VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590 018

A DBMS MINI PROJECT REPORT
on

"Vehicle Parking System"

Submitted in partial fulfillment of the requirements for the 5th semester of
Bachelor of Engineering in Computer Science and Engineering
Of Visvesvaraya Technological University, Belgavi

SUBMITTED BY

Sripradha	**1RN19CS155**
Tejaswini B M	**1RN19CS167**

Under the Guidance of

Mr. Karanam Sunil Kumar
Assistant Professor
Dept. of CSE

Dr. H R Shashidhara
Associate Professor
Dept. of CSE

Department of Computer Science and Engineering
RNS Institute of Technology
Channasandra, Dr.Vishnuvardan Road, Bengaluru-560 098

2021-2022

R N S Institute of Technology

Channasandra, Dr.Vishnuvardan Road, Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(NBA ACCREDITED FOR ACADEMIC YEARS 2018-19, 2019-20, 2020-21, 2021-22)

CERTIFICATE

Certified that the **DBMS Mini Project** work entitled **" Visitor Parking System "** has been successfully carried out by **Sripradha** and **Tejaswini B M** bearing USN 1RN19CS155 and 1RN19CS167 respectively, bonafide students of **RNS Institute of Technology,** currently studying in 3rd year **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2021-22. It is certified that all corrections and suggestions indicated for internal assessment by **Dr. H R Shashidhara** and **Mr. Karanam Sunil Kumar** have been incorporated in the report. The project report has been approved as it satisfies the mini-project requirements of the DBMS Lab of 5th semester, BE in CSE.

Mr. Karanam Sunil Kumar **Assistant Professor** **Dept. of CSE**	**Dr. H R Shashidhara** **Associate Professor** **Dept. of CSE**	**Dr. Kiran P** **Professor & HOD** **Dept. of CSE**

External Viva:

Name of the Examiners	**Signature with Date**
1.	
2.	

ACKNOWLEDGMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We would like to thank **Dr. M K Venkatesha**, Principal, RNSIT, Bangalore, for his support towards completing this mini project. We would like to thank **Dr. Kiran P**, Prof. & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express sincere gratitude to our guide **Mr. Karanam Sunil Kumar**, Assistant Professor and mentor **Dr. H R Shashidhara,** Associate Professor, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We also would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date : Sripradha 1RN19CS155

Place : Bengaluru Tejaswini B M 1RN19CS167

ABSTRACT

Vehicle Parking System is a system to manage the records of the incoming and outgoing vehicles in a parking area. The objective of this project is to design a Vehicle Parking System that enables the control of vehicles with the help of vehicle registration number. This system keeps track of the entry and exit of cars and maintains the record about cars within the parking lot.

Now days in many public places such as malls, multiplex system, hospitals and offices there is a crucial problem of vehicle parking. Moreover this involves, lot of manual labour and investment. Instead of vehicle caught in towing the vehicle can park on safe and security with low cost. These features are hereby very necessary nowadays to secure your car and also to evaluate the fee structure for every vehicles entry and exit.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Vehicle Parking System is a system to manage the records of the incoming and outgoing vehicles in a parking slot. The four most basic operations that can be performed with database systems are Create, Read, Update, and Delete, they are backbone for interacting with any database.

The following operations can be performed on Database:

- Creating the database
- Reading from the database
- Updating the database
- Deleting the database
- Inserting to the database
- Searching the database

1.1 PROBLEM STATEMENT & DESCRIPTION

To design a Vehicle Parking System using PHP and MySQL along with HTML, CSS, JavaScript and Bootstrap.

The Vehicle Parking System project in PHP focuses mainly on keeping track of vehicle's parking. Also, the system displays all the vehicle's entry and outgoing records. In addition, the system allows adding vehicle categories too. This project only contains an admin panel. In an overview of this web application, an admin has all control over the system. He helps to maintain the flow of the system. An admin can simply add vehicle categories by providing category names. After the management of vehicle categories, the admin can simply insert the vehicle's entry. For this, he has to enter the vehicle's number, model name, category, and owner's information.

After setting up the vehicle's parking, now the admin can manage outgoing vehicles. Here, the admin, who is an user can simply take an action for checking out the parking. In order to exit a vehicle's parking, the admin has to enter the total parking charge with remarks manually. Once a vehicle exits the parking, those records can be found under the outgoing vehicle records. Here, the system generates invoice receipts for each and every vehicle. The user can view details of each available vehicle. Also, the user can view and print out the parking receipt of each. Each record consists of parking number, total charge, vehicle registration number, owner details, and more.

On the other hand, an admin can list out the reports between dates. These reports help out to state vehicle's parking records between the selected dates. The steps are simple, the user only needs to select from and to date.

And after that, the system displays it all between those mentioned dates where the user can also view their details. Additionally, the admin can view total earnings to date. And also, the system displays the current date total earning. He/she can search out the vehicle's record using the vehicle's registration number from the sidebar. Besides, an admin can have an overview records of the total vehicle's entry with a number of in and out vehicles and total parking within a 24 hours span time. The system represents overall records using graphical charts like piechart to present a summary of records.

And finally additional features in this system includes updating the user's profile where user can update his name and contact details. Also, an admin can change the password by entering the current password with a new and confirmation password. The last feature is about company information. The system allows the user to update company settings. It includes of company's name, the company's email address, website URL, and company's address. Hence the available features are as follows:

- Dashboard
- Vehicle Entry
- IN Vehicles
- OUT Vehicles
- Vehicle Category
- View Reports
- Total Income

1.2 ADVANTAGES OF PROPOSED SYSTEM

The advantages of Vehicle Parking System are as follows:

- It becomes easier for administrator to retrieve information about in-vehicles and out-vehicles.
- It reduces Operational and Management Costs.
- Easy to manage and maintain
- Authentication during log-in ensures security.
- Graphical Representation of Data helps in analysing.

1.3 APPLICATIONS OF PROPOSED SYSTEM

- In public places such as malls, multiplex complex, hospitals, hotels, offices, market areas.
- Part of this system has implementation in Tollgate Management System.
- In implementation of Automated Parking System.

CHAPTER 2

REQUIREMENT

2.1 HARDWARE REQUIREMENT

The Hardware requirements are very minimal and the program can be run on most of the machine.

Processor Type	Pentium iv or above for optimum performance
System RAM	2.00GB and above
Input Device	Standard Input Device Keyboard
Output Device	Standard Output Device Monitor(display screen)

2.2 SOFTWARE REQUIREMENT

The software requirements are as follows:

Operating System	Windows 10
XAMPP Server	MySQL, phpMyAdmin
IDE	Visual Studio Code
Browser	Google Chrome

2.2.1 Operating System:

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs. One such operating system is Windows 10. WINDOWS 10 provides desktop oriented interface. It is a major release of the Windows NT operating system developed by Microsoft. It is the successor to Windows 8.1, released nearly two years earlier, and was released to manufacturing on July 15, 2015.

2.2.2 XAMPP Server:

XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl, and it allows you to build WordPress site offline, on a local web server on your computer. This simple and lightweight solution works on Windows, Linux, and Mac – hence the “cross-platform” part.. Xampp server allows you to setup a local server with the same characteristics as your production.

2.2.3 MySQL:

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes.
- It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to updates the table indexes automatically.
- It uses many SQL queries and combines useful information from multiple tables for the end-users.

MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client-server model. The core of the MySQL database is the MySQL Server. This server is available as a separate program and responsible for handling all the database instructions, statements, or commands.

The working of MySQL database with MySQL Server are as follows:

- MySQL creates a database that allows you to build many tables to store and manipulate data and defining the relationship between each table.
- Clients make requests through the GUI screen or command prompt by using specific SQL expressions on MySQL.
- Finally, the server application will respond with the requested expressions and produce the desired result on the client-side.

2.2.4 phpMyAdmin:

It is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

Features provided by the program include:

- Web interface
- MySQL and MariaDB database management
- Import data from CSV, JSON and SQL
- Export data to various formats: CSV, SQL, XML, JSON, PDF (via the TCPDF library), ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, LaTeX, SQL, and others
- Administering multiple servers
- Creating PDF graphics of the database layout
- Creating complex queries using query-by-example (QBE)
- Searching globally in a database or a subset of it
- Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
- Live charts to monitor MySQL server activity like connections, processes, CPU/memory usage, etc.
- Network traffic to the SQL server
- Working with different operating systems like Windows*, Linux*, OS/2, Free BSD* Unix* (such as Sun* Solaris* , AIX) and others.
- Make complex SQL queries easier.

2.2.5 Visual Studio Code:

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine.

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor and language support. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol.

2.2.6 Google Chrome:

Google Chrome is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, built with free software components from Apple WebKit and Mozilla Firefox. It was later ported to Linux, macOS, iOS, and Android, where it is the default browser. The browser is also the main component of Chrome OS, where it serves as the platform for web applications.

2.3 FUNCTIONAL REQUIREMENT

2.3.1 HTML:

HTML (Hypertext Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behaviour (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, , , <img>, <nav>, < , <ol>, <li> and many others.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects like interactive forms can be embedded into the rendered page. It provides a way to create structured documents by denoting structural semantics for the text like headings, paragraphs, lists, links, quotes and other items.

2.3.2 CSS:

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

2.3.3 JavaScript:

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js. Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

2.3.4 Bootstrap:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking.

Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-coloured tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the five predefined fixed widths, depending on the size of the screen showing the page:

- Smaller than 576 pixels
- 576–768 pixels
- 768–992 pixels
- 992–1200 pixels
- Larger than 1200 pixels

Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.

2.3.5 PHP:

PHP is a popular general-purpose scripting language that is especially suited to web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response.

PHP is very popular language because of its simplicity and open source. There are some important features of PHP given below:

- Performance:
  PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.
- Open Source
  PHP source code and software are freely available on the web. You can develop all the versions of PHP according to your requirement without paying any cost.
- Embedded
  PHP code can be easily embedded within HTML tags and script.
- Platform Independent
  PHP is available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.
- Database Support
  PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc.
- Web servers Support
  PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS.
- Security
  PHP is a secure language to develop the website. It consists of multiple layers of security to prevent threads and malicious attacks.
- Control
  Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like you can make changes easily whenever you want.

CHAPTER 3

SYSTEM DESIGN

3.1 ER DIAGRAM

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases.

A real-world thing either living or non-living that is easily recognizable and non-recognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.

ER diagrams are created based on three basic concepts: entities, attributes and relationships. ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

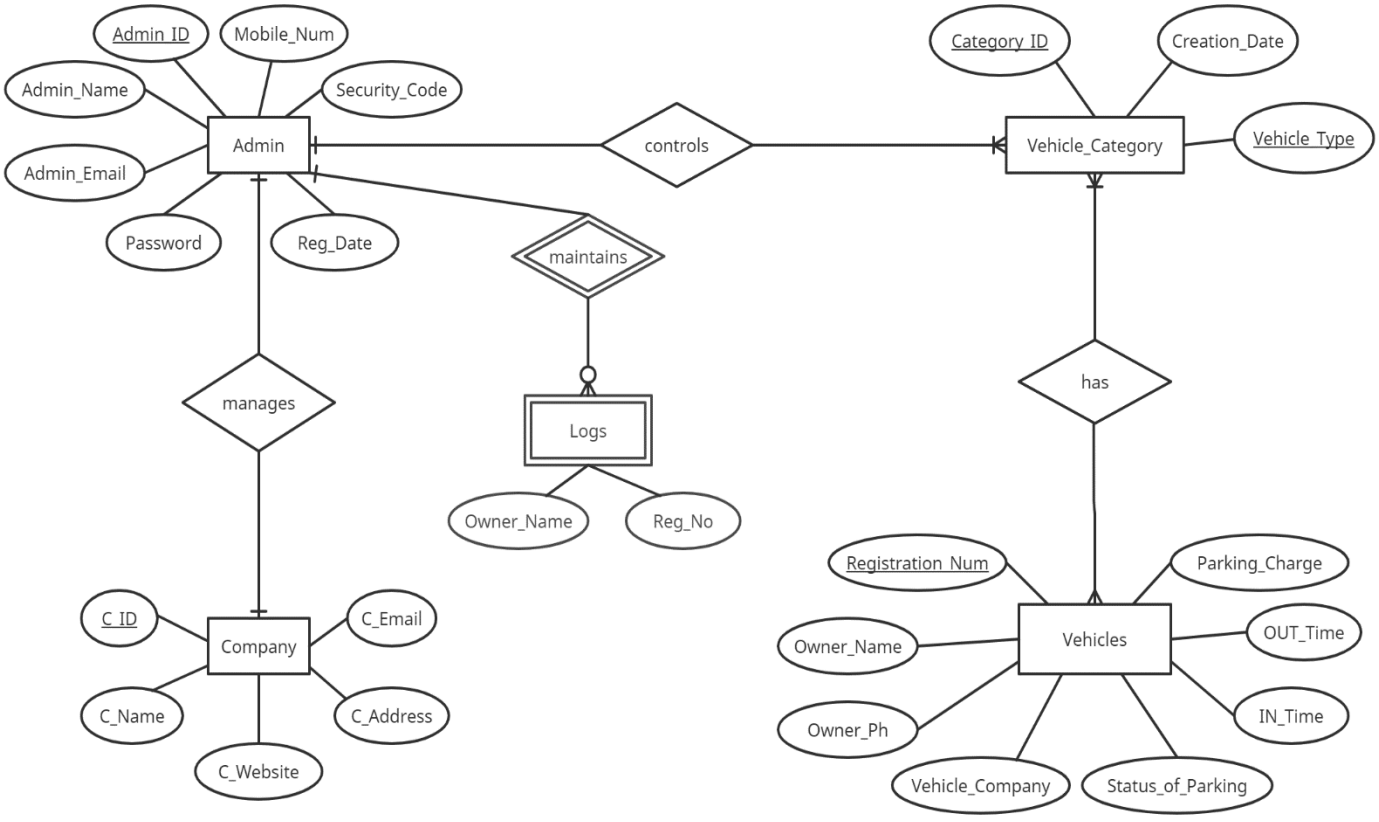

Fig: 3.1.1 ER Diagram

3.2 SCHEMA DIAGRAM

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organised using the schema diagram. A database schema, along with primary key and foreign key dependencies, can be depicted by schema diagrams.

A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

A database schema can be divided broadly into two categories −
Physical Database Schema − This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
Logical Database Schema − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

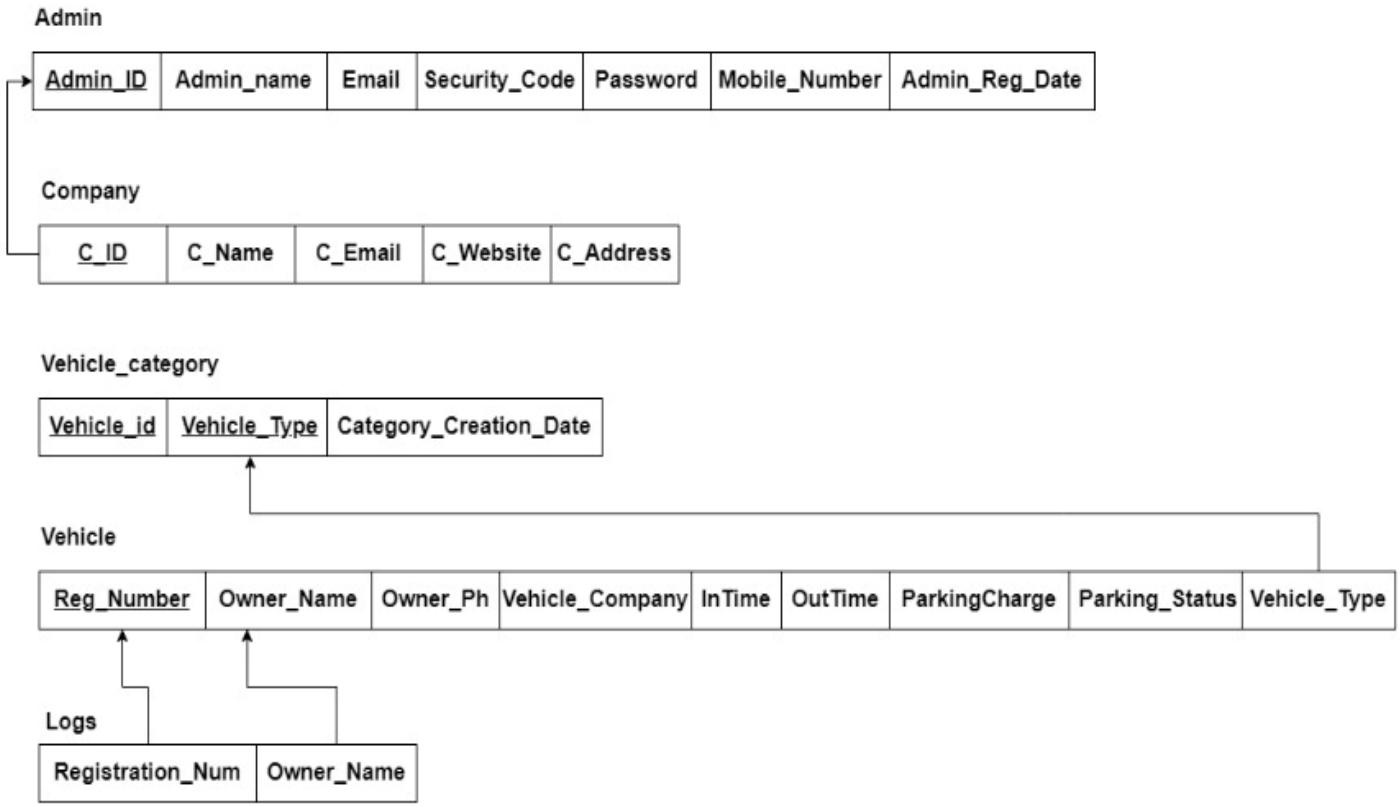

Fig: 3.2.1 Schema Diagram

3.3 USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behaviour of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

The rules that must be followed while drawing a use case diagram are a pertinent and meaningful name should be assigned to the actor or a use case of a system. The communication of an actor with a use case must be defined in an understandable way. Specified notations to be used as and when required. The most significant interactions should be represented among the multiple no of interactions between the use case and actors.

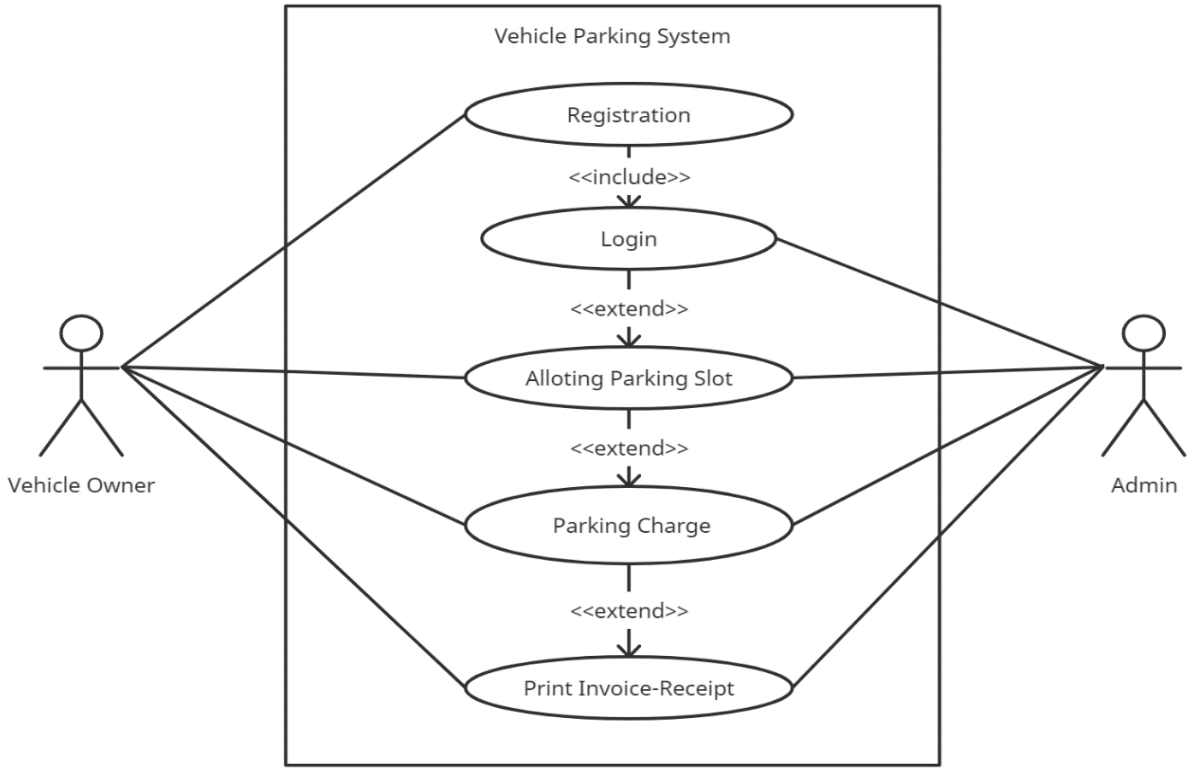

Fig: 3.3.1 Use Case Diagram

CHAPTER 4

IMPLEMENTATION

4.1 CODE

index.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconn.php');
if(isset($_POST['login']))
  {
    $adminuser=$_POST['username'];
    $password=md5($_POST['password']);
    $query=mysqli_query($con,"SELECT ID from admin where  UserName='$adminuser' &&
Password='$password' ");
    $ret=mysqli_fetch_array($query);
    if($ret>0){
      $_SESSION['vpmsaid']=$ret['ID'];
     header('location:dashboard.php');
    }
    else{
    $msg="Login Failed !!";
    }
  }
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>VPS</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/datepicker3.css" rel="stylesheet">
    <link href="css/styles.css" rel="stylesheet">
    <script src="js/html5shiv.js"></script>
    <script src="js/respond.min.js"></script>
</head>
<body>
    <div class="row">
        <center><b><h2> Vehicle<img src="carparking.png" align='top' width="75" >arking
System</b></h2></center>
          <div class="login-panel panel panel-default">

              <div class="panel-heading"><font color="#1e90ff">Log In</font></div>
              <div class="panel-body">
                 <form method="POST">
                 <?php if($msg)
                    echo "<div class='alert bg-danger' role='alert'>
                    <em class='fa fa-lg fa-warning'> </em>
```

```
                $msg
                  <a href='#' class='pull-right'>
                  <em class='fa fa-lg fa-close'>
                  </em></a></div>" ?>
                  <fieldset>
                    <div class="form-group">
                      <input class="form-control" placeholder="Username" name="username" type="text">
                    </div>
                    <div class="form-group">
                      <input class="form-control" placeholder="Password" name="password" type="password"
value="">
                    </div>
                    <div class="checkbox">
                      <a href="forgot-password.php" style="text-decoration:none;">Forgot Password?</a>
                    </div>
                    <button class="btn btn-success" type="submit" name="login">Login</button></fieldset>
                </form>
            </div>
          </div>
    </div>
    <script src="js/jquery-1.11.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

in-vehicle.php

```
<?php
    session_start();
    error_reporting(0);
    include('includes/dbconn.php');
    if (strlen($_SESSION['vpmsaid']==0)) {
        header('location:logout.php');
        } else {
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>VPS</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
        <?php include 'includes/navigation.php' ?>
     <?php
        $page="in-vehicle";
        include 'includes/sidebar.php'
        ?>
        <div class="row">
            <ol class="breadcrumb">
                <li><a href="dashboard.php">
                </a></li>
                <li class="active">Incoming Vehicle Management</li>
```

```
          </ol>
        </div>
        <div class="row">
          </div>
        </div><!--/.row-->
        <div class="row">
                <div class="panel panel-default">
                  <div class="panel-heading">Incoming Vehicles</div>
                  <div class="panel-body">
                  <table id="example" class="table table-striped table-hover table-bordered"
style="width:100%">
        <thead>
          <tr>
            <th>#</th>
            <th>Vehicle No.</th>
            <th>Company</th>
            <th>Category</th>
            <th>Parking Number</th>
            <th>Vehicle's Owner</th>
            <th></th>
          </tr>
        </thead>
        <tbody>
        <?php
        $ret=mysqli_query($con,"SELECT * FROM vehicle_info WHERE Status='' ORDER BY InTime
DESC");
        $cnt=1;
        while ($row=mysqli_fetch_array($ret)) {
        ?>
          <tr>
          <td><?php echo $cnt;?></td>
          <td><?php  echo $row['RegistrationNumber'];?></td>
          <td><?php  echo $row['VehicleCompanyname'];?></td>
          <td><?php  echo $row['VehicleCategory'];?></td>
          <td><?php  echo 'CA-'.$row['ParkingNumber'];?></td>
          <td><?php  echo $row['OwnerName'];?></td>
          <td><a href="update-incomingdetail.php?updateid=<?php echo $row['ID'];?>"><button type="button"
class="btn btn-sm btn-danger">Take Action</button></a>
          </td>
          </tr>
            <?php $cnt=$cnt+1;}?>
        </tbody>
</table>
              </div>
        <?php include 'includes/footer.php'?>
    </div>  <!--/.main-->
    <script>
        $(document).ready(function() {
    $('#example').DataTable();
} );
    </script>
</body>
</html>
<?php }  ?>
```

dbconn.php

```
<?php
   $con=mysqli_connect("localhost", "root", "", "vehicle_db");
   if(mysqli_connect_errno()){
   echo "Connection Failed".mysqli_connect_error();
   }
 ?>
```

out-vehicle.php

```
<?php
   session_start();
   error_reporting(0);
   include('includes/dbconn.php');
   if (strlen($_SESSION['vpmsaid']==0)) {
      header('location:logout.php');
      } else {
?>
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>VPS</title>
</head>
<body>
      <?php include 'includes/navigation.php' ?>
      <?php
      $page="out-vehicle";
      include 'includes/sidebar.php'
      ?>
      <div class="row">
         <ol class="breadcrumb">
            <li><a href="dashboard.php">
            </a></li>
            <li class="active">Outgoing Vehicle Management</li>
         </ol>
      </div>
      <div class="row">
         <div class="col-lg-12">
         </div>
      <div class="row">
            <div class="col-lg-12">
               <div class="panel panel-default">
                  <div class="panel-heading">Outgoing Vehicles</div>
                  <div class="panel-body">
                  <table id="example" class="table table-striped table-hover table-bordered"
style="width:100%">
      <thead>
         <tr>
            <th>#</th>
            <th>Vehicle No.</th>
```

```
                <th>Company</th>
                <th>Category</th>
                <th>Parking Number</th>
                <th>Charge</th>
                <th>Vehicle's Owner</th>
                <th></th>
            </tr>
        </thead>
    </table>
                    </div>
                </div>
            </div>
</div>
        <?php include 'includes/footer.php'?>
    </div>
    <script>
        $(document).ready(function() {
    $('#example').DataTable();
} );
    </script>

</body>
</html>

<?php }  ?>
```

vehicle_category.php

```
<?php
    session_start();
    error_reporting(0);
    include('includes/dbconn.php');
    if (strlen($_SESSION['vpmsaid']==0)) {
        header('location:logout.php');
        } else {
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>VPS</title>
</head>
<body>
        <?php include 'includes/navigation.php' ?>

        <?php
        $page="vehicle-category";
        include 'includes/sidebar.php'
        ?>
        <div class="row">
            <ol class="breadcrumb">
                <li><a href="dashboard.php">
```

```
                <em class="fa fa-home"></em>
              </a></li>
              <li class="active">Vehicle Category Management</li>
            </ol>
          </div>

          <div class="row">
            <div class="col-lg-12">
              <!-- <h1 class="page-header">Vehicle Management</h1> -->
            </div>
          </div><!--/.row-->

          <div class="row">
                <div class="col-lg-12">
                  <div class="panel panel-default">
                    <div class="panel-heading">Vehicle Categories <a href="add-category.php" type="button"
class="btn btn-sm btn-primary">Add New Vehicle Category</a></div>
                    <div class="panel-body">
                    <table id="example" class="table table-striped table-hover table-bordered"
style="width:100%">

          <thead>
            <tr>
              <th>#</th>
              <th>Vehicle Category</th>
              <th>Published On</th>
              <th>Actions</th>
            </tr>
          </thead>
          <tbody>
          <?php
          $ret=mysqli_query($con,"SELECT * from  vcategory");
          $cnt=1;
          while ($row=mysqli_fetch_array($ret)) {
          ?>
            <tr>
            <td><?php echo $cnt;?></td>
            <td><?php  echo $row['VehicleCat'];?></td>
            <td><?php  echo $row['CreationDate'];?></td>
            <td><a href="update-category.php?editid=<?php echo $row['ID'];?>"> <button class="btn btn-success
btn-sm"><i class="fa fa-edit"></i></button> </a>
            <a href="remove-category.php?editid=<?php echo $row['ID'];?>"> <button class="btn btn-danger btn-
sm"><i class="fa fa-trash"></i></button> </a>
            </td>
            </tr>
              <?php $cnt=$cnt+1;}?>
          </tbody>
      </table>
                    </div>
                  </div>
                </div>
```

CHAPTER 5

RESULT ANALYSIS

5.1 WEB INTERFACE SNAPSHOTS

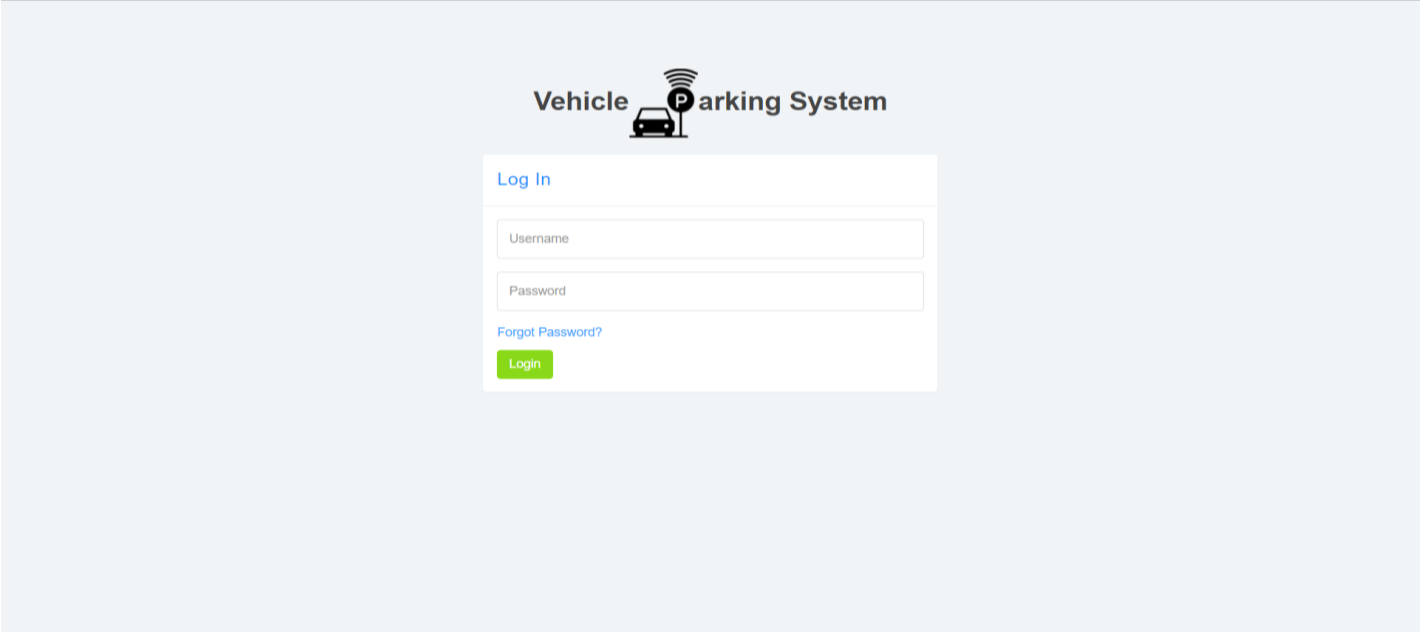

Fig: 5.1.1 Login Page

Login page is where the admin can login into the website with the credentials like username and password. After the login button is pressed, admin can login successfully.

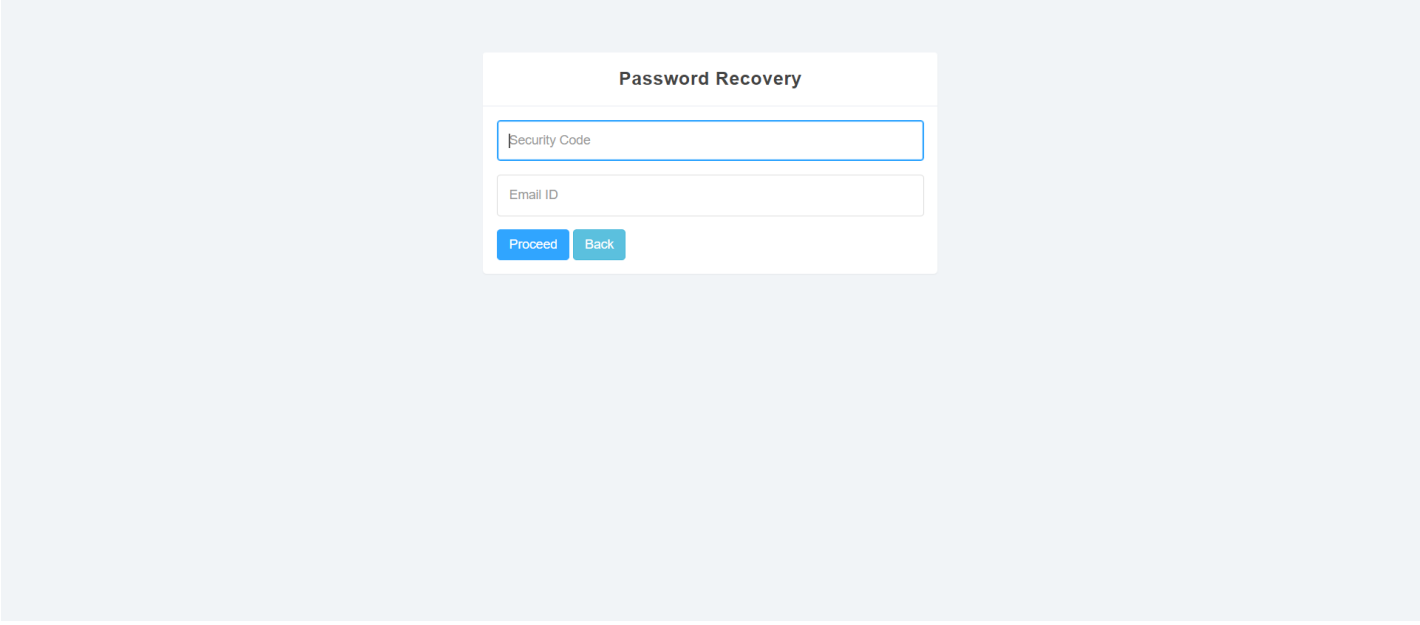

Fig: 5.1.2 Password Recovery

If in case, admin forgets the password then, he can update the password using security code provided by the company and his email ID.

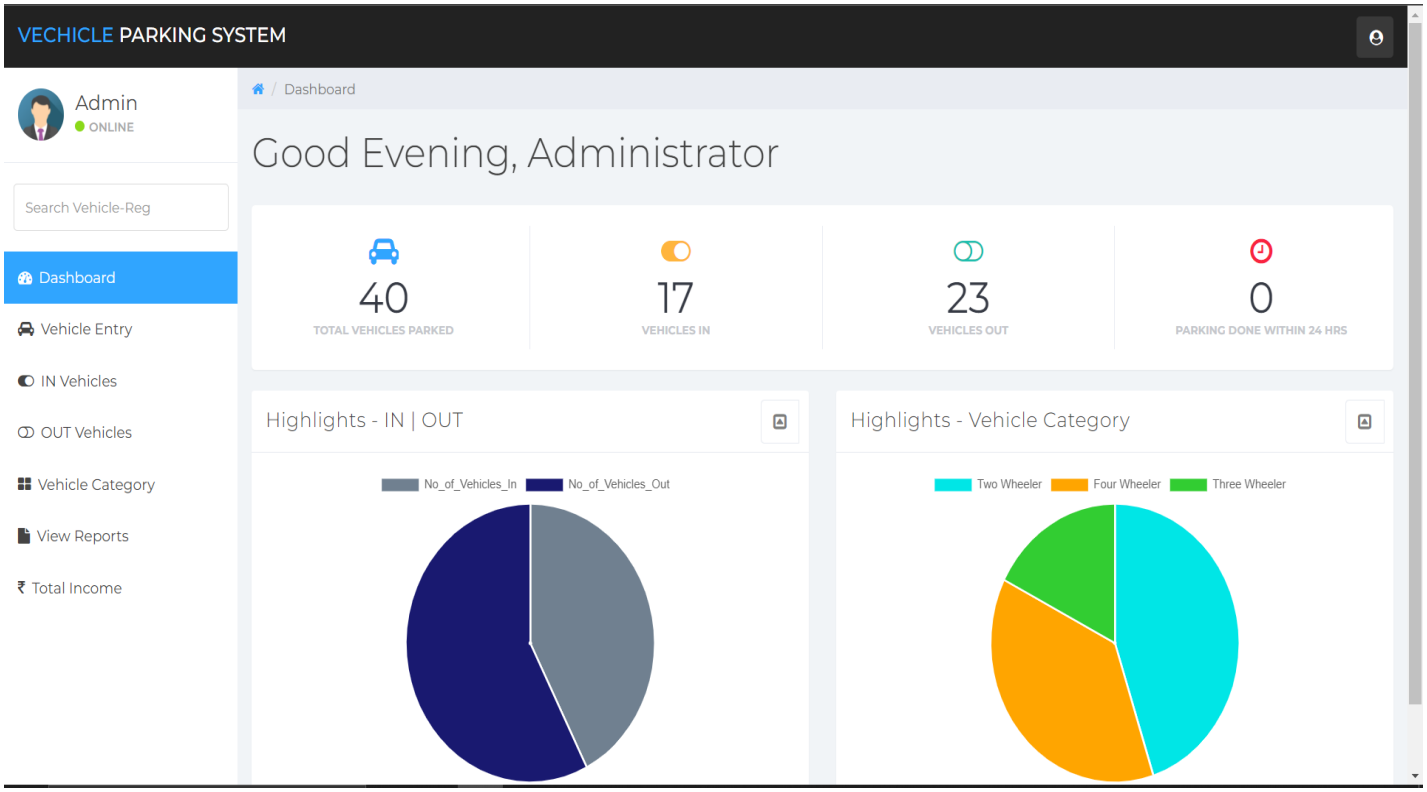

Fig: 5.1.3 Dashboard

This snapshot shows the dashboard features available. The vehicle category pie chart gives information about number of vehicles parked according to the category. The in-out pie chart differentiates between number of in and out vehicles.

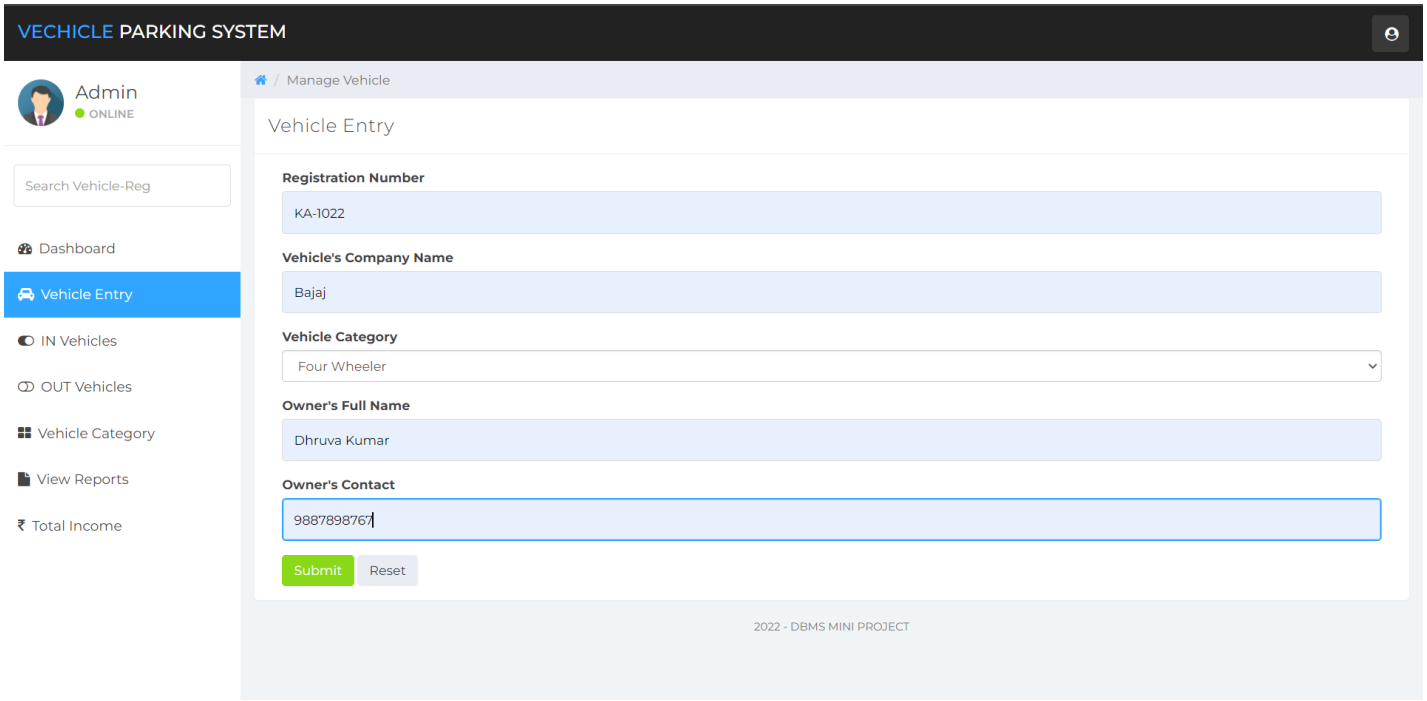

Fig: 5.1.4 Vehicle Entry

With the help of vehicle entry feature, details about the vehicle entered into the parking area and its owner is registered.

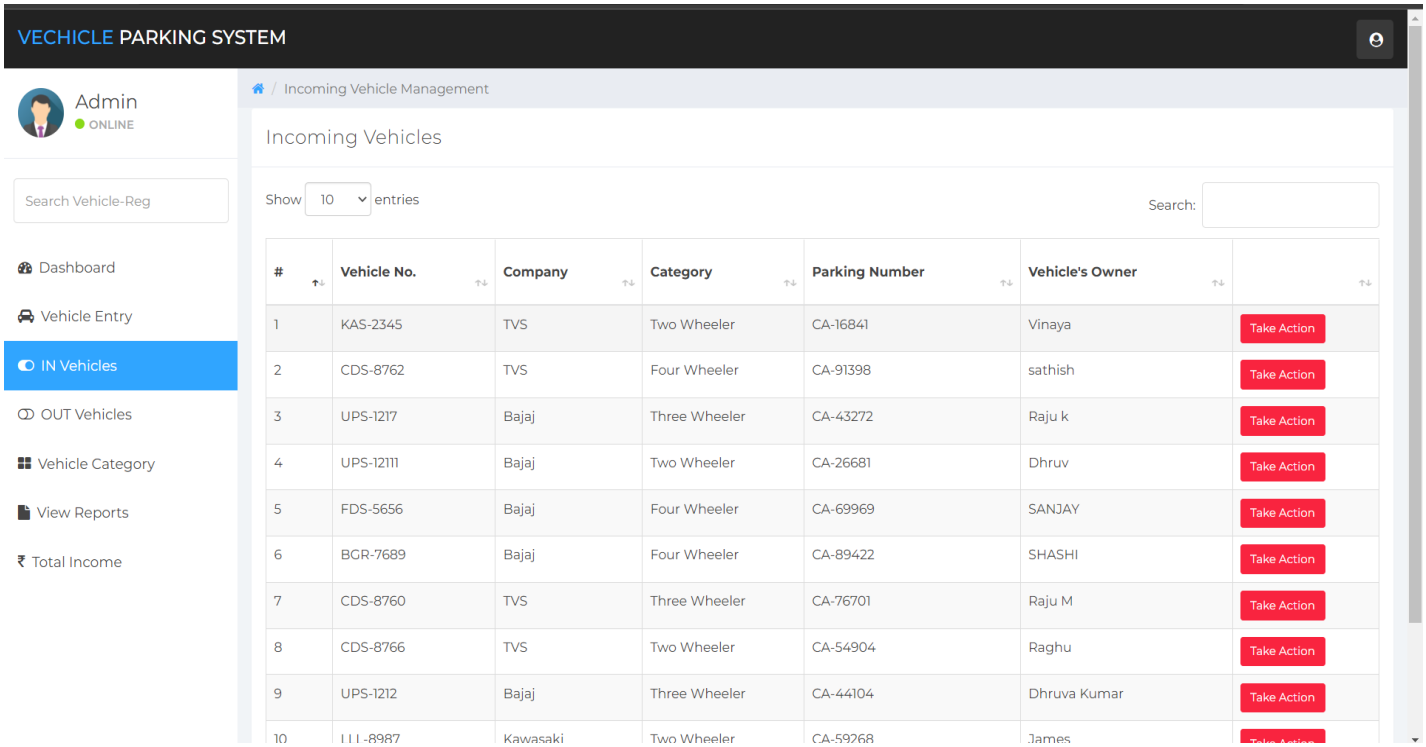

Fig: 5.1.5 List of in vehicles

With the help of in-vehicle feature, details about all the vehicles in the parking area along with the owner name is listed. Here we can limit the entries shown using the option available.

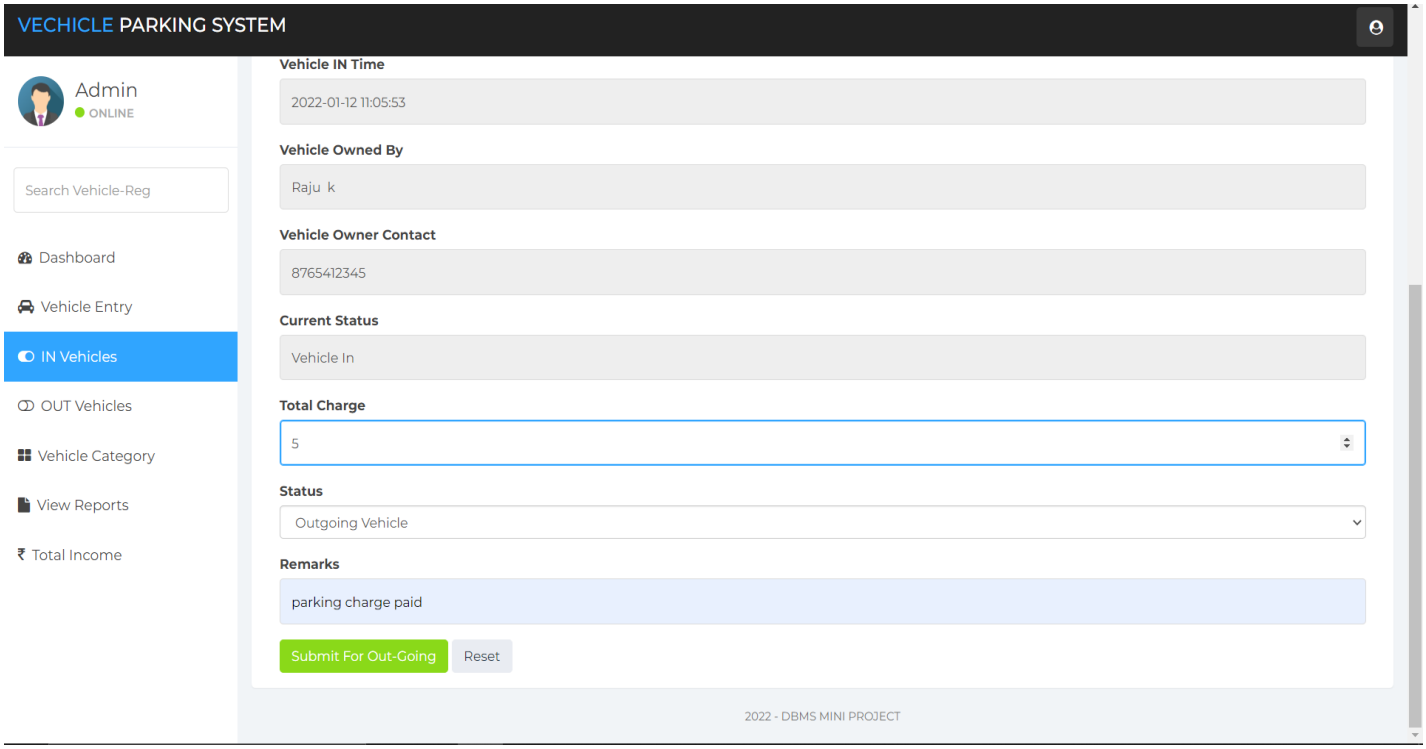

Fig: 5.1.6 Take action for in vehicles

For the in-vehicles, admin can take action that is, if the vehicle is moving out then parking charge need to be entered by admin and provide a remark.

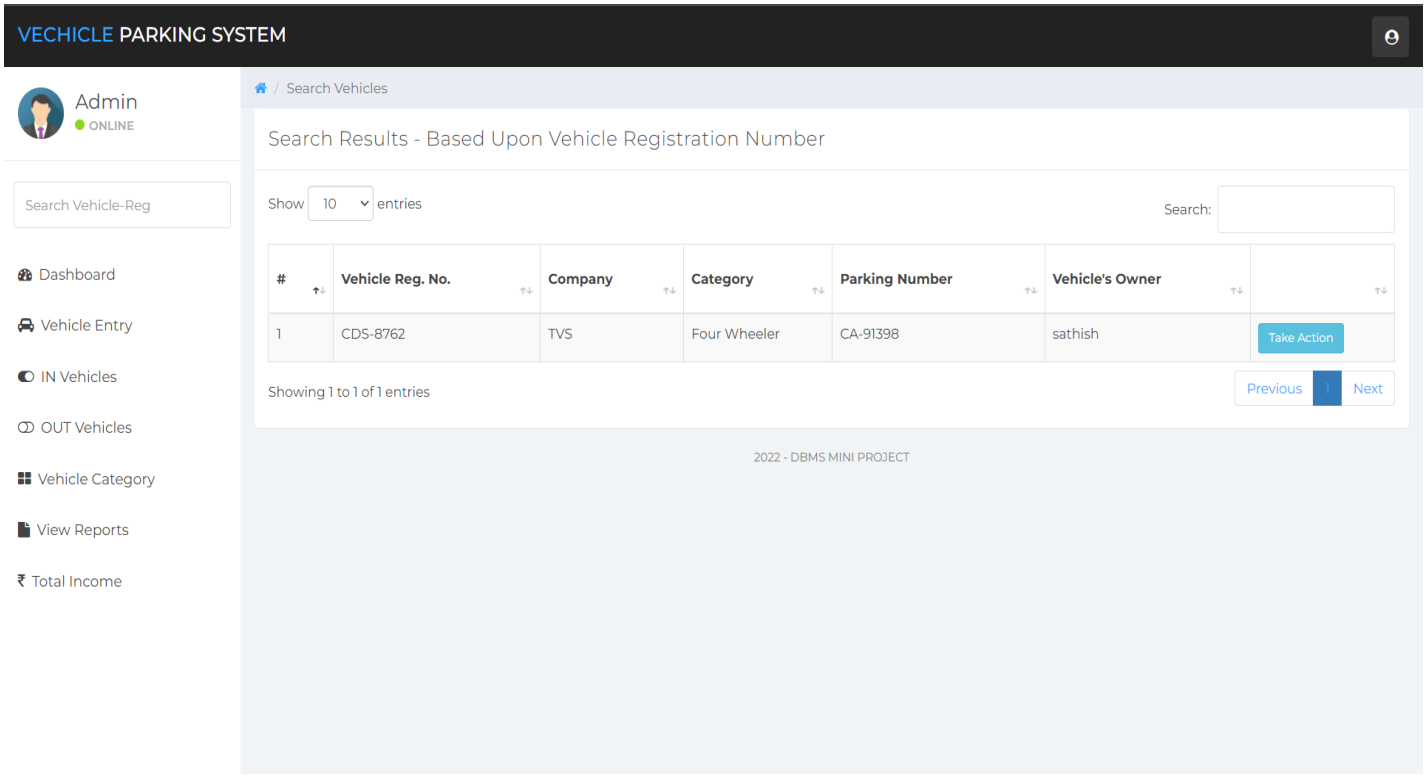

Fig: 5.1.7 Searching for a vehicle details

There is search option available in in-vehicles feature. Admin can search for a particular vehicle using parking number or vehicle registration number.

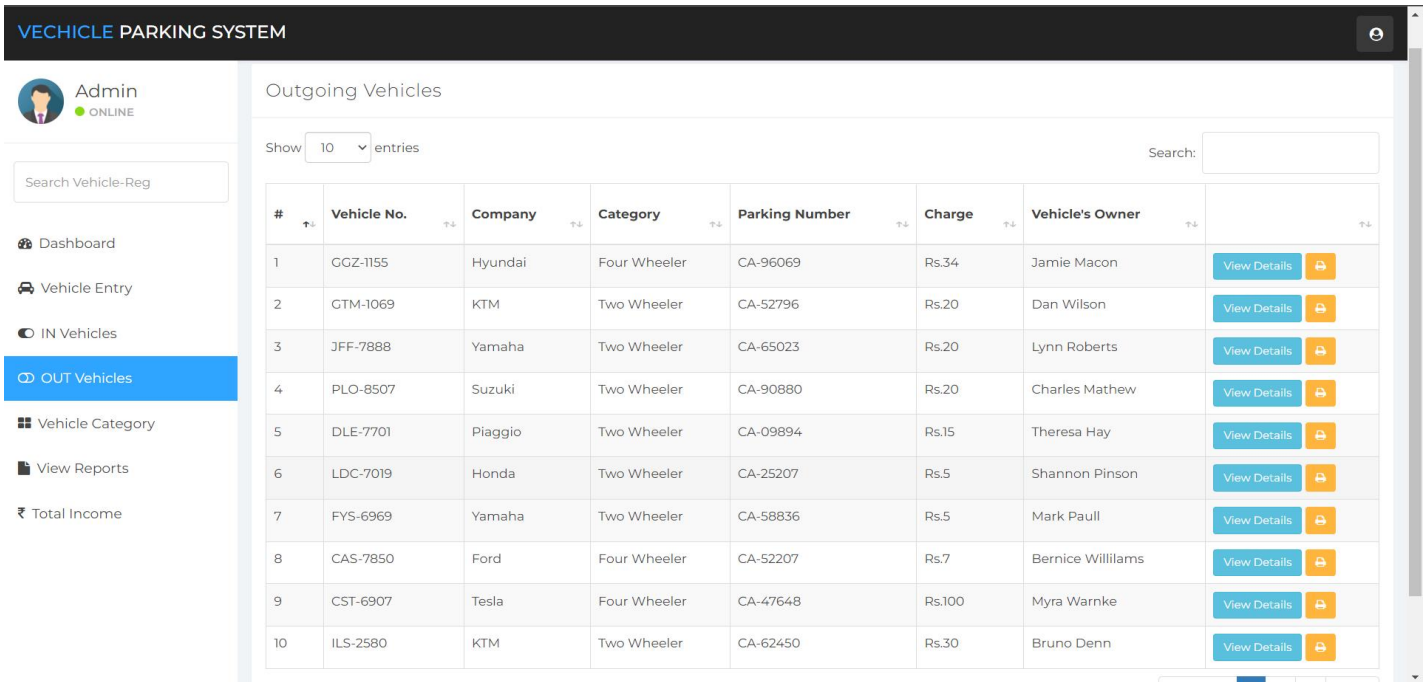

Fig: 5.1.8 List of out vehicles

With the help of out-vehicle feature, details about all the vehicles moved out of parking area along with the owner name is listed. Here we can limit the entries shown using the option available. And also search for particular vehicle using parking number or vehicle registration number.

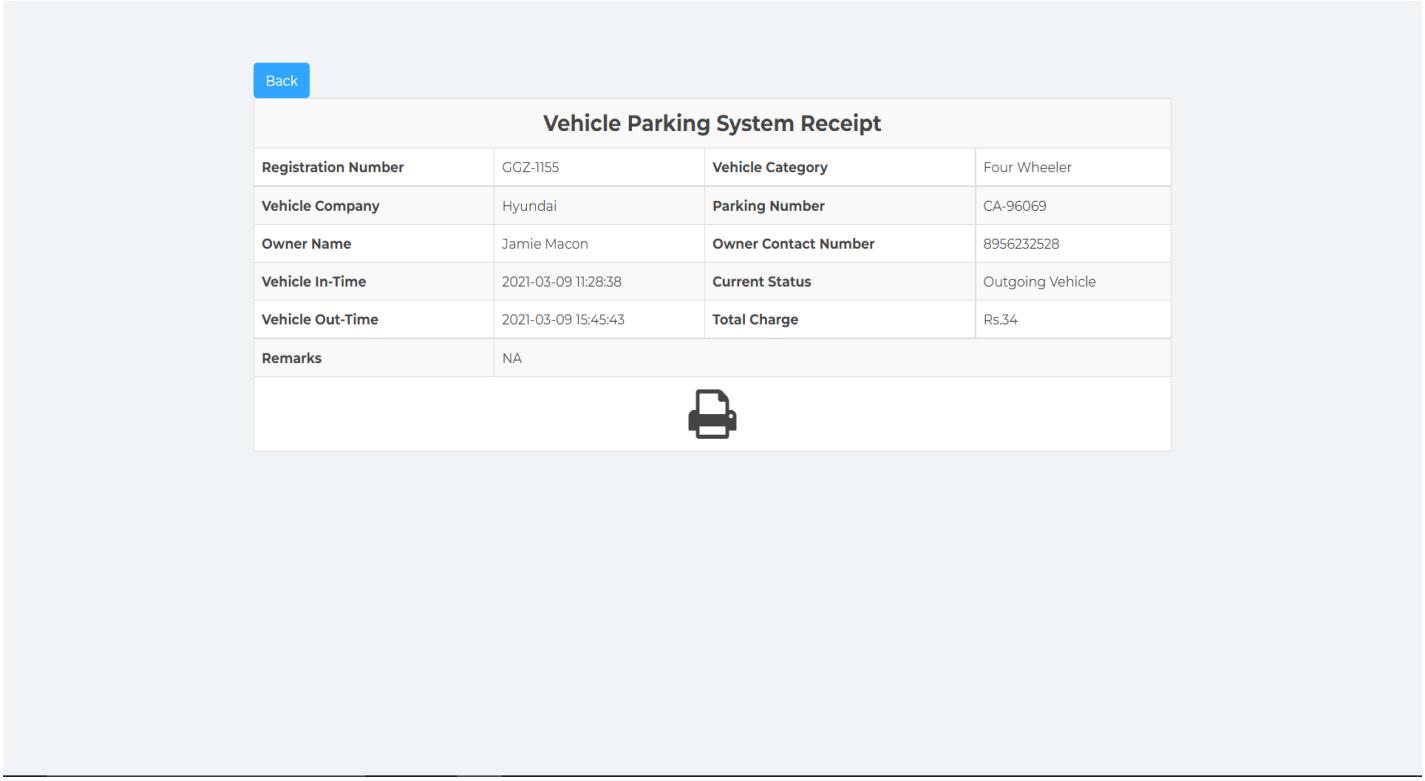

Fig: 5.1.9 Receipt

Receipt can be generated with the view-details option available in out-vehicle bar. Details about vehicles, owner along with vehicle in-time and out-time is available.

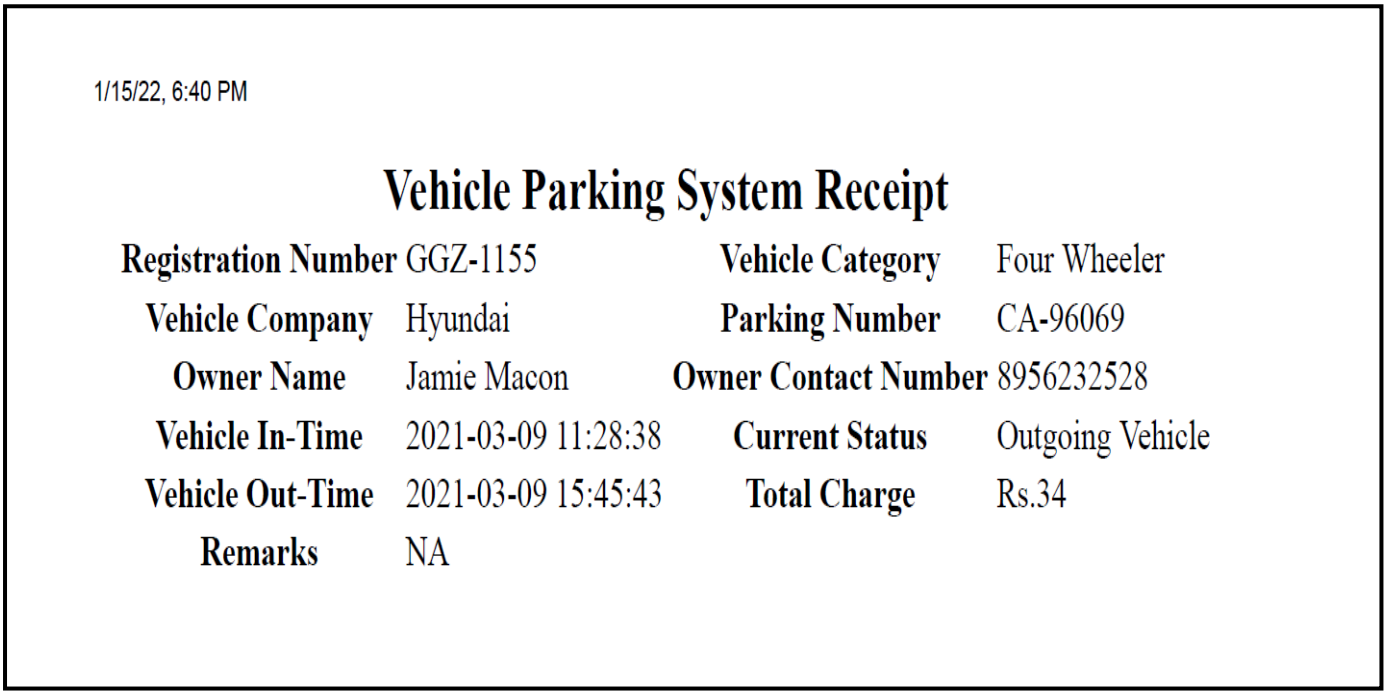

Fig: 5.1.10 PDF Copy of Receipt

Details about vehicles, owner along with vehicle in-time and out-time available can be printed. And the pdf format of the receipt looks like the above.

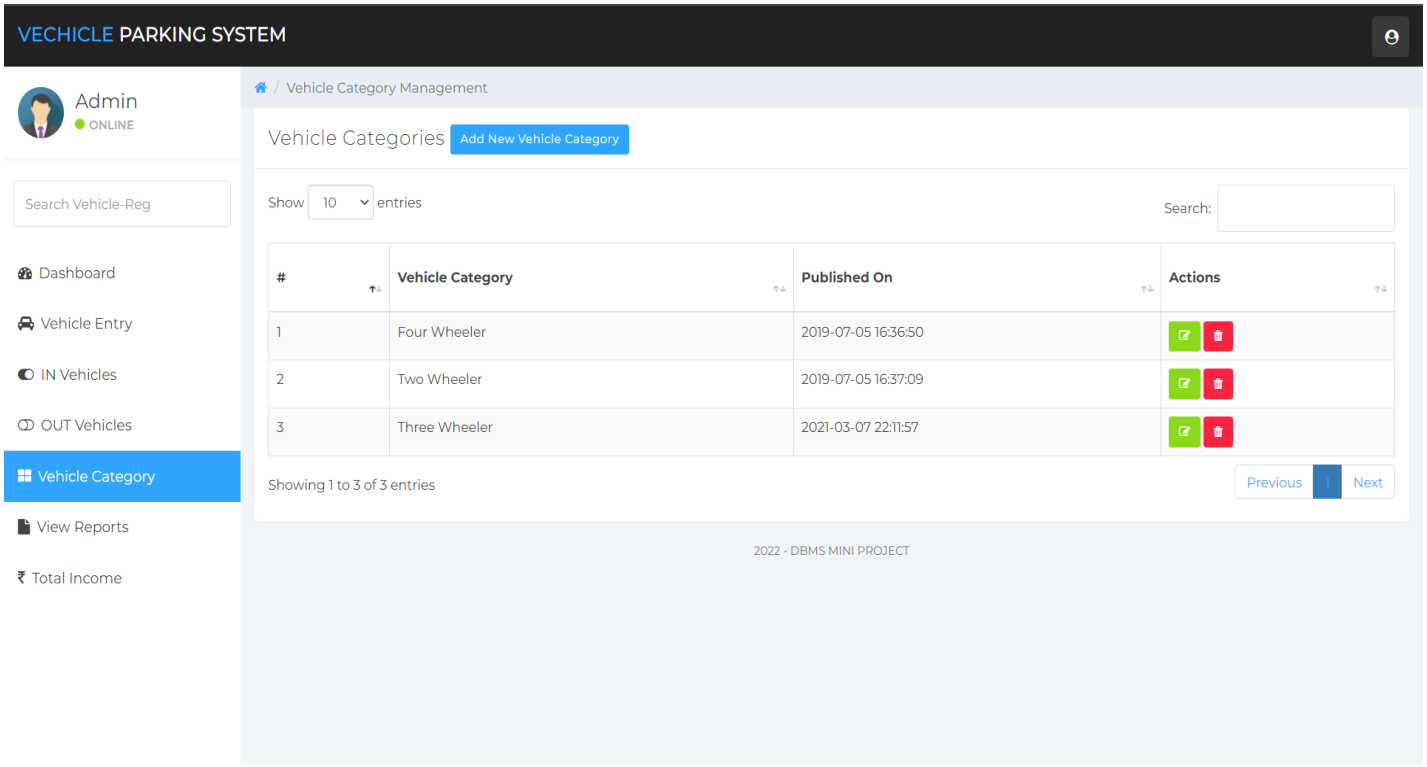

Fig: 5.1.11 List of vehicle categories

New vehicle category can be added or existing can be deleted with the vehicle category feature. And even we can search for the vehicle category with the search option.

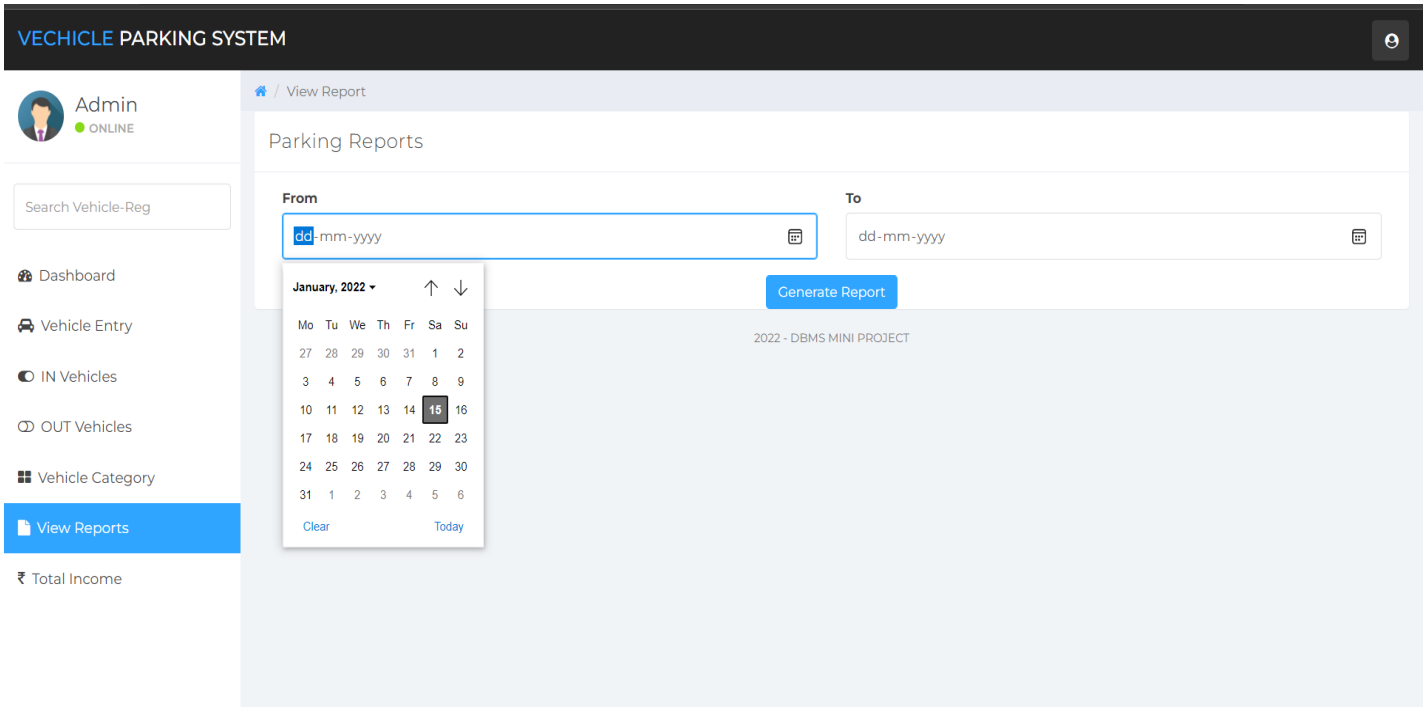

Fig: 5.1.12 View reports with in particular period

Admin can view the reports within particular period by specifying the from date and to date in the mentioned format dd-mm-yyyy only.

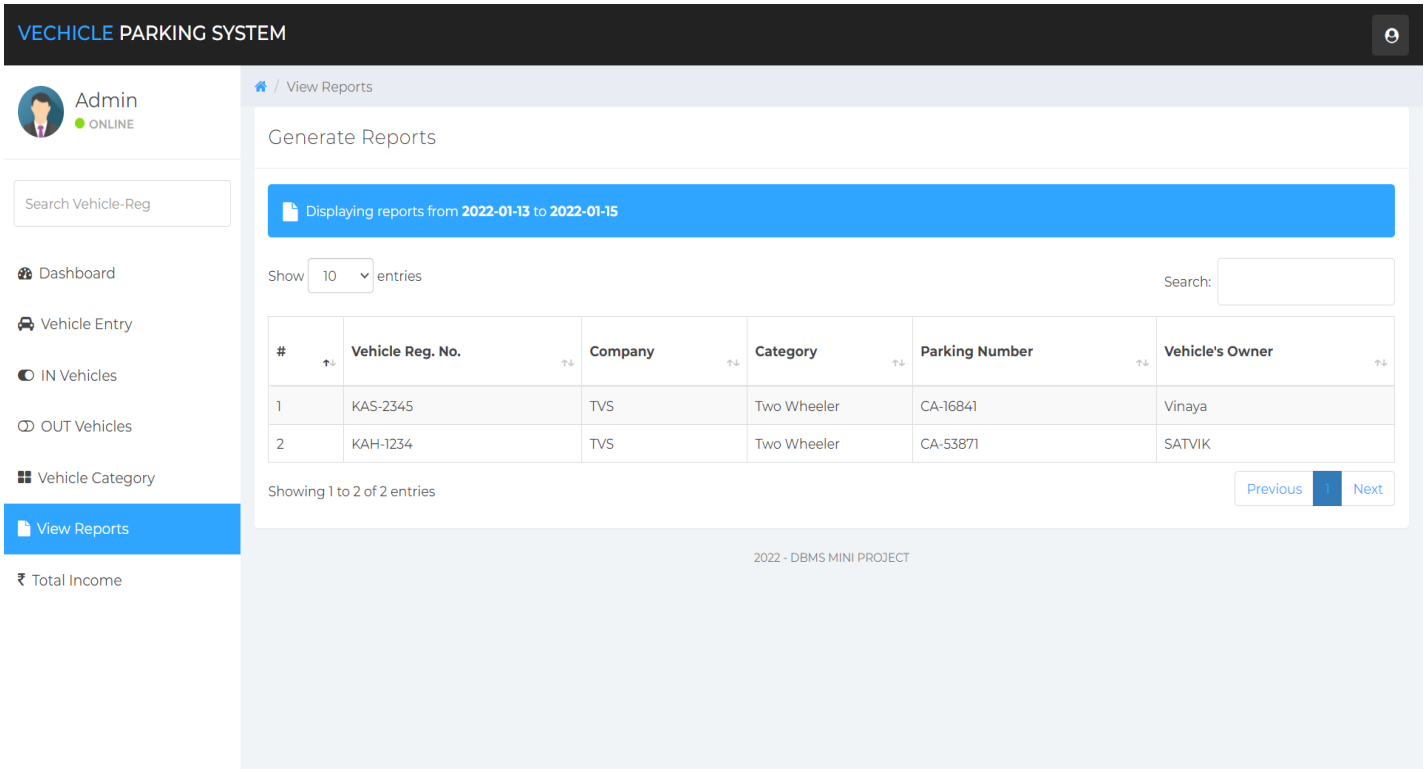

Fig: 5.1.13 Vehicle Report

Admin can view the reports within particular period by specifying the dates. After pressing generate button, list of vehicle entries can be seen.

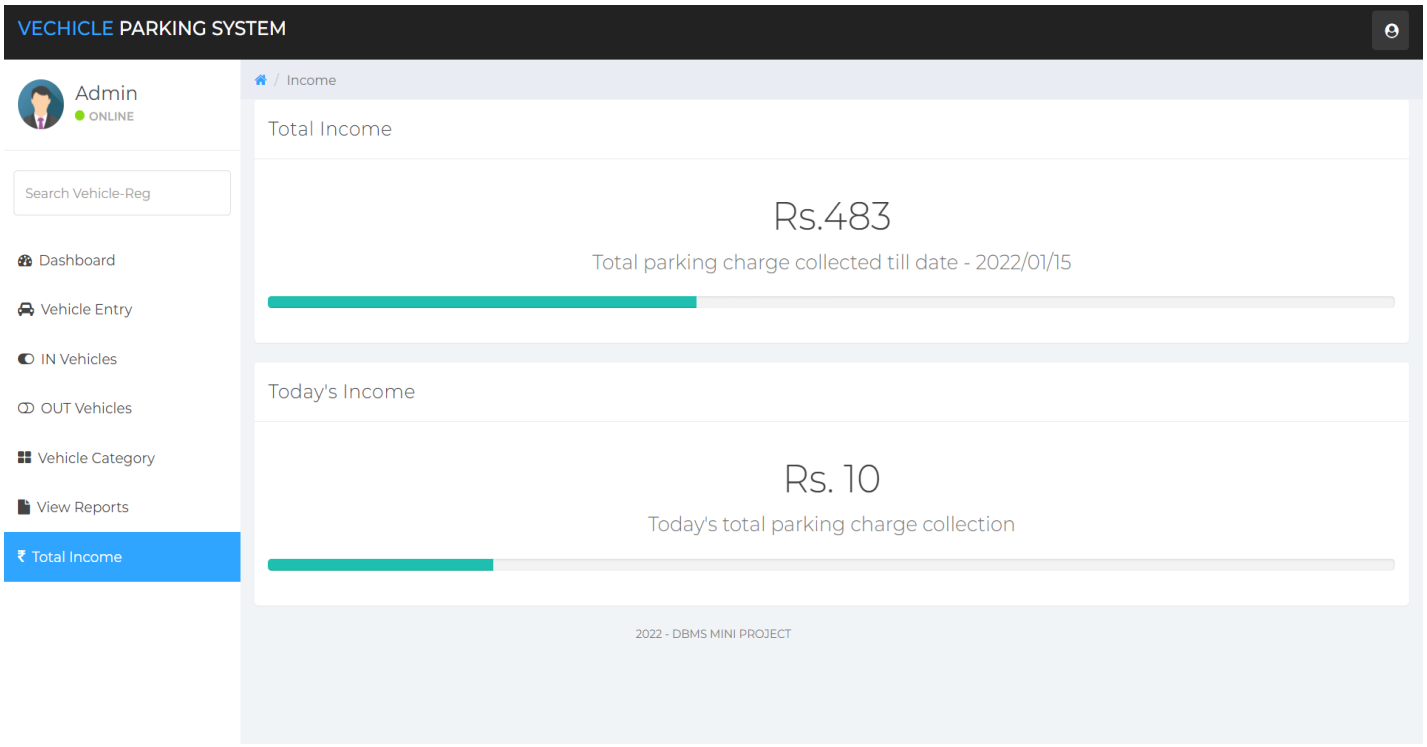

Fig: 5.1.14 Total income

With the total income feature, total income earned till date as well as that days income can be seen by admin.

5.2 DATABASE SNAPSHOTS

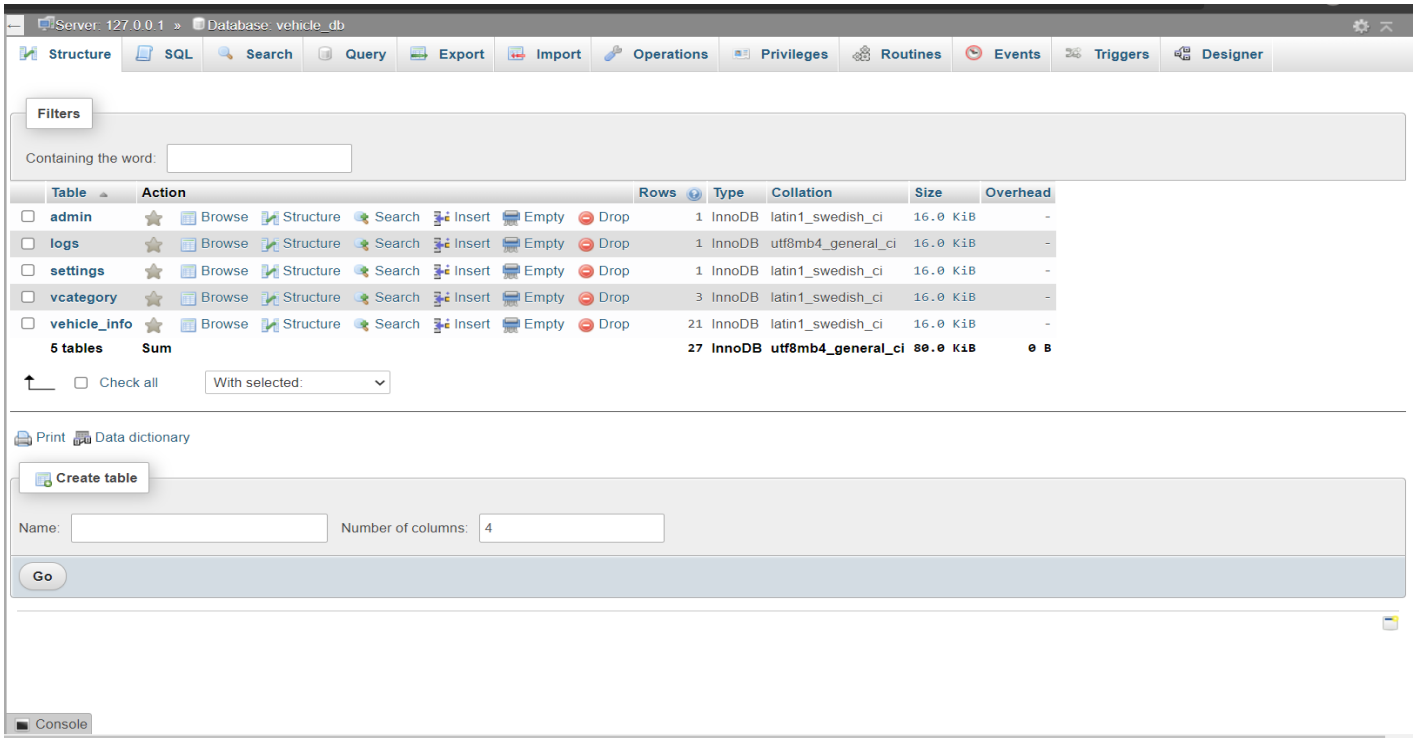

Fig: 5.2.1 List of tables in Vehicle_Database

The relations that are present in Vehicle_Database are listed. There are five relations required and they are admin, logs, company settings, vehicle category and vehicle details.

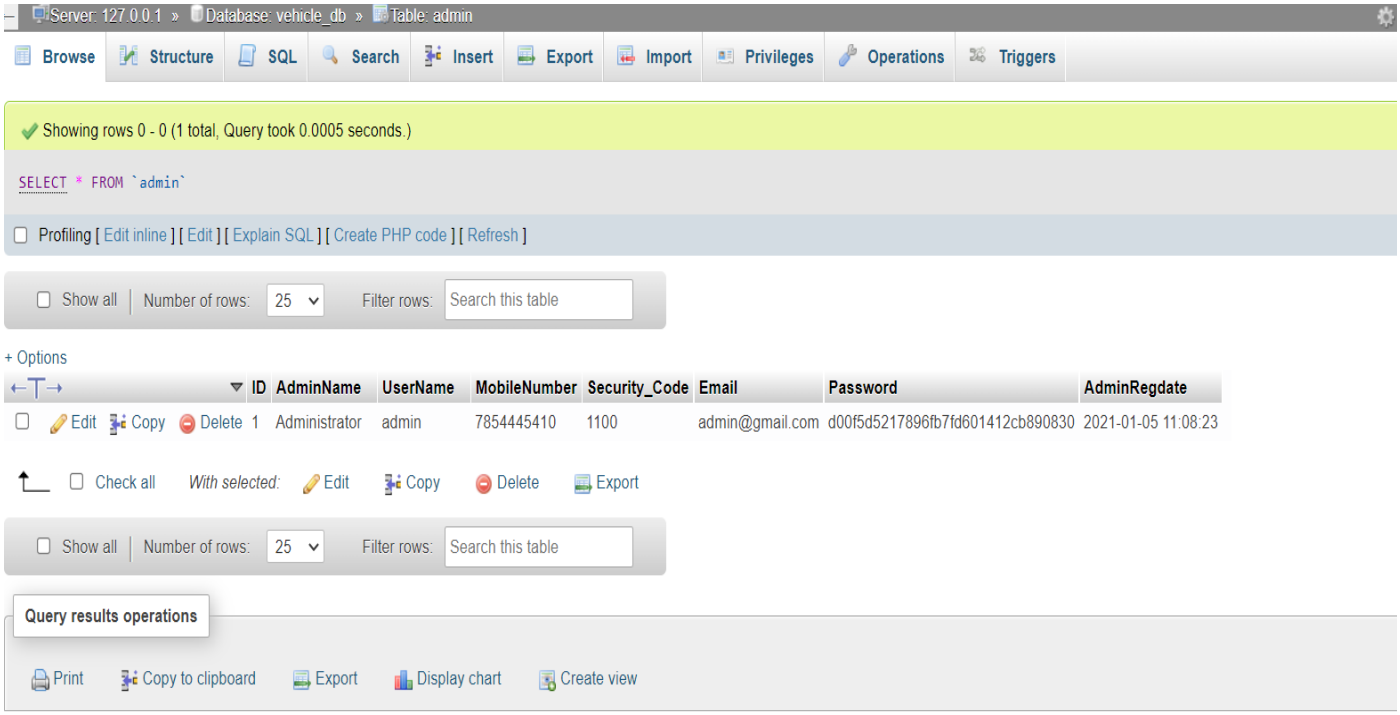

Fig: 5.2.2 Content of Admin Table

In admin table the key element is security code. Here the security code given by the company is known only by the admin.

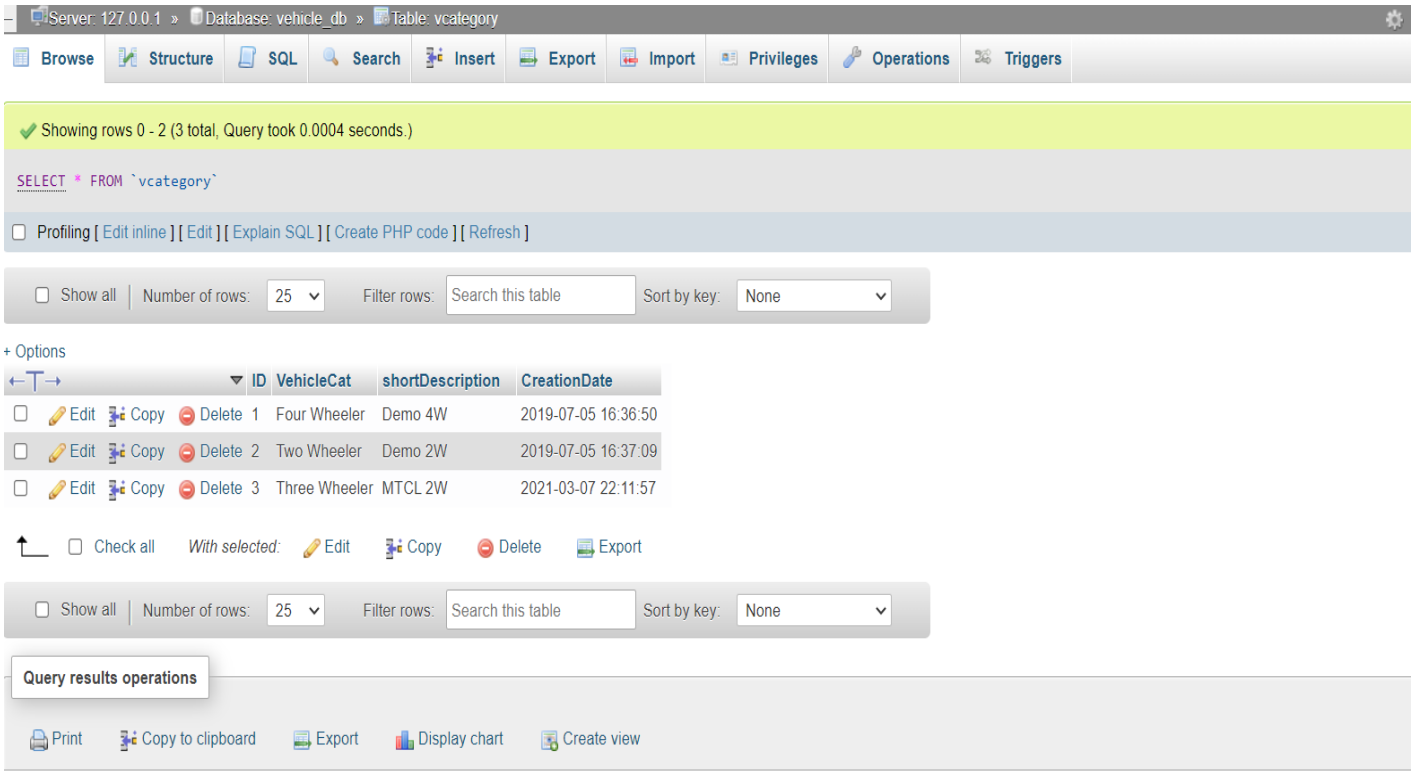

Fig: 5.2.3 Content of Vehicle Category Table

There are three vehicle categories entered in the relation. We can further add the vehicle category or delete the existing category.

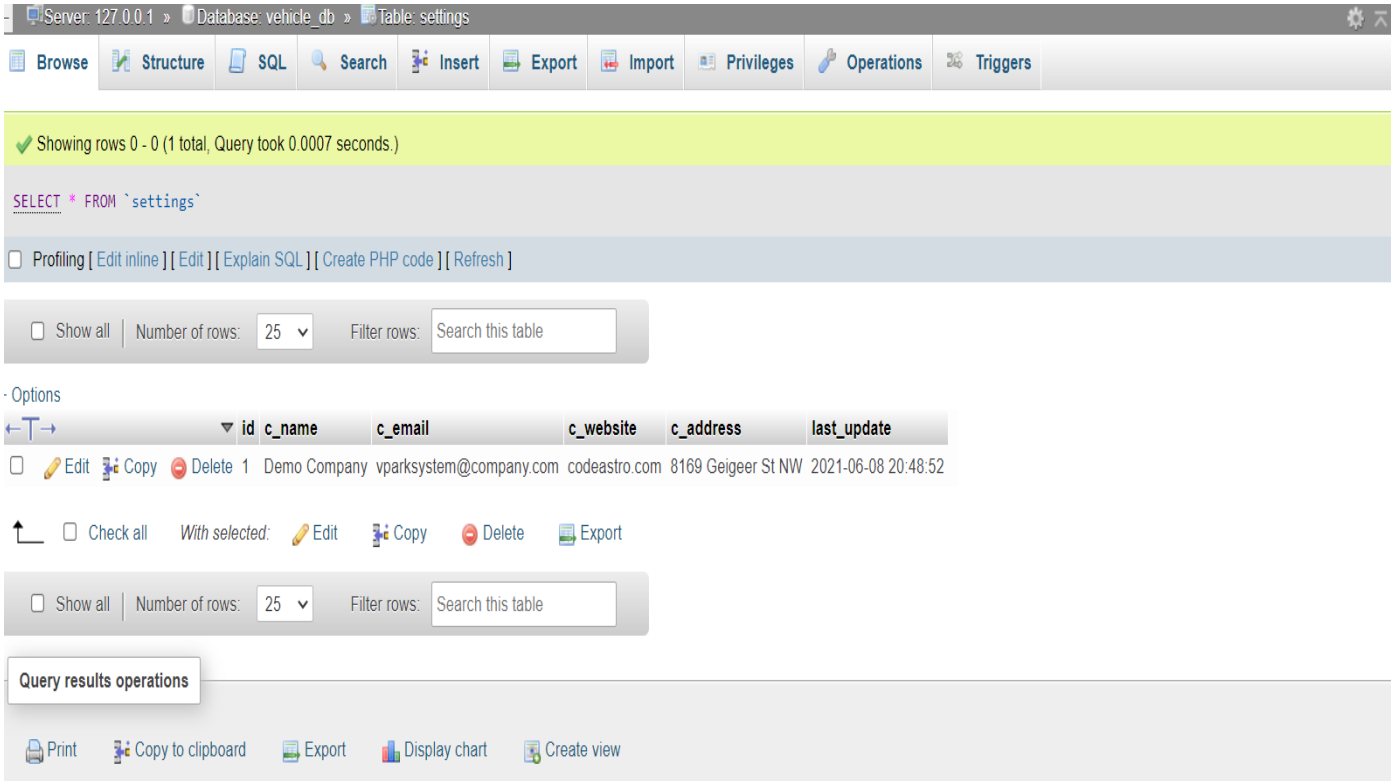

Fig: 5.2.4 Content of Company Table

This relation contains company details like company name, email ID, website, address. We have considered one company with one admin which is one-one relationship.

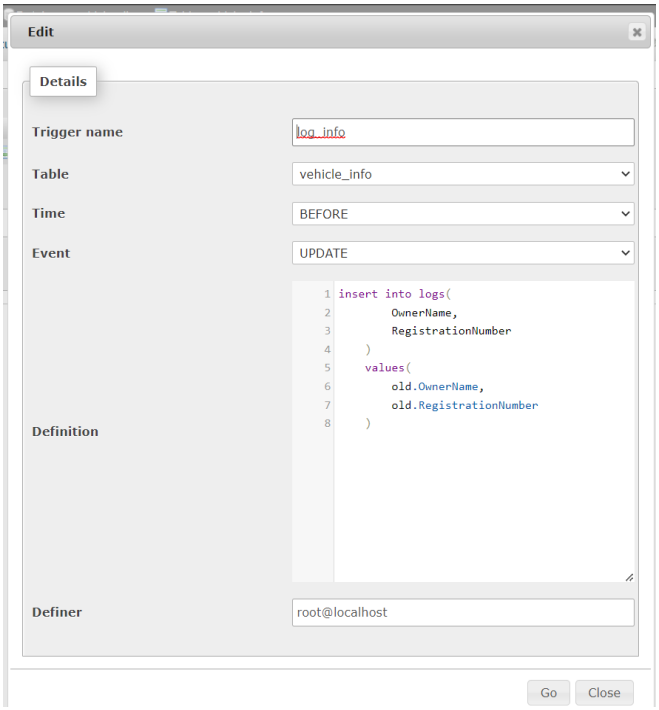

Fig: 5.2.5 Log Table is updated using trigger

A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. The contents of log is updated using trigger. In log table, only owner name and vehicle registration number is listed. For quick retrieval of details this relation can be used.

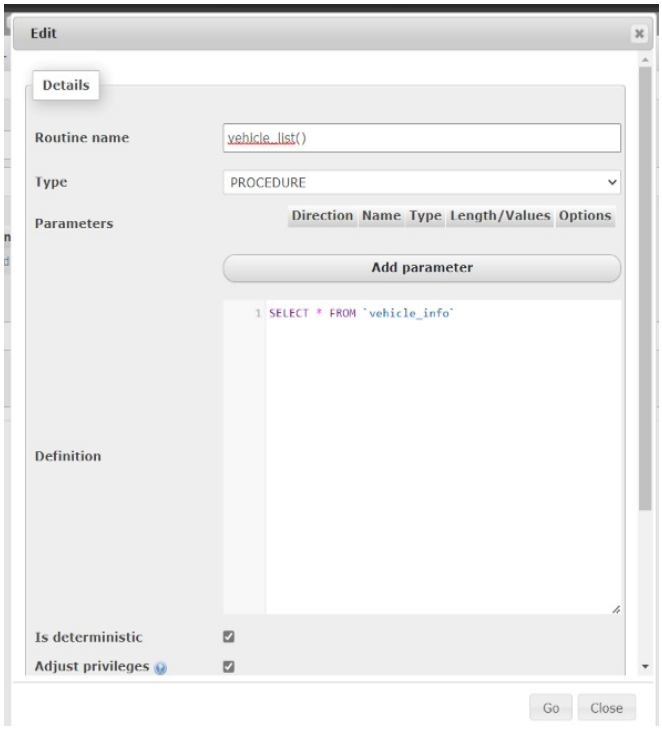

Fig: 5.2.6 Demonstration of procedure

A stored procedure is a prepared SQL code that we can save, so the code can be reused over and over again. We have used procedure in order to list the vehicles information.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Vehicle Parking System in PHP focuses mainly on keeping track of vehicle's parking. Also, the system displays all the vehicle's entry and outgoing records, along with adding vehicle categories. This system reduces manual labour and investment and ensures safety of vehicles. The efficiency of the developed system can be enhanced with some minor modifications like implementing with Automated Parking System.

6.2 LIMITATION AND FUTURE WORK

This Vehicle Parking System have manual entry of parking charge. So in future we need to determine and update parking charge based on vehicle-in-time and vehicle category. And also to automate the system.

BIBLIOGRAPHY

[1] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson

[2] Database Management Systems, Ramakrishnan and Gehrke, 3rd Edition, 2014, McGraw Hill

[3] Silberschatz Korth and Sudharshan: Database System Concepts, 6th Edition, Mc-Graw Hill, 2013.

[4] HTML, CSS & JavaScript Web Publishing by Laura lemay, Rafe Colburn, Jennifer Kyrnin

[5] HTML & CSS design and build websites by Jon Duckett

[6] Learning PHP, MySQL & JavaScript: With JQuery, CSS & HTML5 by Robin Nixon

[7] https://www.w3schools.com/php/default.asp

[8] https://www.geeksforgeeks.org/php-tutorials/