

VEHICLE COMPUTERIZATION SYSTEM

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Science degree in Computer Science

By

L.ANITHA (Reg.No.39290009)

R.ANUSHIYA (Reg.No.39290011)



DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND

TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI-600 119, TAMIL NADU

MARCH - 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE SCHOOL OF COMPUTING.

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **L.ANITHA (Reg.No.39290009)** and **R.ANUSHIYA (Reg.No.39290011)** who carried out the project entitled "VEHICLE COMPUTERIZATION SYSTE" under my supervision from Decmber 2021 to march 2022.

Internal Guide

Dr. ASHA.

Head of the Department

Dr. LAKSHMANAN.M.E,Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

L.ANITHA (Reg.No.39290009) hereby declare that the Project Report entitled ” **VEHICLE
COMPUTERIZATION SYSTEM** “ done by me under the guidance of Dr./Prof./Mr./Ms.

Dr.M.SELVI is submitted in partial fulfillment of the requirements for the award of Bachelor of
Science degree in **COMPUTER SCIENCE**.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean, School of Computing** **Dr. L. Lakshmanan M.E., Ph.D. ,** and **Dr.S.Vigneshwari M.E., Ph.D. Heads** of the Department of Computer Science and Engineering for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr./Mr./Ms for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

VEHICLE COMPUTERIZATION SYSTEM

ABSTRACT

The real power of this project lies not in direct selling of products, but in the formation of stronger relationships with customers and delivering of a high level of service and support, which in turn improves organization sales and its goodwill. A service organization is a business entity that takes care of servicing a customer mechanism in the after sales domain.

As the number of customers and size of operations increases, the organization divides the geographical area into service areas and branch locations, to allow Engineers to be more responsive to the customer-needs. Vehicle computerizationSystem is web application developed in ASP.Net and it is implemented by using windows appliance. This application provides an easy and simple way to sell or purchase vehicles online.

Customers can view all the details of the vehicles and can purchase the vehicle. Customers can view all the details of the vehicle and send a request if he wants to purchase the vehicle. The main aim of developing this Vehicle computerizationSystem is to provide efficient automation service to the vehicle companies and vehicle servicing centers.

SL.NO	TITLE DESCRIPTION	PAGE NO
1.	INTRODUCTION	
	1.1 ORGANIZATION PROFILE	
	1.2 SYSTEM SPECIFICATION	
	1.2.1 HARDWARE CONFIGURATION	
	1.2.2 SOFTWARE SPECIFICATION	
	1.2.3 SOFTWARE DESCRIPTION	
2.	SYSTEM STUDY	
	2.1 EXISTING SYSTEM	
	2.1.1 DRAWBACKS	
	2.2 PROPOSED SYSTEM	
	2.2.1 FEATURES	
3.	SYSTEM DESIGN AND DEVELOPMENT	
	3.1 FILE DESIGN	
	3.2 INPUT DESIGN	
	3.3 OUTPUT DESIGN	
	3.4 DATABASE DESIGN	
	3.5 SYSTEM DEVELOPMENT	
	3.5.1 DESCRIPTION OF MODULES	
4.	TESTING AND IMPLEMENTATION	
	4.1 SYSTEM TESTING	
	4.2 SYSTEM IMPLEMENTATION	
5.	CONCLUSION	
6.	BIBLIOGRAPHY	
7.	APPENDICES	
	A. DATA FLOW DIAGRAM	
	B. TABLE STRUCTURE	

	C. SAMPLE INPUT	
	D. SAMPLE OUTPUT	

1.1 ABOUT THE PROJECT

Vehicle computerization 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. 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 Card.

The main aim of this project is to build a portal that compares the features and characteristics of different vehicles. In this fast moving world vehicle plays a very powerful role in the field of transportation. It is a symbol of sophistication. It is a part of luxurious life. Especially working class people and businessmen always prefer to have a vehicle. For some people it is a long term goal to buy a vehicle. It is a long lasting product and people can't effort to change it every year.

So it is always a best practice to buy the best vehicle as per our comfort and requirement. If you are dreaming about buy a new vehicle, it is very important to know your expectation from your dream vehicle. This Vehicle computerization system is fully customized desktop application where company staff can view each customer order and give a solution to those vehicle problems.

There are many factors to decide on while purchasing the vehicle like cost, fuel efficiency appearance, performance, safety, comfort, reliability, mileage, resale value etc. As the priority varies for different individuals, it is necessary to organize the list in terms of how important the criteria are to you. It is very important to have a clear picture of the vehicle that you wish to buy. Without clear idea if you go to the dealer you would most likely get the automobile the dealer could not sell. That would not be a dream vehicle instead it would be a nightmare for you.

1.2 SYSTEM SPECIFICATION

1.2.1 HARDWARE SPECIFICATION

Processor : Pentium Core 2 Duo

RAM : 2 GB DDR RAM

Hard Disk : 500 GB

Key Board : Standard IOS Keys

Monitor : 14' color monitor

Mouse : 3 Button mouse

1.2.2 SOFTWARE SPECIFICATION

Operating System : MS-Windows

XP/7. Front End : PHP

Back End : MYSQL

1.2.3 SOFTWARE DESCRIPTION

WHAT IS PHP?

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

PHP is a recursive acronym for "PHP: Hypertext Preprocessor".

PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.

It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.

PHP is forgiving: PHP language tries to be as forgiving as possible.

PHP Syntax is C-Like.

COMMON USES OF PHP

PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.

PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.

You add, delete, and modify elements within your database through PHP. Access cookies variables and set cookies.

Using PHP, you can restrict users to access some pages of your website. It can encrypt data.

CHARACTERISTICS OF PHP

Five important characteristics make PHP's practical nature possible –

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

Web Server – PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here

– <http://httpd.apache.org/download.cgi>

Database – PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.

PHP Parser – In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

PHP PARSER INSTALLATION

Before you proceed it is important to make sure that you have proper environment setup on your machine to develop your web programs using PHP.

Type the following address into your browser's address box. If this displays a page showing your PHP installation related information then it means you have PHP and Web server installed properly. Otherwise you have to follow given procedure to install PHP on your computer.

This section will guide you to install and configure PHP over the following four platforms –

[PHP Installation on Linux or Unix with Apache](#)

[PHP Installation on Mac OS X with Apache](#)

[PHP Installation on Windows NT/2000/XP with IIS](#)

PHP Installation on Windows NT/2000/XP with Apache

Apache Configuration

If you are using Apache as a Web Server then this section will guide you to edit Apache Configuration Files.

PHP.INI File Configuration

The PHP configuration file, php.ini, is the final and most immediate way to affect PHP's functionality.

Windows IIS Configuration

To configure IIS on your Windows machine you can refer your IIS Reference Manual shipped along with IIS.

Escaping to PHP

The PHP parsing engine needs a way to differentiate PHP code from other elements in the page. The mechanism for doing so is known as 'escaping to PHP'. There are four ways to do this –

Canonical PHP tags

The most universally effective PHP tag style is `<?P>`. If you use this style, you can be positive that your tags will always be correctly interpreted.

Short tags are, as one might expect, the shortest option. You must do one of two things to enable PHP to recognize the tags –

Choose the `--enable-short-tags` configuration option when you're building PHP. Set the `short_open_tag` setting in your php.ini file to on. This option must be disabled to parse XML with PHP because the same syntax is used for XML tags.

ASP-style tags

ASP-style tags mimic the tags used by Active Server Pages to delineate code blocks. To use ASP-style tags, you will need to set the configuration option in your php.ini file.

HTML script tags

Commenting PHP Code

A comment is the portion of a program that exists only for the human reader and stripped out before displaying the programs result. There are two commenting formats in PHP –

Single-line comments – They are generally used for short explanations or notes relevant to the local code. Here are the examples of single line comments.

Multi-lines comments – They are generally used to provide pseudocode algorithms and more detailed explanations when necessary. The multiline style of commenting is the same as in C.

Frame Work is collection of software or program, that trigger off easy coding and implementing the code. It helps to programmer to achieve goals in short period of time. If PHP code is integrated with frame works, you can do anything with php coding skills.

Some of frame works:

FuelPHP

Fuel PHP works based on Model View Control and having innovative plug ins. FuelPHP supports router based theory where you might route directly to a nearer the input Uri, making the closure the controller and giving it control of further execution.

CakePHP

Cake PHP is a great source to build up simple and great web application in an easy way. Some great features which are inbuilt in php are input validation, SQL injection prevention that keeps you application safe and secure.

FEATURES

- Build Quickly
- No need to configure
- MIT license
- MVC Model
- Secure

MYSQL DEFINITION

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational Database Management System (RDBMS)** is a software that:

Enables you to implement a database with tables, columns and indexes.

Guarantees the Referential Integrity between rows of various tables.

Updates the indexes automatically.

Interprets an SQL query and combines information from various tables.

RDBMS TERMINOLOGY:

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

Database: A database is a collection of tables, with related data.

Table: A table is a matrix with data. A table in a database looks like a simple spreadsheet.

Column: One column (data element) contains data of one and the same kind, for example the column postcode.

Row: A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.

Redundancy: Storing data twice, redundantly to make the system faster.

Primary Key: A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.

Foreign Key: A foreign key is the linking pin between two tables.

Compound Key: A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.

Index: An index in a database resembles an index at the back of a book.

Referential Integrity: Referential Integrity makes sure that a foreign key value always points to an existing row.

MYSQL DATABASE:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It is developed, marketed, and supported by MySQL AB, which is a SweHall company. MySQL is becoming so popular because of many good reasons:

MySQL is released under an open-source license. So you have nothing to pay to use it.

MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

MySQL works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development.

MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

MySQL works very well in combination of various programming languages like PERL, C, C++, JAVA and PHP. Out of these languages, PHP is the most popular one because of its web application development capabilities.

This tutorial focuses heavily on using MySQL in a PHP environment. If you are interested in MySQL with PERL, then you can look into [PERL and MySQL Tutorial](#).

PHP provides various functions to access MySQL database and to manipulate data records inside MySQL database. You would require to call PHP functions in the same way you call any other PHP function.

When writing PHP applications that need to connect to the MySQL server there are several API options available. This document discusses what is available and how to select the best solution for your application.

2.1 EXISTING SYSTEM

In the existing Vehicle system, all the work is done manually which takes a lot of time and effort. It is very difficult to users to view the details of the vehicle and to purchase the vehicle. The existing system only provides text-based interface, which is not as user-friendly as Graphical user Interface. Since the system is implemented in Manual, so the response is very slow. The transactions are executed in off-line mode, hence on-line data capture and modification is not possible. Off-line reports cannot be generated due to batch mode execution

2.1.1 DRAWBACKS

- It is difficult to maintain important information in books.

- More manual hours need to generate required reports.

- It is tedious to manage historical data which needs much space to keep all the previous years' ledgers, books etc.

- Daily sales and purchases details must be entered into books are very difficult to maintain.

2.2 PROPOSED SYSTEM

The proposed Vehicle system provides an easy way to find the information of the vehicle. This is an automated system which enables users to view and purchase a vehicle from anywhere and at any time. All the data will be stored in the database in which there will have not the scope of losing the stored information. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of online sales and purchases in the databases.

2.2.1FEATURES

This system will help the user to easily find out best vehicle.
It helps user to analyze the vehicles effectively and user can make best decision before buying.

User can check out varieties of features and can compare it.
The system has an additional advantage of providing and suggesting the best accessories available for the respective vehicle.

User can select vehicle based on his preference.It is very easy to use.
This online vehicle comparison system helps in back office administration by streamlining and standardizing the procedures.

It saves a lot of time, money and labour
Automatic update to database once reservation is made or new customer registered: Whenever there's new reservation or new registration, the system should be able update the database without any additional efforts from the admin

FEASIBILITY STUDY

Feasibility study is conducted once the problem is clearly understood. Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.

The system has been tested for feasibility in the following points.

1. Technical Feasibility

2. Economic Feasibility

3. Operational Feasibility.

1. TECHNICAL FEASIBILITY

The project entitles "Vehicle computerizationsystem" is technically feasibility because of the below mentioned feature. The project was developed in Java which Graphical User Interface. It provides the high level of reliability, availability and compatibility. All these make ASP.NET an appropriate language for this project. Thus the existing software Java is a powerful language.

2. ECONOMIC FEASIBILITY

The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90% approximately. The costs incurred of not creating the system are set to be great, because precious time can be wanted by manually.

3. OPERATIONAL FEASIBILITY

In this project, the automation will know the details of each project where he may be presented and the data will be maintained as decentralized and if any inquires for that particular contract can be known as per their requirements and necessities.

3.SYSTEM DESIGN AND DEVELOPMENT

3.1 FILE DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made.

These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

3.2 INPUT DESIGN

Input design process is to design the various inputs needed into a machine-oriented format. The main objective is to create an input layout that is easy to follow and to avoid operator errors. In the system design phase, the expanded data flow diagram identifies logical data flows, data stores, sources and destination. A system flowchart specifies the flow of the program. Input data are collected and organized into groups of similar data. Once identified, appropriate input media are selected for processing. Verify in appendix

- A source document differs from a turnaround document in that the former contains data that change the status of a resource while the latter is a machine readable document.
- Transaction throughput is the number of error-free transactions entered during a specified time period.

- A document should be concise because longer documents contain more data and so take longer to enter and have a greater chance of data entry errors.

Numeric coding substitute's numbers for character data mnemonic coding represents data in a form that is easier for the user to understand and remember.

The input designs which is follows are

- Vehicle
- Customer
- Vehicle Reservation
- Testimonials
- Bookings
- Pages
- User Creation
- Service Charges
- Submission
- Sales Manager

3.3 OUTPUT DESIGN

The output design is another important aspects, which designs the screen separately for each purpose. Verify in the appendix. This separate designing of the screen helps the data entry easy and more accurate and also avoid confusion and errors. The purpose of outputs has been understood and the efficiency of information contained should be analyzed and conformed.

These guidelines apply for the most part to both paper and screen outputs. Output design is often discussed before other aspects of design because, from the client's point of view, the output is the system. Output is what the client is buying when he or she pays for a development project. Inputs, databases, and processes exist to provide output.

The system provides the Assured Purposeful Output and it meaningful to user. And also it provides appropriate quantity and appropriate distribution, assured timeliness

The system is designed with output to serve the intended purpose; it does not produce unwanted at all. The Design output is to fit the users. It meets users' requirements as much as possible. The system delivers the appropriate quantity of

output. Large amounts of output do not always guarantee productiveness. It assures that the output is where it is needed. Here

output should be directed to the right person only. The system provides the output on time. Her output should arrive where it's needed on time for making decisions. And here the output will be in an appropriate format for users. . The output designs which is follows are

Vehicle report
Customer report
Vehicles report
Booking report

3.4 DATABASE DESIGN

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database automation system (DBMS).The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- ❖ Determine the relationships between the different data elements.
- ❖ Superimpose a logical structure upon the data on the basis of these relationships

THE DESIGN PROCESS

1. Determine the purpose of your database - This helps prepare you for the remaining steps.
2. Find and organize the information required - Gather all of the types of information you might want to record in the database, such as product name and order number.
3. Divide the information into tables - Divide your information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.
4. Turn information items into columns - Decide what information you want to store in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.
5. Specify primary keys - Choose each table's primary key. The primary key is a column that is used to uniquely identify each row. An example might be Product ID or Order ID.

3.5 SYSTEM DEVELOPEMENT

3.5.1 DESCRIPTION OF MODULES

- Admin
- Vehicle
- Customer
- Vehicle Reservation
- Testimonials
- Bookings
- Pages
- User Creation
- Service Charges
- Submission
- Sales Manager
- Report

ADMIN

Admin can get log in with valid username and password. Admin can view all registered dealer details. Admin can upload vehicle information like Vehicle name, Manufacture Year, Model Color, Cost

DEALER

The dealer needs to register with the application to get unique username and password. The dealer can enter into the application with valid username and password. The dealer can upload vehicle information such as vehicle name, manufacture year, model color and cost.

VEHICLE

A vehicle needs to register with the application to get unique username and password. The vehicle can get log in with valid username and password. The vehicle can view vehicle information like Vehicle name, Manufacture Year, Model Color, Cost.

CUSTOMER

Customer needs to register with the application to get unique username and password. The customer can get log in with valid username and password. Customer views Vehicles and sends a request to sales manager for cost and delivery date. The customer can view sales manager response

VEHICLE RESERVATION

This tool through which customers can reserve available cars prior to their expected pick-up date or time.

CUSTOMER'S REGISTRATION

A registration portal to hold customer's details, monitor their transaction and used same to offer better and improve services to them.

VEHICLE POST

In this module used to post vehicles on this site by customers or subscribers. In this module add the menus like vehicle title, vehicle overview, select brand, full type, seat type, model year and etc.

BOOKINGS

In this module used manages the all operations of bookings. Admin can only manage the bookings category. In this form contains the menus like name, email, testimonials, and postdate etc.

TESTIMONIALS

In this module used manages the all operations of testimonials. Admin can only manage this category. In this form contains the menus like name, date, address, from date and to date and posting date etc.

PAGES

In this module used to manage the vehicle information's. it manages the operation on vehicle's terms and condition, about vehicles, and FAQ

USER CREATION

In this module we are Registration the username, password, phone and vehicle no, warranty and branch no' are user creations.

SERVICE CHARGES

In this module we are Registered the Customer Name, Vehicle no, Parts and Servicing Charges as well as paying the money.

SUBMISSION

In this module the dealer is handover to the Customer and before registering the Customer name, Vehicle no and phone.

SALES MANAGER

Sales manager need to register with the application to get unique username and password. A sales manager can get log in with valid username and password. Sales Manager can view customer requests and send vehicle cost and delivery date.

REPORTS

This module takes care of report generation for various transactions carried out in the system. This also helps with a detailed report on the current information of various details.

- Vehicle report
- Customer report
- Cars report
- Booking report

4. TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team. System testing is important because of the following reasons: System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole. The application is tested thoroughly to verify that it meets the functional and technical specifications. The application is tested in an environment that is very close to the production environment where the application will be deployed. System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

UNIT TESTING

Unit testing verification efforts on the smallest unit of software design, module. This is known as “Module Testing”. The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be working satisfactorily as regard to the expected output from the module.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

REGRESSION TESTING

Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation. The intent of regression testing is to ensure that a change,

such as a bug fix should not result in another fault being uncovered in the application.

Regression testing is important because of the following reasons: Minimize the gaps in testing when an application with changes made has to be tested. Testing the new changes

to verify that the changes made did not affect any other area of the application. Mitigates risks when regression testing is performed on the application. Test coverage is increased without compromising timelines. Increase speed to market the product.

ACCEPTANCE TESTING

This testing is done to verify the readiness of the system for the implementation. Acceptance testing begins when the system is complete. Its purpose is to provide the end user with the confidence that the system is ready for use. It involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

Tools to special importance during acceptance testing include:

Test coverage Analyzer – records the control paths followed for each test case.

Timing Analyzer – also called a profiler, reports the time spent in various regions of the code are areas to concentrate on to improve system performance.

Coding standards – static analyzers and standard checkers are used to inspect code for deviations from standards and guidelines.

ALPHA TESTING

This test is the first stage of testing and will be performed amongst the teams (developer and QA teams). Unit testing, integration testing and system testing when combined together are known as alpha testing. During this phase, the following aspects will be tested in the application: Spelling Mistakes, Broken Links, Cloudy Directions The Application will be tested on machines with the lowest specification to test loading times and any latency problems.

BETA TESTING

This test is performed after alpha testing has been successfully performed. In beta testing, a sample of the intended audience tests the application. Beta testing is also known as pre-release testing.

Beta test versions of software are ideally distributed to a wide audience on the Web, partly to give the program a "real-world" test and partly to provide a preview of the next release. In this phase, the audience will be testing the following: Users will install, run the application and send their feedback to the project team.

Typographical errors, confusing application flow, and even crashes. Getting the feedback, the project team can fix the problems before releasing the software to the actual users. The more issues you fix that solve real user problems, the higher the quality of your application will be. Having a higher-quality application when you release it to the general public will increase customer satisfaction.

4.2 SYSTEM IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification.

It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation. The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

MAINTENANCE AND ENVIRONMENT

As the number of computer based systems, grievance libraries of computer software began to expand. In house developed projects produced tones of thousand soft program source statements. Software products purchased from the outside added hundreds of thousands of new statements. A dark cloud appeared on the horizon. All of these programs, all of those source statements-had to be corrected when false were detected, modified as user requirements changed, or adapted to new hardware that was purchased. These activities were collectively called software Maintenance.

The maintenance phase focuses on change that is associated with error correction, adaptations required as the software's environment evolves, and changes due to enhancements brought about by changing customer requirements. Four types of changes are encountered during the maintenance phase.

- Correction
- Adaptation
- Enhancement
- Prevention

CORRECTION:

Even with the best quality assurance activities is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects.

Maintenance is a set of software Engineering activities that occur after software has been delivered to the customer and put into operation. Software configuration automation is a set of tracking and control activities that began when a software project begins and terminates only when the software is taken out of the operation.

We may define maintenance by describing four activities that are undertaken after a program is released for use:

- Corrective Maintenance
- Adaptive Maintenance

- Perfective Maintenance or Enhancement

ADAPTATION

Over time, the original environment (E>G., CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate change to its external environment.

ENHANCEMENT

As software is used, the customer/user will recognize additional functions that will provide benefit. Perceptive maintenance extends the software beyond its original function requirements.

PREVENTION

Computer software deteriorates due to change, and because of this, preventive maintenance, often called software re-engineering, and must be conducted to enable the software to serve the needs of its end users. In essence, preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced.

5. CONCLUSION

The project “VEHICLE COMPUTERIZATIONSYSTEM” has been developed in ASP.NET as front end and MS-SQL as back end. It is user friendly and provides necessary information. Necessary information needed for later reference is also stored. Various reports as needed by the company are generated.

A person who has little knowledge about computer, about Visual Basic and production field also can use this software efficiently. Though enhancement can be done to this software. The software developed is well secured with user password and this doesn't allow anyone to enter into the software.

This software can be operated easily. Since the success of the project depends largely on how accurately the problem is defined, thoroughly investigated and properly carried out through the choice of solutions, the existing system is analyzed in detail.

SCOPE FOR FUTURE ENHANCEMENT

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

With the proposed software of database Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner. The following are the future scope for the project.

6. BIBLIOGRAPHY

BOOK REFERENCES

- Domain-Driven Design in by Carlos Buenosvinos , Christian Soronellas , **Keyvan Akbary** Publisher: Picket Published: June 2017
- **ASP.NET** Objects, Patterns, and Practice, Fifth Edition by Matt Zandstra Publisher: Apress Published: December 2016
- Doyle, Matt. Beginning ASP.NET 5.3 (Wrox Programmer to Programmer), 2009.
- Lerdorf, Rasmus. ASP.NET Pocket Reference, O'Reilly, 2000.
- Nixon, Robin. Learning ASP.NET, MS-SQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites, O'Reilly, 2012.
- Nixon, Robin. Robin Nixon's ASP.NET Crash Course: Learn ASP.NET in 14 easy lectures, O'Reilly, 2012.
- Ullman, Larry. ASP.NET Advanced and Object-Oriented Programming: Visual QuickPro Guide (3rd Edition), 2012.
- Ullman, Larry. ASP.NET for the Web: Visual QuickStart Guide, 2011.
- Welling, Luke and Thomson, Laura. ASP.NET and MS-SQL Web Development, Third Edition, Sams, 2008.

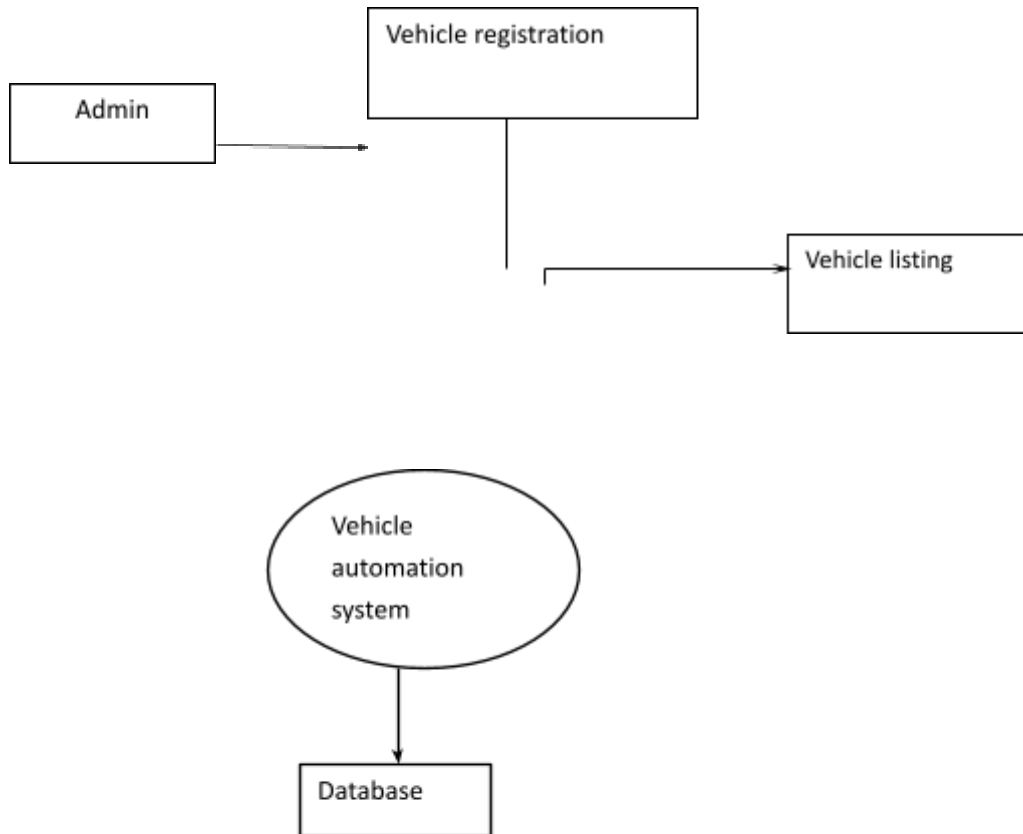
WEBSITE REFERENCES

- <https://www.w3schools.com/ASP.NET/default.asp>
- <https://www.codecademy.com/en/tracks/ASP.NET>
- <https://www.codeschool.com/learn/ASP.NET>
- <https://www.sololearn.com/Course/ASP.NET/>
- <https://www.guru99.com/ASP.NET-tutorials.ASP.NET>

7. APPENDICES

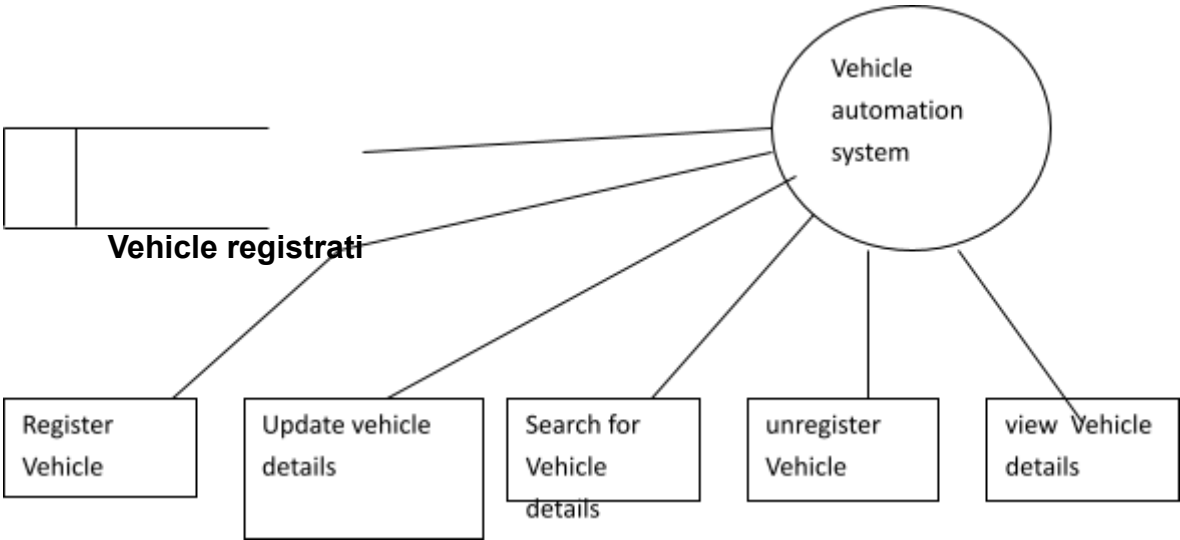
A.DATA FLOW DIAGRAM

LEVEL 0



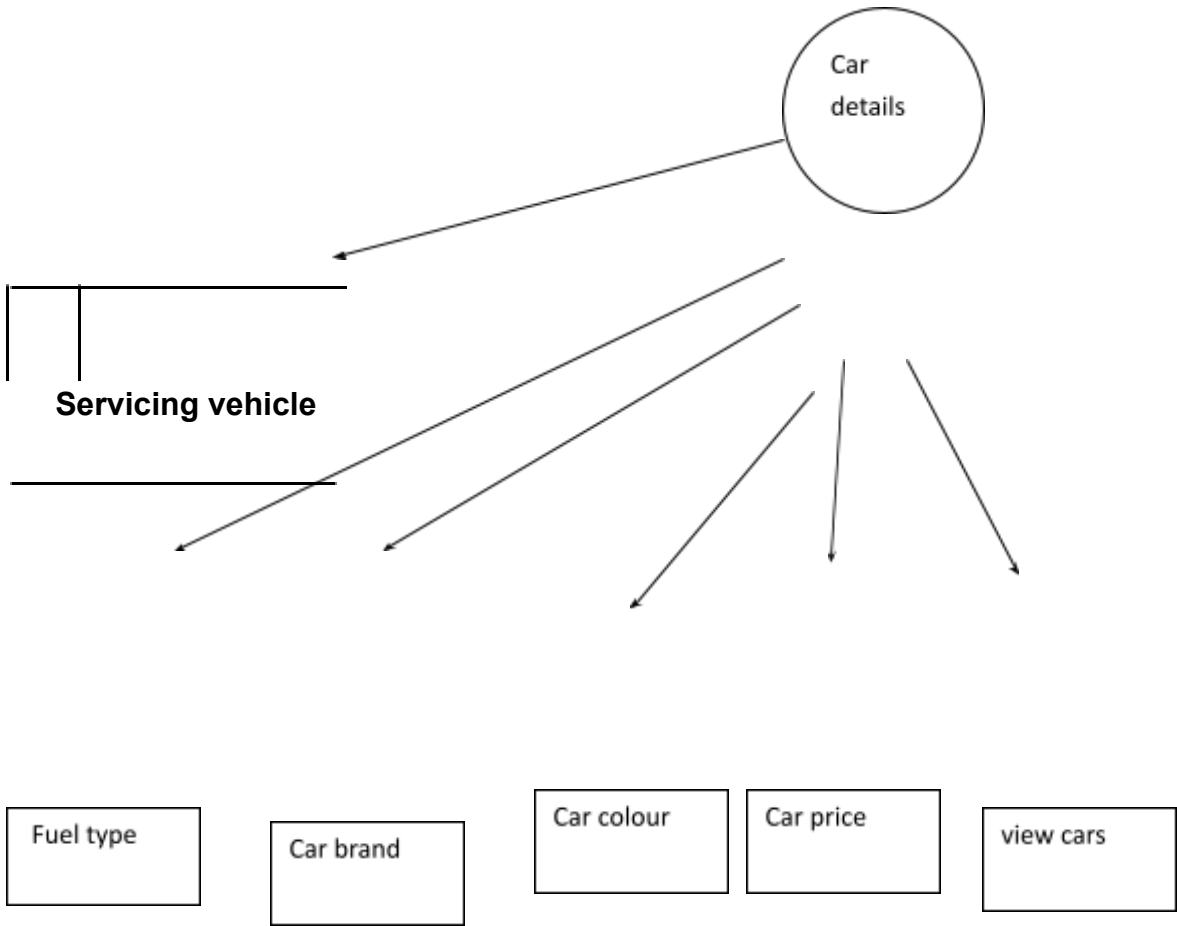
LEVEL 1

Vehicle registration



LEVEL 2

`Booking vehicle



B. TABLE STRUCTURE

TABLE: Admin

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPTION
1	id	int	11	Primary key	id
2	UserName	varchar	20	Not Null	User Name
3	Password	varchar	20	Not Null	Password
4	updatetime	Date/Time	-	Not Null	update Date
5	currpassword	varchar	20	Not null	Current password
6	Npassword	Var char	20	Not null	New password
7	Con password	varchar	20	Notnull	Conform password

TABLE: tblbooking

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPTION
1	id	int	11	Primary key	id
2	userEmail	varchar	30	Not Null	User Email
3	VehicleId	int	11	Not Null	Vehicle Id
4	FromDate	Date/Time	-	Not Null	From Date
5	ToDate	Date/Time	-	Not Null	To Date
6	message	varchar	500	Not Null	message

7	Status	varchar	500	Not Null	Status
8	PostingDate	Date/Time	-	Not Null	Posting Date

TABLE: tblbrands

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPTION
1	id	int	11	Primary key	id
2	BrandName	varchar	20	Not Null	Brand Name
3	CreationDate	Date/Time	-	Not Null	Creation Date
4	UpdationDate	Date/Time	-	Not Null	Updation Date

TABLE: tblcontactusinfo

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPTION
1	id	int	11	Primary key	id
2	Address	varchar	50	Not Null	Address

3	EmailId	varchar	20	Not Null	Email Id
4	ContactNo	int	10	Not Null	Contact Number

TABLE: tblcontactusquery

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPTION
1	id	int	11	Primary key	id
2	name	varchar	30	Not Null	name
3	EmailId	varchar	11	Not Null	Email Id
4	ContactNumber	int	10	Not Null	Contact Number
5	Message	varchar	500	Not Null	Message
6	PostingDate	Date/Time	-	Not Null	Posting Date
7	status	varchar	500	Not Null	status

TABLE: tblpages

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPTION
1	id	int	11	Primary key	id

2	PageName	varchar	50	Not Null	Page Name
3	type	varchar	20	Not Null	type
4	detail	varchar	1000	Not Null	detail

TABLE: tbltestimonial

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPRTION
1	id	int	11	Primary key	id
2	UserEmail	varchar	30	Not Null	User Email
3	Testimonial	varchar	11	Not Null	Testimonial
4	PostingDate	Date/Time	10	Not Null	Posting Date
5	status	varchar	500	Not Null	status

TABLE: tblcontactusquery

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPRTION
1	id	int	11	Primary key	id
2	FullName	varchar	20	Not Null	Full Name
3	EmailId	varchar	30	Not Null	Email Id
4	Password	varchar	20	Not Null	Password
5	ContactNo	int	10	Not Null	Contact Number
6	dob	varchar	10	Not Null	Date of birth
7	Address	varchar	50	Not Null	Address
8	City	varchar	20	Not Null	City

9	Country	varchar	10	Not Null	Country
10	RegDate	Date/Time	-	Not Null	Register Date
11	UpdationDate	Date/Time	-	Not Null	Updation Date

TABLE: tblvehiclequery

PRIMARY KEY: id

SNO	FIELD NAME	DATA TYPE	SIZE	CONSTRAINT	DESCRIPRTION
1	Id	int	11	Primary key	id
2	VehiclesTitle	varchar	25	Not Null	Vehicles Title
3	VehiclesBrand	varchar	25	Not Null	Vehicles Brand
4	VehiclesOverview	varchar	150	Not Null	Vehicles Overview
5	PricePerDay	int	5	Not Null	Price Per Day

6	FuelType	varchar	10	Not Null	Fuel Type
7	ModelYear	date	4	Not Null	Model Year

C. SAMPLE SCREENS

```
<?php
session_start
();

$_SESSION = array();

if (ini_get("session.use_cookies")) {

$params =
session_get_cookie_params();
setcookie(session_name(), "", time() -
60*60,

$params["path"], $params["domain"],

$params["secure"], $params["httponly"]

);

}

unset($_SESSION['login']);
session_destroy(); // destroy
session
header("location:index.php");

?>
```

```
<?php
session_start();
error_reporting(0)
;

include('includes/config.php');
if(strlen($_SESSION['alogin'])==
0)
{

header('location:index.php');

}

else{

// Code for change
password
if(isset($_POST['submit'])
)
```

```
{
```

```
$password=md5($_POST['password']);
```

```
$newpassword=md5($_POST['newpassword']);
```

```
$username=$_SESSION['alogin'];
```

```
$sql ="SELECT Password FROM admin WHERE UserName=:username and  
Password=:password";
```

```
$query= $dbh ->prepare($sql);
```

```
$query->bindParam(':username', $username, PDO::PARAM_STR);
```

```
$query->bindParam(':password', $password, PDO::PARAM_STR);
```

```
$query->execute();
```

```
$results = $query
```

```
->fetchAll(PDO::FETCH_OBJ); if($query
```

```
->rowCount() > 0)
```

```
{
```

```
$con="update admin set Password=:newpassword where UserName=:username";
```

```
$chngpwd1 = $dbh->prepare($con);
```

```
$chngpwd1->bindParam(':username', $username, PDO::PARAM_STR);
```

```
$chngpwd1->bindParam(':newpassword', $newpassword, PDO::PARAM_STR);
```

```
$chngpwd1->execute();
```

```
$msg="Your Password succesfully changed";
```

```
}
```

```
else {
```

```
$error="Your current password is not valid.";
```

```
}
```

```
}
```

```
?>
```

```
<!doctype html>
```

```
<html lang="en" class="no-js">
```

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1,
minimum-scale=1, maximum-scale=1">

<meta name="description" content="">

<meta name="author" content="">

<meta name="theme-color" content="#3e454c">

<title>Vehicle computerizationSystem | Admin Change Password</title>

<!-- Font awesome -->

<link rel="stylesheet" href="css/font-awesome.min.css">

<!-- Sandstone Bootstrap CSS -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<!-- Bootstrap Datatables -->

<link rel="stylesheet" href="css/dataTables.bootstrap.min.css">

<!-- Bootstrap social button library -->

```
<link rel="stylesheet" href="css/bootstrap-social.css">
```

```
<!-- Bootstrap select -->
```

```
<link rel="stylesheet" href="css/bootstrap-select.css">
```

```
<!-- Bootstrap file input -->
```

```
<link rel="stylesheet" href="css/fileinput.min.css">
```

```
<!-- Awesome Bootstrap checkbox -->
```

```
<link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">
```

```
<!-- Admin Style -->
```

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

```
<script
```

```
type="text/javascript">
```

```
function valid()
```

```
{

if(document.chngpwd.newpassword.value!= document.chngpwd.confirmpassword.value)

{

alert("New Password and Confirm Password Field do not match !!");
document.chngpwd.confirmpassword.focus();

return false;

}

return true;

}

</script>

<style>

.errorWrap {
padding: 10px;
margin: 0 0 20px
0; background:
#fff;

border-left: 4px solid #dd3d36;

-webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
}
```

```
.succWrap{
```

```
padding: 10px;
```

```
margin: 0 0 20px
```

```
0; background:
```

```
#fff;
```

```
border-left: 4px solid #5cb85c;
```

```
-webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<?php include('includes/header.php');?>
```

```
<div class="ts-main-content">
```

```
<?php include('includes/leftbar.php');?>
```

```
<div class="content-wrapper">
```

```
<div class="container-fluid">
```

```
<div class="row">
```

```
<div class="col-md-12">
```

```
<h2 class="page-title">Change Password</h2>
```

```
<div class="row">
```

```
<div class="col-md-10">
```

```
<div class="panel panel-default">
```

```
<div class="panel-heading">Form fields</div>
```

```
<div class="panel-body">
```

```
<form method="post" name="chngpwd" class="form-horizontal"  
onSubmit="return valid();">
```

```
<?php if($error){?><div  
class="errorWrap"><strong>ERROR</strong>:<?php echo  
htmlentities($error); ?></div><?php }
```

```
else if($msg){?><div class="succWrap"><strong>SUCCESS</strong>:<?php  
echo htmlentities($msg); ?></div><?php }?>
```

```
<div class="form-group">
```

```
<label class="col-sm-4 control-label">Current Password</label>
```

```
<div class="col-sm-8">
```

```
<input type="password" class="form-control" name="password" id="password" required>
```

</div>

</div>

<div class="hr-dashed"></div>

<div class="form-group">

<label class="col-sm-4 control-label">New Password</label>

<div class="col-sm-8">

<input type="password" class="form-control" name="newpassword"
id="newpassword" required>

</div>

</div>

<div class="hr-dashed"></div>

<div class="form-group">

<label class="col-sm-4 control-label">Confirm Password</label>

<div class="col-sm-8">

<input type="password" class="form-control"
name="confirmpassword" id="confirmpassword" required>

</div>

</div>

<div class="hr-dashed"></div>

<div class="form-group">

<div class="col-sm-8 col-sm-offset-4">

<button class="btn btn-primary" name="submit" type="submit">Save changes</button>

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Loading Scripts -->

<scriptsrc="js/jquery.min.js"></script>

<scriptsrc="js/bootstrap-select.min.js"></script>

<scriptsrc="js/bootstrap.min.js"></script>

<scriptsrc="js/jquery.dataTables.min.js"></script>

<scriptsrc="js/dataTables.bootstrap.min.js"></script>

<scriptsrc="js/Chart.min.js"></script>

<scriptsrc="js/fileinput.js"></script>

```
<scriptsrc="js/chartData.js"></script>
```

```
<scriptsrc="js/main.js"></script>
```

```
</body>
```

```
</html>
```

```
<?php } ?><?php
```

```
session_start();
```

```
error_reporting(0);
```

```
include('includes/config.php
```

```
');
```

```
if(strlen($_SESSION['alogin'])==0)
```

```
{
```

```
header('location:index.php');
```

```
}
```

```
else{
```

```
if(isset($_GET['del']))
```

```
{
```

```
$id=$_GET['del'];
```

```
$sql = "delete from tblbrands WHERE id=:id";
```

```
$query = $dbh->prepare($sql);
```

```
$query ->bindParam(':id',$id, PDO::PARAM_STR);
```

```
$query ->execute();
```

```
$msg="Page data updated successfully";
```

```
}
```

```
?>
```

```
<!doctype html>
```

```
<html lang="en" class="no-js">
```

```
<head>
```

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">

<meta name="description" content="">

<meta name="author" content="">

href="css/bootstrap.min.css">

<!-- Bootstrap Datatables -->

<link rel="stylesheet" href="css/dataTables.bootstrap.min.css">

<!-- Bootstrap social button library -->

<link rel="stylesheet" href="css/bootstrap-social.css">

<!-- Bootstrap select -->

```
<link rel="stylesheet" href="css/bootstrap-select.css">
```

```
<!-- Bootstrap file input -->
```

```
<link rel="stylesheet" href="css/fileinput.min.css">
```

```
<!-- Awesome Bootstrap checkbox -->
```

```
<link rel="stylesheet" href="css/awesome-bootstrap-checkbox.css">
```

```
<!-- Admin Style -->
```

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

```
<style>
```

```
.errorWrap {
```

```
padding: 10px;
```

```
margin: 0 0 20px
```

```
0; background:
```

```
#fff;
```

```
border-left: 4px solid #dd3d36;
```

```
-webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
}
```

```
.succWrap{
```

```
padding: 10px;
```

```
margin: 0 0 20px
```

```
0; background:
```

```
#fff;
```

```
border-left: 4px solid #5cb85c;
```

```
-webkit-box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
box-shadow: 0 1px 1px 0 rgba(0,0,0,.1);
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<?php include('includes/header.php');?>
```

```
<div class="ts-main-content">
```

```
<?php include('includes/leftbar.php');?>
```

```
<div class="content-wrapper">
```

```
<div class="container-fluid">
```

```
<div class="row">
```

```
<div class="col-md-12">
```

```
<h2 class="page-title">Registered Users</h2>
```

```
<!-- Zero Configuration Table -->
```

```
<div class="panel panel-default">
```

```
<div class="panel-heading">Reg Users</div>
```

```
<div class="panel-body">
```

```
<?php if($error){?><div
```

```
class="errorWrap"><strong>ERROR</strong>:<?php echo  
htmlentities($error); ?></div><?php }
```

```
else if($msg){?><div class="succWrap"><strong>SUCCESS</strong>:<?php  
echo htmlentities($msg); ?></div><?php }?>
```

```
<table id="zctb" class="display table table-striped table-bordered  
table-hover" cellpadding="0" width="100%">
```

```
<thead>
```

```
<tr>
```

<th>#</th>

<th> Name</th>

<th>Email </th>

<th>Contact no</th>

<th>DOB</th>

<th>Address</th>

<th>City</th>

<th>Country</th>

<th>Reg Date</th>

</tr>

<th>#</th>

<th> Name</th>

<th>Email </th>

<th>Contact no</th>

<th>DOB</th>

<th>Address</th>

<th>City</th>

<th>Country</th>

<th>Reg Date</th>

</tr>

</tr>

</tfoot>

<tbody>

<?php \$sql = "SELECT * from tblusers ";

\$query = \$dbh ->prepare(\$sql);

\$query->execute();

```
$results=$query->fetchAll(PDO::FETCH_OBJ);
```

```
$cnt=1;
```

```
if($query->rowCount() > 0)
```

```
{
```

```
foreach($results as $result)
```

```
{  
    ?>
```

```
<tr>
```

```
<td><?php echo htmlentities($cnt);?></td>
```

```
<td><?php echo htmlentities($result->FullName);?></td>
```

```
<td><?php echo htmlentities($result->EmailId);?></td>
```

```
<td><?php echo htmlentities($result->ContactNo);?></td>
```

```
<td><?php echo htmlentities($result->dob);?></td>
```

```
<td><?php echo htmlentities($result->Address);?></td>
```

```
<td><?php echo htmlentities($result->City);?></td>
```

```
<td><?php echo htmlentities($result->Country);?></td>
```

<td><?php echo htmlentities(\$result->RegDate);?></td>

</tr>

<?php \$cnt=\$cnt+1; }} ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Loading Scripts -->

<scriptsrc="js/jquery.min.js"></script>

<scriptsrc="js/bootstrap-select.min.js"></script>

<scriptsrc="js/bootstrap.min.js"></script>

<scriptsrc="js/jquery.dataTables.min.js"></script>

<scriptsrc="js/dataTables.bootstrap.min.js"></script>

<scriptsrc="js/Chart.min.js"></script>

<scriptsrc="js/fileinput.js"></script>

<scriptsrc="js/chartData.js"></script>

<scriptsrc="js/main.js"></script>

</body>

</html>

<?php } ?>

