CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The Project "**AUTOMATION OF GARAGE AND ALIGNMENT MANAGEMENT SYSTEM**" is a web based application which is developed for fixing an appointment earlier and view the requires product is available or not and to get an estimation of price. Here the administrator can access this website and they can update product for garage and service purpose. This will help the customers to view the required spares is available or not, if available fix an appointment and save their time.

Nowadays everyone likes online acknowledgement because customers have no time for moving here and there to shop in the busy lifestyle. Therefore, every shopkeeper needs to develop their own online shopping portal. Every competitive organization needs a centralized store of shop information, a complete profile of shop that contains the all information about product and the service that has been provided by the shop.

The activities in garage shop is presently managed manually. It is always a hardest job to both the owner and customer. The reason is when the customer has a complaint in his vehicle then he need to approach the garage shop. So need to visit the shop physically and then he makes an enquiry and service the vehicle supposes if any vehicle has been in the queue of servicing list, then you need to wait for there until the vehicle is getting services. So more times it makes customers to wait there even though we need to go for any urgent work. Also the service provider also has the struggle of handling bunch of customer at a same time, in festival season and weekend time it is more hardest time for the administrator.

1.2 OBJECTIVE

The goal of the project is to create a web application that will reduce the amount of manual effort involved in managing appointment, choosing product, and service. It keeps all information related to appointment, products, and customers. This project is entirely.

Administrator. Only registered users have access to their accounts, which they may use to choose product, make appointment, and add to their final service list. It keeps track of all product, service list, and customer information.

1.3 PROBLEM STATEMENT

Since the shop is associated with the lives of ordinary people and their day-to-day routines, I decided to work on this project. The manual handling of the record is time-consuming and highly prone to error. To automate, the admin can easily add the products, retrieve the customer data. I have tried my best to make the complicated process of Automation of Garage and Alignment Management System as simple as possible using Structured & Modular techniques & Menu-oriented interface. I have tried to design the software in such a way that users may not have any difficulty using this package & further expansion is possible without much effort.

The activities in garage shop is presently managed manually. It is always a hardest job to both the owner and customer. The reason is when the customer has a complaint in his vehicle then he need to approach the nearby garage shop. So need to visit the shop physically and then he makes an enquiry and service the vehicle supposes if any vehicle has been in the queue of servicing list, then you need to wait for there until the vehicle is getting services. So more times it makes customers to wait there even though we need to go for any urgent work. Also the service provider also has the struggle of handling bunch of customer at a same time, in festival season and weekend time it is more hardest time for the administrator.

1.4 BACKGROUND

Currently, the process related to this shop has been done manually. This process will get difficult when the number orders increase, it is very difficult to maintain the records. During the report-writing process, errors and inaccuracies can occur. It is possible that the report will not be clear and clean. The reports' generation could take a long time. Data redundancy has piqued my interest. The database is dumped as the corporate turnover increases year after year, and human entry is too slow to access. With the help of this application one can view the availability of time to service his vehicle and also choose some accessories that needs to fixed or gets added with our vehicle. For admin also it is very easy to view the next day’s timeline, these timelines tell about that those who are going to view our shop and for what problem they reaching out shop. If any of the requires products is not available in the store, then the admin by viewing the timeline he can understood the product is need and bought from any other showrooms earlier. By this application the customer also knows the estimated amount. The use of technology (computer) for improved marketing production is known as internet marketing.

SUMMARY

The **" Automation of Garage and Alignment Management System "** was created to address the constraints of the traditional manual system, according to the overview of this chapter. The goal of this programme is to remove or, in some circumstances, reduce the problems that this system is now experiencing. This system can ensure that items are managed in an error-free, safe, and timely manner. In this chapter, the objects, issue states, and background are all thoroughly explained. The project's system analysis was presented in Chapter 2.

CHAPTER 2

SYSTEM ANALYSIS

The main purpose of this system analysis is to provide a detailed explanation of the existing system, it includes following things, issues faced by users in manual work, loopholes of the existing system, a list of system drawbacks. Implementation analysis can overcome the drawbacks of the existing system, it can also help to benchmark the system requirements and feasibility of the system.

2.1 EXISTING SYSTEM

In an existing System, the activities in this shop are presently managed manually. The existing system is not automated. It has to do a lot of manual work. They need to visit the show to verify what the problem is, even sometimes they need to wait for a huge period to check the fault in his machine. Nowadays, People are very busy and they don't find much time to go to the shop and waits for huge time to service his vehicle.

The existing system is fully based on manual work. All the details stored and maintained by a paper. This system takes a lot of time to update the information. All the data is handled by manual so, easily data is lost. Less security for data. The database is dumped as the corporate turnover increases year after year, and human entry is too slow to access. Users cannot service their vehicle at anywhere and anytime by using the existing systems. In a manual database system, information must be found by hand rather than electronically. Looking for information in a manual database may take hours searching for particular piece of data. A manual database is very vulnerable to fire or natural adversities.

2.1.1 Drawbacks of Existing System

The existing system we have to wait for a long time to service our vehicle. Also calculating the amount for serviced product in manual way may sometime cause mistake. Updating and maintaining price list for each and every product is a garage is a huge task. Maintaining and updating order details takes more time. Customer need to visit much more time to verify the vehicle has been serviced or not. In this system all work is finished by manual but now all types of process are done by the computerized equipment. These are the main drawbacks of this system.

2.2 PROPOSED SYSTEM

The proposed system is computerized and has been developed using advance language therefore it gives more facilities than present system. It provides quick access to any data. In this system user have to enter the data only once and then it get linked with all files. This reduces the workload of user and it is also a time saving process. The information about any Subscriber can be easily retrieved. The system maintains all records easy.

2.2.1 Advantages of Proposed System

The retrieval of appliances prices is very fast in this application and it is also easy for the users. The users can fix an appointment through this web application. The customer can also choose the product that suitable to fix the problem in our vehicle. The prices of the product will be updated periodically. The updating of prices is very easy in this application. Using this application, users can view the spare prices anywhere. The application needs one-time updates per day. The simple clicking options improve the interface of the app. By maintaining all the work on mobile will increase our accuracy as well as speed of our work. It will be easily used and the time consumed is decreased.

- All the information about service will be maintain properly inthis system.

- All manual calculation of sale or all the management will beperformed by the computer automatically.
- This system will provide timely report information.
- It will produce report for service.
- We can fix an appointment through the application.
- User can enquire the amount need for the servicing the vehicle.
- The computer can hold amount of data in its storage device.
- The operation and speed of the computer is very high.
- We can calculate result and print any report within seconds.
- Any difficulties we can solve easily.
- A database application can be stored in computer effectively.
- It is very user friendly and easy to handle.

So the computerized system is more suitable than the manual system.

2.3 FEASIBILITY STUDY

The feasibility study deals with all analysis that takes up in developing the project. Each structure has to be thought of in the development of the project, as it has to serve the end user in a friendly manner. A feasibility study determines whether the proposed solution is feasible basedon the priorities of the requirements of the organization. A feasibility study culminates in a feasibility report that recommends a solution. It helps you to evaluate the cost-effectiveness of a proposed system.

The feasibility study is carried out to test if the proposed system is worth being implemented. Given unlimited resources and infinite time, all projects are feasible. After performing a Preliminary Investigation, gathering and interpreting data and details concerning the project, a Feasibility Check is done which involves a series of steps to check the Technical, Financial and Operational feasibilities. One must know the type of

information to be gathered and system analysis consists of collecting, organizing and evaluating facts about a system and its environment. Three considerations involved in feasibility are

- Economic Feasibility
- Operational Feasibility
- Technical Feasibility

2.3.1 Economic Feasibility

The cost and benefit analysis may be conducted to know whether the computerized system is favorable in today's fast-moving world. This application is developed with Html, Css, Javascript, Php, Mysql which is one of the open-source languages and an Html, Css, Javascript for building user interfaces or for frontend development. The economic feasibility is carried out to check the economic impact that the system will have on the organization. In the proposed system many users can make make an appointment through online without visiting to the shop physically. By doing this activities the user can save the money that needed to spend to visit the shop until the vehicle has been serviced.

2.3.2 Operational Feasibility

The Proposed system solves the problems that occurred in existing system. The current day-to-day operations of the organization can be fit into the system. Mainly operational feasibility should include on analysis of how the proposed system will affect the organizational structure and procedures. The developed website is user-friendly to the customers because they can easily view the menu. The purpose of operational feasibility is a measure of how people feel about the system.

Operational feasibility criteria measure the urgency of the problem or the acceptability of a solution, Operational feasibility is dependent upon determining human resources for the project. If any issue arises then the admin could solve it an easy manner because the language used to develop his application is very much easy to understand, learn and to fix any issues. For the system, it is not necessary that the user must be a computer

expert, but any operator given a little bit of knowledge and training can easily operate.

2.3.3 Technical Feasibility

The technical feasibility study basically centers on an alternative for hardware, software, and design approaches to determine the functional aspects of the system. A useful way to present the results of the feasibility studies is using a feasibility assessment matrix. This shows the percentage score for each criterion and feasibility study of technical, operational etc., It helps in troubleshooting the project before commencing work. The study identifies potential challenges and uncovers ways to overcome them. It also helps in long-term planning, as it can serve as a flowchart for how products and services evolve before they reach the market.

SUMMARY

This chapter describes the existing system and its disadvantages, such as the fact that the company's current work is done manually, which takes more time and effort, and that all of these issues can be resolved by using this application, which offers a number of benefits such as time savings, reduced manual work, and so on. The system's economic, operational, and technical feasibility studies are also discussed. The system requirement for this project is described in Chapter 3.

CHAPTER 3

SYSTEM REQUIREMENTS

The main objective of this chapter is to satisfy the needs of the system requirements and to handle all user's needs. Numeric analysis is also taken into consideration to benchmark the system hardware and software. The Hardware and Software use matrix has been kept as per the users need. The details are followed accordingly in this chapter.

3.1 SYSTEM REQUIREMENTS

A declaration that describes the capability that a system needs in order to satisfy the customer's requirements is known as a system requirement. Whether talking about the system requirements for specific machines, software, or business operations in general. Taking it all the way down to the hardware and coding that operates the software. System requirements are the most efficient way to address user needs while lowering implementation costs. System requirements have the potential to save a company a lot of money and time, but they also have the potential to squander money and effort. They are the most significant component of any project, because if the system requirements aren't met, the project isn't finished.

3.2 HARDWARE REQUIREMENTS

This section gives the details and specification of the hardware on which the systempredictable to work.

Processor	:	Intel, AMD
RAM	:	Minimum 2 GB
Hard disk	:	500 GB
Monitor	:	15"inch VGA/SVGA monitor
Keyboard	:	Standard 104 keys

Mouse : 3 Buttons

3.3 SOFTWARE REQUIREMENTS

The software for the project is selected considering the factors such as working front end environment, flexibility in the coding language, database knowledge enhanced in backend technology etc.

Operating System	: Windows 7, Windows 10
Front-End	: HTML, CSS, JQuery, AJAX
Back-End	: PHP, MySQL
Tools	: Visual Studio, XAMPP

3.4 PRESENT USAGE OF REQUIREMENTS

The following hardware and software requirements are used to construct the present system, which is stratified to meet all of the needs, and only then will the system produce the desired outcomes and complete the project. This system can accommodate a large number of users. The database is being handled on the admin side.

3.5 AFTER DEPLOY

After implementing the system, which is currently operational, all required software and hardware will be required, as well as one additional item that I believe will be critical: maintaining a large amount of data from both the admin and user perspectives in a small amount of memory (system memory) to manage a large volume of data by using a centralized server for data maintenance on the developer side Only the system memory and data saved in the system will be managed by the administrator.

SUMMARY

The system requirements are fully explained in detail for each piece of hardware and software that was used to meet the user and system requirements. The detailed information regarding the system design will be presented in the following Chapter 4.

CHAPTER 4

SYSTEM DESIGN

This chapter gives a complete overview of the module. Dataflow and ER diagrams are also included, as well as a system architecture diagram that illustrates the front end and back-end tools. It also includes a number of designs, such as database, table, input, and output designs, all of which will be thoroughly described.

4.1 MODULE DESCRIPTION

A module description provides detailed information about the module and its supported components, which is accessible in different manners. In this application it contains two main modules with many sub-modules. The two main and sub-modules are

Controller Module

- Spares
- Customer

Customer Module

- Choose Issue
- Fix Appointment
- Cart
- Report

4.1.1 Controller Module

Admin module allows system administrators to set up the back-end of the system and perform basic system configuration, mainly definition of predefined drop-down fields, definition of classes time schedule, etc. Admin module helps to set up the system's backend and execute basic system administration, such as defining predefined drop-down fields, class time schedules, and so on.

Spares Module

In this module admin can create a new product in the website with the detailed Manner like price, size, brand and category of the product. Admin can also see the list of products which were created before. Whatever product created in this product module will be displayed in the user site. In the product module, there are some sub-modules.
Sub-modules in the product modules are

- Adding Spares
- Update Spares
- Delete Spares

Categories Module

In this category the module specifies what type of product this is, for example, a tire or garage spares. The product can be added to the appropriate category by the administrator. It assists the user in quickly locating the product. The admin can also create a new product to be added to as well as add the product to the newly created category. Admin can also generate report from particular date.

Customers Module

In this customer module, admin can see the customers profile details and order details. This module provides the details of customers like ID, Name, E-mail, contact and address. Admin can also delete the customer details if it doesn't need anymore.

4.1.2 Customer Module

The User module is one of the main modules in this application. In this module, the user can use their email ID and password to enter and access the screen**.** Before access the application user should do the registration process by giving users details.

Choose Issue

The Choose Issue module contains problems in different parts. In this page, users can see categories which contain multiple category problems which relate to these garage like engine sound, timing chain, chain sprocket and tire products like front and back tires. After user selected correct issue then users are able to view the list of products and price of each product.

Fix Appointment

Fix appointment module is the important module in this user site. The customer can fix appointment to their service through this module. Add to service option is available so that users can add their multiple service to the cart section, after fixing their appointment the user can complete their order. The customer will not be limited to adding a specific limited number of appointment to their cart.

Cart Module

A Cart class with properties like product id, total price, and categories. After fixing an appointment the list is updated to the Cart. The cart can create a new page that displays the cart items details like item name, price, and product id appointment time. Implement cart action can add actions like "cancel appointment" to the cart page. These actions will update the Cart class instance and the UI accordingly. Persist the cart data can use local storage or a backend API to persist the cart data across sessions.

Report

In this module the final serviced details are available in a table format which contains category, product name, customer name, price amount, serviced mechanic name and serviced mechanic contact. All these details are displayed only when admin accepts the final serviced list, also in the setting module the user can update the credentials of his own.

4.2 DATAFLOW DIAGRAM

The Data Flow Diagram provides information about the inputs and outputs of each entity and process itself.

LEVEL 0

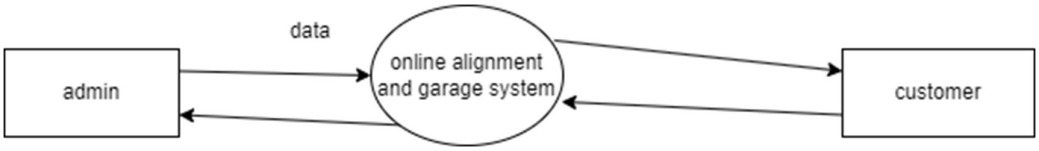

Figure 4.1 Level 0 Data Flow Diagram

In Figure 4.1 Admin and Customer Can Login to the system and make use of their respective process in the system. high-level data flow between the entities involved in the garage shop project. The customer browses issue and fix an appointment on the garage shop and adds it to their cart using the Cart Module. When the vehicle is serviced, they proceed to the report.

LEVEL 1

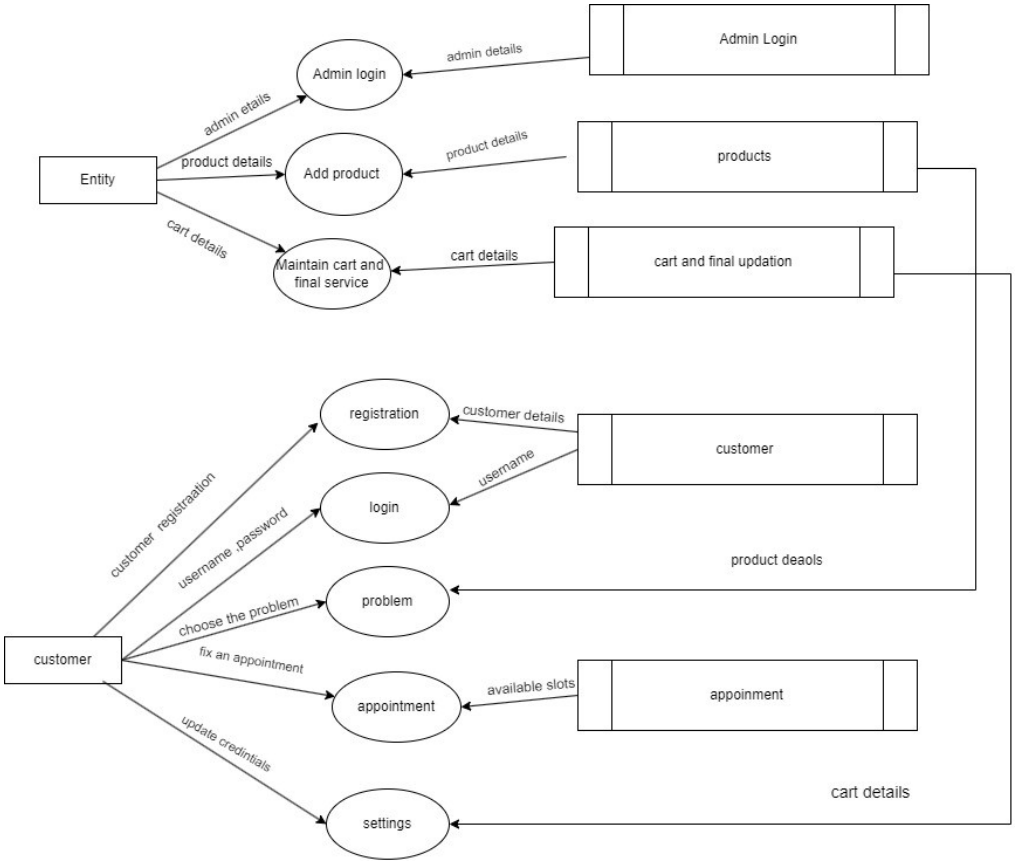

Figure 4.2 Level 1 Data Flow Diagram

In Figure 4.2 Admin can Login and manage the product and view appointments and manage the user. The user can also login to their account and fic an appointment and view the report and manage the account. admin and the customer. The admin is responsible for managing the products and orders, while the customer can view the products, add items to the cart, and place an order.

4.3 ENTITY RELATIONSHIP DIAGRAM

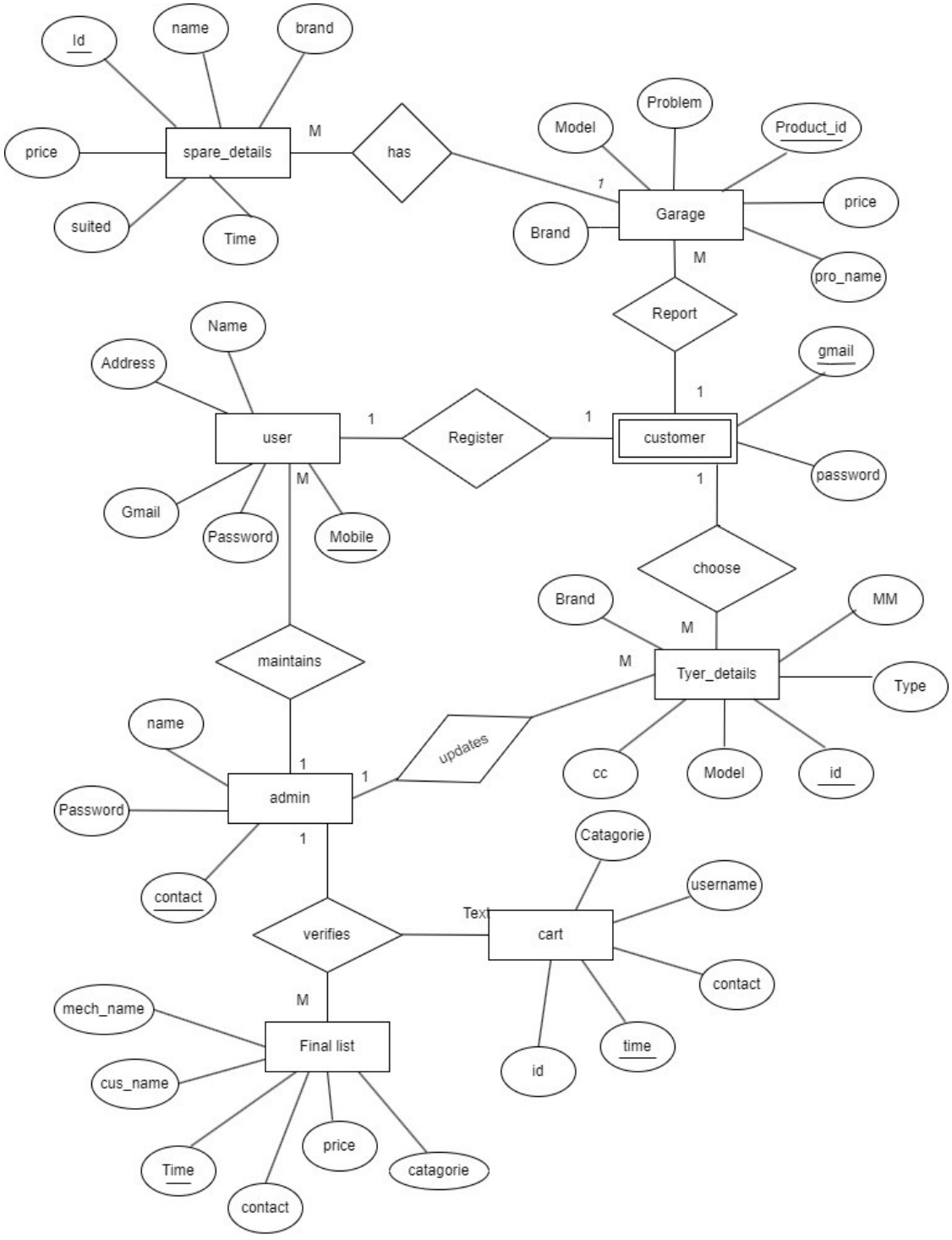

Figure 4.3 ER Diagram

In Figure 4.3 ER Diagram there are Admin, user process is described. Each has its own different process and connection with each other. Admin can manage users and products. Users can order a product. Product is stored in the order table. Admin can process the product based on users input.

4.4 SYSTEM ARCHITECTURE

Front End

PHP is the web development language written by and for web developers. PHP stands for PHP: Hypertext Preprocessor. The product was originally named Personal Home Page Tools, and many people still think that's what the acronym stands for, but as it expanded in scope, a new and more appropriate name was selected by community vote. PHP is currently in its sixth major rewrite, called PHP6 or just plain PHP.

Introduction to PHP

PHP is a widely-used, open-source scripting language used for scripts that are executed on theserver and it is freeware. It is a server side scripting language used to develop attractive and dynamicweb pages. PHP is widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.We make available with database used with PHP is MYSQL - which is also an open source which isan added advantage. PHP's simple programming style, we attempt to design in a way that enables anyone with basic programming knowledge to learn and shift to never-ending opportunity available. PHP started out as a small open-source project that evolved as more and more people found out howuseful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

Compared to many other programming languages, PHP makes it easy to develop powerful web applications quickly (this is a blessing and a curse). Many of the most useful specific functions (such as those for opening a connection to an Oracle database or fetching email from an Internet Message Access Protocol [IMAP] server) are predefined. Most advanced PHP users (including most of the development team members) are diehard hand coders.

PHP & MySQL are also both stable in the sense of feature stability. Their respective development teams have thus far enjoyed a clear vision of their project and refused to be distracted by every new fad and ill-through-out user demand that comes along. Much of the effort goes into incremental performance improvements, communicating with more major databases, or adding better OOP support**.** PHP is a general-purpose scripting language that is especially suited to server-side web
development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere.

- PHP is a recursive acronym for "PHP: Hypertext Pre-processor".
- 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 ecommerce 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 hugeresult sets in record-setting time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Back End

MySQL is an open source, SQL relational database management system (RDBMS) that is free for many uses (more detail on that later). In its history, MySQL occasionally faced opposition because of its lack of support for some core SQL constructs such as sub selects and foreign keys. Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use. Its acceptance was aided in part by the wide variety of other technologies such as PHP, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions. Databases are generally useful, perhaps the most consistently useful family

of software products (the “killer product”) in modern computing. Like many competing products, both free and commercial, MySQL isn’t a database until it gives it some structure and form.

4.5 REQUIREMENTS PERFORMANCE

Performance requirements define acceptable response times for system functionality. The load time for user interface screens shall take just a few seconds. Queries shall return the results very faster.

Security

Security is the most important thing on the network. The system verifies the username and password and then allows the administrator to access the data if only the username and password is correct; otherwise, it will display the error message. It provides more security to the administrator for accessing the data.

Maintainability

This system is being developed using HTML that supports all types of data in a different format, and which has a high community support. Therefore, it is easy to maintain all data. Maintenance of the system is very flexible and cost effective.

Reliability

Reliability describes the ability of a system or component to function under stated conditions for a specified period of time. Reliability is closely related to availability, which is typically described as the ability of a component or system to function at a specified moment or interval of time.

4.6 DATABASE DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database

management system manages the data accordingly. The term database design can be used to describe many different part of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structure used to store the data. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system.

Data Independence

Data independence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of applications and to allow modifications to application programs without recognizing the physical data.. The tables needed for each module were designed and the specification of each and every column was given based on the records and details collected during record specification of the system study, as shown in the below.

4.8 TABLE STRUCTURE

Table 4.1: Admin Details

S.NO	FIELD NAME	TYPE	SIZE	CONSTRAINT
1	Contact	Int	10	primary key
2	Name	Varchar	15	unique key, not null
3	Password	Varchar	15	not null

TABLE NAME: admin

PRIMARY KEY: contact

TABLE DESCRIPTION: This table Contains the Admin details. This table lists

the fields that consists of admin name, password, contact .

Table 4.2: Tier Details

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
1.	Id	Int	5	Primary key
2.	Brand	Varchar	20	Not null
3.	Mm	Varchar	10	Not null
4.	Type	Varchar	5	Not null
5.	Model	Varchar	10	Not null
6.	Cc	Int	4	Not null

TABLE NAME: Tier Details

PRIMARY KEY: id

TABLE DESCRIPTION: This table Contains the tier details. This table lists the fields that category table consists of Brand Name, Tier Size, Tier Type, Model and cc are the details stored in this table.

Table 4.3: Spare Details

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
1.	Product_id	Int	5	Primary key

2.	Pro_name	Varchar	10	Not null
3.	Brand	Varchar	10	Not null
4.	Model	Varchar	10	Not null
5.	Problem	Varchar	20	Not null
6.	Price	Int	5	Not null

TABLE NAME: Spares

PRIMARY KEY: id

TABLE DESCRIPTION: This table Contains the spare details. This table lists the fields that spare table consists of product id, product name, brand, model, problem and price are the details stored in this table.

Table 4.4 : Tier Details

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
1.	Username	Varchar	15	Not null
2.	Contact	Int	10	Not null
3.	Brand	Varchar	10	Not null
4.	Model	Varchar	15	Not null
5.	Mm	Varchar	10	Not null

6.	Price	Int	6	Not null
7	Time	Timestamp	None	Not null
8.	Product_id	Int	4	Foreign key, not null
9.	Catogorie	Varchar	5	Not null

TABLE NAME: Tier

PRIMARY KEY: Time

FOREIGN KEY : id (Tier details)

TABLE DESCRIPTION: This table Contains the ordered tier history. This table lists the fields that tier table consists of username, contact, product id, brand, model, mm, price, time and category are the details stored in this table.

Table 4.5 : Garage Details

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
1.	Username	Varchar	10	Not null
2.	Contact	Int	10	Not null
3.	Brand	Varchar	10	Not null
4.	Model	Varchar	10	Not null
5.	Problem	Varchar	15	Not null

6.	Price	Int	6	Not null
7.	Product	Varchar	20	Not null
8.	Product_id	Int	5	Foreign key
9.	Time	Timestamp	None	Primary key
10.	Category	Varchar	10	Not null

TABLE NAME: Garage

PRIMARY KEY: Time

FOREIGN KEY: Id (spares)

TABLE DESCRIPTION: This table Contains order details from garage and alignment section. This table lists the fields thatproduct review table consists of username, contact, brand, model, problem, price, product, product_id, time and category are the details stored in this table.

Table 4.6 : Final Service Table

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
1.	Name	Varchar	15	Not null
2.	Contact	Int	10	Not null
3.	Catagorie	Varchar	8	Not null
4.	Details	Varchar	50	Not null

5.	Price	Int	6	Not null
6.	Date	Int	11	Not null
7.	Servicer_name	Varchar	15	Not null
8.	Servicer_contact	Int	10	Not null

TABLE NAME: Final Service

TABLE DESCRIPTION: This table Contains the details of the customer who had successfully completed their vehicle service. This table lists the fieldsthat customer name, contact, category, details, price, date, servicer name, servicer contact are thedetails stored in this table.

Table 4.7 : Customers List Table

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT
1.	Username	Varchar	20	Not null
2.	Mobile	Int	10	Primary key
3.	Gmail	Varchar	15	Not null
4.	Password	Varchar	15	Not null

TABLE NAME: Signup

PRIMARY KEY: mobile

TABLE DESCRIPTION: This table Contains the user details. This table lists the fields that

consists username User Gmail, user contact no, user password, user Address, password are the details stored in this table.

4.8 INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer understandable format. Input design is one of the most expensive phases of the operation of a computerized system and is often the major problem of a system. A large number of problems with a system can usually be tracked back to fault input design and method. Every moment of input design should be analyzed and designed with utmost care. The system should be user friendly to gain appropriate information to the user. The decisions made during the input design are the project gives the low time consumption to make sensitive applications simple. Thus, the developed system is well within the budget. This was achieved because most of the technologies used are freely available. Only the customized product had to be purchased. In the project, the forms are designed with easy-to-use options. The coding is being done such that proper validation is made to get the perfect input. No error inputs are accepted.

User Login Form

In the user login form, the customer can login with their registered email id and password to access the portal.

User Register Form

In the user register form, the customer should register themselves for further notification about their order details.

4.9 OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; it should be understandable with the enhanced format. Computer output is the most important direct source of information to the user. Output design deals with form design. Efficient output design should improve the interfacing with user. The term output applies to any information produced by an information system in terms

of data displayed. When analysts design system output, they identify the specific output that is needed to meet the requirements of the end user. Previewing the output reports by the user is extremely important because the user is the ultimate judge of the quality of the output and, in turn, the success of the system. When designing output, system analysis accomplishes more things like, to determine what applications, websites or documents are blocked or allowed. The output is designed in such a way that is attractive, convenient and informative.

SUMMARY

This chapter contains a detailed description of the module. Login, manage user, manage product, user module, and payment module are all mentioned in the module description. Dataflow and ER diagrams, as well as a system architecture that explains front end and back-end tools, are illustrated. It also includes several designs, such as database, table, input, and output designs, which are all covered in detail. The coding will be discussed in Chapter 5.

CHAPTER 5

CODING

In this chapter, the objective of a code description is to summarize the code or to clarify the programmer's meaning. The code is not repeated or discussed in sufficient detail in the comments. Coding styles are defined by coding standards. A coding standard normally isn't concerned with what is correct or incorrect in general. Exception handling is a method or procedure for dealing with and processing errors in programming.

5.1 CODE DESCRIPTION

Code description can be used to summarize code or to explain the programmer's intent. Good comments don't repeat the code or explain it. They clarify its intent. Comments are sometimes processed in various ways to generate documentation external to the source code itself by document generator or used for integration with systems and other kinds of external programming tools.

5.2 STANDADIZATION OF THE CODE

Coding standards define a programming style. A coding standard does not usually concern itself with wrong or right in a more abstract sense. It is simply a set of rules and guidelines for the formatting of source code. The other common type of coding standard is the one used in or between development teams. Professional code performs a job in such a way that it is easy to maintain and debug. All the coding standards are followed while creating this project. Coding standards become easier, the earlier you start. It is better to do a neat job than cleaning up after all is done. Every coder will have a unique pattern than he adheres to. Such a style might include the conventions he uses to name

variables and functions ($userName, $username or $user_name for example), and how he comments his work. When the said pattern and style is standardized, it pays off the effort well in the long.

5.3 EXCEPTION HANDLING

Exception handling is a process or method used for handling the abnormal statements in the code and executing them. It also enables to handle the flow control of the code/program. For handling the code, various handlers are used that process the exception and execute the code. Mainly if-else block is used to handle errors using condition checking. if-else catch errors as a conditional statement. In many cases there are many corner cases which must be checking during a execution but “if-else” can only handle the defined conditions. In if-else, conditions are manually generated based on the task.

An error is a serious problem that an application doesn’t usually get past without incident. Errors cause an application to crash, and ideally send an error message offering some suggestions to resolve the problem and return to a normal operating state, like asking users to restart the application, refresh their browser tab, or log out and back in again. There’s no way to deal with errors “live” or in production — the only solution is to detect them via error monitoring and bug tracking and dispatch a developer or two to sort out the code.

Exceptions, on the other hand, are exceptional conditions an application should reasonably be expected to handle. Programming languages allow developers to include try…catch statements to handle exceptions and apply a sequence of logic to deal with the situation instead of crashing. And when an application encounters an exception that there’s no workaround for, that’s called an unhandled exception, which is the same thing as an error.

SUMMARY

In this chapter shows explained that the purpose of a code description is to summarise the code or to clarify the programmer's intent. Good comments don't repeat or explain the code. A programming style is defined by coding standards. A coding standard isn't usually concerned with what's proper or bad in a broader sense. Exception handling is a method or process for dealing with and executing anomalous statements in code. In Next Chapter 6 shown about the Testing.

CHAPTER 6

TESTING

This chapter will go through testing in depth, covering system testing, unit testing, user acceptability testing, and the test cases that were accomplished for this project. Once the source code is complete, testing will be completed and recorded as applicable data structures. Testing and validation of the finished project is required, which involves both subtle and overt attempts to uncover flaws. During unit testing, each unit of the system is examined and its functionality is validated, guaranteeing that it functions as intended.

6.1 SYSTEM TESTING

After the source code has been completed, it is documented as related data structures. The completed the project must undergo testing and validation where there is subtle and definite attempt to get errors. The project developer treads lightly, designing and executing test that will demonstrate that the program works rather than uncovering errors. Unfortunately, errors will be present and if the project developer doesn't find them, the user will find out. The project developer is always responsible for testing the individual units i.e., modules of the program. In many cases, developer also conducts integration testing i.e., the testing step that leads to the construction of the complete program structure.

6.2 UNIT TESTING

In unit testing, each unit of the system is tested, and its functionality is verified and ensures that it functions as expected. Unit testing is performed by the web developer while developing the application. "Online Mobile and Laptop Sales and Service Management System" has all modules are tested separately. In unit testing, testers have to

test the programs making use of the system. For this reason, unit testing is sometimes called as program testing. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function, Unit testing first works on the modules independently of one another, to locate errors. Each test case is tested independently in an isolated environment, as to ensure a lack of dependencies in the code.

In this product we made the unit testing by testing each file separately, like login, sigh up, fix an appointment, cart page and report page. Firstly, in login and signup page we test the functionality of the page and in the admin we will test all the admin enabled services functionality like add new product, add new product, delete the product and user information, update the data of user and product.

6.3 USER ACCEPTANCE TESTING

User Acceptance testing is a critical level of testing in the testing process since it involves significant participation of the developer with end users in order to make sure that the software/application fulfils the functional requirements of the users, and also ensures whether the final developed software provides ease of access and control over the application to the users**.**

(i) Application Functionality ideally, all the key functionality should get validated and to verify that a software application performs and functions correctly according to design specifications. During functionality testing we check the core application functions, text input, menu functions and installation and setup on localized machines, etc.

(ii)Contractual –After all the above testing is getting completed we will have done this contractual, here in to different system we open the application, in one system we open the user page and in another we open the admin interface. After in customer system we will select the product and fix the appointment and con currently in the admin page we will done the report generation.

6.4 TEST CASE

Table 6.1 Test Case

Test Case ID	Test Objective	Test Data	Expected Result	Actual Result	Test Result
TC_001	Check Email	admin@gmail.com	Accept	Accept	Pass
TC_002	Check Password	**********(Incorrect password)	Show error	Show error (Enter validPassword)	Pass
TC_003	Check Password & Email	NULL	Show error	Show error (Enter valid Email and Password)	Pass
TC_004	Check password	**********(correct password)	Successfully login	Successfully login	Pass
TC_005	Check Email	nandha@gmail.com	Show Error	Show Error (Please EnterValid Email)	Pass
TC_006	Fix an appointment in Past time	04:30 6/7/2021	Doesn't enable	The button hide	pass

TC_007	Fix an appointment at predefined time	NULL	Appointment already fixed	Show alert message(Appointment already fixed)	Pass
TC_008	Logging in without sigh in	nandha@gmail.com 12345	Please signup first	Please signup first	Pass
TC_009	Log in at admin page with user credentials	kk@gmail.com 12345	Invalid Credentials	Invalid Credentials	Pass
TC_010	Submitting without entering required field	NULL	Field required	Field required	Pass
TC_011	Submitting without selecting any filter		Field required	Field required	Pass
TC_012	Choose an vehicle to fix an appointment	NULL	Goes to next page	Goes to next page	Pass

TC_013	Fixing appointment with out clicking the checkbox	NULL	Time box do not apply	Time box doesn't apply	Pass
TC_014	Delete an fixed appointment	NULL	Delete an appointment	Delete an appointment	Pass
TC_015	Printing an empty Report	NULL	Report list empty	Report list empty	Pass
TC_016	Clicking add user	NULL	Redirect to add user page	Redirect to add user	Pass
TC_017	Empty data in Input field	NULL	Field is required	Field is required	Pass
TC_018	Remove an existing data in update field	NULL	Field is required	Field is required	Pass

TC_019	Entering alphabet in numeric field	NULL	Data doesn't show	Data doesn't show	Pass
TC_020	Submitting without end date in report generation	NULL	Select end date	Select end date	Pass

In Table 6.1, Test case of the table is shown. Each field is tested, the email id and password that matches each other for login purpose, which shows that the test case pass and login will work properly.

SUMMARY

The preceding chapter discusses the many types of testing, including system testing, unit testing, and user acceptability testing. During the system evaluation following the completion of the source code, it is documented as associated data structures. The final project must go through testing and validation, which includes both subtle and overt attempts to find problems. Each unit of the system is evaluated, and its functionality is confirmed during unit testing, ensuring that it performs as expected.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The project **"AUTOMATION OF ALIGNMENT AND GARAGE MANAGEMENT SYSTEM"** has been designed and developed as per the specification. The project is very simple and gives clear understanding. The code written in this project is very clear. The system is tested with various sample data. From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient GUI based component. They can also read product reviews as well as create their own. The overall project covers the scope and viability of the shop, focusing on the financial, technological and market potential. This component can be easily plugged in many other systems. It is user- friendly application to user. It will reduce the time of user and effort also. The new system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. The system is very fast according to the user wish that can be viewed or retaken at any level.

7.2 FUTURE SCOPE

Future modifications to this application will be straight forward to develop, because it was designed in such a way. This project has a bright future ahead of it. A mobile application can be designed for this system. More features, such as phone-based customer assistance, could be added to the system. This application may also be registered on sites like Olx which could boost the growth of the company. Artificial intelligence technologies can also be utilized to enhance their business. A chat box for public benefit is also a great idea via which people can directly have a conversation with some officials regarding any type of queries.

APPENDIX

A.SAMPLE CODING

Index.php

```
<html>
 <head>
 <title>
 welcome to bike garage shop
 </title>
</head>
 <body>
 <style>
 .top
{
display:flex;
background-color:black;
justify-content: space-around;
padding-bottom: 14px;
PADDING-TOP:10PX;
}
.ul
{
display:flex;
}

  .s1
{z-index:1px;
position:absolute;
display:none;
min-width: 70px;
}
```

```
.s1 a{
 display:block;
 padding:5px;

  }
 .h1
 {
 display:inline-block;

  }
.h1:hover .s1{
display:block;

 border-color:r;
 text-align: center;
  min-width: 95px;
 background-color:cyan;
 border:double;

 }
 .box{
 margin-top: 20px;
padding-top: 10px;
width:400px;
}
.s1:hover{
 background-color: orange;
 }
 .button{
 padding: 3px;
background-color:lavender;
color:black;
border:double;
```

```
font-family:serif;
border-color:aqua;

}
top{
background-color: ;
 }

.h1{
 padding-right: 20px;
}
Li
{
padding-right: 90px;

}
.content{
display:none;
font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial,
sans-serif;   min-width: 30px;
}
.button:hover
{
background-color:darksalmon;
}

.bu{
padding:5px;
border: double;
border-color: red;
}
.contents {
 max-width: 1050px;
```

```
 margin: auto;
line-height: 2;
font-family:-apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu,
Cantarell, 'Open Sans', 'Helvetica Neue', sans-serif
    }
   { float:right;
   height:300px;
   width:300px;
   padding-right:50px;
  }
  .mySlides {display:none;}
  .i2,img
  { float:left;
 height:150px;
 width:150px;
  padding-right:20px;
 padding-top: 5px;
  padding-bottom: 5px;
  padding:10px;
  }
 .content a{
  display: block;
  padding: 1px;
   text-align: center;
   }

    #but
     {
  padding-top:10px;
  margin-bottom: 0px;
 }
 .ul{
  display:flex;
```

```
  }
.button
  {
   height:22px;
  width: 20px;
  }
   .content{
    min-width: 10px;
    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
     z-index: 1;
     }
    .u1{
     background-color: white;
      }

    #logo
    {
       height:10px;
       width:20px;
    }
    </style>

     <body style="width:100%" >

    <div class="top" style="color:white;">
    <img  id="logo" src="logo.jpg" alt="check network "
style="padding:1%;height:2%;width:6%">
    <p style="padding-right:25%;color:gold ; font-family: 'Segoe UI', Tahoma, Geneva,
Verdana, sans-serif;">sri selvanayaki <br>    garage  </p>
    <span class="ser">

    </span>
```

```
   <div class="options">

   <ul id="ul">

 <li class ="h1"><button  id="garage" class="button"style="min-width:
100px;">GARAGE</button></li><script type="text/javascript">
 document.getElementById("garage").onclick = function () {
  location.href = "garage.php";
  };
  </script>
 <li class="h1"><button class="button" id='tyres' style="min-width: 100px;" >TYRES
</button></li><script type="text/javascript">
 document.getElementById("tyres").onclick = function () {
  location.href = "tyres.php";
  };
  </script>
  <li class ="h1"><button class="button"  id="allignment" style="min-width: 100px;"
>ALLIGNMENT</button></li><script type="text/javascript">
  document.getElementById("allignment").onclick = function () {
  location.href = "allignment.php";
};
   </script>
  <li class ="h1"><button class="button" id="cart" style="min-width: 100px;" >CART
</button></li><script type="text/javascript">
  document.getElementById("cart").onclick = function () {
   location.href = "cart.php";
  };
   </script>
    <li class ="h1"><button class="button" id="aboutus" style="min-width: 100px;"
>ABOUT US  </button></li><script type="text/javascript">
  document.getElementById("aboutus").onclick = function () {
   location.href = "home.html";
```

```
  };
  </script>
  <li class ="h1"> <button class="button" id="settings" style="min-width: 100px;"
>SETINGS</button> </li><script type="text/javascript">
 document.getElementById("settings").onclick = function () {
   location.href = "settings.php";
  };
 </script>
</ul>
 </div>
 </div>
  <h3> </h3>
 <script>
 var myIndex = 0;
 carousel();
function fun()
{
let a=document.forms["form1"]["search"].value;
if(a=="")
{
alert("enter valid values");
}
}
function carousel() {
var i;
var x = document.getElementsByClassName("mySlides");
 for (i = 0; i < x.length; i++) {
 x[i].style.display = "none";
      }
   myIndex++;
   if (myIndex > x.length) {myIndex = 1}
   x[myIndex-1].style.display = "block";
  setTimeout(carousel, 5000); // Change image every 2 seconds
```

```
 }
 </script>
<i><u style="color:darkorange"> Garage and services :</u> </i></h1>
 <h1><i><u style="color:darkorange"> Whell allignments :</u> </i></h
 <h1><i><u style="color:darkorange"> Tyres sales :</u> </i></h1>
</section>
 </body>
 <section class="footer" style="padding-top: 50px; ">
  <div class="f1"><ul>
   <p style="color:aqua">our services </p>
  <li><a href=""><p style="color:white">Garage</p></a> </li>
  <li><a href=""><p style="color:white">Allignment</p></a> </li>
  <li><a href=""><p style="color:white">Tyres</p></a> </li>
  <li><a href=""><p style="color:white">linked stores</p></a> </li>
  </ul></div>
  <div class="f1"><ul>
   <p style="color:aqua">Galery</p>
   <li><a href=""><p style="color:white">workers</p></a> </li>
   <li><a href=""><p style="color:white">Equipment</p></a> </li>
    <li><a href=""><p style="color:white">our stores</p></a> </li>
    <li><a href=""><p style="color:white">explore more</p></a> </li>
    </ul></div>
  <div class="f1"><ul>
    <p style="color:aqua">contact us </p>
    <li><a href=""><p style="color:white">twitter</p></a> </li>
    <li><a href=""><p style="color:white">watsapp</p></a> </li>
    <li><a href=""><p style="color:white">Instagram</p></a> </li>
     <li><a href=""><p style="color:white">linkedIn</p></a> </li>
   <li><a href=""><p style="color:white">Youtube</p></a> </li>
  </ul></div>
  <br>
  <br>
  </section>
```

```
   </body>
</html>

<?php
include'../includes/sidebar.php';
error_reporting(0);
?>


</div><br><br><br><br>
   <center>    <form method="post">
<input type="date"  name="start_time" placeholder="Enter the  start Time"  ><br><br>
<input type="date"  name="end_time" placeholder="Enter the  end Time"  ><br><br>
<input type="submit" name="generate" value="generate report ">
</form>
</center>

<?php

$conn = mysqli_connect("localhost","root","","scms");

//$conn = mysqli_select_db($connection,"sri_selvanayanki");


if(isset($_POST['delete_btn'])) {
    $contact = $_POST['contact'];
    $expected_time = $_POST['expected_time'];
    $catogorie=$_POST['catogorie'];
  $sql = "DELETE FROM garage  WHERE contact='$contact' and
expected_time='$expected_time' and catogorie='$catogorie' ";
  $count=mysqli_query($conn, $sql);
//echo("hihihh");
//echo($sql);
```

```
 $res=mysqli_affected_rows($conn);
 if($res)
 {
    echo"<script>alert('Deleted succesfully ');</script>";
 }
}

$sql = "SELECT * FROM garage";
$result = mysqli_query($conn, $sql);
$record=mysqli_affected_rows($conn);
$sql1 = "SELECT * FROM tyer";
$result1 = mysqli_query($conn, $sql1);
$record1=mysqli_affected_rows($conn);

 if(isset($_POST['generate']))
 {
$start=$_POST['start_time'];
$end=$_POST['end_time'];
$name="select * from transaction where date between '$start' and '$end'";

$res=mysqli_query($conn,$name);

$count=mysqli_affected_rows($conn);

if($count==0)
{
   echo '<script>alert("NO VALUES PRESENT")</script>';
}
else{
echo "<br><br><table border ='1' cellspacing='0' cellpadding='10' style='width:100%'>";
echo
"<tr><th>Username</th><th>Contact</th><th>Catogorie</th><th>Details</th><th>Servi
cer Name</th><th>Servicer Contact</th><th>Serviced Time</th><th>Price</th></tr>";
```

```
while($row = mysqli_fetch_assoc($res)) {
  echo
"<tr><td>".$row['username']."</td><td>".$row['contact']."</td><td>".$row['catogorie']."<
/td><td>".$row['details']."</td><td>".$row['servicer_name']."</td><td>".$row['servicer_c
ontact']."</td><td>".$row['date']."</td><td>".$row['price']."</td></tr>";
}
}
}
```

B. SCREENSHOTS

HOME PAGE

Figure B.1 Home Page

This is the home page design for this application, after sign up our new account user can enter into this page by login. This page has options like My Account, there user details available in that page, Product with different categories, cart page and logout option also available is shown in the Figure B.1

LOGIN PAGE

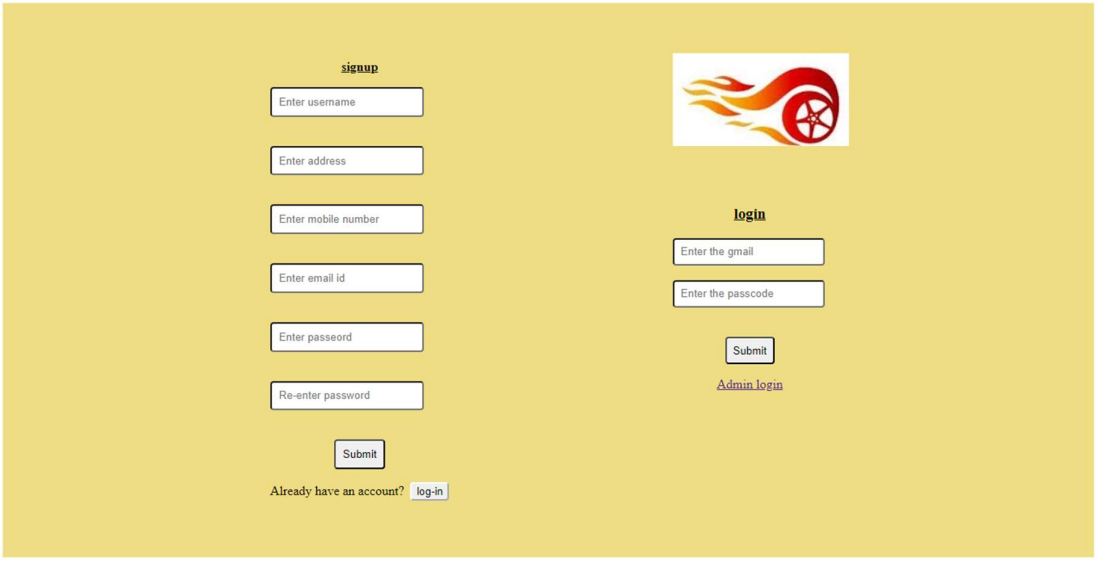

Figure B.2 Login Page

This is the Sign-Up page design for this application, here customer can enter the details to access the portal, new user registers their account before access this application. User should give their name, email address, password and confirm password is shown in the Figure B.2

CHOOSING TIER PAGE

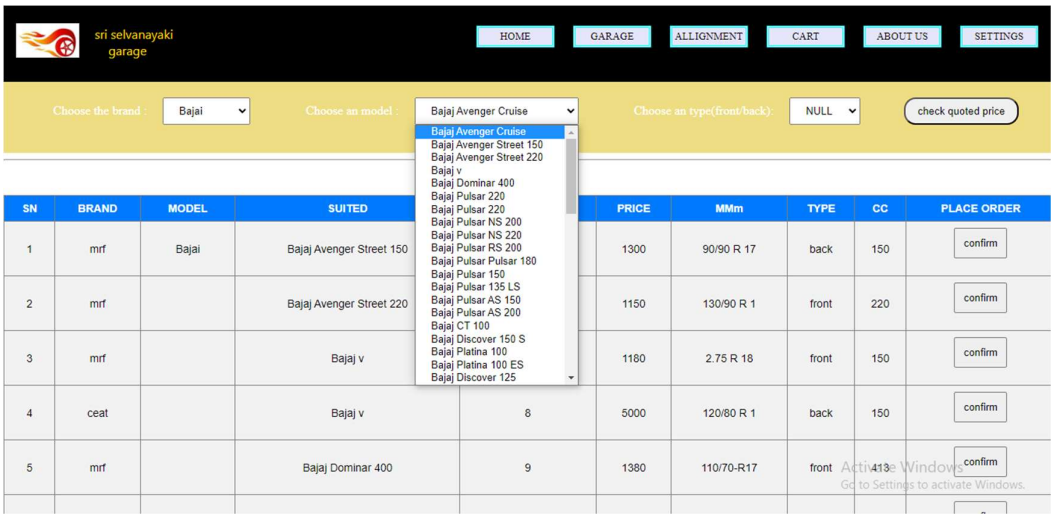

Figure B.3 Select Tier Page

This is the product page design for this application, here user can see the available tier. Different brands of tier are available in this website. Therefore, user can buy whatever prefer. Also search option available so, user no need manually looks for the list of products to select a specific product as shown in the Figure B.3.

FIXING APPOINTMENT

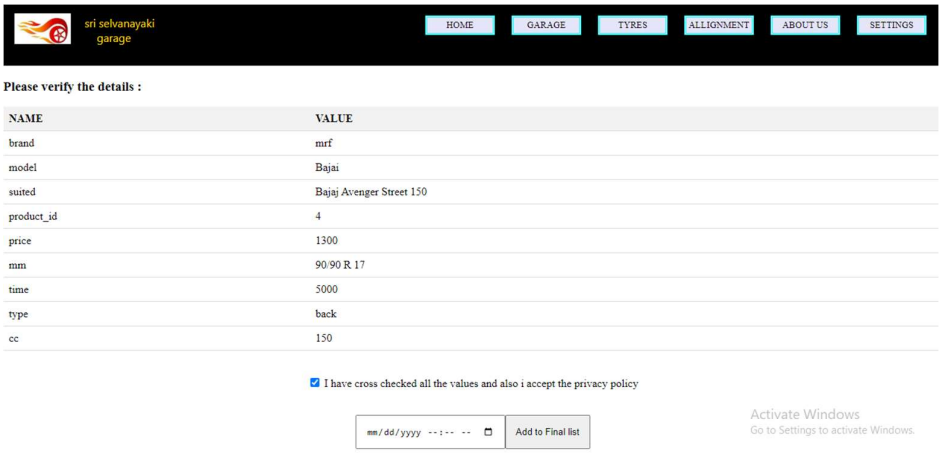

Figure B.4 Verifying Details Page

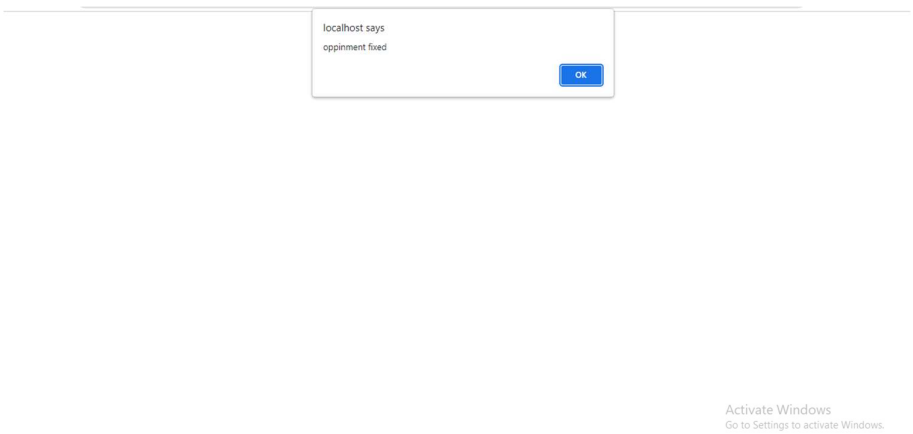

Figure B.5 Fixing Appointment Page

In this page, the user can see the details of selected products in the cart. This page shows product list with their total amount price and specifications. After verifying all the details the user can fix an appointment is shown in the Figure B.5

CHOOSE ISSUE:

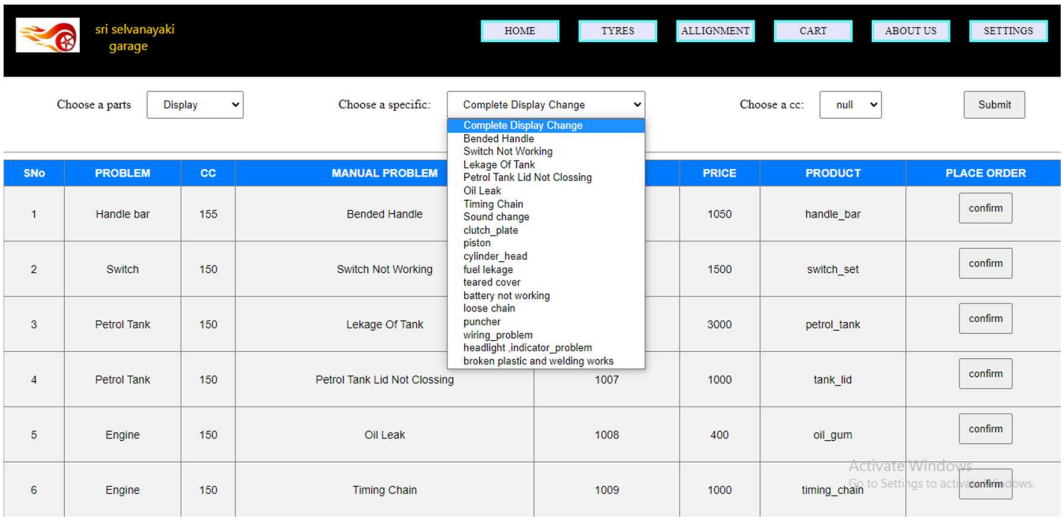

Figure B.6 Garage Page

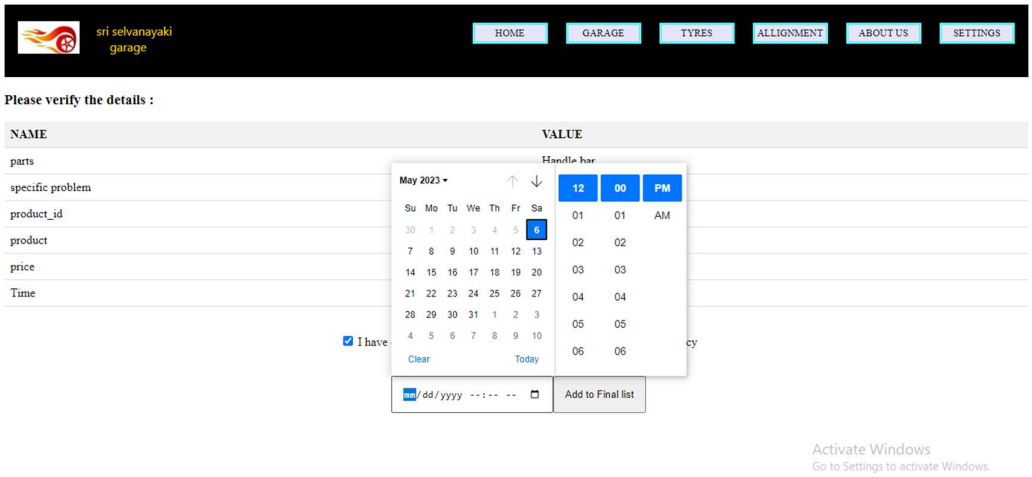

Figure B.7 Appointment Page

In this page user can choose the spares based on the problem in the vehicle as shown in Figure B.6. This product is redirected to the fix appointment page. Then the user verifies the details and fix a final appointment as shown in Figure B.7

ALIGNMENT PAGE

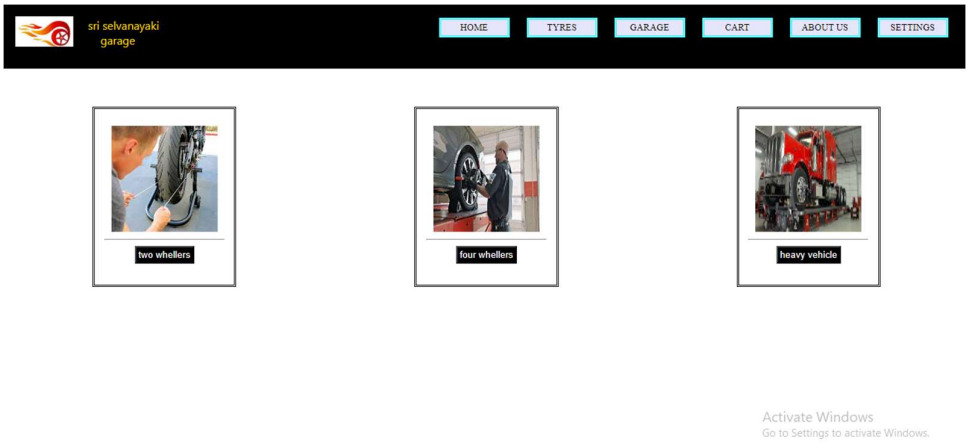

Figure B.8 Alignment Page

In this page, vehicle type is chosen for an alignment and fix an appointment for the alignment as shown in Figure B.8.

CART PAGE

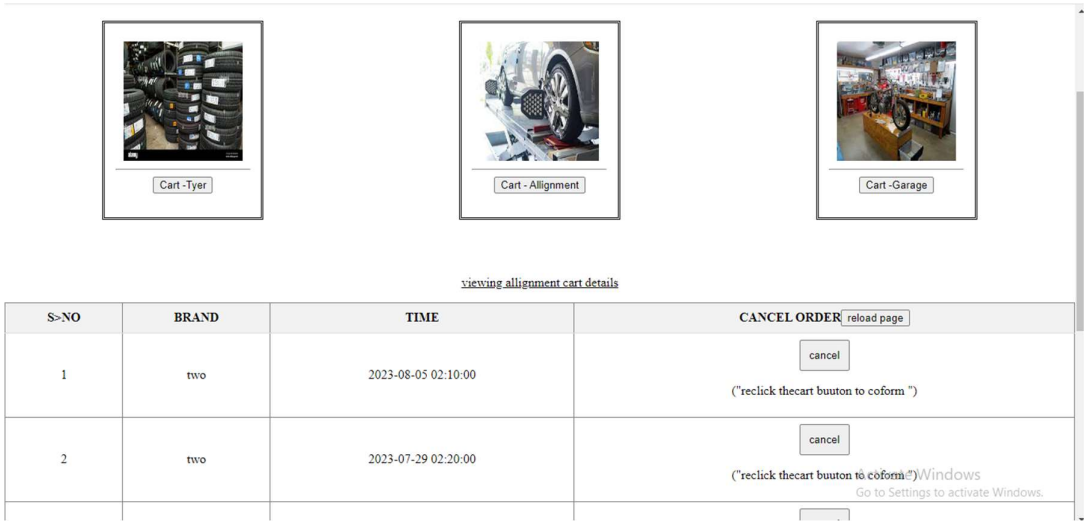

Figure B.9 Cart Page

In this cart page, user can view the prefixed appointment with all the complete details of the product which includes price, id, name, brand name and category as shown in Figure B.9

REPORT PAGE

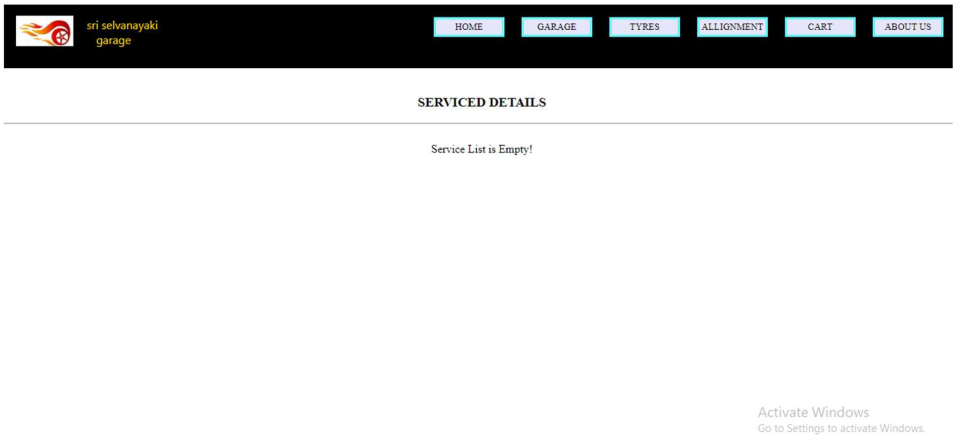

Figure B.10 Report Page

In the page, The user can view the list of serviced product list .If none of the services made Then the list will displayed an empty result as shown in Figure B.10.

ADMIN LOGIN

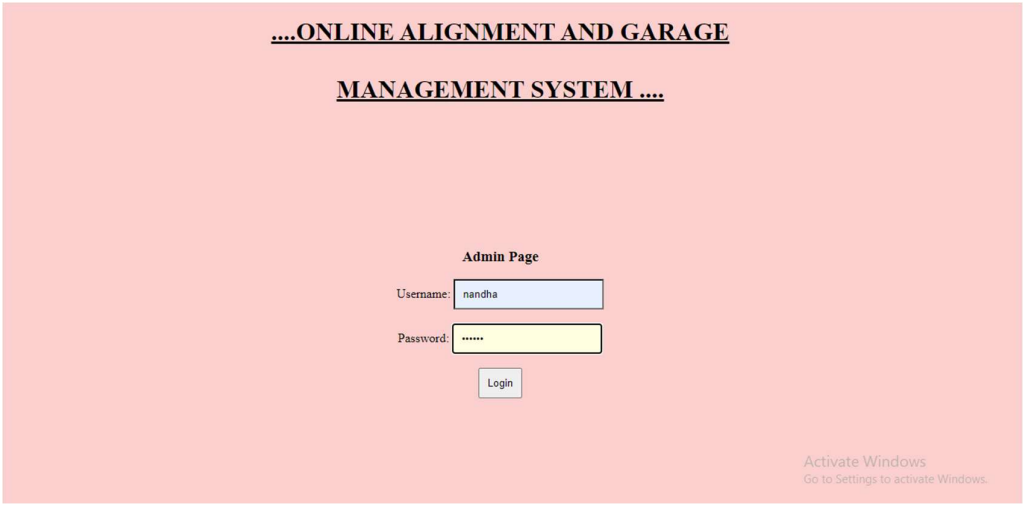

Figure B.11 Admin-login Page

This is an admin login page, Where the admin can use his credentials and after successful login ,the admin can also edit or delete the tire and spares as shown in the Figure B.11

ADMIN – EDIT USER PAGE

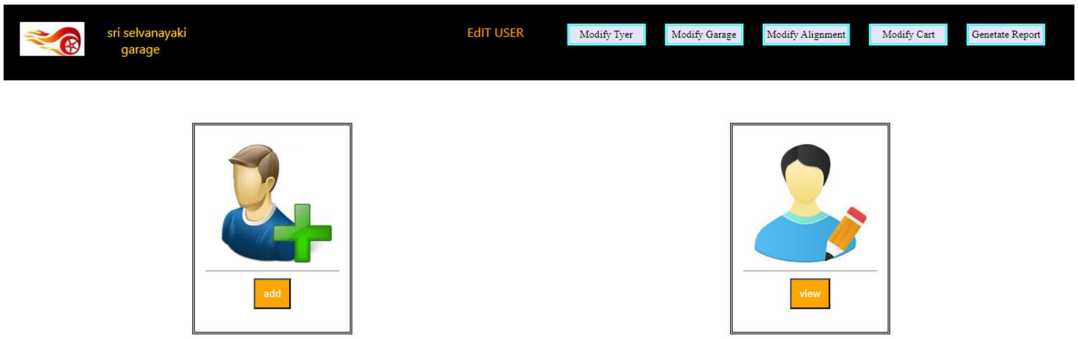

Figure B.12 Admin – Edit User Page

In the page, The admin can add a new user or update and delete an existing user as shown in Figure B.12.

ADMIN – EDIT GARAGE PAGE

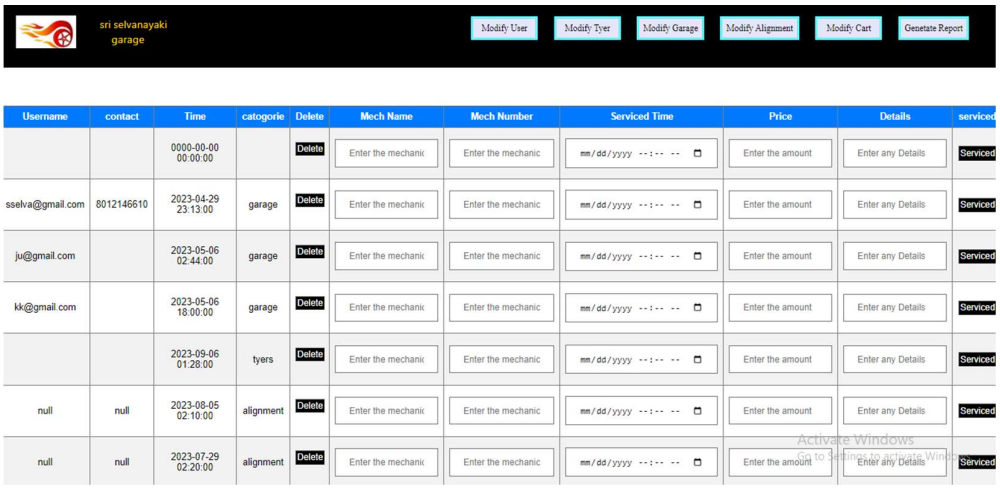

Figure B.13 Admin – Edit Product Page

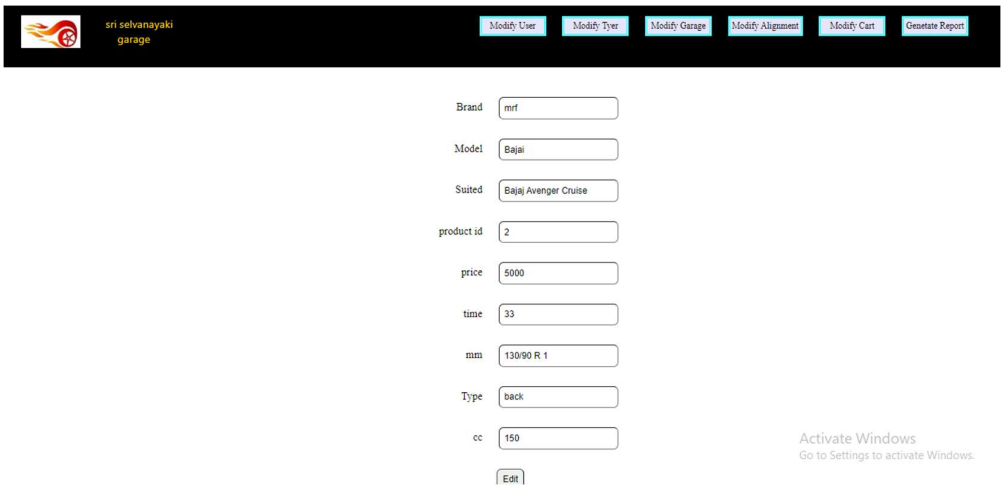

Figure B.14 Admin – Edit Product Page

In this page, Admin can see the complete details of spares if admin wishes he could add, update and delete a product as shown in Figure B.14.

REPORT GENERATION

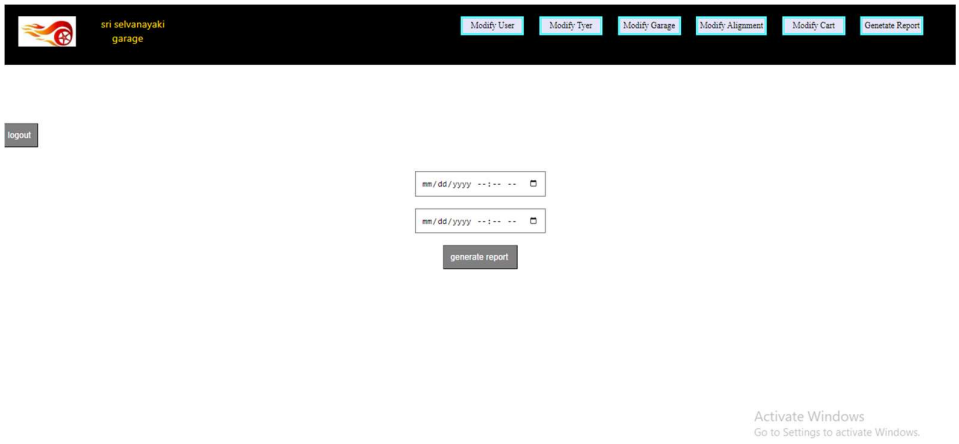

Figure B.15 Admin – Report Generation Page

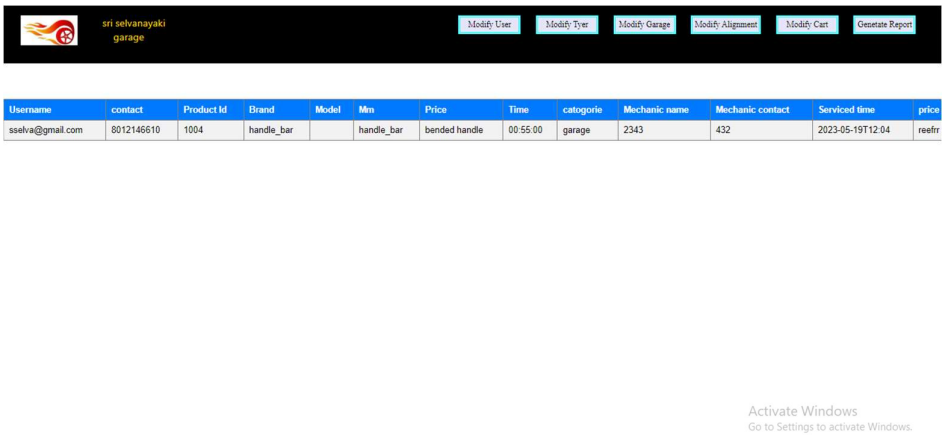

Figure B.16 Admin – Report Generation Page

In this page, Admin can generate and view report between two different dates are shown in the Figure B.16.

SETTINGS PAGE

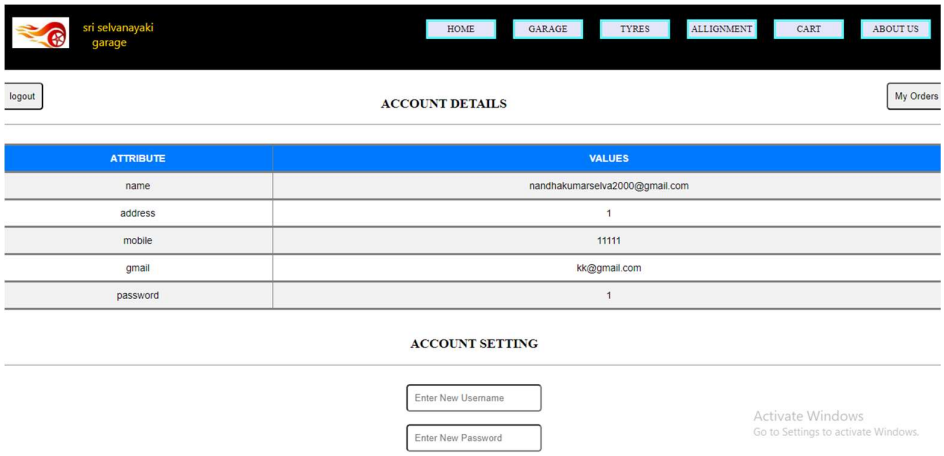

Figure B.17 Settings Page

In this page users can view his own credentials, if user wish then he could change he credentials in the settings page as shown in the Figure B.17.

REFERENCES

[1.] Rasmus Lerdorf, "Programming PHP 2 Edition, published 2002

[2.] Huge E. Williams, "Web Database Applications with PHP and MySQL". 2" Edition published 2002.

[3.] David Powers, "PHP Object-Oriented solutions", 3 Edition, published 2008

[4.] Williams Sanders, "Learning PHP Design Patterns", 3 Edition, published 2013.

[5.] Tim Converse, Joyce Park and Clark Morgan, "PHIPS and MySQL Bible", Wiley Publishing, Inc, 2004.

[6.] Larry Edward Ullman," PHP and MYSQL for Dynamic Websites", Pench pit press, Fifth Edition, 2003.

[7.] Don Gosselin, "Java Script", McGraw Hill, Fifth Edition, 2005

[8.] Jeffrey CJackson, "Web Technologies", Pearson, First Editon, 2000.

[9.] http://www.w3schools.com/php/php_syntax.asp

[10.] http://html.net/tutorials/php/lesson3.php

[11.] http://php.net/manual/en/tutorial.firstpage.php

[12.] https://1000projects.com