**Title of the Project :**

**Online Vehicle Showroom Management System**

**Introduction and objective of the project:**

1. **Project Description**

This is a live project which was developed for a local Vehicle Agent. It is useful to the Agent to maintain the Customers details, Sales Details, Company items and services and also Vehicle Details. If any customer buy a vehicle in loan that persons vehicle will recognizes with the help of vehicle no.

Vehicle Recognition System, which will cover the following modules

The agent will enter the Vehicle No It will help the user to search the vehicle details and the required details of the customer .The agent enters the customer details and maintain company details.

Vehicle showroom management system is ideal for dealers or resellers of any size. The vehicle showroom control panel can be installed on any Operating System. The product features include: easy stock inventory updates, full user management, photograph uploads, image optimization, marketing reports, and easy installation. This software does not require special templates, frames, coding, or server-side technology.

**Abstract of the project:**

Regarding the registration module, it contains the information about newly joined User details like name of the user, password etc. Vehicle Details module contains the details like Name of the Vehicle, send from, send to etc. Billing and Payment Detail will contain the details like Actual Amount to be paid, emi. Searching is having the details of the customer. Any agent will login and search the details of the customer it will show all details like customer Details, vehicle name, payment type etc.

This “Vehicle Showroom Management System” project mainly contains the vehicle information and loan recovering details. The vehicle is purchased that vehicle amount will be cleared or not. It will clear then its ok. Otherwise the agent will search the details of the customer and recover the loan.

**According these three Phases, there are three modules in this system. They are:**

1. Admin module:
   * Registration:
     + - 1. Employee Registration
         2. User Registration,
         3. Vehicles Registration,
   * Booking Detail:
     + - 1. Online booking
         2. Manual booking
   * Booking Operation:
     + - 1. Booking Conformation
         2. Booking Cancellation
   * Search
2. User module:

* Vehicles Catalogs
* Booking
  + - * 1. Online booking
        2. Manual booking
* Search Vehicle

1. Reports

* Daily Report
* Weekly Report
* Monthly Report
* Yearly Report

The proposed system facilitates the customers to fill up their details, and to give a brief description of a vehicle they want to book. This new system is very helpful for customers who want to hire their vehicles through this site.

Functional requirements: The systems have the following inputs:

* First the customer has to make a reservation and later on in the process has to do registration.
* Second if the customer had already registered himself then he can continue booking in his own account by giving his customer id or mail id.
* Thirdly, the customer can amend details or update his details.
* The main outputs are whether the booking is confirmed or not and regular information’s services to the users of the site.

Advantages: The project can be easily used in the process of decision making. Different types of reports can be generated which help the management to take correct decision and reduce the time delay which automatically increases the company’s work standards as well as the economical state of the company.

**Problem Statement:**

A vehicle showroom is a vehicle that can be used temporarily for a fee during a specified period. Getting a rental vehicle helps people get around despite the fact they do not have access to their own personal vehicle or don't own a vehicle at all. The individual who needs a vehicle must contact a rental vehicle company and contract out for a vehicle. This system increases customer retention and simplify vehicle and staff management.

**Aims & Objectives**

* To produce a web-based system that allow customer to register and reserve vehicle online and for the company to effectively manage their vehicle showroom business.
* To ease customer’s task whenever they need to rent a vehicle.

**Scope for the project Online Vehicle showroom System**

This project traverses a lot of areas ranging from business concept to computing field, and required to perform several researches to be able to achieve the project objectives.

**The area covers include:**

* Vehicle showroom industry: This includes study on how the vehicle showroom business is being done, process involved and opportunity that exist for improvement.
* PHP Technology used for the development of the application.
* General customers as well as the company’s staff will be able to use the system effectively.
* Web-platform means that the system will be available for access 24/7 except when there is a temporary server issue which is expected to be minimal.

**How Vehicle showroom Services Work**

A vehicle showroom is a vehicle that can be used temporarily for a period of time with a fee. Renting a vehicle assists people to get around even when they do not have access to their own personal vehicle or don't own a vehicle at all. The individual who want to rent a vehicle must first contact the vehicle showroom company for the desire vehicle. This can be done online. At this point, this person has to supply some information such as; dates of rental, and type of vehicle. After these details are worked out, the individual renting the vehicle must present a valid Identification Vehicled.

Most companies throughout the industry make a profit based of the type of vehicles that are rented. The rental vehicles are categorized into economy, compact, compact premium, premium and luxury. And customers are free to choose any vehicle of their choice based on their purse and availability of such vehicle at the time of reservation.

**Benefits of Online Vehicle showroom Services**

* This online vehicle showroom solution is fully functional and flexible.
* It is very easy to use.
* This online vehicle showroom system helps in back office administration by streamlining and standardizing the procedures.
* It saves a lot of time, money and labour.
* Eco-friendly: The monitoring of the vehicle activity and the overall business becomes easy and includes the least of paper work.
* The software acts as an office that is open 24/7.
* It increases the efficiency of the management at offering quality services to the customers.
* It provides custom features development and support with the software.

**Functional Requirements**

Requirement analysis is a software engineering technique that is composed of the various tasks that determine the needs or conditions that are to be met for a new or altered product, taking into consideration the possible conflicting requirements of the various users.

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data should the system holds and the interfaces with the user. The functional requirements identified are:

1. Customer’s registration: The system should allow new users to register online and generate membership vehicled.
2. Online booking of vehicle: Customers should be able to use the system to make booking and.
3. Feedbacks to customers: It should provide means for customers to leave feedback.
4. Booking Registration : Registration of booking
5. Vehicle Management : Manage and add vehicles online

**Non-Functional Requirements**

It describes aspects of the system that are concerned with how the system provides the functional requirements. They are:

1. Security**:** The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the company can gain access to the company’s secured page on the system; and only users with valid password and username can login to view user’s page.
2. Performance and Response time:The system should have high performance rate when executing user’s input and should be able to provide feedback or response within a short time span usually 50 seconds for highly complicated task and 20 to 25 seconds for less complicated task.
3. Error handling:Error should be considerably minimized and an appropriate error message that guides the user to recover from an error should be provided. Validation of user’s input is highly essential. Also the standard time taken to recover from an error should be 15 to 20 seconds.
4. Availability:This system should always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that the business process is not severely affected.
5. Ease of use:Considered the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and required less training.

**Project Category:**

Relational Database Management System (RDBMS) : This is an RDBMS based project which is currently using MySQL for all the transaction statements. MySQL is an opensource RDBMS System.

**Brief Introduction about RDBSM:**

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMSs have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand and use. However, relational databases have been challenged by object databases, which were introduced in an attempt to address the object-relational impedance mismatch in relational database, and XML databases.

**System Purpose**

The company wants to find a solution to reduce its operating costs. The system being developed is a system to handle the business needs of renting out vehicles to customers, maintaining records and data on vehicle fleet, operating the customer portal website, and reporting the state of the system to the company. The system does not fulfill any other needs of the business.

**System Scope**

The functional scope of the system is represented in four different aspects of the system: Customer Service Module (CS), Web Portal Module (WP), Fleet Management Module (FM) and Reports and Analytics Module (RA).

* The scope of the Fleet Management module is keeping track of the rental vehicle fleet. This module’s purpose will be to contain the data on the vehicle fleet and information about the specific vehicles. The Fleet Management module will not handle the renting of the vehicles.
* The scope of the Customer Web Portal is to rent vehicles to customers online in the absence of an employee. The module will interface with the fleet management module but will not perform any of that module’s duties. The same goes for the Customer Service module.
* The scope of the Customer Service Module is to provide a clear and easy to use layout for employees and customers to follow along with as they work out a rental. As mentioned above the module will interface with the fleet management module but will not perform any of that module’s duties.
* The scope of the Reports and Analytics Module is to provide a predetermined spot that will contain all of the reports made from employees on vehicle showrooms. It will help the store managers and corporate office when they need to file reports and check on rentals.

**System Overview**

**System Context**

The system has been designed with four modules in mind. These modules are the customer service module, the web portal module, the fleet management module, and the reports and analytics module. These four modules will make up the structure of the system. The customer service module will be the part of the system that is supposed to provide the employee with everything they need to perform their duties. The web portal module will be the part of the system that handles the website where the customers will be able to go through an automated process to rent vehicles. The fleet management module will do as its name suggests and manage the rental fleet. The reports and analytics module will be the part of the system that will be used to generate reports from the remaining three modules for the purpose organizing the data on the status of the company. The system will be able to interact with outside systems to process payment on behalf of the company.

**System Functions**

The system will be able to rent out vehicles. The system will be able to accept payments. The system will keep track of the rental fleet. The system will generate reports for the employees.

**User Characteristics**

The users will be customers which can include traveling business people, out of town visitors, and local residents in need of a vehicle. Other users of the system will be the employees both at the store level and a headquarters. The employees will all be using windows desktops to conduct their business on the system but the system will need to accommodate the variety of devices that the customers will have. The customers will have mobile devices running several different operating software including android, iOS, and blackberry. The full website also must accommodate these different operating systems.

**Definitions of Vehicle showroom System:**

* Customer Service Module: CS
* Customer Web Portal: WP
* Fleet Management: FM
* Reports and Analytics: RA

**Actors**

* Employee: Emp
* Manager: Man
* Customer: Cus

**Requirements External Interfaces**

**Hardware Interfaces**

* R1: The system will have 3 terminals per store, with each one having a touch screen monitor, keyboard, credit vehicled scanner and a cash register.
* R2: The system will have a 45 minute battery backup at each terminal.
* R3: The system will have a 2 hour battery backup at the server at headquarters.

**System Interfaces**

* R4: The system will interact with the banking network for the purpose of processing payments.
* R5: The systems data needs will be supported by a connection to the headquarters server.

**User Interfaces**

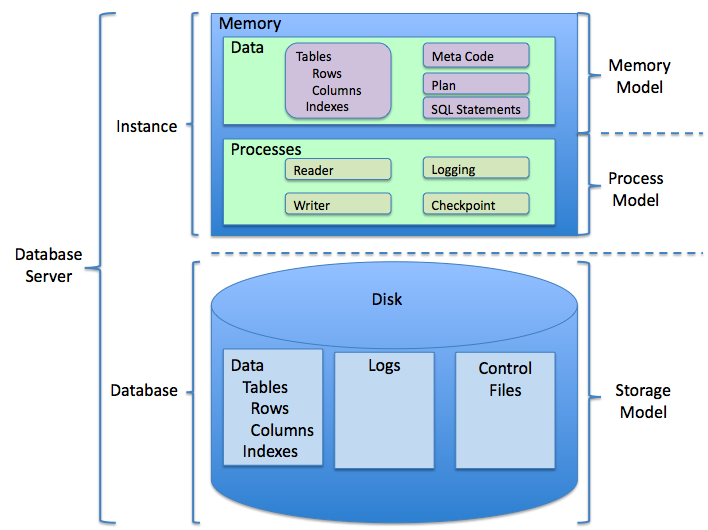
* R6: The System will consist of Four (4) modules; Customer Service, Customer Portal Website, Fleet Management, Reports & Analytics.

**Functional Requirements**

* R7: The System will contain a Customer Service module that will allow Store and Corporate employees to provide information to customers
* R8: The System will contain a Customer Service module that will allow Store and Corporate employees’ access to the system for the purpose of Creating "Rental Agreements"
* R9: The System will contain a Customer Service module that will allow Store and Corporate employees to collect payments
* R10: The System will contain a Customer Portal Website that will provide information to the public and customers about the company and operations of the company.
* R10.1: This information will include Locations, Hours of operation, contact information, available rental vehicles, rental packages, price estimate, and any other information deemed necessary by the company.
* R11: The customer portal website will provide the customer with the closest location to their zip code and up to the next five (5) closest locations in addition to hours of operations, contact info and directions.
* R12: The System will generate an inventory capacity report for the desired time and location.
* R13: The System will allow for new vehicles to be added to the inventory of the company.
* R14: The Fleet management module will allow the system to track vehicle maintenance. Maintenance will be tracked through mileage and or time as decided by the company.
* R15: The System will contain a Report and Analytics Module that will serve the purpose of generating reports on the operations of the company.
* R16: The System will contain a Report and Analytics Module that will be capable of generating reports for items such as sales, maintenance, vehicle history, operations, rentals, and insurance.
* R17: The System will contain a Report and Analytics Module that will be capable of generating reports for a desired time and location
* R18: The system will batch the daily operations to headquarters at the end of each business day.
* R19: The system will generate a backup of all data weekly for the purpose of data loss prevention at the headquarters.
* R20: The system will allow the customer to input the desired date, location and package options to determine vehicle availability. If vehicle is available then the system will move onto the next step. If vehicle is not available then the system will prompt the customer to pick another vehicle.
* R21: The system will display only vehicles that are currently available to rent to the customer during the rental process.
* R22: The system will create a customer account if the customer is not in the system.
* R23: The customer account will require name, address, date of birth, driver’s License, and credit vehicled number.
* R24: The system will require a credit vehicled be on file but will allow customer to use different forms of payment at time of rental.
* R25: The system will provide a report of the vehicles that are scheduled to be rented out and returned for the specified day.
* R26: The system will allow for a block reservation of more than one (1) vehicle at a time.
* R27: at time of pickup the system will allow the employee to bring up the specified reservation and to print out a "rental agreement"
* R28: The system will require the employee to verify the information on the "Rental Agreement" and to then input into the system a notice that the agreement has been gone over. This will ensure that the employee has in fact gone over the agreement and everything is in place.
* R29: The system will ask if the employee has obtained the customer Signature on the "rental agreement". Customer signature is required on the rental agreement by company policy.
* R30: The system will allow for preliminary payment to be collected at time of pickup if company policies states that payment is to be collected at time of pickup.
* R31: The customer will liable for a "no show" charge if they fail to pick up the vehicle at the specified time. The system will require that the charge be process before the reservation can be closed out in the case of a "no show"
* R32: The customer will have twenty four (24) hours before pickup time to call and cancel the reservation before the system requires the "no show" charge
* R33: At time of return of rental vehicle the system will allow the employee to print out a "Vehicle inspection form". This form is to be filled out at the vehicle and then input back into the system.
* R34: The "vehicle inspection form" will require information including model, license plate number, vin number, and damage to the vehicle, level of fuel in the gas tank, mileage, and a section for any other issues.
* R35: The system will allow for a vehicle to be returned to a different location than the original rental location for the charge of a fee to be determined by the company.
* R36: The system will identify and additional charges that need to be charged for any damage or issue to the rental vehicle.
* R37: For company accounts the system will allow for the invoice (bill) to be sent to the company.
* R38: Company billing will be required to be set up in advance with the company to prevent fraud.
* R39: After the vehicle inspection is done then the system will print off a final invoice for the costumer to sign. This invoice will contain the total amount charged by the rental company including primary rental and additional charges. Physical copy will be saved as proof of transaction.
* R40: The system will provide employees with a login.
* R41: The system will provide employees with a logout.
* R42: The system will provide employees with the option to reset their login password.
* R43: The system will provide employees with the option to change their login password.
* R44: The system will allow employees with quick an efficient access to customer information.
* R45: The system will provide employees with a screen view of all customer pickups for a specified day.
* R46: The system will provide employees with a screen view of an existing reservation of a customer.
* R47: The system will provide customers to login on the web portal and view existing reservation in the system.
* R48: The system will allow employees to check vehicle availability from inventory for customers.
* R49: The system will allow employees to get specific vehicle information for customers.

**Performance Requirement**

The local system should be capable of running, at minimum, 1000 processes per minute (PPM). The Headquarter (HQ) system should be able to process and allocate information to designated terminal within a 10 second refresh rate. The system response time should be no longer than 1.0 seconds in order to keep up with the actions given by the User. As the company begins to acquire more customers and becomes more popular, the system must be able to scale in speed, size, and versatility in order to accommodate the rising needs/wants of the customer basis.

**RDBMS Architecture Diagram**

**Tools/Platform, Hardware and Software Requirement specifications:**

For setting this portal, it requires certain technical requirements to be met for the store to operate properly. First, a web server must be created to make the ecommerce store publicly available on the web. Domain names and hosting services can easily be purchased for an affordable price. When selecting a hosting service, you should check to see that these server requirements are provided and installed on their web servers:

**Software Details for running Portal**

* Web Server (preferably Apache)
* PHP (at least 5.2)
* MySQL
* Curl

**Tools/Platform, Hardware and Software Requirement specifications:**

For setting this ecommerce portal, it requires certain technical requirements to be met for the store to operate properly. First, a web server must be created to make the online vehicle showroom system publicly available on the web. Domain names and hosting services can easily be purchased for an affordable price. When selecting a hosting service, you should check to see that these server requirements are provided and installed on their web servers:

**Software Details for running Online Vehicle showroom System**

* Web Server (preferably Apache)
* PHP (at least 5.2)
* MySQL
* Curl

**Required PHP libraries / modules**

* Curl
* ZIP
* Zlib
* GD Library
* Mcrypt
* Mbstrings

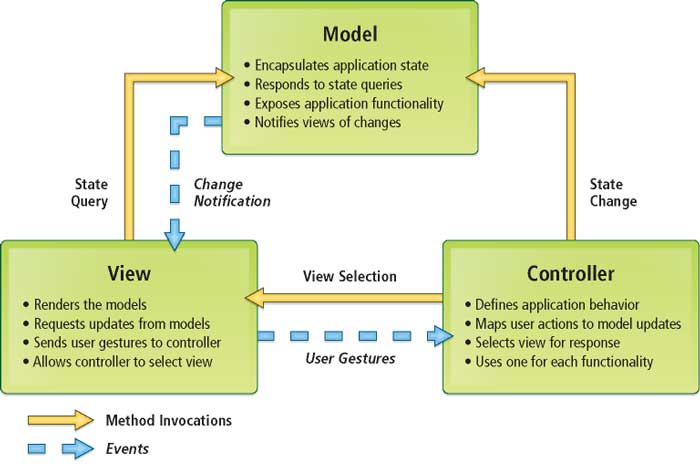
The above PHP extensions should be available by almost all hosting providers, during the install processa it will check you have them all enabled. You should contact your hosting provider if one is missing.

**Implementation methodology:**

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

* **Model** - The lowest level of the pattern which is responsible for maintaining data.
* **View** - This is responsible for displaying all or a portion of the data to the user.
* **Controller** - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.

**MVC (Model View Controller Flow) Diagram**

**Implementation of security mechanisms at various levels**

**Online Transaction Security System :** The online transaction security system will provide a safe and secure method for online shoppers to make credit vehicled purchases on the website. When a credit purchase is made using the website, the credit vehicled information will be encrypted using Secure Socket Layer (SSL) and transmitted to the bank for processing. This security system will also provide access control for website visitors, which will allow only registered users to make purchases. The registered customers account information will be stored in the customer information database mentioned earlier.

**Data Security System:** The data security system will allow data to be securely transmitted between the various components of the portal. This includes transmission of product, merchant and customer information from the content management system to the website, and also the transmission of data from the website to the content management system.

**Folder Security using .htaccess:** The .htaccess file in your sites directory is a configuration file you can use to override the settings on your web server. With the right commands, you can enable/disable extra functionality and features to protect your site from spammers, hackers and other threats.

**Prevent Directory Browsing:** Protecting directories from being listed is, at best, security by obscurity. That is, its hiding your stuff from view, preventing meddling visitors from browsing through your directories. Really, its the web equivalent of hiding your cash under your mattress.Still, its good practice to prevent directory browsing, along with implementing other measures to secure your site.

To disable browsing of your directories, add this to your .htaccess file:

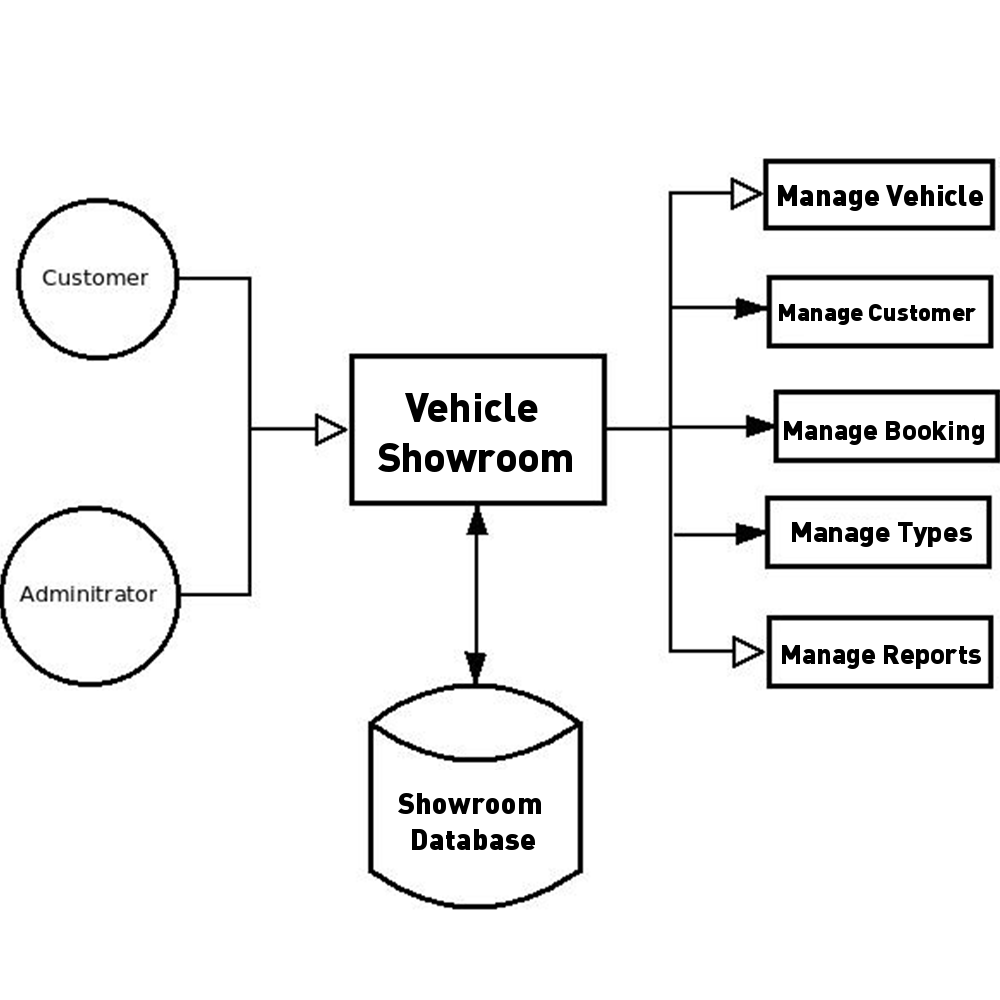
Options All –Indexes

**Restrict Access to Your Admin Area :**  A simple way to restrict access if your internet has a fixed IP address and you always access your site form the same location is by creating a new .htaccess file with the following snippet:

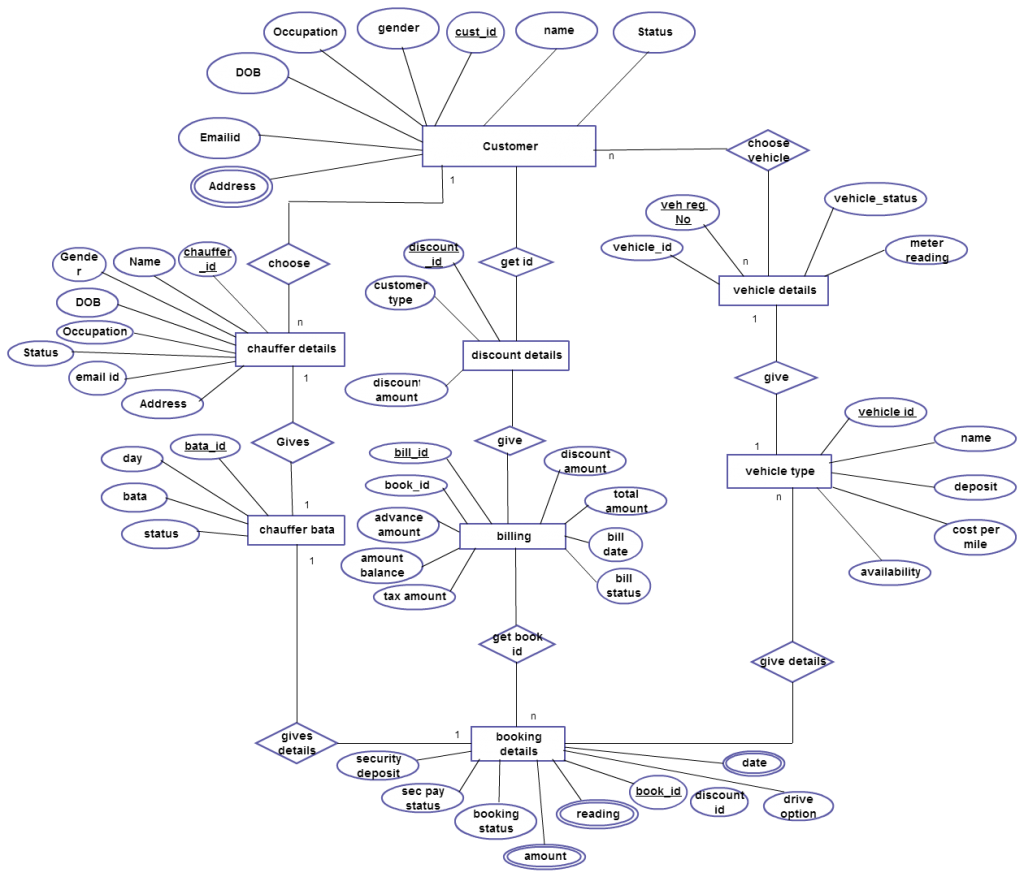
order deny,allow

allow from 192.168.5.1

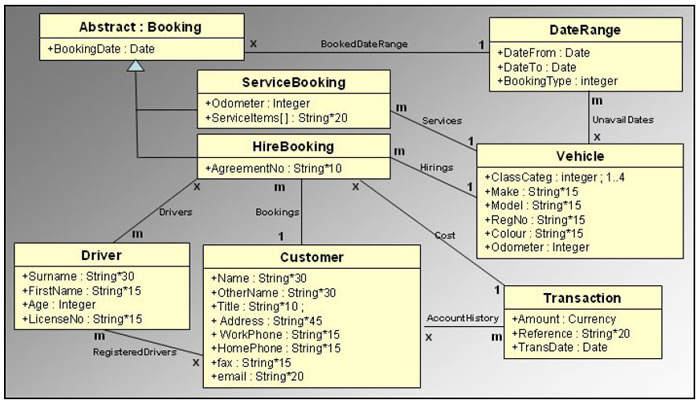
deny from all

****

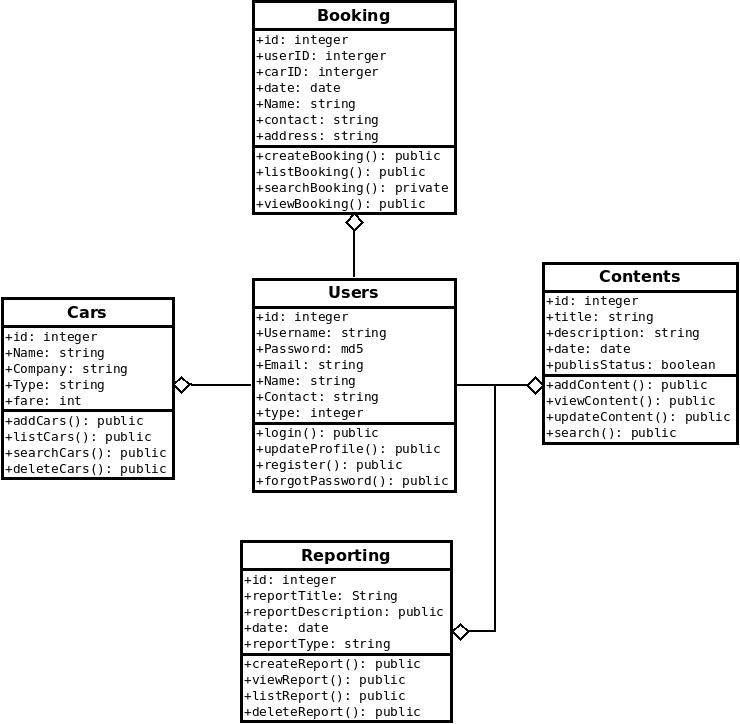
**Zero Level DFD**



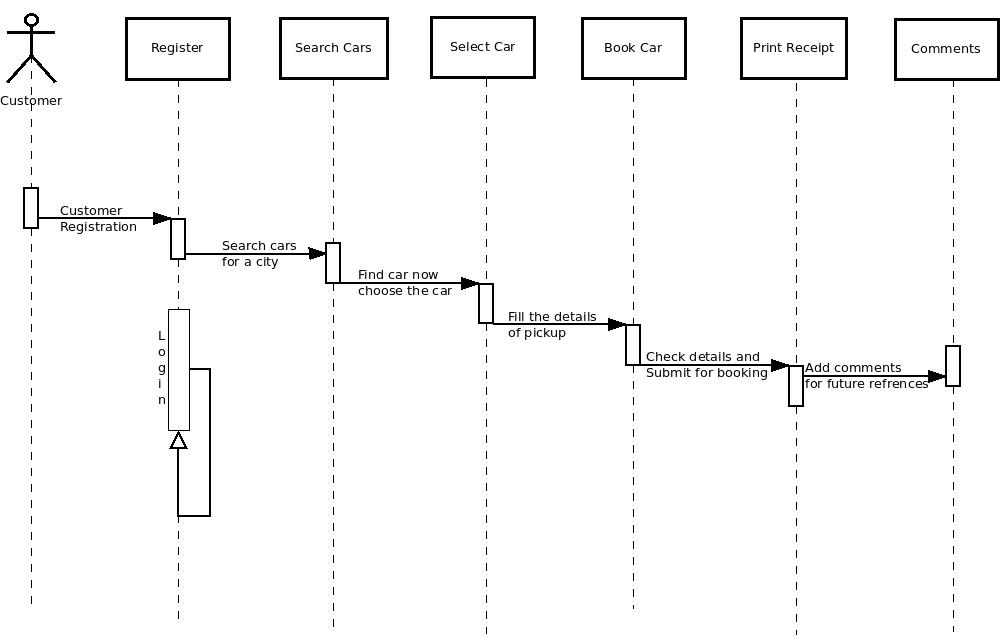
**ER-Diagram**



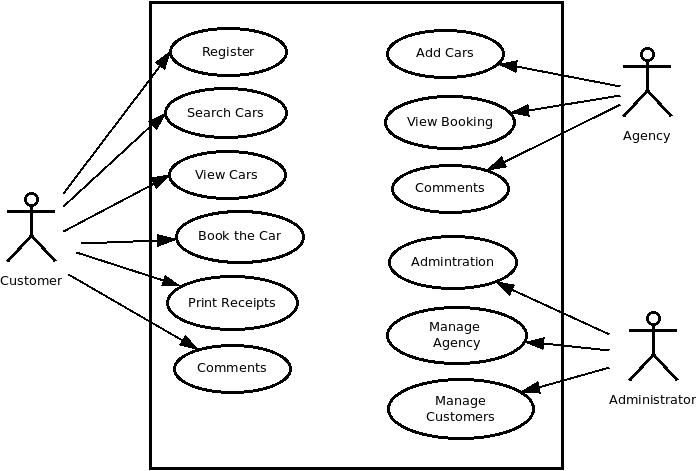
**Class Diagram**



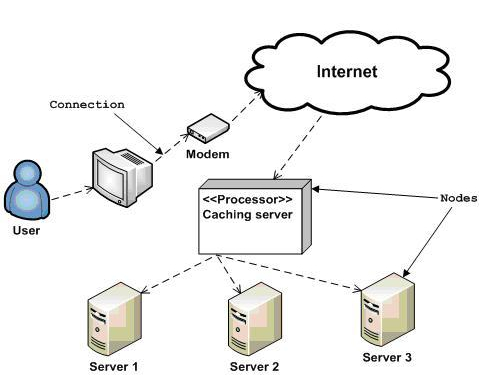
**Object Diagram**



**Sequence Diagram**



**Use Case Diagram**



**Deployment Diagram**

**SYSTEM ANALYSIS**

The analysis model is a concise, precise abstraction of what the desired system must do, and not how it will be done after the study of the existing system is completed. This basically includes system study and the requirement analysis. Interacting with the clients regarding their requirements and expectations from the system does requirement analysis.

**The steps, which are essential for system analysis, are:**

1. Research and define essential components.
2. Analyze current processes and identify gaps.
3. Interview users, Trainee, Trainers and other concerned personnel regarding essential components and current processes.
4. Write requirements document.
5. Define standards for standards, policies, and procedures.
6. Review draft requirements document with users, Trainee, Trainers and other concerned personnel.
7. Update and expand project plan.

**IDENTIFICATION OF NEED**

Online Library Management Systems described above can lead to error free, secure reliable and fast management system for library .

It can assist the staff to concentrate on their related(for library) activities rather to concentrate on the records and reports of issuing books, registration and suppliers. This will help organization in better utilization of human resources.

In this phase, the user identifies the need for a new or improved system. In large organizations this identification may be part of a systems planning process.

**PRELIMINARY INVESTIGATION**

**Benefit to Organization**

The organization will obviously be able to gain benefits such as savings in operating cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill. The other benefits are improved service and faster and better access to up-to-date information.

**The Initial Cost**

The initial cost of setting up the system will include the cost of hardware (server/clients, network adapter and related hardware), software (server OS, add-on software, utilities) & labor (setup & maintenance). The same has to bear by the organization.

**Running Cost**

Besides, the initial cost the long term cost will include the running cost for the system including the AMC, book charges, cost for human resources, cost for update/renewal of various related software.

**Need for Training**

The users along with the administrator need to be trained at the time of implementation of the system for smooth running of the system. The client will provide the training site.

**Depending upon this definition,** we analyzed the present Library Management System, during the analysis process; we had to go through the entire manual system that was being followed in the Organization for past few years.

We talked to the management people who were managing a the financial issues of the center, staff who were keeping the records in lots of registers and the reporting manager regarding their existing system, their requirements and their expectations from the new proposed system. Then, we did the system study of the entire system based on their requirements and the additional features they wanted to incorporate in this system.

The manual system was very complex to be managed and searching or updating the information was also very difficult.

There was a lengthy procedure for entering the details of books, members, visitors, and then making their timings and Bills.

The other problem with the manual system was of report generation. The reports to be generated on a daily/weekly/monthly basis required lot of paper work and calculations etc. Thus, Reporting Manager had to manually calculate all the consultation fees of every player at the end of every day/week/month. Even a small calculation mistake was leading to reanalysis and recreation of whole work.

**FEASIBILITY STUDY**

At this stage, the analyst estimates the urgency of the project and estimates the development cost.

In feasibility analysis, we have to study the following:

1. **Technical Feasibility:**

Technical feasibility is concerned with the availability of hardware and software required for the development of the system, to see compatibility and maturity of the technology proposed to be used and to see the availability of the required technical manpower to develop the system.

After the study we came to conclusion that we proceed further with the tools and development environment chosen by us. This was important in our case as we were working on two various phases of the department that will need to be integrated in future to make an extended system.

1. **Operational Feasibility:**

Operational feasibility is all about problems that may arise during operations. There are two aspects related with this issue:

* + What is the probability that the solution developed may not be put to use or may not work?
  + What is the inclination of the management and end users towards the solution? Though, there is very least possibility of management being averse to the solution, there is a significant probability that the end users may not be interested in using the solution due to lack of training, insight etc.

1. **Economic Feasibility:**

It is the measure of cost effectiveness of the project. The economic feasibility is nothing but judging whether the possible benefit of solving the problems is worthwhile of not. At the feasibility study level, it is impossible to estimate the cost because member’s requirements and alternative solutions have not been identified at this stage. However, when the specific requirements and solutions have been identified, the analyst weighs the cost and benefits of all solutions, this is called “cost benefit analysis.

**PROJECT PLANNING**

Software project plan can be viewed as the following:

1. **Within the organization:** How the project is to be implemented? What are various constraints (time, cost, staff)? What is market strategy?
2. **With respect to the customer:** Weekly or timely meetings with the customer with presentation on status reports. Customers feedback is also taken and further modification and developments are done. Project milestones and deliverables are also presented to the customer.

For a successful software project, the following steps can be followed:

* + Select a project
    - Identifying project’s aims and objectives
    - Understanding requirements and specification
    - Methods of analysis, design and implementation
    - Testing techniques
    - Documentation
  + Project milestones and deliverables
  + Budget allocation
    - Exceeding limits within control
* Project Estimates
  + Cost
  + Time
  + Size of code
  + Duration
* Resource Allocation
* Hardware
* Software
* Previous relevant project information
* Digital Library
* Risk Management
  + Risk avoidance
  + Risk detection

**PROJECT SCHEDULING**

An elementary Gantt chart or Timeline chart for the development plan is given below. The plan explains the tasks versus the time (in weeks) they will take to complete.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | January | | | | February | | | | March | | | |
| Requirement Gathering |  | |  | |  | | | |  | | | |
| Analysis |  | |  | |  | | | |  | | | |
| Design |  | | | |  |  | | |  | | | |
| Coding |  | | | |  |  | | |  |  | | |
| Testing |  | | | |  | | | |  |  | |  |
| Implement |  | | | |  | | | |  | | |  |
|  | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 |

W*i‘*s are weeks of the months, for i =1, 2, 3, 4

**SOFTWARE REQUIREMENT SPECIFICATION**

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

**1.  Introduction**

* 1. **Purpose** – Themain purpose, to make this software is to over come of problems of data base handling, maintain registers of members, books, and information about worker which are works at Library etc.
  2. **Intended Audience –** This SRS is meant for basically my final semester project report of MCA. Besides, it is obviously targeted for the managers of mentioned center, the client, to get and overall description and advantage of the proposed system over the existing manual system.
  3. **Product Vision and Scope –** The proposed system is being developed keeping in mind the requirements/need of the client to automate its existing system for record keeping, report generation and management level information system. Keeping in mind the needs, the system has been developed as per guidelines laid by the client’s center. The system is capable enough to handle library records, book’s records, member’s records, visitors records, report generation, and storing other related information.

1. **Overall Description**

2.1 Product Perspectives:

The proposed system falls under RDBMS (Relational Data Base Management System) category. I have adopted PHP as front end for the software and MYSQL as back end.

PHP is at present one of the most popular development platform for web based system that is efficient for web programming.

MYSQL is at present the most reliable and secure RDBMS tool. MYSQL Server works to efficiently manage its resource, a database of information, among the multiple clients requesting and sending data in the network. MYSQL has many important features that make it not only an exceptional database management system but also an excellent database server choice for client/server database computing.

So the overall system will prove to reliable, secure and efficient for the organization.

**2.2 Product Functions**

**Functionalities:**

2.2.1 Any person can query for books availability according to specified conditions.

* + 1. A Person can book/cancel the book only after he logs in.
    2. 2.2.4 A user can sign up for a profile if he doesn’t have one already.
    3. On logging in, the user has options to

1. Book issued .

2. Edit Profile information

3. Cancellation of his booked issued.

4. View all current books booked by him.

5. Logout

* + 1. A person can get all information regarding a library if he keys in it.
* A person can get all information about a books if he keys in the book id.
* A person can get the availability of all books for the next 20 days.
* Official members are suppose to do that work which distributed by the administrator
* Administrator or assigned official members can add/modify/delete library information.
* Administrator or assigned official members can add/modify/delete seat type according their availability and generate report.
* Administrator or assigned official members can add/modify/delete library information.
* Administrator or assigned official members can define and manage charges information.
* Administrator or assigned official members can define scheme and modify time to time and generate report.
* Administrator can add/modify/delete official member and generate report.
* Administrator can add/modify/delete user information and generate report.

**2.3. Operating Environment –** The proposed software is to run on client/server model network.

A client/server can deliver the better performance than the file server system because a client application and database server work together to split processing load of applications (thus the term distributed processing). The server manages the database among the number of clients, while the client send, request, and analyze the data entry form with small specific data set, such as rows in a table not file as in the file server system. A database server is intelligent enough so that it lock and return only the rows a client request, which ensure concurrency, minimize the network traffic and increase the system performance.

* 1. **Assumptions and Dependencies –** While cost estimation of the proposed system it has been assumed that the cost hardware and for license of Operating System and back end will be met by client (the organization). Hence only the cost incurred for the proposed software is included therein.

The followings are identified as some of the potential risk factors or dependencies:

* 1. Non-availability of required resources.
  2. Power cuts.
  3. Slippage of schedule due to unpredictable holidays, etc.

**2.5 Life Cycle Model –** I am using SDLC model that begin at system level and progresses through analysis, design, coding, testing, implementation and maintenance.

1. **External Interface Requirements**
   1. **User Interfaces –** It has been required that every form’s interface should be user friendly and simple to use. Besides, there should be facility of accessing the system through keyboard along with the mouse i.e. keyboard shortcuts.
   2. **Software Interfaces –** It has been required that there could be a necessity of using the stored data for some kind of report that is not supported by proposed system at present. So the proposed system is required to export its data as text file so that some other application software can import the data.

**4. System Future Requirement** – Other than descriptions provided above, the following features were required by the client:

1. The system should be secured enough to rely upon.
2. Users should not be allowed to delete/modify any records.
3. Users should not be allowed to take financial reports.
4. Every users report should keep the tracks of user inputting the record.
5. System should provide facility of exporting its data in text format.
6. System should be able to integrate with its Phase II developments.

5. **Other Nonfunctional Requirements**

* 1. **Performance Requirements –** As it is going to be used by all the concerned employees within the organization, the system should have a good performance in terms of speed and accuracy. The proposed system should be accurate and fast enough to handle huge data. It should provide fast communication between server and clients.
  2. **Safety Requirements –** As the system is going to handle records for a long run eliminating the manual system, it is supposed to ensure the retaining of data avoiding or eliminating any probable cause for data loss.
  3. **Security Requirements –** The software should not allow unauthorized access to any module of the system. Besides, it should maintain the privileges granted to users at various user levels.
  4. **Software Quality Attributes –** The prioritization of the software quality attributes are assumed as under:

1. Accurate and hence reliable.
2. Secured.
3. Fast speed.
4. Compatibility.
5. Portability.

**System Design**

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the customer’s requirements into a logically working system. Normally, design is performed in the following in the following two steps:

1. **Primary Design Phase:** In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.
2. **Secondary Design Phase :** In the secondary phase the detailed design of every block is performed.

**The general tasks involved in the design process are the following:**

* 1. Design various blocks for overall system processes.
  2. Design smaller, compact and workable modules in each block.
  3. Design various database structures.
  4. Specify details of programs to achieve desired functionality.
  5. Design the form of inputs, and outputs of the system.
  6. Perform documentation of the design.
  7. System reviews.

**User Interface Design**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

**The following steps are various guidelines for User Interface Design:**

1. The system user should always be aware of what to do next.
2. The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
3. Message, instructions or information should be displayed long enough to allow the system user to read them.
4. Use display attributes sparingly.
5. Default values for fields and answers to be entered by the user should be specified.
6. A user should not be allowed to proceed without correcting an error.
7. The system user should never get an operating system message or fatal error.

**TESTING**

Testing is the process in which the system is run on manually created input so that the system is correctly working as desired or not.

During systems testing, the system is used experimentally to ensure that the software does not fail. In other words, we can say that it will run according to its specifications and in the way users expect. Special test data are input for processing, and the results examined.

A limited number of users may be allowed to use the system so that analyst can see whether they try to use it in unforeseen ways. It is desirable to discover any surprises before the organization implements the system and depends on it.

Testing of a system is generally done in two phases – One is **Unit Testing** which is done for each module independently on its completion and the other one is **System Testing** which is done at the end of a project.

**VALIDATION CRITERIA**

The validation criteria in this project are as follows...

In Portal Systemalso, the user inputs are validated before storing them, and then further for displaying etc. The main validations that are done in Portal **System** are as follows –

All the screens have a similar look and feel. They all have the almost same color combinations in its background. This provides a better user interface to the users.

1. The primary key values cannot be duplicated.
2. All the entries in any combo box have been sorted in alphabetical order. This helps a user while selecting a value from the combo box.

**IMPORTANCE OF TESTING**

During systems testing, the system is used experimentally to ensure that the software does not fail. In other words, we can say that it will run according to its specifications and in the way users expect. Special test data are input for processing, and the results examined.

The importance of system testing is that the system is expected to run according to member’s requirement before delivering it to the customer.

The System is tested on the basis of specification so that it does not fail on user site.

**Implementation and Testing**

**Implementation**

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

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

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

**Programming and Testing** - This activity encompasses actual development, writing, and testing of program units or modules.

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

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

**Installation Phase** - In this phase the new Computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

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

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

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

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

**THE STEPS IN THE SOFTWARE TESTING**

**The steps involved during Unit testing are as follows:**

1. Preparation of the test cases.
2. Preparation of the possible test data with all the validation checks.
3. Complete code review of the module.
4. Actual testing done manually.
5. Modifications done for the errors found during testing.
6. Prepared the test result scripts.

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

1. Functionality of the entire module/forms.
2. Validations for user input.
3. Checking of the Coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source files.

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

**The steps involved during System testing are as follows:**

* Integration of all the modules/forms in the system.
* Preparation of the test cases.
* Preparation of the possible test data with all the validation checks.
* Actual testing done manually.
* Recording of all the reproduced errors.
* Modifications done for the errors found during testing.
* Prepared the test result scripts after rectification of the errors.

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

1. Functionality of the entire system as a whole.
2. User Interface of the system.
3. Testing the dependent modules together with all the possible test data scripts.
4. Verification and Validation testing.
5. Testing the reports with all its functionality.

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

Thus, we reached the final phase of the project delivery.

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

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

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

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

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

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

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

**COST ESTIMATION OF THE PROJECT**

Software cost comprises a small percentage of overall computer-based system cost. There are a number of factors, which are considered, that can affect the ultimate cost of the software such as - human, technical, Hardware and Software availability etc.

The main point that was considered during the cost estimation of **project** was its sizing. In spite of complete software sizing, function point and approximate lines of code were also used to "size" each element of the Software and their costing.

The cost estimation done by me for **Project** also depend upon the baseline metrics collected from past projects and these were used in conjunction with estimation variables to develop cost and effort projections.

We have basically estimated this project mainly on two bases -

**1) Effort Estimation -** This refers to the total man-hours required for the development of the project. It even includes the time required for doing documentation and user manual.

**2) Hardware Required Estimation -** This includes the cost of the PCs and the hardware cost required for development of this project.

**GANTT CHART**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | January | | | | February | | | | March | | | |
| Requirement Gathering |  | |  | |  | | | |  | | | |
| Analysis |  | |  | |  | | | |  | | | |
| Design |  | | | |  |  | | |  | | | |
| Coding |  | | | |  |  | | |  |  | | |
| Testing |  | | | |  | | | |  |  | |  |
| Implement |  | | | |  | | | |  | | |  |
|  | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 |

W*i‘*s are weeks of the months, for i =1, 2, 3, 4

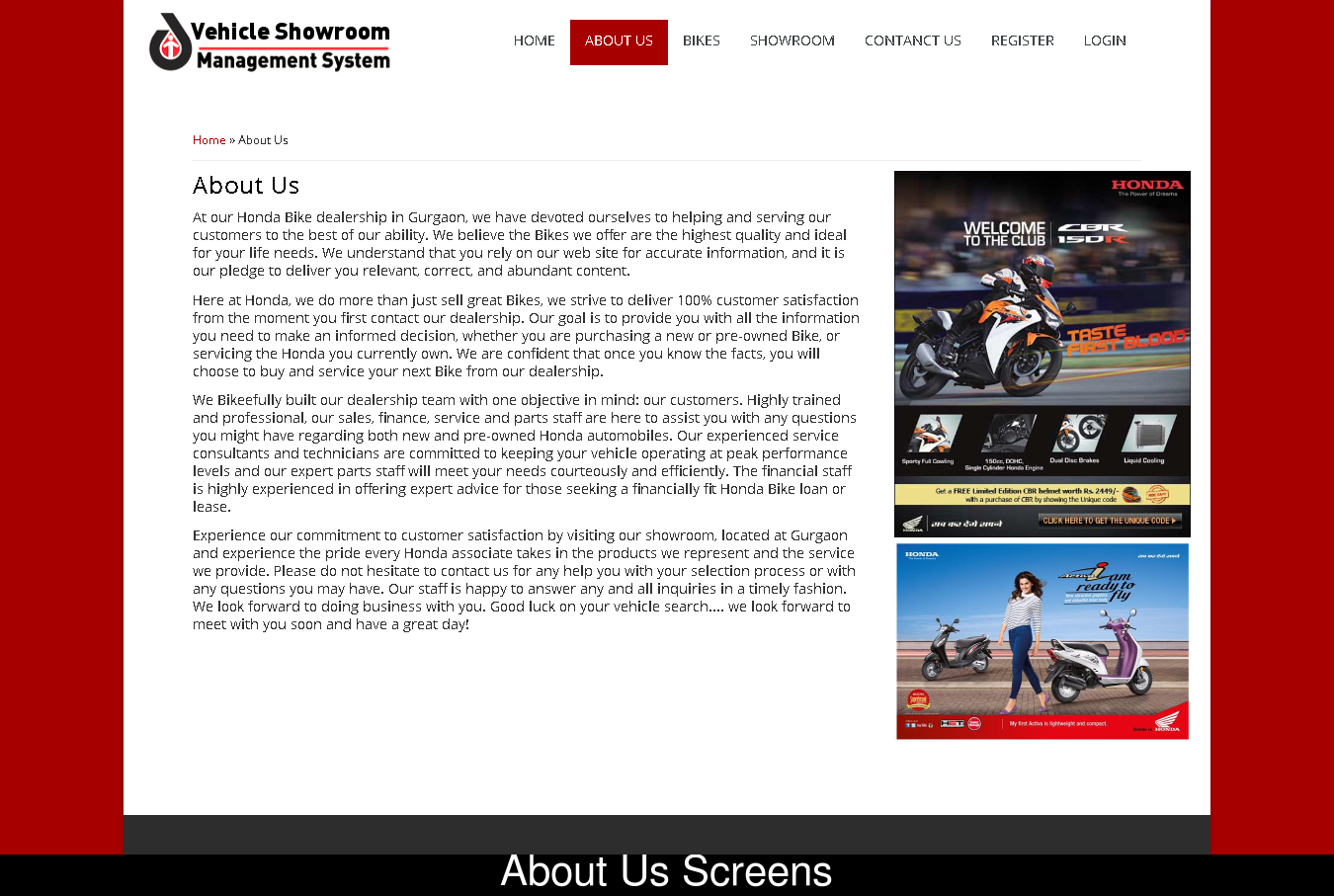
**CONCLUSION**

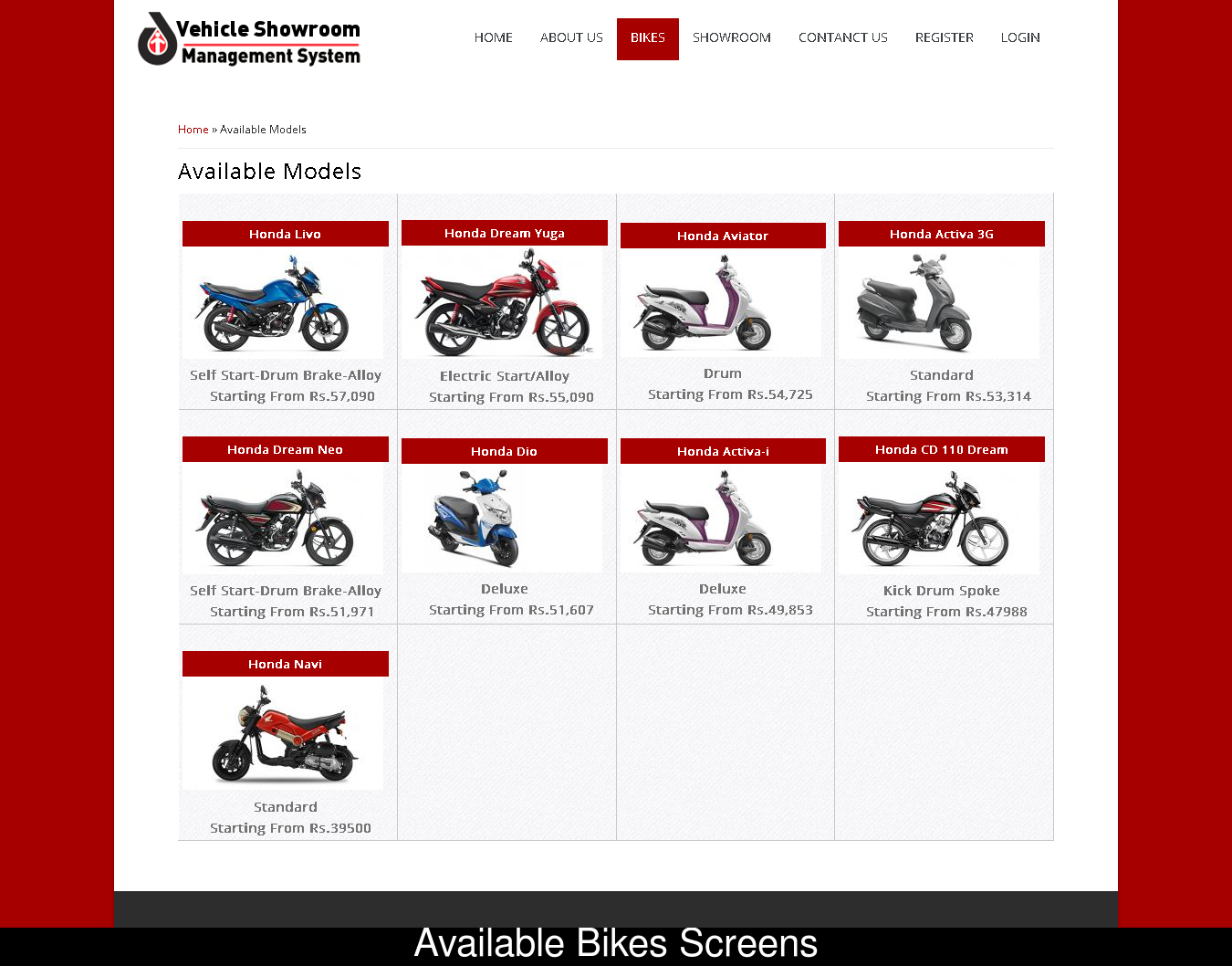
Vehicle showroom business has emerged with a new goodies compared to the past experience where every activity concerning vehicle showroom business is limited to a physical location only. Even though the physical location has not been totally eradicated; the nature of functions and how these functions are achieved has been reshaped by the power of internet. Nowadays, customers can reserve vehicles online, rent vehicle online, and have the vehicle brought to their door step once the customer is a registered member or go to the office to pick the vehicle.

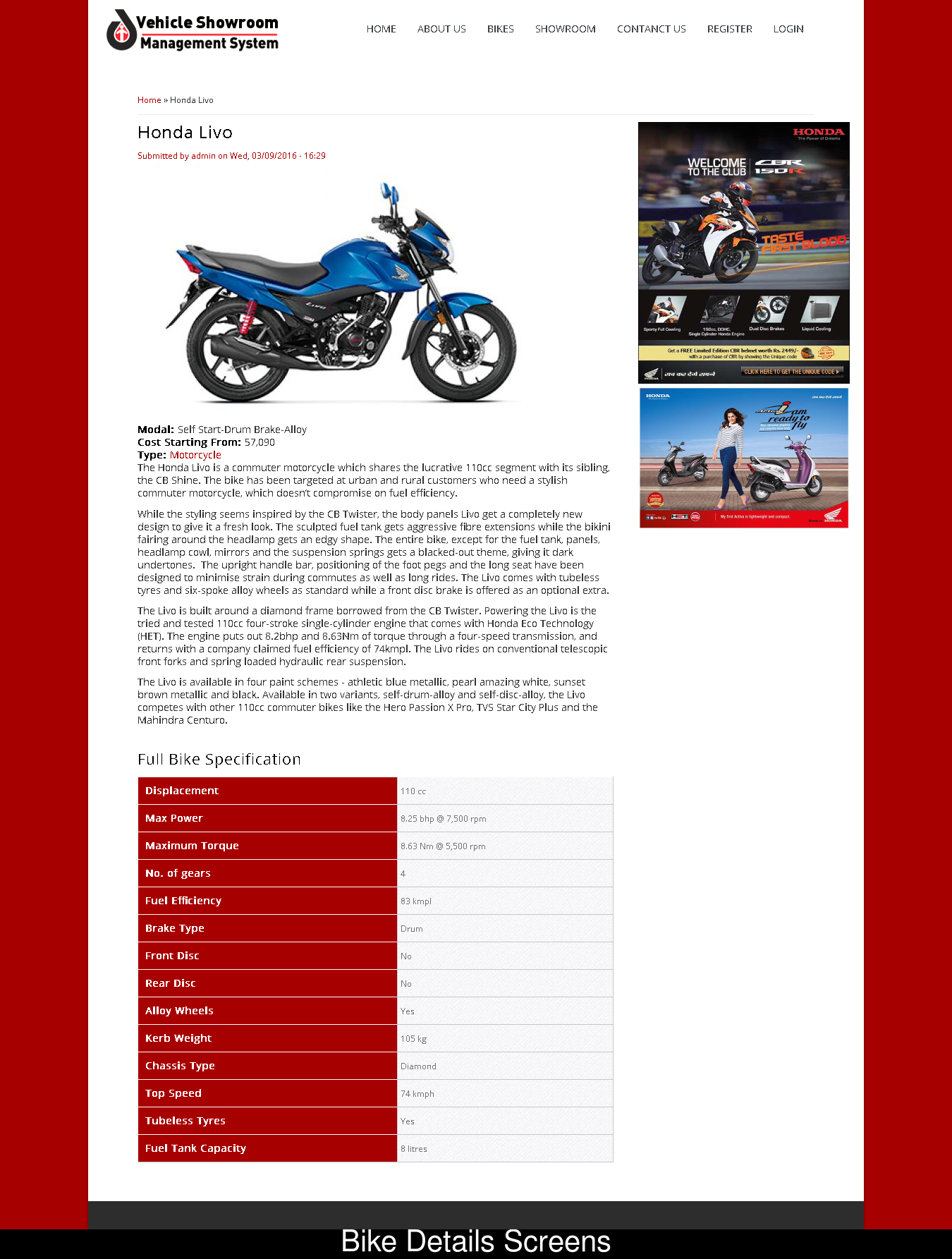
The web based vehicle showroom system has offered an advantage to both customers as well as Vehicle showroom Company to efficiently and effectively manage the business and satisfies customers’ need at the click of a button.

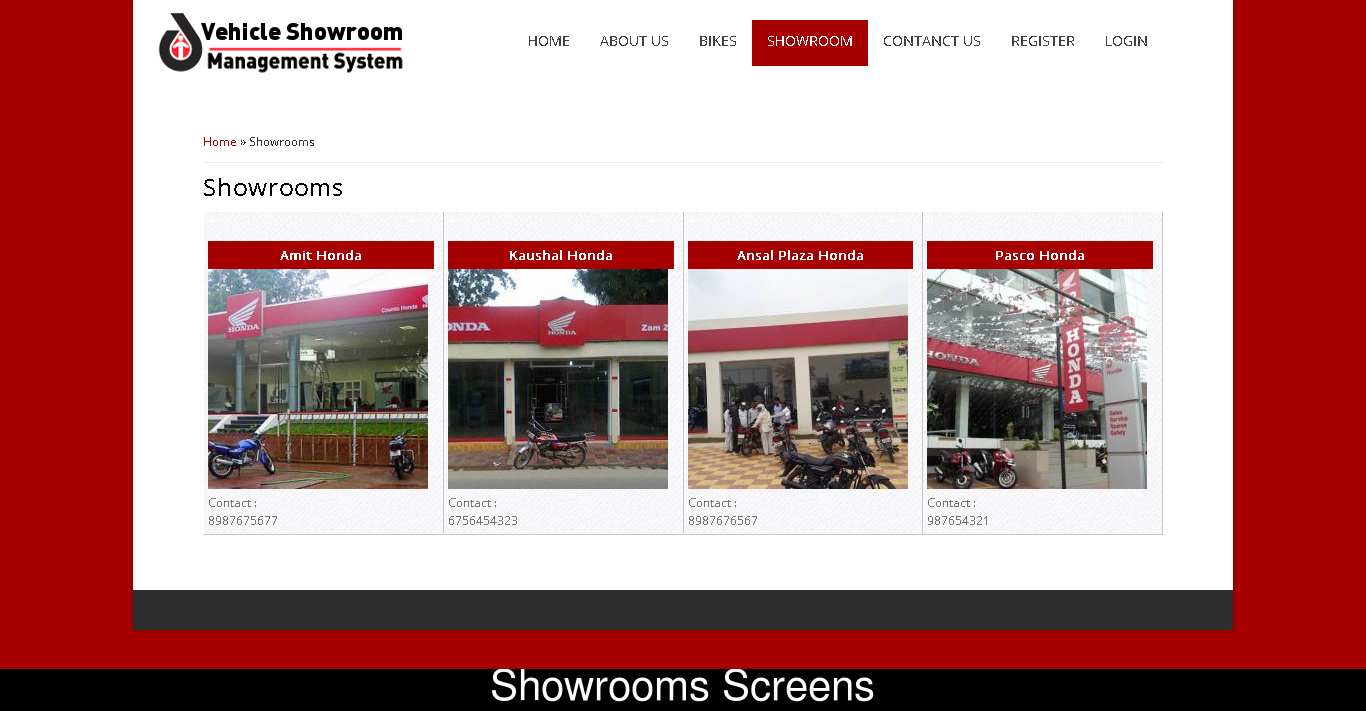


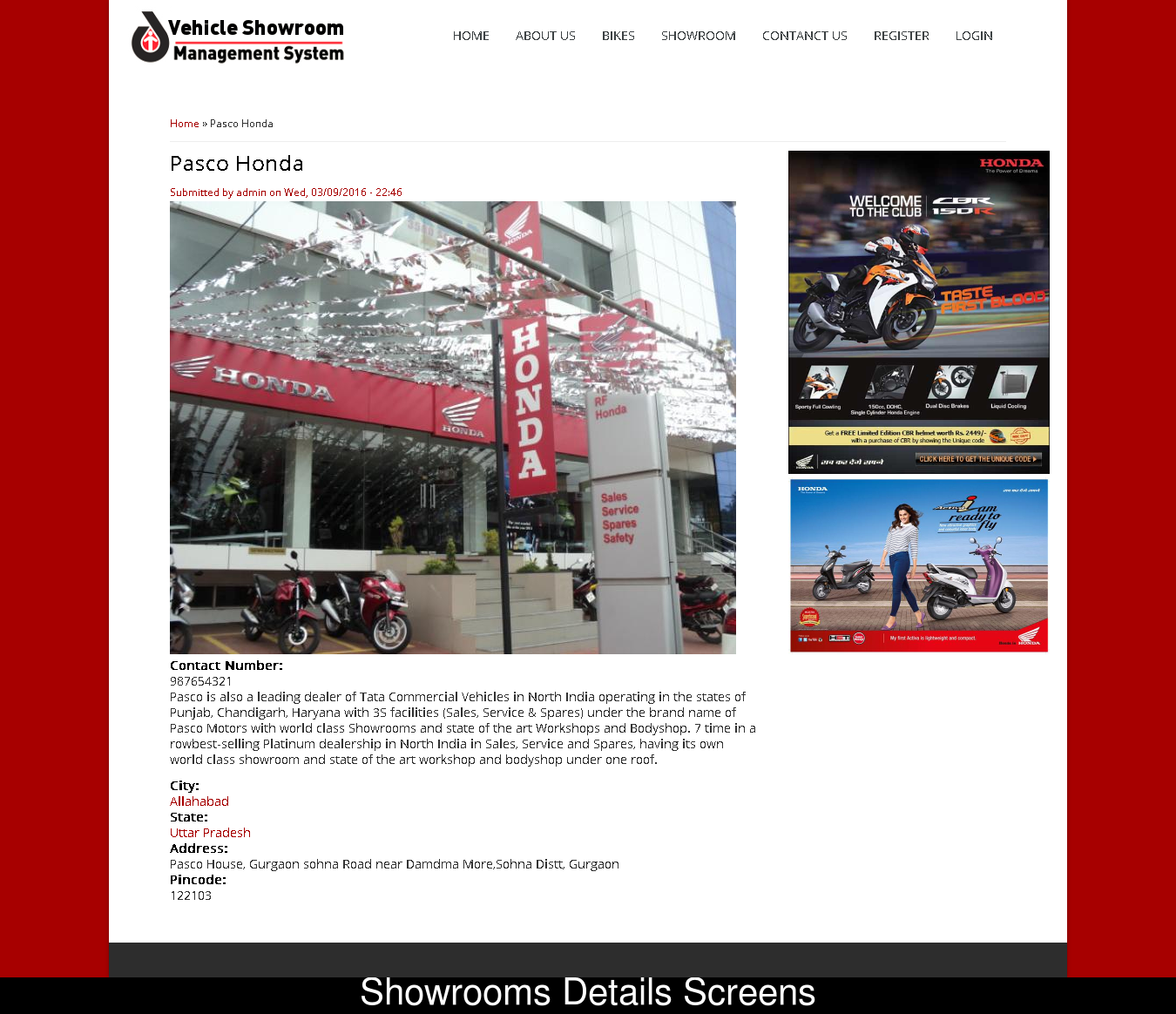


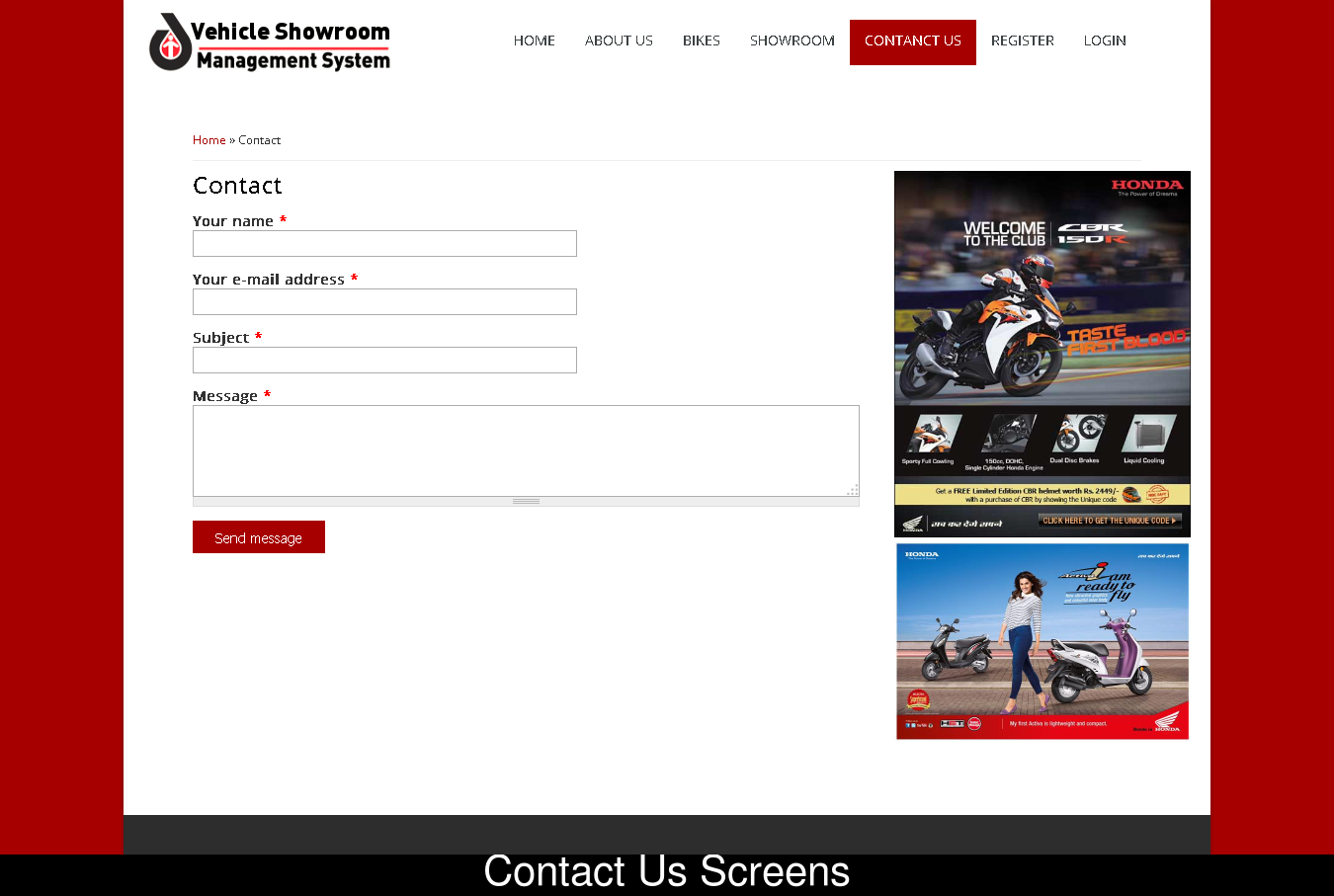




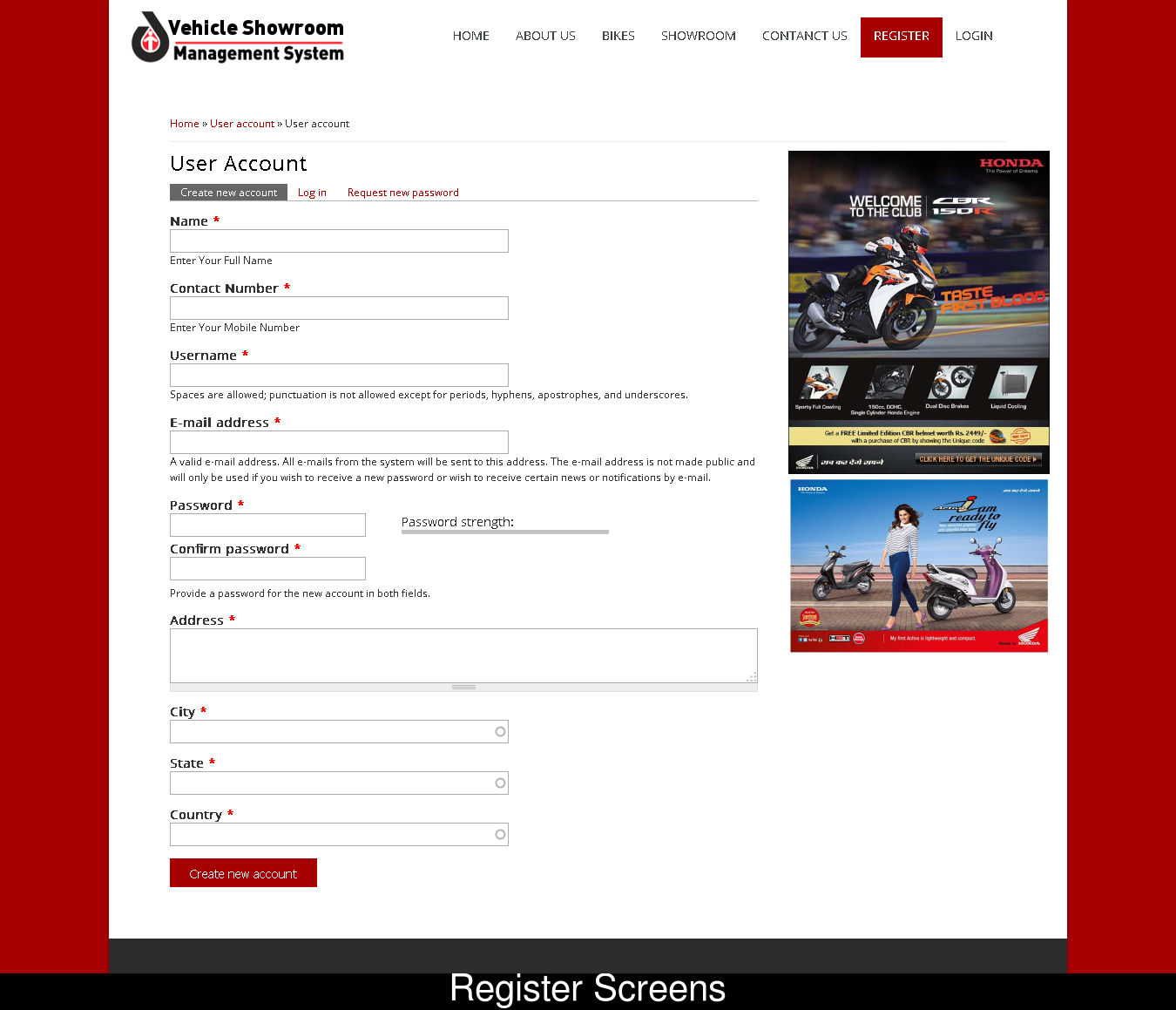


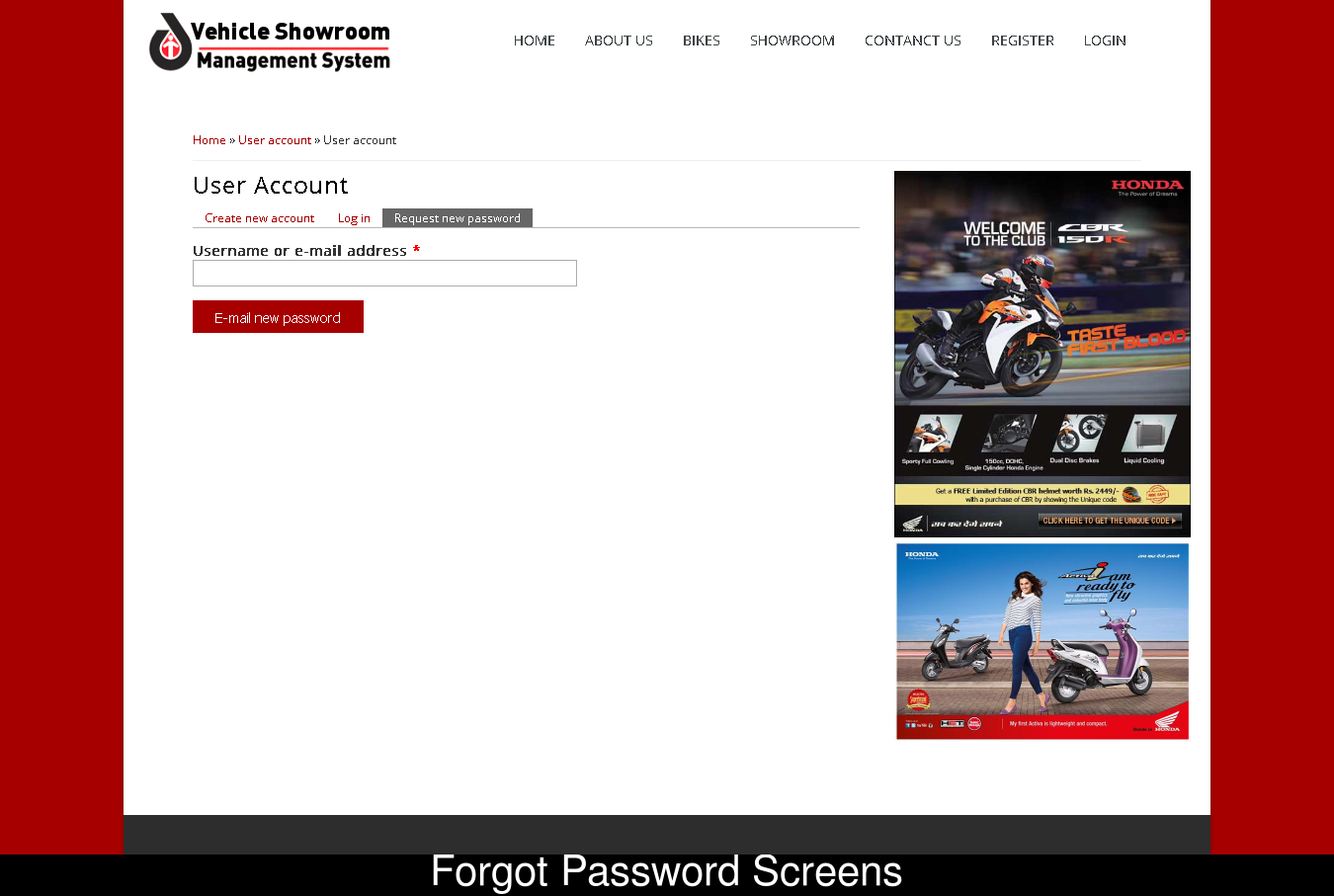


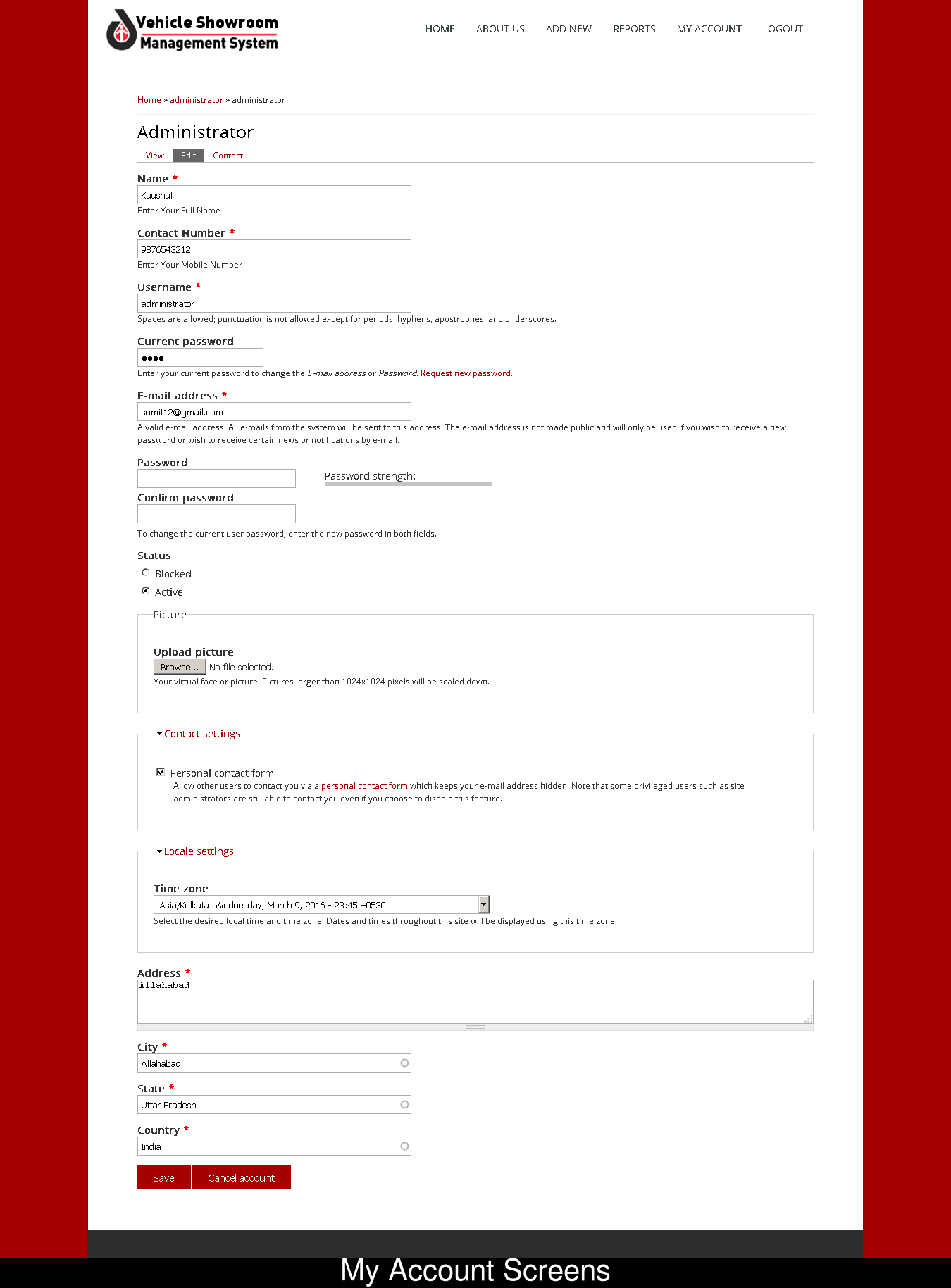


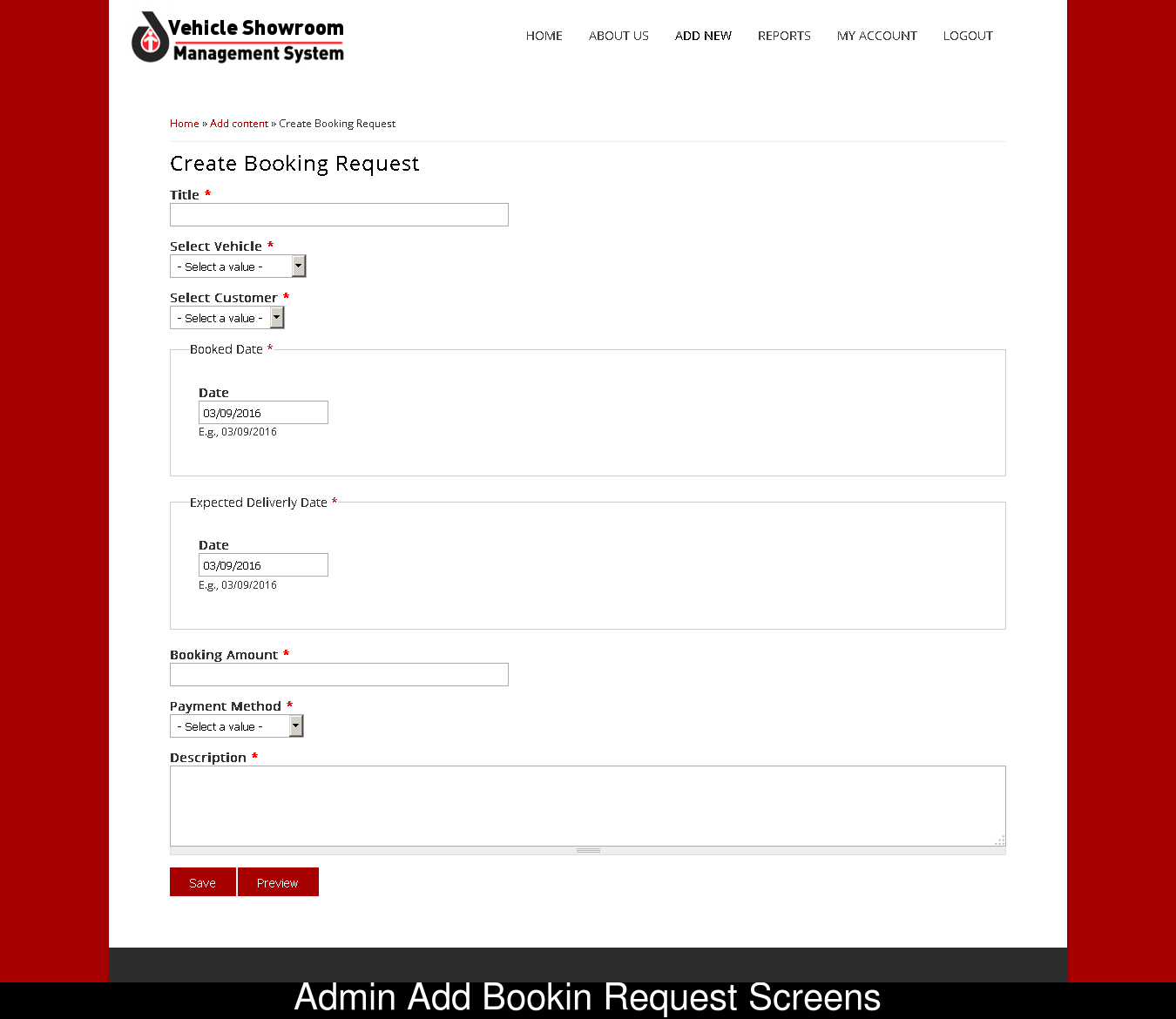


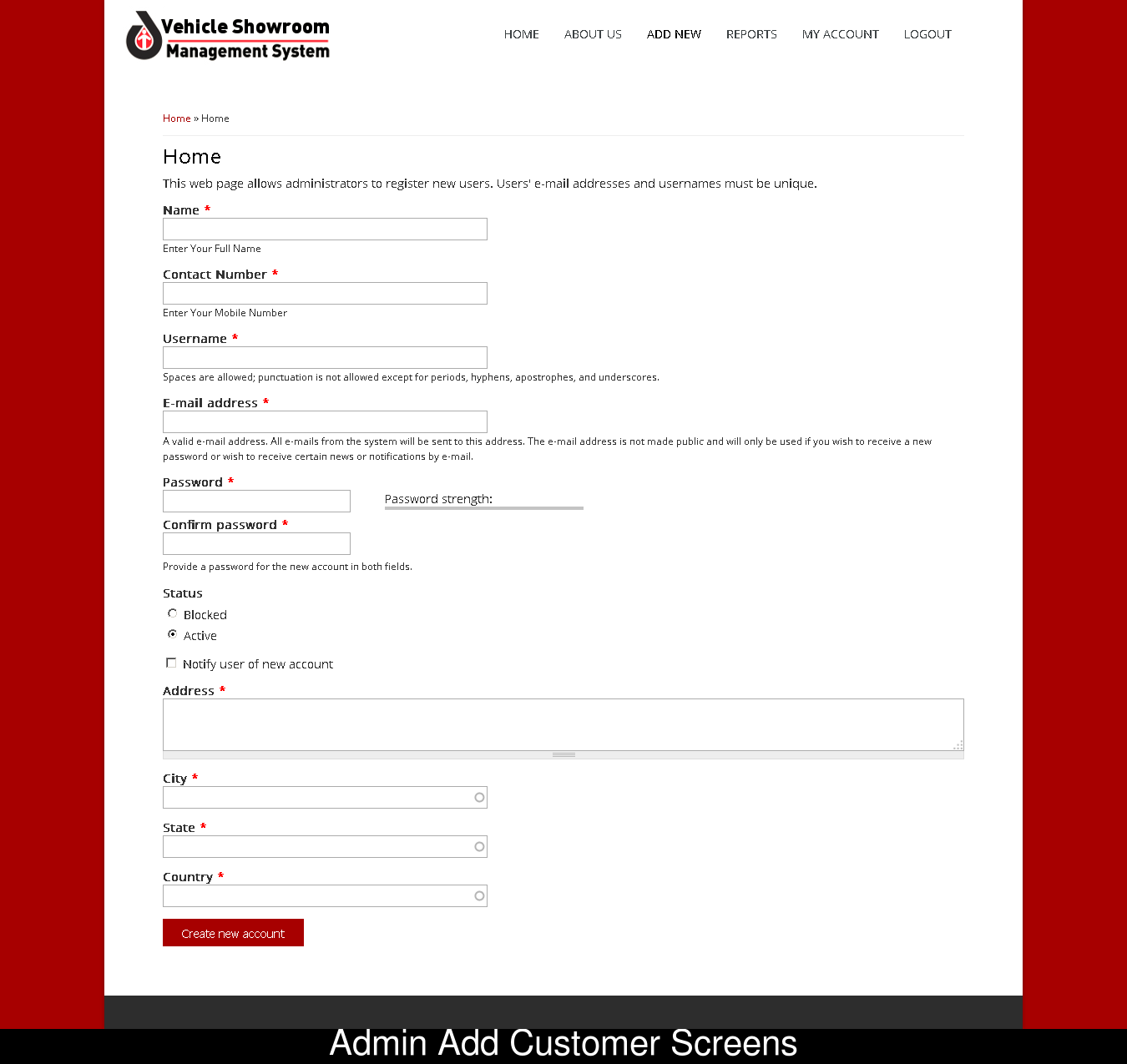


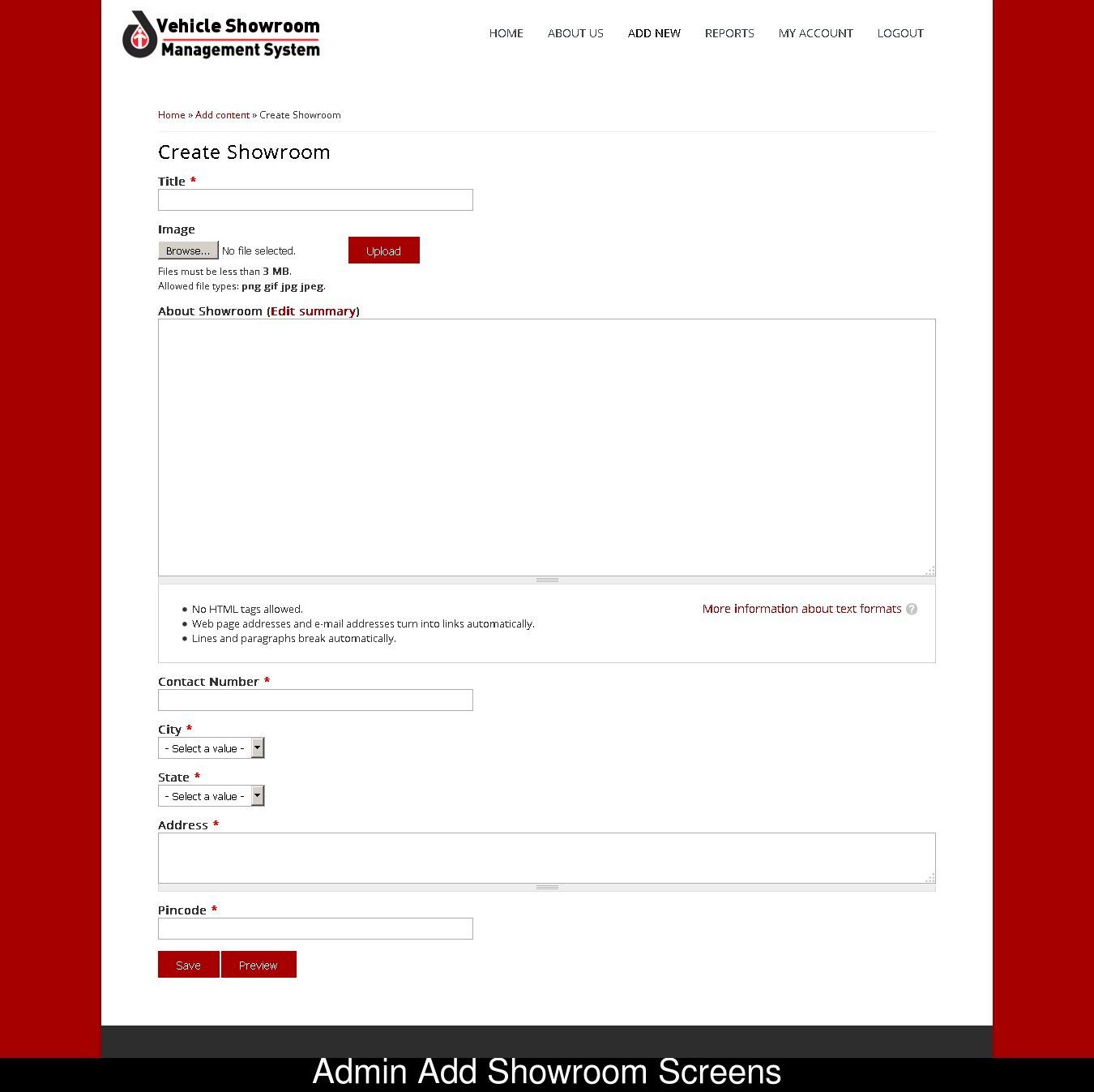


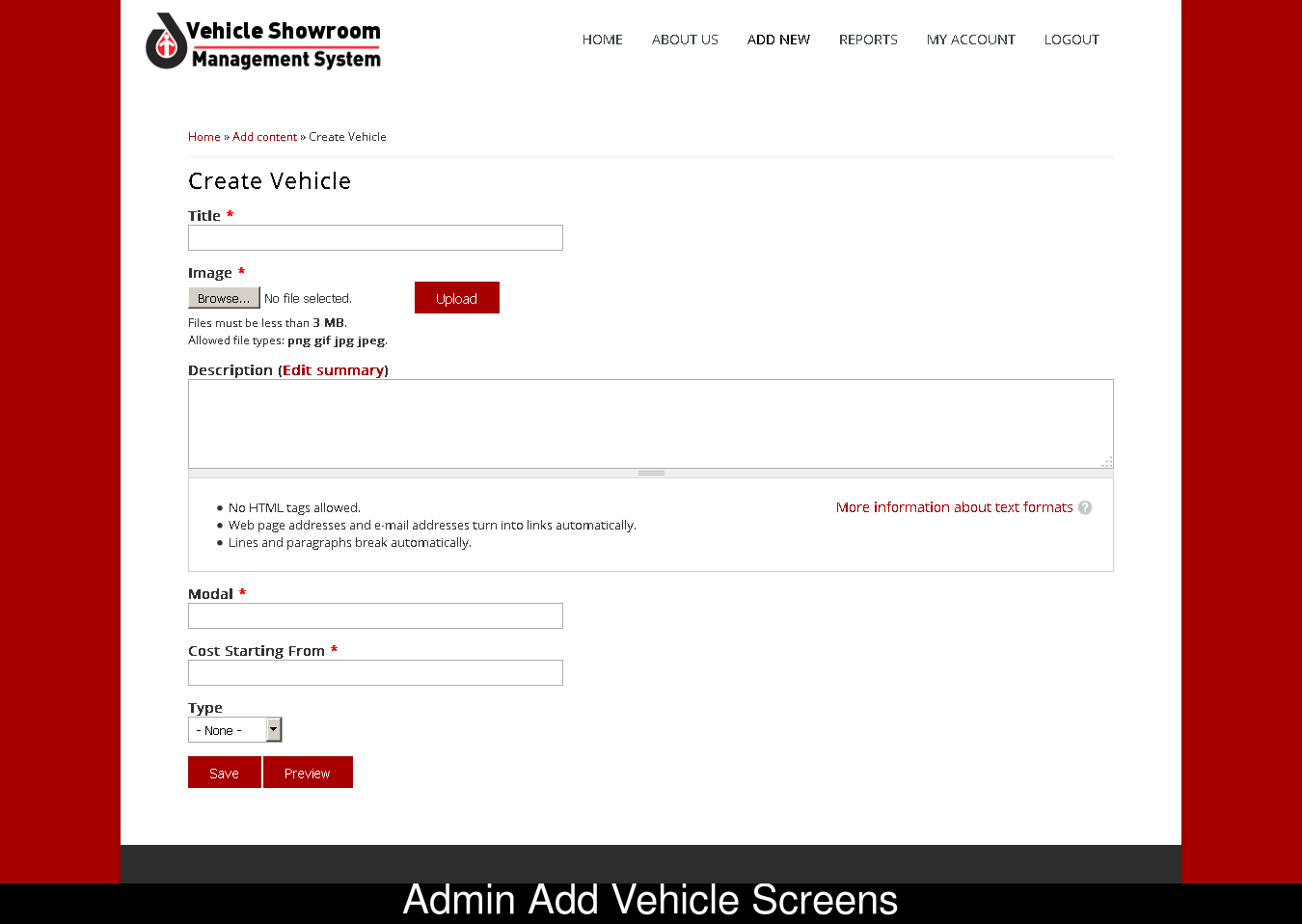


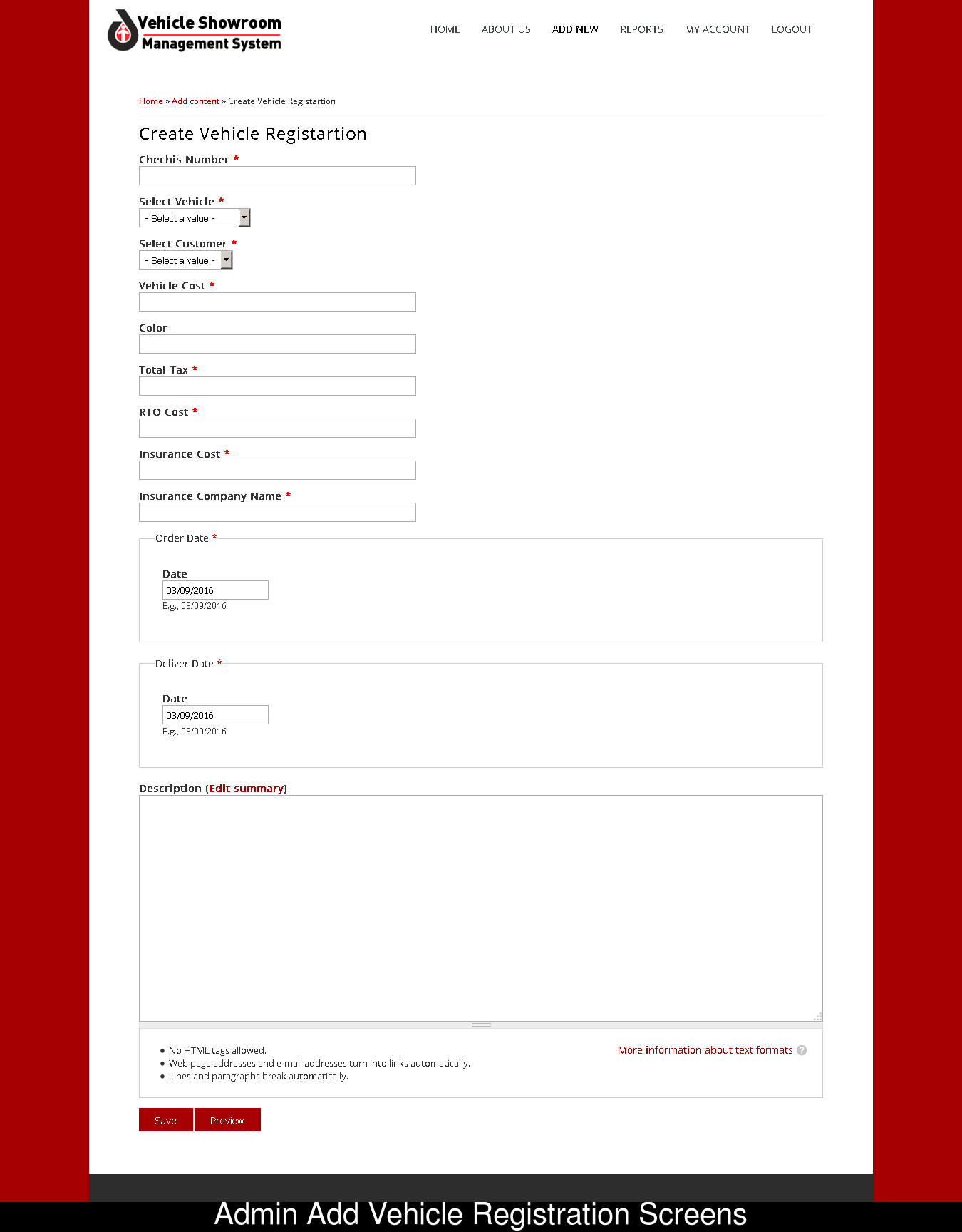


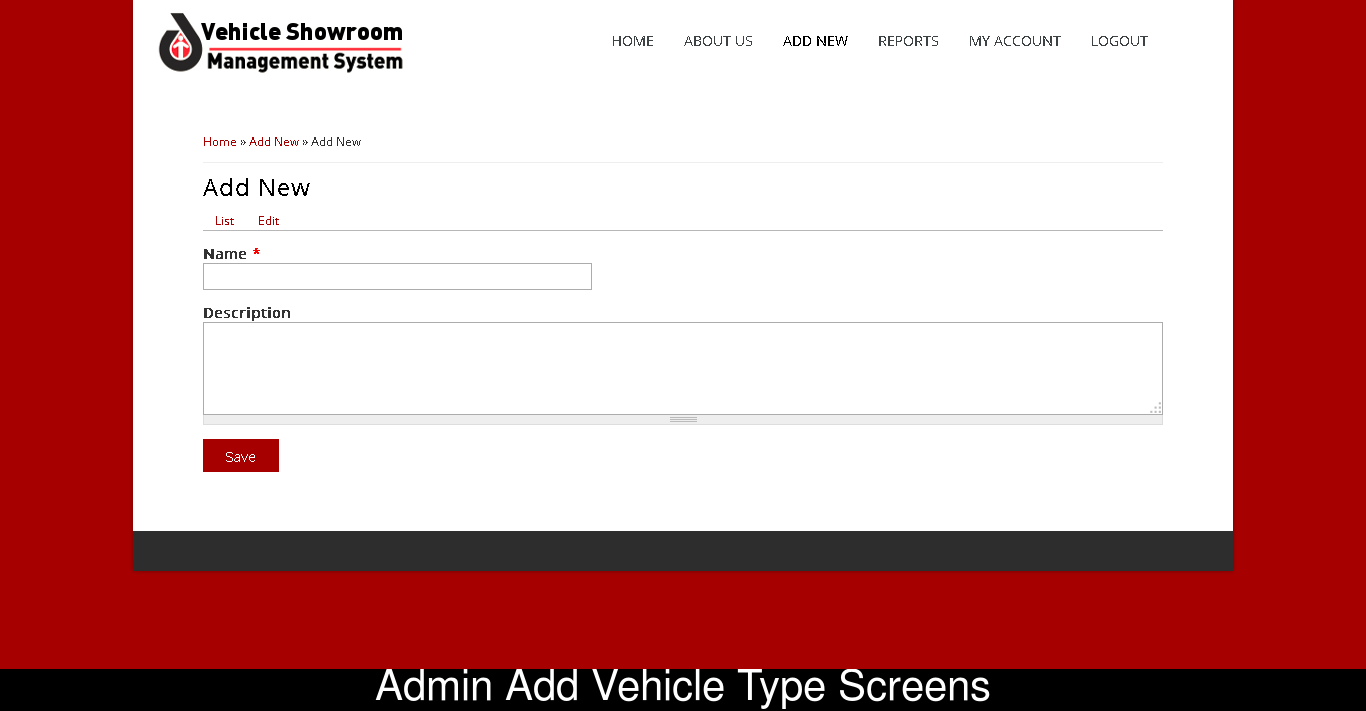


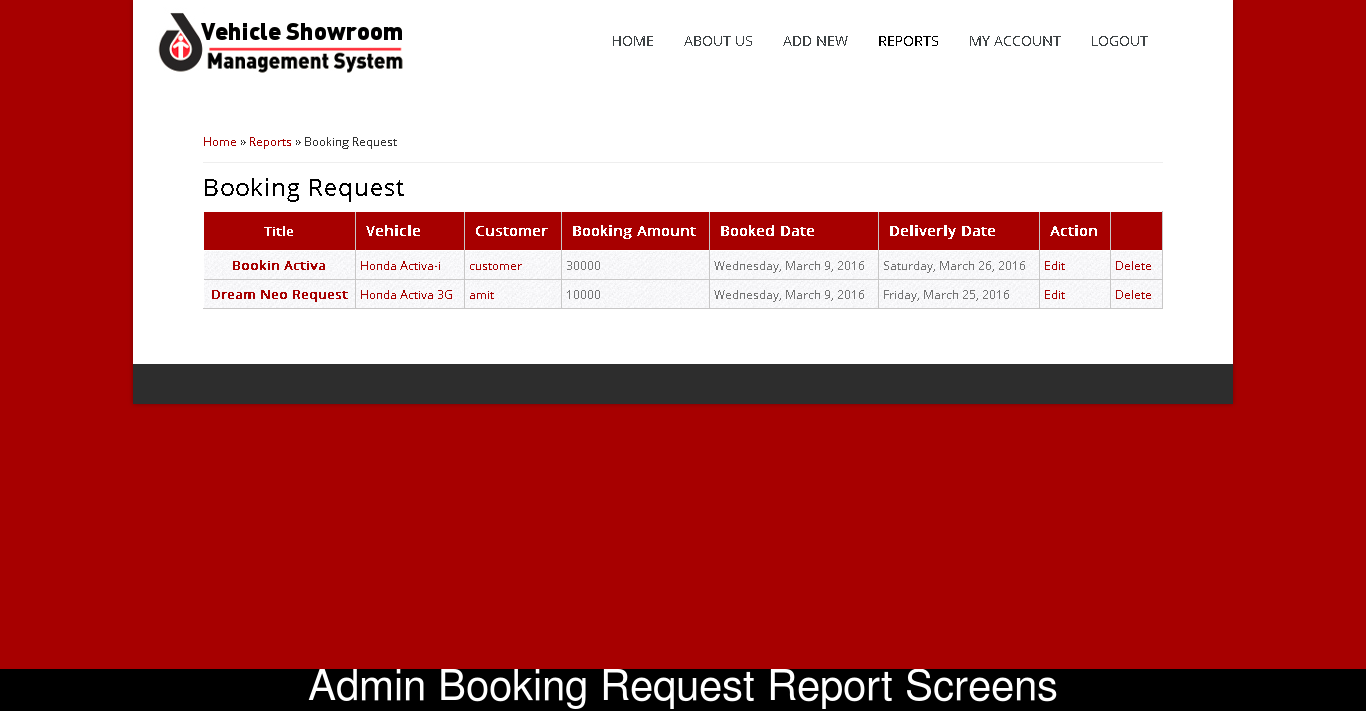


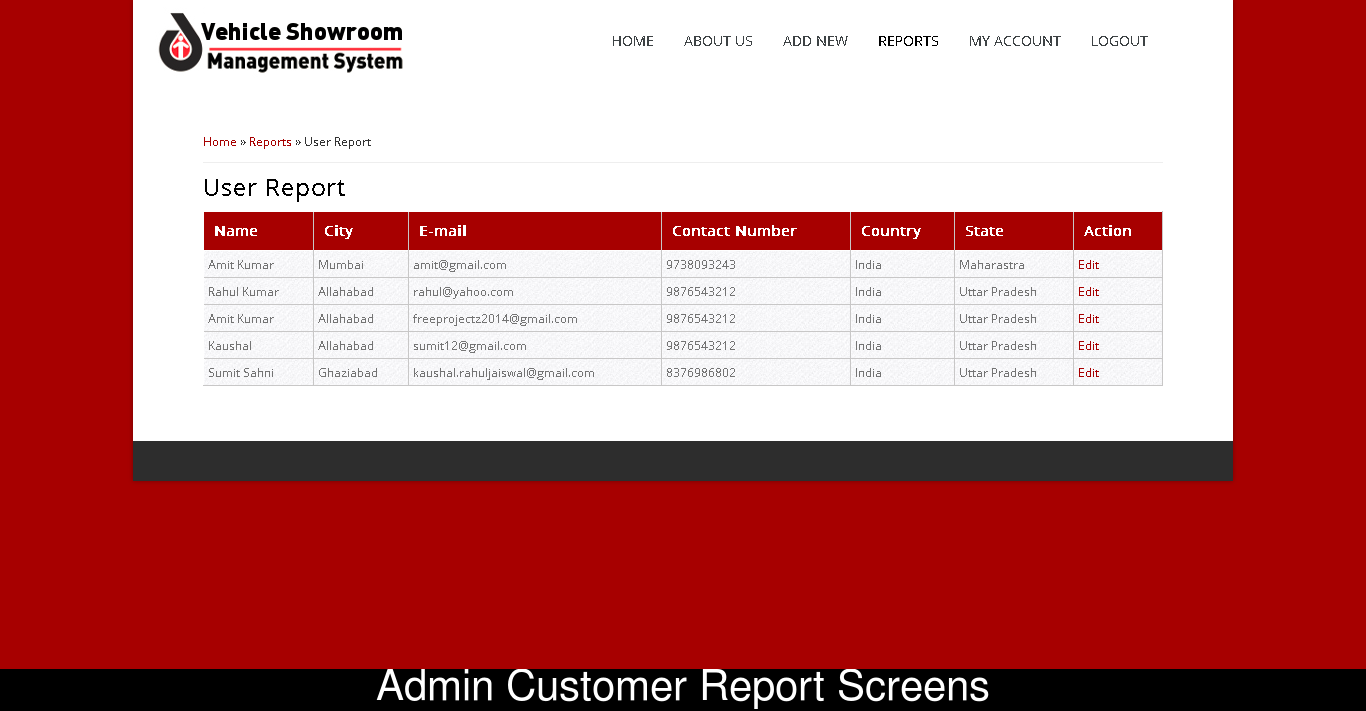




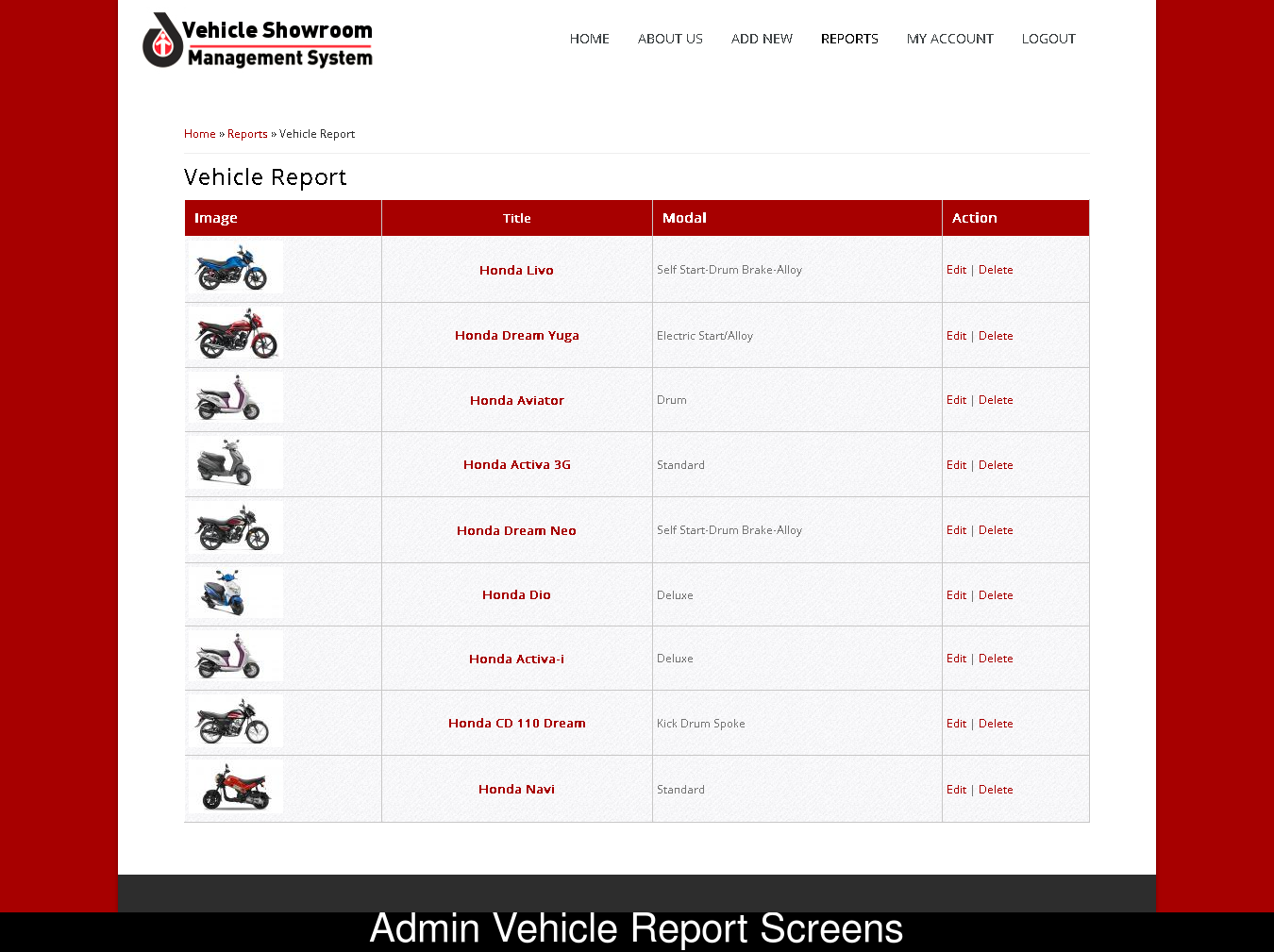


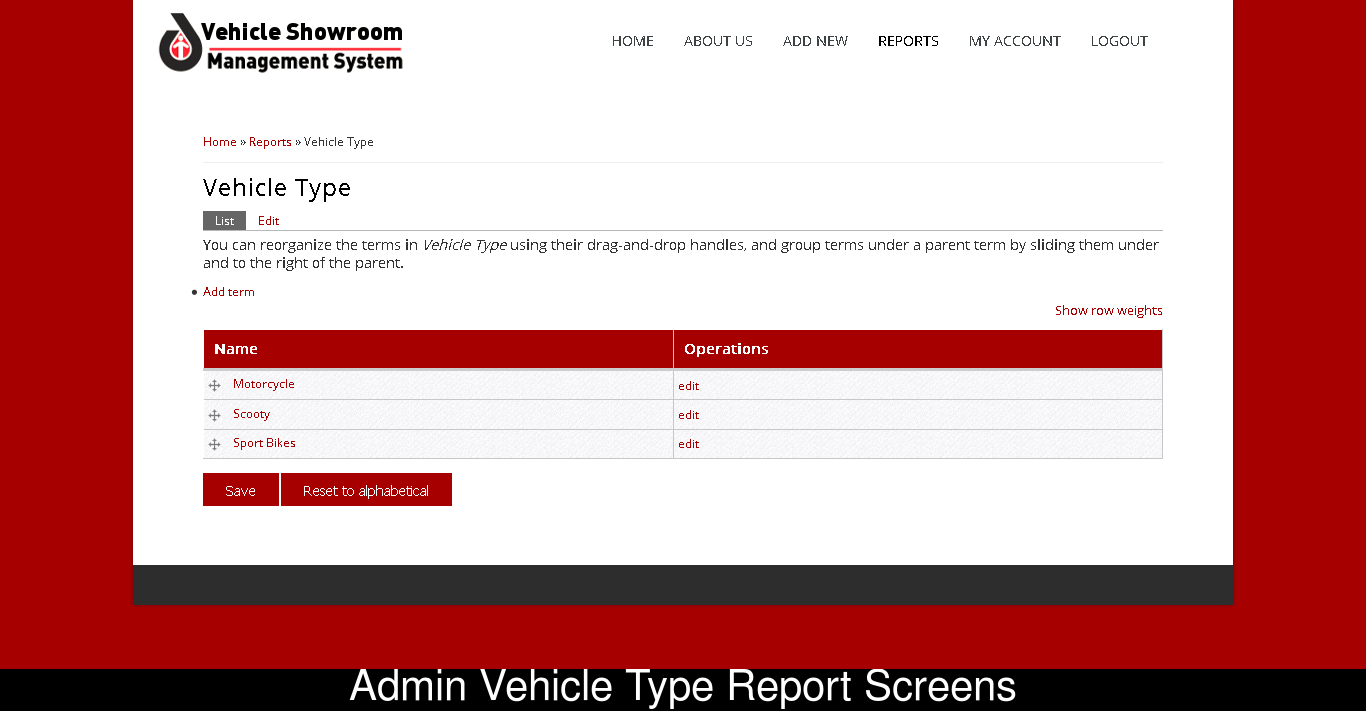


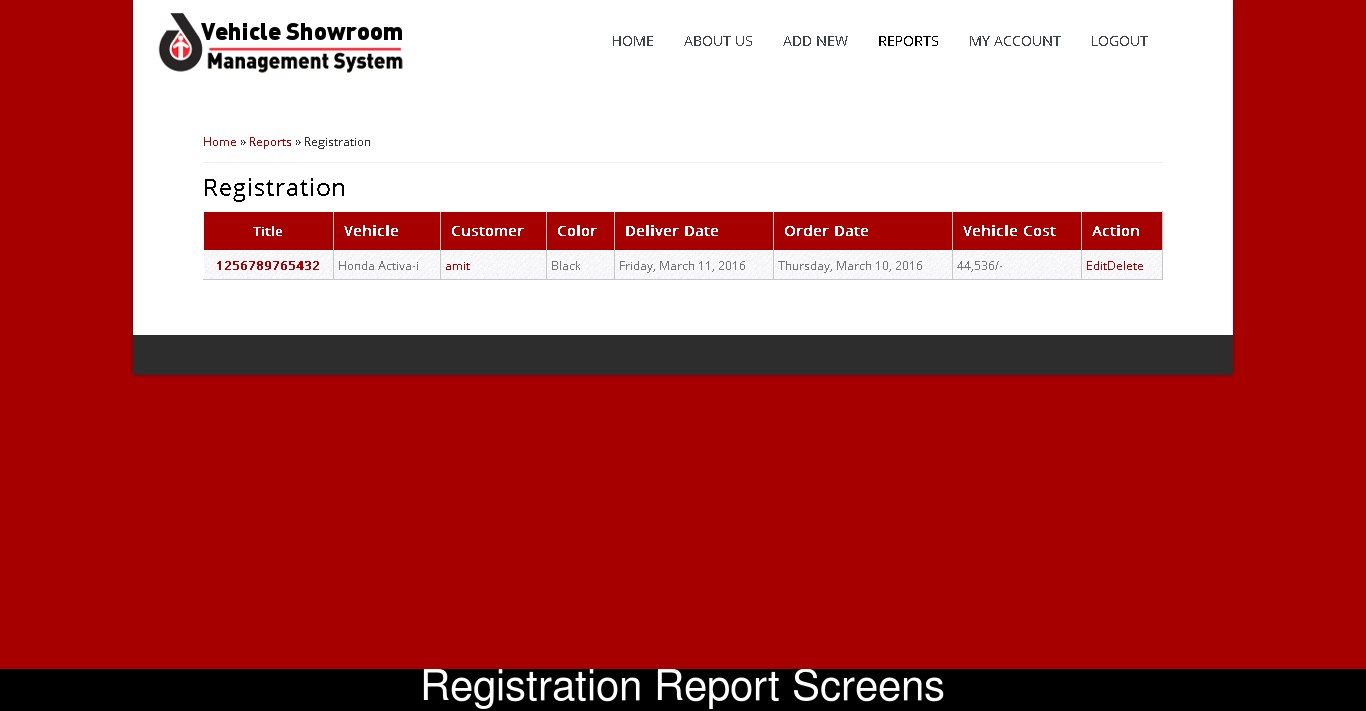












**Code for booking.php**

<?php

include\_once("../includes/db\_connect.php");

include\_once("../includes/functions.php");

if($\_REQUEST[act]=="save\_booking")

{

save\_booking();

exit;

}

if($\_REQUEST[act]=="delete\_booking")

{

delete\_booking();

exit;

}

if($\_REQUEST[act]=="get\_report")

{

get\_report();

exit;

}

###Code for save booking#####

function save\_booking()

{

$R=$\_REQUEST;

if($R[booking\_id])

{

$statement = "UPDATE `booking` SET";

$cond = "WHERE `booking\_id` = '$R[booking\_id]'";

$msg = "Data Updated Successfully.";

}

else

{

$statement = "INSERT INTO `booking` SET";

$cond = "";

$msg="Data saved successfully.";

}

$SQL= $statement."

`booking\_name` = '$R[booking\_name]',

`booking\_mobile` = '$R[booking\_mobile]',

`booking\_email` = '$R[booking\_email]',

`booking\_pickup` = '$R[booking\_pickup]',

`booking\_drop` = '$R[booking\_drop]',

`booking\_from\_date` = '$R[booking\_from\_date]',

`booking\_to\_date` = '$R[booking\_to\_date]',

`booking\_vehicle\_id` = '$R[booking\_vehicle\_id]'".

$cond;

$rs = mysql\_query($SQL) or die(mysql\_error());

if($R[booking\_id] == "") {

$booking\_id = mysql\_insert\_id();

}

header("Location:../booking\_confirmation.php?booking\_id=".$booking\_id);

}

#########Function for delete booking##########3

function delete\_booking()

{

/////////Delete the record//////////

$SQL="DELETE FROM booking WHERE booking\_id = $\_REQUEST[booking\_id]";

mysql\_query($SQL) or die(mysql\_error());

header("Location:../booking-report.php?msg=Deleted Successfully.");

}

##############Function for reporting ##################3

function get\_report()

{

$fname = 'myCSV.csv';

$fp = fopen($fname,'w');

$column\_name = '"ID","booking\_name","booking\_add1","booking\_add2","booking\_state","booking\_email","booking\_city","booking\_mobile","booking\_gender","booking\_dob","booking\_nl\_id","booking\_image"'."\n\r";

fwrite($fp,$column\_name);

$SQL="SELECT \* FROM booking,city WHERE booking\_city = city\_id";

$rs=mysql\_query($SQL);

while($data=mysql\_fetch\_assoc($rs))

{

$csvdata=implode(",",$data)."\n\r";

fwrite($fp,$csvdata);

}

fclose($fp);

header('Content-type: application/csv');

header("Content-Disposition: inline; filename=".$fname);

readfile($fname);

}

?>

**Code for User.php:**

<?php

session\_start();

include\_once("../includes/db\_connect.php");

include\_once("../includes/functions.php");

if($\_REQUEST[act]=="save\_user")

{

save\_user();

exit;

}

if($\_REQUEST[act]=="delete\_user")

{

delete\_user();

exit;

}

if($\_REQUEST[act]=="get\_report")

{

get\_report();

exit;

}

###Code for save user#####

function save\_user()

{

$R=$\_REQUEST;

///Checking Username Exits or not ////

$SQL="SELECT \* FROM user WHERE user\_username = $\_REQUEST[user\_username]";

$rs=mysql\_query($SQL);

$data=mysql\_fetch\_assoc($rs);

if($data['user\_username'] && $R['user\_id'] == "") {

header("Location:../user.php?msg=Username Already Exits. Kindly choose another....");

return;

}

/////////////////////////////////////

$image\_name = $\_FILES[user\_image][name];

$location = $\_FILES[user\_image][tmp\_name];

if($image\_name!="")

{

move\_uploaded\_file($location,"../uploads/".$image\_name);

}

else

{

$image\_name = $R[avail\_image];

}

//die;

if($R[user\_id])

{

$statement = "UPDATE `user` SET";

$cond = "WHERE `user\_id` = '$R[user\_id]'";

$msg = "Data Updated Successfully.";

$condQuery = "";

}

else

{

$statement = "INSERT INTO `user` SET";

$condQuery = "`user\_username` = '$R[user\_username]',

`user\_password` = '$R[user\_password]',";

$cond = "";

$msg="Data saved successfully.";

}

$SQL= $statement."

`user\_level\_id` = '$R[user\_level\_id]',

".

$condQuery

."

`user\_name` = '$R[user\_name]',

`user\_add1` = '$R[user\_add1]',

`user\_add2` = '$R[user\_add2]',

`user\_city` = '$R[user\_city]',

`user\_state` = '$R[user\_state]',

`user\_country` = '$R[user\_country]',

`user\_email` = '$R[user\_email]',

`user\_mobile` = '$R[user\_mobile]',

`user\_gender` = '$R[user\_gender]',

`user\_dob` = '$R[user\_dob]',

`user\_image` = '$image\_name'".

$cond;

$rs = mysql\_query($SQL) or die(mysql\_error());

if($\_SESSION['login']!=1)

{

header("Location:../login.php?msg=You are registered successfully. Login with your credential !!!");

exit;

}

else if($\_SESSION['user\_details']['user\_level\_id'] == 3) {

header("Location:../user.php?user\_id=".$\_SESSION['user\_details']['user\_id']."&msg=Your account updated successfully !!!");

exit;

}

header("Location:../user-report.php?msg=$msg");

}

#########Function for delete user##########3

function delete\_user()

{

$SQL="SELECT \* FROM user WHERE user\_id = $\_REQUEST[user\_id]";

$rs=mysql\_query($SQL);

$data=mysql\_fetch\_assoc($rs);

/////////Delete the record//////////

$SQL="DELETE FROM user WHERE user\_id = $\_REQUEST[user\_id]";

mysql\_query($SQL) or die(mysql\_error());

//////////Delete the image///////////

if($data[user\_image])

{

unlink("../uploads/".$data[user\_image]);

}

header("Location:../user-report.php?msg=Deleted Successfully.");

}

##############Function for reporting ##################3

function get\_report()

{

$fname = 'myCSV.csv';

$fp = fopen($fname,'w');

$column\_name = '"ID","user\_name","user\_add1","user\_add2","user\_state","user\_email","user\_city","user\_mobile","user\_gender","user\_dob","user\_nl\_id","user\_image"'."\n\r";

fwrite($fp,$column\_name);

$SQL="SELECT \* FROM user,city WHERE user\_city = city\_id";

$rs=mysql\_query($SQL);

while($data=mysql\_fetch\_assoc($rs))

{

$csvdata=implode(",",$data)."\n\r";

fwrite($fp,$csvdata);

}

fclose($fp);

header('Content-type: application/csv');

header("Content-Disposition: inline; filename=".$fname);

readfile($fname);

}

?>

**Code for Login.php:**

<?php

session\_start();

include\_once("../includes/db\_connect.php");

if($\_REQUEST[act]=="check\_login")

{

check\_login();

}

if($\_REQUEST[act]=="logout")

{

logout();

}

if($\_REQUEST[act] == "change\_password")

{

change\_password();

}

####Function check user#######

function check\_login()

{

$user\_user=$\_REQUEST[user\_user];

$user\_password=$\_REQUEST[user\_password];

$SQL="SELECT \* FROM user WHERE user\_username = '$user\_user' AND user\_password = '$user\_password'";

$rs = mysql\_query($SQL) or die(mysql\_error());

if(mysql\_num\_rows($rs))

{

$\_SESSION[login]=1;

$\_SESSION['user\_details'] = mysql\_fetch\_assoc($rs);

header("Location:../index.php");

}

else

{

header("Location:../login.php?msg=Invalid User and Password.");

}

}

####Function logout####

function logout()

{

$\_SESSION[login]=0;

$\_SESSION['user\_details'] = 0;

header("Location:../login.php?msg=Logout Successfullly.");

}

#####Function for changing the password ####

function change\_password() {

$R = $\_REQUEST;

if($R['user\_confirm\_password'] != $R['user\_new\_password']) {

header("Location:../change-password.php?msg=Your new passsword and confirm password does not match!!!");

exit;

}

$SQL = "UPDATE `user` SET user\_password = '$R[user\_new\_password]' WHERE `user\_id` = ".$\_SESSION['user\_details']['user\_id'];

$rs = mysql\_query($SQL) or die(mysql\_error());

header("Location:../change-password.php?msg=Your Password Changed Successfully !!!");

print $SQL;

die;

}

?>

**Code for vehicle.php**

<?php

include\_once("../includes/db\_connect.php");

include\_once("../includes/functions.php");

if($\_REQUEST[act]=="save\_vehicle")

{

save\_vehicle();

exit;

}

if($\_REQUEST[act]=="delete\_vehicle")

{

delete\_vehicle();

exit;

}

if($\_REQUEST[act]=="get\_report")

{

get\_report();

exit;

}

###Code for save vehicle#####

function save\_vehicle()

{

$R=$\_REQUEST;

$image\_name = $\_FILES[vehicle\_image][name];

$location = $\_FILES[vehicle\_image][tmp\_name];

if($image\_name!="")

{

move\_uploaded\_file($location,"../uploads/".$image\_name);

}

else

{

$image\_name = $R[avail\_image];

}

$vehicle\_nl\_id=implode(",",$R[vehicle\_nl\_id]);

if($R[vehicle\_id])

{

$statement = "UPDATE `vehicle` SET";

$cond = "WHERE `vehicle\_id` = '$R[vehicle\_id]'";

$msg = "Data Updated Successfully.";

}

else

{

$statement = "INSERT INTO `vehicle` SET";

$cond = "";

$msg="Data saved successfully.";

}

$SQL= $statement."

`vehicle\_name` = '$R[vehicle\_name]',

`vehicle\_description` = '$R[vehicle\_description]',

`vehicle\_number` = '$R[vehicle\_number]',

`vehicle\_company` = '$R[vehicle\_company]',

`vehicle\_city` = '$R[vehicle\_city]',

`vehicle\_type` = '$R[vehicle\_type]',

`vehicle\_seats` = '$R[vehicle\_seats]',

`vehicle\_price\_per\_day` = '$R[vehicle\_price\_per\_day]',

`vehicle\_image` = '$image\_name'".

$cond;

$rs = mysql\_query($SQL) or die(mysql\_error());

header("Location:../vehicle-report.php?msg=$msg");

}

#########Function for delete vehicle##########3

function delete\_vehicle()

{

$SQL="SELECT \* FROM vehicle WHERE vehicle\_id = $\_REQUEST[vehicle\_id]";

$rs=mysql\_query($SQL);

$data=mysql\_fetch\_assoc($rs);

/////////Delete the record//////////

$SQL="DELETE FROM vehicle WHERE vehicle\_id = $\_REQUEST[vehicle\_id]";

mysql\_query($SQL) or die(mysql\_error());

//////////Delete the image///////////

if($data[vehicle\_image])

{

unlink("../uploads/".$data[vehicle\_image]);

}

header("Location:../vehicle-report.php?msg=Deleted Successfully.");

}

##############Function for reporting ##################3

function get\_report()

{

$fname = 'myCSV.csv';

$fp = fopen($fname,'w');

$column\_name = '"ID","vehicle\_name","vehicle\_add1","vehicle\_add2","vehicle\_state","vehicle\_email","vehicle\_city","vehicle\_mobile","vehicle\_gender","vehicle\_dob","vehicle\_nl\_id","vehicle\_image"'."\n\r";

fwrite($fp,$column\_name);

$SQL="SELECT \* FROM vehicle,city WHERE vehicle\_city = city\_id";

$rs=mysql\_query($SQL);

while($data=mysql\_fetch\_assoc($rs))

{

$csvdata=implode(",",$data)."\n\r";

fwrite($fp,$csvdata);

}

fclose($fp);

header('Content-type: application/csv');

header("Content-Disposition: inline; filename=".$fname);

readfile($fname);

}

?>

**Code for functions.php:**

<?php

$SERVER\_PATH = "http://127.0.0.1:81/vehicle\_rental\_system/";

##Function for generating the dynamic options #######

function get\_new\_optionlist($table,$id\_col,$value\_col,$selected=0)

{

$SQL="SELECT \* FROM $table ORDER BY $value\_col";

$rs=mysql\_query($SQL);

$option\_list="<option value=''>Please Select</option>";

while($data=mysql\_fetch\_assoc($rs))

{

if($selected==$data[$id\_col])

{

$option\_list.="<option value='$data[$id\_col]' selected>$data[$value\_col]</option>";

}

else

{

$option\_list.="<option value='$data[$id\_col]'>$data[$value\_col]</option>";

}

}

return $option\_list;

}

##Function for generating the dynamic options #######

function get\_checkbox($name,$table,$id\_col,$value\_col,$selected=0)

{

$selected\_array=explode(",",$selected);

$SQL="SELECT \* FROM $table ORDER BY $value\_col";

$rs=mysql\_query($SQL);

$option\_list="";

while($data=mysql\_fetch\_assoc($rs))

{

if(in\_array($data[$id\_col],$selected\_array))

{

$option\_list.="<input type='checkbox' value='$data[$id\_col]' name='".$name."[]' id='$name' checked>$data[$value\_col]<br>";

}

else

{

$option\_list.="<input type='checkbox' value='$data[$id\_col]' name='".$name."[]' id='$name'>$data[$value\_col]<br>";

}

}

return $option\_list;

}

?>

**Code for header.php**

<?php

session\_start();

include\_once("db\_connect.php");

include\_once("functions.php");

ini\_set("display\_errors",0);

error\_reporting(E\_ERROR);

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Online Vehicle showroom System</title>

<!-- // Stylesheets // -->

<link rel="stylesheet" type="text/css" href="./css/style.css" />

<link rel="stylesheet" type="text/css" href="./css/ddsmoothmenu.css" />

<link rel="stylesheet" type="text/css" href="./css/contentslider.css" />

<link rel="stylesheet" type="text/css" href="./css/jquery.fancybox-1.3.1.css" />

<link rel="stylesheet" type="text/css" href="./css/slider.css" />

<link rel="stylesheet" type="text/css" href="./css/jquery-ui.css">

<!-- // Javascripts // -->

<script type="text/javascript" src="js/jquery-1.10.2.js"></script>

<script type="text/javascript" src="./js/jquery.easing.1.2.js"></script>

<script type="text/javascript" src="./js/jquery.anythingslider.js"></script>

<script type="text/javascript" src="./js/anythingslider.js"></script>

<script type="text/javascript" src="./js/animatedcollapse.js"></script>

<script type="text/javascript" src="./js/ddsmoothmenu.js"></script>

<script type="text/javascript" src="./js/menu.js"></script>

<script type="text/javascript" src="./js/contentslider.js"></script>

<script type="text/javascript" src="./js/ddaccordion.js"></script>

<script type="text/javascript" src="./js/acrodin.js"></script>

<script type="text/javascript" src="./js/jquery.fancybox-1.3.1.js"></script>

<script type="text/javascript" src="./js/lightbox.js"></script>

<script type="text/javascript" src="./js/menu-collapsed.js"></script>

<script type="text/javascript" src="./js/cufon-yui.js"></script>

<script type="text/javascript" src="./js/trebuchet\_ms\_400-trebuchet\_ms\_700-trebuchet\_ms\_italic\_700-trebuchet\_ms\_italic\_400.font.js"></script>

<script type="text/javascript" src="./js/cufon.js"></script>

<script type="text/javascript" src="./js/jquery.validate.js"></script>

<script type="text/javascript" src="js/jquery-ui.js"></script>

</head>

<body>

<div id="wrapper\_sec">

<div id="masthead">

<div class="logo">

<a href="./index.html"><img src="./images/logo.png" alt="" /></a>

</div>

<div class="slogan"></div>

<div class="clear"></div>

<div class="navigation">

<div id="smoothmenu1" class="ddsmoothmenu">

<ul>

<li><a href="./index.php">Home</a>

</li>

<li><a href="./about.php">About Us</a></li>

<li><a href="./book\_vehicle.php">Book A Vehicle</a></li>

<li><a href="./vehicle-list.php">Vehicles List</a></li>

<?php if($\_SESSION['user\_details']['user\_level\_id'] == 1) {?>

<li><a href="./contact.php">Administration</a>

<ul>

<li><a href="vehicle.php">Add New Vehicle</a></li>

<li><a href="book\_vehicle.php">Book A Vehicle</a></li>

<li><a href="user.php">Add System User</a></li>

</ul>

</li>

<li><a href="./contact.php">Reports</a>

<ul>

<li><a href="vehicle-report.php">Vehicle Reports</a></li>

<li><a href="booking-report.php">Booking Reports</a></li>

<li><a href="vehicle-list.php">Vehicle Listing</a></li>

<li><a href="user-report.php">User Reports</a></li>

</ul>

</li>

<?php } if($\_SESSION['login'] == 1) {?>

<li><a href="./user.php?user\_id=<?php echo $\_SESSION['user\_details']['user\_id']; ?>">My Account</a></li>

<li><a href="change-password.php">Change Password</a></li>

<li><a href="./lib/login.php?act=logout">Logout</a></li>

<?php } else { ?>

<li><a href="./user.php">Register</a></li>

<li><a href="./login.php">Login</a></li>

<li><a href="./contact.php">Contact Us</a></li>

<?php }?>

</ul>

<br style="clear: left" />

</div>

<!--

<ul class="searchsec">

<li><input type="text" value="Search" id="searchBox" name="s" onblur="if(this.value == '') { this.value = 'Search'; }" onfocus="if(this.value == 'Search') { this.value = ''; }" class="bar" /></li>

<li><input type="image" src="./images/go.gif" alt="" class="go" /></li>

</ul>

-->

</div>

</div>

**Code for footer.php:**

<div class="clear"></div>

<div id="footer">

<ul class="right">

<li>© Online Vehicle showroom System</li>

</ul>

<div class="clear"></div>

</div>

</body>

</html>

**Code for db\_connect.php**

<?php

error\_reporting(E\_ERROR);

$db=mysql\_connect("localhost","root","") or die(mysql\_error());

$db\_sel=mysql\_select\_db("vehicle\_rental\_system") or die(mysql\_error());

?>

**Code for booking\_confirmation.php**

<?php

include\_once("includes/header.php");

if($\_REQUEST[booking\_id])

{

$SQL="SELECT \* FROM vehicle,company,booking WHERE vehicle\_company = company\_id AND vehicle\_id = booking\_vehicle\_id AND booking\_id = $\_REQUEST[booking\_id]";

$rs=mysql\_query($SQL) or die(mysql\_error());

$data=mysql\_fetch\_assoc($rs);

}

$R = $\_REQUEST;

?>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content\_sec">

<div class="col1">

<?php if($\_SESSION['login'] != 1) {?>

<div style="color:#ff0000; font-size:14px; font-weight:bold;padding:10px 0">Your booking was succesfull. We will contact you soon regarding your booking.<br> Your booking details are as below:</div>

<?php } ?>

<div class="news">

<h4 class="heading colr">Your booking details and receipt</h4>

<div class="featured\_news">

<div class="thumb"><span class="featured">&nbsp;</span><a href="#"><img src="<?=$SERVER\_PATH.'uploads/'.$data[vehicle\_image]?>" alt="" style="width:149px; height:149px" /></a></div>

<div class="cont">

<p class="featuredate" style="margin-left:322px"><?=$data['booking\_from\_date']?></p>

<h4><?=$data[vehicle\_name]?></h4>

<div style="font-weight:bold; font-size:12px;">Price Per Day : <?=$data[vehicle\_price\_per\_day]?></div>

<div style="font-weight:bold; font-size:12px;">Company Name : <?=$data[company\_name]?></div>

<div style="font-weight:bold; font-size:12px;">Number of Seats : <?=$data[vehicle\_seats]?></div>

<p class="txt" style="margin-top:5px;">

<?=$data['vehicle\_description']?>

</p>

</div>

</div>

</div>

<div class="contact">

<h4 class="heading colr">Pickup/Dropoff Address Details</h4>

<form action="lib/booking.php" enctype="multipart/form-data" method="post" name="frm\_vehicle">

<ul class="forms">

<li class="txt">Booking ID</li>

<li class="inputfield" style="font-size:12px; font-weight:bold"><?=$data['booking\_id']?></li>

</ul>

<ul class="forms">

<li class="txt">Name</li>

<li class="inputfield" style="font-size:12px; font-weight:bold"><?=$data['booking\_name']?></li>

</ul>

<ul class="forms">

<li class="txt">Mobile</li>

<li class="inputfield" style="font-size:12px; font-weight:bold"><?=$data['booking\_mobile']?></li>

</ul>

<ul class="forms">

<li class="txt">Email</li>

<li class="inputfield" style="font-size:12px; font-weight:bold"><?=$data['booking\_email']?></li>

</ul>

<ul class="forms">

<li class="txt">IC Number</li>

<li class="inputfield" style="font-size:12px; font-weight:bold"><?=nl2br($data['booking\_pickup'])?></li>

</li>

</ul>

<ul class="forms">

<li class="txt">Collect Your Vehicle From Address :</li>

<li class="inputfield" style="font-size:12px; font-weight:bold">

No 27 Jalan PJS 3/26<br>

Taman Buana <br>

4600 Petaling Jaya<br>

Selangor

</li>

</ul>

<ul class="forms" style="display:none">

<li class="txt">Drop Off Address</li>

<li class="inputfield" style="font-size:12px; font-weight:bold"><?=nl2br($data['booking\_drop'])?></li>

</ul>

<div class="clear"></div>

<ul class="forms">

<li class="txt">&nbsp;</li>

<li class="textfield"><input type="button" value="Print Details" onclick="window.print()" class="simplebtn"></li>

</ul>

</form>

</div>

</div>

<div class="col2">

<?php include\_once("includes/sidebar.php"); ?>

</div>

</div>

<?php include\_once("includes/footer.php"); ?>

**Code for book\_vehicle.php**

<?php

include\_once("includes/header.php");

if($\_SESSION['login'] != 1) {

header("Location:login.php?msg=Login first to book your vehicle !!!");

}

$R = $\_REQUEST;

if(isset($R[pick\_up])) {

$SQL="SELECT \* FROM vehicle, company where vehicle\_company = company\_id AND vehicle\_id NOT IN (SELECT booking\_vehicle\_id FROM booking WHERE booking\_from\_date BETWEEN '$R[pick\_up]' AND '$R[drop\_off]' OR booking\_to\_date BETWEEN '$R[pick\_up]' AND '$R[drop\_off]') ORDER BY vehicle\_id DESC";

$rs=mysql\_query($SQL) or die(mysql\_error());

}

global $SERVER\_PATH;

?>

<script>

jQuery(function() {

jQuery( "#pick\_up" ).datepicker({

changeMonth: true,

changeYear: true,

yearRange: "-0:+1",

dateFormat: 'yy-mm-dd'

});

jQuery( "#drop\_off" ).datepicker({

changeMonth: true,

changeYear: true,

yearRange: "-0:+1",

dateFormat: 'yy-mm-dd'

});

});

</script>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content\_sec">

<div class="col1">

<div class="contact">

<h4 class="heading colr">Book Your Vehicle</h4>

<form action="book\_vehicle.php" enctype="multipart/form-data" method="post" name="frm\_vehicle">

<ul class="forms">

<li class="txt">Pick Up Date</li>

<li class="inputfield"><input name="pick\_up" id="pick\_up" type="text" class="bar" required/></li>

</ul>

<ul class="forms">

<li class="txt">Drop Off Date</li>

<li class="inputfield"><input name="drop\_off" id="drop\_off" type="text" class="bar" required/></li>

</ul>

<div class="clear"></div>

<ul class="forms">

<li class="txt">&nbsp;</li>

<li class="textfield"><input type="submit" value="Search Vehicle" class="simplebtn"></li>

<li class="textfield"><input type="reset" value="Reset" class="resetbtn"></li>

</ul>

<input type="hidden" name="act" value="save\_vehicle">

<input type="hidden" name="vehicle\_id" value="<?=$data[vehicle\_id]?>">

</form>

</div>

<?php if($\_REQUEST['pick\_up']) { ?>

<div class="vehicletsecs" style="clear:both; width:100%">

<h4 class="heading colr">Choose Your Vehicle</h4>

<?php if(!mysql\_num\_rows()) { ?>

No Vehicle Found. Choose different date range !!!

<?php } ?>

<ul>

<?php

while($data = mysql\_fetch\_assoc($rs))

{

?>

<li>

<div class="thumb">

<a href="#"><img src="<?=$SERVER\_PATH.'uploads/'.$data[vehicle\_image]?>" alt="" style="width:92px; height:78px"/></a>

</div>

<div class="conts">

<a href="#" class="black bold"><?=$data[vehicle\_name]?></a>

<p>Company : <?=$data[company\_name]?></p>

<p class="bold">Seats : <?=$data[vehicle\_seats]?> Seaters</p>

<p class="bold">Price Per Day : <?=$data[vehicle\_price\_per\_day]?></p>

</div>

<div style="float:right; padding:20px 57px 20px 100px; border-left:1px solid #cccccc">

<a href="booking\_details.php?vehicle\_id=<?=$data[vehicle\_id]?>&pick\_up=<?=$\_REQUEST['pick\_up'];?>&drop\_off=<?=$\_REQUEST['drop\_off'];?>" class="simplebtn left">Book Your Vehicle Now</a>

</div>

</li>

<?php } ?>

</ul>

<div class="clear"></div>

</div>

<?php } ?>

</div>

<div class="col2">

<?php include\_once("includes/sidebar.php"); ?>

</div>

</div>

<?php include\_once("includes/footer.php"); ?>

**Code for User\_report.php**

<?php

include\_once("includes/header.php");

include\_once("includes/db\_connect.php");

$SQL="SELECT \* FROM user";

$rs=mysql\_query($SQL) or die(mysql\_error());

global $SERVER\_PATH;

?>

<script>

function delete\_user(user\_id)

{

if(confirm("Do you want to delete the user?"))

{

this.document.frm\_user.user\_id.value=user\_id;

this.document.frm\_user.act.value="delete\_user";

this.document.frm\_user.submit();

}

}

</script>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content\_sec">

<div class="col1" style="width:100%">

<div class="contact">

<h4 class="heading colr">User Reports</h4>

<form name="frm\_user" action="lib/user.php" method="post">

<div class="static">

<table style="width:100%">

<tbody>

<tr class="tablehead bold">

<td scope="col">Sr. No.</td>

<td scope="col">Image</td>

<td scope="col">Name</td>

<td scope="col">Mobile</td>

<td scope="col">Email</td>

<td scope="col">Date Of Birth</td>

<td scope="col">Action</td>

</tr>

<?php

$sr\_no=1;

while($data = mysql\_fetch\_assoc($rs))

{

?>

<tr>

<td style="text-align:center; font-weight:bold;"><?=$sr\_no++?></td>

<td><img src="<?=$SERVER\_PATH.'uploads/'.$data[user\_image]?>" style="heigh:50px; width:50px"></td>

<td><?=$data[user\_name]?></td>

<td><?=$data[user\_mobile]?></td>

<td><?=$data[user\_email]?></td>

<td><?=$data[user\_dob]?></td>

<td style="text-align:center"><a href="user.php?user\_id=<?php echo $data[user\_id] ?>">Edit</a> | <a href="Javascript:delete\_user(<?=$data[user\_id]?>)">Delete</a> </td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

<input type="hidden" name="act" />

<input type="hidden" name="user\_id" />

</form>

</div>

</div>

</div>

<?php include\_once("includes/footer.php"); ?>

**Code for user.php**

<?php

include\_once("includes/header.php");

if($\_REQUEST[user\_id])

{

$SQL="SELECT \* FROM user WHERE user\_id = $\_REQUEST[user\_id]";

$rs=mysql\_query($SQL) or die(mysql\_error());

$data=mysql\_fetch\_assoc($rs);

}

?>

<script>

jQuery(function() {

jQuery( "#user\_dob" ).datepicker({

changeMonth: true,

changeYear: true,

yearRange: "-65:-10",

dateFormat: 'd MM,yy'

});

jQuery('#frm\_user').validate({

rules: {

user\_confirm\_password: {

equalTo: '#user\_password'

}

}

});

});

</script>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content\_sec">

<div class="col1">

<div class="contact">

<h4 class="heading colr">User Registration</h4>

<?php

if($\_REQUEST['msg']) {

?>

<div class="msg"><?=$\_REQUEST['msg']?></div>

<?php

}

?>

<form action="lib/user.php" enctype="multipart/form-data" method="post" name="frm\_user">

<ul class="forms">

<li class="txt">Name</li>

<li class="inputfield"><input name="user\_name" type="text" class="bar" required value="<?=$data[user\_name]?>"/></li>

</ul>

<ul class="forms" id="user\_level">

<li class="txt">User Level</li>

<li class="inputfield">

<select name="user\_level\_id" id="user\_level\_id" class="bar" required/>

<?php echo get\_new\_optionlist("role","role\_id","role\_name",$data[user\_level\_id]); ?>

</select>

</li>

</ul>

<?php

if(!(isset($\_REQUEST['user\_id'])) || $\_REQUEST['user\_id'] == "") { ?>

<ul class="forms">

<li class="txt">Username</li>

<li class="inputfield"><input name="user\_username" type="text" class="bar" required value="<?=$data[user\_username]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">Password</li>

<li class="inputfield"><input name="user\_password" id="user\_password" type="password" class="bar" required value="<?=$data[user\_password]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">Confirm Password</li>

<li class="inputfield"><input name="user\_confirm\_password" id="user\_confirm\_password" type="password" class="bar" required value="<?=$data[user\_password]?>"/></li>

</ul>

<?php } ?>

<ul class="forms">

<li class="txt">Mobile</li>

<li class="inputfield"><input name="user\_mobile" type="text" class="bar" required value="<?=$data[user\_mobile]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">Email</li>

<li class="inputfield"><input name="user\_email" type="text" class="bar" required value="<?=$data[user\_email]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">Date of Birth</li>

<li class="inputfield"><input name="user\_dob" id="user\_dob" type="text" class="bar" required value="<?=$data[user\_dob]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">Address Line 1</li>

<li class="inputfield"><input name="user\_add1" type="text" class="bar" required value="<?=$data[user\_add1]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">Address Line 2</li>

<li class="inputfield"><input name="user\_add2" type="text" class="bar" required value="<?=$data[user\_add2]?>"/></li>

</ul>

<ul class="forms">

<li class="txt">City</li>

<li class="inputfield">

<select name="user\_city" class="bar" required/>

<?php echo get\_new\_optionlist("city","city\_id","city\_name",$data[user\_city]); ?>

</select>

</li>

</ul>

<ul class="forms">

<li class="txt">State</li>

<li class="inputfield">

<select name="user\_state" class="bar" required/>

<?php echo get\_new\_optionlist("state","state\_id","state\_name",$data[user\_state]); ?>

</select>

</li>

</ul>

<ul class="forms">

<li class="txt">Country</li>

<li class="inputfield">

<select name="user\_country" class="bar" required/>

<?php echo get\_new\_optionlist("country","country\_id","country\_name",$data[user\_country]); ?>

</select>

</li>

</ul>

<ul class="forms">

<li class="txt">Photo</li>

<li class="inputfield"><input name="user\_image" type="file" class="bar"/></li>

</ul>

<div class="clear"></div>

<ul class="forms">

<li class="txt">&nbsp;</li>

<li class="textfield"><input type="submit" value="Submit" class="simplebtn"></li>

<li class="textfield"><input type="reset" value="Reset" class="resetbtn"></li>

</ul>

<input type="hidden" name="act" value="save\_user">

<input type="hidden" name="avail\_image" value="<?=$data[user\_image]?>">

<input type="hidden" name="user\_id" value="<?=$data[user\_id]?>">

</form>

</div>

</div>

<div class="col2">

<?php include\_once("includes/sidebar.php"); ?>

</div>

</div>

<?php

if($\_SESSION['user\_details']['user\_level\_id'] != 1)

{

?>

<script>

jQuery( "#user\_level\_id" ).val(3);

jQuery( "#user\_level" ).hide();

</script>

<?php

}

?>

<?php include\_once("includes/footer.php"); ?>

**Code for login.php**

<?php

include\_once("includes/header.php");

if($\_REQUEST[vehicle\_id])

{

$SQL="SELECT \* FROM vehicle WHERE vehicle\_id = $\_REQUEST[vehicle\_id]";

$rs=mysql\_query($SQL) or die(mysql\_error());

$data=mysql\_fetch\_assoc($rs);

}

?>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content\_sec">

<div class="col1">

<div class="contact">

<h4 class="heading colr">Login To Your Account</h4>

<div class='msg'><?=$\_REQUEST['msg']?></div>

<form action="lib/login.php" method="post" name="frm\_vehicle">

<ul class="forms">

<li class="txt">Username</li>

<li class="inputfield"><input name="user\_user" type="text" class="bar" required /></li>

</ul>

<ul class="forms">

<li class="txt">Password</li>

<li class="inputfield"><input name="user\_password" type="password" class="bar" required /></li>

</ul>

<ul class="forms">

<li class="txt">&nbsp;</li>

<li class="inputfield"><a href='user.php' style="color:#790101; font-weight:bold;">Click here for register</a></li>

</ul>

<div class="clear"></div>

<ul class="forms">

<li class="txt">&nbsp;</li>

<li class="textfield"><input type="submit" value="Submit" class="simplebtn"></li>

<li class="textfield"><input type="reset" value="Reset" class="resetbtn"></li>

</ul>

<input type="hidden" name="act" value="check\_login">

</form>

</div>

</div>

<div class="col2">

<?php include\_once("includes/sidebar.php"); ?>

</div>

</div>

<?php include\_once("includes/footer.php"); ?>

**Code for change\_password.php:**

<?php include\_once("includes/header.php"); ?>

<div class="crumb">

</div>

<div class="clear"></div>

<div id="content\_sec">

<div class="col1">

<div class="contact">

<h4 class="heading colr">Change Your Account Password</h4>

<div class='msg'><?=$\_REQUEST['msg']?></div>

<form action="lib/login.php" method="post" name="frm\_vehicle">

<ul class="forms">

<li class="txt">New Password</li>

<li class="inputfield"><input name="user\_new\_password" type="password" class="bar" required /></li>

</ul>

<ul class="forms">

<li class="txt">Confirm Password</li>

<li class="inputfield"><input name="user\_confirm\_password" type="password" class="bar" required /></li>

</ul>

<div class="clear"></div>

<ul class="forms">

<li class="txt">&nbsp;</li>

<li class="textfield"><input type="submit" value="Change Password" class="simplebtn"></li>

<li class="textfield"><input type="reset" value="Reset" class="resetbtn"></li>

</ul>

<input type="hidden" name="act" value="change\_password">

</form>

</div>

</div>

<div class="col2">

<?php include\_once("includes/sidebar.php"); ?>

</div>

</div>

<?php include\_once("includes/footer.php"); ?>

**Future scope and further enhancement of the project:**

Today, the market place is flooded with several vehicle showroom options for shoppers to choose from. A variety of innovative products and services are being offered spoiling customers for choice. Online vehicle showroom system is no more a privilege enjoyed by your friends and family. Today, it is a reality in India. In the last couple of years, the growth of vehicle showroom system industry in India has been phenomenal as more shoppers have started discovering the benefits of using this platform. There is enough scope for online businesses in the future if they understand the Indian shoppers psyche and cater to their needs.

**Bibliography:**

* **Books Used:**
* Software Engineering - R.S. Pressman
* PHP For Dummies
* PHP Begineers Guide By McGrawhill Publication
* Javascript By McGrawhill Publication