VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A DBMS Mini-Project

ON

"VEHICLE PARKING SYSTEM"

A Mini project report submitted in partial fulfillment of the requirements for the

5 th semester of ***Bachelor of Engineering in Computer Science Engineering***

Of Visvesvaraya Technological University, Belagavi

Submitted by

SRUJAN M S AATHREYA 1ST20CS122

Under the guidance of

Dr. Sanjeetha. R

Associate Professor

Department of Computer Science & Engineering

SAMBHRAM INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SAMBHRAM INSTITUTE OF TECHNOLOGY

M.S. PALYA, JALAHALLI, BENGALURU 560097.

2022-2023

SAMBHRAM INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the mini project work entitled **"Vehicle Parking System''** has been successfully carried out by Srujan M S, bearing USN 1ST20CS122, bonafide student of **Sambhram Institute of Technology** in partial fulfilment of the requirements for the **5th semester** of Bachelor **of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the laboratory requirements of 5th semester BE, CSE.

Dr. Sanjeetha. R	**Dr. T John Peter**
Associate Professor	**HOD**
Dept. of CSE	**Dept. of CSE**

External Viva:

Name of the Examiners

1.

2. **Signature with Date**

ACKNOWLEDGEMENT

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 me in carrying out this project work. We would like to take this opportunity to thank them all.

I would like to thank **Dr. H.G. Chandrakanth**, Principal, SaIT, Bangalore, for his moral support towards completing my project. I would like to thank **Dr. T. John Peter**, Prof & Head, Department of Computer Science & Engineering, SaIT, Bangalore, for his valuable suggestions and expert advice. I deeply express my sincere gratitude to my guide to Prof. **Dr. Sanjeetha. R** Associate Professor, SaIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

I would like to thank all the teaching and non-teaching staff of Department of Computer Science & Engineering, SaIT, Bengaluru for their constant support and encouragement.

Date:

Place: Bengaluru **SRUJAN M S AATHREYA**

TABLE OF CONTENTS

List of Figures

List of Tables:

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.

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 vehicle Parking Management system is a web-based technology that will manage the records of the incoming and outgoing vehicles in an parking house. It's an easy for Admin to retrieve the data if the vehicle has been visited through number he can get that data. Vehicle parking management system is an automatic system that delivers data processing in very high speed in a systematic manner.

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 Overview

This system vehicle parking management system is designed to book a parking space for a vehicle and admin can manage the parked vehicle by adding incoming vehicle and removing outgoing vehicle. Although financial transactions is done cash in hand method but our team believe that this project would help a lot in parking using computerized method instead of old parking system. And for the clients/drivers it would be easier for them to book a parking space online. This system Vehicle Parking Management System is designed to book a parking space for a vehicles and also admin can manage the parked vehicle by adding incoming vehicle and removing outgoing vehicle. Although financial transactions is done by cah in hand method but we team habb believe that this project would help a lot in parking using computerized method instead of old parking system. And for the clients/drivers it would be easier for them to book a parking space online

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 record of the total vehicle's entry with a number of in and out vehicles and total parking within a 24 hours span time.

1.3 Objectives

In other words we can say that out project has the following objectives

- Maintain records in short time of period
- Enhances the visitors experience
- Easy operations for operator of the system
- Centralized database management
- Reduce time consumption
- No paper work requirement.

2. LITERATURE SURVEY

Vehicle Parking System using PHP program has lots of codes, Using Internet in gathering information partially contributed to the success of this project. Due to the fact that PHP is an open source program. development of hostel management system was not too difficult. However, thanks to the cyber world (Internet) that makes it possible to study and make comparison in needs of some code function

2.1 Tools and Technologies

The Following Project "Vehicle Parking System" is built using three different platforms such as Front-end, Back-end associated with Database Technology & Server Setup Applications. For a basic web page we have chosen HTML i.e. the hypertext markup language, we have made use of CSS i.e. cascading style sheets for logic and alignment and to enhance the look we have used a bit of JavaScript.

- **HTML:** is a standard markup language for documents designed to be displayed in a web browser HTML is used to create static pages with text, tables, lists, images, links, and so on. HTML can be assisted by technologies such as Cascading Style Sheets (CSS).

- **CSS:** CSS is designed to enable the separation of content and presentation, including layout, colours, and fonts. This separation can improve content accessibility provide more flexibility and control in the specification of presentation characteristics ,CSS is used to style and layout web pages that is to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.

- **JavaScript:** is a programming language that is often used as client side programming language and also to create dynamic an interactive web content like applications in browsers. Which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without communicating with the server.

- **My SQL**: is an application under the Oracle Corporation, which is not only used in small scale industry but also used in the large scale industries. MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. 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.

- **PHP** formerly known as Personal Home Page which now abbreviates as Hypertext Pre-processor is a programming language used to create web servers, that works with Apache to help to create dynamic web content 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.

- **Apache** is a free and open-source software that allows users to deploy their websites on the Internet. Apache is not any physical server; it is software that executes on the server. However, we define it as a web server. Apache is software that is highly **customizable**. It contains the module-based structure. Various modules permit server administrators for turning additional functionality off and on. Apache includes modules for caching, security, password authentication, URL rewriting, and other purposes

3. REQUIREMENT SPECIFICATION

3.1 Functional Requirements

Interface of the system, the services and feature of the system.User can check whether the system provides the services according to the requirements that they proposed.In this project, we use PHP and Mysql database.

Admin Module:

Dashboard: In this section, admin can see all detail in brief like the total registered user, Total Enquiry, Total Mechanics and Total Services.

Enquiry Search: In this section, admin can search enquiry with the help of customer phone number, email id or contact number.

Service Search: In this section, admin can search for services with help of customer phone number, email id or contact number.

Mechanics: In this section, admin can manage mechanics (add, delete, and update).

Vehicle Category: In this section, admin can manage vehicle category (add, delete and update).

Reg Users: In this section, admin can view user detail and update user detail.

Service Request: In this section, admin can add service prices (service charge, parts charge additional (if any)). Admin can view services on the basis of status (pending services, rejected services and complete services). Admin also can approve pending policy

Customer Query: In this section, admin can respond to the customer query and also view the customer responded query and none responded query.

3.2 Non - Functional Requirements

3.2.1 Software Requirements

The software requirements are as follows:

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

3.2.2 Hardware Requirements:

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)

3.3 Database Requirements

Apartment building has many parking spaces, Each parking spaces have radium parking lights. Parking Managed by apartment association. Those who wants to park their vehicles will contact association manager and other issues. Entity parking system have attributes such as Registration_Num Entity Vehicles and Entity Vehicle_category are subclasses, defined under the superclass. Logs sharing common attributes such as owner_name, Reg_number.

Entity Logs is a weak entity of parking system having attributes owner_name, reg_id. Entity vehivle_category have attributes category_id, creation_date, vehicle_type, in this vehicle_type is a primary key. Total participation is obtained at admin, vehicle category, company. Entity admin has admin_ID, smob_number, admin_name, admin_email, reg_date.

4. SYSTEM DESIGN

4.1 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.

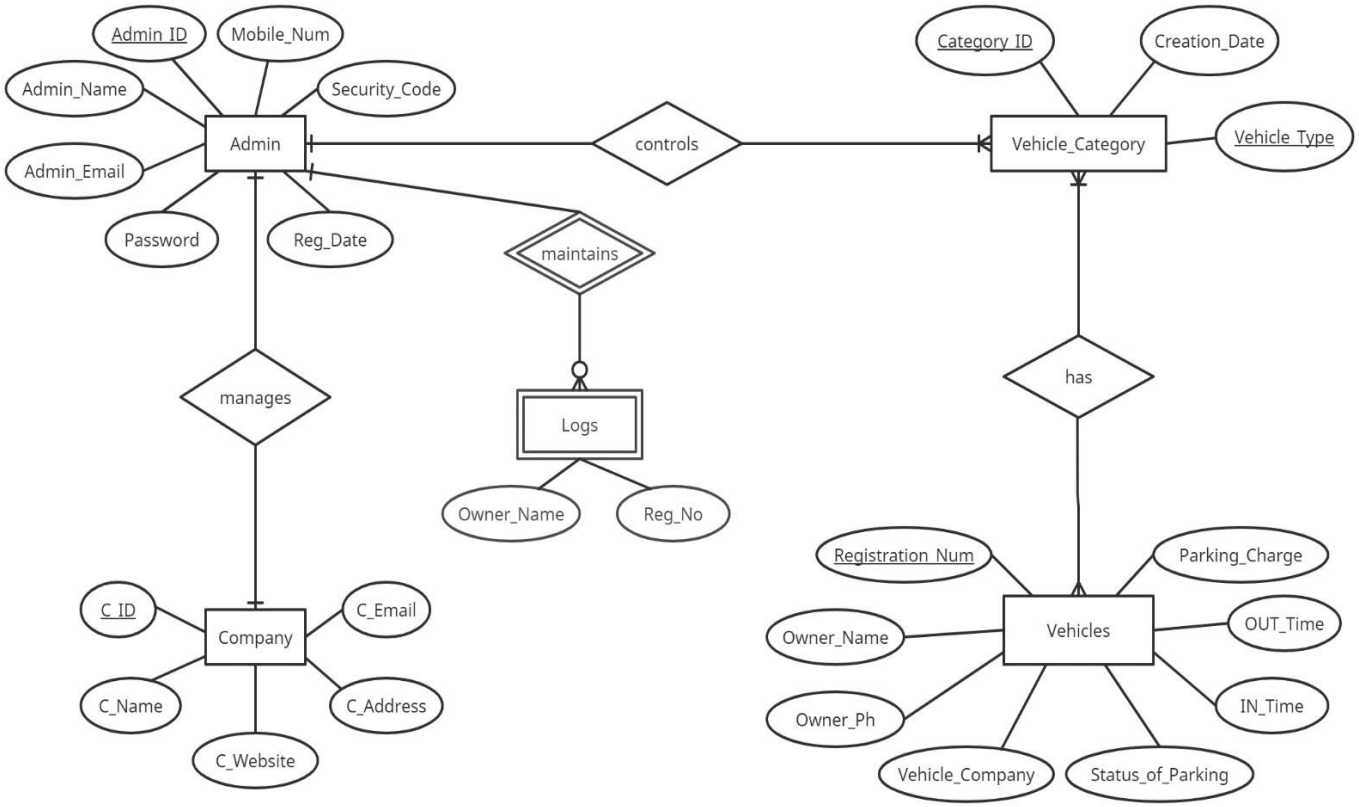

Figure 4.1.1 : E-R Diagram of Vehicle Parking System

4.2. RELATION SCHEMA

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.

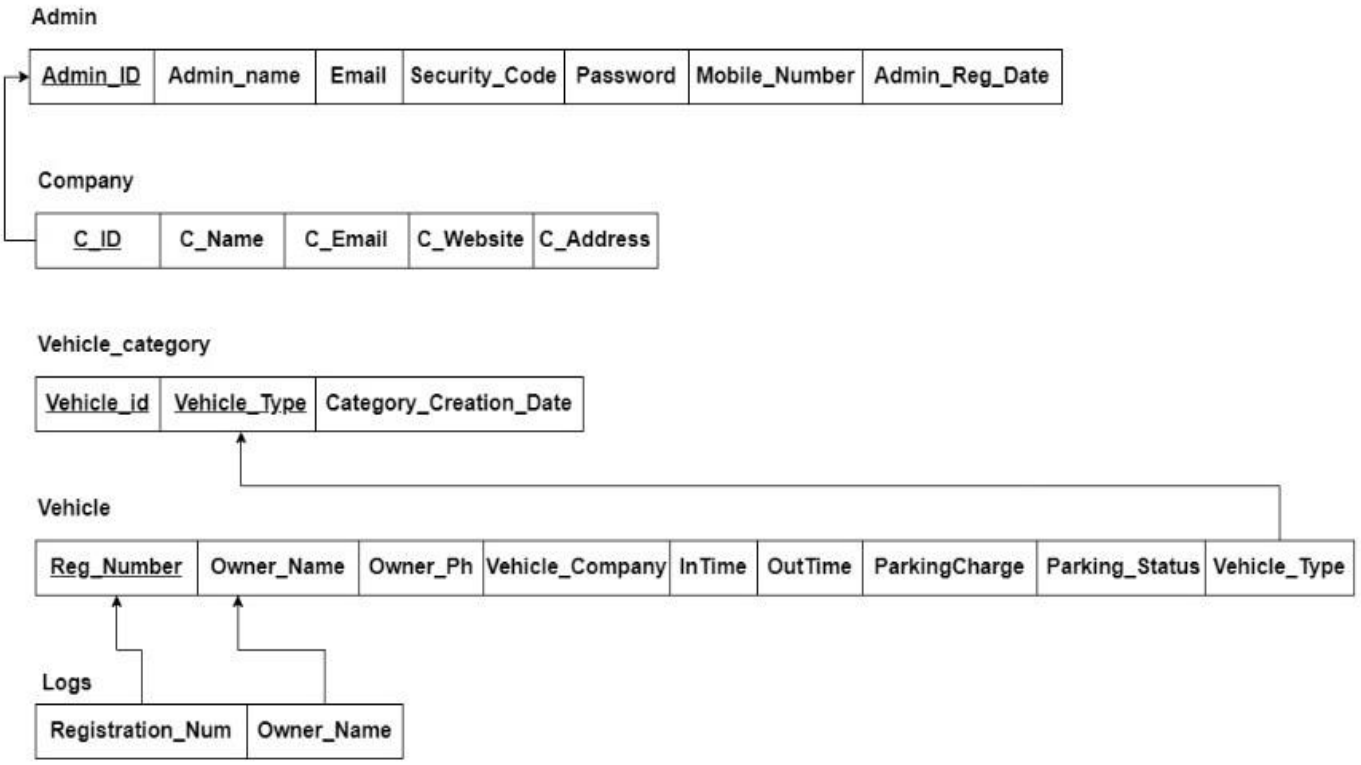

Figure 4.1.2 : Relation Schema of Vehicle parking System

5. IMPLEMENTATION

Cost Benefit Analysis: Cost benefit analysis (CBA) estimates and total up the equivalent money value of the benefits and the cost invested to for implementation the software. Cost benefit analysis(CBA) is the weighing scale approach to decision-making. All the plus points (such as cash flow and other intangible benefits) are put on one side all the minus points (the cost and disadvantages) are put on the other side. Both sides should be weighed and benefits should be evaluated.

Cost Estimation: A cost estimate is the approximation of the cost of a program, project, or operation. The cost estimates is the product of the cost estimating process. The cost estimate has a single total value and may have identifiable component values. For a given set of requirements, it is desirable to know how much it will cost to develop the software to satisfy a given requirement, and how much time development will take. The cost of a project is a function of many parameters. Foremost among them is the size of the project. Other factors that effects the cost are programmer ability, experience of the developers in the area complexity of the project, and reliability requirements of the software, hardware and human resources

Benefits:

- Improves business processes leading to annual cost decrease.
- Due to availability of information, better decision making is possible leading to additional cash flows.

5.1 Frontend

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>
```

Figure 5.1.1: Code for out vehicle

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 }  ?>
```

Figure 5.1.2: Code for in vehicle

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 btnsm"><i class="fa fa-trash"></i></button> </a>
            </td>
            </tr>
                <?php $cnt=$cnt+1;}?>
        </tbody>
    </table>
```

Figure 5.1.3: Code for Vehicle Category

5.2 Backend

5.2.1: Details of admin table:

```
CREATE TABLE `admin` (
  `ID` int(10) NOT NULL,
  `AdminName` varchar(120) DEFAULT NULL,
  `UserName` varchar(120) DEFAULT NULL,
  `MobileNumber` bigint(10) DEFAULT NULL,
  `Security_Code` int(55) NOT NULL,
  `Email` varchar(200) DEFAULT NULL,
  `Password` varchar(120) DEFAULT NULL,
  `AdminRegdate` timestamp NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `admin` (`ID`, `AdminName`, `UserName`, `MobileNumber`, `Security_Code`, `Email`, `Password`, `AdminRegdate`) VALUES
(1, 'Administrator', 'admin', 7854445410, 1100, 'admin@gmail.com', 'd00f5d5217896fb7fd601412cb890830', '2021-01-05 05:38:23');
```

Table 5.2.1: Details of Admin Table

ID	AdminName	UserName	MobileNumber	Security_Code	Email	Password	AdminRegdate
1	Administrator	admin	7854445410	1100	admin@gmail.com	e807f1fcf82d132f9bb018ca6738a19f	Tue Jan 05 2021 11:08:23 GMT+0530 (India Standard Time)

5.2.2: Details of Setting table:

```
CREATE TABLE `settings` (
  `id` int(11) NOT NULL,
  `c_name` varchar(255) NOT NULL,
  `c_email` varchar(55) NOT NULL,
  `c_website` varchar(55) NOT NULL,
  `c_address` varchar(255) NOT NULL,
  `last_update` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `settings` (`id`, `c_name`, `c_email`, `c_website`, `c_address`, `last_update`) VALUES
(1, 'Demo Company', 'vparksystem@company.com', 'codeastro.com', '8169 Geigeer St NW', '2021-06-08 20:48:52');
```

Table 5.2.2: Details of Setting Table

id	c_name	c_email	c_website	c_address	last_update
1	Curio	info.curio@company.com	curio.com	Bangalore	Tue Jun 08 2021 20:48:52 GMT+0530 (India Standard Time)

5.2.3: Details of v_category table:

```
CREATE TABLE `vcategory` (
  `ID` int(10) NOT NULL,
  `VehicleCat` varchar(120) DEFAULT NULL,
  `shortDescription` varchar(50) NOT NULL,
  `CreationDate` timestamp NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


INSERT INTO `vcategory` (`ID`, `VehicleCat`, `shortDescription`, `CreationDate`) VALUES
(1, 'Four Wheeler', 'Demo 4W', '2023-01-05 11:06:50'),
(2, 'Two Wheeler', 'Demo 2W', '2023-01-05 11:07:09'),
(3, 'Two Wheeler', 'MTCL 2W', '2023-01-07 16:41:57');
```

Table 5.2.3: Details of v_category table

ID	VehicleCat	shortDescription	CreationDate
1	Four Wheeler	Demo 4W	Mon Jan 16 2023 16:36:50 GMT+0530 (India Standard Time)
2	Two Wheeler	Demo 2W	Thu Jan 05 2023 16:37:09 GMT+0530 (India Standard Time)
6	Two Wheeler	MTCL 2W	Sun Jan 08 2023 22:11:57 GMT+0530 (India Standard Time)

5.2.4: Details of vehicle_info table:

```
CREATE TABLE `vehicle_info` (
  `ID` int(10) NOT NULL,
  `ParkingNumber` varchar(120) DEFAULT NULL,
  `VehicleCategory` varchar(120) NOT NULL,
  `VehicleCompanyname` varchar(120) DEFAULT NULL,
  `RegistrationNumber` varchar(120) DEFAULT NULL,
  `OwnerName` varchar(120) DEFAULT NULL,
  `OwnerContactNumber` bigint(10) DEFAULT NULL,
  `InTime` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `OutTime` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  `ParkingCharge` varchar(120) NOT NULL,
  `Remark` mediumtext NOT NULL,
  `Status` varchar(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `vehicle_info` (`ID`, `ParkingNumber`, `VehicleCategory`, `VehicleCompanyname`, `RegistrationNumber`, `OwnerName`, `OwnerContactNumber`,
 `InTime`, `OutTime`, `ParkingCharge`, `Remark`, `Status`) VALUES
(1, '96069', 'Four Wheeler', 'Hyundai', 'GGZ-1155', 'Jamie Macon', 8956232528, '2023-03-09 05:58:38', '2023-03-09 10:15:43', '34', 'NA', 'Out'),
(2, '52796', 'Two Wheeler', 'KTM', 'GTM-1069', 'Dan Wilson', 8989898989, '2023-03-09 08:58:38', '2023-03-09 14:16:26', '20', 'NA', 'Out'),
(3, '65023', 'Two Wheeler', 'Yamaha', 'JFF-7888', 'Lynn Roberts\n', 7845123697, '2023-01-09 08:58:38', '2023-03-09 12:16:31', '20', 'Vehicle Out', 'Out'),
(4, '90880', 'Two Wheeler', 'Suzuki', 'PLO-8507', 'Charles Mathew', 2132654447, '2023-01-09 08:58:38', '2023-03-09 13:58:38', '20', 'Vehicle Out', 'Out'),
(5, '09894', 'Two Wheeler', 'Piaggio', 'DLE-7701', 'Theresa Hay\n', 4654654654, '2023-01-09 08:58:38', '2023-03-09 14:58:38', '15', 'none', 'Out');
```

Table 5.2.4: Details of vehicle-info table

ID	ParkingNumber	VehicleCategory	VehicleCompanyname	RegistrationNumber	OwnerName	OwnerContactNumber	InTime	OutTime	ParkingCharge	Remark	Status
1	96069	Four Wheeler	Hyundai	GGZ-1155	Jamie Macon	8956232528	Mon Jan 09 2023 11:28:38 GMT+0530 (India Standard Time)	Mon Jan 09 2023 15:45:43 GMT+0530 (India Standard Time)	34	NA	Out
2	52796	Two Wheeler	KTM	GTM-1069	Dan Wilson	8989898989	Wed Jan 04 2023 14:28:38 GMT+0530 (India Standard Time)	Wed Jan 04 2023 19:46:26 GMT+0530 (India Standard Time)	20	NA	Out
3	65023	Two Wheeler	Yamaha	JFF-7888	Lynn Roberts	7845123697	Sun Jan 01 2023 14:28:38 GMT+0530 (India Standard Time)	Sun Jan 01 2023 17:46:31 GMT+0530 (India Standard Time)	20	Vehicle Out	Out
4	90880	Two Wheeler	Suzuki	PLO-8507	Charles Mathew	2132654447	Fri Jan 06 2023 14:28:38 GMT+0530 (India Standard Time)	Fri Jan 06 2023 19:28:38 GMT+0530 (India Standard Time)	20	Vehicle Out	Out
5	09894	Two Wheeler	Piaggio	DLE-7701	Theresa Hay	4654654654	Sat Jan 14 2023 14:28:38 GMT+0530 (India Standard Time)	Sat Jan 14 2023 20:28:38 GMT+0530 (India Standard Time)	15	none	Out

6. SNAPSHOTS

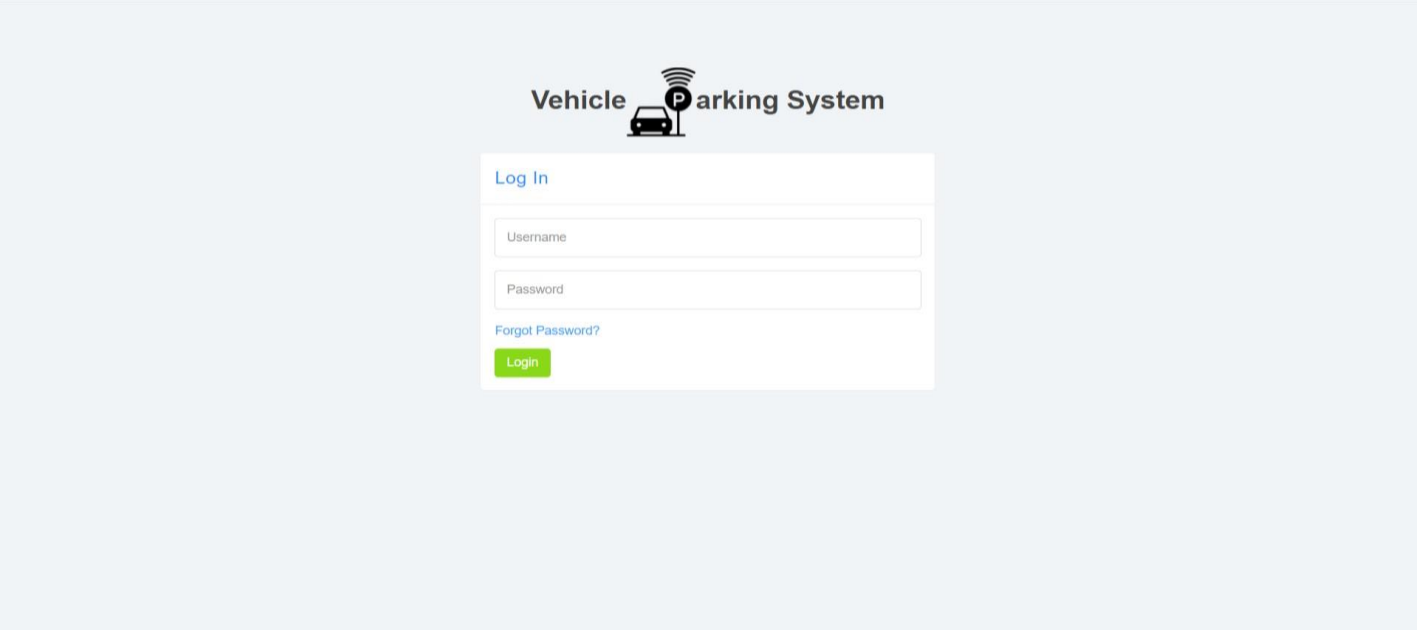

Figure: 6.1.1 Login Page

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.

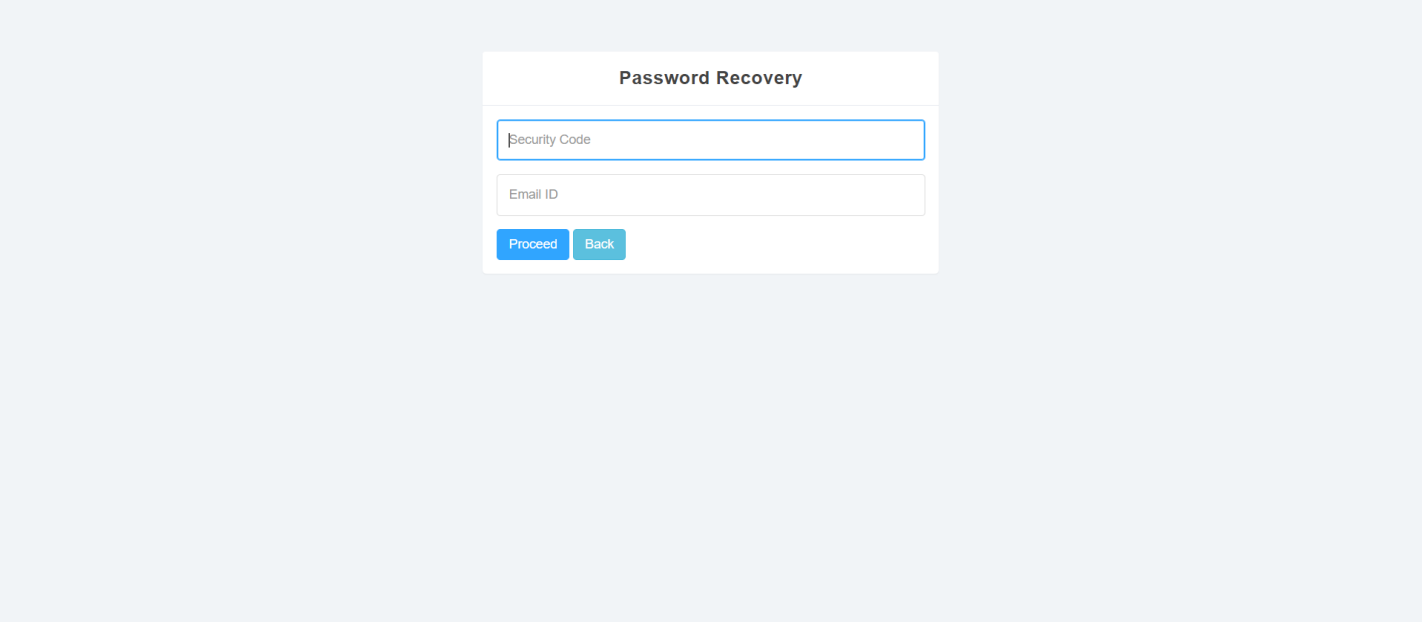

Figure: 6.1.2 Password Recovery

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

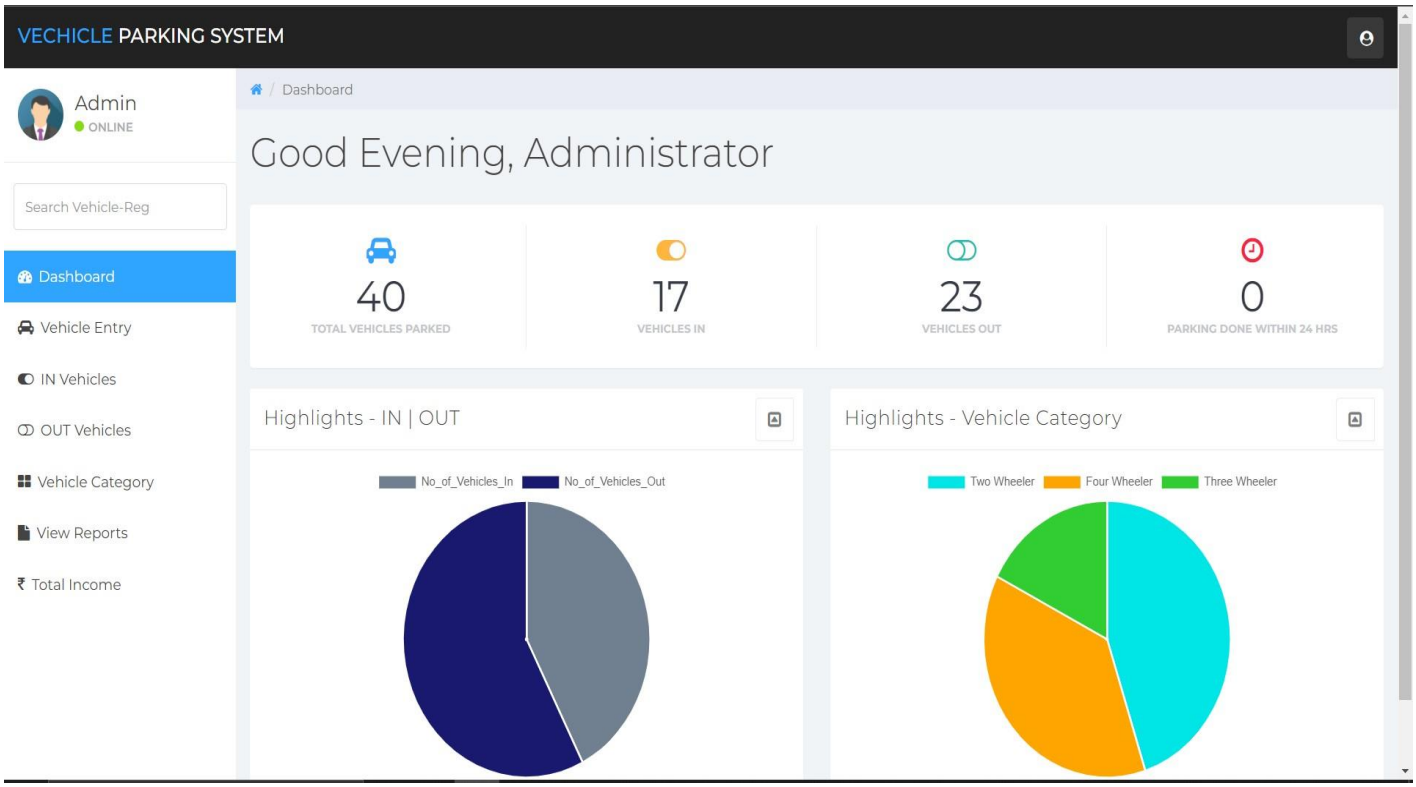

Figure: 6.1.3 Dashboard

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.

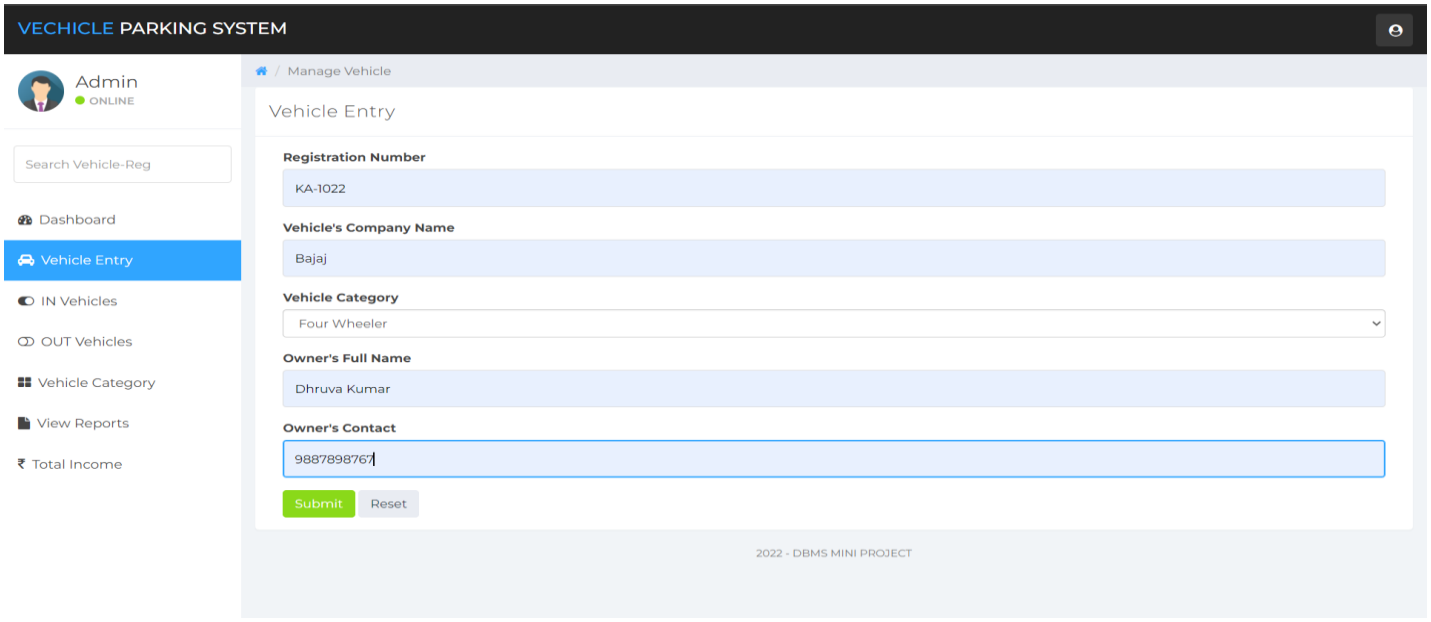

Figure: 6.1.4 Vehicle Entry

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

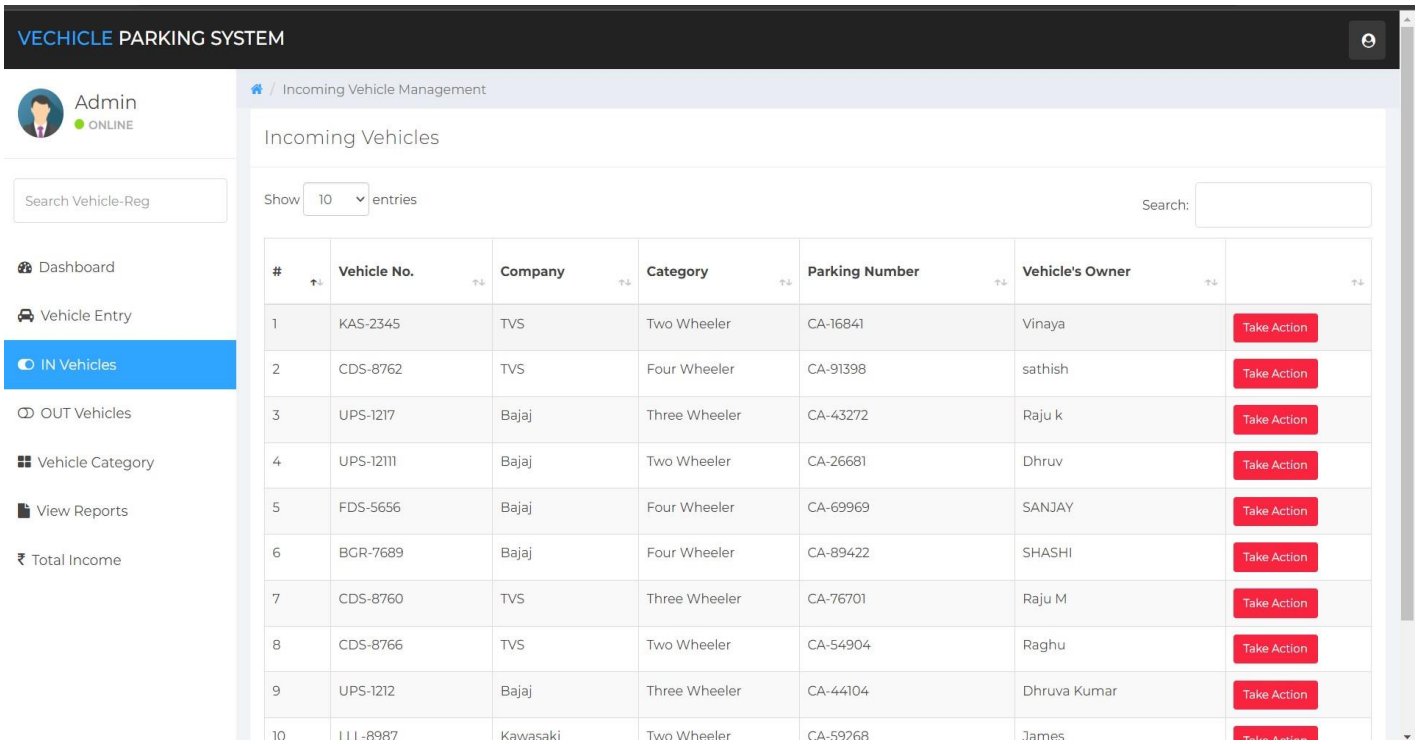

Figure: 6.1.5 List of in vehicles

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.

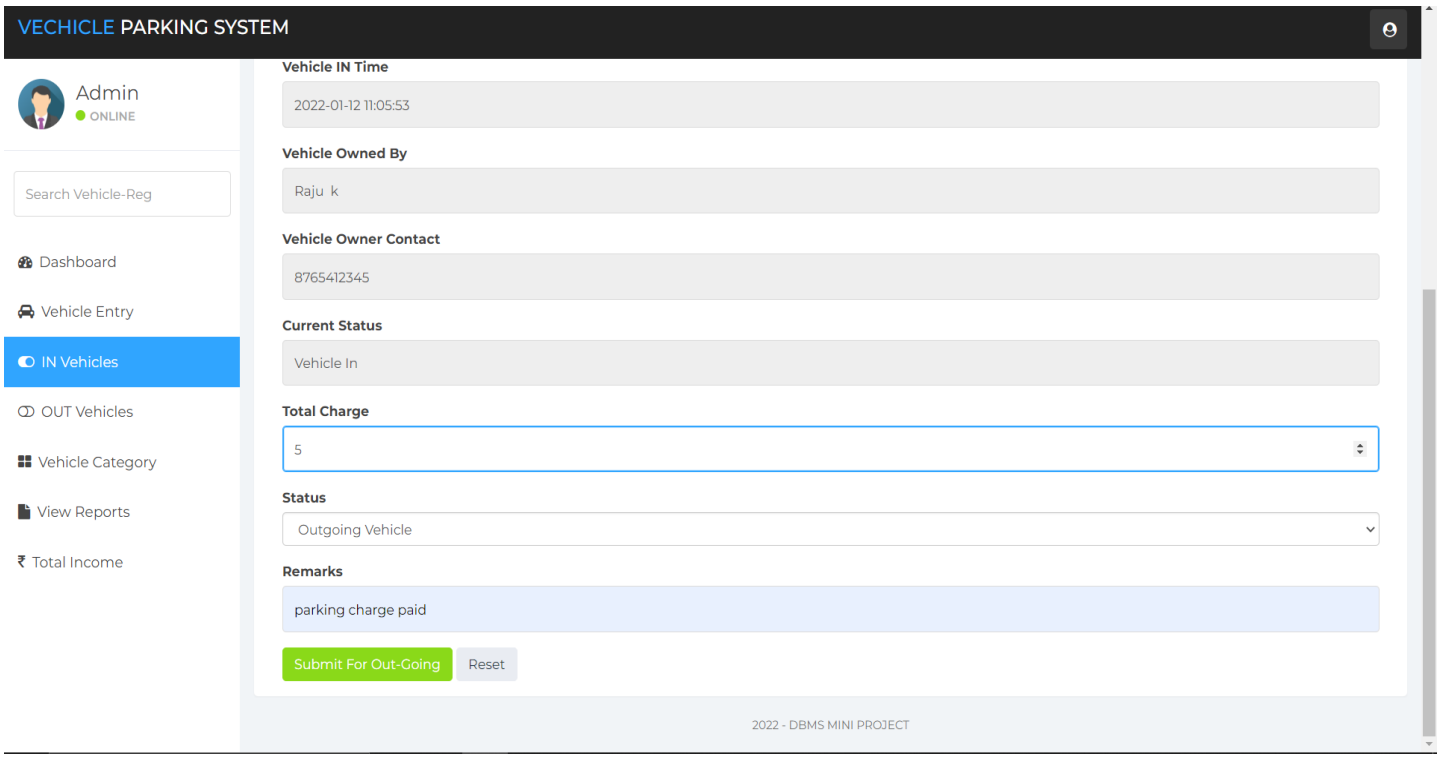

Figure: 6.1.6 Take action for in vehicles

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

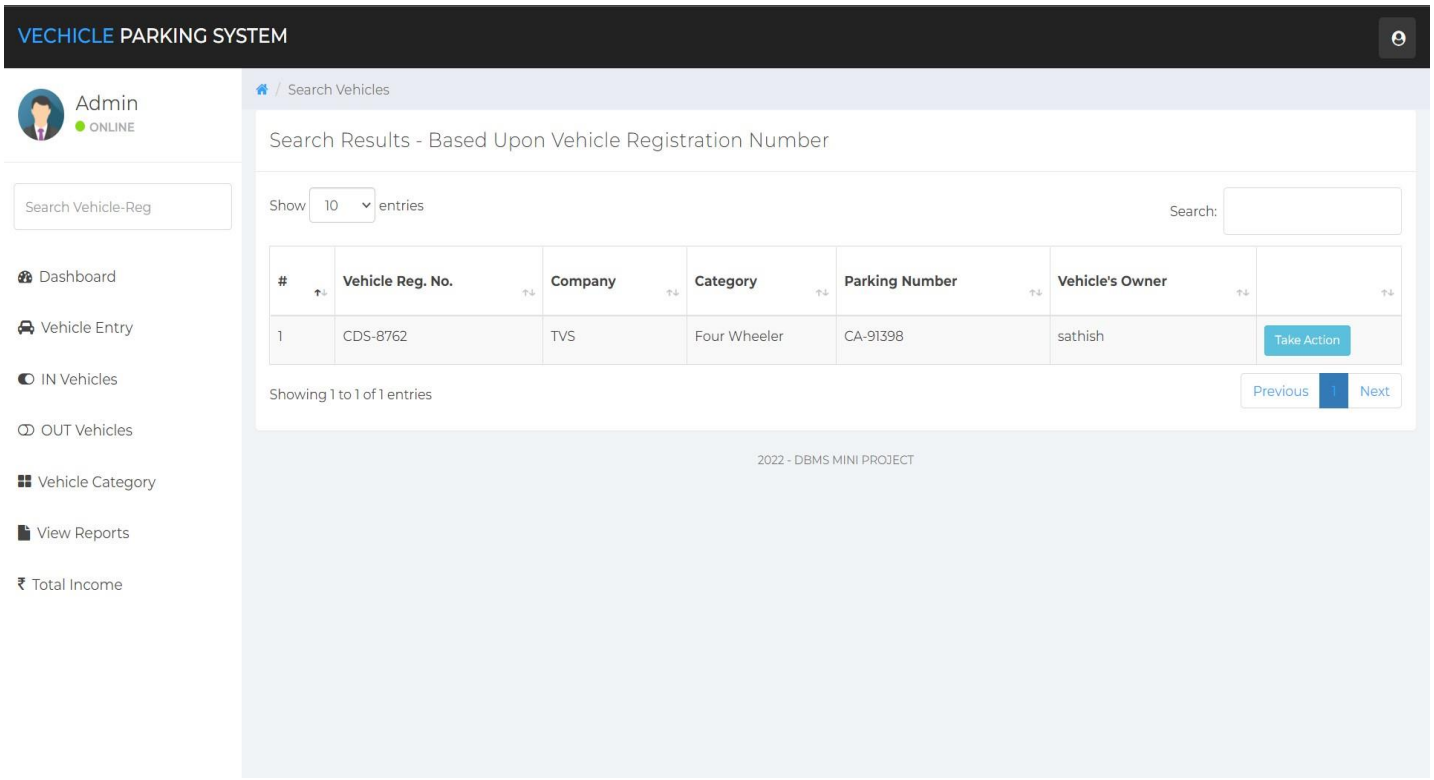

Figure: 6.1.7 Searching for a vehicle details

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

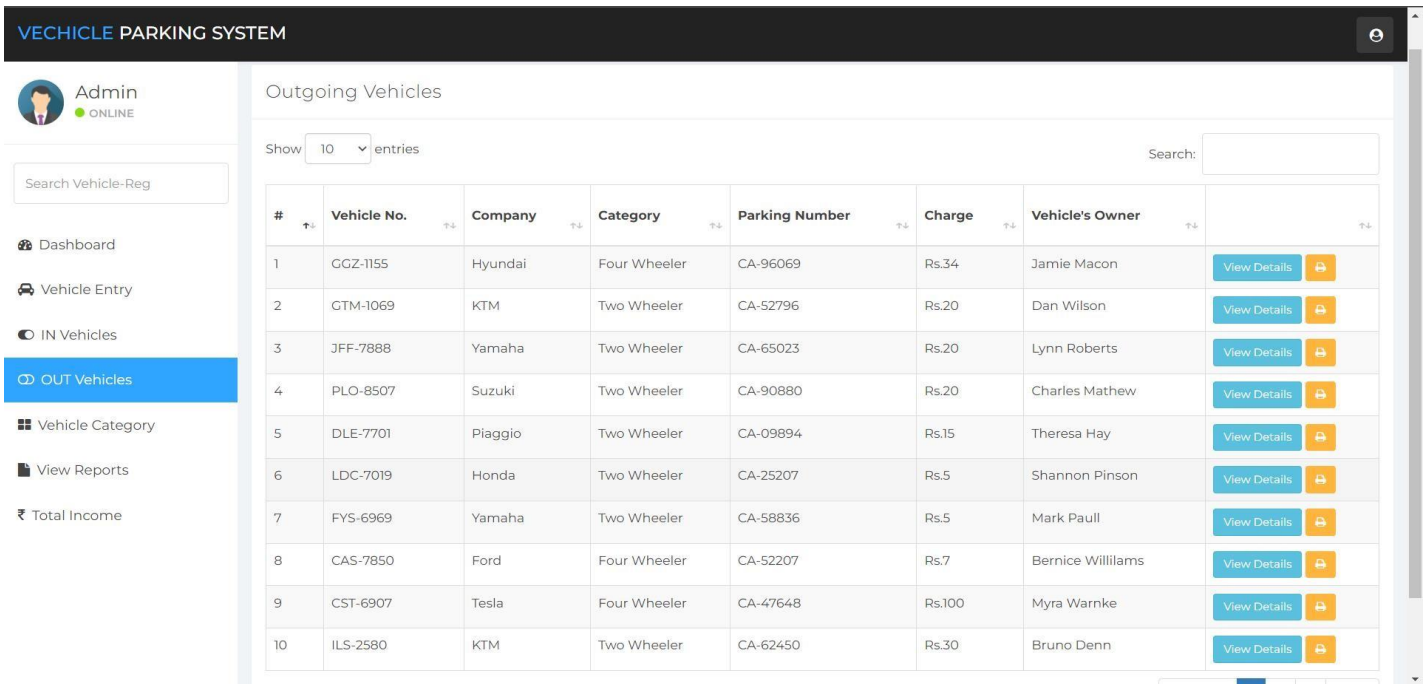

Figure: 6.1.8 List of out vehicles

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.

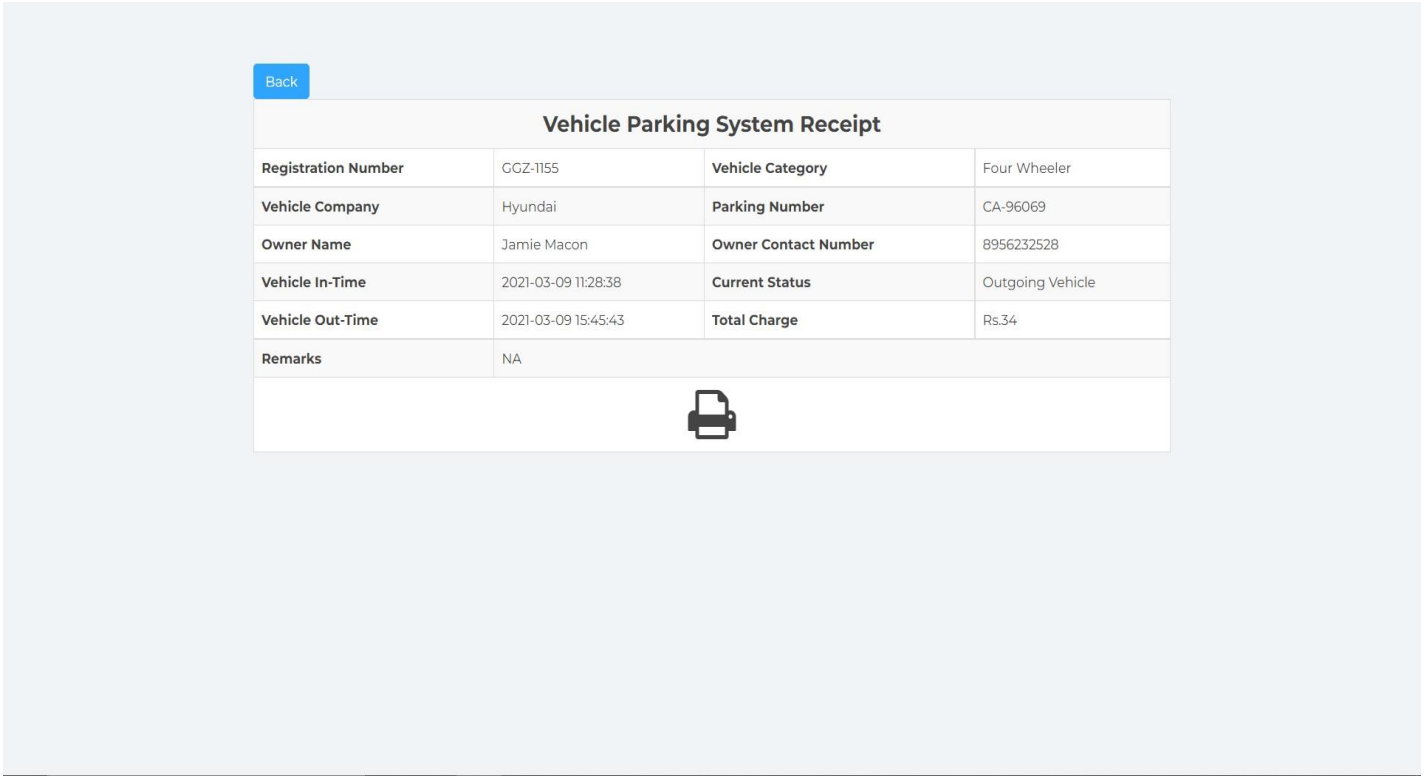

Figure: 6.1.9 Receipt

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.

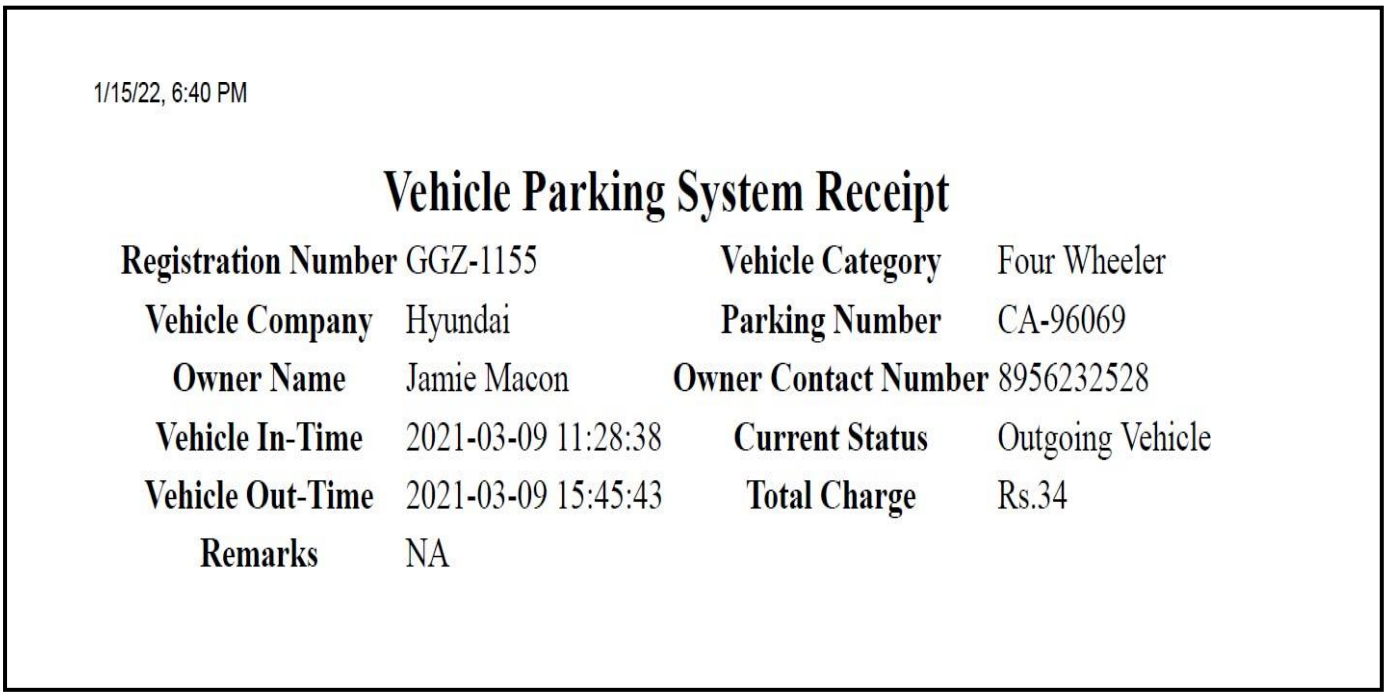

Figure: 6.1.10 PDF Copy of Receipt

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

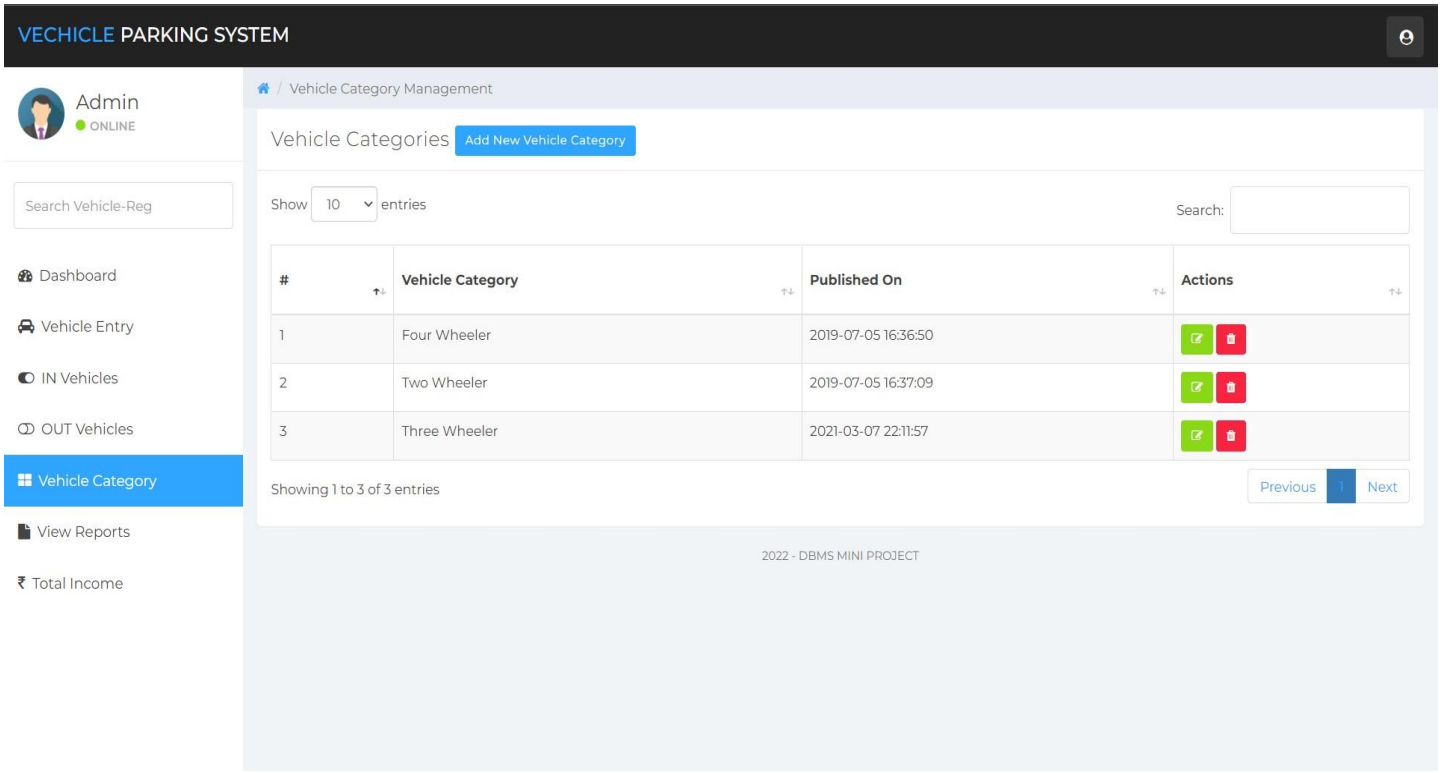

Figure: 6.1.11 List of vehicle categories

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.

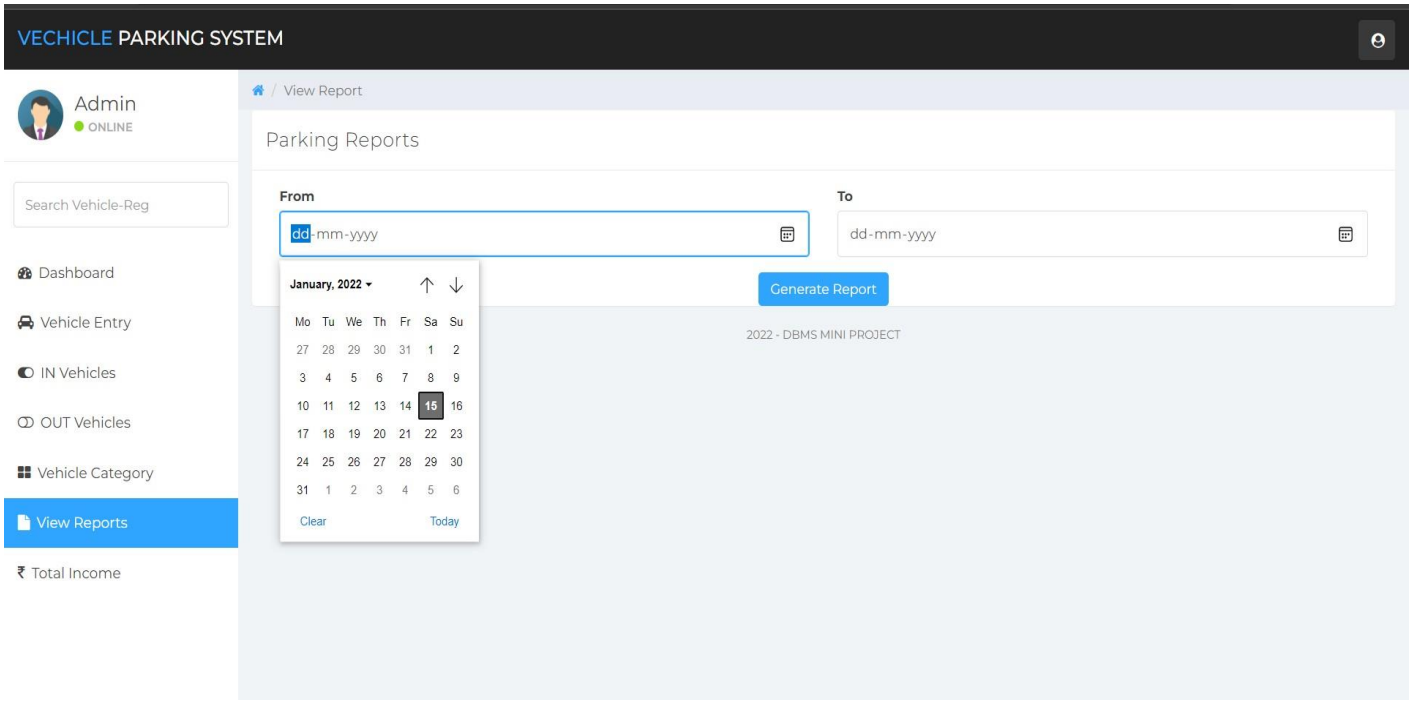

Figure: 6.1.12 View reports with in particular period

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

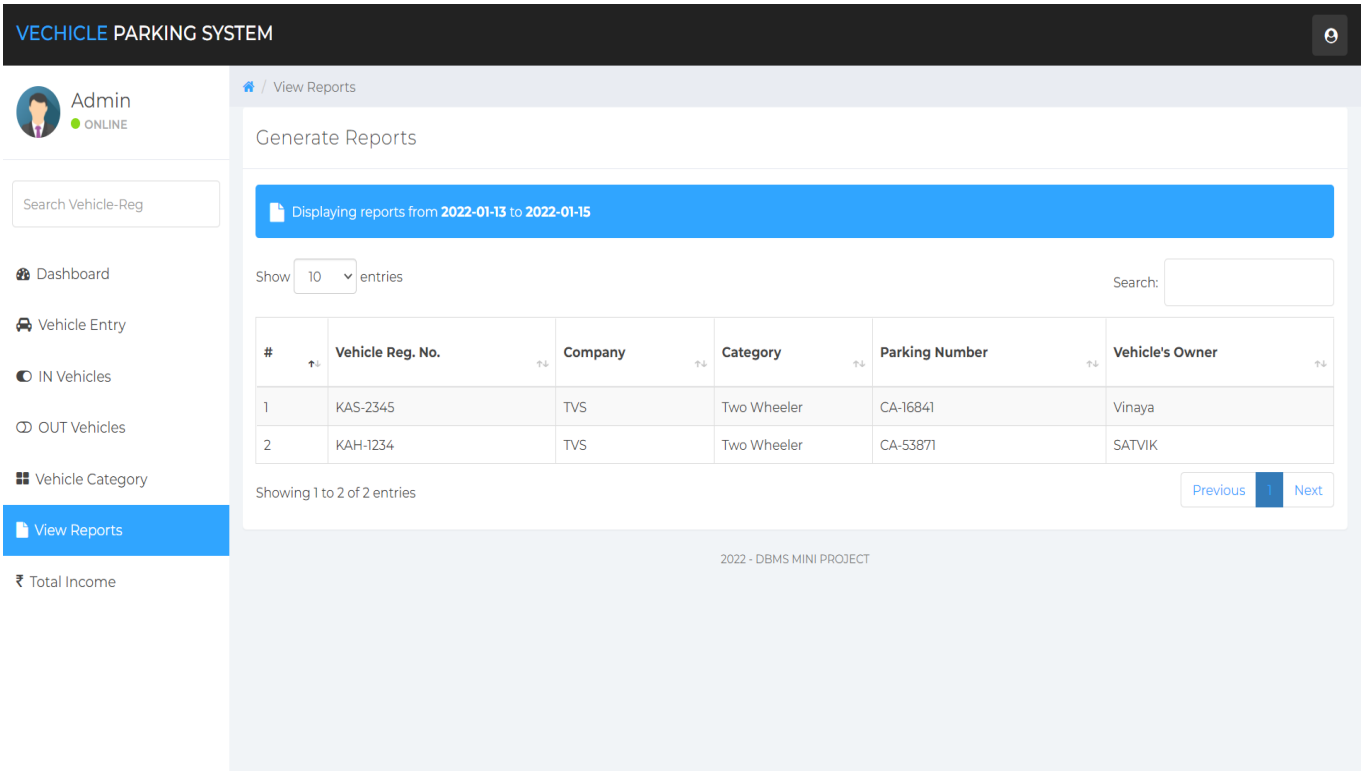

Figure: 6.1.13 Vehicle Report

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

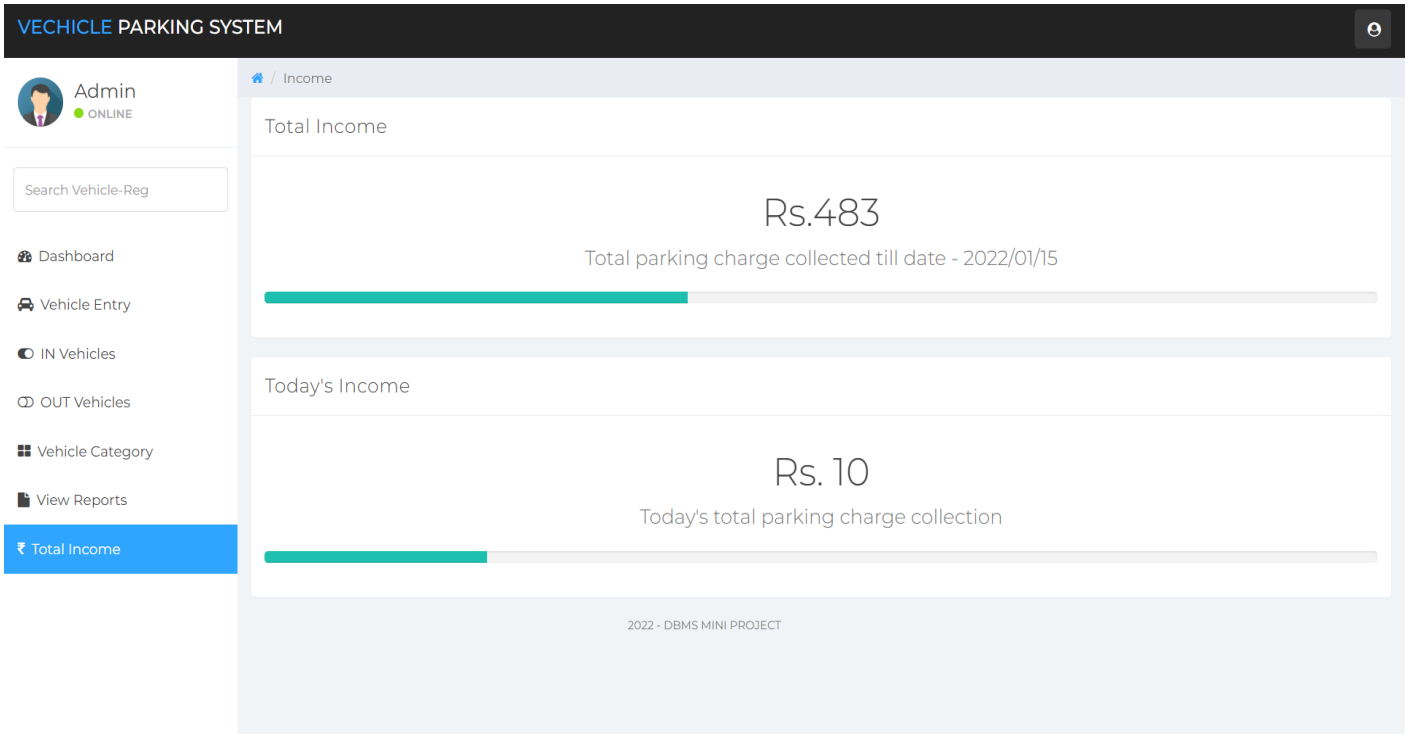

Figure: 6.1.14 Total income

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

7. Conclusion & Future Work

To conclude the description about the project: The project, developed using PHP and MySQL is based on the requirement specification of the users and the analysis of the existing system, with flexibility for future enhancement. The expanded functionality of today's software requires an appropriate approach towards software development. This hostel management software is designed for people who want to manage various activities in the hostel. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually. Identification of the drawbacks of the existing system leads to the designing of computerized systems that will be compatible to the existing system with the system which is more user friendly and more GUI oriented.

8. REFERENCES

[1] Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.

[2] Monson-Haefel, R. (2007). J2EE Web services. Boston. Mass: Addison-Wesley. Silberschatz A., Korth H. F., & Sudarshan S. (2011).

[3] Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.

[4] Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education,

[5] David F. (2011). JavaScript: The Definitive Guide Sixth edition.

[6] https://www.w3schools.com

[7] https://www.canvasjs.com

[8] https://getbootstrap.com/

[9] https://fontawesome.com