is the Entity relationship diagram for the new system.

###### ER Diagram

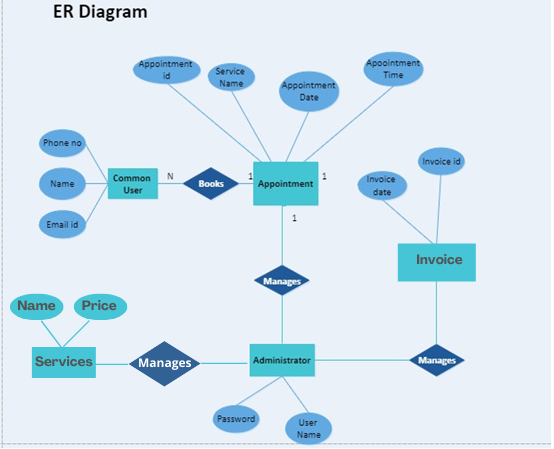


Figure 4.1: Entity Relationship Diagram for the System

A diagram of a computer

Description automatically generated

Figure 4.2: Entity Relationship Tables

# Chapter 5: Testing

###### Testing

After the successful implementation of a system for its accuracy the system is needed to be verified and validated. Software testing is performed to verify that the completed software functions according to the expectations defined by the requirements. This chapter describes the test plan used in the development of the system, and a sample of a test result is also provided as evidence that the specifications have been met.

**Requirement for A Good System Evaluation**

Before the system develops into a business environment it must test in various ways to ensure the system does not gain any type of error. System testing is performed on the entire system in the context of a Functional Requirement Specification(s) and/or a System Requirement Specification. System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

These tests were helped to check,

* + - The functions developed are according to the specification.
    - The identification and reporting of errors that occur in the system and correcting them as needed.
    - Improving the performance and efficiency of the system.

##### Test Plan

The test plan describes the testing strategies and the approaches to testing quality assurance will be used to validate the quality of the system. While the coding system, it was carried out the unit testing to ensure that functionalities are working as expected. This was done by testing the code unit wise.

An integration testing has been done for the test modules in the system. It was integration of a few units of code. And all the modules together tested as a system.

###### Test Strategies

Testing to confirm all the modules work together. And also as the system users are able to access via any browsers need to be tested at least most common web browsers like Chrome, Firefox and Edge etc.

###### Unit testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process.

Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended [19].

###### Black box testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher-level testing, but can also dominate unit testing as well [20].

###### White box testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-

testing). In white-box testing an internal perspective of the system, as well as programming

skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit,

e.g. in-circuit testing.

###### Integration testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

###### System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

#### Test Case and Test Result for Service

**In table 5.1 it shows the test cases and test results for service . Mainly adding service, editing service category, deleting service category and validation tested here.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description/steps to test | Expected Result/s | Actual Result/s | Status |
| 1 | Click Service Category on left navigation in  Service Module | Service category view should be loaded in to service window | See Figure 5.1 | Pass |
| 2 | Add service category Values: Category Name- Face Description - Facial treatments  Branch - Boralesgamuwa Click save | 1-Should be saved to the database table 2-Text fields should be cleared.   1. Add Category view should be closed. 2. New row with the entered values should be added in to table   on view. | 1. Record added to the database table 2. Text fields get cleared 3- View get closed 4-New record added to the table on view | Pass |
| 3 | Add new service category-  Click save with empty text fields | 1-Please fill this field with an error message. | See Figure 5.2 | Pass |
| 4 | Add new category Values: “Name, Branch” field and click save | 1-Should be saved to the database table 2-text fields should be clear.  3-Table in view should be added a new row with entered values. | 1. Record added to the database table 2. Text fields got cleared 3-record added to the db | Pass |
| 5 | Add new test category- Values: only to “Description” text field and click  save | 1. Fields validations should be fired 2. Data should not be insert in to database table | Red colour border appear around text box validation messages shown | Pass |
| 6 | Delete service category- Click Delete button in table | 1. Delete confirmation message should be shown to the users 2. If click yes - selected category should be removed from view 3. If click no – selected category should not be removed from view | 1. See Figure 5.3 2. Deleted row removed from table in view 3. See Figure 5.4 Service Added successfully | Pass |
| 7 | Edit service Click Edit button in table | 1-selected service categories values should be filled into service category form. | Selected service category loaded to the form. | Pass |

Table 5.1: Test Cases for Service

###### Bellow Figure 5.1, 5.2, 5.3, 5.4 images are refers to “Actual Result” column in table

A screenshot of a computer

Description automatically generated

Figure 5.1: Service Category View

A screenshot of a computer

Description automatically generated

Figure 5.2: Validation for Required Fields

A screenshot of a computer

Description automatically generated

Figure 5.3: Delete Confirmation Message

A screenshot of a computer

Description automatically generated

Figure 5.4: Service Add Confirmation Message

**CHAPTER 6: ROLES AND RESPONSIBILITIES**

#### Project Roles and Responsibilities ROLE

* Work as a developer, designer, and tester of the application.

#### RESPONSIBILITIES

* Work on definition of development requirements and priorities.
* Data migration.
* Interfaces with other systems.
* Reporting configuration and deployment.
* Setup and maintenance of security rights and access permission.
* Contributing to technical strategy,policy and procedure.
* Development and operation of technical testing programmes.
* Production of technical documentation to agreed quality standards.
* Reporting on progress/issues to management and users

## CHAPTER 7: CONCLUSION

###### Overview

From start to end of this project, it was prototyped to the client to ensure that his requirement has been fulfilled by the implementation of the system in each phase. This constant checking with the client is assured that the developed system met the requirements of business that were identified in the requirement analysis phase and client requested.

By user reviewing the functional and non-functional requirements that were discovered during the analysis phase and checking back with the functionalities implemented in the developed system, it can be said that all the requirements of the user have been satisfied.

The simple user interfaces that were designed and developed are easy to learn and use proved to be satisfactory for the user.

###### Lessons Learnt

I was not familiar with the php language or Visual Studio development environment. So, I had to learn php and JavaScript frameworks to fulfil this project.

I found some great video tutorials and some examples and view them and follow examples to grab lessons.

Mostly inexperience of project planning and scheduling identify and have great advice from my supervisors to improve proper time management and planning.

**Future Enhancements**

The following future enhancements in the system are as follows.

1. Create attractive separate web sites for each branch using data collected in the current version.
2. Improve appointment booking using mobile app Enhance the features of the system

**APPENDICES: CODING AND SNAPSHOTS**

#### Homepage

A screenshot of a computer

Description automatically generated

**Book Appointment**

Create Bookings

Booking Dashboard From Appointment Form User have to share

* + Name
  + Phone no
  + Email Id

And have to select,

* + A date for booking
  + An service for booking
  + A time for booking

Then after submitting, the user gets an appointment no.

A screenshot of a computer

Description automatically generated

### ADMIN PANEL

#### System Login

Employees must be login to the system with correct credentials. In this case the system administrator password will be provided by the developer.

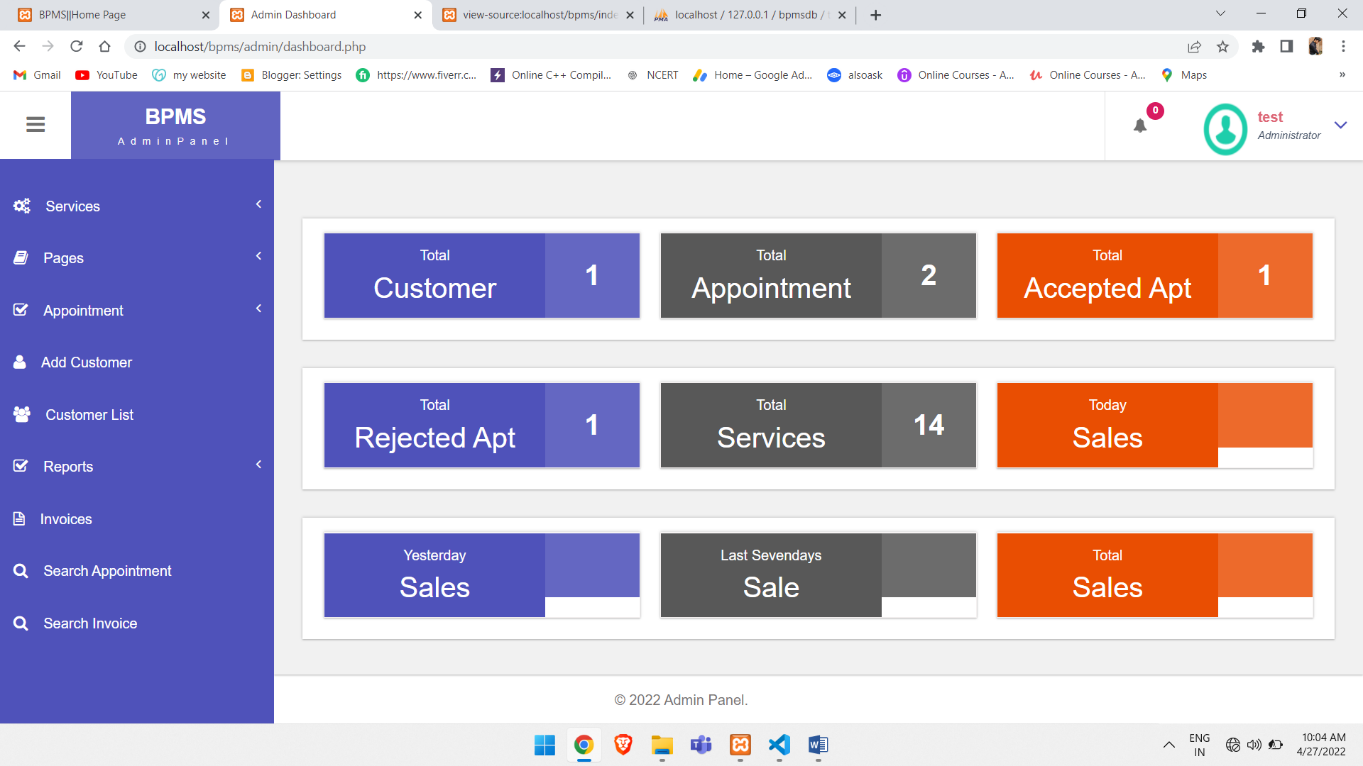
A screenshot of a login page

Description automatically generated

#### ADMIN HOMEPAGE

When the system administrator login to the system successfully it will appear the system home . On the home page the user will see all the modules and some important details in the category view.

It’s included module



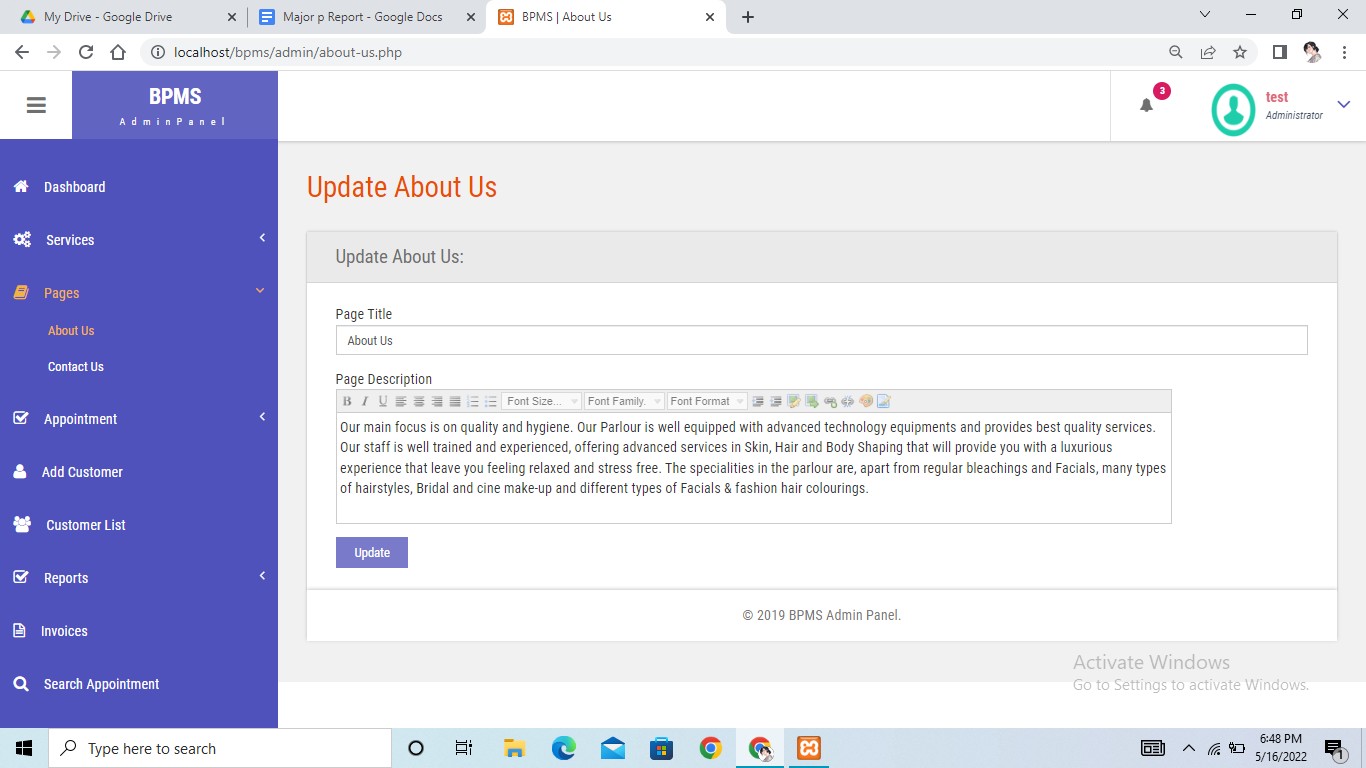
#### Services

It includes add services and manage services.

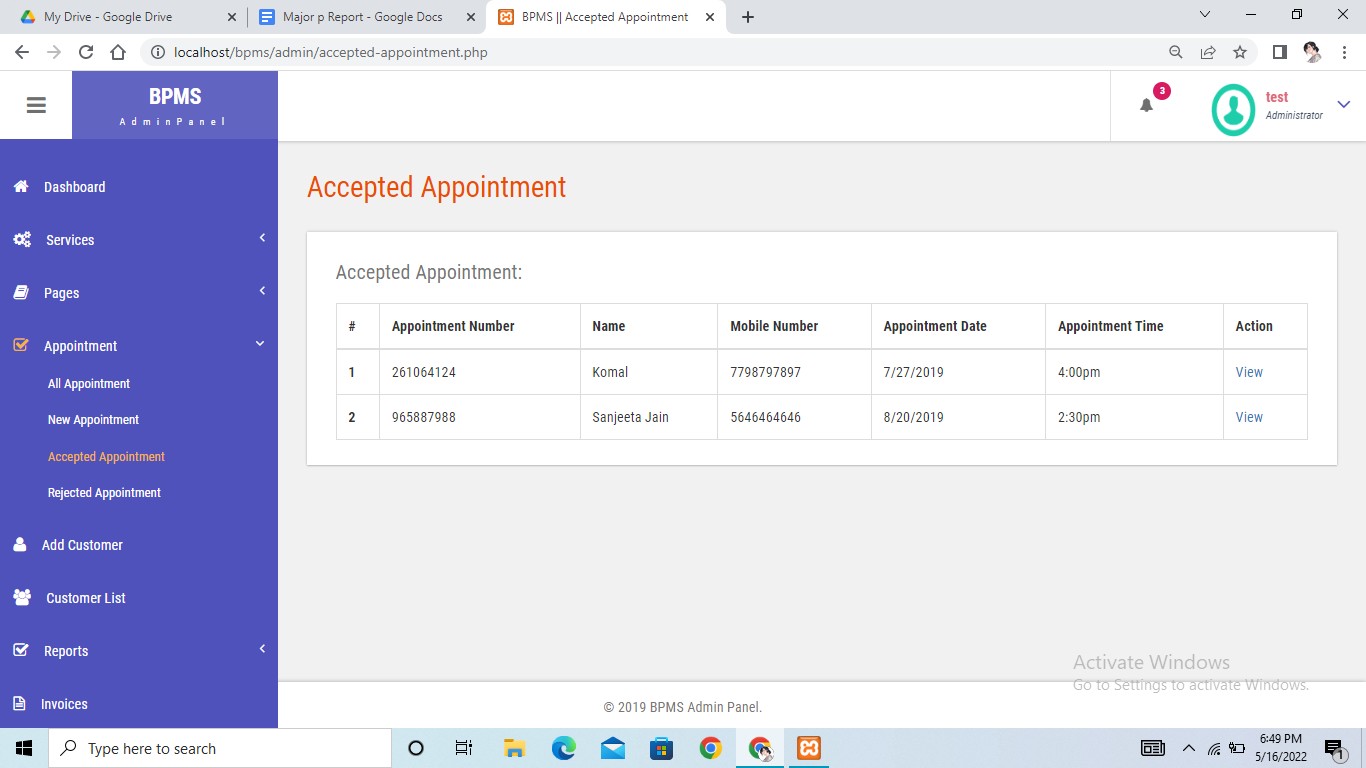
A screenshot of a computer

Description automatically generated

### Manage Pages

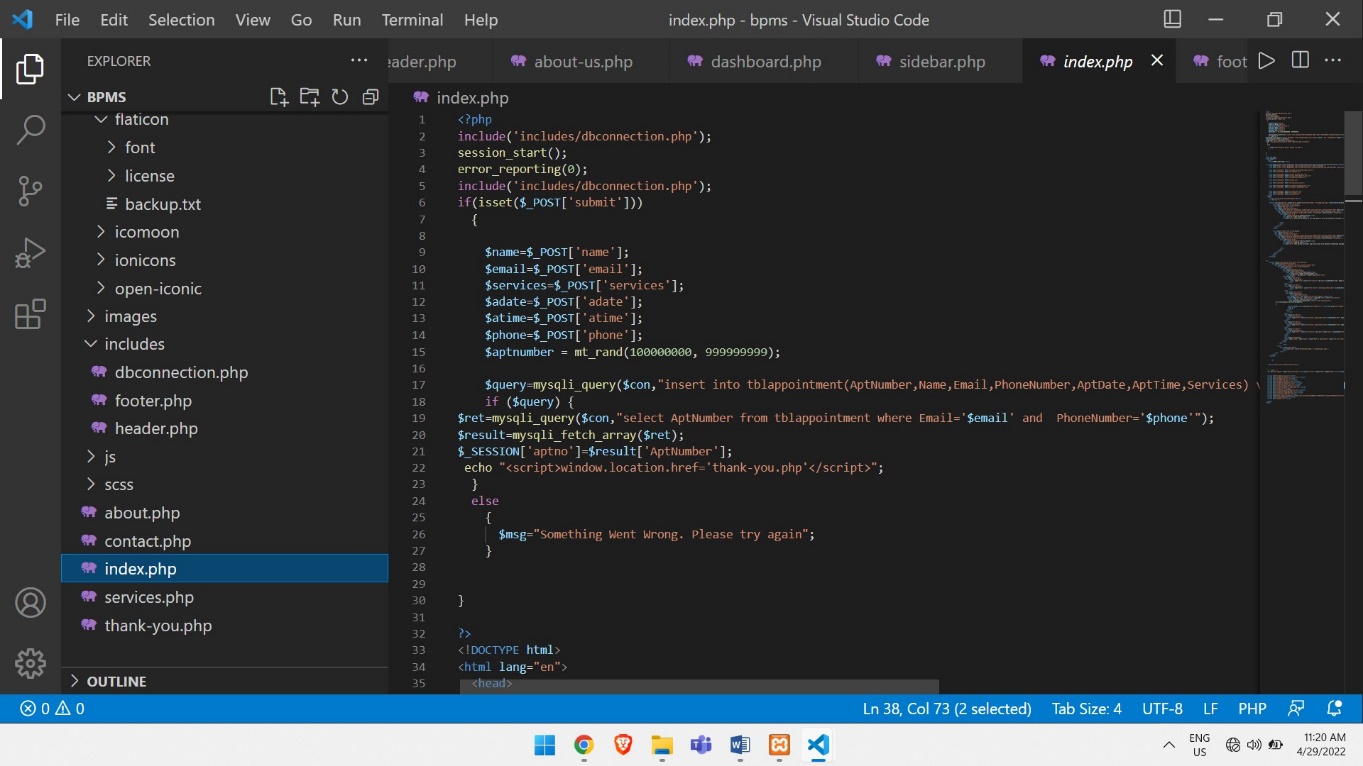


* **Manage Appointment**

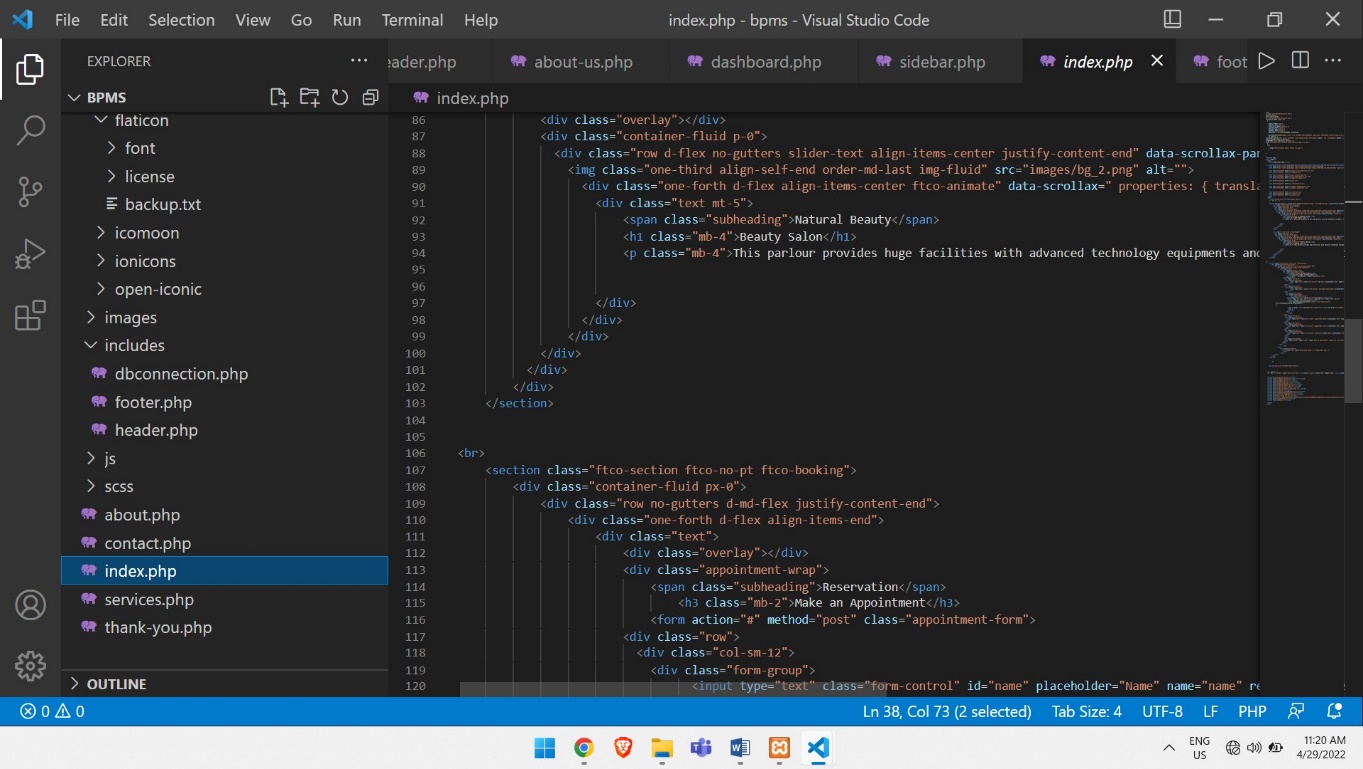


# Coding

### Index page

A screenshot of a computer

Description automatically generated

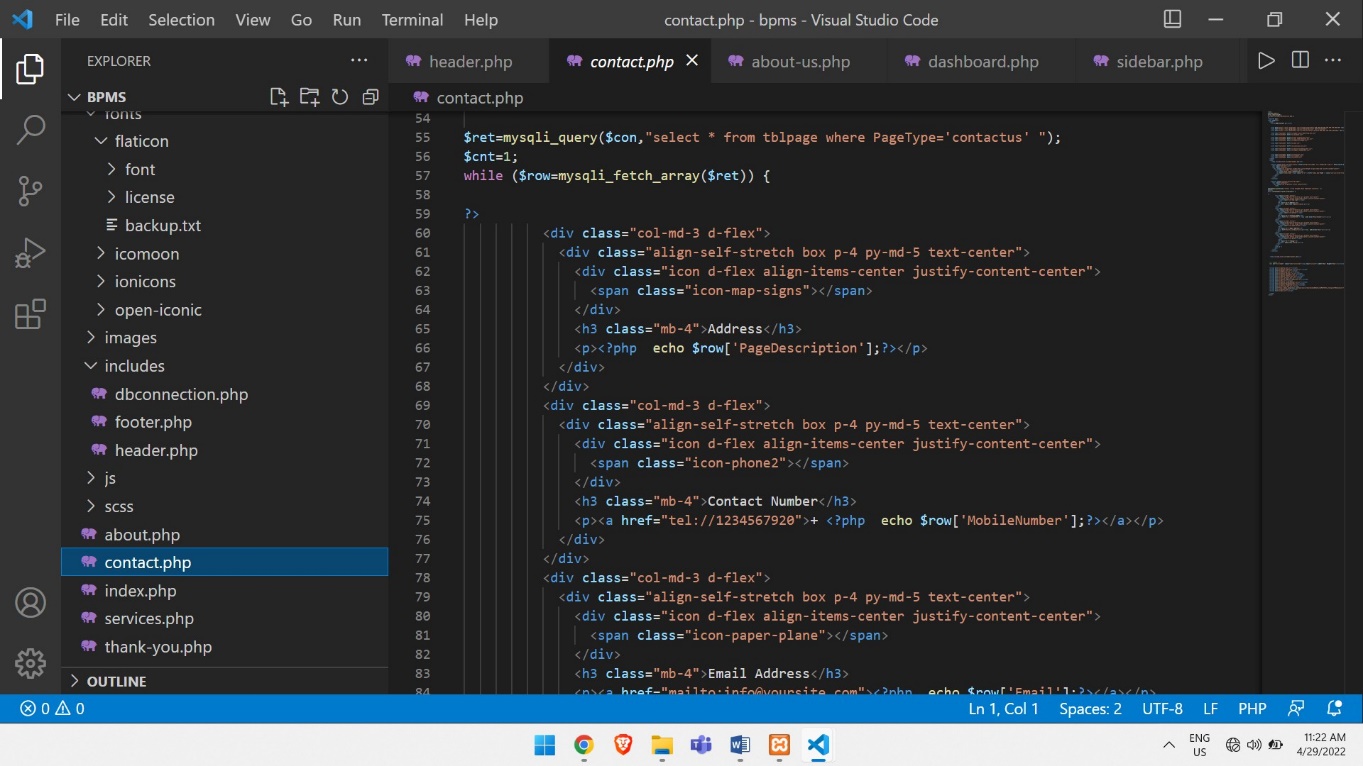
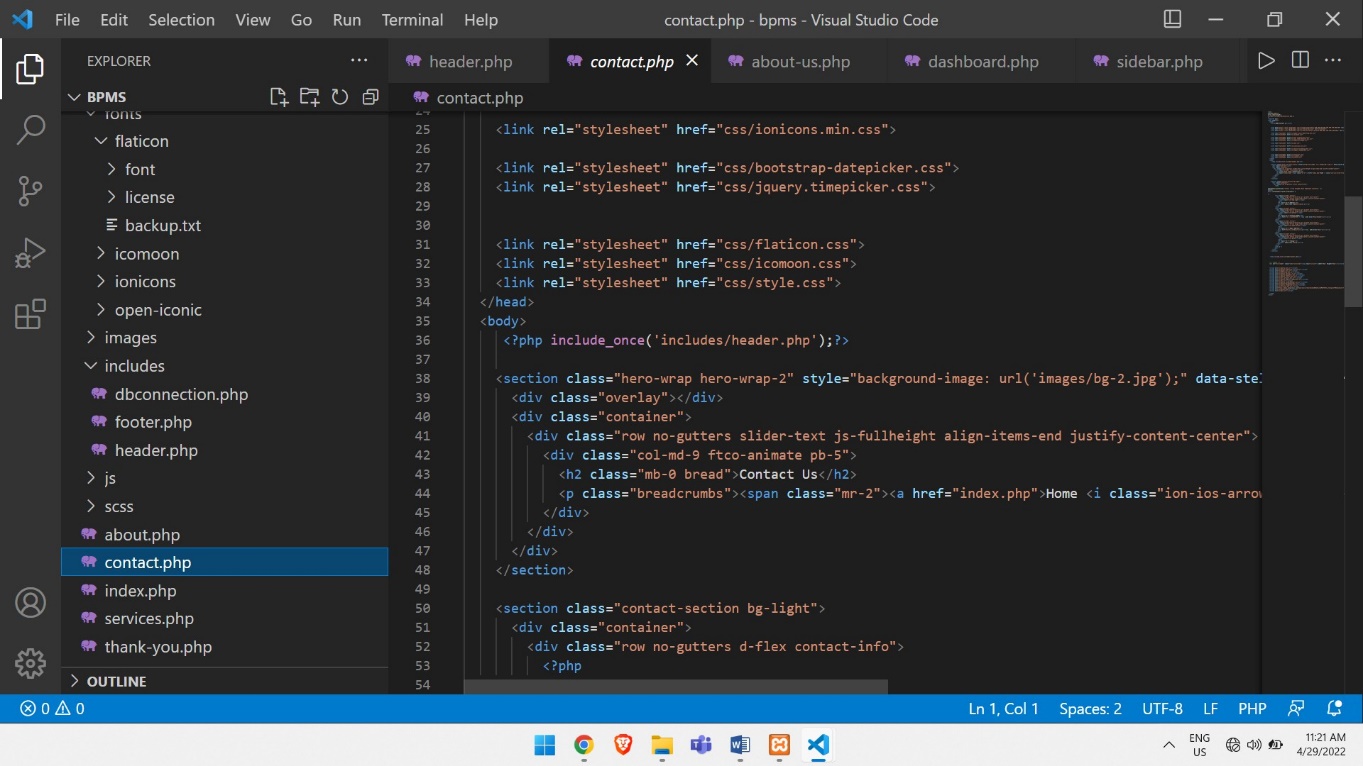


**Header**

A screenshot of a computer

Description automatically generated

### Contact us



**BIBLIOGRAPHY**

* + [http://www.google.co.in](http://www.google.co.in/)
  + [http://www.youtube.co.in](http://www.youtube.co.in/)
  + [http://www.wikipedia.com](http://www.wikipedia.com/)
  + [http://www.w3schools.com](http://www.w3schools.com/)