VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A DBMS Mini-Project Report
On

"Vehicle Management 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, Belagavi

Submitted by:

GALI KEERTHI REDDY	**1RN19CS052**
ALLAPARTHI SHARON KUMAR	**1RN19CS013**

Under the Guidance of:

Mrs.S Mamatha Jajur
Assistant Professor
Dept. of CSE

Mr.Sanju D J
Assistant Professor
Dept. of CSE

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

2021-2022

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road,
Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that the DBMS mini-project work entitled **"Vehicle Management System"** has been successfully carried out by **GALI KEERTHI REDDY** bearing USN **1RN19CS052** and **ALLAPARTHI SHARON KUMAR** bearing USN **1RN19CS013**, bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **5th semester Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment 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.

Mrs.S Mamatha Jajur	**Mr.Sanju D J**	**Dr. Kiran P**
Assistant Professor	**Assistant Professor**	**Prof. and Head**
Dept. of CSE	**Dept. of CSE**	**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 are grateful to **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 our sincere gratitude to our guide **Mrs.S Mamatha Jajur,** Assistant Professor,**Mr.Sanju D J**, Assistant Professor, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We 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 :
Place : Bengaluru

Gali Keerthi Reddy 1RN19CS052
Allaparthi Sharon Kumar 1RN19CS013

ABSTRACT

The purpose of “Vehicle Management system” website is to provide an online portal for booking of cars and buses. People can book their vehicles by logging into the website. The website provides two views i.e. user and admin.where user have signup option to register and by logging in they can book their vehicles where admin can manage the bookings and also add the new drivers and vehicles and confirming the bookings and billing the trip.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 DATABASE TECHNOLOGIES

The essential feature of database technology is that it provides an internal representation (model) of the external world of interest. Examples are the representation of a particular date/time/flight/aircraft in an airline reservation or of the item code/item description/quantity on hand/reorder level/reorder quantity in a stock control system.

The technology involved is concerned primarily with maintaining the internal representation consistent with external reality; this involves the results of extensive R&D over the past 30 years in areas such as user requirements analysis, data modelling, process modelling, data integrity, concurrency, transactions, file organisation, indexing, rollback and recovery, persistent programming, object-orientation, logic programming, deductive database systems, active database systems...and in all these (and other) are as there remains much more to be done. The essential point is that database technology is a CORE TECHNOLOGYwhich has links to:

- Information management/processing
- Data analysis/statistics
- Data visualization/presentation
- Multimedia and hypermedia
- Office and document systems
- Business processes, workflow, CSCW(computer-supportedcooperativework)

Relational DBMS is the modern base technology for many business applications. It offers flexibility and easy-to-use tools at the expense of ultimate performance. More recently relational systems have started extending their facilities indirections like information retrieval, object-orientation and deductive/active systems which lead to the so-called 'Extended Relational Systems'.

Information Retrieval Systems began with handling library catalogues and then extended to full free-text by utilizing inverted index technology with a lexicon or thesaurus. Modern systems utilize some KBS(knowledge-basedsystems) techniques to improve the retrieval.

Object-Oriented DBMS started for engineering applications in which objects are complex, have versions and need to be treated as a complete entity. OODBMSs share many of the OOPL features such as identity, inheritance, late binding, overloading and overriding. OODBMSs have found favours in engineering and office systems but haven't been successful yet in traditional application areas.

Deductive / Active DBMS have evolved over the last 20 years and combines logic programming technology with database technology. This allows the database itself to react to the external events and also to maintain its integrity dynamically with respect to the real world.

1.2CHARACTERISTICS OF DATABASE APPROACH

Traditional form included organising the data in file format. DBMS was a new concept then, and all kinds of research were done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics−

- Real-world entity − A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses behaviour and attribute too. For example, a school databasemay use students as an entity and the irage as an attribute.

- Relation-based tables − DBMS allows entities and relations to form tables.
  A user can understand the architecture of a database by just looking at the table names.

- Isolation of data and application − A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.

- Less redundancy − DBMS follows the rules of normalization, which splits a relation when any of its attributes has redundancy in its values. Normalization is a mathematicallyrich and scientific process that will reduce the data redundancy.

- Consistency − Consistency is a state where every relation in a database remains consistent. There exists methods and techniques, that can detect an attempt of leaving database in an in consistent state. DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.

- Query Language − DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and the filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.

- ACID Properties − DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database to stay healthy in multi-transactional environments and also incase of failure.

- Multiuser and Concurrent Access − DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.

- Multiple views − DBMS offers multiple views for different users. A user in the Sales department will have a different view of the database from the person working in the Production department. This feature enables the users to have a concentrate view of the database according to the requirements.

- Security − Features like multiple views offer security to certain extent when users are unable to access the data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. It can also be helpful in deciding how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

1.3 APPLICATIONS OF DBMS

Applications of Database Management Systems:

- **Telecom**: There is a database to keeps track of the information regarding the calls made, network usage, customer details etc. Without the database system it is hard to maintain such huge amount s of data which gets updated every millisecond.

- **Industry**:Whether it is a manufacturing unit, a warehouse or a distribution centre, each one needs a database to keep the records of the ins and outs. For example, a distribution centre should keep a track of the product units that were supplied to the centre as well as the products that got delivered from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System**: For storing information regarding a customer, keeping a track of his/her day to day credit and debit transactions, generating bank statements etc is done with through Database management systems.

- **Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding the student, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is lots of inter-related datathatneeds to bestored and retrievedinanefficient manner.

- **Online shopping**: You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

1.4 PROBLEM DESCRIPTION/STATEMENT

The purpose of "Vehicle Management system" website is to provide an online portal for booking of cars and buses. People can book their vehicles by logging into the website. The website provides two views i.e. user and admin.

The user can browse through the website and book the vehicles of his/her choice. The admin can manage the bookings. He can add, delete and update the bookings and their details. Along with login option there is also a sign-up option which can be used by a new user to create a new account.

CHAPTER 2
REQUIREMENT ANALYSIS

2.1 HARDWARE REQUIREMENTS

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

Processor	:	Pentium 4 processor
Processor Speed	:	2.4GHz
RAM	:	1GB
Storage space	:	40GB
Monitor Resolution	:	1024*768 or1336*768 or 1280*1024

2.2 SOFTWARE REQUIREMENTS

1. Operating System used : Windows10
2. Language : HTML, CSS,PHP,JavaScript
3. WAMPPServer : MySQL, PhpMyAdmin
4. IDEused : VS COD

2.3 FUNCTIONAL REQUIREMENTS

2.3.1 Major Entities

User: Users is the entity that signs up as a user and make booking vehicle for a perod of time using his/her credentials. Few attributes are user_id,password,username,email etc.

admin: Admin manages the booking done by the user and confirms the vehicle booking and for generating bills. Few attributes are password,username,admin_id etc.

booking:bookings will have details of booking id and its details. Few attributes are book_id,req_date,req_time,veh_reg etc

vehicles: It has vehicle details like vehicle id,registration and type.

Driver: Users can also book the drivers for their booked vehicles. Few attributes are driver_id,D_name,drlicense.

Tripcost:Total cost taken for the trip and its details. Few attributes are username,Book_id etc.

Bill: bill will have the details of total bill that was effected.

Few attributes are bill_id,usrename,oil,salary,equipment.

2.3.2 End User Requirements

The technical requirements for the project are mentioned below.

2.3.2.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from a local storage and render them to multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

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. HTML elements are delimited by tags that are written within angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Other tags such as ... surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the

content of the page. HTML can also embed programs written in a scripting language such as JavaScript which affect the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content.

2.3.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language which is used for describing the presentation of a document written in a markup language. Although most often its used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is also applicable to rendering inspeech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of presentation and
content, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share the formatting by specifying the relevantCSS in a separate .css file, and reduce complexity and repetition in the structural content.

2.3.2.3 JAVASCRIPT

JavaScript often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. 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. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamictyping,prototype-based object-orientation, and first-class functions. It is multi-paradigm, supportingevent-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).

2.3.2.4 PHP

PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language. Originally created by RasmusLerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally

stood for Personal Home Page, but it now stands for the recursive acronym PHP:Hypertext Pre-processor.

PHP code can be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code can also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

The standard PHP interpreter, powered by the Zend Engine, is a free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers, on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone into creating a formal PHP specification. HP development began in 1995 when RasmusLerdorf wrote several Common Gateway Interface (CGI) programs in C, which he used in order to maintain his personal homepage. He extended them to work with web forms and to communicate with databases, and called this implementation "Personal Home Page/Forms Interpreter" or PHP/FI.

PHP/FI could help to build simple, dynamic web applications. To accelerate bug reporting and to improve the code, Lerdorf initially announced the release of PHP/FI as "Personal Home Page Tools (PHP Tools) version 1.0" on the Usenet discussion group on June 8, 1995 This release already had the basic functionality that PHP has as of 2013. This included Perl-like variables, form handling, and the ability to embed HTML. The syntax resembled that of Perl but was simpler,more limited and less consistent.

2.3.2.5 MySQL

MySQL is a Relational Database Management System (RDBMS). MySQL server can manage many databases at the same time. In fact, many people might have different databases managed by a single MySQL server. Each database consists of a structure to hold onto the data itself. A data-base can exist without data, only a structure, be totally empty, twiddling its thumbs and waiting for data to be stored in it.

Data in a database is stored in one or more tables. You must create the data-base and the tables before you can add any data to the database. First you create the empty database. Then you add empty tables to the database. Database tables are organized in rows and columns. Each row represents an entity in the database, such as a customer, a book, or a project. Each column contains an item of information about the entity, such as a customer name, a book name, or a project start date. The place where a particular row and column intersect, the individual cell of the table, is called a field. Tables in databases can be related. Often a row in one table is related to several rows in another table. For instance, you might have a database containing data about books you own. You would have a book table and an author table. One row in the author table might contain information about the author of several books in the book table. When tables are related, you include a column in one table to hold data that matches data in the column of another table. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation Open Source company that unites Open Source values and methodology with a successful business model.

MySQL is a database management system. A database is a structured collection of data. It can be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and“SQL:2003” refers to the current version of the standard. We use the phrase “the SQLstandard” to refer to the current version of the SQL Standard.

MySQL software is Open Source. Open Source means that it is possible for anyone to use

and modify the software. Anybody can download the MySQL software from the Internet and useit without paying anything. If you wish, you may study the source code and change it to suit yourneeds. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations.

MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases and has been successfully used in highly demanding production environments for several years. MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

MySQL Server works in a client/server or embedded systems. The MySQL Database Software is a client/server system which consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools,and a wide range of application programming interfaces(APIs).

2.3.2.6 XAMPP Server

Xampp server installs a complete, ready-to-use development environment. Xampp server allows you to fit your needs and allows you to setup a local server with the same characteristics as your production.

While setting up the server and PHP on your own, you have two choices for the method of connecting PHP to the server. For many servers, PHP has a direct module interface (also called SAPI). These servers include Apache, Microsoft Internet Information Server, Netscape and iPlanet servers. Many other servers support ISAPI, the Microsoft module interface (Omni HTTP d for example). If PHP has no module support for your web server, you can always use it as a CGI or FastCGI processor. This means you set up your server to use the CGI executable of PHP toprocess all PHP file requests on the server.

CHAPTER 3

DATABASE DESIGN

3.1 Entities, attributes and relationships

From the above list of entities, we can conclude that admin,booking,tripcost,user,bill,driver,vehicle are the 7 core entities in our vehicle management system.Their attributes are as follows:

admin:

Admin_id,

password,

username

Booking:

Book_id,

username,

Driver_id,

type,

Req_date,

Veh_reg,

Req_time

Tripcost:

id,

Book_id,

Oil_cost,

totalcost,

Username

User:

username,

User_id,

admin,

password,

email,

First_name,

Last_name

Bill:

Bill_id,

id,

username,

oil,

salary,

equipment,

tcost

Driver:

Driver_id,

D_name,

draddress,

drlicensevalid,

drnid,

drlicense

Vehicle:

Veh_id,

Veh_photo,

Veh_type,

Veh_reg,

Veh_available

3.2 ER and schema

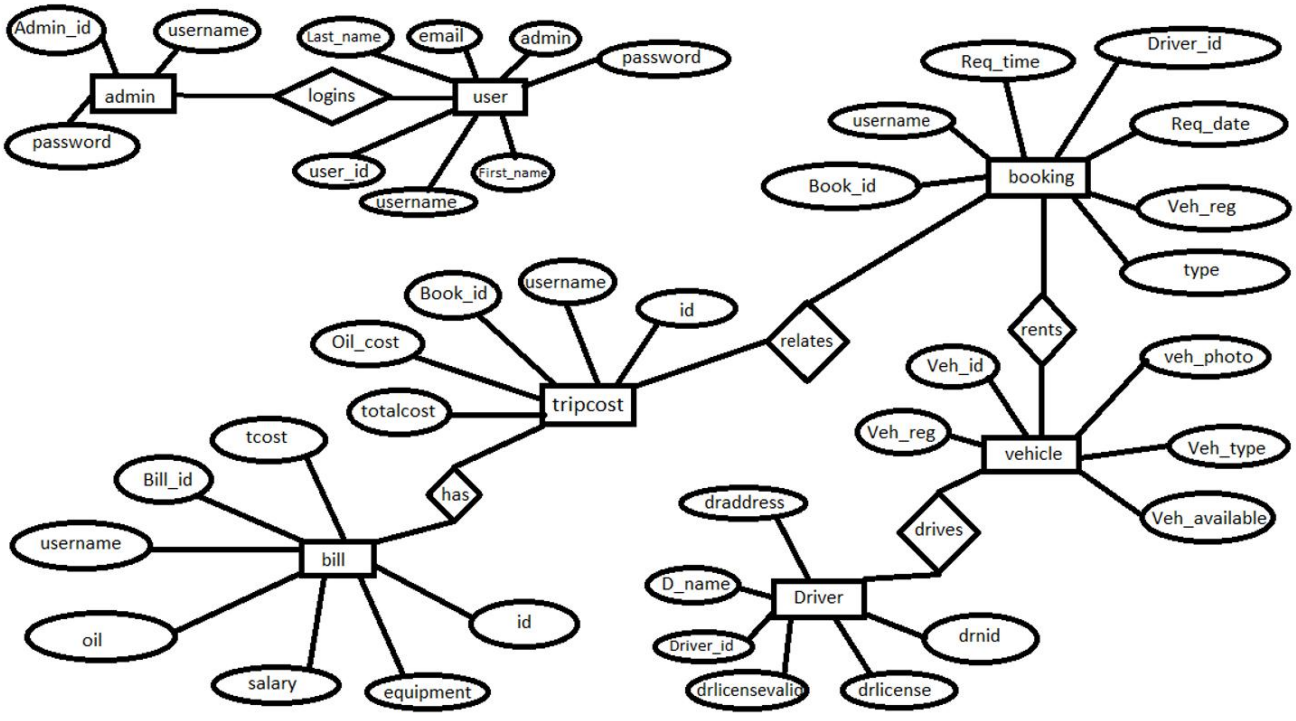

Fig.3.1 ER Diagram for Vehicle Management System

3.3 Relational Schema

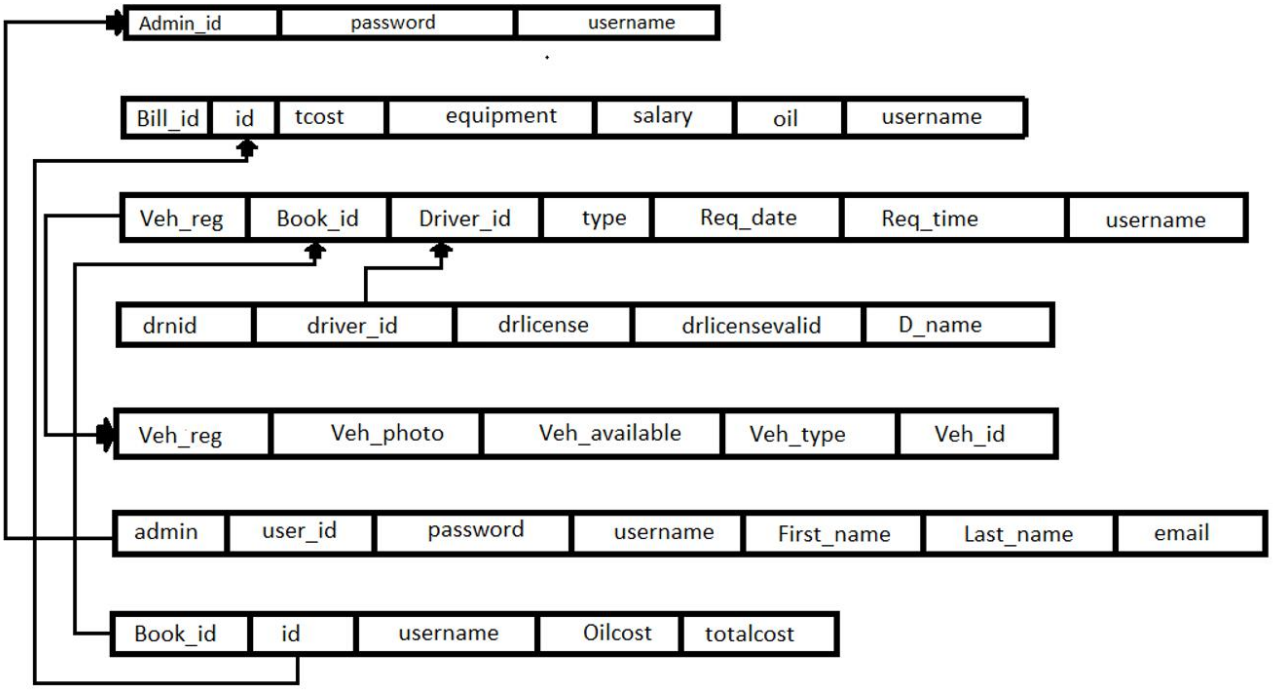

Fig.3.2. Schema for Vehicle Management System

CHAPTER 4:

IMPLEMENTATION

4.1 Creating Database Connection

1. PHP provides built-in database connectivity for a wide range of databases – MySQL, PostgreSQL, Oracle, Berkeley DB, Informix, Lotus Notes, and more.

2. public Connection

```php
<?php
$connection=mysqli_connect("localhost","root","","vehicle_management");

session_start();
?>
```

Architecture used (4-TIER architecture)

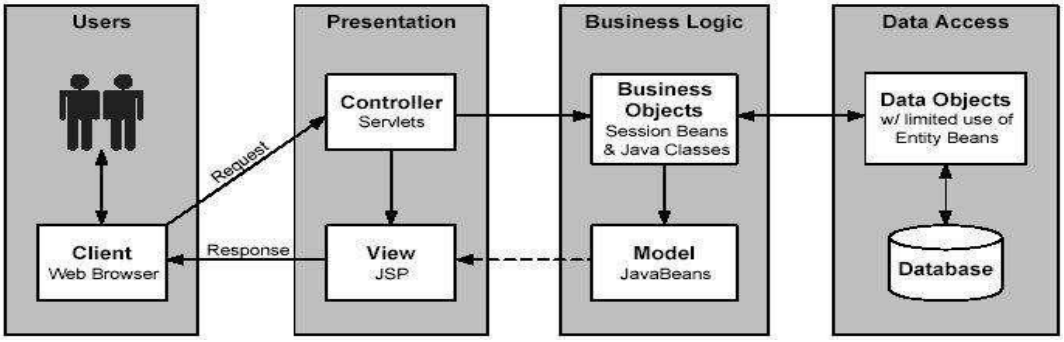

Figure 4.1 The 4-TIER architecture

Four Tier architecture is a client–server architecture in which presentation, application processing, and data management functions are physically separated. Four-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application.

Presentation layer

This is the topmost level of the application. The presentation tier displays information related to services such as browsing merchandise, purchasing and shopping cart contents. It also communicates with other tiers and puts out the results to the browser/client tier and to all other tiers in the network. In simple terms, it is a layer which users can access directly (such as a web page, or an operating system'sGUI).

Business layer

Business layer or domain logic is the part of the program that encodes the real-world business rules which determine how data can be created, stored, and changed. It is contrasted with the remainder of the software that might be concerned with lower-level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program.

Data access layer

A Data Access Layer (DAL) in computer software is a layer of computer program which provides simplified access to data stored in persistent storage.

For example, the DAL might return a reference to an object (in terms of object-oriented programming) with its attributes instead of a row of fields from a database table. This allows the client (or user) modules to be created with a higher level of abstraction. This kind of model could be implemented by creating a class of data access methods that directly reference a corresponding set of database stored procedures. Another implementation could potentially retrieve or write records to or from a file system. The DAL hides the complexity of the underlying data store from the external world.

Control layer The control layer is responsible for the communication between business and presentation layer. It connects logic and data with each other and provides a better connectivity and separationbetween layers.

4.2 Pseudo Code for Major Functionalities

Login page: It is used for login purposes. When we enter the correct email and password itwill go to the next page.We can use signup to create an account.

```
    <?php
    session_start();
    $connection=mysqli_connect("localhost","root","","vehicle_management");

    $msg="";
    if(isset($_POST['submit'])){
        $username=mysqli_real_escape_string($connection,strtolower($_POST['username']));

        $password=mysqli_real_escape_string($connection,$_POST['password']);

        $login_query="SELECT * FROM `user` WHERE username='$username' and password='$password'";

        $login_res=mysqli_query($connection,$login_query);
        if(mysqli_num_rows($login_res)>0){
            $_SESSION['username']=$username;
            header('Location:index.php');
        }
        else{
             $msg= '<div class="alert alert-danger alert-dismissable" style="margin-top:30px";>
                    <a href="#" class="close" data-dismiss="alert" aria-label="close">×</a>
                    <strong>Unsuccessful!</strong> Login Unsuccessful.
                  </div>';
        }
    }

?>
```

Login admin page :

```
<?php
    session_start();
    $connection=mysqli_connect("localhost","root","","vehicle_management");

    $msg="";
    if(isset($_POST['submit'])){
        $username=mysqli_real_escape_string($connection,strtolower($_POST['username']));

        $password=mysqli_real_escape_string($connection,$_POST['password']);

        $login_query="SELECT * FROM `admin` WHERE username='$username' and password='$password'";

        $login_res=mysqli_query($connection,$login_query);
        if(mysqli_num_rows($login_res)>0){
            $_SESSION['username']=$username;
            header('Location:admin.php');
        }
        else{
             $msg= '<div class="alert alert-danger alert-dismissable" style="margin-top:30px";>
                    <a href="#" class="close" data-dismiss="alert" aria-label="close">×</a>
                    <strong>Unsuccessful!</strong> Login Unsuccessful.
                  </div>';
        }
    }
?>
```

Sign up page:

```php
  <?php
$connection=mysqli_connect("localhost","root","","vehicle_management");

session_start();
$msg="";

if(isset($_POST['submit'])){
    $firstname= mysqli_real_escape_string($connection,strtolower($_POST['firstname']));
    $lastname= mysqli_real_escape_string($connection,strtolower($_POST['lastname']));
    $email= mysqli_real_escape_string($connection,strtolower($_POST['email']));
    $username= mysqli_real_escape_string($connection,strtolower($_POST['username']));
    $password= mysqli_real_escape_string($connection,strtolower($_POST['password']));


    $signup_query= "INSERT INTO `user`(`user_id`, `first_name`, `last_name`, `email`, `username`, `password`)
VALUES ('','$firstname','$lastname','$email','$username','$password')";

    $check_query= "SELECT * FROM `user` WHERE username='$username' or email='$email'";

    $check_res=mysqli_query($connection,$check_query);

    if(mysqli_num_rows($check_res)>0){
         $msg= '<div class="alert alert-warning" style="margin-top:30px";>
                  <strong>Failed!</strong> Username or Email already exists.
                  </div>';

    }

    else{
        $signup_res= mysqli_query($connection,$signup_query);
             $msg= '<div class="alert alert-success" style="margin-top:30px";>
                  <strong>Success!</strong> Registration Succefull.
                  </div>';

    }

}
?>
```

Home page:

```
       <?php
    $connection=mysqli_connect("localhost","root","","vehicle_management");

    session_start();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>vehicle management system</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
    <script src="https://code.jquery.com/jquery-2.2.0.min.js" type="text/javascript"></script>
     <script src="https://unpkg.com/scrollreveal/dist/scrollreveal.min.js"></script>
    <link rel="stylesheet" type="text/css" href="./slick/slick.css">
    <link rel="stylesheet" type="text/css" href="./slick/slick-theme.css">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" href="animate.css">
    <link rel="stylesheet" href="style.css">

</head>
```

Insert: This is for insertion into the tables

```
      <?php
     $connection= mysqli_connect('localhost','root','','vehicle_management');
    session_start();
    $id= $_GET['id'];
     $msg="";
      if(isset($_POST['submit'])){
        $username= $_POST['username'];
        $total_km=$_POST['total_km'];
        $oil_cost=$_POST['oil_cost'];
        $extra_cost=$_POST['extra_cost'];
        $total_cost=$_POST['total_cost'];
    }
    $sql="INSERT INTO `tripcost`(`booking_id`,`username`, `total_km`, `oil_cost`, `extra_cost`, `total_cost`)
VALUES ('$id','$username','$total_km','$oil_cost','$extra_cost','$total_cost')";
     $result= mysqli_query($connection,$sql);
   if($result==true){
        $msg= "<script language='javascript'>
                                    swal(
                                         'Success!',
                                         'Registration Completed!',
                                         'success'
                                         );
                        </script>";

        }
        else{
            die('unsuccessful' .mysqli_error($connection));
        }


?>
```

Update: This is for updating the table entries

```php
<?php

$connection=mysqli_connect("localhost","root","","vehicle_management");
session_start();

$msg="";
$id=$_GET['id'];

$query= "SELECT * FROM `tripcost` WHERE booking_id='$id'";
$query1="UPDATE `booking` SET `paid`='1' WHERE booking_id='$id'";
//echo $query;
$result= mysqli_query($connection,$query);
$result1= mysqli_query($connection,$query1);
$row= mysqli_fetch_assoc($result);
?>
```

Delete: This is for deleting the table entries

```php
<?php

$id= $_GET['id'];

$conn=mysqli_connect('localhost','root','','vehicle_management');
$sql="DELETE FROM `booking` WHERE booking_id='$id'";
 echo $sql;
$result=mysqli_query($conn,$sql);
if(mysqli_query($conn,$sql)){
    header("Location: bookinglist.php");
}else{
      echo "Not deleted";
}
```

Trigger: Trigger is added after the booking where it calculate the cost of the trip.

Figure 4.7 Trigger

CHAPTER 5:

RESULTS , SNAPSHOTS AND DISCUSSIONS

Login page: This page allows the users to login into the application as shown in fig. 5.1.

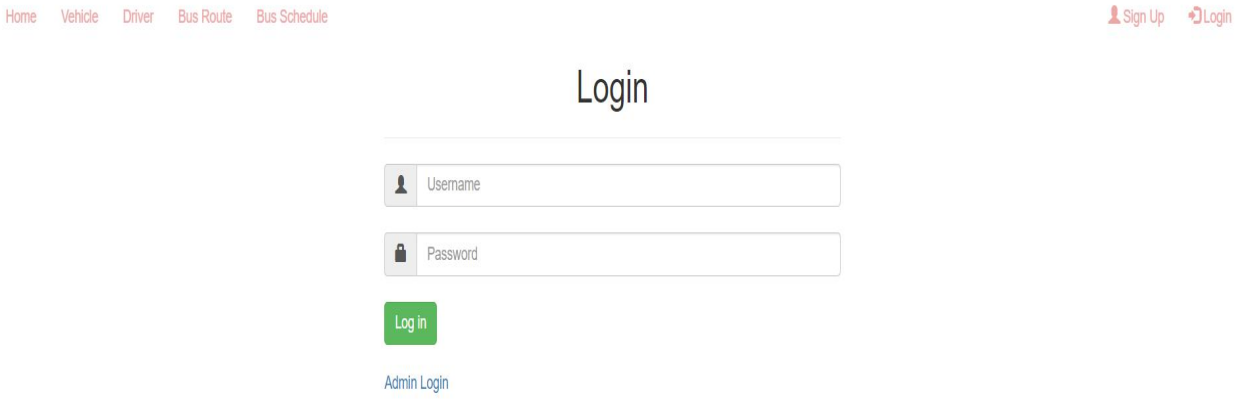

Fig. 5.1 Login page

Signup page: This page allows users to create a new account if the account does not exist as shown in fig. 5.2.

Fig. 5.2 Signup page

Homepage: Allows the users to enter their basic details and also to view their entries as shown in fig. 5.3.

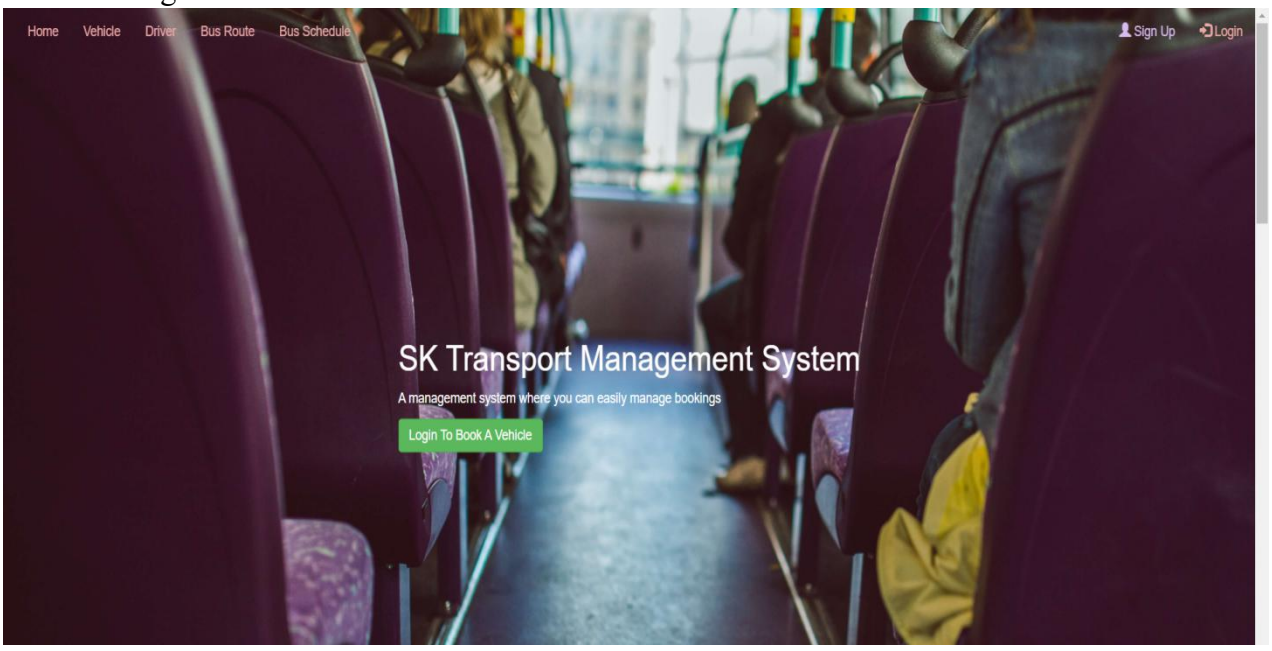

Fig. 5.3 Homepage

Admin Login: Allows the admin to enter their username and password as shown in fig. 5.4.

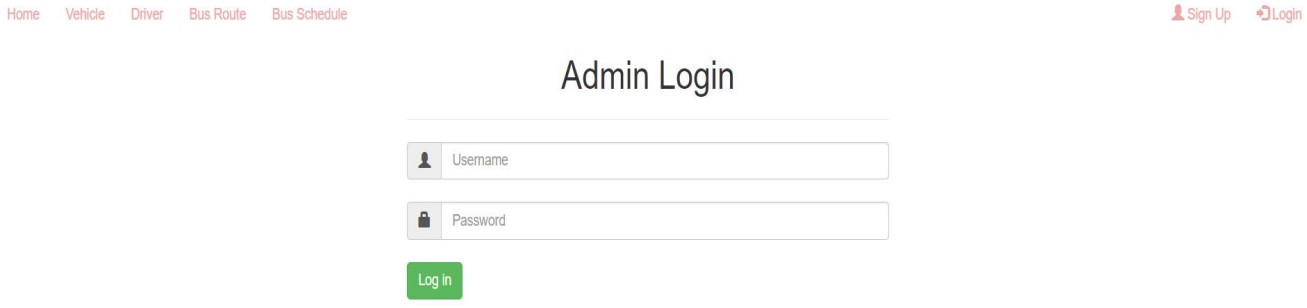

Fig. 5.4 Admin Login

Vehicle List: User can view the bus details and admin can insert the new vehicles fig. 5.5.

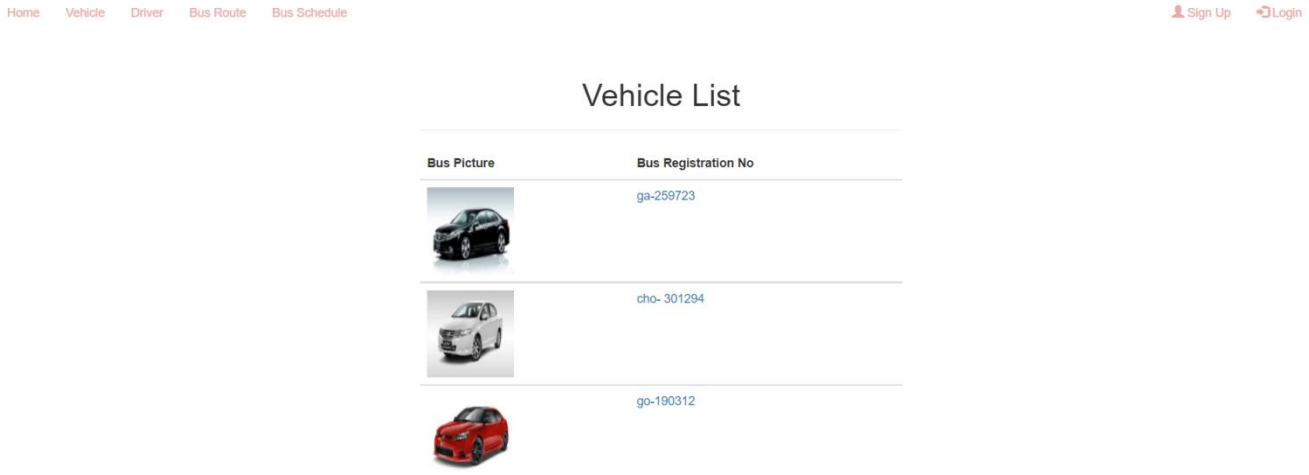

Fig. 5.5 Vehicle List

Admin Panel: admin can use these portal to change the data or add to it as shown in fig. 5.6.

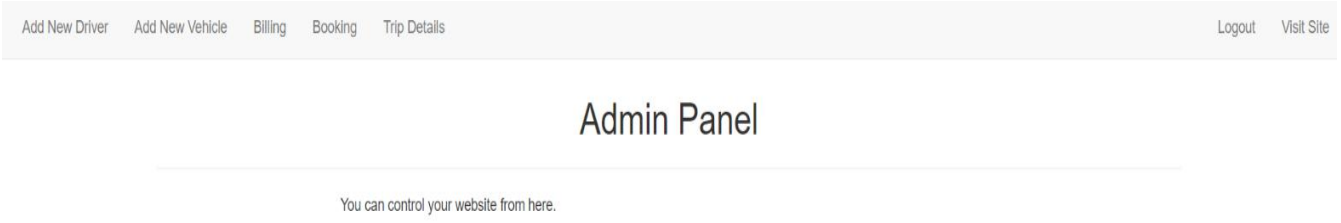

Fig. 5.6 Admin Panel

Billing Information: Admin can view these information for conformation.

Fig. 5.7 Billing Information

Billing List: Admin can view these information for conformation.

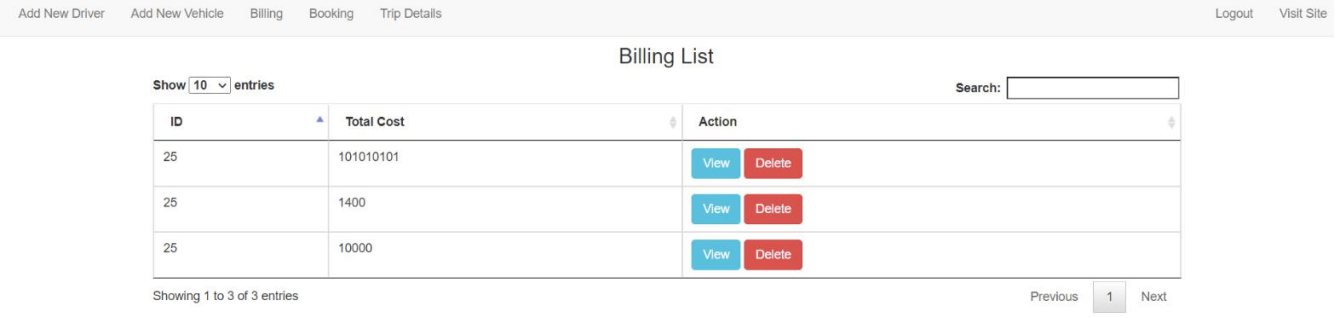

Fig. 5.8 Billing List

Booking List: Admin can delete or confirm the trip by releasing the vehicle and checks the bill payments.

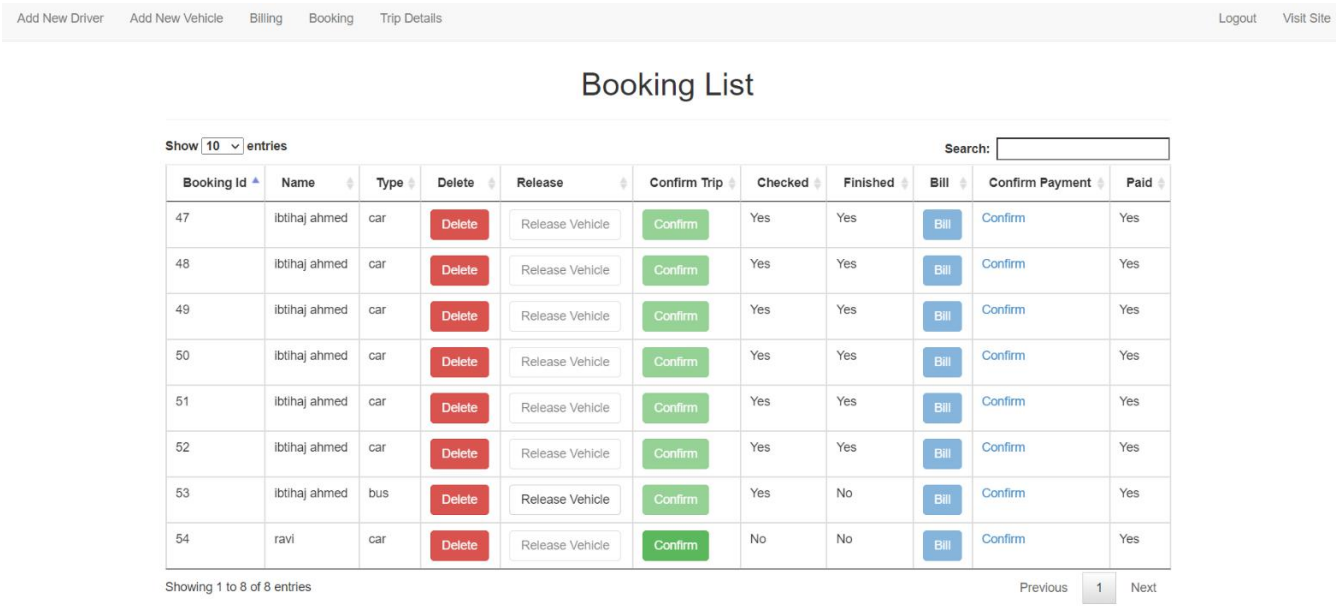

Fig. 5.9 Booking List

Booking form:

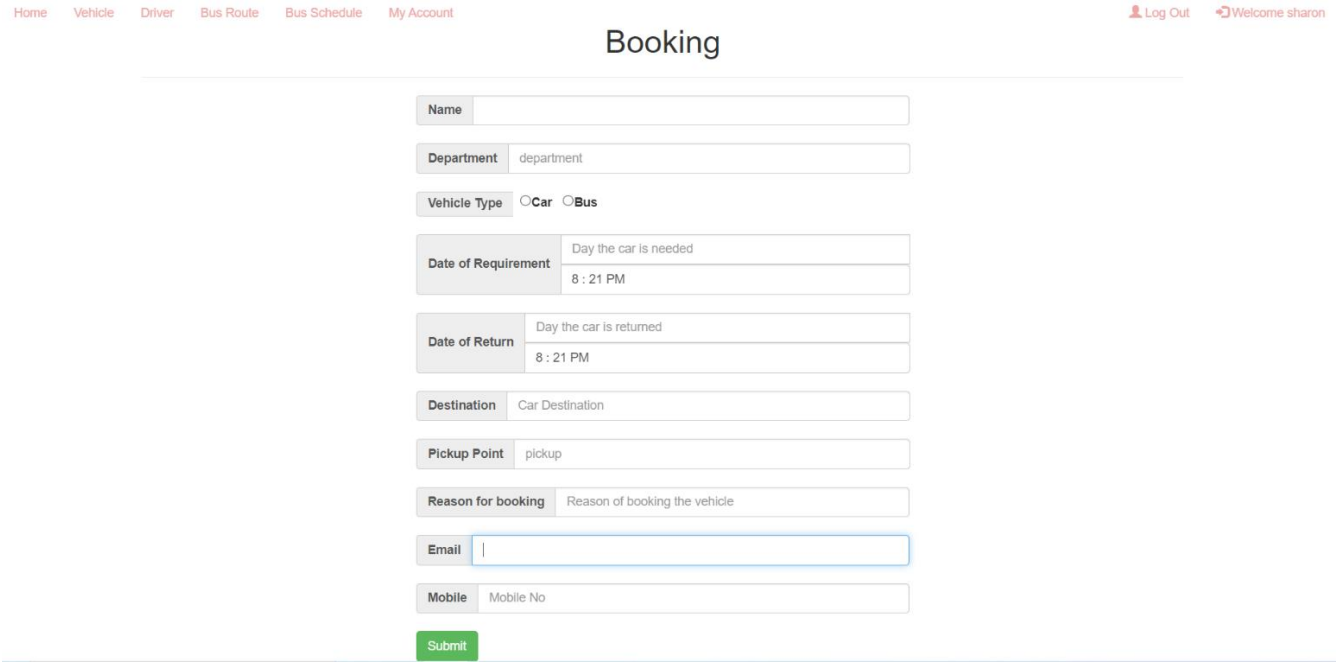

Fig. 5.10 Booking form

Bus Schedule:

Home Vehicle Driver Bus Route Bus Schedule Sign Up Login

Permanent Bus Schedule

NO	PERIOD	FIRST	SECOND	THIRD
01	1st JANUARY to 31st JANUARY	3-45	4-45	6-15
02	1st FEBRUARY to 28/29th FEBRUARY	4-00	5-00	6-30
03	1st MARCH to 31st MARCH	4-15	5-15	6-45
04	1st APRIL to 30th APRIL	4-30	5-30	7-00
05	1st MAY to 31st MAY	4-45	5-45	7-15
06	1st JUNE to 31st JULY	5-00	6-00	7-30
07	1st AUGUST to 15th AUGUST	5-00	6-00	7-15
08	16th AUGUST to 31st AUGUST	4-45	5-45	7-00
09	1st SEPTEMBER to 15th SEPTEMBER	4-30	5-30	6-45
10	16th SEPTEMBER to 30th SEPTEMBER	4-15	5-15	6-35
11	1st OCTOBOR to 15th OCTOBOR	4-00	5-00	6-15
12	16th OCTOBOR to 31st OCTOBOR	3-45	4-45	6-00
13	1st NOVEMBER to 31st DECEMBER	3-30	4-30	5-45

Fig. 5.11 Bus Schedule

Driver:

Home Vehicle Driver Bus Route Bus Schedule Sign Up Login

Drive List

Profile Picture	Driver Name
	fahim
	Arman

Fig. 5.12 Booking List

Bus Route:

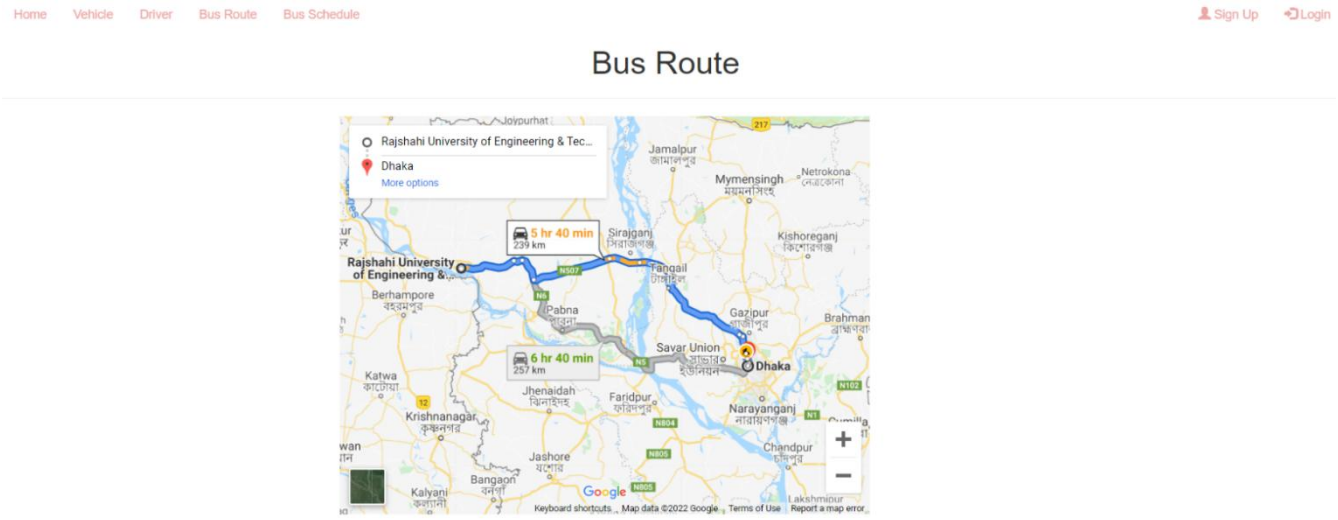

Fig. 5.13 Bus Route

Add New Driver:

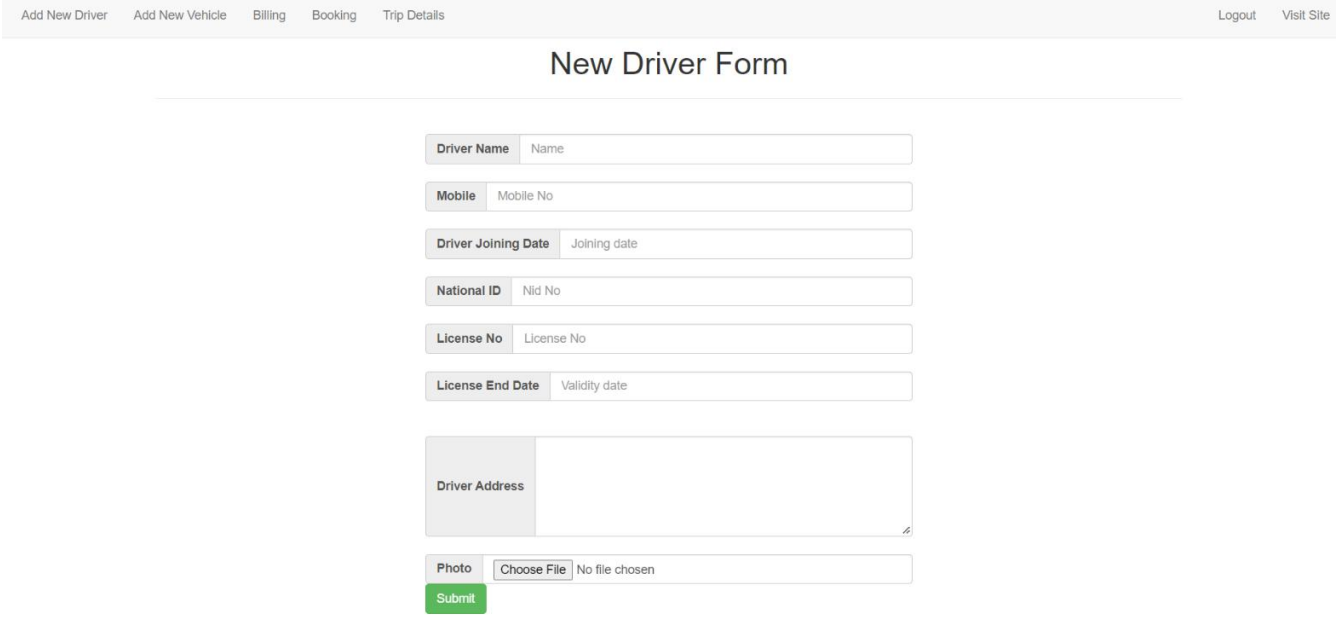

Fig. 5.14 Add New Driver

Trip Details:

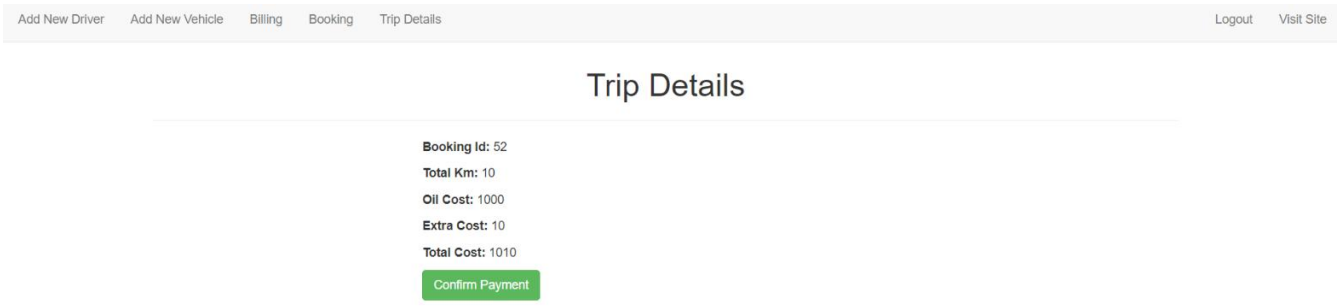

Fig. 5.15 Trip Details

Add New Vehicle:

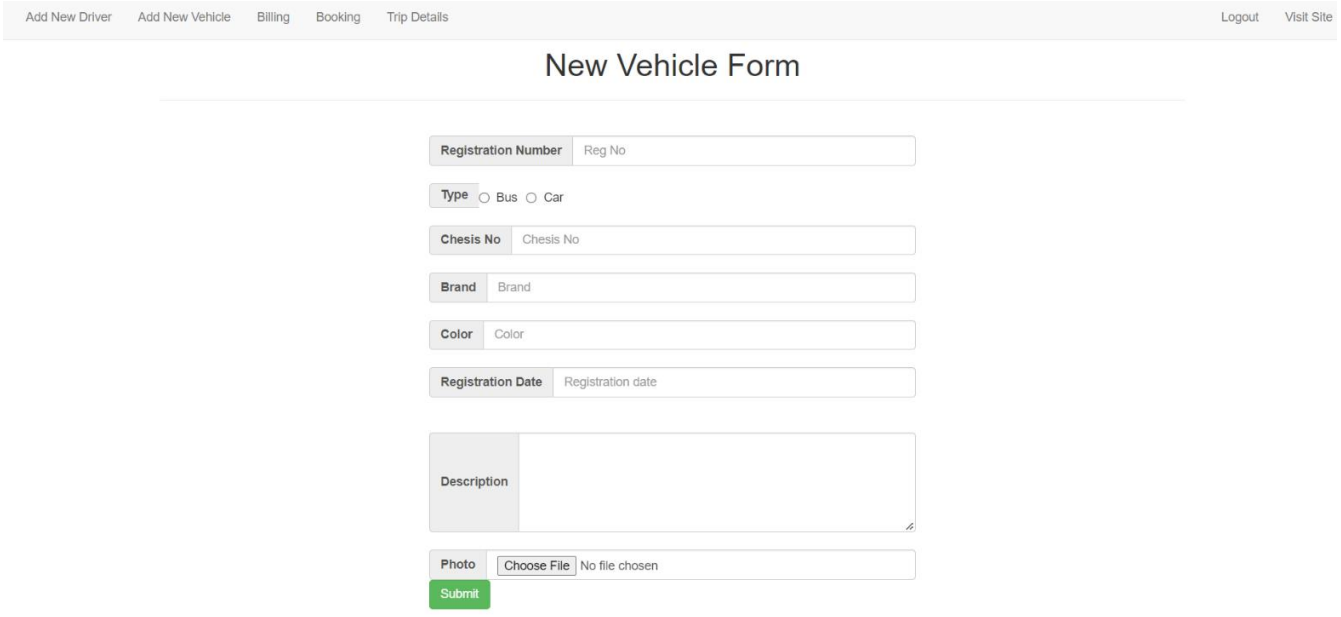

Fig. 5.16 Add New Vehicle

Trip Details :

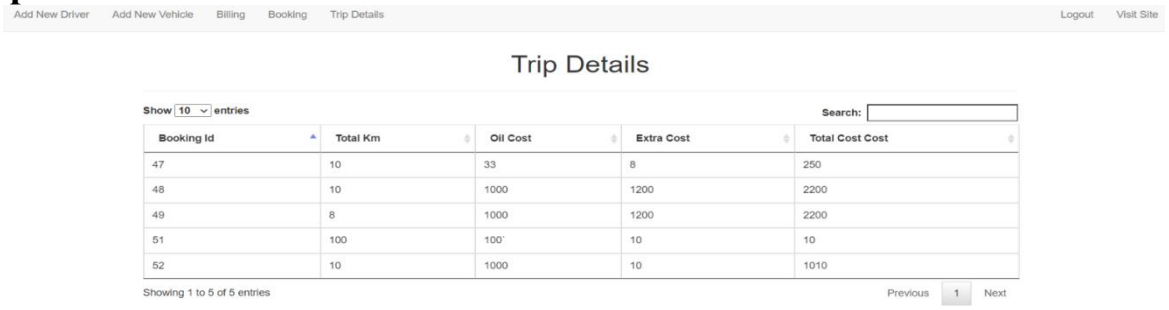

Fig. 5.16 Trip Details

Table Entries:

Users table: Fig 5.8 shows entries of Users table.

	user_id	first_name	last_name	email	username	password	admin
Edit Copy Delete	10	ibtihaj	ahmed	fahad@gmail.com	fahad	asdasd	0
Edit Copy Delete	13	dad	ahmed	fahaha@gmail.com	asdasd	asdasdad	0
Edit Copy Delete	14	rakibul	hoque	rakib@gmail.com	rakib24	1234	0

Fig. 5.17

Tripcost:

	id	booking_id	username	total_km	oil_cost	extra_cost	total_cost	paid
Edit Copy Delete	13	47	fahad	10	33	8	250	1
Edit Copy Delete	15	48	fahad	10	1000	1200	2200	1
Edit Copy Delete	18	49	fahad	8	1000	1200	2200	1
Edit Copy Delete	19	51	fahad	100	100`	10	10	1
Edit Copy Delete	20	52	fahad	10	1000	10	1010	1

Fig. 5.18

Driver:

	driverid	drname	drjoin	drmobile	drnid	drlicense	drlicensevalid	draddress	drphoto	dr_available
Edit Copy Delete	20	fahim	03/04/2018	01717172398712	91238912093097812	1093q098091839	03/26/2018	creasent road, dhaka, bangladesh.	nsf.jpg	1
Edit Copy Delete	21	Arman	03/04/2018	0123084982	kls3893809839082	1093q098091839	03/04/2018	Oktroy Mor, Rajshahi.	rkb.jpg	0

Fig. 5.19

Booking:

	booking_id	name	username	department	type	req_date	req_time	ret_date	ret_time	destination	pickup_point	resons	email
Edit Copy Delete	47	ibtihaj ahmed	fahad	CSE	car	05/01/2018	7 : 19 AM	05/02/2018	7 : 19 AM	Katakhali	Ruet Gate	Education	fahad@gmail.com
Edit Copy Delete	48	ibtihaj ahmed	fahad	CSE	car	05/01/2018	7 : 22 AM	05/02/2018	7 : 22 AM	dd	dd	dd	fahad@gmail.com
Edit Copy Delete	49	ibtihaj ahmed	fahad	CSE	car	05/24/2018	7 : 32 AM	05/25/2018	7 : 32 AM	ra	as	Trip	fahad@gmail.com
Edit Copy Delete	50	ibtihaj ahmed	fahad	cse	car	07/09/2018	8 : 55 PM	07/10/2018	8 : 55 PM	kajla	kajla	emni	fahad@gmail.com
Edit Copy Delete	51	ibtihaj ahmed	fahad	dd	car	07/09/2018	9 : 05 PM	07/09/2018	9 : 05 PM	dd	dd	dd	fahad@gmail.com
Edit Copy Delete	52	ibtihaj ahmed	fahad	dd	car	07/02/2018	9 : 06 PM	07/02/2018	9 : 06 PM	dd	dd	sd	fahad@gmail.com
Edit Copy Delete	54	ravi	rakib24	cse	car	02/25/2022	7 : 32 PM	02/26/2022	7 : 32 PM	delhi	pune	tour	rakib@gmail.com

Fig. 5.20

Bill :

	bill_id	id	username	salary	equipment	oil	tcost
Edit Copy Delete	8	25		1000	1000001	1010	101010101
Edit Copy Delete	9	25		1000	100	300	1400
Edit Copy Delete	10	25		10000	500	4000	10000

Fig. 5.21

Admin:

	admin_id	username	password
Edit Copy Delete	1	mihaf24	1234
Edit Copy Delete	2	sharon	6543

Fig. 5.22

Vehicle:

	veh_id	veh_reg	veh_type	chesisno	brand	veh_color	veh_regdate	veh_description	veh_photo	veh_available
Edit Copy Delete	25	ga-259723	car	101-12303.20201	corolla	Black	25/03.17	hello this is a black car.	1.jpg	1
Edit Copy Delete	26	cho- 301294	car	101-12309.23981	axio	white	26.9.15	hello this is a white car.	2.jpg	0
Edit Copy Delete	35	go-190312	car	101321-138713.3291823	Corolla	Red	10/17/2017	This is a red car.	images.jpg	1

Fig. 5.23

CHAPTER 6:

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

Today the world is considered as a competitive world where everybody seeks for accuracy in least Time.Earlier paper work was the means to keep various records.It was very time consuming and not even that accurate.So we decided to design and develop the project called vehicle management system which eliminates the paper work and provides better option to the people for their vehicle records.It deals with the maintenance of the records of the different categories of vehicles and their bookings.The user of this program can book their vehicles and they can access the platform with ease.

Our back-end tech is reliable and efficient enough to not create any hurdles from the time the user logs into the website to the time when he/she has found the information needed. We hope that the users have a seamless experience.

6.2 Future Enhancements

The project database could be improved by adding some more attributes and tables.

Tables can further be improved by adding another trigger to calculate the bill

automatically.

Further changes can be easily done by changing the code.

The front end of the website can be made more attractive by using more advanced concept of css.

BIBLIOGRAPHY

1. RamezElmarsi and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.

2. https://stackoverflow.com/

3. https://www.w3schools.com/

4. www.php.net

5.https://trainings.internshala.com/web-development