

# **MINI PROJECT REPORT**

**ON**

## **CAR SERVO**



**GLA University, Mathura**

**Department of Computer Science and Engineering**

Submitted in partial fulfilment of the requirements for the  
degree of Bachelor Technology

### **Team Members**

Yatin Mishra (171500401)  
Ayushman Awasthi (171500081)  
Sakshi Srivastava (171500285)  
Aman Mishra (171500035)

### **Supervised By**

Ms. Harvender Kaur  
(Trainer)



## DECLARATION

We hereby declare that the work which is being presented in the Mini Project “**Car Servo**”, in partial fulfilment of the requirements for Mini-Project Lab, is an authentic record of our own work carried under the supervision of Ms. Harvender Kaur GLA University, Mathura

Sakshi Srivastava  
Roll. No.: 171500285  
Course: B.tech  
Year: 3rd  
Semester: 6<sup>th</sup>

Ayushman Awasthi  
Roll. No.: 171500081  
Course: B.tech  
Year: 3rd  
Semester: 6<sup>th</sup>

Aman Mishra  
Roll. No.: 171500035  
Course: B.tech  
Year: 3rd  
Semester: 6<sup>th</sup>

Yatin Mishra  
Roll. No.: 171500401  
Course: B.tech  
Year: 3rd  
Semester: 6<sup>th</sup>



## CERTIFICATE

This is to certify that the project entitled “**Car Servo**” carried out in Mini Project – II Lab is a bona fide work done by **Ayushman Awasthi (171500081)**, **Aman Mishra (171500035)**, **Yatin Mishra (171500401)**, and **Sakshi Srivastava (171500285)**.

and is submitted in partial fulfilment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

**Ms. Harvender Kaur**  
(Trainer)

## **ABSTRACT**

This mini project in PHP named Car Servo was developed by me and my friend for project submission during the 6th semester. This project Car Servo is basically developed for customers to book spares and to get car services.

User can book their car for water wash on the day that user wishes. User can also modify their car by giving details that what type of modification that user requires. A Registered customer can choose and order the spares which are in the spares list.

The main purpose behind developing this application is to help people for their car services. This platform will surely reduce the stress of servicing or modifying cars.

## **ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to Dr (Prof). Anand Singh Jalal, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

We owe special debt of gratitude to Ms. Harvender Kaur, for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. She has showered us with all her extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies. Acknowledgement to any other faculty member Mr. Pankaj Kapoor . We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project.

# CONTENTS

Certificate.....	ii
Abstract.....	iii
Acknowledgments.....	iv
1. Introduction.....	1
1.1. Overview.....	1
1.2. Motivation.....	1
2. Software Requirement Analysis.....	2-5
2.1. Introduction.....	2
2.2. Module.....	2
3. System Specifications.....	6-19
3.1. Hardware Specification.....	6
3.2. Software Specification.....	6-8
3.2.1. WAMP.....	6
3.2.2. Php Designer.....	7
3.2.3. Web Browser .....	8
3.3. Language Used.....	8-19
3.3.1. HTML.....	8
3.3.2. CSS.....	10
3.3.3. PHP.....	13
3.3.4. JAVA SCRIPT.....	15
3.3.5. BOOTSTRAP.....	16
3.3.6. MYSQL.....	16
4. Software Design.....	20-30
4.1. DFD.....	20
4.2. Use Case.....	22

4.3. ER Diagram.....	23
4.4. Relation Schema.....	25
4.5. Database Design .....	26
4.6. Database Connectivity.....	27
4.7. Implementation of Database Operations.....	28
4.8. Trigger and Stored Procedure.....	29
4.8.1. Trigger Code.....	29
4.8.2. Procedure Code.....	29
5. Testing.....	31-32
5.1. Unit Testing.....	31
5.2. Integration Testing.....	32
5.3. System Testing.....	32
6. Implementation and User Interface.....	33-45
7. Conclusions.....	46
8. References/Bibliography.....	47

## **CHAPTER-1**

### **INTRODUCTION**

#### **1.1. OVERVIEW**

Online Car Servo is a Database application which is helpful for customers to book spares and to get car services. This mini-Project is implemented using HTML and PHP. The project also showcases embedded multimedia and Cascaded Style Sheet.

Operations supported by the application are insert, delete, update and retrieve. User can book their car for water wash on the day that user wishes. User can also modify their car by giving details that what type of modification that user requires. Admin has rights to insert and delete the spares from spares list and it is automatically updated in user page. Admin can add new spare parts for his customers. When spares are unavailable, Admin can delete the spares and he can show that they are out of stock. A Registered customer can choose and order the spares which are in the spares list. Admin can view the orders placed by the customer and they can deliver spares to the address specified by user.

#### **1.2. MOTIVATION**

People faced difficulty in centralizing the data used for managing the tasks performed in an online shopping and online car service booking. The theoretical process involved in database design has been practically implemented.



## CHAPTER-2

### SOFTWARE REQUIREMENT ANALYSIS

#### 2.1. INTRODUCTION

In this chapter, a complete description of the project development is discussed. The requirements of the project identified are showcased. The database design is done Using High-Level Conceptual Data Models.

#### 2.2. MODULE

A module is a separate unit of software or hardware. Typical characteristics of modular components include portability, which allows them to be used in a variety of systems, and interoperability, which allows them to function with the components of other systems. The term was first used in architecture.

This project will include the following modules:

---

- Sign up
- User Details.
- Home page
- Log in
- Accessories.
- Cart.
- Confirmation
- Modification.

## Car-Servo

- Water wash details.
- Payment page

### Functionalities of Modules:

---

- **Sign up (New User):** There is a signup module for the new users one can get himself/herself authenticated by signing up and getting him/her account on this website. Authenticated users will be able to view others stories or upload their videos and photography.
- **User Details:** User details table holds the information of the user who are signed up to your car servo.
- **Home Page:** We can see some car services and accessories sales here. In order to book any car service or if user needs to buy some spares then user needs to login with his account. If not, user has to register his details. Without registering his account, he cannot book the service or buy the car spares.
- **Log in:** If the user already has his account, he can login and get the service through this page.
- **Accessories:** Accessories table holds the description and prices of the car spares.
- **Cart:** When user want to buy the product later then, user can add the item into Cart table.
- **Confirmation:** After filling out the water wash booking form, user needs to confirm whether any changes required for his water wash. If any changes he required he can just change the water wash type, date of wash and garage via change button. Otherwise he can confirm his booking. He can pay the amount after the service.
- **Modification:** Modification table holds the booking information of modification and water wash respectively.
- **Water wash details:** Water wash table holds the booking information of modification and water wash respectively.
- **Payment Page:** Here user can buy the accessories and user can pay us via Paytm app.

### Functional Requirements

Functional requirements of a software project interpret the function of a part. It defines its functions, input and output. The typical functional requirements include: Application contains 2 modules:

- Admin module
- Customer module

#### Admin module

- Admin can able to register the product information.
- Admin can able to update the product information.
- Admin can able to view the details of the customers who are all logged in.
- Admin can able to see bills generated.

#### Customer module

- Customer must be register to the application.
- Customer can be able to get the all type of car services from the given garages.
- Customer can able to view the product details present in the accessory's hub.
- Customer can able to add the product to cart.

### Non- Functional Requirements

A non-functional requirement specifies the canon of the particular process not the particular judgment of the system and particular behavior of the process. Non-functional requirements define how the system works.

- This application is developed to make car service and shopping easier to customer so that it will save the time.
- This application work efficiently it works on all logical paths and independently and it should use the mobile data efficiently.

## Car-Servo

- This application is available during all the time at shopping.
- To run this application efficiently mobile network is the main important factor.
- Using of application is secure, because it displays appropriate information about the product.
- The system should capable to enhance with further technology in future to improve its features compared to the existing system such as offline billing.

## CHAPTER 3

### SYSTEM SPECIFICATIONS

#### 3.1. HARDWARE SPECIFICATIONS

Computer system with minimum requirements:

- Hard Disk: 20 GB
- RAM: 2 GB

#### 3.2. SOFTWARE REQUIREMENTS

**3.2.1. WAMP:** WAMP is an acronym that stands for Windows, Apache, MySQL, and PHP. It's a software stack which means installing WAMP installs Apache, MySQL, and PHP on your operating system (Windows in the case of WAMP). Even though you can install them separately, they are usually bundled up, and for a good reason too. What's good to know is that WAMP derives from LAMP (the L stands for Linux). The only difference between these two is that WAMP is used for Windows, while LAMP for Linux based operating systems.

Let's quickly go over what each letter represents:

---

- “W” stands for Windows, there's also LAMP (for Linux) and MAMP (for Mac).
- “A” stands for Apache. Apache is the server software that is responsible for serving web pages. When you request a page to be seen by you, Apache grants your request over HTTP and shows you the site.
- “M” stands for MySQL. MySQL's job is to be the database management system for your server. It stores all of the relevant information like your site's content, user profiles, etc.
- “P” stands for PHP. It's the programming language that was used to write WordPress.

Instead of installing and testing WordPress on your hosting account, you can do it on your personal computer (localhost). WAMP acts like a virtual server on your computer. It allows you

## Car-Servo

to test all WordPress features without any consequences since it's localized on your machine and is not connected to the web.

First of all, this means that you don't need to wait until files are uploaded to your site, and secondly – this makes creating backups much easier.

WAMP speeds up the work process for both developers and theme designers alike. What is more, you also get the benefit of playing around with your site to your heart's content.

However, to actually make the website go live, you need to get some form of hosting service and a Domain. See our beginner-friendly article about web hosting for more information. Or if you're already prepared to go online, in essence, WAMP is used as a safe space to work on your website, without needing to actually host it online.

WAMP also has a control panel. Once you install the software package, all of the services mentioned above (excluding the operating system that is) will be installed on your local machine. Whether you use WAMP or software packages for the other operating systems, it's a great way to save time. You won't have to upload files to a site and will be able to learn how to develop in a safe and care-free environment.

**3.2.2. PHP DESIGNER 8:** It is a lightning fast and powerful PHP IDE and PHP editor boosted with all the features to help you create amazing websites. Combine all your web-development into phpDesigner 8 with built-in editors for PHP, HTML5, CSS3 and JavaScript.

- **Live parsing:** The live parsing is a great help. As you type your code, phpDesigner is parsing and it and checking it for syntax errors. It reports any errors on the bottom of the screen as you type.
- **Function IntelliSense and descriptions:** As you begin typing a PHP function name, the IntelliSense will give you a list of available functions for the letters that you have already typed. It will also display a short description for that function, which is very handy if you are not sure of the exact description to choose. If you have are returning to a script, you can click on a PHP function name and the description will appear.
- **Interface:** The interface is visually pleasant and fairly well laid out.

- Project management tools: phpDesigner comes equipped with project manager tools like a debugging tracker.

There are loads of features with phpDesigner, as listed on their website. The above is just a short list of my favourites.

**3.2.3. WEB BROWSER:** A web browser (commonly referred to as a browser) is a software application for accessing information on the World Wide Web. When a user requests a particular website, the web browser retrieves the necessary content from a web server and then displays the resulting web page on the user's device. A web browser is not the same thing as a search engine, though the two are often confused. For a user, a search engine is just a website, such as Google Search, Bing, or DuckDuckGo, that stores searchable data about other websites. However, to connect to a website's server and display its web pages, a user must have a web browser installed.

The purpose of a web browser is to fetch information resources from the Web and display them on a user's device. This process begins when the user inputs a Uniform Resource Locator (URL), such as <https://en.wikipedia.org/>, into the browser. Virtually all URLs on the Web start with either http: or https: which means the browser will retrieve them with the Hypertext Transfer Protocol. In the case of https the communication between the browser and the web server is encrypted for the purposes of security and privacy.

Another URL prefix is file: which is used to display local files already stored on the user's device. Once a web page has been retrieved, the browser's rendering engine displays it on the user's device. This includes image and video formats supported by the browser.

Web pages usually contain hyperlinks to other pages and resources. Each link contains a URL, and when it is clicked, the browser navigates to the new resource. Thus, the process of bringing content to the user begins again.

### 3.3. LANGUAGE USED

**3.3.1. HTML:** HTML is short for Hypertext Mark-up Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page

contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: `<p>`

In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" `<p>` and "end tag" `</p>`. The text content of the element, if any, is placed between these tags.

Tags may also enclose further tag mark-up between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element.

The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the `<img>` used to embed images, the reference to the image resource.

Some elements, such as the line break `<br>`, do not permit any embedded content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

Many tags, particularly the closing end tag for the very commonly used paragraph element `<p>`, are optional. An HTML browser or other agent can infer the closure for the end of an element from the context and the structural rules defined by the HTML standard. These rules are complex and not widely understood by most HTML coders.

The general form of an HTML element is therefore: `<tag attribute1="value1" attribute2="value2">"content"</tag>`. Some HTML elements are defined as empty elements and take the form `<tag attribute1="value1" attribute2="value2">`. Empty elements may enclose no content, for instance, the `<br>` tag or the inline `<img>` tag. The name of an HTML element is the



name used in the tags. Note that the end tag's name is preceded by a slash character, /, and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case.

**3.3.2. CSS:** CSS stands for Cascading Style Sheets. It describes how HTML elements are to be displayed on screen, paper, or in other media. It saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

CSS helps you to keep the informational content of a document separate from the details of how to display it. The details of how to display the document are known as its style.

You keep the style separate from the content so that you can:

---

- Avoid duplication
- Make maintenance easier
- Use the same content with different styles for different purpose

Your web site might have thousands of pages that look similar. Using CSS, you store the style information in common files that all the pages share. When a user displays a web page, the user's browser loads the style information along with the content of the page. When a user prints a web page, you might provide different style information that makes the printed page easy to read.

In general, you use HTML to describe the content of the document, not its style; you use CSS to specify its style, not its content. There are exceptions to this rule, of course, and HTML also provides some ways to specify style. For example, in HTML you can use a <b> tag to make text bold, and you can specify the background colour of a page in its <body> tag. When you use CSS, you normally avoid using these HTML style features so that all your document's style information is in one place.

### Top Uses of CSS:

- **Compatibility:** While using CSS, a user can be stress-free with its older language versions as it is well compatible even with the older versions. Because of this, even if the applications of CSS is developed in some older version of programming language and if

the developers integrate the same with new enhancements then CSS can be easily incorporated and implemented with those changes easily. So, the changes associated with the predecessor of CSS, need not be removed and a developer can update the existing code successfully.

- **E-Commerce Domain:** Uses of CSS has played a very important role in the Ecommerce Domain. There is a different scale of industries present in the Ecommerce domain and CSS has helped the application frameworks styling and look-views which are used from small to large scale industries. CSS styling can be used to associate directly with e-commerce web. Also, different add-ins and library files that are present can be modified and updated by CSS libraries and useful while creating the web applications and the source codes associated with it can be implemented by uses of CSS in HTML framework to develop ecommerce web platform from the scratch.
- **Website Maintenance:** Another most common reason, why the uses of CSS are important? CSS plays a very important role while performing website maintenance. It makes website maintenance much easier. The CSS file makes the website look and feels more flexible and it can be altered in a more convenient manner. Also, it makes the HTML formatting and the corresponding data elements modification easier. Because of which, the website maintenance becomes more convenient from the development perspective.
- **Social Media Impact:** Uses of CSS in HTML frameworks are also incorporated with the social media website developments as well. Facebook applications can directly correlate with the corresponding frameworks. The HTML client library files can be implemented based on CSS stylings to develop the application and can work on different extensions. These social media-based platforms can be inter-related with the frameworks and can be used to develop few enhancements from the end user's perspective. Thus, the CSS styling and updating the user interphase becomes easier and this creates a direct impact on the social media platforms.
- **Web-based Online Community and UI Approach:** Based on the current standard, to develop any web-based application or any online community, the application of CSS can be implemented in different ways. There are different style sheet frameworks present in CSS and those can be implemented easily over here. Using CSS, the styling of the own

online community can be developed. There are also different add-ins present and those can be incorporated by the uses of CSS frameworks to develop the look and feel of the web-based community.

- **Easy Accessibility:** From the accessibility point of view, applications of CSS provides much better solutions that allow users to update the user interphase to suit the business requirements. It also allows the web pages to be easily rendered by different devices like speaking browser, PDAs etc. This actually provides a much deeper impact while considering the look and feel modification of a web page from end-user and business perspective.
- **Image File Handling:** In the case of image handling, uses of CSS provides the styling Library, and this helps to output images along with the XML to the browser. Initially, it was a bit difficult to update and style the existing image. But now using the CSS files, output images can be received in different formats like jpeg, png, and gif and this can be modified to the specified styling formats as per the requirement. This feature will also allow editing different image types and is also used to create thumbnails, watermarks, image cropping etc.
- **Handling Dynamic Website Templates:** Dynamic Website Templates can be created and handled by uses of CSS in HTML framework and it basically helps while adding and editing web pages and elements to the web server and sites. By adding CSS extension to the concerned HTML pages and incorporating the same with the server-side templates, it makes easier to handle the dynamically allocated elements. So, using these templates the handling of dynamic elements can be more organized and properly implemented by CSS frameworks. These elements can also be used to style the pages in a dynamic pattern while using the CSS templates
- **Handling Flash Animation and Effects:** With the help of CSS Flash files, flash elements can be directly placed and handled in the website. There is a presence of inbuilt frameworks and styling sheets in CSS which can be used and to handle the situation. The animations can be directly created, and effects can be updated using the CSS frameworks. The required flash files can be processed and implemented to create movies and animation in the web pages with the help of CSS properties.

- **End User and Server-side Representation:** CSS files can directly interact with the user end server-side response and can be used for web server-side interphase styling purpose. This allows producing the better web representation from the end user's perspective.

After discussing all the above uses of CSS, it can be concluded that CSS can be very useful while doing the web application interphase representation and styling across different domains. Thus, based on the project need, business requirements, CSS can be implemented to achieve the desired goal.

**3.3.3. PHP:** PHP stands for Hypertext Pre-processor (no, the acronym doesn't follow the name). It's an open source, server-side, scripting language used for the development of web applications. By scripting language, we mean a program that is script-based (lines of code) written for the automation of tasks.

- What does open source mean? Think of a car manufacturer making the secret to its design models and technology innovations available to anyone interested. These design and technology details can be redistributed, modified, and adopted without the fear of any legal repercussions. The world today might have developed an amazing supercar!
- Web pages can be designed using HTML. With HTML, code execution is done on the user's browser (client-side). On the other hand, with PHP server-side scripting language, it's executed on the server before it gets to the web browser of the user.
- PHP can be embedded in HTML, and it's well suited for web development and the creation of dynamic web pages for web applications, e-commerce applications, and database applications. It's considered a friendly language with abilities to easily connect with MySQL, Oracle, and other databases.

### PHP Use:

PHP scripts can be used on most of the well-known operating systems like Linux, Unix, Solaris, Microsoft Windows, MAC OS and many others. It also supports most web servers including Apache and IIS. Using PHP affords web developers the freedom to choose their operating system and web server.

In PHP, server-side scripting is the main area of operation. Server-side scripting with PHP involves:

- PHP Parser: a program that converts source and human readable code into a format easier for the computer to understand
- Web server: a program that executes files that form web pages from user requests
- Web browser: an application used to display content on the World Wide Web

In this instance, with the use of just a PHP parser, the PHP script can be executed without a server program or browser. This use of the PHP script is normally employed for simple text processing tasks, like task schedulers.

PHP can also be used for creating client-side applications, like desktop applications. Desktop applications are usually characterized by a graphic user interface. With knowledge in using the advanced features of PHP, such as PHP-GTK, these client-side applications can be developed.

There are a lot of reasons to know and love PHP, probably the most potent and valid of which is this: it's used and runs everywhere the web does. Your cheap little \$3 per month hosting account may let you run a web application in Python or Ruby if you shop carefully. But it'll definitely run PHP. This means that you can count on it wherever you are. And because it runs everywhere, and is easy to get started with, a lot of very popular software is written in PHP. WordPress is the example that's both largest and most familiar to me, but tools like Joomla, Drupal, Magento, Expression Engine, Vbulletin (yep, that's still around), MediaWiki, and more are all running PHP on the server. And there are more PHP application frameworks than you can shake a stick at as well: Symfony, Zend, Laravel, Aura, CakePHP, Yii, and even the venerable CodeIgnitor. Surely you can make a list of web frameworks of some length for almost any other language. And for the commonly used web languages like Python, Ruby, or Node/JavaScript you may even be able to amass a numerically competitive list. But the sheer volume of sites running PHP is immense.

WordPress proudly boasts that it powers more than 30% of the internet. You don't need to even trust that fact to realize that a lot of the internet must be using PHP if that fact is even conceivably true.

A lot of people new to programming are looking for very general face-offs of languages. So, here they come. All the languages I'm going to write up here have the following traits in common with PHP:

- They're open-source. What this means is that you can use the underlying language for free (no cost), and you are able to see and understand the underlying program if you wish to.
- They're often used for web development. Pretty straight-forward, these are languages used a lot for web development. Some are also widely used outside of that venue, but not all.
- They're high-level, loosely- and dynamically-typed. This means that a variable can change types, and that you don't have to define when you define a variable what type of things (numbers vs strings vs objects, etc.) it'll store. This is generally favored for web programming, but not universally.
- Their communities are good-sized or better. There are a lot of interesting languages that have the qualities listed above, but which don't have a large community of practice. I'm leaving them aside here.

**3.3.4. JAVA SCRIPT:** JavaScript is one of the most popular programming languages in the world. Along with HTML and CSS, JavaScript forms the basis of front-end web development, allowing the creating of interactive elements. This is something of a simplistic view, and in this tutorial, I'm going to help you figure out exactly what is JavaScript used for.

Unfortunately, there is a lot of low-quality information out there that claims to be from 'authority sources'. This is especially true when it comes to things like online JavaScript courses which can to help you discover what is JavaScript used for.

JavaScript is one of the most popular programming languages in the world. Although it was designed to help front-end web developers create interactive elements, the use cases of JavaScript have rapidly expanded to include things like back-end web development, game creation, and even mobile app development. If any of these things interest you, you should consider learning JavaScript.

**3.3.5. BOOTSTRAP:** Bootstrap is a framework to help you design websites faster and easier. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc. It also gives you support for JavaScript plugins.

Here's what you need to know:

First of all, Bootstrap is the most popular framework for creating layouts. Here are some additional reasons to use Bootstrap:

- Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
  - Mobile-first styles are part of the framework
  - Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)
- Bootstrap has a big community and friendly support.
  - Bootstrap is easy to set up and create a working layout in less than an hour
  - You don't need to know HTML and CSS well to use bootstrap, it's a plus if you're a backend developer and need to do some UI changes.
  - It's fully customizable, I can choose which components I'd like to use and use variables file to get do even more colour and behaviour customization.
  - When you update the version of Bootstrap, you won't see tons of errors because their core team cares about backwards compatibility.
  - Bootstrap offers a lot of helper classes that make development of a responsive website easy and fast.
  - Bootstrap's components are well-adopted to the ecosystem of popular JS MVC Frameworks like Angular. Bootstrap provides several ways to include it into project

**3.3.6. MY SQL:** MySQL is a relational database management system based on SQL Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.

The most common use for MySQL however, is for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products

for an online store. In association with a scripting language such as PHP or Perl (both offered on our hosting accounts) it is possible to create websites which will interact in real-time with a MySQL database to rapidly display categorized and searchable information to a website user.

### Reasons to Choose MySQL:

- **Secure Money Transactions:** MySQL transactions work as a single unit, which means unless and until every individual operational stage is successfully completed, the transaction is not cleared. So, if an operation fails at any stage, the entire transaction happening within that group fails. MySQL ensures that financial transactions have data integrity, so customers can make worry-free transactions online. The money is not debited until the entire process is completed and in case of failure, every process is reverted to the previous stage.
- **On-Demand Scalability:** MySQL comes with the advantage of unmatched flexibility that facilitates efficient management of deeply embedded applications, even in gigantic data centres that stack tremendous amounts of mission-critical information. It enables complete customization to cater to the unique requirements of e-commerce businesses with a much smaller footprint. MySQL provides ultimate platform flexibility to enterprises who need additional features and functionalities for their database servers.
- **High Availability:** Consistent availability is the stalwart feature of MySQL enterprises that deploy it can enjoy round-the-clock uptime. MySQL comes with a wide variety of cluster servers and master-slave replication configurations that enable instant failover for uninterrupted access. Whether you run an e-commerce website or a high-speed processing system, MySQL is designed to process millions of queries and thousands of transactions while ensuring unique memory caches, full-text indexes and optimum speed.
- **Rock-Solid Reliability:** Protecting sensitive business information is the primary concern of every enterprise. MySQL ensures data security with exceptional data protection features. Powerful data encryption prevents unauthorized viewing of data and SSH and SSL supports ensure safer connections. It also features a powerful mechanism that restricts server access to authorized users and has the ability to block users even at the man-machine level. Finally, the data backup feature facilitates point-in-time recovery.



- **Quick-Start Capability:** You can go from software download to complete installation in just 15 minutes. MySQL is exceptionally quick, regardless of the underlying platform. It features self-management capabilities like auto restart, space expansion and automatic configuration changes for ease of management. It also comes with a comprehensive set of migration tools and a fully loaded graphical management suite. MySQL enables real-time performance monitoring for timely troubleshooting of operational issues from a single workstation.

For all of these reasons, organizations are using MySQL to instantly develop and launch apps. From retail and finance, to healthcare and manufacturing, many industries are capitalizing on the cost-effectiveness, efficiency and reliability of MySQL to deliver seamless services and boost their revenue. But when it comes to optimizing MySQL deployments for performance and availability, enterprises face the following challenges, because scaling MySQL needs much more than just a database.

In regard to the general definition, MySQL is an open source relational database management system (RDBMS) with a client-server model. RDBMS is a software or service used to create and manage databases based on a relational model. Now, let's take a closer look at each term:

- **Database:** A database is simply a collection of structured data. Think of taking a selfie: you push a button and capture an image of yourself. Your photo is data, and your phone's gallery is the database. A database is a place in which data is stored and organized. The word "relational" means that the data stored in the dataset is organized as tables. Every table relates in some ways. If the software doesn't support the relational data model, just call it DBMS.
- **Open source:** Open source means that you're free to use and modify it. Anybody can install the software. You can also learn and customize the source code to better accommodate your needs. However, The GPL (GNU Public License) determines what you can do depending on conditions. The commercially licensed version is available if you need more flexible ownership and advanced support.
- **Client-server model:** Computers that install and run RDBMS software are called clients. Whenever they need to access data, they connect to the RDBMS server. That's the "client-server" part.

MySQL is one of many RDBMS software options. RDBMS and MySQL are often thought to be the same because of MySQL's popularity. To name a few big web applications like Facebook, Twitter, YouTube, Google, and Yahoo! all use MySQL for data storage purposes. Even though it was initially created for limited usage, now it's compatible with many important computing platforms like Linux, macOS, Microsoft Windows, and Ubuntu.

MySQL and SQL are not the same. Be aware that MySQL is one of the most popular RDBMS software's brand name, which implements a client-server model. So, how do the client and server communicate in an RDBMS environment? They use a domain specific language – Structured Query Language (SQL). If you ever encounter other names that have SQL in them, like PostgreSQL and Microsoft SQL server, they are most likely brands which also use SQL syntax. RDBMS software is often written in other programming languages, but always use SQL as their primary language to interact with the database. MySQL itself is written in C and C++.

Computer scientist Ted Codd developed SQL in the early 1970s with an IBM based relational model. It became more widely used in 1974 and quickly replaced similar, then-outdated languages, ISAM and VISAM. History aside, SQL tells the server what to do with the data. It is similar to your WordPress password or code. You input it into the system to gain access to the dashboard area.

In this case, SQL statements can instruct the server to perform certain operations:

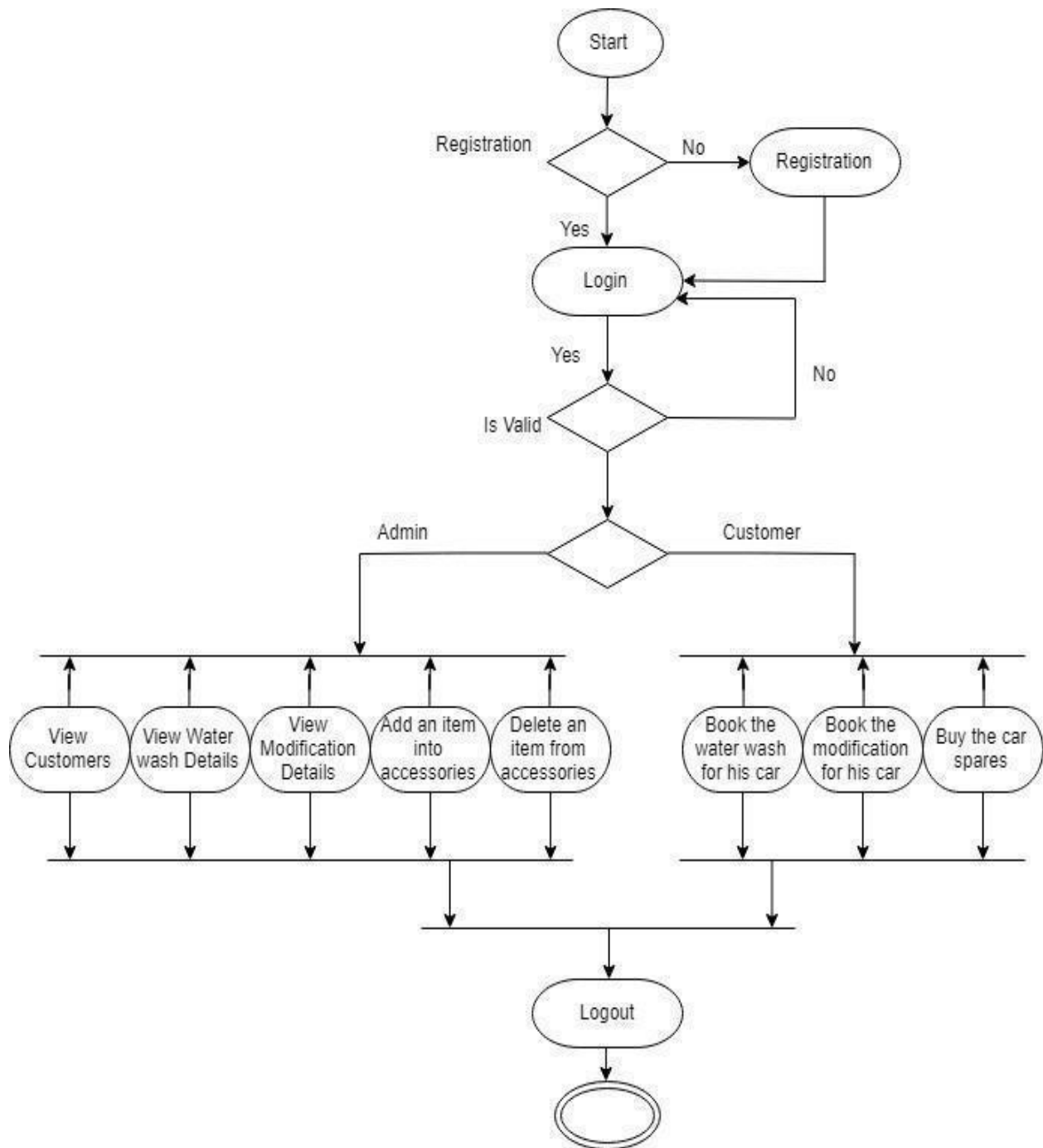
- **Data query:** requesting specific information from the existing database.
- **Data manipulation:** adding, deleting, changing, sorting, and other operations to modify the data, the values or the visuals.
- **Data identity:** defining data types, e.g. changing numerical data to integers. This also includes defining a schema or the relationship of each table in the database.
- **Data access control:** providing security techniques to protect data, this includes deciding who can view or use any information stored in the database.

## CHAPTER 4

### SOFTWARE DESIGN

#### 4.1. DFD (Data flow diagram)

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or sy

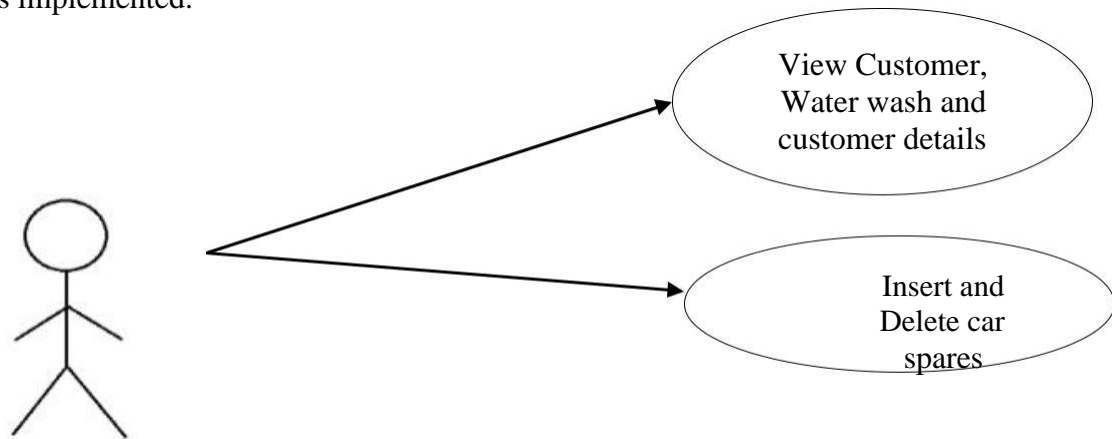


stems.

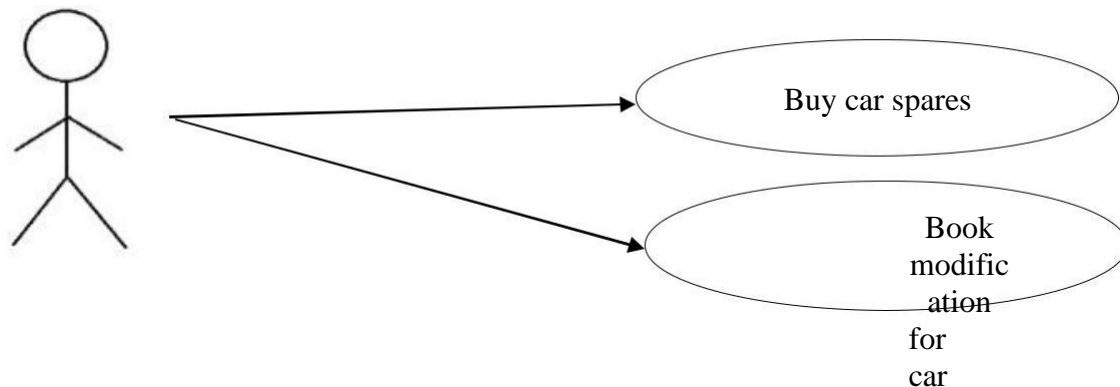
Fig 1: DFD

## 4.2. USE CASE DIAGRAM

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration. They provide a good high-level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.



Admin



Customer



Dept. of CEA, GLAU, Mathura

Fig 2: Use Case Diagram

### **4.3. ER DIAGRAM**

Entity relationship diagram displays the relationships of entity set stored in a database. In other words, we can say that ER diagrams help you to explain the logical structure of databases. At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make



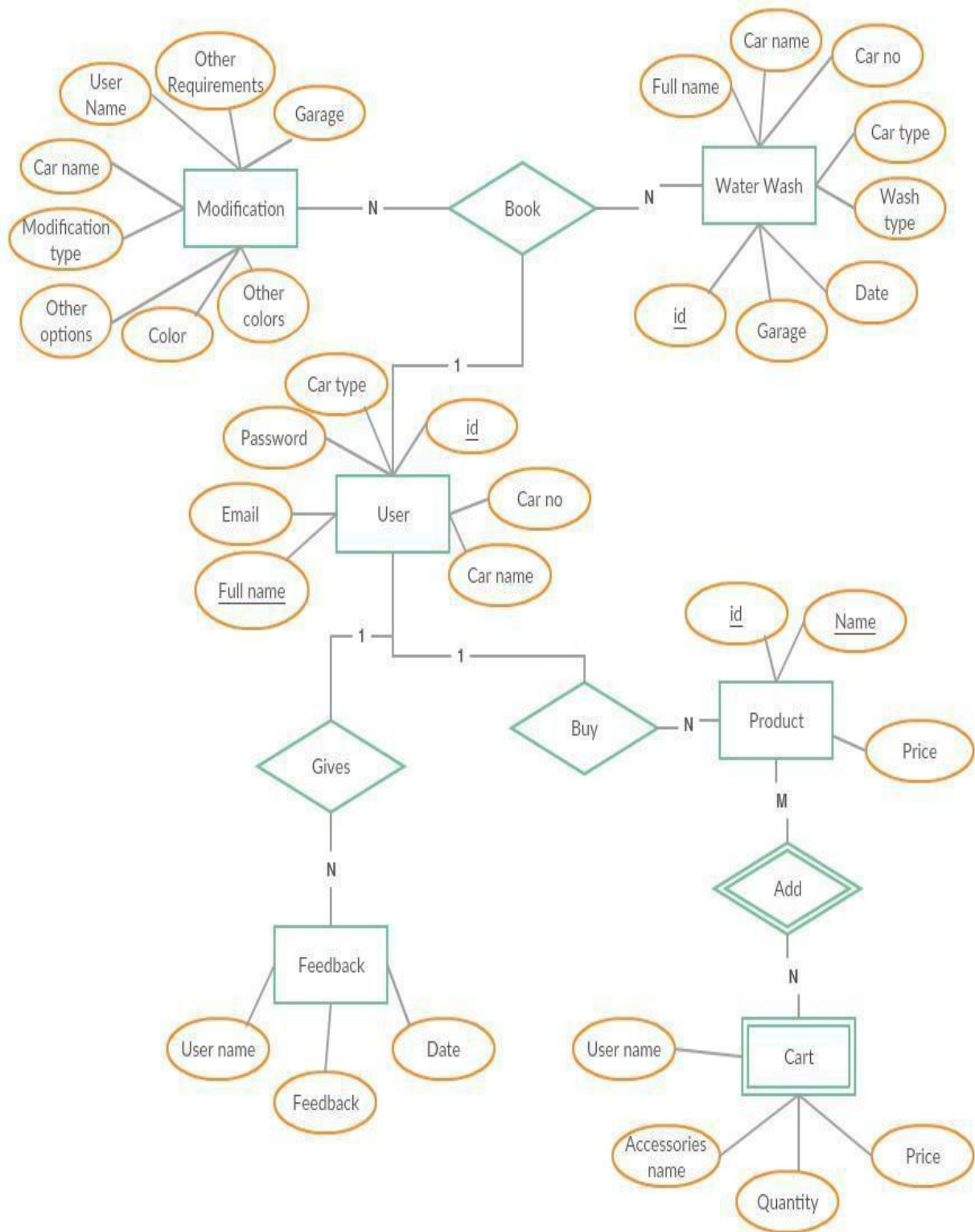


Fig 3: ER Diagram

#### 4.4.Relational Schema

The relational schema diagram has been derived from the ER-Diagram in Figure 4 using the ER-Relational mapping algorithm

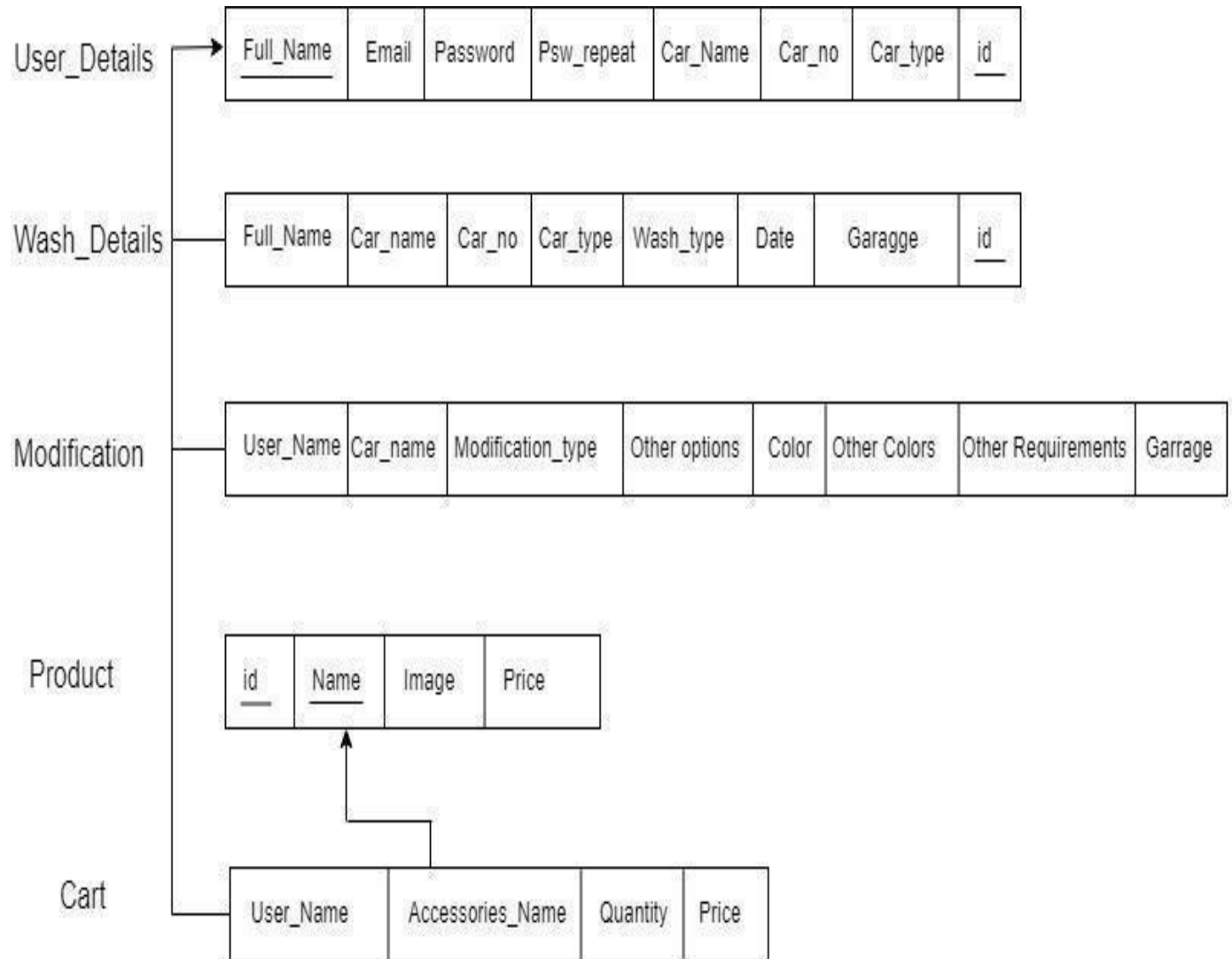


Fig 4: Relational Schema Diagram Showing the Primary key and Foreign key relationships

## 4.5. Database Design

### 1. USER DETAILS:

```
CREATE TABLE user_details (Full Name varchar(30) NOT NULL,  
Email varchar(30) NOT NULL,  
Password varchar(30) NOT NULL,  
psw_repeat varchar(30) NOT NULL,  
Car_name varchar(30) NOT NULL,  
Car_no varchar(30) NOT NULL,  
Car_type varchar(30) NOT NULL,  
idint(11) NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (id));
```

### 2. WASH DETAILS:

```
CREATE TABLE wash_details (  
Full_Name varchar(30) NOT NULL,  
cname varchar(20) NOT NULL,  
cno varchar(20) NOT NULL,  
ctype varchar(20) NOT NULL,  
wtype varchar(20) NOT NULL,  
datew date NOT NULL,  
gd varchar(35) NOT NULL,  
idint(10) NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (id));
```

### 3. PRODUCT DETAILS:

```
CREATE TABLE tbl_product (  
idint(11) NOT NULL AUTO_INCREMENT,  
name varchar(255) NOT NULL,  
image varchar(50) NOT NULL,  
price varchar(255) NOT NULL,  
PRIMARY KEY (id));
```

#### **4. CART:**

```
CREATE TABLE cart (  
    user_name varchar(20) NOT NULL DEFAULT "",  
    accessories_name varchar(40) NOT NULL DEFAULT  
    "", quantity int(11) DEFAULT NULL, price double(10,2)  
    DEFAULT NULL,  
    PRIMARY KEY (user_name, accessories_name));
```

#### **5. MODIFICATION DETAILS:**

```
CREATE TABLE modification (  
    user_name varchar(50) NOT NULL,  
    customer_name varchar(50) NOT NULL,  
    current_car varchar(20) NOT NULL,  
    Modification_type varchar(100) NOT NULL,  
    other_options varchar(30) NOT NULL,  
    color varchar(30) NOT NULL,  
    other_colors varchar(20) NOT NULL,  
    other_req varchar(50) NOT NULL,  
    garage_name varchar(50) NOT NULL);
```

### **4.6. Database Connectivity**

The following code is used to connect the database by giving username, password, server name and database name.

```
<?php  
// Create connection  
$conn = new mysqli('localhost', 'root', "", "car_servo");  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
echo "Connected successfully."  
mysqli_select_db($conn, "car_servo");
```

## Car-Servo

```
echo "DB selected Successfully.";
mysqli_close($conn);
?>
```

### 4.7. Implementation of Database Operations

Insert, delete, update and retrieve php code used in the project to implement database operations are as follows.

#### //PHP code to insert data

```
$sql = "INSERT INTO wash_details (Full_Name,cname,cno,ctype,wtype,datew,gd)
VALUES ('$uname','$cname','$cno','$ct','$wt','$dt','$gd')";
```

#### //PHP code to delete data

```
$sql="DELETE from tbl_product WHERE name='$u'";
```

#### //PHP code to update data

```
$sql="UPDATE wash_details set wtype='$wt',datew='$dt',gd='$gd' where Full_name='$u' AND
id='$co'";
```

#### //PHP code to view data

```
$sql = mysqli_query($conn,"SELECT * FROM wash_details where id=(SELECT max(id) from
wash_details)");
```

```
while($row=mysqli_fetch_array($sql,MYSQLI_ASSOC))
```

```
{
```

```
echo nl2br("Hello " . $row["Full_Name"])."<br>";
```

```
echo nl2br("Your Car, " . $row["cname"])."<br>";
```

```
echo nl2br("has booked for " . $row["wtype"])." wash<br>";
```

```
echo nl2br("on " . $row["datew"])."<br>";
```

```
echo nl2br("at " . $row["gd"])." service station<br>";}
```

## 4.8.Trigger and Stored Procedure

Following trigger and stored procedure have been implemented in the project.

### 4.8.1. TRIGGER CODE

This trigger code will update the number of users after new user gets registered.

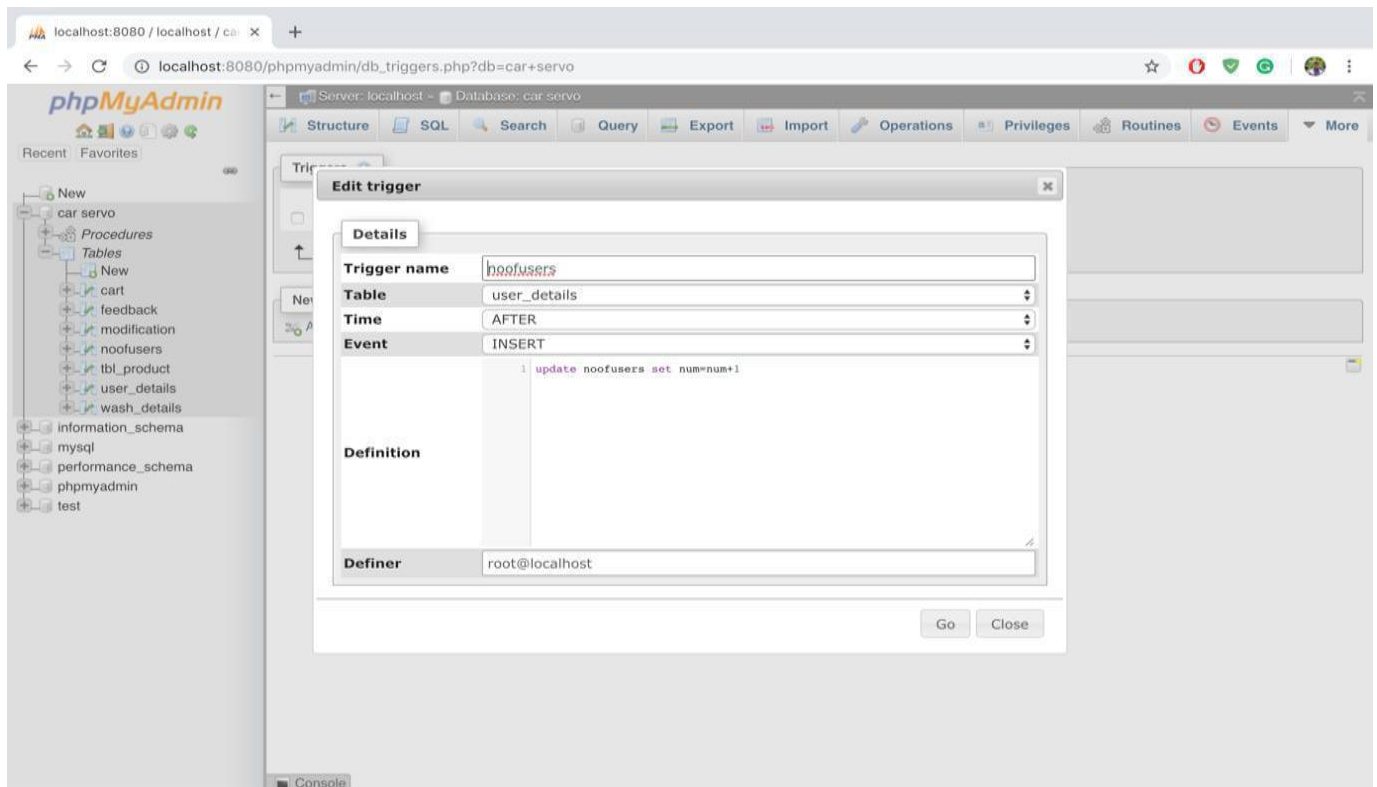


Fig 5: Adding trigger in MYSQL

### 4.8.2. PROCEDURE CODE

This procedure code displays the number of user's present in our project and that is displayed in the admin page.

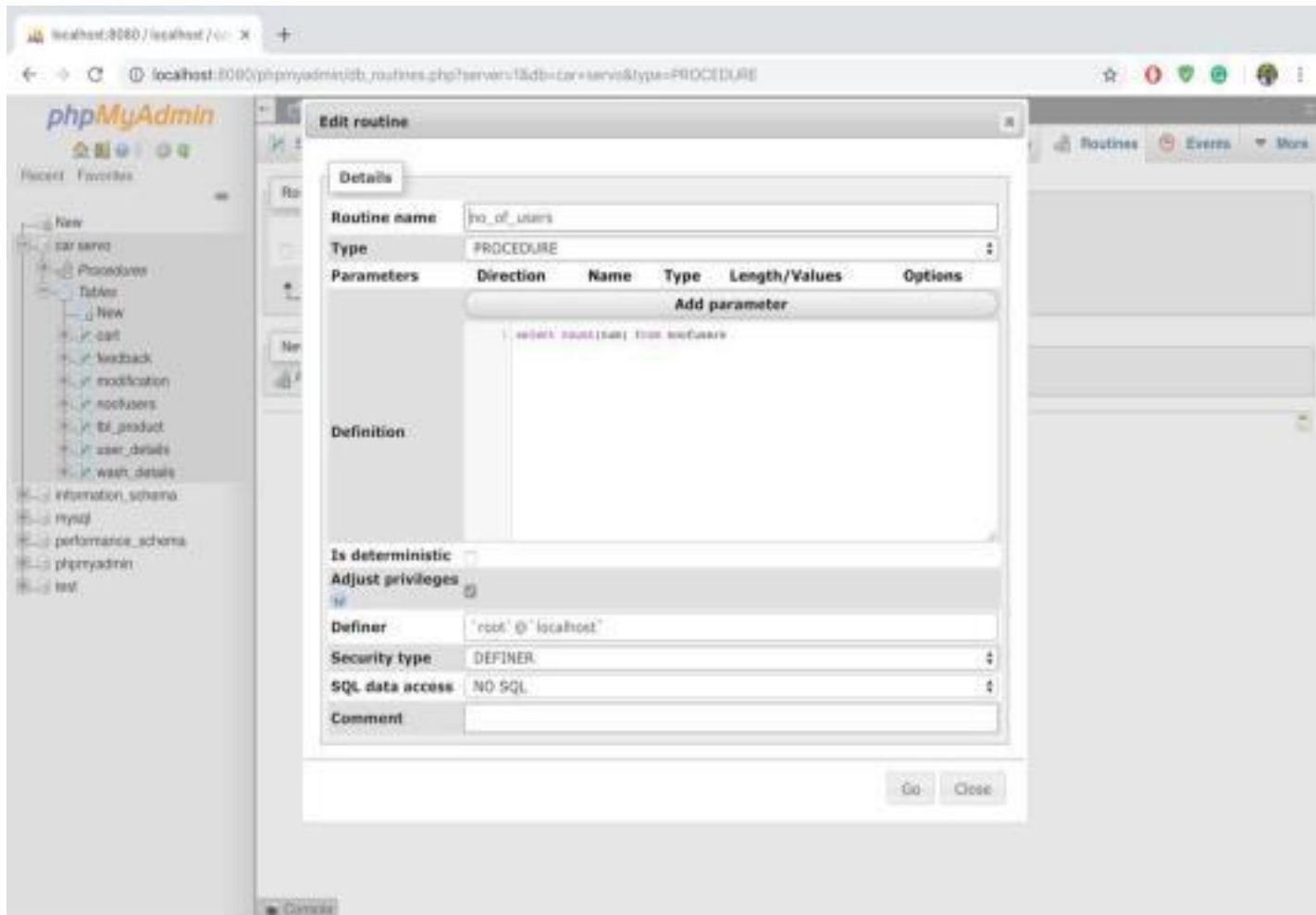


Fig 6: Stored procedure for number of user's present currently.

## CHAPTER 5

### TESTING

A process of executing a program with the explicit intention of finding errors, that is making the program fail. It is the process of detecting errors and performs a very critical role for quality assurance, also for ensuring the reliability of software. The results of testing are used later on during maintenance also. Testing Approaches: There are three types of software testing approaches.

- **White Box Testing:** It is also called as Glass Box, Clear Box, and Structural Testing. White Box Testing is based on applications internal code structure. In white-box testing, internal perspectives of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level.
- **Black Box Testing:** It is also called as Behavioral/Specification-Based/Input/output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.
- **Grey Box Testing:** Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

#### 5.1. UNIT TESTING

Unit Testing is done to check whether the individual modules of the source code are working properly. i.e. testing each and every unit of the application separately by the developer in the developer's environment. It is AKA Module Testing or Component Testing. It concentrates on each unit of software as implemented in source code and is a white box oriented. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. In the unit testing, the steps can be conducted in parallel for multiple components in this project we tested all the modules individually related to main function codes and attacks also.



## **5.2. INTEGRATION TESTING**

Here focus is on design and construction of the software architecture. Integration Testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions. In this project the main system is formed by integrating all the modules. When integrating all the modules we have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before integration.

## **5.3. SYSTEM TESTING**

It's a black box testing. Testing the fully integrated application this is also called as end to end scenario testing. To ensure that the software works in all intended target systems. Verify thorough testing of every input in the application to check for desired outputs. Testing of the users experiences with the application

## CHAPTER 6

### IMPLEMENTATION AND USER INTERFACE

In this chapter the results of the project are discussed. The snapshot of the project showing various functionalities like insert, delete, update and retrieval are showcased.

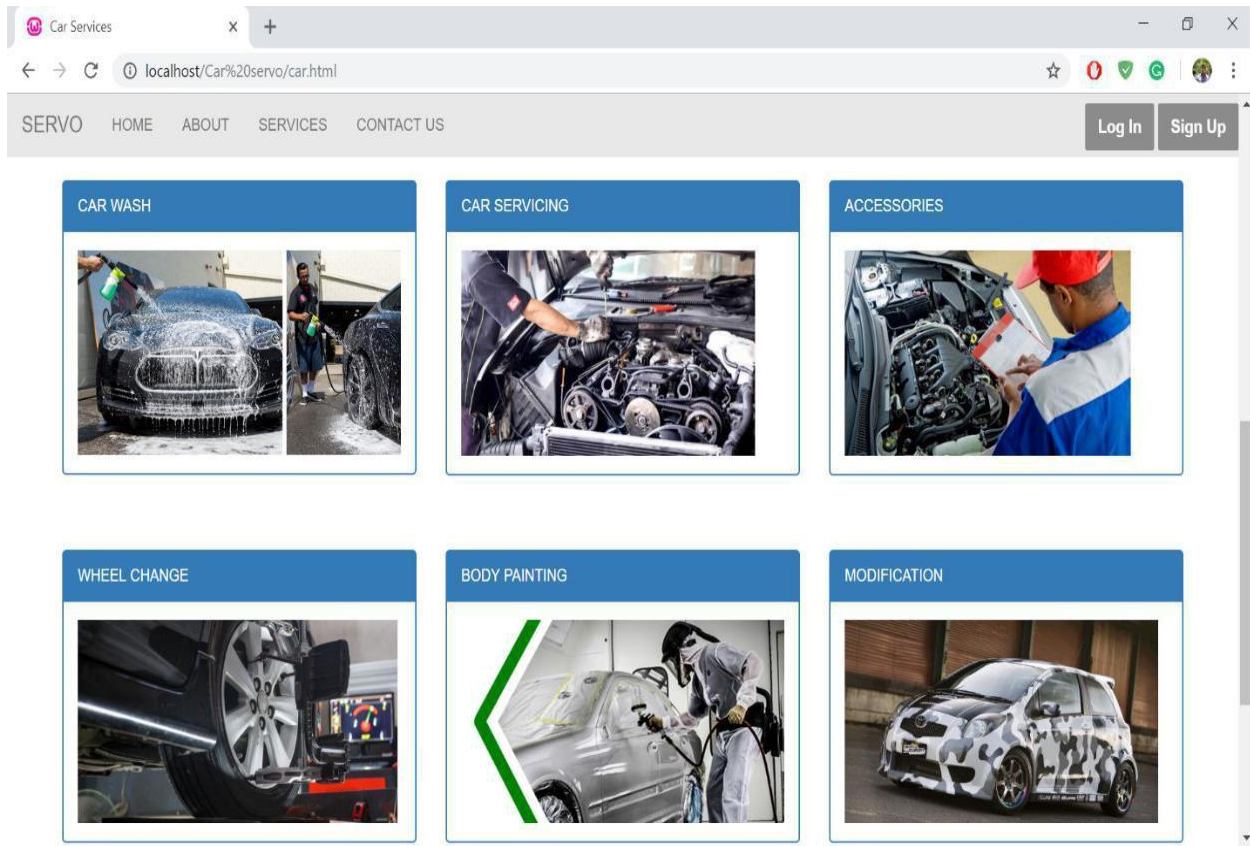
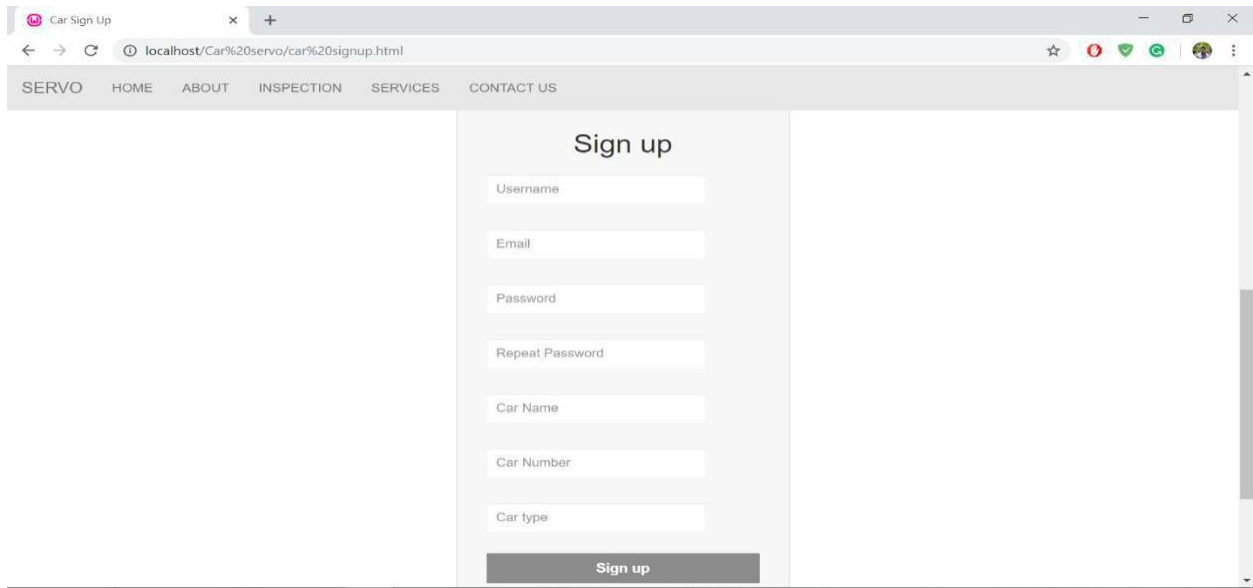


Fig 7: Home page of project

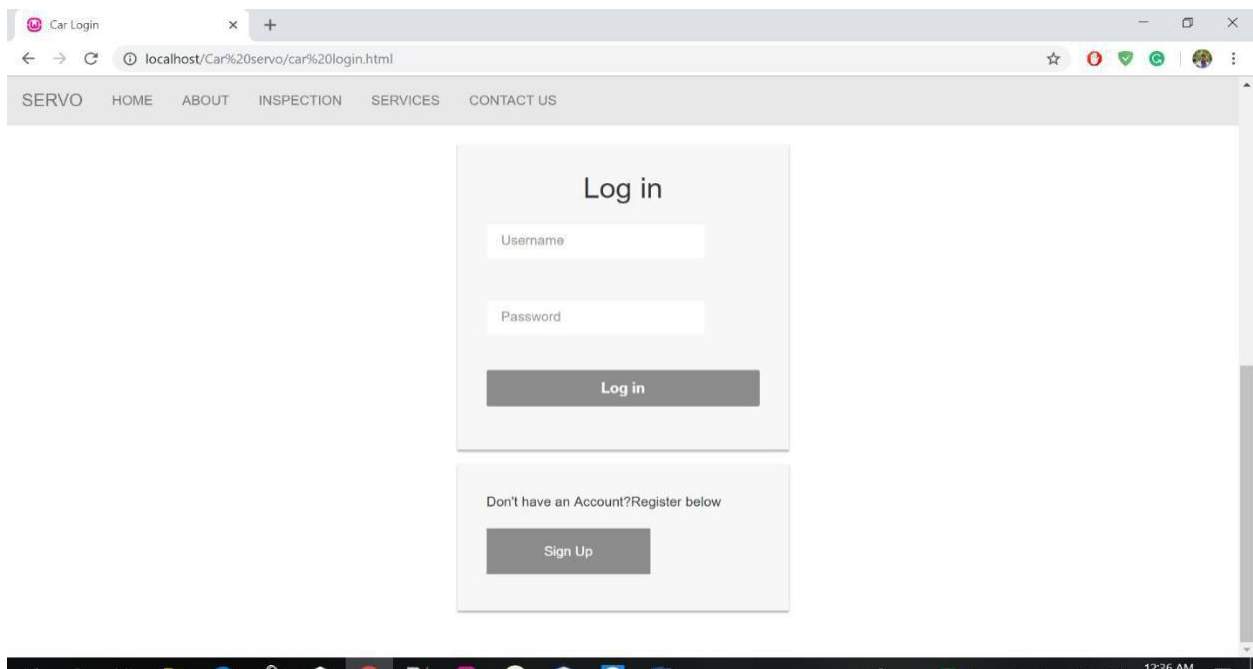
The figure 7 shows the home page of the project. We can see some car services and accessories sales here. In order to book any car service or if user needs to buy some spares then user needs to login with his account. If not, user has to register his details. Without registering his account, he cannot book the service or buy the car spares.



The screenshot shows a web browser window with the title 'Car Sign Up'. The address bar displays 'localhost/Car%20servo/car%20signup.html'. A navigation bar at the top contains links: SERVO, HOME, ABOUT, INSPECTION, SERVICES, and CONTACT US. The main content area features a 'Sign up' form with the following fields: Username, Email, Password, Repeat Password, Car Name, Car Number, and Car type. A 'Sign up' button is located at the bottom of the form.

Fig 8: Sign up page of the project

Figure 8 shows the registration page of the project. In this page new user can registered his/her account.



The screenshot shows a web browser window with the title 'Car Login'. The address bar displays 'localhost/Car%20servo/car%20login.html'. A navigation bar at the top contains links: SERVO, HOME, ABOUT, INSPECTION, SERVICES, and CONTACT US. The main content area features a 'Log in' form with the following fields: Username and Password. A 'Log in' button is located below the password field. Below the login form, there is a link 'Don't have an Account? Register below' and a 'Sign Up' button.

Fig 9: Log in page of the project

Figure 9 shows the login page for the user. If the user already has his account, he can login and get the service through this page.

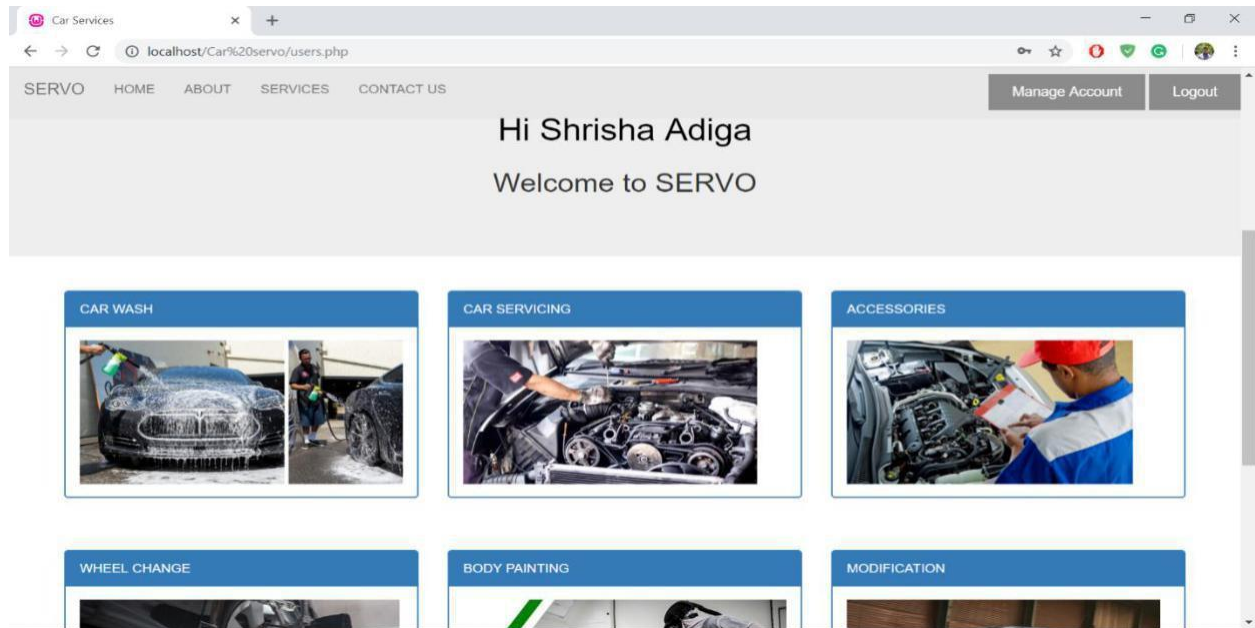


Fig 10: User Home page of the project.

Figure 10 shows the user home page where he can book a car service or book modifications or he can buy spare parts for his/her car.

A form titled 'Fill Out This Form' for booking a water wash service. It contains the following fields and options: 'Car Name' (text input), 'Car Number' (text input), 'Car Type' with radio buttons for 'HatchBack' (selected) and 'Sedan Class /SUV', 'Wash Type' with a dropdown menu showing 'Interior', 'Date of Wash' with a date input field (placeholder 'mm/dd/yyyy'), and 'Garrage Details' with a text input field (placeholder 'Bosch Service Station'). At the bottom is a large grey button labeled 'BOOK'.

Fig 11:Water wash booking form.

## Car-Servo

Here user can book his water wash on his convenient date at the nearest service station give in the website.

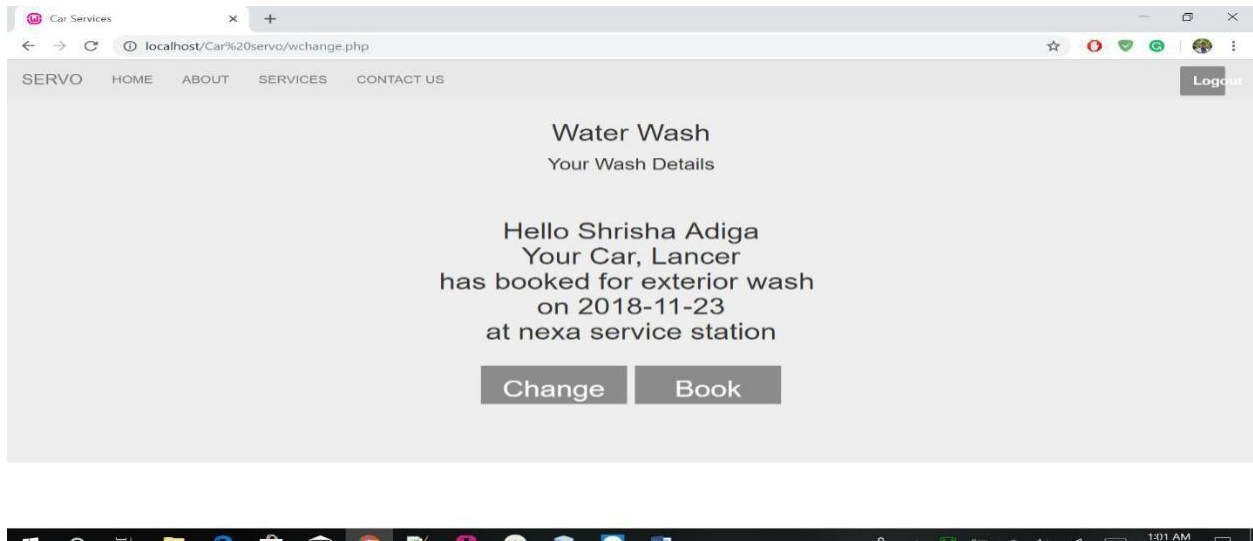


Fig 12: Confirmation of Water wash booking form

After filling out the water wash booking form, user needs to confirm whether any changes required for his water wash. If any changes he required he can just change the water wash type, date of wash and garage via change button. Otherwise he can confirm his booking. He can pay the amount after the service.

The screenshot shows a web browser window with the URL `localhost/Car%20servo/details.pdf`. The page has a navigation bar with links: SERVO, HOME, ABOUT, SERVICES, CONTACT US, and a Logout button. The main content area displays the following text:

SERVO SERVICES

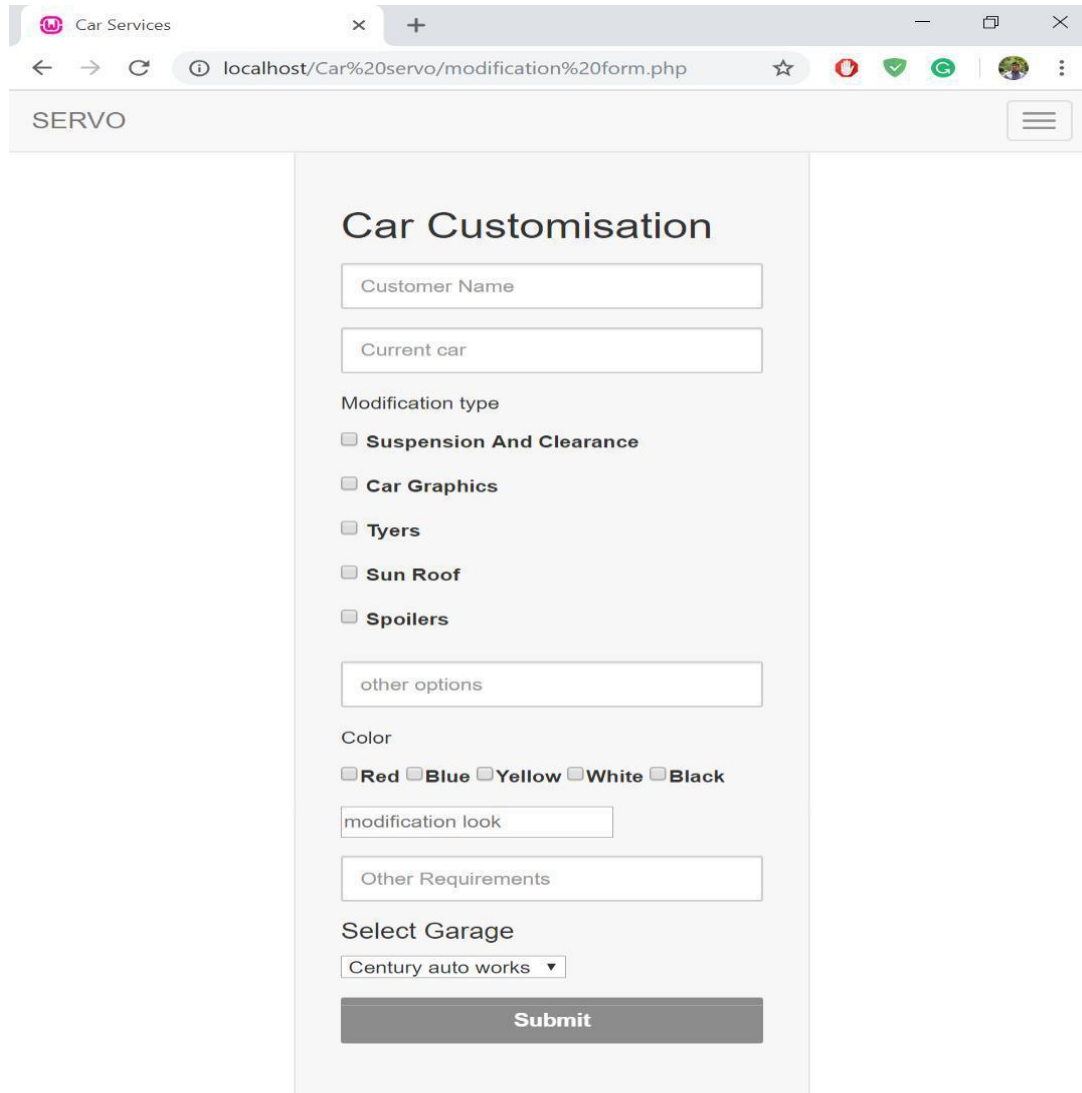
GARRAGE INFORMATION

NAME	PHONE NO	CAR BRANDS	WATER WASH COST (in Rs)	SERVICE COST (In Rs)
BOSCH SERVICE CENTER	08182-223345	SUZUKI, TATA, HYUNDAI, BMW, FORD.	300-500	1000-40000
NEXA SERVICE CENTER	08182-223355	SUZUKI, TATA, HYUNDAI, BMW, FORD, BENZ, VOLVO.	300-600	2500-60000
NISSAN SERVICE CENTER	08182-223390	SUZUKI, TATA, HYUNDAI, BMW, FORD, BENZ, VOLVO, CHEVORLET.	300-700	2000-70000
SUPREME SERVICE CENTER	08182-224456	SUZUKI, TATA, FORD, BENZ, VOLVO, FIAT, CHEVORLET.	300-600	2000-50000
DC SERVICE CENTER	08182-223344	SUZUKI, TATA, HYUNDAI, HONDA, FORD.	300-500	1000-45000

Fig 13: Garage Information.

## Car-Servo

In figure 4.7 we had given the complete details about the garage. Here we had mentioned the address, phone number, water wash cost and car servicing cost. Car servicing includes both major and minor services hence we have not included this servicing part in our project.



The screenshot shows a web browser window with the title 'Car Services' and the URL 'localhost/Car%20servo/modification%20form.php'. The page has a header with the word 'SERVO' and a hamburger menu icon. The main content area is titled 'Car Customisation' and contains the following form elements:

- Customer Name
- Current car
- Modification type
  - ☐ Suspension And Clearance
  - ☐ Car Graphics
  - ☐ Tyers
  - ☐ Sun Roof
  - ☐ Spoilers
- other options
- Color
  - ☐ Red ☐ Blue ☐ Yellow ☐ White ☐ Black
- modification look
- Other Requirements
- Select Garage
  - ▼
- 

Fig 14:Modification Booking form.

Figure 4.8 shows the modification booking form for the user. Here user can customize his car by including details in this form and he can take his car to garage and he can customize his car. Here also payment is done after the customization.

## Car-Servo

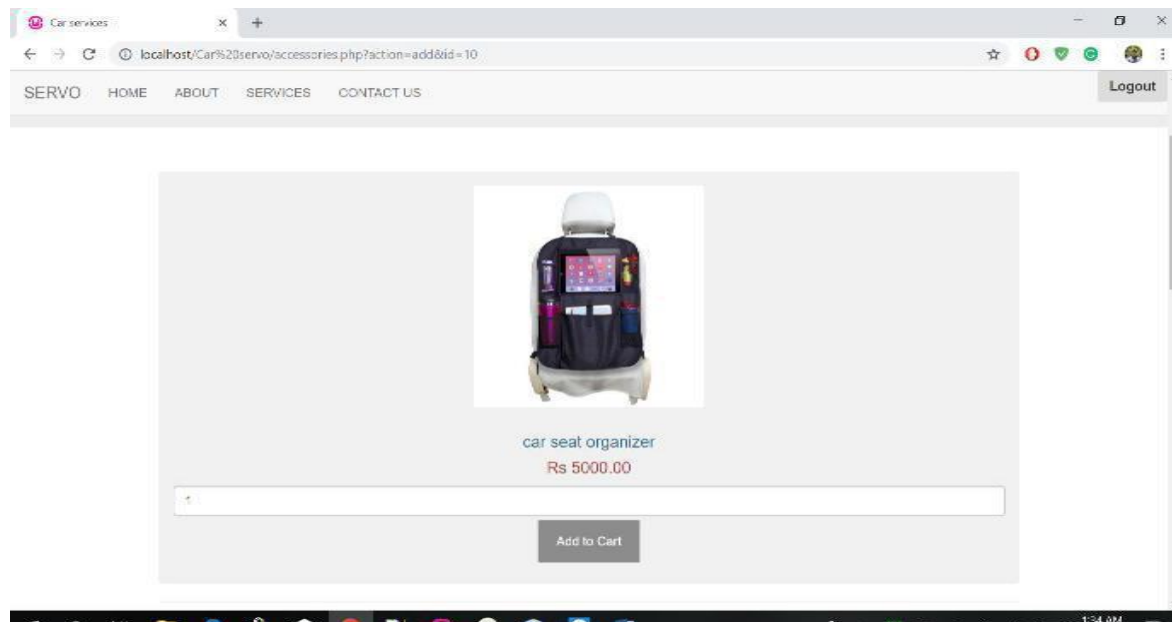


Fig 15: Spares Hub.

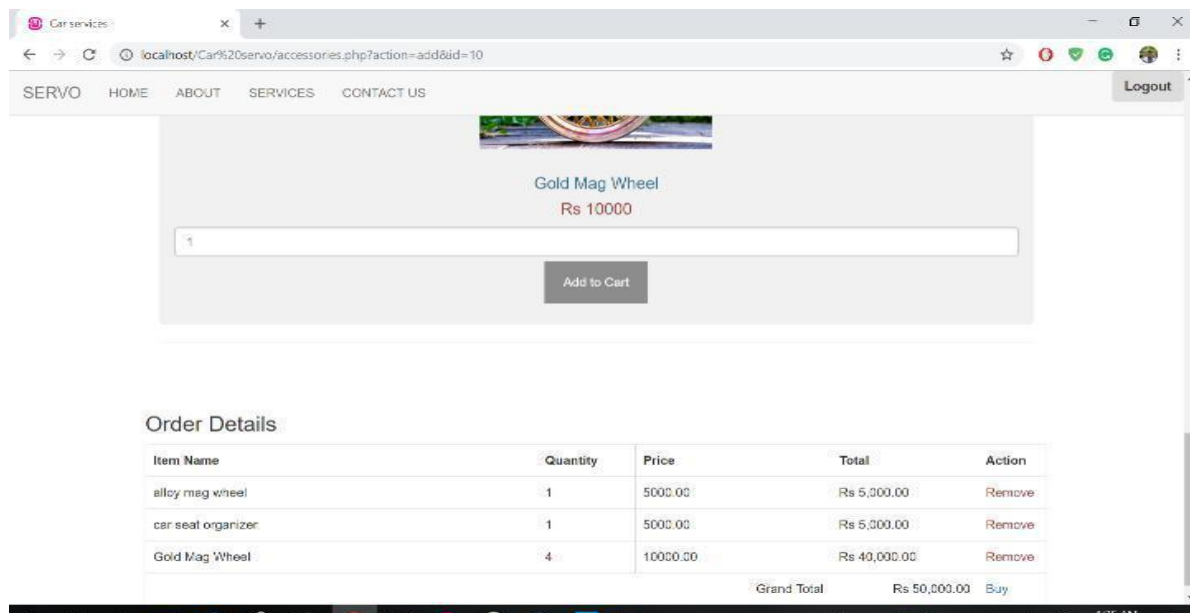


Fig 16: Spares Hub.

Figure 16 shows the car spares. Here user can buy any products. Before buying, user just adds the product to the cart. If user doesn't want the product, he can just remove it from the cart. Then

## Car-Servo

the product will be delivered to his given address. The payment is made either by debit card or through Paytm.

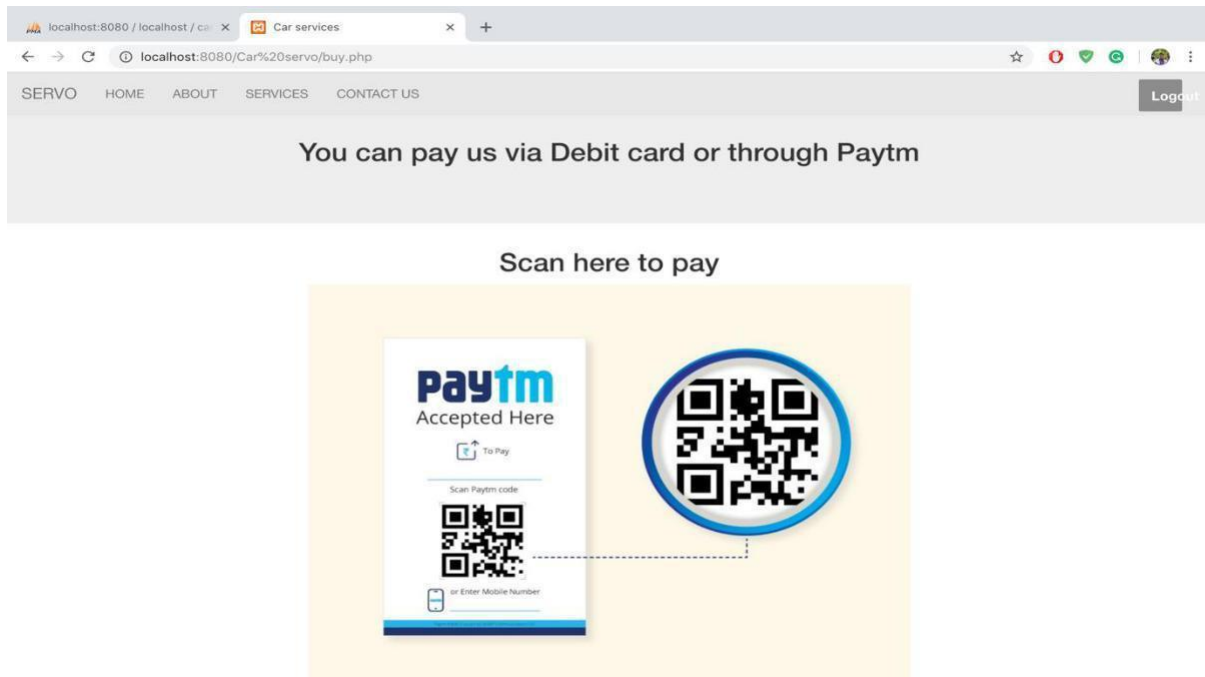


Fig 17: Paytm payment page.

Figure 17 shows the Paytm payment page. Here user can buy the accessories and user can pay us via Paytm app.

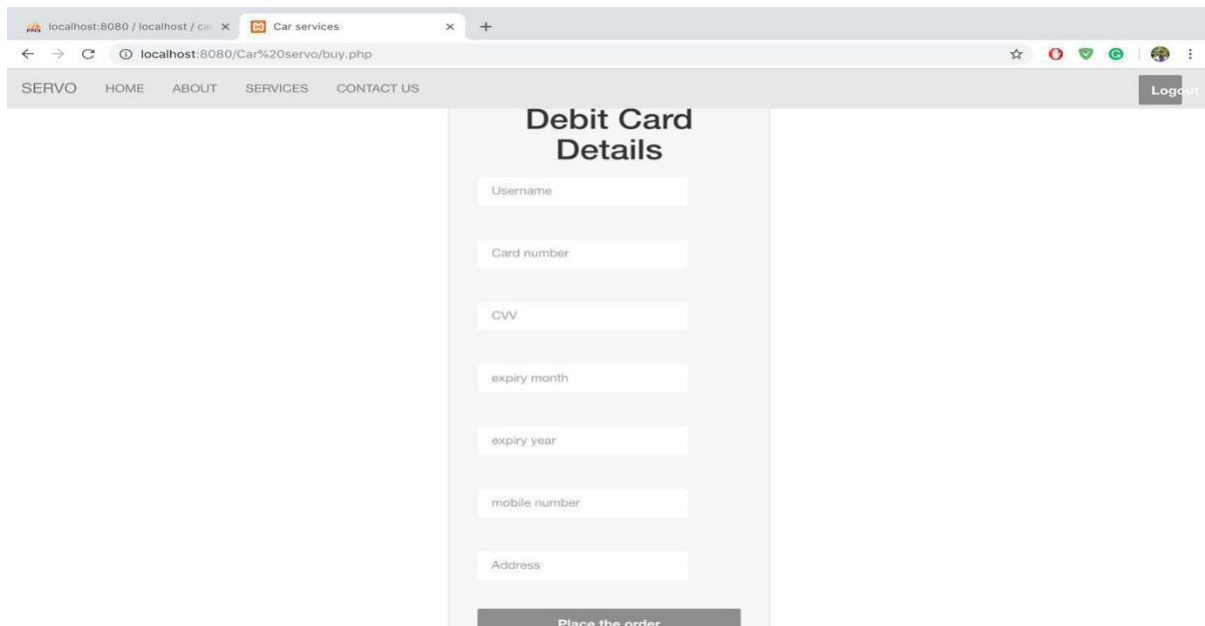




Fig 18: Debit card details page.

Figure 18 shows the debit car details page. User can also pay us via debit card by giving the correct details to us.

The screenshot shows a web browser window with the URL `localhost:8080/Car%20servo/feedback.php`. The page has a navigation bar with links: SERVO, HOME, SERVICES, CONTACT US, and FEEDBACK. On the right side of the navigation bar are buttons for 'Manage Account' and 'Logout'. The main heading of the page is 'FEEDBACK FORM'. Below this, there is a feedback form consisting of a text input field with the placeholder 'Enter feedback here...', a 'Post Feedback' button, and a list of feedback options: 1. Superb, 2. Good, and 3. Better. To the right of the feedback options is a table showing the dates of previous feedback submissions.

Feedback	Posted on
1. Superb	2018-11-14
2. Good	2018-11-15
3. Better	2018-11-15

Fig 19: User's Feedback Page.

Figure 19 shows that the user can give the feedback for us and he can also view his/her previously given feedbacks.

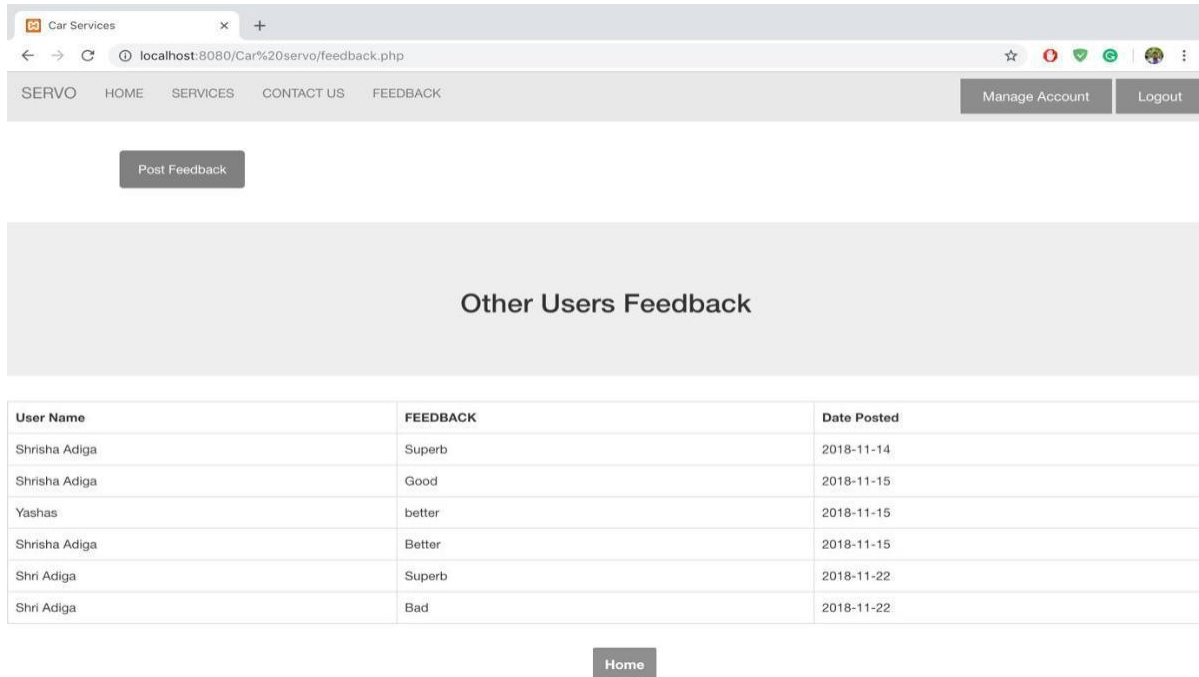


Fig 20: Other User's Feedback Page.

Figure 20 shows that the user can also view other user's feedback.

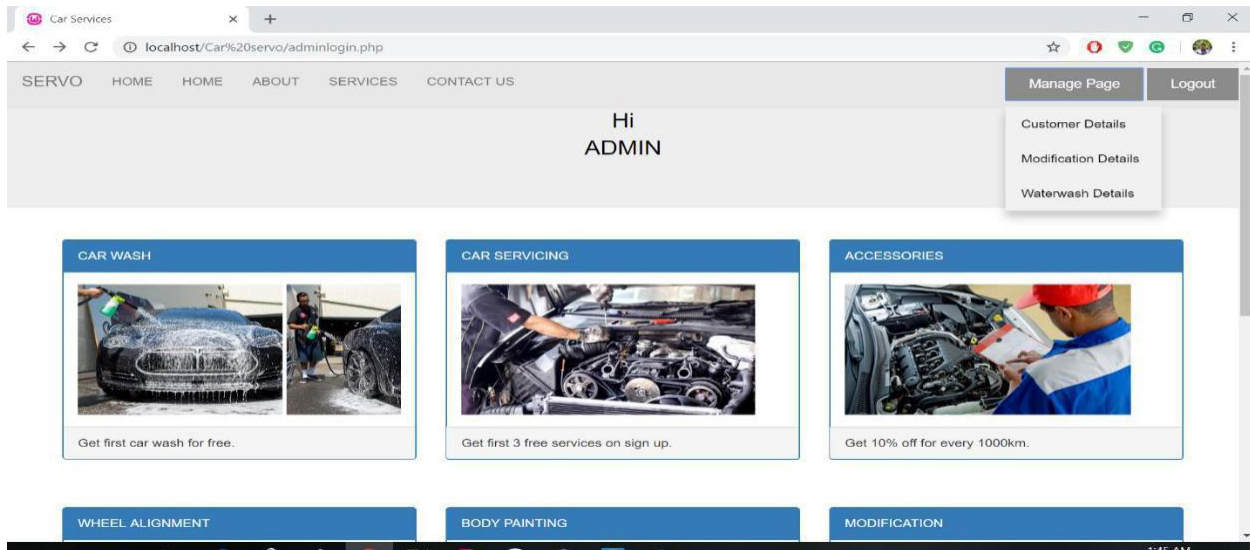
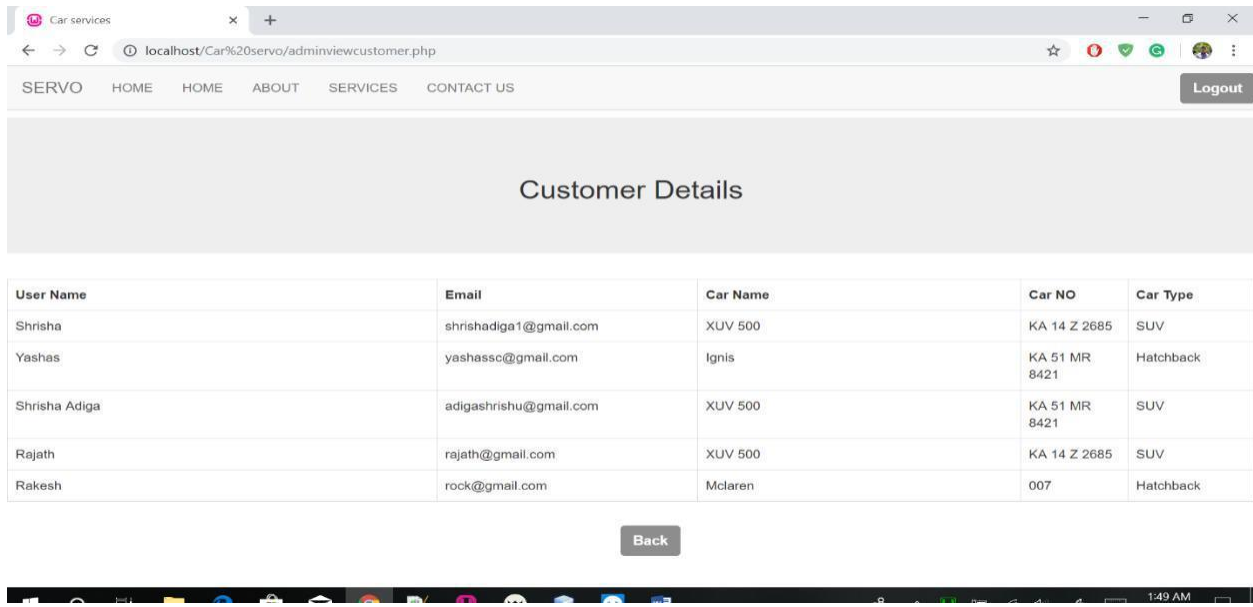


Fig 21: Admin Home page.

Figure 21 shows the admin home page. Here admin can view his customers. Admin can also view water wash and modification details. Admin can also add the product into user page or he can also delete the product from the user page if is out of stock.

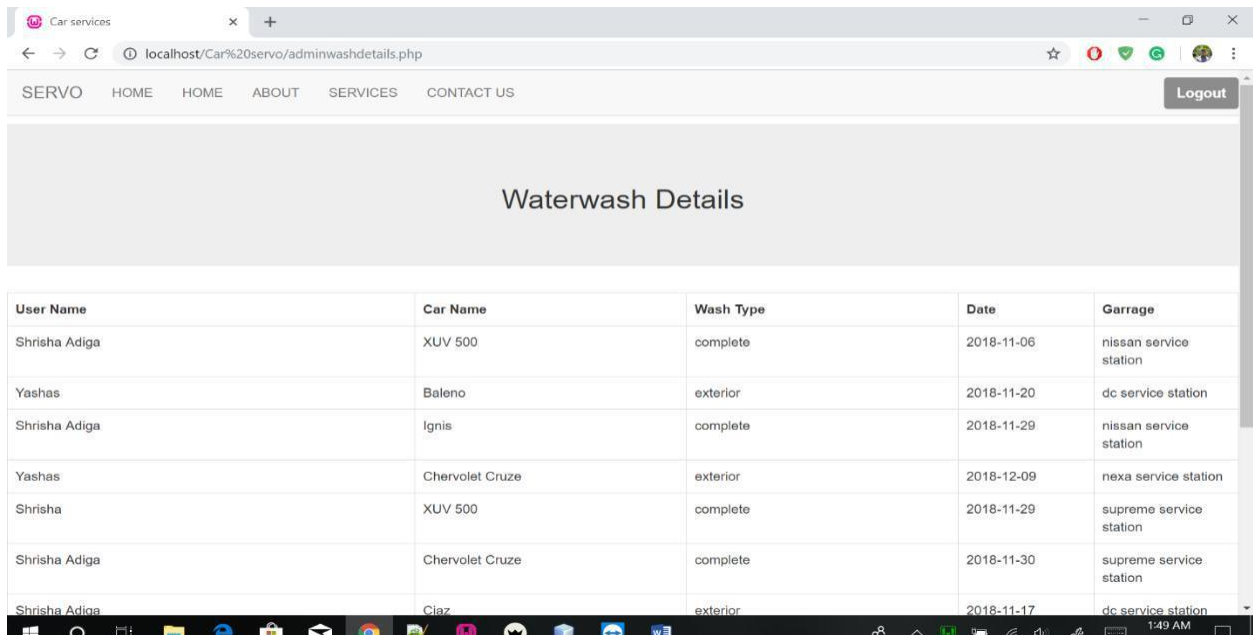


User Name	Email	Car Name	Car NO	Car Type
Shrisha	shrishadiga1@gmail.com	XUV 500	KA 14 Z 2685	SUV
Yashas	yashassc@gmail.com	Ignis	KA 51 MR 8421	Hatchback
Shrishha Adiga	adigashrishu@gmail.com	XUV 500	KA 51 MR 8421	SUV
Rajath	rajath@gmail.com	XUV 500	KA 14 Z 2685	SUV
Rakesh	rock@gmail.com	Mclaren	007	Hatchback

Back

Fig 22: Customer's details page.

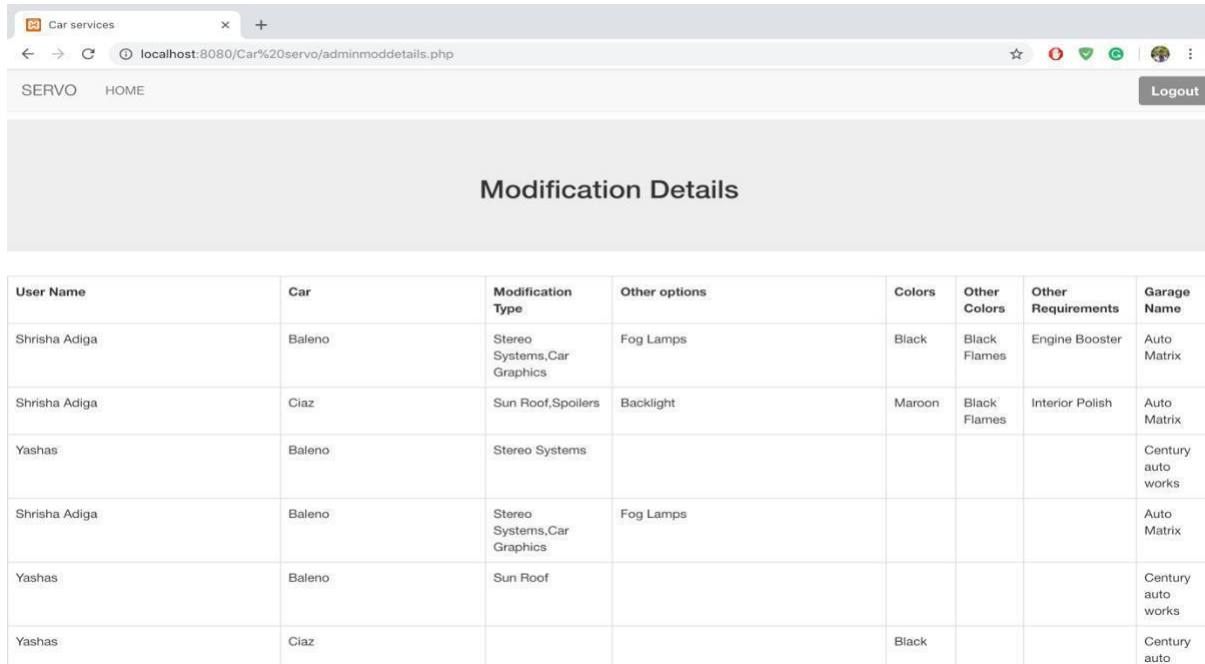
Figure 22 shows that admin can view his customers.



User Name	Car Name	Wash Type	Date	Garraige
Shrishha Adiga	XUV 500	complete	2018-11-06	nissan service station
Yashas	Baleno	exterior	2018-11-20	dc service station
Shrishha Adiga	Ignis	complete	2018-11-29	nissan service station
Yashas	Chervolet Cruze	exterior	2018-12-09	nexa service station
Shrishha	XUV 500	complete	2018-11-29	supreme service station
Shrishha Adiga	Chervolet Cruze	complete	2018-11-30	supreme service station
Shrishha Adiga	Ciaz	exterior	2018-11-17	dc service station

Fig 23: Customer's Water wash details page.

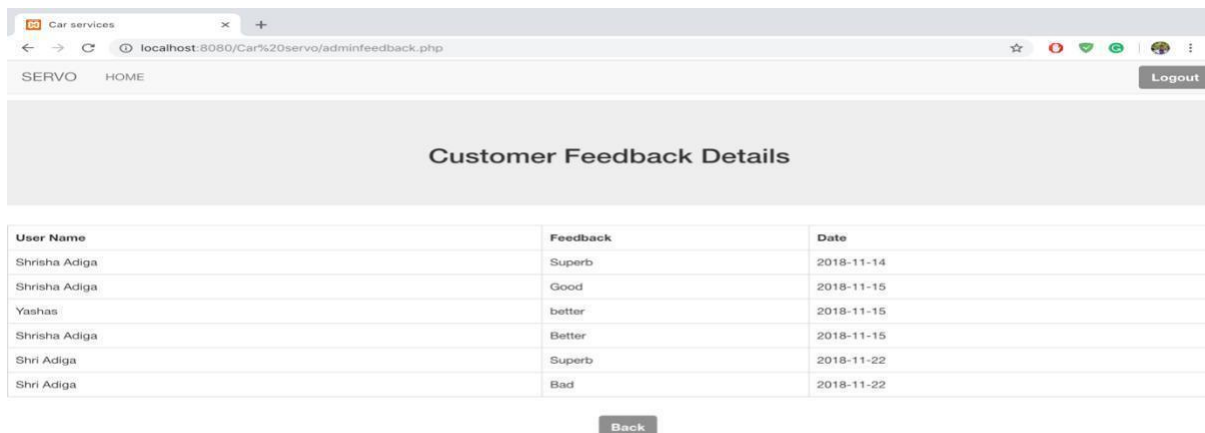
Fig 23 shows that admin can also view the water wash details, who has booked the water wash on which date and on which service station.



User Name	Car	Modification Type	Other options	Colors	Other Colors	Other Requirements	Garage Name
Shrisha Adiga	Baleno	Stereo Systems, Car Graphics	Fog Lamps	Black	Black Flames	Engine Booster	Auto Matrix
Shrisha Adiga	Ciaz	Sun Roof, Spoilers	Backlight	Maroon	Black Flames	Interior Polish	Auto Matrix
Yashas	Baleno	Stereo Systems					Century auto works
Shrisha Adiga	Baleno	Stereo Systems, Car Graphics	Fog Lamps				Auto Matrix
Yashas	Baleno	Sun Roof					Century auto works
Yashas	Ciaz			Black			Century auto

Fig 24: Customer's Modification details page.

Figure 24 shows that admin can also view the modification details, who has booked the modification on which date and on which service station.



User Name	Feedback	Date
Shrisha Adiga	Superb	2018-11-14
Shrisha Adiga	Good	2018-11-15
Yashas	better	2018-11-15
Shrisha Adiga	Better	2018-11-15
Shri Adiga	Superb	2018-11-22
Shri Adiga	Bad	2018-11-22

[Back](#)

Fig 25 Customer's Feedback page.

Figure 25 shows that admin can also view the customer's feedback. So that admin can improve his quality of service.

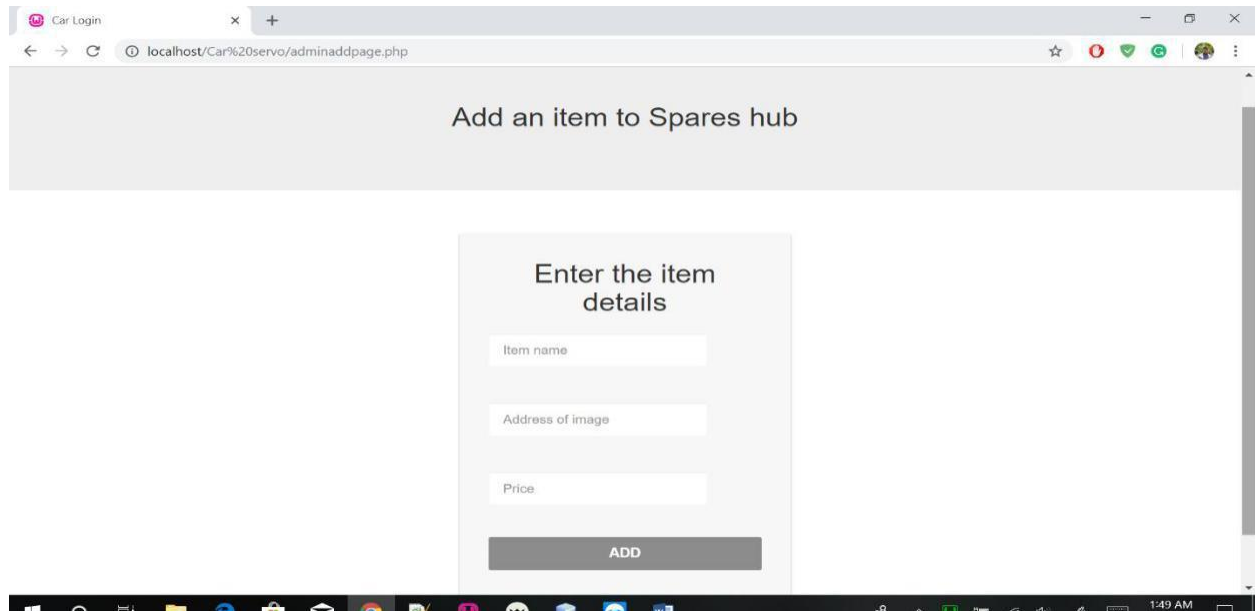


Fig 26: Admin add item page.

Figure 26 shows that admin can add new items to the user's page. Admin can add the item by just giving the item name and admin should choose an image then finally he needs to fix the prize. After clicking add button the item will be added to user's page.

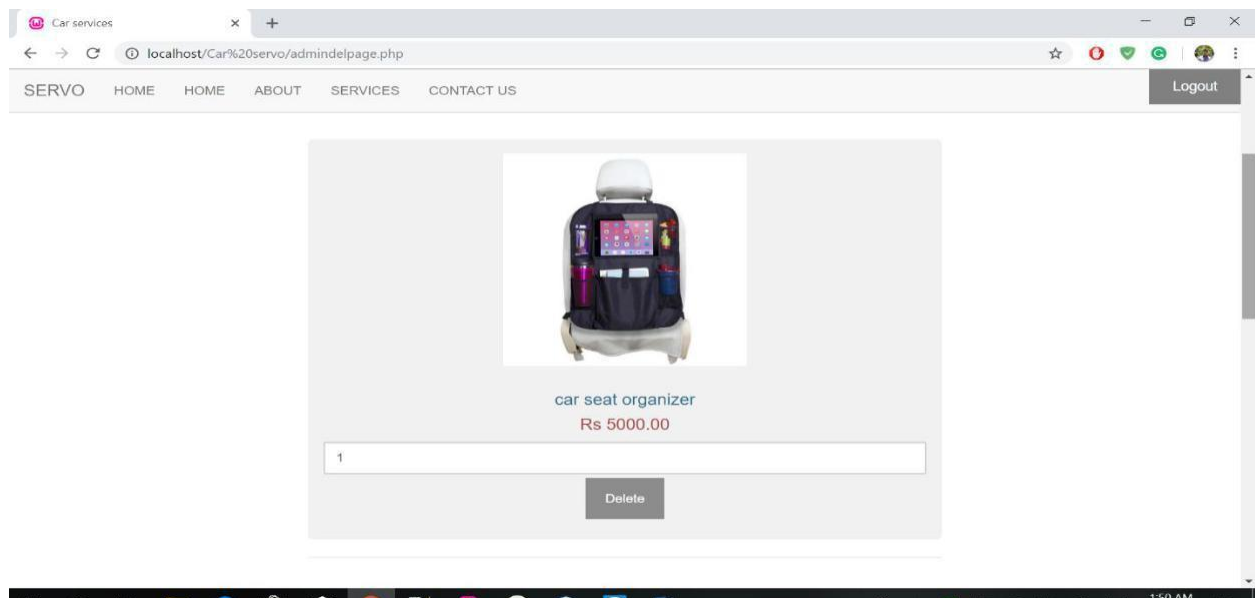


Fig 27 Admin delete item page.

## Car-Servo

Figure 27 shows that admin can delete the products from the user page if the product goes out of stock. By just clicking on delete button the item is going to be deleted from the user's page.

## **CHAPTER 7**

### **CONCLUSION**

Online Car Service and Spares Sales Management System helps us in centralizing the data used for managing the tasks performed in an online shopping and online car service booking. The theoretical process involved in database design has been practically implemented. The project provides user friendly interface for the users to interact with the database. All database operations including insertion, deletion, updating and Retrievals are supported along with support for trigger and stored procedure.

## **CHAPTER 8**

### **REFERENCES**

[1] Database systems Models, Languages, Design and Application Programming, RameezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.

[2] [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

[3] [http://www.w3schools.com/css/css\\_background.asp](http://www.w3schools.com/css/css_background.asp)

[4] [http://www.w3schools.com/sql/sql\\_update.asp](http://www.w3schools.com/sql/sql_update.asp)

[5] [http://www.w3schools.com/php/php\\_forms.asp](http://www.w3schools.com/php/php_forms.asp)



# Role Played --

**Project Planning** - Entire Team

**Project Monitoring** - Ayushman Awasthi

- Aman Mishra

**Project Testing** - Yatin Mishra

- Sakshi Srivastava

**Web Page Designing** - **Car.php** (Main Module) Completed by collaboration of entire Team.

**Final Report** - (Ayushman Awasthi, Yatin Mishra, Sakshi Srivastava)

**Mid Term Report + PPT** (Sakshi Srivastava, Ayushman Awasthi)

**Synopsis** – Aman Mishra

<b>Aman Mishra</b> <ul style="list-style-type: none"> <li>- Car login</li> <li>- Car Logout</li> <li>- Car Wash</li> <li>- C name check</li> <li>- C name fail</li> <li>- C name log</li> <li>- C password check</li> <li>- Car Fail</li> <li>- Check log</li> <li>- Check details</li> <li>- Contact</li> <li>- Details</li> <li>- Place</li> </ul>	<b>Sakshi Srivastava</b> <ul style="list-style-type: none"> <li>- Car Signup</li> <li>- Feedback</li> <li>- Feedback1</li> <li>- Login check</li> <li>- Login Fail</li> <li>- Mg</li> <li>- Modch1</li> <li>- Mod change</li> <li>- Modification</li> <li>- Modification form</li> <li>- Modification</li> <li>- Mod Submit</li> </ul>
<b>Ayushman Awasthi</b> <ul style="list-style-type: none"> <li>- Admin login</li> <li>- Admin Accessories</li> <li>- Admin Add Item</li> <li>- Admin Add Page</li> <li>- Admin Delete Item</li> <li>- Admin Delete Page</li> <li>- Admin Feedback</li> <li>- Admin Mod Details</li> <li>- Admin view Customer</li> <li>- Admin Wash Details</li> <li>- Buy</li> <li>- Payment</li> </ul>	<b>Yatin Mishra</b> <ul style="list-style-type: none"> <li>- Signup</li> <li>- Signup1 Failure</li> <li>- Signup Access</li> <li>- User password</li> <li>- User pl</li> <li>- Users</li> <li>- Wash check</li> <li>- Wash change</li> <li>- Wash change1</li> <li>- Place1</li> <li>- Place mod</li> <li>- Remove</li> </ul>

Dept. of CEA, GLAU, Mathura