Chapter 1

INTRODUCTION

AIM & OBJECTIVE:

- To produce a web-based system that allow customer to register and Booking car online and for the company to effectively manage their car rental business.
- To ease customer's task whenever they need to rent a car.

ABOUT PROJECT:

Welcome to our Car Rental System! Our platform is designed to provide you with a convenient and hassle-free way to rent a car for your travel needs. Whether you're going on a business trip, planning a family vacation, or simply need a temporary set of wheels, our Car Rental System offers a wide range of vehicles to suit your preferences and budget.

With our user-friendly interface and streamlined booking process, renting a car has never been easier. Our system allows you to browse through a diverse fleet of cars, including sedans, SUVs, luxury vehicles, and more. You can view detailed information about each car, such as its specifications, features, and availability, enabling you to make an informed decision.

Here, User has to Login To book a car. The user can search for cars easily and book. For bookings, the user has to provide information such as Booking Dates and Text Message. All car details are provided and it also includes Car's feature and Overview. The user can also post their Testimonials and the user can update their Profile as well as passwords anytime they want from the site. Admin can Add/Manage car brands, manage cars, bookings, testimonial, pages and many more. It's easy to operate and understand by users. This site makes customers easy for car rental. The design is pretty simple and the user won't find it difficult to understand, use and navigate.

Reason for the Project :

- Enhance Business Processes: To be able to use internet technology to project the rental company to the global world instead of limiting their services to their local domain alone, thus increase their return on investment (ROI).
- Online Car Reservation: A tools through which customers can Booking available cars online prior to their expected pick-up date or time.
- Customer's registration: A registration portal to hold customer's details, monitor their transaction and used same to offer better and improve services to them.
- Group Booking Event Management: Allows the customer to book space for a group in the case of weddings or corporate meetings.

Chapter 2

LITERATURE SURVEY

2.1 Existing System

Car Rental system Service is an innovative thought to simplify the Transportation problems of Employees of an organization. In the present System, Organization do maintain a person for the allocating and proper functioning of transportation .The Person appointed needs to look after the assigning and movement of cabs. Authorized person maintains the transportation details in papers, which is a tedious task if any updations or changes need to be done.

- Details are stored in Papers.
- Maintenance is a huge problem.
- Updation, changes in details is a tedious task.
- Performance is not achieved up to the requirements.

2.2 Proposed System

In the Previous System, Details are Stored Manually in papers, to share the details between employees was a financial drawback. Updations in the details is a tedious task. But a new system was proposed to overcome the above drawbacks.

Functionalities and advantages of proposed system are:

- Data is Centralized which has overcome the Sharing problem in previous system.
- As data is Maintained electronically, it's easy for a person to update the details, which has overcome the tedious updation in previous system.
- Maintenance is easy and performance is good.

2.3 Technologies Used

Here we will be discussing about the requirement of making this application possible and response as we wanted it to this is only done through the thinking of the developer. In this we will be also understanding the platform on which our application is running and on which it is being developed.

2.3.1 About PHP

PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor.

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. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

The standard PHP interpreter, powered by the Zend Engine, is 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, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification.

PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use. PHP is an amazing and popular language!

It is powerful enough to be at the core of the biggest blogging system on the web WordPress!. It is deep enough to run the largest social network Facebook!. It is also easy enough to be a beginner's first server side language.

PHP files can contain text, HTML, CSS, JavaScript, and PHP code. PHP code is executed on the server, and the result is returned to the browser as plain HTML. PHP files have extension ".php".

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

USES:

- PHP runs on various platforms Windows, Linux, Unix, Mac OS X, etc.
- PHP is compatible with almost all servers used today Apache, IIS, etc.
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

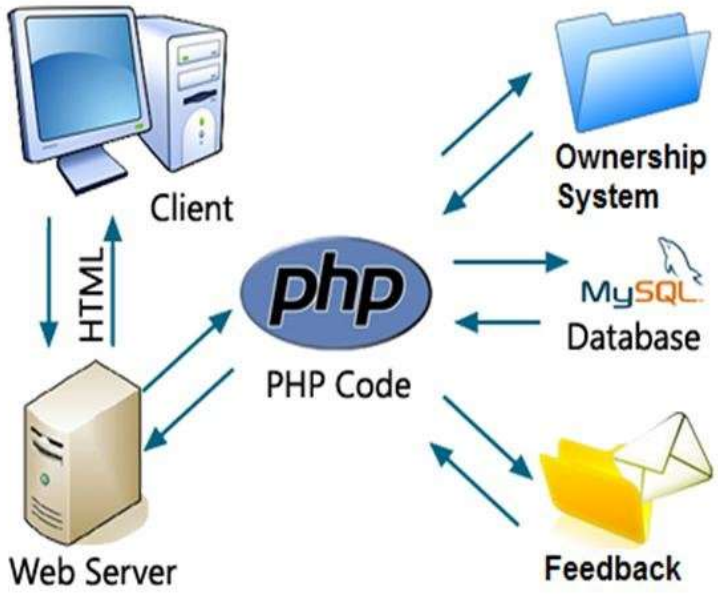

Fig 2.3.1 PHP- Hypertext Preprocessor

PHP is a server-side scripting language designed to be used for web purposes. Originally PHP was short for "Personal Home Page" but over time it evolved to include that in its recursive current expansion "PHP: Hypertext Preprocessor". Server-side scripting languages interpret scripts on the server side rather than client-side like JavaScript.

Doing so provides a customized interface for each user and adds functionality beyond what HTML can offer. Scripting languages are programming languages that are interpreted rather than needing to be compiled before execution.

2.3.2 About CSS

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed. This tutorial will teach you CSS from basic to advanced.

CSS stands for Cascading Style Sheets, CSS describes how HTML elements are to be displayed on screen, paper, or in other media. CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files. A CSS rule consists of a selector and a declaration block.

CSS syntax

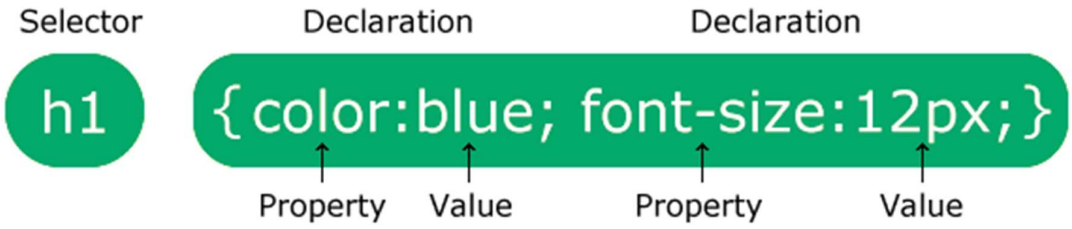

Fig 2.3.2 Syntax of CSS

The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

Example Explained

p is a selector in CSS it points to the HTML element you want to style: <p>.

color is a property, and red is the property value

text-align is a property, and center is the property value

CSS Selectors

CSS selectors are used to "find" or select the HTML elements you want to style. We can divide CSS selectors into five categories:

- Simple selectors select elements based on name, id, class
- Combinator selectors select elements based on a specific relationship between them
- Pseudo-class selectors select elements based on a certain state
- Pseudo-elements selectors select and style a part of an element
- Attribute selectors select elements based on an attribute or attribute value
- This page will explain the most basic CSS selectors.

The CSS element Selector

The element selector selects HTML elements based on the element name.

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash # character, followed by the id of the element.

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file! Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.

Inline CSS

An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

2.3.3 About SQL

An SQL database, also known as a relational database, is a type of database management system (DBMS) that uses Structured Query Language (SQL) for managing and manipulating data. SQL databases are organized into tables that consist of rows and columns, where each row represents a record and each column represents a specific attribute or field of the data.

Here are some key concepts and components of an SQL database:

Tables: Tables are the fundamental component of an SQL database. They consist of rows (also called records or tuples) and columns (also called fields or attributes). Each table represents a specific entity or concept in the database schema, and each row in the table represents an instance or entry of that entity.

Schema: The database schema defines the structure and organization of the database. It includes the tables, their columns, data types, relationships between tables, constraints, and other metadata. The schema provides a blueprint for how the data is stored and accessed in the database.

Data Types: SQL databases support various data types, such as integer, floating-point numbers, character strings, dates, and Boolean values. These data types define the kind of values that can be stored in each column of a table.

Primary Keys: A primary key is a unique identifier for each record in a table. It ensures that each row has a unique identity and provides a way to reference and relate records across different tables. Typically, primary keys are created using a single or a combination of columns.

Relationships: SQL databases support relationships between tables through keys. The most common relationship types are:

One-to-One: Each record in one table is related to exactly one record in another table.

One-to-Many: Each record in one table can be related to multiple records in another table.

Many-to-Many: Multiple records in one table can be related to multiple records in another table, requiring a join table.

Chapter 3

SYSTEM ANALYSIS

3.1 Software Requirement Specification

3.1.1 Introduction

This Software Requirement Specification document provides a complete description of all the functionalities and the specifications of the "Car Rental System". The following section provides an overview of the derived Software Requirements Specification (SRS) for the application. To begin with, the purpose of the document is presented, and its intended audience is outlined. Subsequently, the scope of the project specified by the document is given with a particular focus on what the resultant software will do and the relevant benefits associated with it. The nomenclature used throughout the SRS is also offered. To conclude, a complete document overview is provided to facilitate increased reader comprehension and navigation.

3.1.2 Purpose

The purpose of a car rental system is to facilitate the rental process of vehicles to individuals or businesses. Here are some key purposes and benefits of a car rental system:

Vehicle Availability: The system ensures that customers can easily access information about available vehicles, including types, models, and rental rates. It allows customers to make informed decisions and choose the vehicle that best suits their needs.

Convenient Booking and Reservations: The system provides a user-friendly interface for customers to make online bookings and reservations at their convenience. This eliminates the need for manual paperwork or phone calls and allows customers to check availability, compare options, and secure their reservations quickly.

Streamlined Rental Process: A car rental system automates various aspects of the rental process, such as customer registration, vehicle check-in/out, documentation, and payment processing. This streamlines the process, reduces manual errors, and saves time for both customers and rental providers.

Efficient Fleet Management: The system helps rental companies effectively manage their vehicle fleet. It provides real-time information on vehicle availability, tracks rental durations, monitors maintenance schedules, and generates reports on fleet utilization. This enables rental companies to optimize their inventory, minimize downtime, and maximize profitability.

Improved Customer Experience: By offering a user-friendly and efficient booking process, a car rental system enhances the overall customer experience. Customers can easily find and reserve vehicles, view rental details, and access support if needed. It leads to increased customer satisfaction and loyalty.

Financial Management: A car rental system includes features for payment processing, invoicing, and generating financial reports. It enables rental companies to accurately track revenues, manage security deposits, handle refunds, and streamline their financial operations.

Business Insights and Analytics: The system provides data and analytics on various aspects of the rental business, such as rental patterns, customer preferences, revenue analysis, and fleet performance. These insights help rental companies make data-driven decisions, identify opportunities for growth, and optimize their operations.

3.1.3 Scope

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

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

3.2 Functional Requirements And Non Functional Requirements

3.2.1 Functional Requirements

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

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

- **Customer's registration:** The system should allow new users to register online and generate membership card
- **Online reservation of cars:** Customers should be able to use the system to make booking and online reservation.
- Automatic update to database once reservation is made or new customer registered Whenever there's new reservation or new registration, the system should be able update the database without any additional efforts from the admin.

3.2.2 Non-Functional Requirements

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

- **Security:** The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the company can gain access to the company's secured page on the system; and only users with valid password and username can login to view user's page.
- **Availability:** This system should always be available for access at 24 hours, 7 days a week. Also in the occurrence of any major system malfunctioning, the system should be available in 1 to 2 working days, so that the business process is not severely affected.

- **Ease of use:** Considered the level of knowledge possessed by the users of this system, a simple but quality user interface should be developed to make it easy to understand and required less training.

Chapter 4

HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements:

XAMPP runs on Windows 10 hence the minimum hardware requirements of windows 10 considered below:

- **Processor:** 1 gigahertz GHz or faster processor.
- **RAM:** 1 gigabyte GB for 32-bit or 2 GB for 64-bit.
- **Hard disk space:** 16 GB for 32-bit OS or 20 GB for 64-bit OS.
- **Graphics card:** DirectX 9 or later with WDDM 1.0 driver.

4.2 Software Requirements:

- Operating System : Windows /iOS/Unix 17
- Web Browser : IE/Google Chrome/Firefox
- Technology : PHP
- Tools : XAMPP
- Web Design : HTML, CSS, JAVASCRIPT
- Back End : MYSQL
- Scripting Language : PHP

Chapter 5

SYSTEM DESIGN

5.1 Modules

i. **Admin**

ii. **Registered users**

Admin :

Admin is the super user of the website who can manage everything on the website.

Admin Features:

- Admin can create car brands
- Manage Car Brands(Edit, Delete)
- Post Car
- Manage car(Edit,Delete)
- Manage Booking(Admin can confirm and Cancel Booking)
- Manage Contact us Query
- Admin Can the details of registered users
- Admin can also update the page content
- Admin can update the contact us details 22
- Manage Subscribers
- Admin Dashboard(Admin can view the count of reg users, total booking, total subscribers, total queries etc)
- Change Password(admin can change own password)
- Logout

Registerd Users:

Anyone can register through the registration page. After a successful registration user can log in with valid email and password. User can recover own password by providing some registered info. After successful login user can do the following things:

- Car Booking
- View Car booking history
- Update His /Her profile
- Update his/her password
- View details of car
- Logout

5.2 Use case Diagram

A use case diagram is a visual representation of the functional requirements of a system, illustrating the interactions between users (actors) and the system itself. It provides a high-level overview of how users interact with the system to achieve specific goals or perform certain tasks. Use case diagrams are commonly used in software development and systems analysis to capture and communicate the system's behavior from a user's perspective.

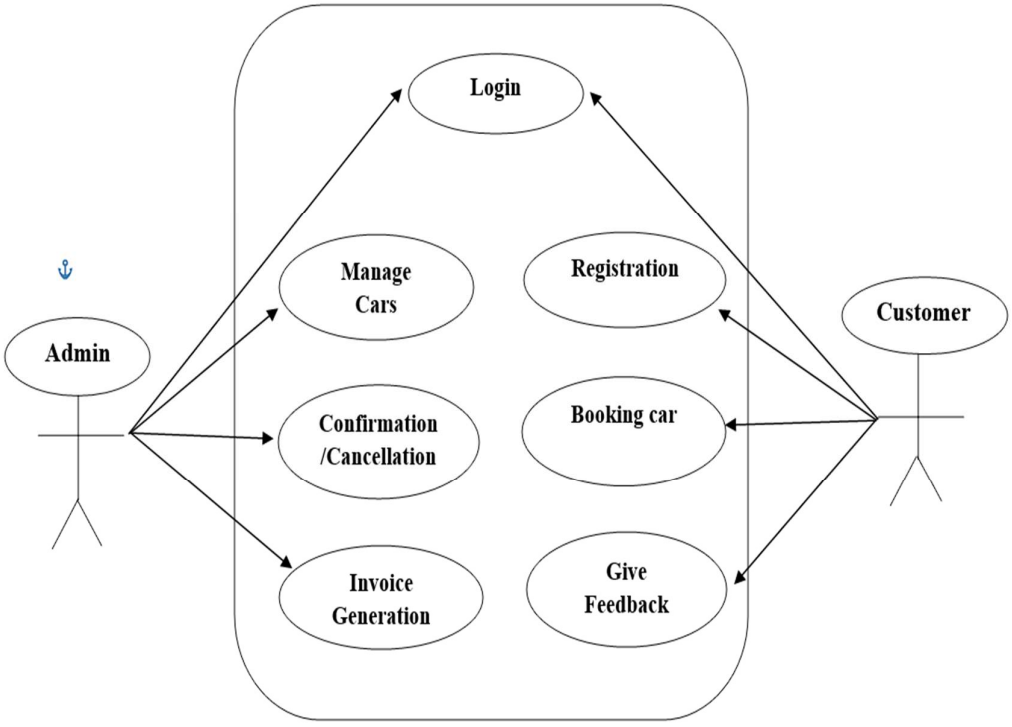

Fig. 5.2 Use Case Diagram

5.3 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) is a graphical representation that depicts the information flow and the transforms that are applied as data moves from input to output.

5.3.1 Zero Level Data Flow Diagram :

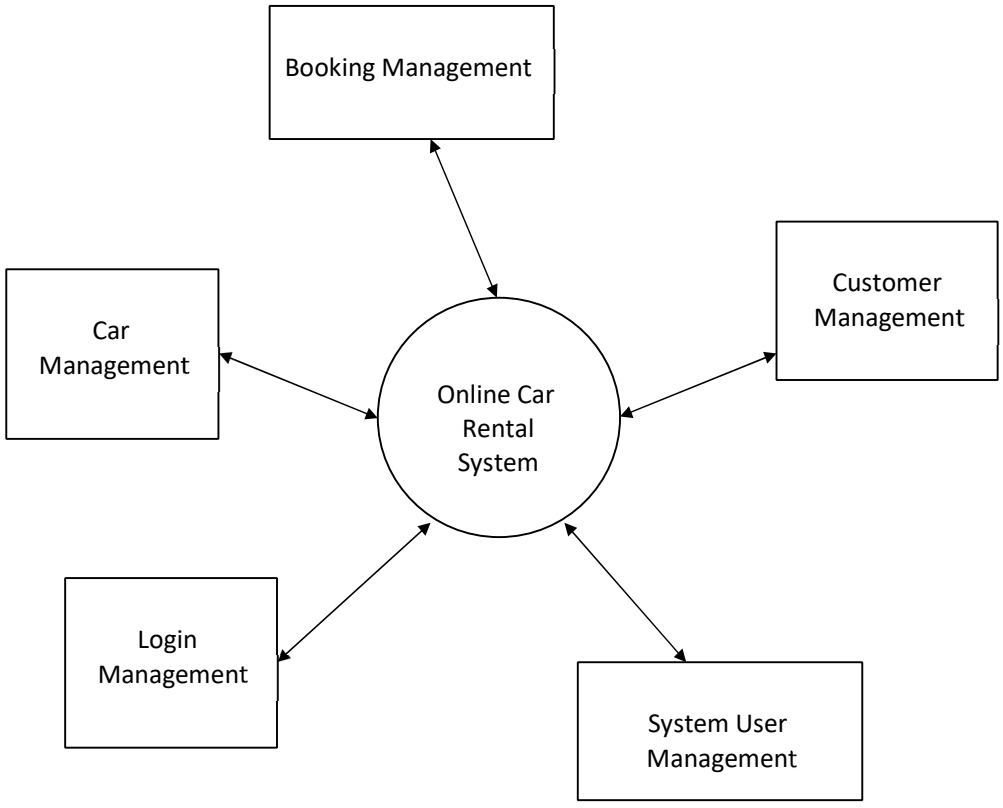

Fig. 3.2.1 Zero Level Data Flow Diagram

Zero Level DFD of online car rental system, it elaborate high level process of online car rental system. It is overview of whole online car rental system there are some high level entities for the process of car rental system.

3.2.2 First Level Data Flow Diagram

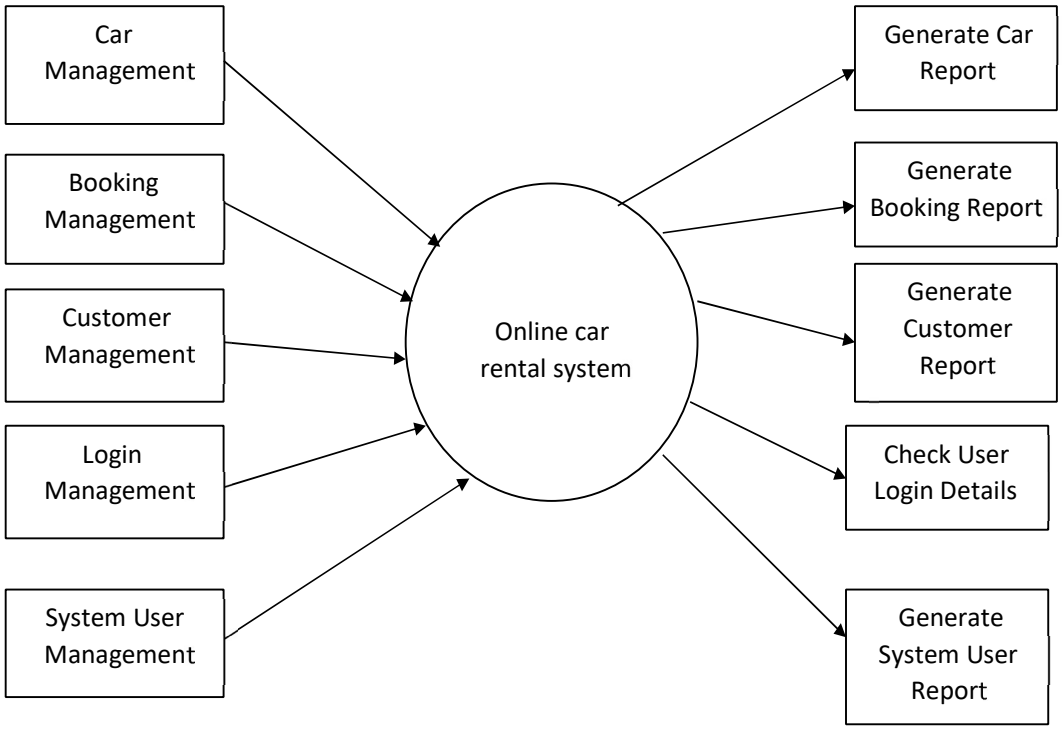

Fig. 3.2.2 First Level Data Flow Diagram

1st Level DFD of online car rental system shows how the system is divided into sub system, each of which deals with one or more of the data flows to or from an external agent which together provide all the functionality of online car rental system as whole, above are some given entities and output of 1st level.

5.3.3 Second Level Data Flow Diagram

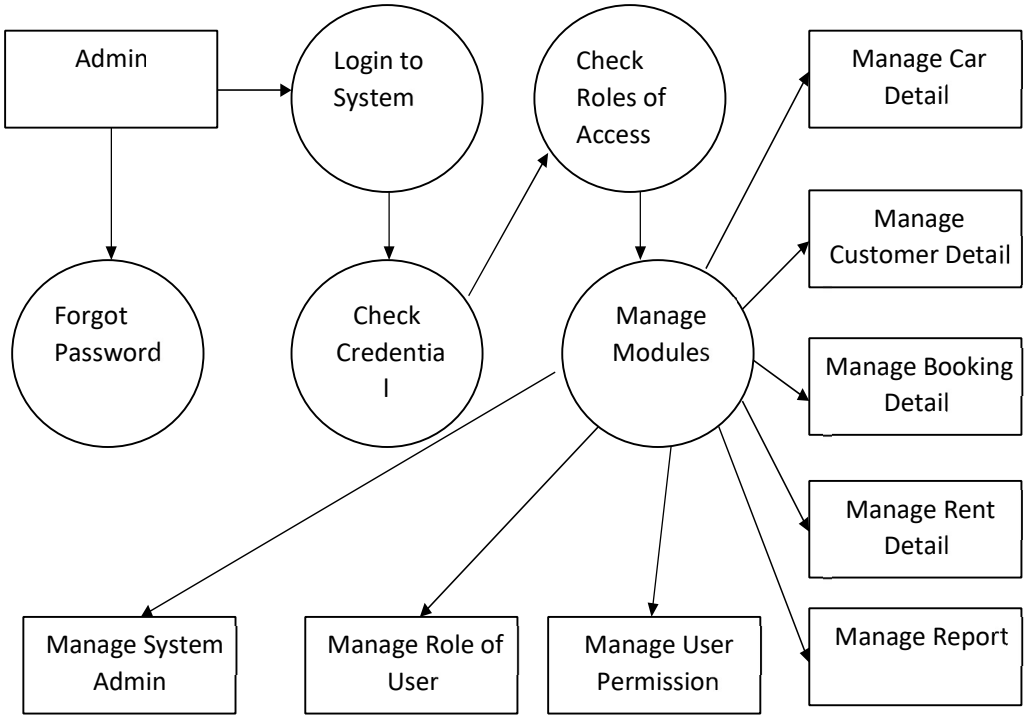

Fig. 5.3.3 Second Level Data Flow Diagram

In a second level DFD for a car rental system, the main processes are typically divided into smaller, more detailed sub-processes. Here is a general overview of the components you might find in such a diagram

5.4 E-R Diagram:

The ER diagram means the entity-relationship diagram in short ERD. It shows all kinds of possible relational data with the user and the admin. The entity-relationship diagram of Car Rental System shows all the visual instrument of database tables and the relations between Booking, Car Routes, Cars, Car booking dates, etc.

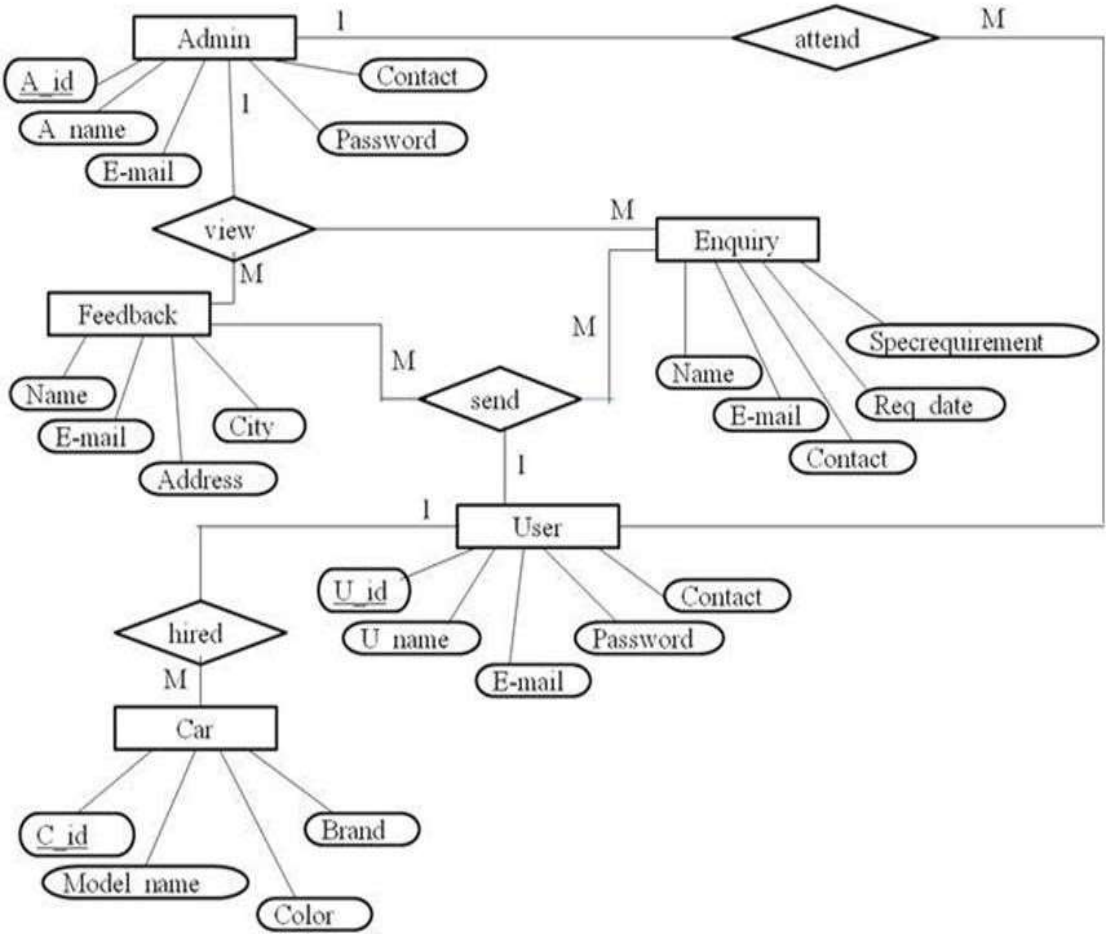

Fig. 5.4 E-R Diagram

5.5 Database Tables

The created tables explain the data hierarchy of the whole project from the admin and the user side. Database tables show the relationship between all the data and it is extremely important to have database design on the website. All the data were arranged in a specific way that both the user and admin can use them when necessary.

Table Name	Admin
Description	This table stores information about Admin.
Primary Key	Id
Foreign Key	-

Sr. No	Field Name	Data type (Size)	Constraints	Description
1	id *(Primary)*	int (11)	Primary Key	It stores Admin id.
2	Username	varchar(100)	Not Null	It stores admin user name.
3	Password	varchar(100)	Not Null	It stores the password of Admin.
4	Updation date	Timestamp	NotNull	It stores the profile updating date.

Table. 5.5.1 Admin Database Table

Table Name	Users
Description	This table provides the information about User registration.
Primary Key	Id
Foreign Key	-

Sr. No	**Field Name**	**Data type(Size)**	**Constraints**	**Description**
1	id *(Primary)*	int(11)	Primary Key	It stores User id.
2	FullName	varchar(120)	Null	It stores User name.
3	Email Id	varchar(100)	Null	It stores email address of User.
4	Password	varchar(100)	Null	It stores Password.
5	ContactNo	char(11)	Null	It stores Contact no.
6	Dob	varchar(100)	Null	It stores Birthdate.

Table. 5.5.2 User Database Table

Table Name	Tblvehicles
Description	This table provides the information about cars.
Primary Key	Id
Foreign Key	-

Sl.No	**Field Name**	**Data Type(size)**	**Constraints**	**Description**
1	id *(Primary)*	int(11)	Primary Key	It stores User id.
2	VehicleTitle	varchar(150)	Null	It stores vehicle title.
3	VehicleBrand	int(11)	Null	It stores vehicle brand id.
4	PricePerDay	int(11)	Null	It stores vehicle rent per day.
5	FuelType	varchar(100)	Null	It stores fuel type of vehicle.
6	Vimage1	varchar(120)	Null	It stores vehicle image 1.
7	Vimage2	varchar(120)	Null	It stores vehicle image 2.

Table. 5.5.3 Vehicle Database Table

Table Name	Tblbooking
Description	This table provides the information about booking.
Primary Key	Id
Foreign Key	-

Sl.No	**Field Name**	**Data Type(size)**	**Constraints**	**Description**
1	Id(primary)	Int(11)	Primary Key	It Stores Booking Id
2	User Email	Varchar(100)	Null	It Stores User Email
3	Vehicle Id	Int(11)	Null	It Stores Vehicle Id
4	From Date	Varchar(20)	Null	It Stores Booking from date
5	To Date	Varchar(20)	Null	It Stores Booking date
6	Message	Varchar(255)	Null	It Stores Message
7	Status	Int(11)	Null	It Stores Confirmation and cancelation Status

Table. 5.5.4 Vehicle Booking Table

Table Name	Tbltestimonial
Description	This table store information about feedback.
Primary Key	F_Id
Foreign Key	-

Sr. No	**Field Name**	**Data type(Size)**	**Constraints**	**Description**
1	id *(Primary)*	int(11)	Primary Key	It stores feedback id.
2	UserEmail	varchar(100)	Not Null	It stores user email.
3	Testimonial	medium text	Not Null	It stores feedback.
4	PostingDate	Timestamp	Not Null	It stores posting date of feedback.
5	Status	int(11)	Not Null	It stores status(0 for inactive and 1 active).

Table. 5.5.5 Testimonial Table

Chapter 6

SYSTEM IMPLEMENTATION

The website is fully dynamic. There is no need of knowing coding for the user and the admin. Basic uses of the internet are required for the client. XAMP is used as a software for running the server locally. Visual studio code is used for viewing and editing the code. Hence, the displayed Figure illustrates the whole project such as the homepage of the web page, car listing, booking, manage testimony as well as inquiry and canceling for the user. Moreover, the dashboard can be used for changing passwords, creating new users, managing dashboards for the admin panel. The graphical interface of XAMP, Bracket, and connected database are snapped in Figures 6.1 to 6.3. Figure 6.1 is the screenshot of XAMP. Figure 6.2 is the view of the code in Visual studio codestudio code is used to write and edit code when necessary. Figure 6.3 shows the connected database in XAMPP. The whole database was created in XAMPP IDE as well.

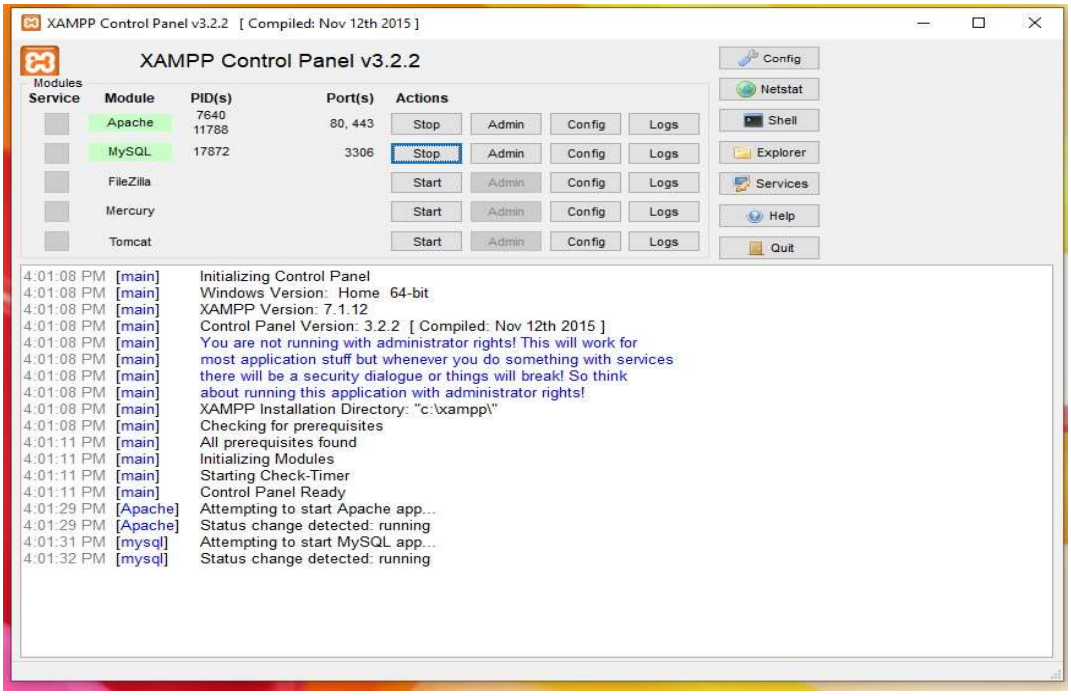

FIG. 6.1 XAMPP Control Panel

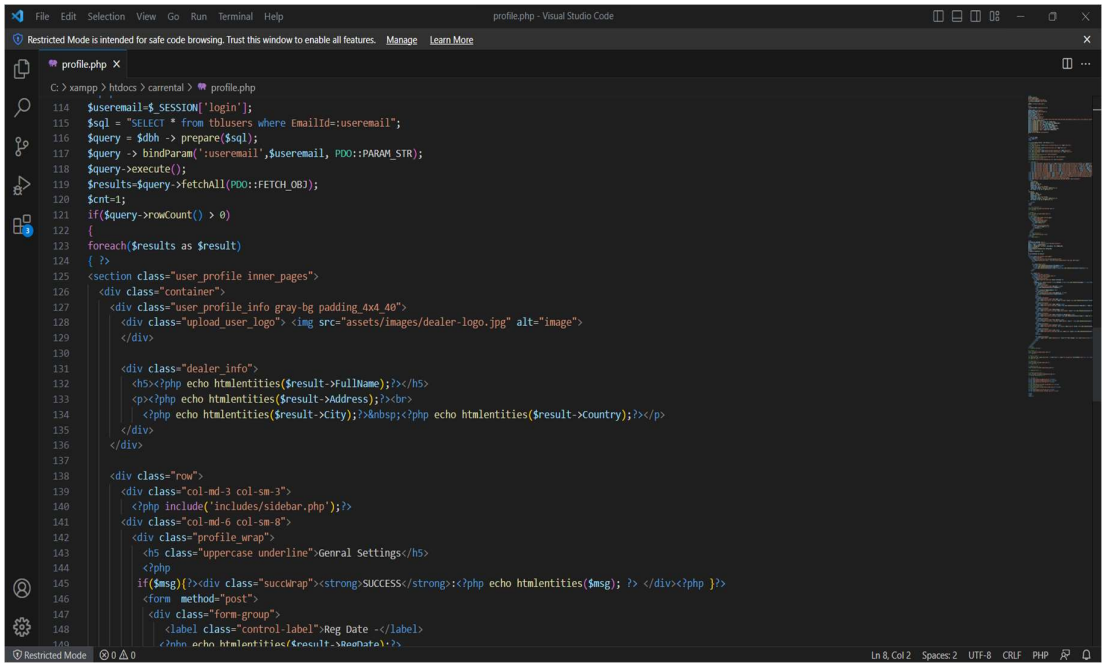

Fig. 6.2 View of the code in Visual Studio Code

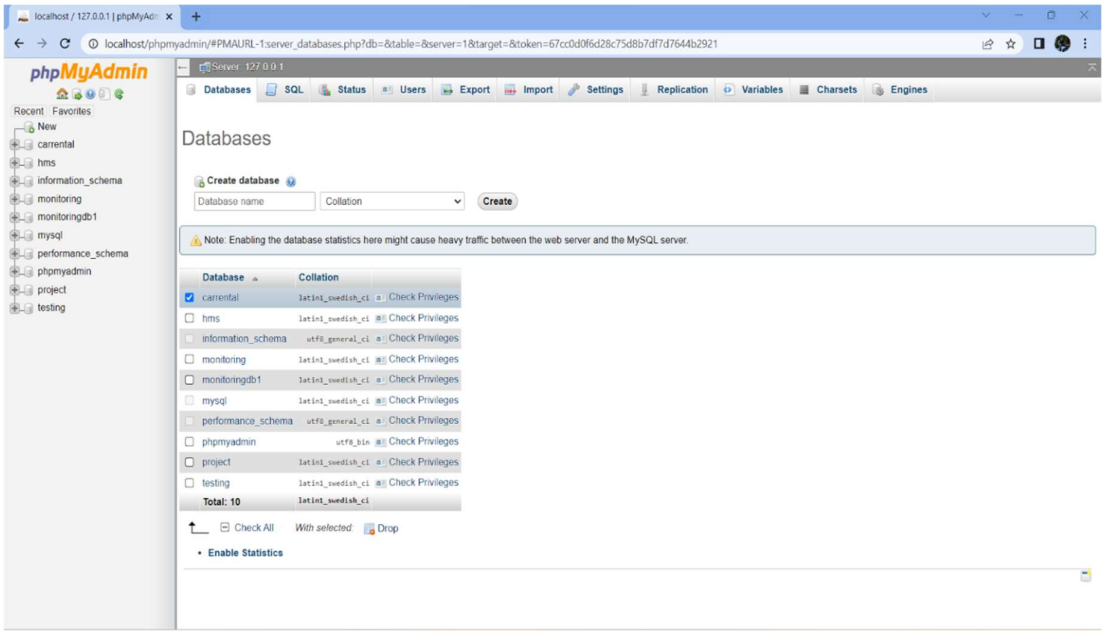

Fig. 6.3 Database connected in XAMPP

6.1 LOGIN SCRIPT

```
<?php

if(isset($_POST[‘login’]))

{

$email=$_POST[‘email’];

$password=md5($_POST[‘password’]);

$sql =”SELECT EmailId,Password,FullName FROM tblusers WHERE EmailId=:email and Password=:password”;

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

$query-> bindParam(‘:email’, $email, PDO::PARAM_STR);

$query-> bindParam(‘:password’, $password, PDO::PARAM_STR);

$query-> execute();

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

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

{

$_SESSION[‘login’]=$_POST[‘email’];

$_SESSION[‘fname’]=$results->FullName;

$currentpage=$_SERVER[‘REQUEST_URI’];

echo “<script type=’text/javascript’> document.location = ‘$currentpage’; </script>”;

} else{

  echo “<script>alert(‘Invalid Details’);</script>”;

}
```

```
}

?>

<div class=”modal fade” id=”loginform”>

<div class=”modal-dialog” role=”document”>

<div class=”modal-content”>

<div class=”modal-header”>

<button type=”button” class=”close” data-dismiss=”modal” aria-label=”Close”><span aria-hidden=”true”>&times;</span></button>

<h3 class=”modal-title”>Login</h3>

</div>

<div class=”modal-body”>

<div class=”row”>

<div class=”login_wrap”>

<div class=”col-md-12 col-sm-6”>

<form method=”post”>

<div class=”form-group”>

<input type=”email” class=”form-control” name=”email” placeholder=”Email address*”>

</div>

<div class=”form-group”>

<input type=”password” class=”form-control” name=”password” placeholder=”Password*”>

</div>
```

<div class=”form-group checkbox”>

<input type=”checkbox” id=”remember”>

<input type=”submit” name=”login” value=”Login” class=”btn btn-block”>

</form>

<div class=”modal-footer text-center”>

Don’t have an account? Signup Here

Forgot Password ?

6.2 REGISTRATION SCRIPT

```
<?php

//error_reporting(0);

if(isset($_POST[‘signup’]))

{

$fname=$_POST[‘fullname’];

$email=$_POST[‘emailid’];

$mobile=$_POST[‘mobileno’];

$password=md5($_POST[‘password’]);

$sql=”INSERT INTO  tblusers(FullName,EmailId,ContactNo,Password)
VALUES(:fname,:email,:mobile,:password)”;

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

$query->bindParam(‘:fname’,$fname,PDO::PARAM_STR);

$query->bindParam(‘:email’,$email,PDO::PARAM_STR);

$query->bindParam(‘:mobile’,$mobile,PDO::PARAM_STR);

$query->bindParam(‘:password’,$password,PDO::PARAM_STR);

$query->execute();

$lastInsertId = $dbh->lastInsertId();

if($lastInsertId)

{

echo “<script>alert(‘Registration I. Now you can login’);</script>”;

}
```

```
else

{

echo “<script>alert(‘Something went wrong. Please try again’);</script>”;

}

}

?>

<script>

function checkAvailability() {

$(“#loaderIcon”).show();

jQuery.ajax({

url: “check_availability.php”,

data:’emailid=’+$(“#emailid”).val(),

type: “POST”,

success:function(data){

$(“#user-availability-status”).html(data);

$(“#loaderIcon”).hide();

},

error:function (){}

});

}

</script>

<script type=”text/javascript”>
```

```
function valid()

{

if(document.signup.password.value!= document.signup.confirmpassword.value)

{

alert("Password and Confirm Password Field do not match  !!");

document.signup.confirmpassword.focus();

return false;

}

return true;

}

</script>

<div class="modal fade" id="signupform">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h3 class="modal-title">Sign Up</h3>

</div>

<div class="modal-body">

<div class="row">

<div class="signup_wrap">
```

<div class=”col-md-12 col-sm-6”>

<form method=”post” name=”signup” onSubmit=”return valid();”>

<input type=”text” class=”form-control” name=”fullname” placeholder=”Full Name” required=”required”>

<input type=”text” class=”form-control” name=”mobileno” placeholder=”Mobile Number” maxlength=”10” required=”required”>

<input type=”email” class=”form-control” name=”35mailed” id=”35mailed” onBlur=”checkAvailability()” placeholder=”Email Address” required=”required”>

<input type=”password” class=”form-control” name=”password” placeholder=”Password” required=”required”>

<input type=”password” class=”form-control” name=”confirmpassword” placeholder=”Confirm Password” required=”required”>

<div class=”form-group checkbox”>

<input type=”checkbox” id=”terms_agree” required=”required” checked=””>

<label for=”terms_agree”>I Agree with Terms and Conditions</label>

<input type=”submit” value=”Sign Up” name=”signup” id=”submit” class=”btn btn-block”>

</form>

<div class=”modal-footer text-center”>

Already got an account? Login Here

6.5 Database connectivity

<?php

$con = new mysqli(“localhost”, “root”, “”, “carrental_db”);

?>

Chapter 7

SYSTEM TESTING

Testing is a crucial process in software development that involves evaluating a system or application to ensure it meets the specified requirements, functions as intended, and operates without errors or issues. The primary goal of testing is to identify defects or bugs in the software so that they can be fixed before the product is released to users.

Testing involves executing the software under controlled conditions and observing its behavior to compare the actual results with the expected results. The process aims to uncover any discrepancies, errors, or failures in the software's functionality, performance, usability, security, or other quality attributes.

Testing can be performed at different levels of software development, including unit testing, integration testing, system testing, and acceptance testing. Each level focuses on different aspects of the software and helps identify issues at different stages of development.

There are various types of testing techniques and methodologies that can be employed, such as functional testing, performance testing, security testing, usability testing, and regression testing. The choice of testing techniques depends on the nature of the software, its requirements, and the goals of the testing process.

Overall, testing plays a critical role in ensuring the quality, reliability, and robustness of software systems. It helps identify and rectify issues early in the development process, minimizing risks, and improving the overall user experience.

SL NO.	DATA OUTPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	PASS/ FAIL
1	All files are empty	Error message: *Indicates compulsory field*	Error message: *Indicates compulsory field*	Pass
2	Email	Error message:	Error message:	Pass

		Invalid email address	Invalid email address	
3	Password and confirm password	Error message: Both password doesn't match	Error message: Both password doesn't match	Pass
4	Login	Login in the system should be try with the login assigned by the admin and the correct password.	Login should be successful and the user should enter into the system.	Fail
		The system give an error and denied from the login.	Login should fail with an **error 'Invalid Details'**	Pass
5	User	Login should be allow and admin get admin home page.	Login successfully and admin get its admin home page.	Pass
		Login should be allow and user get visitor side user page.	Login successfully and user get its user home page.	Pass
6	Validation Test cases	Pre-defined format must be required in control.	System give error to enter valid output.	Pass
		Enter data in a compulsory field with required field validations.	Data must be filled in compulsory field otherwise its messages are displayed.	Pass

Table. 7.1 Test Case table

Chapter 8

RESULTS & SNAPSHOTS

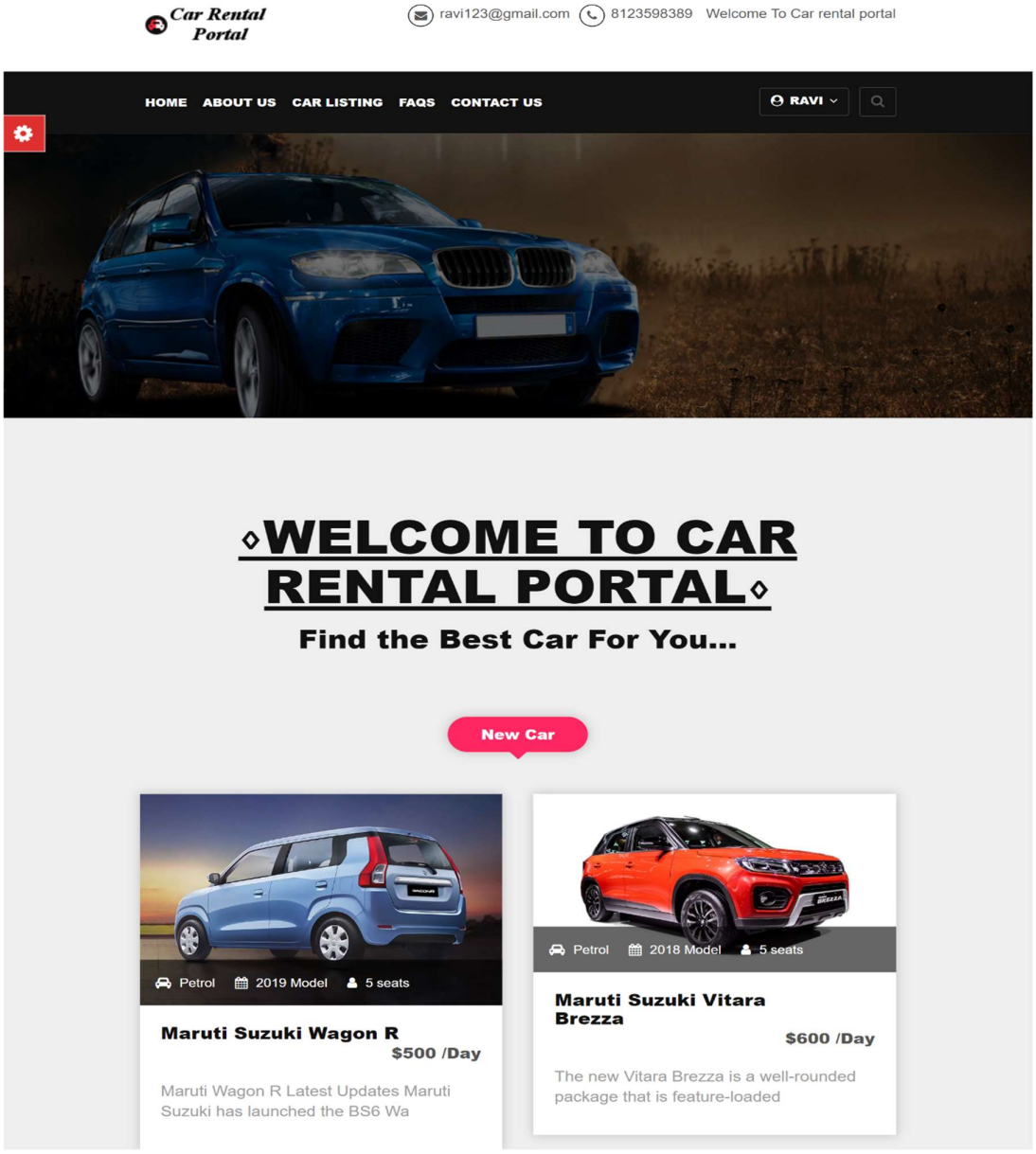

Fig. 8.1 Home page

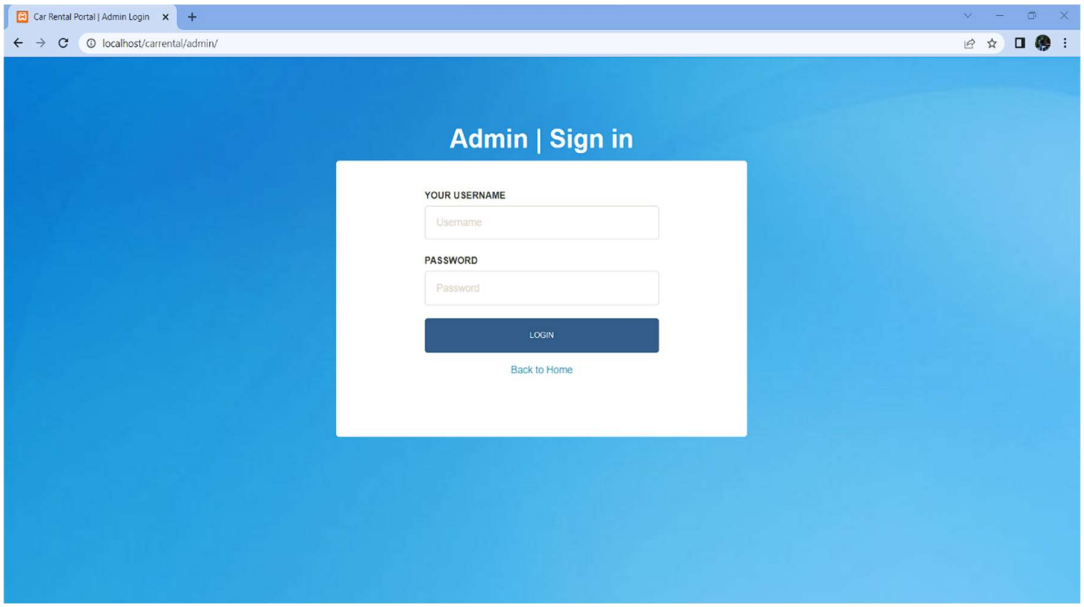

Fig. 8.2 Login page

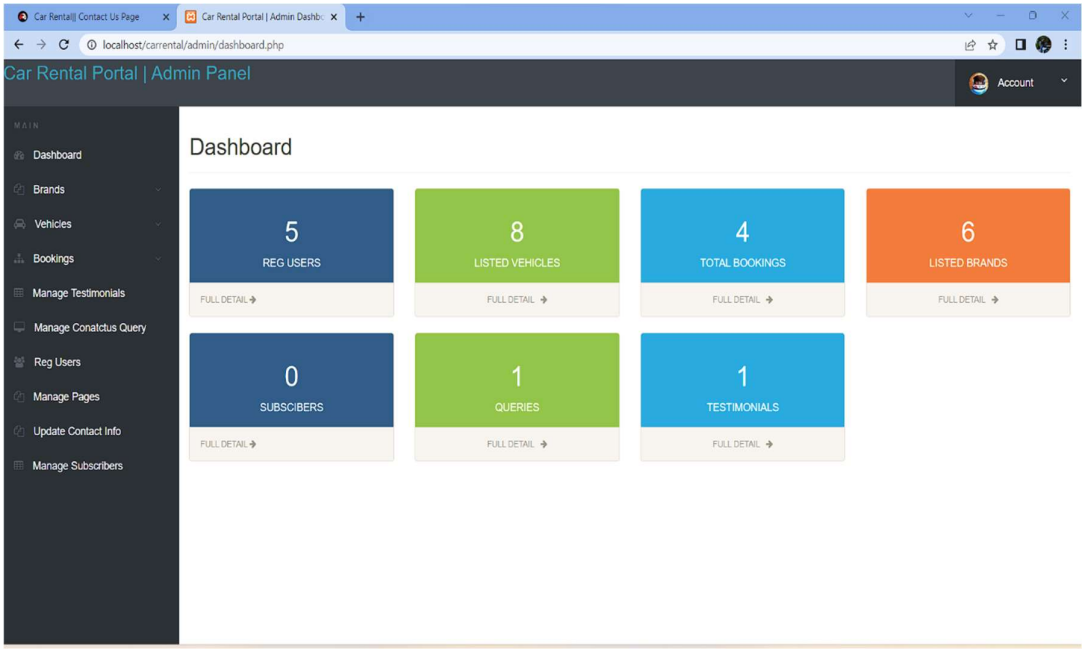

Fig. 8.3 Admin Panel

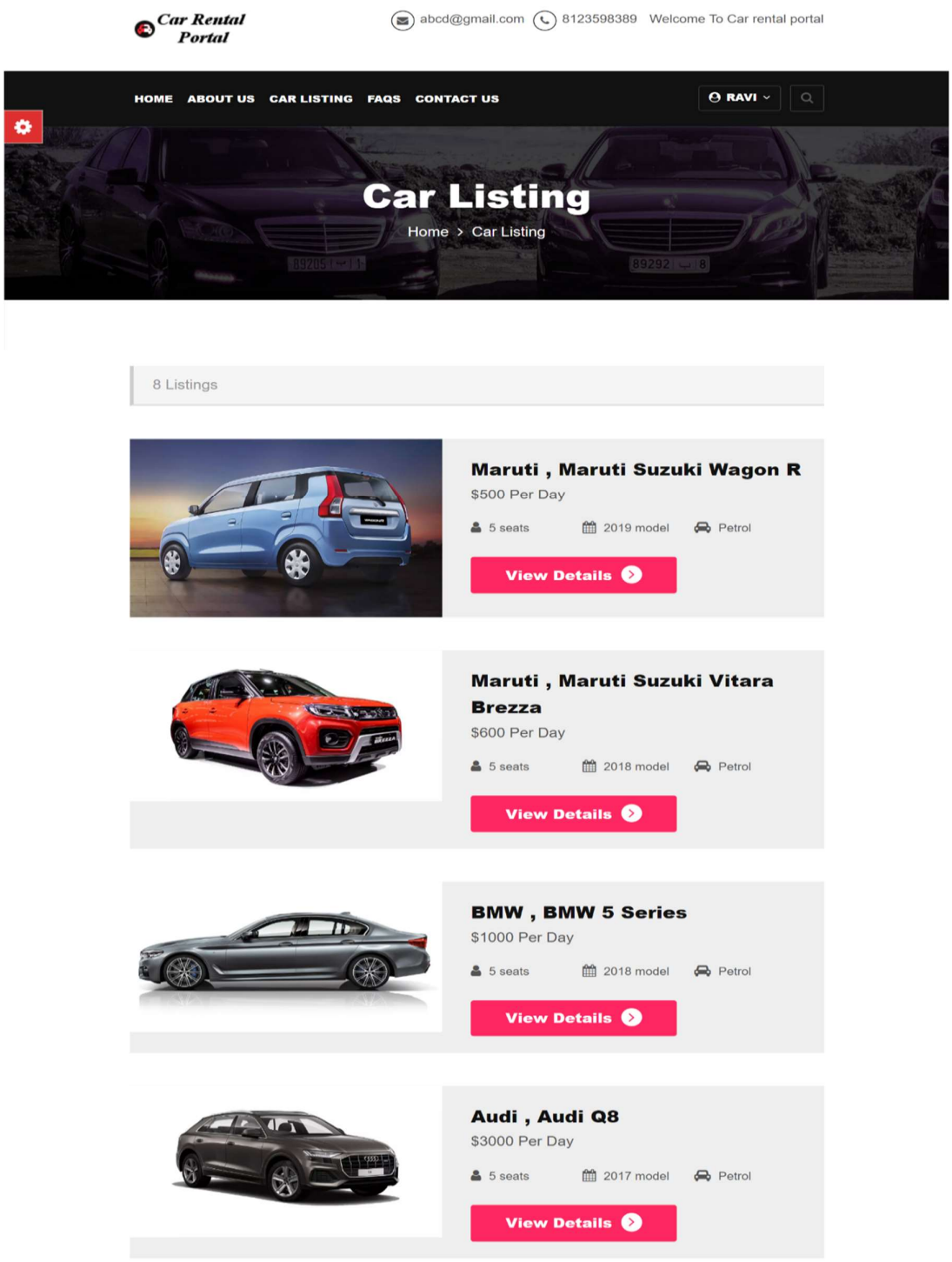

Fig. 8.4 Car listings

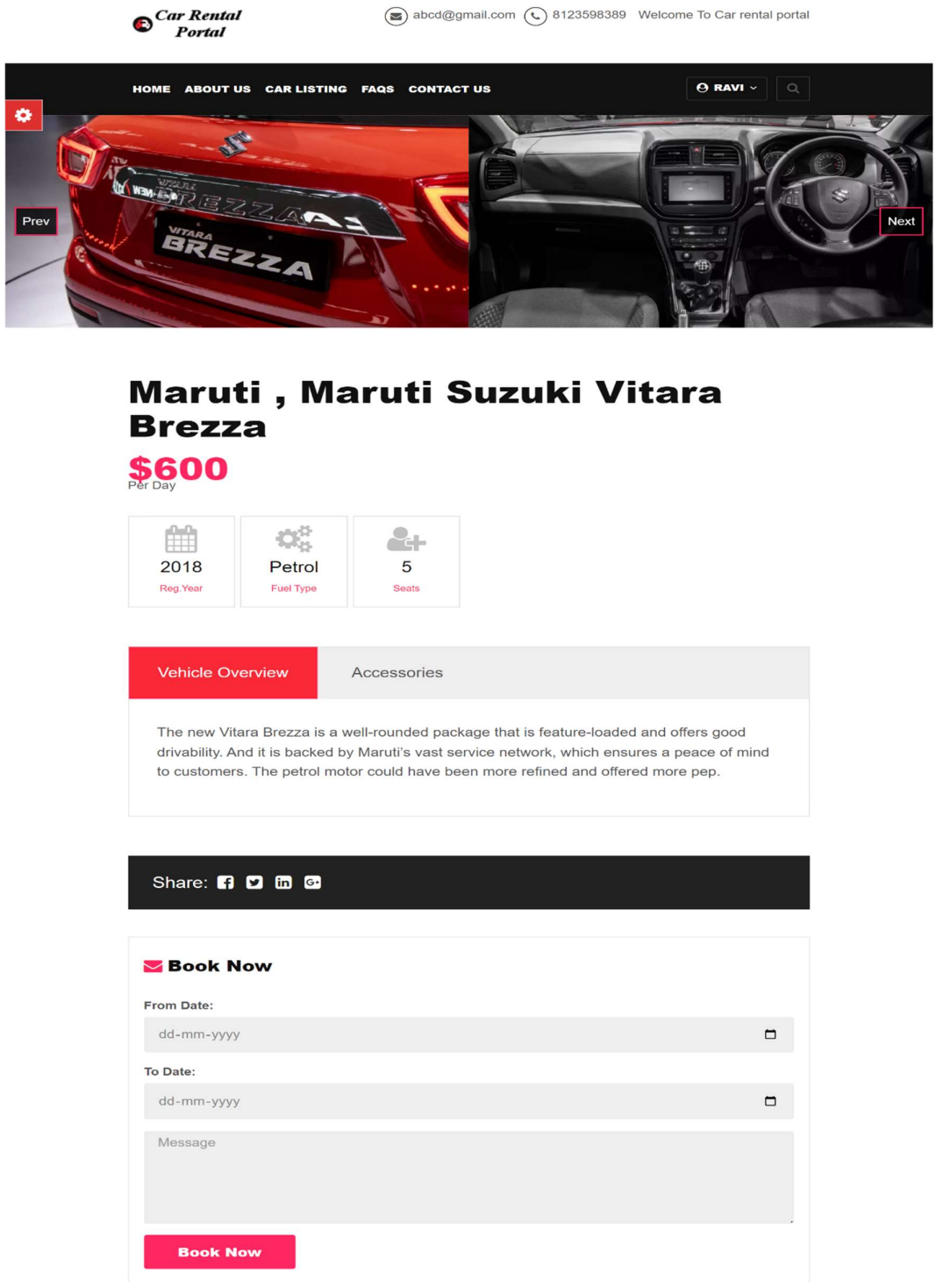

Fig. 8.5 Car booking page

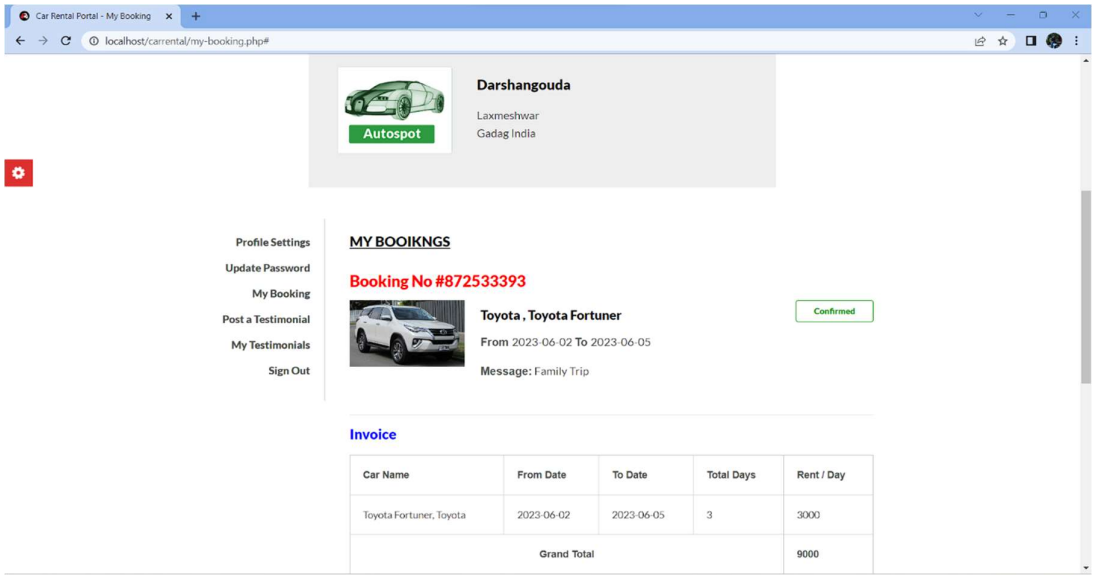

FIG. 8.6 Booking confirmation

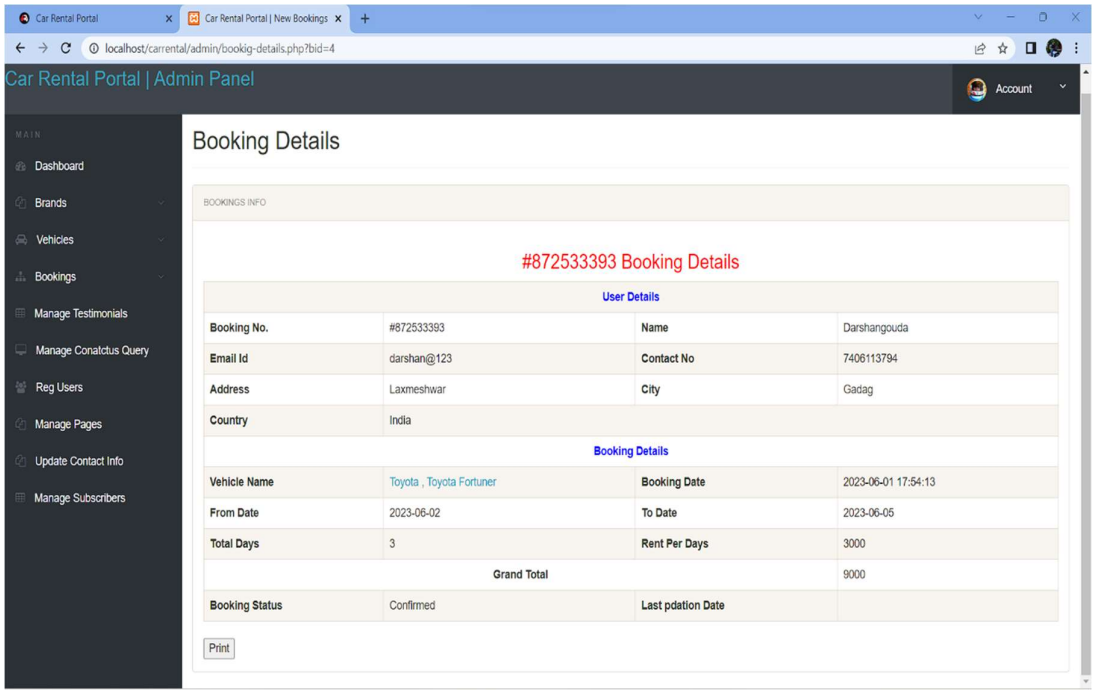

Fig. 8.7 Receipt generation page

Chapter 9

CONCLUSION & FUTURE ENHANCEMENTS

9.1 Conclusion

Car rental business has emerged with a new goodies compared to the past experience where every activity concerning car rental business is limited to a physical location only. Even though the physical location has not been totally eradicated; the nature of functions and how these functions are achieved has been reshaped by the power of internet.

Nowadays, customers can Booking cars online, rent car online, and have the car brought to their door step once the customer is a registered member or go to the office to pick the car. The proposed project Car Rental System is a web based online platform to provide car services for people in need. This project is implemented using PHP and MySQL.

The web based car rental system has offered an advantage to both customers as well as Car Rental Company to efficiently and effectively manage the business and satisfies customer's need at the click of a button.

9.2 Future Enhancements

In near future, we are planning to hire cars daily bases. So that clients can give their car to the customer on daily bases. We are planning to add new feature i.e. pay after the trip. We are working to increase automation in the system to increase user experience great and also we are trying to install GPS based online tracking system in cars for security.

Bibliography

Text Books Referred

• Software Engineering - R.S. Pressman

• PHP For Dummies

• PHP Begineers Guide By McGrawhill Publication

• Javascript By McGrawhill Publication References Used

References Used:

http://www.carrentingsolutions.com/

http://www.flashvortex.com/

http://www.imscart.com/car_rental_software.html/

http://www.wikipedia.org/

http://www.w3schools.com/

http://www.phpgurukul.com/