Chapter 1

INTRODUCTION

1.1 Background

Purpose

The car rental management system is a web based application which provides a user friendly and interactive system to book a car on rent per day basis. This web application facilitates an easy way for user to rent a car. It allows a user to rent a car without having to go to and offline car rental business and through the tedious process of searching the car he wants, bargaining the price for it and then renting the car. Not only does it make it easier for the user but also allows someone with an offline car rental business to provide its services on an online platform. Therefore this system makes it easier for both the client and the service provider.

Scope

The car rental management system can be accessed by an user who has an account. Anyone can create an account in the signup page. The user accounts are maintained in the MySQL database which also stores the history of the users' previous bookings. This record can be accessed by the admin only. It also allows the admin to access current active bookings.

1.2 Introduction about the project

This web application can be adapted by any car rental business to take their business to an online platform. It is aimed at making renting cars easier. Not only does it allow the business owner to provide their cars through this web application but also allows the user to rent a car from any location they want and for any number of days. The payment for booking can also be done online.

Chapter 2

ER DIAGRAM AND RELATIONAL SCHEMA DIAGRAM

2.1 Description of ER Diagram

An Entity-Relationship Diagram, usually referred to as an ER Diagram represents the attributes, entities and relationship in a relational schema diagram.

- Entities like Customer, Admin and Booking are represented using rectangular boxes in the ER Diagram.
- The attributes which characterize the entities are represented in ovals, each attached to the entity type using a straight line. The attribute which is designed as the primary key is identified by underlining it within the oval.
- Relationships like views, manages and does are represented in diamond boxes which are attached to the entity types participating in the relationship using straight lines.
- The total participation of the entities participating in a relationship is identified by two straight lines from the entity type to the diamond. Whereas, the partial participation is identified by a single line.
- The cardinality ratios are as follows:
    1. Customer : Booking is of the cardinality 1 : n as each customer may book a vehicle n number of times.
    2. Customer : Payment is of the cardinality 1 : n as each customer, in order to make a booking has to do payment for each booking and as established earlier each customer can make n bookings.
    3. Customer : Feedback is of the cardinality 1 : n as each customer can give n number of feedbacks.
    4. Customer : Models is of the cardinality 1 : n as each customer can view details of n number of car models that are made available.
    5. Booking : Payment is of the cardinality 1 : 1 as each booking is associated with a corresponding payment as vice versa.

6. Admin : Booking is of the cardinality 1 : n as each admin manages n number of bookings made by customers.
7. Admin : Models is of the cardinality 1:n as each admin can manage n Car models present in the inventory.
8. Admin : Feedback is of the cardinality 1 : n as each admin can view number of feedbacks given by the customers.

Fig 2.1 Shows the ER Diagram of Car Rental System with relationships and cardinality ratios.

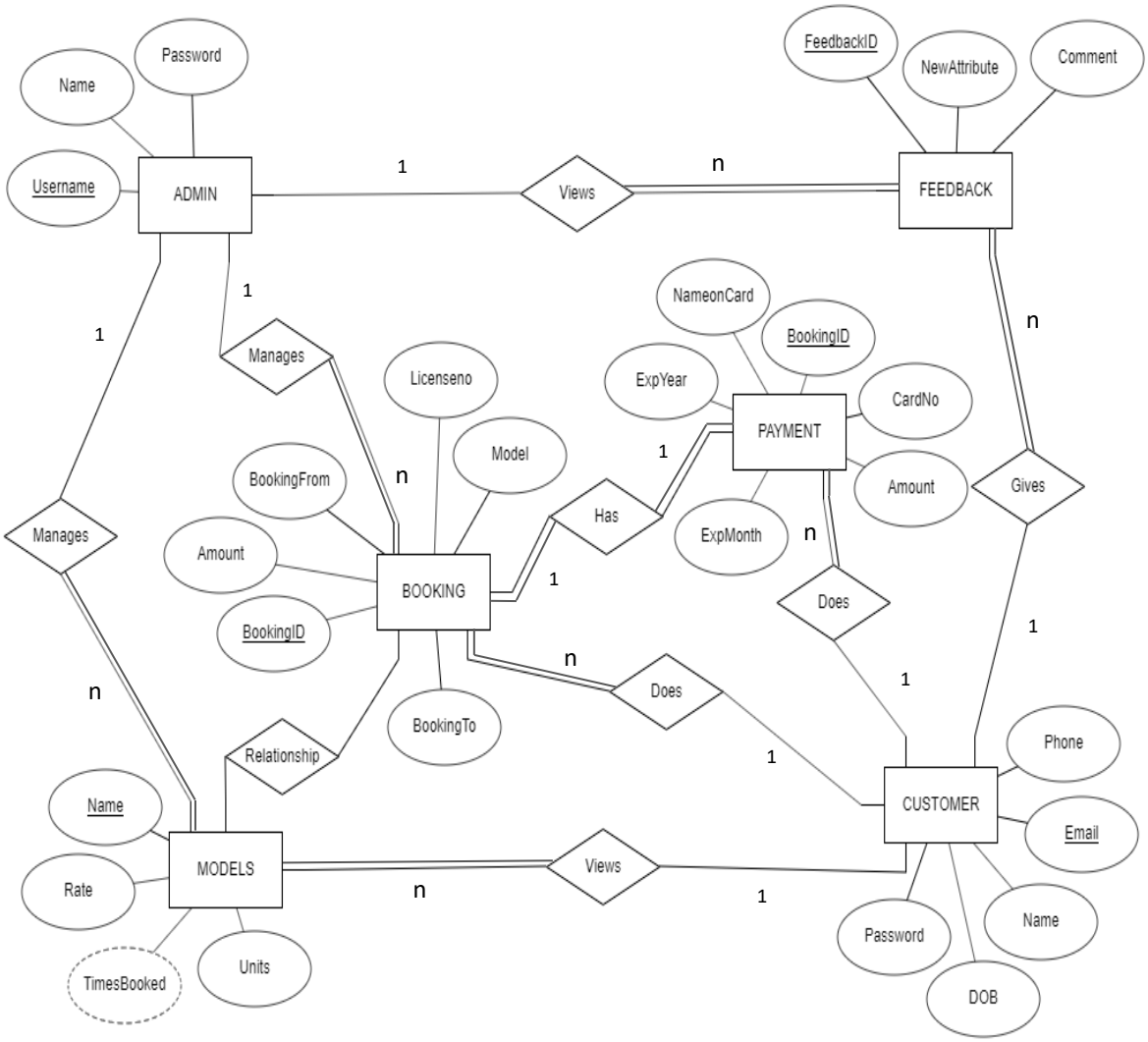


Fig 2.1 : ER Diagram of Car Rental System

2.2 Description of Relational Schema Diagram

The term database Schema refers to the description of the database that includes the database structure and various constraints on the database. The Schema diagram is in turn an illustrative display of the database schema. The primary keys are underlined and the referential integrity constraints are depicted by arrows pointing to the keys they reference.

Figure 2.2 shows the Relational Schema Diagram along with primary keys and referential integrity constraints.

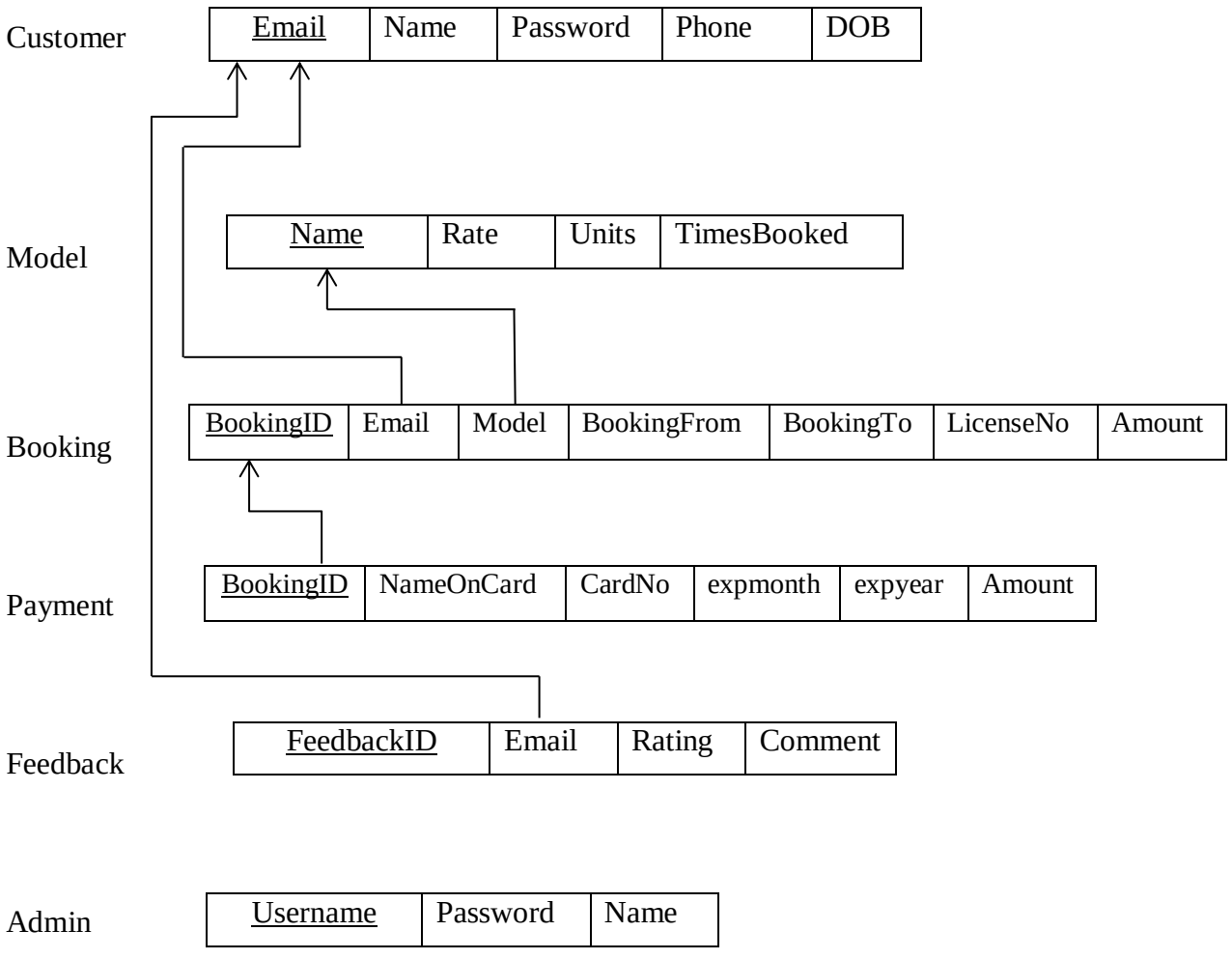


Fig 2.2 : Schema Diagram of Car Rental System

The list of tables are:

- Customer : This table stores the personal information concerning each registered user and their passwords. The Primary Key is Email as it is unique to every customer.
- Admin : This table stores the login details of each Administrator of the Car Rental System. The Primary Key is Username of the Administrator.
- Models : This table stores the details of each Car Model available for rent like their hiring rent per day, units of each model available. The name of each model acts as the Primary Key of the table.
- Booking : This table stores the details of each booking made by any customer. The Primary Key is a unique Booking ID for each booking.
- Payment : This table stores the payment information for every corresponding booking made. The Primary Key here is Booking ID as a payment is always mandatory for each Booking ID.
- Feedback : This table stores all the feedbacks given by the customers. The Primary Key is a unique Feedback ID for each feedback.

Chapter 3

SYSTEM DESIGN

3.1 Table Description

1. Customer

Field	Type	Null	Key	Extra
name	varchar(40)	NO		
email	varchar(40)	NO	PRI	
phone	varchar(10)	NO		
dob	date	NO		
password	varchar(20)	NO		

The above table stores the personal information concerning each registered user and their passwords.

2. Admin

Field	Type	Null	Key	Extra
username	varchar(20)	NO	PRI	
password	varchar(20)	NO		
name	varchar(20)	NO		

The above stores the login details of each Administrator of the Car Rental System. The Primary Key is Username of the Administrator.

3. Feedback

Field	Type	Null	Key	Extra
feedbackid	int(11)	NO	PRI	AUTO_INCREMENT
email	varchar(40)	NO		
rating	text	NO		
comment	varchar(100)	YES		

The above table stores all the feedbacks given by the customers. The Primary Key is a unique Feedback ID for each feedback.

4. Booking

Field	Type	Null	Key	Extra
bookingid	int(11)	NO	PRI	AUTO_INCREMENT
email	varchar(40)	NO		
model	varchar(20)	NO		
bookingfrom	date	NO		
bookingto	date	NO		
licenseno	varchar(15)	NO		
amount	int(11)	NO		

The above table stores the details of each booking made by any customer.

5. Payment

Field	Type	Null	Key	Extra
bookingid	int(11)	NO	PRI	AUTO_INCREMENT
nameoncard	varchar(40)	NO		
cardno	varchar(20)	NO		
expmonth	int(2)	NO		
expyear	int(4)	NO		
amount	int(11)	NO		

The above table stores the payment information for every corresponding booking made.

6. Models

Field	Type	Null	Key	Extra
name	varchar(15)	NO	PRI	
rate	int(11)	NO		
units	int(11)	NO		
timesbooked	int(11)	NO		

The above table stores the details of each Car Model available for rent like their hiring rent per day, units of each model available.

3.2 Stored Procedure and Triggers

Stored Procedure

A Stored Procedure is a set of SQL statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.

```
CREATE PROCEDURE ‘cancelbooking’ ( IN  ‘id’  INT )
NO SQL  SQL SECURITY DEFINER
delete from booking where bookingid = id;
```

The above stored procedure used in Car Rental System will delete the record from booking table corresponding to the booking which has been requested for cancellation from the customer. The procedure takes an argument, named as ‘id’ which is used to select the particular booking that needs to be deleted.

Triggers

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database.

Trigger 1 : newbooking

```
CREATE TRIGGER  ‘newbooking’
AFTER INSERT ON  ‘booking’
FOR EACH ROW
update models
set timesbooked = timesbooked + 1
where name = new.model;
```

The above trigger used in the application will perform its operation every time an insertion operation takes place in ‘booking’ table. This trigger will increment the attribute ‘timesbooked’ by 1 in the ‘models’ table which is used to keep a count of number of times a particular Car Model has been booked.

Trigger 2 : cancelbooking

```
CREATE TRIGGER ‘cancelbooking’
AFTER DELETE ON ‘booking’
FOR EACH ROW
update models
set timesbooked = timesbooked – 1
where name = old.model;
```

The above trigger used in the application will perform its operation every time a deletion operation takes place in ‘booking’ table. This trigger will decrement the attribute ‘timesbooked’ by 1 in the ‘models’ table which is used to keep a count of number of times a particular Car Model has been booked.

Chapter 4

IMPLEMENTATION

4.1 Front end and back end used

Front end: HTML and CSS

HTML is used as the front end tool to design web pages because:

- It is easy to write, use and understand.
- HTML also allows the use of templates, which makes designing a webpage easy.
- All browsers support HTML.

CSS is used along with html to design the web pages as it is relatively easy to learn and produces better and cleaner code than applying all the styles directly to the HTML code. Also the following reasons make CSS for helpful:

- Easy to maintain and update.
- Greater consistency in design and formatting options.
- Greater accessibility.

Back end: PHP and MySQL

MySQL is a free-to-use, open-source database that facilitates effective management of databases by connecting them to the software. It is a stable, reliable and powerful solution with advanced features like the following:

- MySQL is globally renowned for being the most secured and reliable DBMS used in popular web applications.
- MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance.
- MySQL tops the list of robust transactional database engines available on the market with features like complete atomic, consistent, isolated, transaction support.

PHP (Hypertext Pre-Processor) is a server-side web programming language that is widely used for web development. MySQL is used with PHP as the back end tool in our web application.

- PHP also has powerful output buffering that further increases over the output flow.
- PHP is dynamic. PHP works in combination of HTML to display dynamic elements on the webpage.
- PHP can be used with a large number of relational database management systems and runs on all of the most popular web servers and is available for many different operating systems.

4.2 Discussion of Code Segment

Code to establish connection with the database

```
<?php
  $host = "localhost";     //Server Name
 $user = "root";           //Username
 $pass = "";               //Password
 $db = "project";          //Database Name

 $conn = new mysqli($host, $user, $pass, $db);
 if($conn->connect_error)
{
   echo "Failed to Connect to MySQL" . $conn->connect_error;
 }
?>
```

The above code establishes connection with the database by taking into account the username and password for the MySQL account and also the name of the database it is trying to establish the connection with. '$conn' is the variable used to establish the connection. The function used is 'new mysqli()'.

4.3 Application of Project Work

- The software helps a lot in business management and provides a convenient service in booking the cars for rental.
- Detailed listing and seamless online booking are the main features of the software for easy usage. Customers can directly see the information about all the cars available for them and the prices.
- The car rental software allows only the authenticated users in. With the integration of the secured payment gateways the transaction becomes safe and reliable.
- Data and booking history of the user is stored securely and it can only be accessed by the user and the admin.
- It allows the owner to track the status of the car.
- By using this car rental management system, it is easy for the business owner to track the record of cash payments, generating online invoices and billing options. As a rental manager, you can keep a track of your cash flow.
- This system makes it very easy to manage sales. There is no need to check your available vehicles and allocated vehicles manually. The software shows all the vehicles that are available for booking and the ones that are already rented out. In this way, it is very easy to manage sales and maintain a record of it.

4.4 Discussion of Results

Fig 4.1 Login Page

The above figure shows the snapshot of the Login Page which contains both User and Admin Login forms along with the option to Sign up for a new account. After pressing the login button, the page authenticates the user based on his login name and password and prints “Login Successful” if the entered details match with the details in database.

Fig 4.2 Message displayed after Successful Login

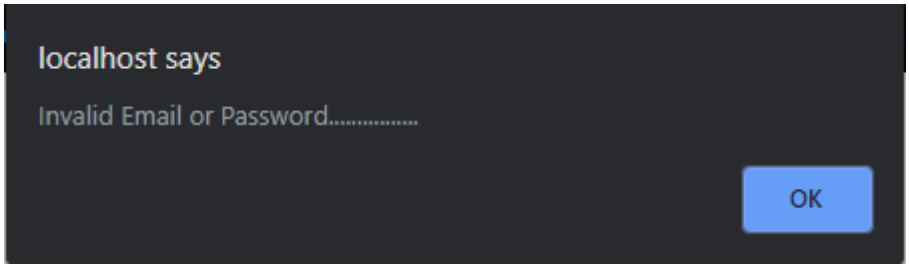


Fig 4.3 Message displayed if details are invalid

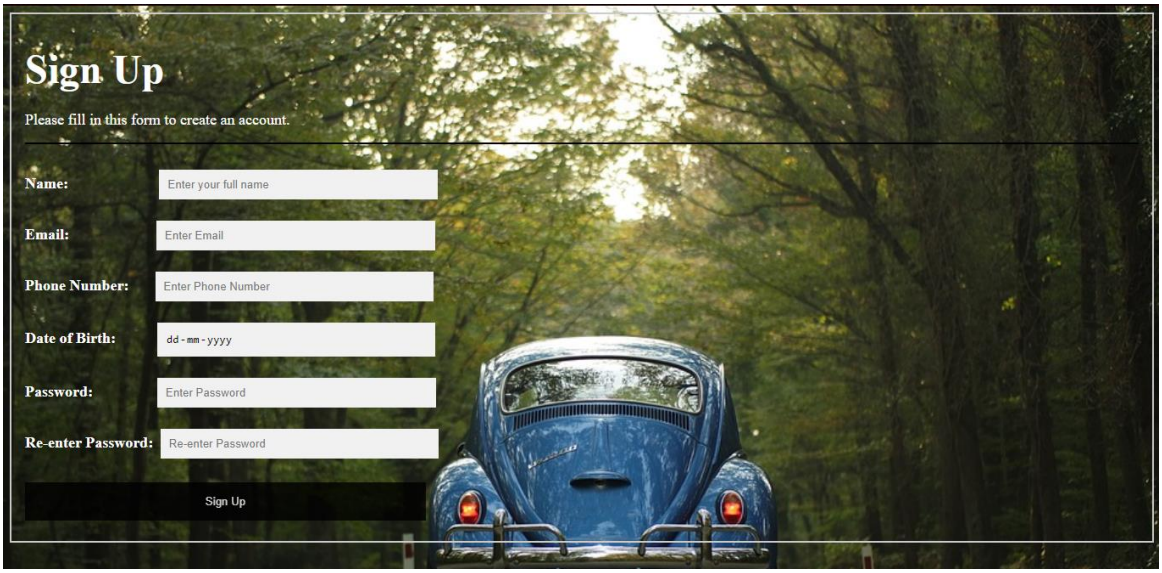


Fig 4.4 Signup Page

The above figure shows the snapshot of the Signup Page. After pressing the signup button the page checks if the email is unique to all the already registered users and prints "Signup Successful" if the entered email is unique.

Fig 4.5 Message displayed after Successful Signup

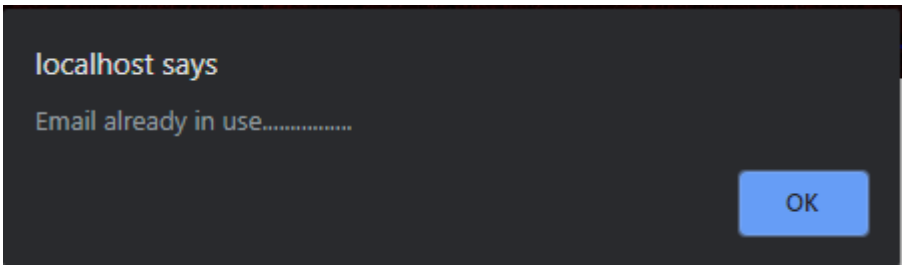
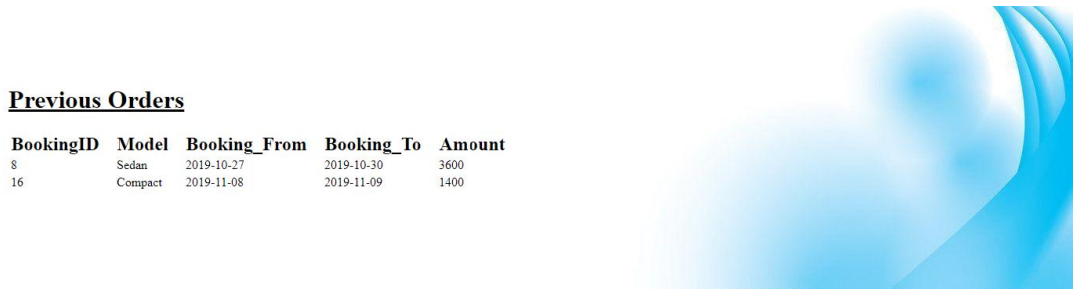


Fig 4.6 Message displayed after Unsuccessful Signup

Fig 4.7 Customer Home Page

The above figure shows the snapshot of the Customer Home Page after the user has successfully logged in. Options to book a car on rent, viewing previous orders made by the user, cancelling a booking and option to give a feedback are provided in this page.

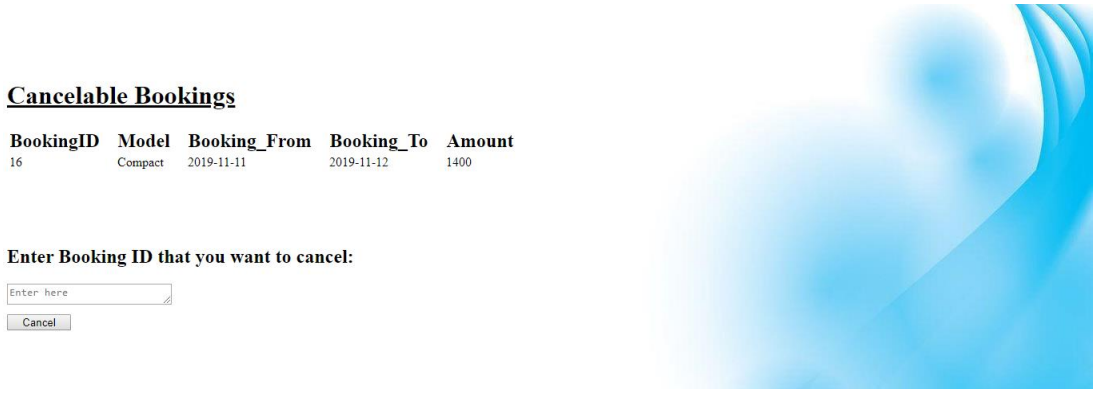


Previous Orders

BookingID	Model	Booking_From	Booking_To	Amount
8	Sedan	2019-10-27	2019-10-30	3600
16	Compact	2019-11-08	2019-11-09	1400

Fig 4.8 Previous Bookings done by a user

The above figure shows the snapshot of the page which displays all the previous and active bookings made by the user.



Cancelable Bookings

BookingID	Model	Booking_From	Booking_To	Amount
16	Compact	2019-11-11	2019-11-12	1400

Enter Booking ID that you want to cancel:

Enter here

Cancel

Fig 4.9 Page to cancel bookings

The above figure shows the snapshot of the page which gives the user option to cancel a particular booking, provided the booking date is still to arrive.

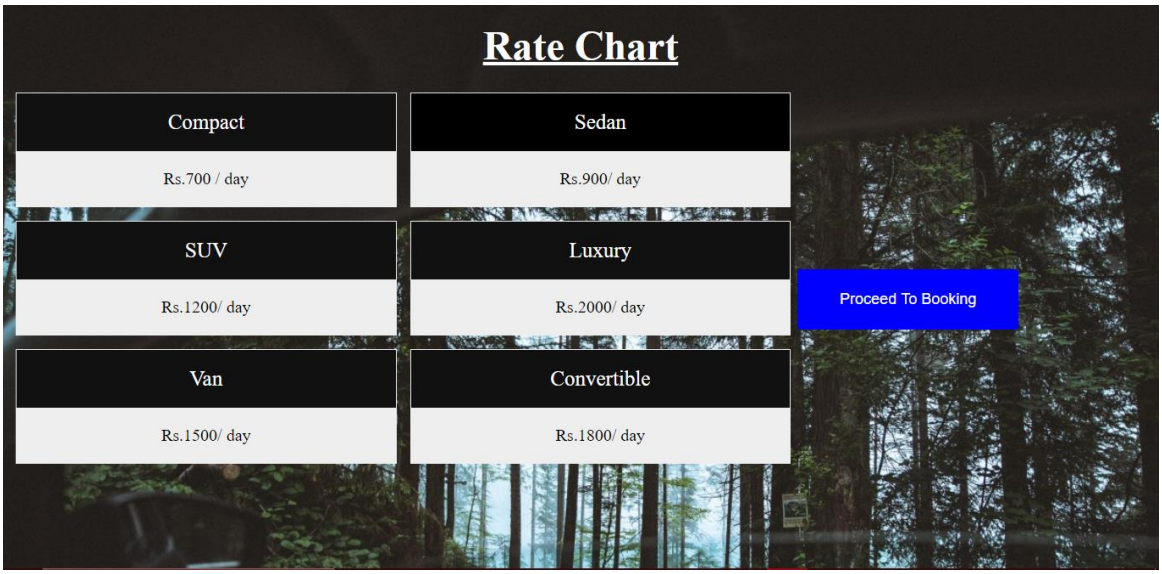


Fig 4.10 Rate Chart of different Models

The above figure shows the snapshot of the page which displays the rate chart of all the available models in the inventory.

Fig 4.11 Booking Page

The above figure shows the snapshot of the Booking Page which checks if on the days specified, the particular model is available for booking or not.

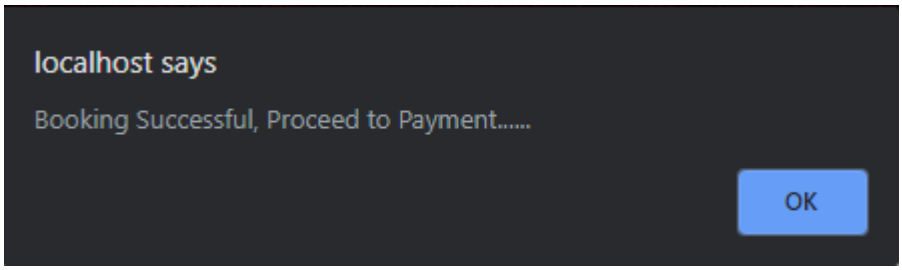


Fig 4.12 Message displayed after Successful Booking

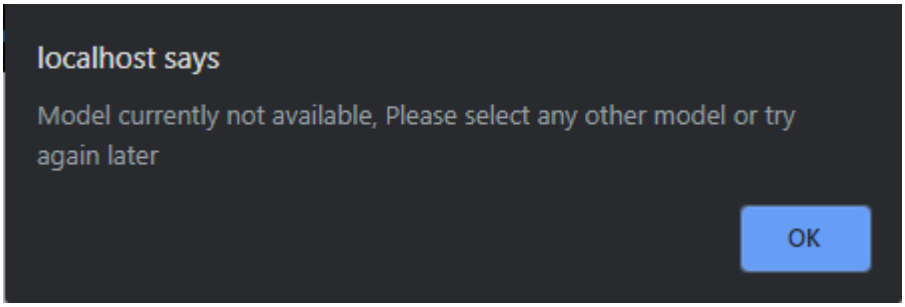


Fig 4.13 Message displayed if selected Model is not available on specified dates

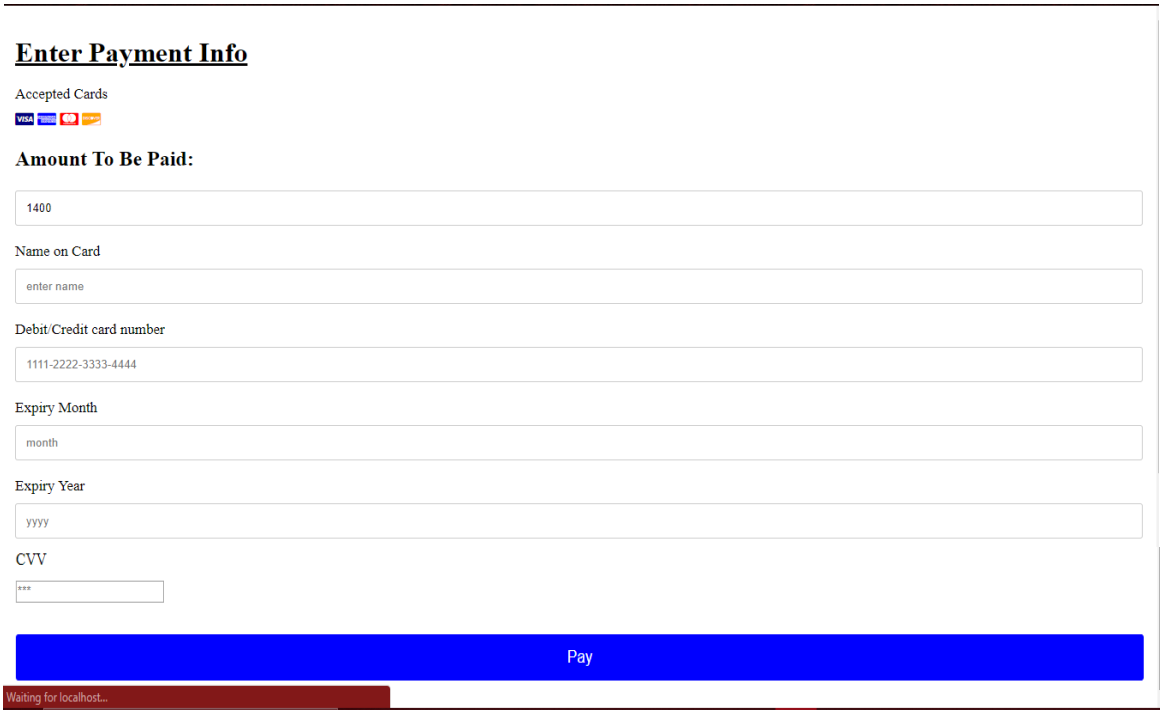


Fig 4.14 Payment Page after Booking is Successful

The above figure shows the snapshot of the Payment Page which displays the total amount to be paid and asks the payment details from the user.

Fig 4.15 Admin Home Page

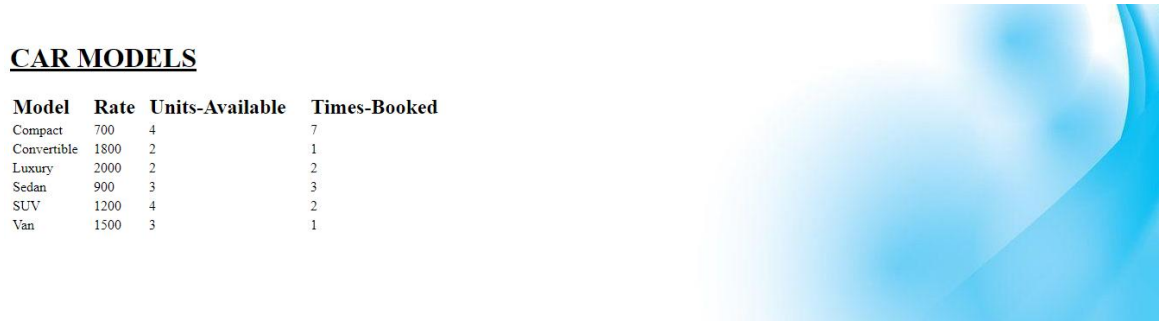
The above figure shows the snapshot of the Admin Home Page after an Admin has successfully logged in. Options to view and manage Feedbacks given by users, Active Bookings, all previous Bookings and Car Details are provided in this page.

Feedbacks

SerialNo.	Username	Rating	Comment
1	nikhilsingh4@gmail.com	5	Very Nice
2	dhanush7865@gmail.com	3	Average

Fig 4.16 Viewing Feedbacks given by users

The above figure shows the snapshot of the page which displays all the feedbacks provided by the users.

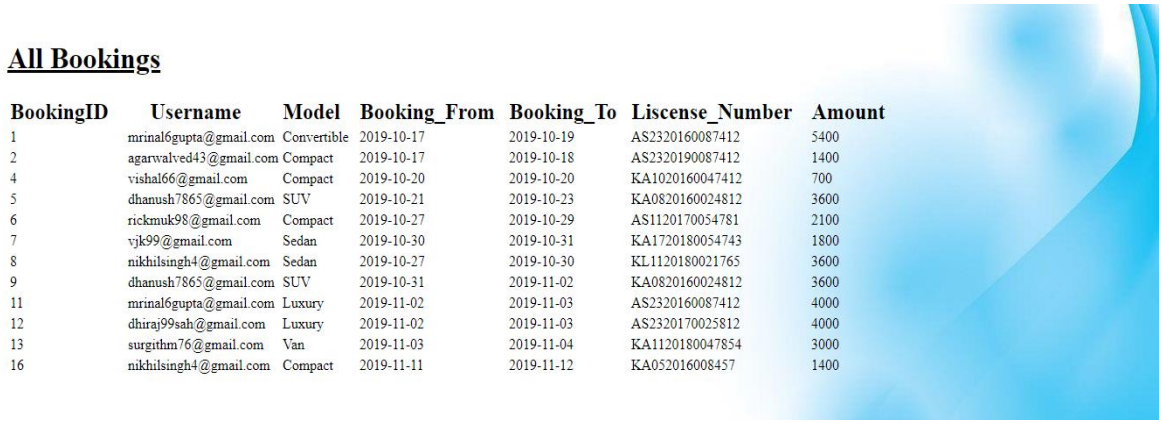


CAR MODELS

Model	Rate	Units-Available	Times-Booked
Compact	700	4	7
Convertible	1800	2	1
Luxury	2000	2	2
Sedan	900	3	3
SUV	1200	4	2
Van	1500	3	1

Fig 4.17 Viewing Details of each Model

The above figure shows the snapshot of the page which displays the details of all the car models available in the inventory along with the number of times each model has been booked.

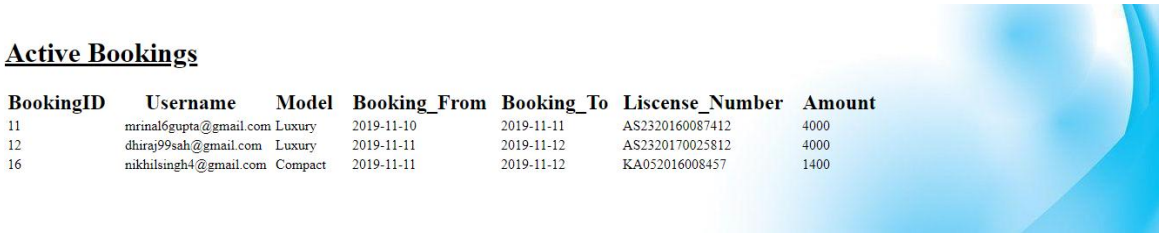


All Bookings

BookingID	Username	Model	Booking_From	Booking_To	Liscense_Number	Amount
1	mrinal6gupta@gmail.com	Convertible	2019-10-17	2019-10-19	AS2320160087412	5400
2	agarwalved43@gmail.com	Compact	2019-10-17	2019-10-18	AS2320190087412	1400
4	vishal66@gmail.com	Compact	2019-10-20	2019-10-20	KA1020160047412	700
5	dhanush7865@gmail.com	SUV	2019-10-21	2019-10-23	KA0820160024812	3600
6	rickmuk98@gmail.com	Compact	2019-10-27	2019-10-29	AS1120170054781	2100
7	vjk99@gmail.com	Sedan	2019-10-30	2019-10-31	KA1720180054743	1800
8	nikhilsingh4@gmail.com	Sedan	2019-10-27	2019-10-30	KL1120180021765	3600
9	dhanush7865@gmail.com	SUV	2019-10-31	2019-11-02	KA0820160024812	3600
11	mrinal6gupta@gmail.com	Luxury	2019-11-02	2019-11-03	AS2320160087412	4000
12	dhiraj99sah@gmail.com	Luxury	2019-11-02	2019-11-03	AS2320170025812	4000
13	surgithm76@gmail.com	Van	2019-11-03	2019-11-04	KA1120180047854	3000
16	nikhilsingh4@gmail.com	Compact	2019-11-11	2019-11-12	KA052016008457	1400

Fig 4.18 Viewing all previous Bookings

The above figure shows the snapshot of the page which displays all the previous and active bookings made by all the users.



Active Bookings

BookingID	Username	Model	Booking_From	Booking_To	Liscense_Number	Amount
11	mrinal6gupta@gmail.com	Luxury	2019-11-10	2019-11-11	AS2320160087412	4000
12	dhiraj99sah@gmail.com	Luxury	2019-11-11	2019-11-12	AS2320170025812	4000
16	nikhilsingh4@gmail.com	Compact	2019-11-11	2019-11-12	KA052016008457	1400

Fig 4.19 Viewing Bookings those are currently active

The above figure shows the snapshot of the page which displays all the active bookings made by all the users.

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusions

The primary goal of this project is to allow the user to rent a self-drive car online without having to manually go through the process of it. It makes it easy not only for the user but also to the business provider to maintain a catalogue of cars available for booking and maintain a record of all the bookings that have been made. Now both the car rental business and website can run smoothly, by displaying available cars, accepting online reservations and managing the entire fleet of cars, all from one single control panel. By offering a highly customizable booking system, customers can see exact vehicle availability and their past and active bookings as well.

The following are the future enhancements that can be made on the project:

- To verify the driving license of the user.
- Develop an iOS and android application for this website.
- Allow renting a car on hourly basis.
- Provide chauffer services with the car.
- To allow payment through various other payment gateways like Paytm, Gpay, PhonePe, etc.
- To provide car rental services from places like airports, railway stations, etc.

REFERENCES

TEXT BOOK

1. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGRAW HILL, 3rd Edition.
2. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.

WEBSITES REFERRED

[1] https://www.geeksforgeeks.org/php-mysql-select-query/

[2] https://stackoverflow.com/questions/8722806/how-to-compare-two-dates-in-php

[3] https://www.w3schools.com/sql/func_sqlserver_getdate.asp