

**DESIGN AND DEVELOPMENT OF WEB APPLICATION ON RESORT
BOOKING: VROMON BILASH**



**DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH SEPTEMBER 2019**

**DESIGN AND DEVELOPMENT OF WEB APPLICATION ON RESORT
BOOKING: VROMON BILASH**

BY

Redwanul Islam

ID: 141-15-094

AND

Tanbir Ahmed Tushar

ID: 143-15-241

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Ms. Shayla Sharmin

Lecturer

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH SEPTEMBER 2019

APPROVAL

This Project titled “**DESIGN AND DEVELOPMENT OF WEB APPLICATION ON RESORT BOOKING: VROMON BILASH**”, submitted by Redwanul Islam, ID:- 141-15-094 and Tanbir Ahmed Tushar, ID:- 143-15-241 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering (BSc) and approved as to its style and contents. The presentation has been held on 16 September 2019.

BOARD OF EXAMINERS

Dr. Syed Akhter Hossain
Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman

Dr. S M Aminul Haque
Associate Professor & Associate Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Saif Mahmud Parvez
Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Dr. Mohammad Shorif Uddin
Professor

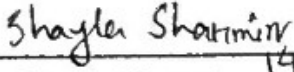
Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

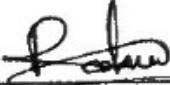
DECLARATION

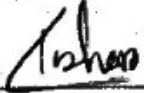
We hereby declare that, this project has been done by us under the supervision of **Ms. Shayla Sharmin, Lecturer, Department of CSE Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:


14.9.2019
Ms. Shayla Sharmin
Lecturer
Department of CSE
Daffodil International University

Submitted by:


Redwanul Islam
ID: 141-15-094
Department of CSE
Daffodil International University


Tanbir Ahmed Tushar
ID: 143-15-241
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound our indebtedness to **Ms. Shayla Sharmin, Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “Information Technology” to carry out this project. His endless patience ,scholarly guidance ,continual encouragement to constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to our Supervisor and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

In modern world we always need modern technology. Now all technology and task are getting computerized day by day, for this web application increasing day by day. Everyone wants more easier technology and everything wants in their hands. By the use of the computer and internet everything is available and everybody can do lot of things from home. Base on this reason our full project is built. This Resort Booking System is a web based application. By using this system anyone can browse resort in a area and see their price based on category and place. They can also book resort by paying the minimum percentage of amount.

Table of Content

CONTENTS	PAGE
Title Page	i
Approval	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcome	2
1.5 Report Layout	2
1.6 Summary	3
CHAPTER 2: BACKGROUND	4-7
2.1 Introduction	4
2.2 Related Work	4
2.3 Figures of related work	5
2.4 Challenges	6
2.5 Summary	7
CHAPTER 3: REQUIREMENT AND SPECIFICATION	8-11
3.1 UML Diagram	8
3.1.1 Use case diagram	8
3.1.2 Description Use case diagram	9
3.1.3 System flow diagram	10
3.2 Design Requirements	11
©Daffodil International University	vi

CHAPTER 4: SYSTEM DETAILS	12-27
4.1 Introduction	12
4.2 Application Feature	12
4.2.1 Registration	13
4.2.2 Login	13
4.2.3 Searching	13
4.2.4 Choosing room	13
4.2.5 Booking	13
4.2.6 Online payment	13
4.2.7 Review place	14
4.2.8 Make profile	14
4.2.9 Add resort	14
4.2.10 Manage room category	14
4.2.11 Manage time configuration	14
4.2.12 Manage room plane	14
4.2.13 Approve booking request	14
4.2.14 Manage reviews	14
4.2.15 Manage places	15
4.2.16 Manage resort	15
4.3 Front-End Design	16-25
4.4 Back-End Design	26
4.5 Implementation Requirements	27
4.6 Hardware Requirement	27
4.7 Software Requirement	27
4.8 Summary	27

CHAPTER 5

IMPLEMENTATION AND TESTING 28-37

5.1 Introduction	28
5.2 Implementation of Database and query	28-31
5.3 Implementation of booking system and time configuration	32-34

LIST OF FIGURES 5-34

Figure:2.3.1 Reference site 1	5
Figure:2.3.2 Reference site 2	6
Figure:3.1 Use case Diagram	8
Figure:3.2 System Flow Diagram	10
Figure:4.3.1 Home page	16
Figure:4.3.2 User Registration	17
Figure:4.3.3 Login Page	17
Figure:4.3.4 Searching System	18
Figure:4.3.5 Resort List	18
Figure:4.3.6 Choosing Room	18
Figure:4.3.7 Resort Info	19
Figure:4.3.8 View Reviews	19
Figure:4.3.9 Order Cart	20
Figure:4.3.10. Booking System	20
Figure:4.3.11 Order summery	20
Figure:4.3.12 Payment Method	21
Figure:4.3.13 User Profile	21
Figure:4.3.14 Admin Dashboard	21

Figure:4.3.15 Banners	22
Figure:4.3.16 Add Rooms	22
Figure:4.3.17 Manage Room Category	23
Figure:4.3.18 Booking Request	23
Figure:4.3.19 Review Management	24
Figure:4.3.20 Manage Resort	24
Figure:4.3.21 Manage Area	24
Figure:4.3.22 Manage Place	25
Figure:4.4.1 Database Relationship	26
Figure:4.4.2 Database Table	26
Figure:5.1-5.8 Implementation of Code and Testing	28-34
5.4 Testing Implementation	35
5.4.1 Unit Testing	35
5.4.2 Integration Testing	35
5.4.3 Functional Testing	35
5.4.4 System Testing	35
5.4.5 Usability Testing	36
5.5 Test Result	36
5.5.1 Result of Unit Testing	36
5.5.2 Result of Integration Testing	36
5.5.3 Result of Functional Testing	36
5.5.4 Result of System Testing	37
5.6 Summary	37

CHAPTER 6	
CONCLUSION AND FUTURE SCOPE	38
6.1 Discussion and Conclusion	38
6.2 Scope and Further Development	38
6.3 Limitation	38
REFERENCES	39
PLAGIARISM REPORT	40

CHAPTER 1

INTRODUCTION

1.1 Introduction

Today we are living in a vast expanse of advancement. We are using development in our step by step needs. As needs be, as a bit of advancement web has ended up being most basic thing. Cause now-a-days every single sort of gadgets is associated with cloud. In like manner, a critical number of our step by step errands are ended up being dependent on the web.

Furthermore, today whatever we need at first, we seek it on the google utilizing web. At that point we gather the essential material or data what we needed. Primary concern is we would prefer not to squander our profitable time and consequently we do it.

Now-a-days many people want to visit many resorts and wants to book that by sitting at home. But there is a common problem for booking resort that is they cannot find the appropriate place. They do not have the proper information and prices of room the quality and the environment and what are necessary for booking a resort.

On the basis of it we have taken a step to solve these kinds of problem by making a web-based Resort booking system for all travelers.

In this system we attempted to tackle a wide range of issue what are looked at the season of booking resort. It will assist the general population with finding the resort and costs of room and all data of resort as indicated by their decision, as per their spending limits and it will likewise help them by giving all fundamental data about their decision.

1.2 Motivation

We have searched all over the internet for finding out a better web application for Resort Booking System. But we have found there is nearly few applications that can help a traveler for booking a resort. Then we discussed it with our honorable Mam Ms.Shayla Sharmin(Lecturer, DIU-UC). After Discussion she told us to do a project on resort booking system where user will get the easiest way to find resort and according to their category and prices. Then we discussed it with our elder brother Mehedi Hasan (Software Engineer). After listening the full idea, he also appreciates us to do this project and gave direction how we can start.

1.3 Objectives

- We have made our web application about resort booking system. By then

there are a couple of objectives that we need to fulfill. Here are a segment of the goals of goals that can be gotten by our project.

- Make this application user friendly.
- Finding necessary resource.
- Set booking management.
- Set payment gateway and booking notification.
- Passing control to the users to maintain their profile.
- Test this application for bugs and other errors.
- Do necessary coding and build up an Application.

1.4 Expected Outcome

This booking system will express some outcome. Here we will discuss about it:

- Making a full furnish web application.
- Easy to use our system.
- User can book resort choosing specific room category.
- User can pay minimum booking amount.
- Get user review for specific resort.
- All user can maintain their profile.
- Various kind of resort will show.

1.5 Report Layout

Here is the report format that used to make this report on web application of resort booking system.

- All the topics are covered with related information.
- All topics are divided with some paragraph thus it can be easy to understand.
- Figures and images are used to get the topics content more easily.
- Required font and size is used to organize all the contents of the report.
- Specific margins and spaces are used to format the content.
- Bullet point are used to describe the content value more exact.
- Required format and menus are used to get all the contents more easily.

1.6 Summary

In this cutting-edge age how might we maintain a strategic distance from innovation. Since innovation is quicker than everything. Along these lines, subsequent to arriving this advanced age we don't have to rely upon anybody for anything. Booking a resort is very important and everybody likes travel various resort. By using our system people can easily find resort and make online booking. For this reason, this system is very much helpful. This system will take all responsibility to give the best idea of resort and the environment and easy to find better place for booking. This is the main reason for creating our system.

CHAPTER 2

BACKGROUND

2.1 Introduction

The fundamental objective of making this resort booking system is that it makes the existence simpler and agreeable for voyager. Along these lines, at that point we are attempting to look through something that has not moved into this sort of system. At that point we discovered some web application about Resort the board system. In any case, they have insufficient data for voyager So we chose to make booking system that will be all the more inviting and compelling to the clients.

2.2 Related Work

There is some related system like our system. We have tried to do something better and unique in our system. Some systems are more attractive but our system has more functionality. We are trying to Compare those systems to our system is primarily not too much attractive but it is user friendly and unique.

Here is some example of resort management system and those websites features and drawbacks.

Website Link:

- <https://www.khujbo.com>.
- <https://www.dhakaholidays.com>

2.3 Figures of related work

Figure 2.3.1 shows a resort management system.



Figure 2.3.1: Reference site 1

Here, figure 2.3.2 shows a resort management system.

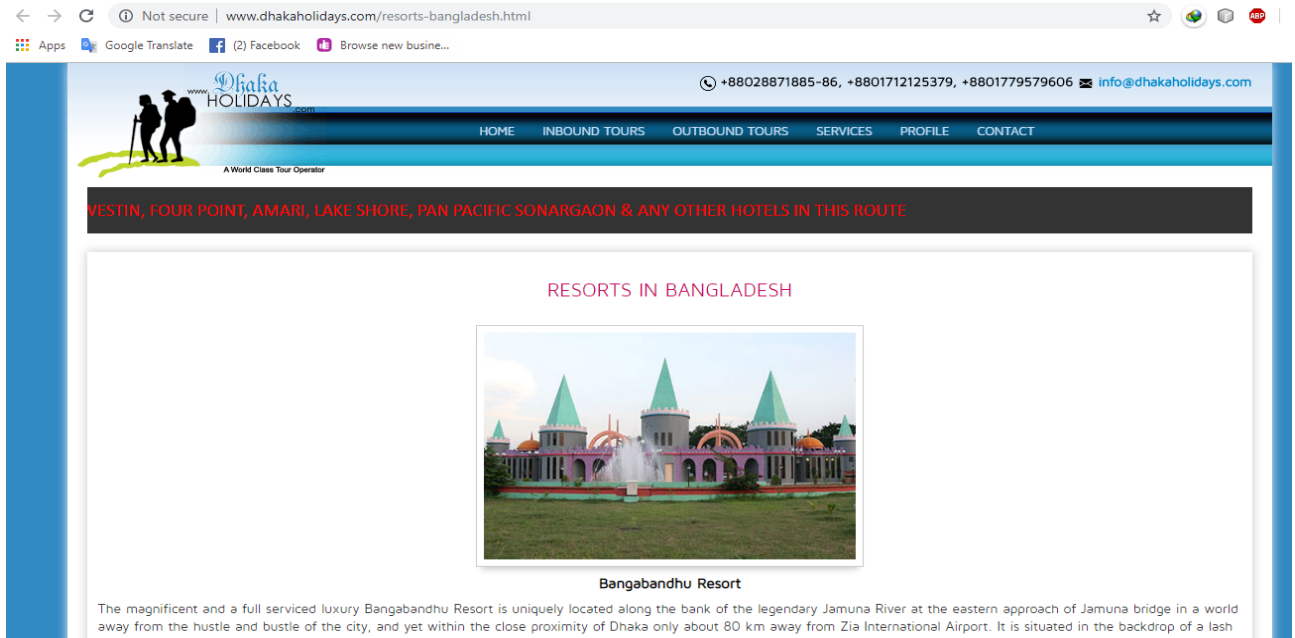


Figure 2.3.2: Reference site 2

2.4 Challenges

In this resort booking system we have confronted such a significant number of difficulties. A few difficulties are too hard to even think about meeting objective and some are absolutely new to us to discover the arrangement like,

- **Setting booking configuration:** It was not easy to manage booking system. If user do not get the proper knowledge about booking management of resort, they can't book resort.
- **Create the relation to all the tables in database:** To make database tables are too much difficult job to do, we tried our best to normalized the database table for our project.
- A standout amongst the most essential test was gathering proper information for this system and give an easy to understand look the system and make this system dependable.

2.5 Summary

Before making a system or develop a system it is imperative to separate the related work of that system. Without extraordinary and suitable looking at no work can't get its climax and can't similarly make the goal. Along these lines, after extensive time span separating, we have made it and we fixed our target which will empty the lacking of existing system.

CHAPTER 3

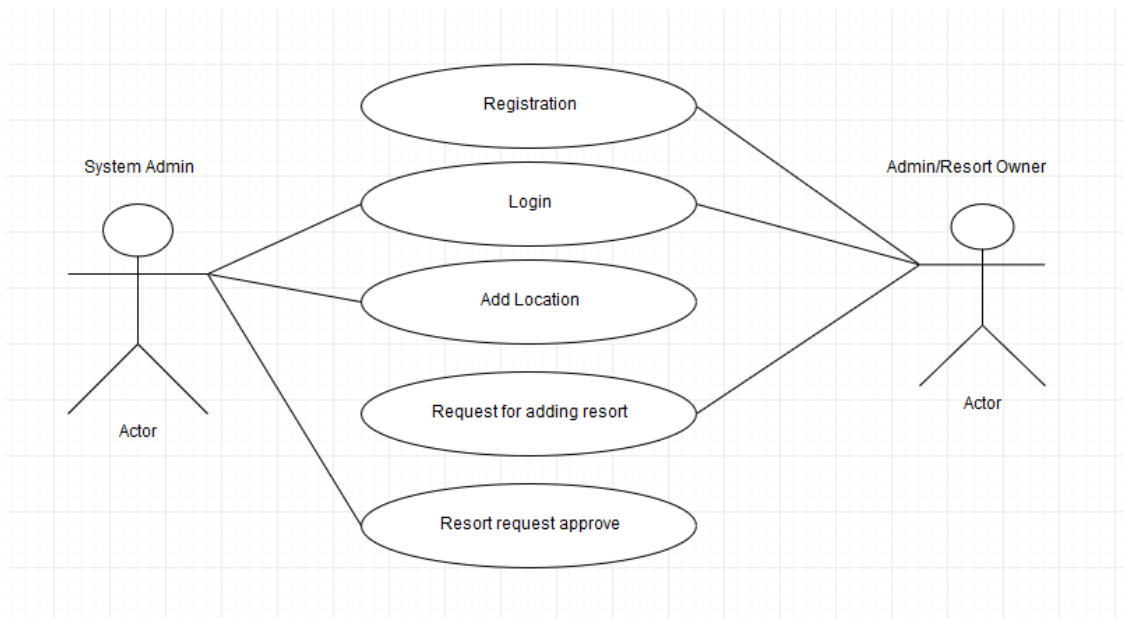
REQUIREMENT AND SPECIFICATION

3.1 UML Diagram

In this section we will examine about UML chart. There are numerous sorts of UML outline. We will talk about two of them. By utilizing what we will talk about our whole system.

3.1.1 Use Case Diagram

Use case diagram represent all the actor whose are related to the system and then show them how they can participate to the system. Figure-3.1 is the use case diagram of our system.



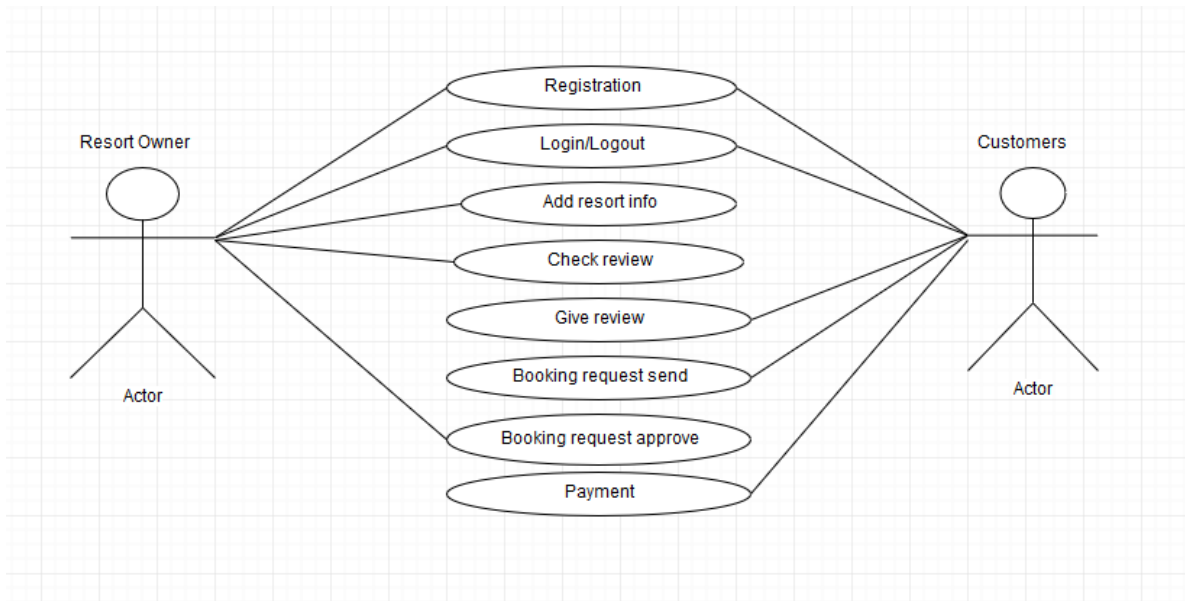


Figure 3.1: use case diagram of resort booking system

3.1.2 Description of Use Case Model

In Use case diagram we can see there is three actors and they are (1) System Admin (2) Admin and (3) User. In this application System admin can log in, manage places and areas and approve resort approval request. Admin can manage room categories and food menus. He also can He also can upload photos of his resort and set the location. He can create room plane. User can browse resort by location or resort name .user can view resort list and its facilities of room and the pricing of room. The price and facilities of room is set by the admin. He can add room on his cart and choose time for booking and send a booking request, he has to pay minimum amount of food price. Admin receive the booking request and send a notification to the user

Use case of Log in

Cases	Details
Actors	System admin, Admin and User
Pre-Condition	Registration is needed.
Post-Condition	After Successful log in System admin will get System admin panel, Admin will get admin panel and user will get user profile.

3.1.3 System Flow Diagram

Now we are going to show our working flow diagram in figure-3.2. The entire system has three fundamental on-screen characters System administrator, Admin and another is client. Enlistment is must be required for two entertainers Admin and User. To start with, we will discuss client. Client must to sign in into the system, in the event that he login he could book a resort else he couldn't book. On the off chance that client is old enrolled, at that point at whatever point he signed in, system will take the client to client landing page. In client landing page client will get a few highlights like, User can peruse resorts by area or resort name .client can view resort rundown and room classifications and the evaluating of rooms. The cost and limits of room is set by the administrator. He can include rooms his truck and pick time for booking and send a booking demand, he needs to pay least measure of cost. He additionally can survey the retreat. Next is admin, he can manage room categories and capacity. He also can manage opening and closing time of his resort. He also can upload photos of his resort and set the location. Admin can receive the booking request and send a notification to the user. System admin can log in, manage places and areas and approve resort approval request.

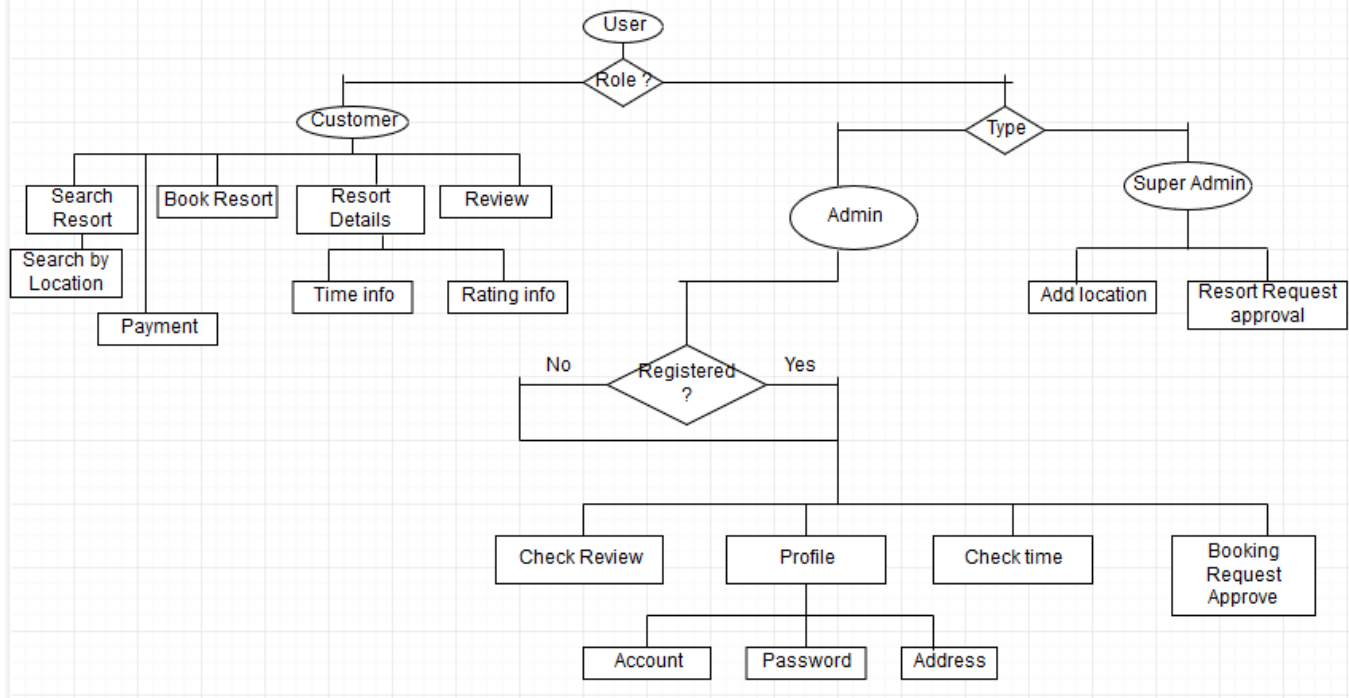


Figure 3.2: System Flow Diagram

3.2 Design Requirements

To develop this venture, we need to complete a couple of structures to make this system all the more straightforward and progressively reliable to customer. Along these lines, this has two sorts of structure and they are front end and back end.

To do the frontend plan we should have been sketching the designs into paper first. At that point when the coding starts of the frontend, at that point it should be incorporate bootstrap, bootstrap js and some jquery code for complete the plan. At that point we have to structure the undertaking as the documentation prerequisites and get done with coding.

By and by for the back end plan we need to consider the database structure and the data on social database. There is in like manner need to get the association between the tables in the database and moreover need to mind redundancy things. With the objective that we need to do some institutionalization to keep up a key good ways from this kind of things. Also, for back end coding we were attempting to seek after the thing organized coding structure. By then find the record structure for the code storage facility. Henceforth, we get our beginning and end the essential on back end structure.

So that's all the design requirement that needed in this project.

3.3 Summary

In this chapter we discussed about the full project using use case diagram and system flow chart. We also described them elaborately.

CHAPTER 4

SYSTEM DETAILS

4.1 Introduction

In this part we will talk about quickly about our system. After totally reading this section anyone who has read this part will get the 70% thought regarding our application. In this part we will talk about our system highlights, front-end configuration, back-end database and execution prerequisite.

4.2 Application Feature

There are various kind of features in our system. At first, we are going to show our features list and we will describe them later. So, the features of this systems are-

- Registration – This is for user and admin to access this system.
- Login into system – Without logging in admin and system admin can't use this system.
- Searching- User can search resort by location.
- Choosing room - User can choose rooms by clicking the room category and it will save into the cart.
- Booking-User can book resort rooms by choosing room category. User can also choose booking time and number of people.
- Online payment- User can pay minimum amount of price.
- Review place – User can give a review after experienced.
- Make profile – All type of user can set up his/her profile.
- Add resort-Any user can add resort.
- Manage room category – Admin can manage room categories and upload pictures.
- Manage time config-Admin can set time configuration.
- Approve booking request-Admin can approve booking request.
- Room pane- Admin can add capacity of his/her resort room.
- Manage reviews- Admin can manage reviews.
- Manage places-System admin can add place and areas
- Manage resort– System admin can approve resort request.

4.2.1 Registration

From the outset client needs to enrollment to enter the system. Client needs to give a legitimate email and secret word to make himself enlisted. On the off chance that any client acknowledge system administrator needs to access on the system with no enlistment then it is essentially unthinkable. Each client need to get enrolled first.

4.2.2 Log in

After complete the enrollment client can go into the system utilizing their email and secret key that was given at the season of enlistment. On the off chance that client put wrong email or secret phrase, client will neglect to go into the system. In this way, it is important to recollect email and secret word. On the off chance that clients need, they can change is secret key whenever by clicking his/her profile yet can't change the email.

4.2.3 Searching

Using this feature user can search resort by location and also can see the all the resorts list their name and how much rating it have.

4.2.4 Choosing room

By this component client can pick rooms by tapping the room classification and it will spare into the truck. When he clicks any room, it will consequently be spared into the truck and figure the genuine evaluating. By this component he could see the aggregate sum and his chose thing.

4.2.5 Booking

After select the room user need to fill a form for booking, user need to write his name address, phone number, number of people that he wants to book and also can choose time for booking. After filled this information user needs to click submit button for confirm the booking.

4.2.6 Online payment

After confirm the booking form user need to pay at least 10% amount of his pricing that has been automatically calculated and shows to the user. User can choose many payment systems by using online payment feature.

4.2.7 Review place

After experiencing any resort and its facilities he/she can write reviews and can give ratings to that resort. User needs to log in for place a review, without login he/she will not be able to place review. Without login user only can view reviews.

4.2.8 Make profile

After login a newly registered user can go to his/her Profile. User can upload profile picture, change password and update his/her mobile number and address. User can also see the booking list that he booked before.

4.2.9 Add resort

In this system any user can add resort by fill a form, user need to give resort name and address of his resort and send a request to the system admin for approval.

4.2.10 Manage room category

After send a solicitation for include resort he/she will naturally go to the retreat dashboard and afterward his job will consequently change to administrator, from the dashboard administrator can include room classification, under this room classification administrator can include offices of room, limits and evaluating. Administrator likewise can alter and erase this data.

4.2.11 Manage time configuration

Admin can add opening and closing time for his/her resort by using this feature. Admin also can select opening and closing day for his/her resort.

4.2.12 Manage room plan

Admin can add the t capacity of his room and also can upload photos.

4.2.13 Approve booking request

There is an important feature for admin that he can see the booking request send by the user. Admin can see the booking information and paid money and unpaid money.

4.2.14 Manage reviews

Admin can see the reviews that is given by the user and can delete the review by using this feature.

4.2.15 Manage places

This feature is used by the system admin. He/she can add places and areas that he wants. User and admin can only view city and areas added by the system admin.

4.2.16 Manage resort

By using this feature system admin can approve resort request, when he/she accept resort request it will be live for the user. System admin also can disable any resort.

4.3 Front End Design

This is the home page for all users. After giving the valid email and password user, admin or system admin can enter the system.

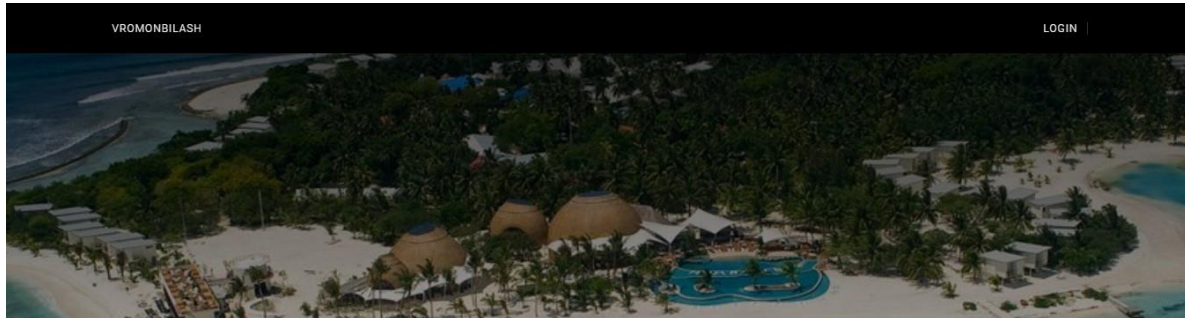
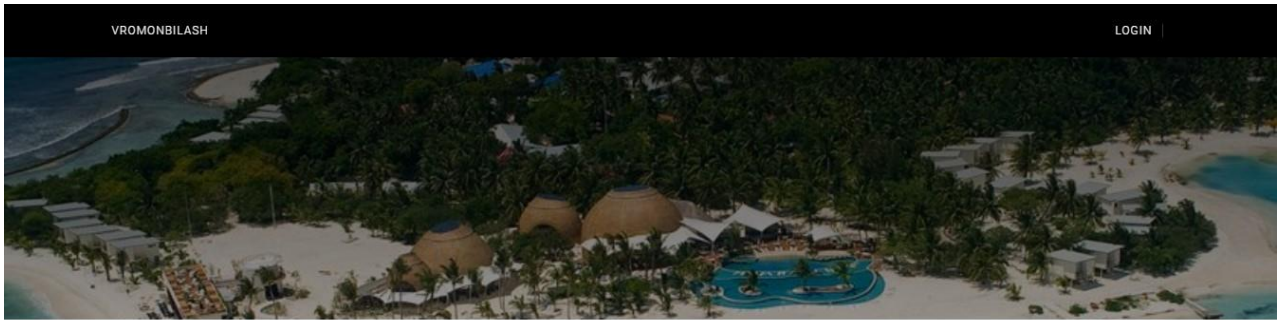


Figure 4.3.1 Home page and login system.

From here, user can registration



Create A New Account

Full Name

Email

Password

Confrim Password

REGISTER

Already have an account ? [Sign In](#)

Figure 4.3.2: User Registration

Account Login

Sign In to your account

Email

Password

LOGIN

[Forgot password ? Create a new account](#)

Figure 4.3.3: Login page

Select Your Location

Select your City Select your Area

Figure 4.3.4: Searching system

The screenshot displays two resort listings side-by-side. The first listing is for 'HERITAGE ECO RESORT' with a rating of 4. The second listing is for 'SOHAG POLLI' with a rating of 0. Both listings feature a description: 'Featuring an outdoor swimming pool, a fitness centre and a sauna, Royal Park Residence Hotel offers accommodation in Dhaka, 300 m from Embassy of Spain. Free WiFi is available throughout the property.' and a location tag 'JOYDEBPUR,GAZIPUR'.

Figure 4.3.5: Resort list

Heritage Eco Resort

Room Category

Single Room
 Double Bed
 Single Deluxe

Single Room

101
Capacity: 2 Persons
Beds: 1 bed
Amenities: Wifi, AC, 50" led tv, refrigerator
Per Night: 1000 Tk

CheckIn Date

Checkout Date

Your Booking

Double Bed

102
Capacity: 4 Persons
Beds: 2 bed

Figure 4.3.6 Choosing room

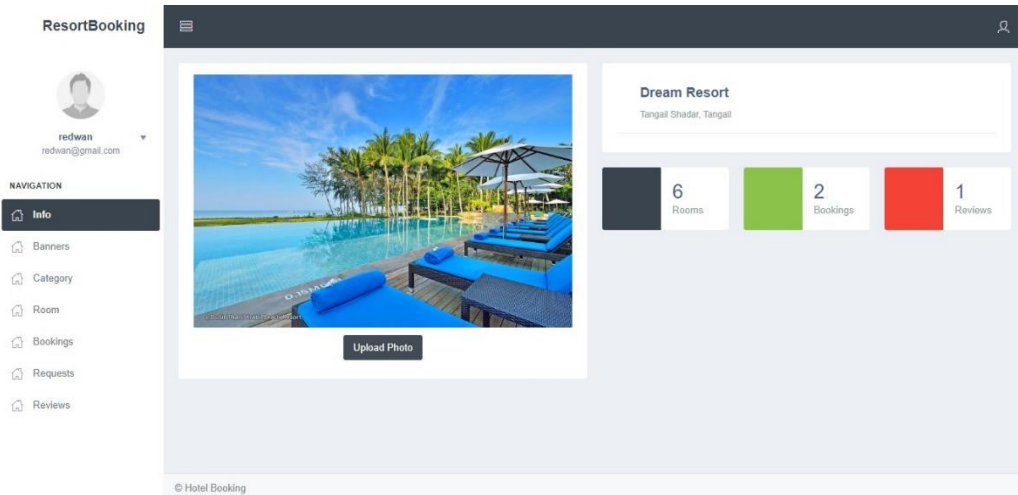
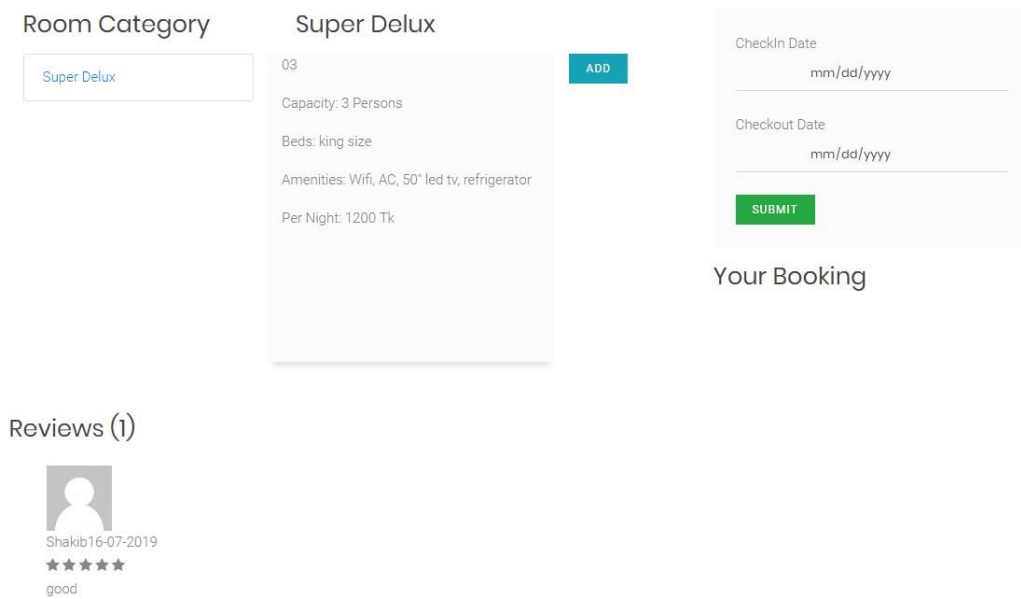


Figure 4.3.7: Resort info



Reviews (1)



Figure 4.3.8: View reviews

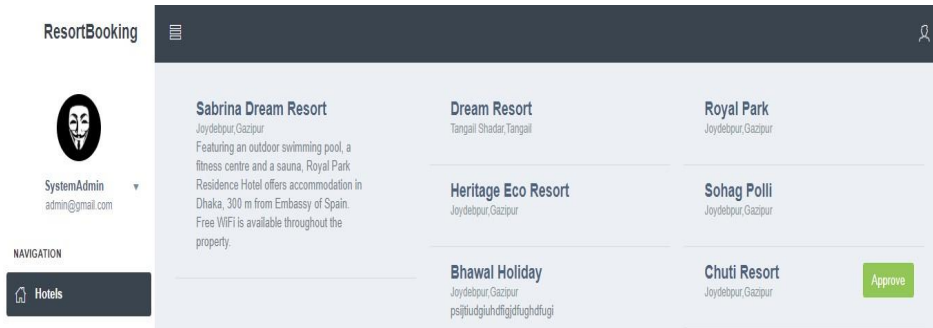


Figure 4.3.9: Order cart

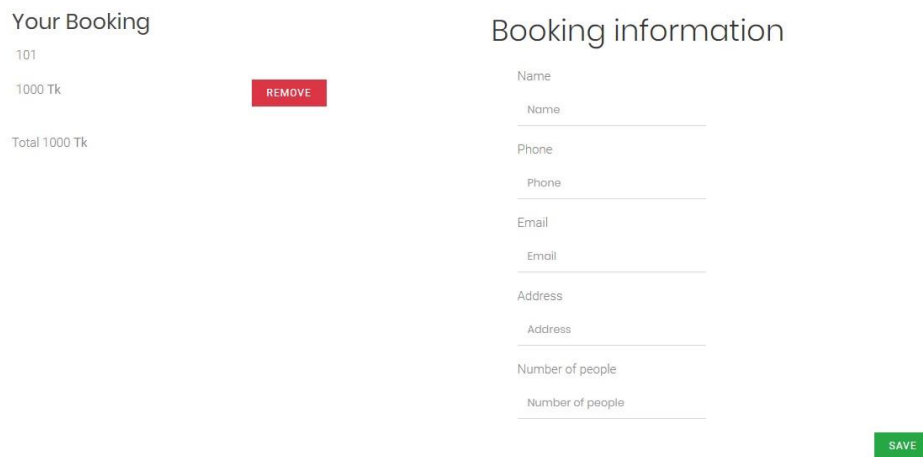


Figure 4.3.10: Booking System



Figure 4.3.11: Order summery



Figure 4.3.12: Payment method

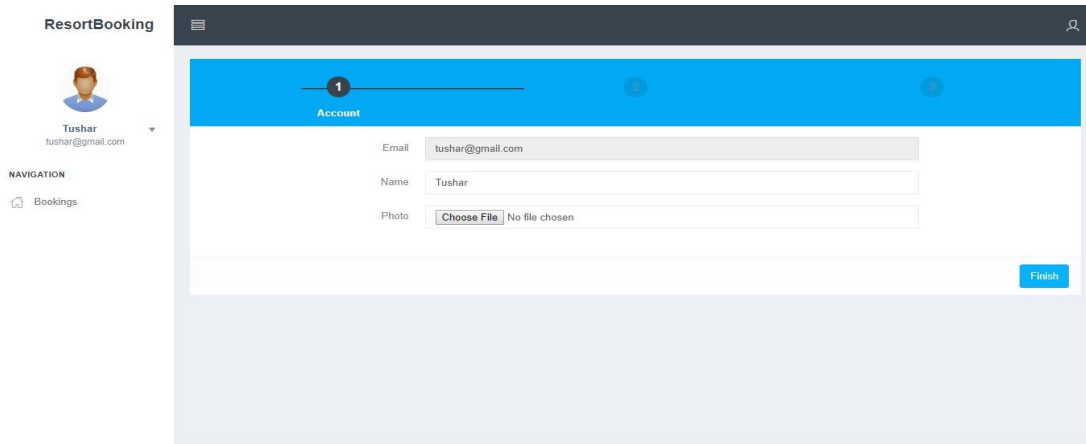


Figure 4.3.13: User profile

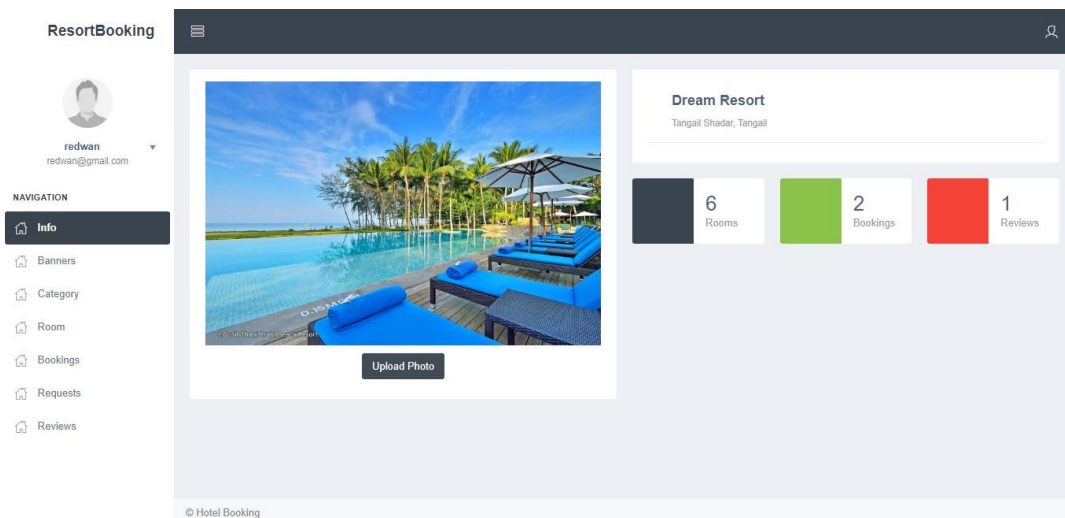


Figure 4.3.14: Admin Dashboard

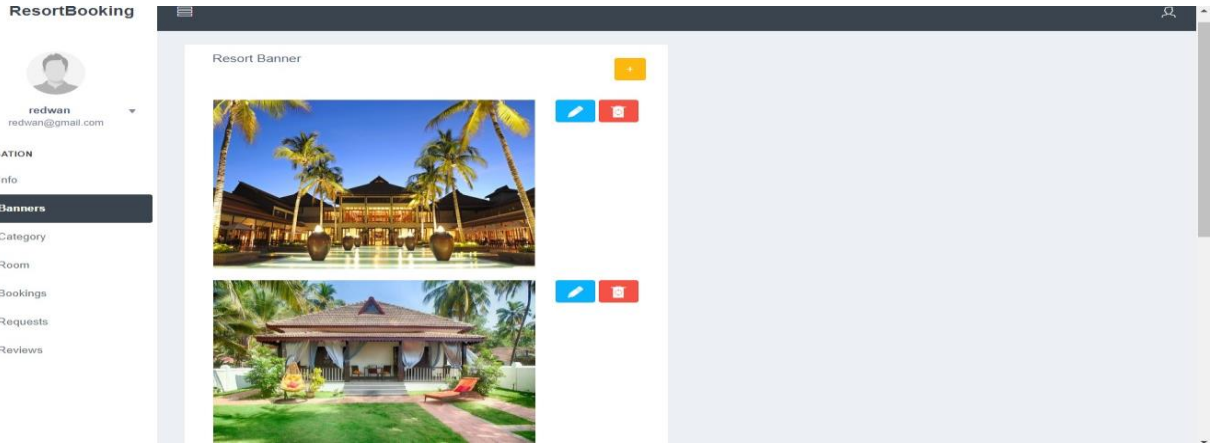


Figure 4.3.15: Banners

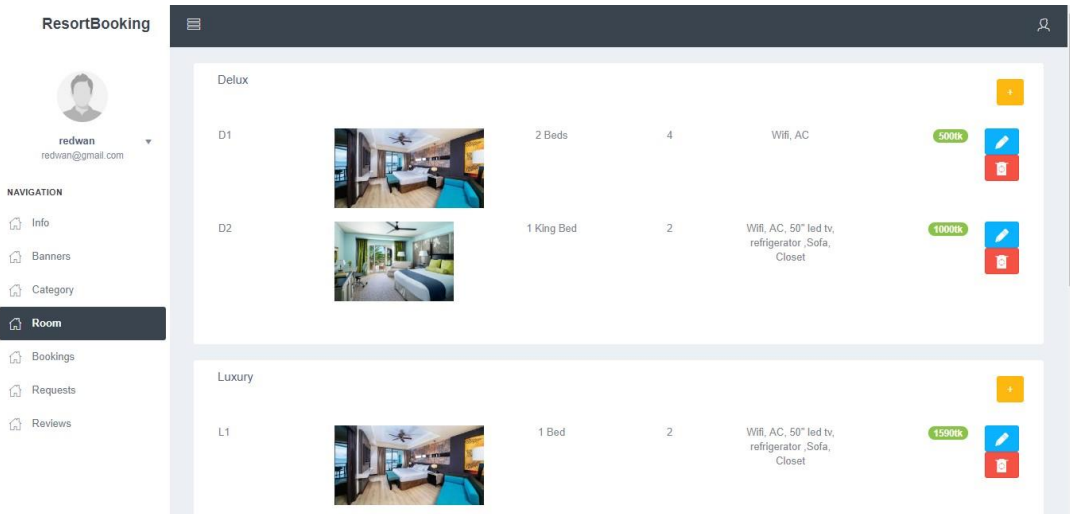


Figure 4.3.16: Add rooms

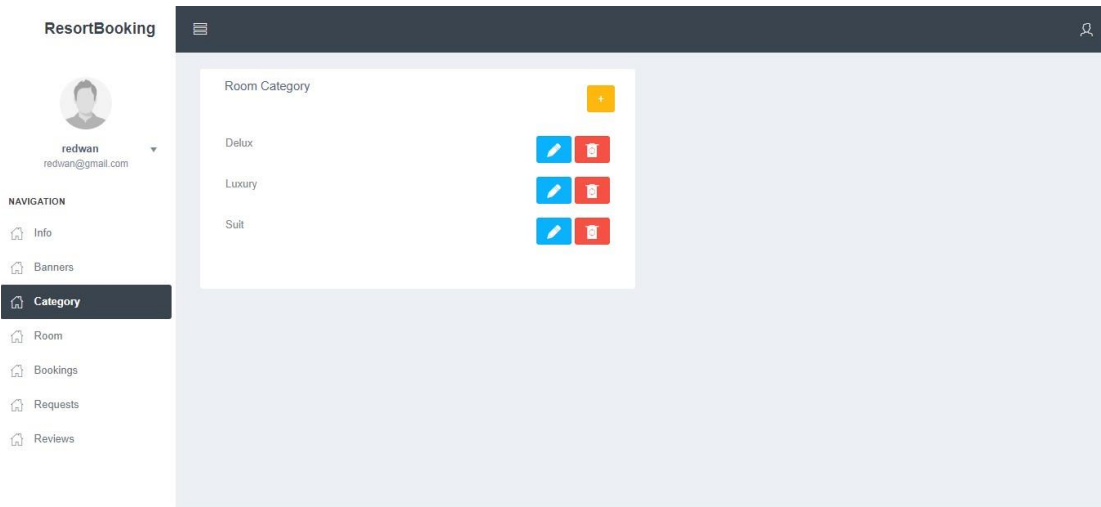


Figure 4.3.17: Manage room category



Figure 4.3.18: Booking request

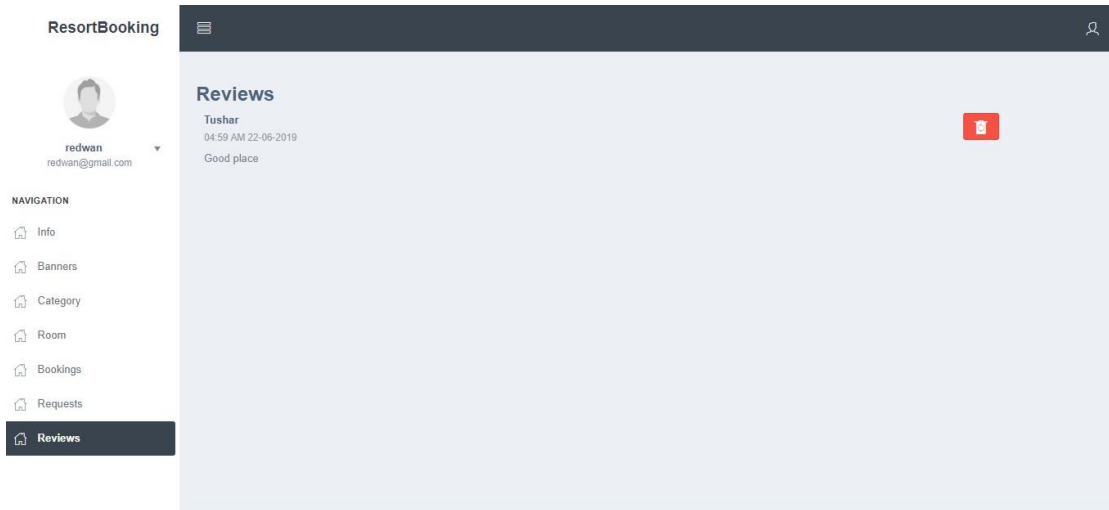


Figure 4.3.19: Review Management

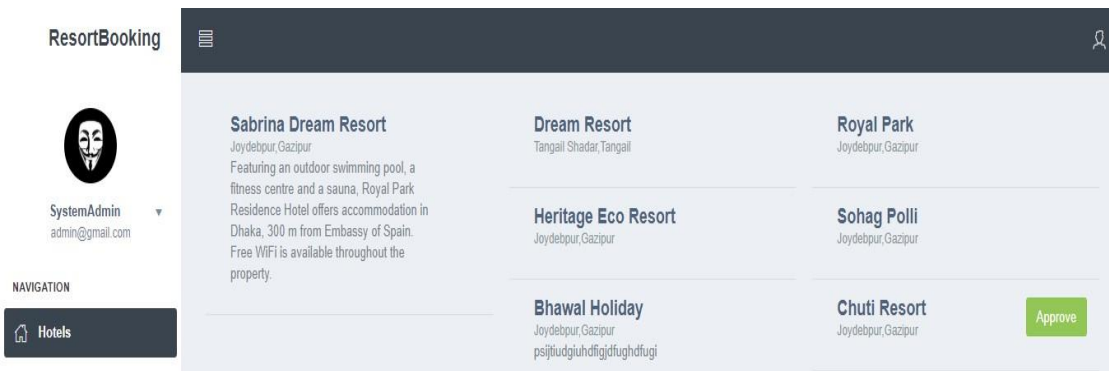


Figure 4.3.20: Manage resort

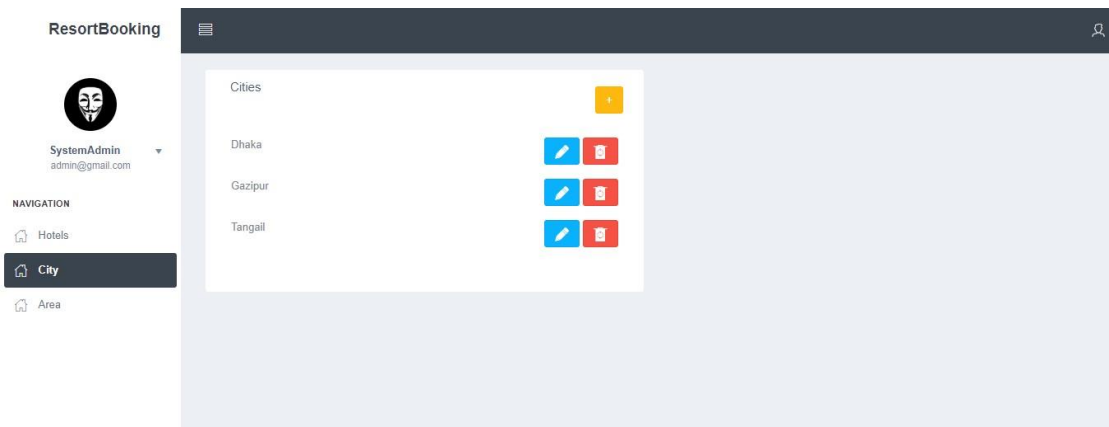


Figure 4.3.21: Manage area.

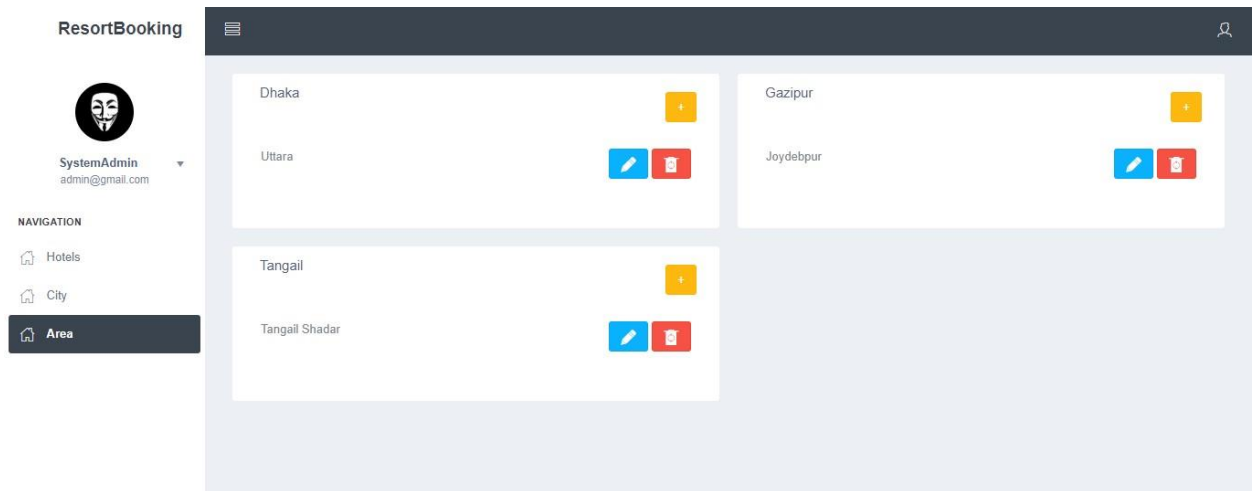


Figure 4.3.22: Manage places.

4.4 Backend Design

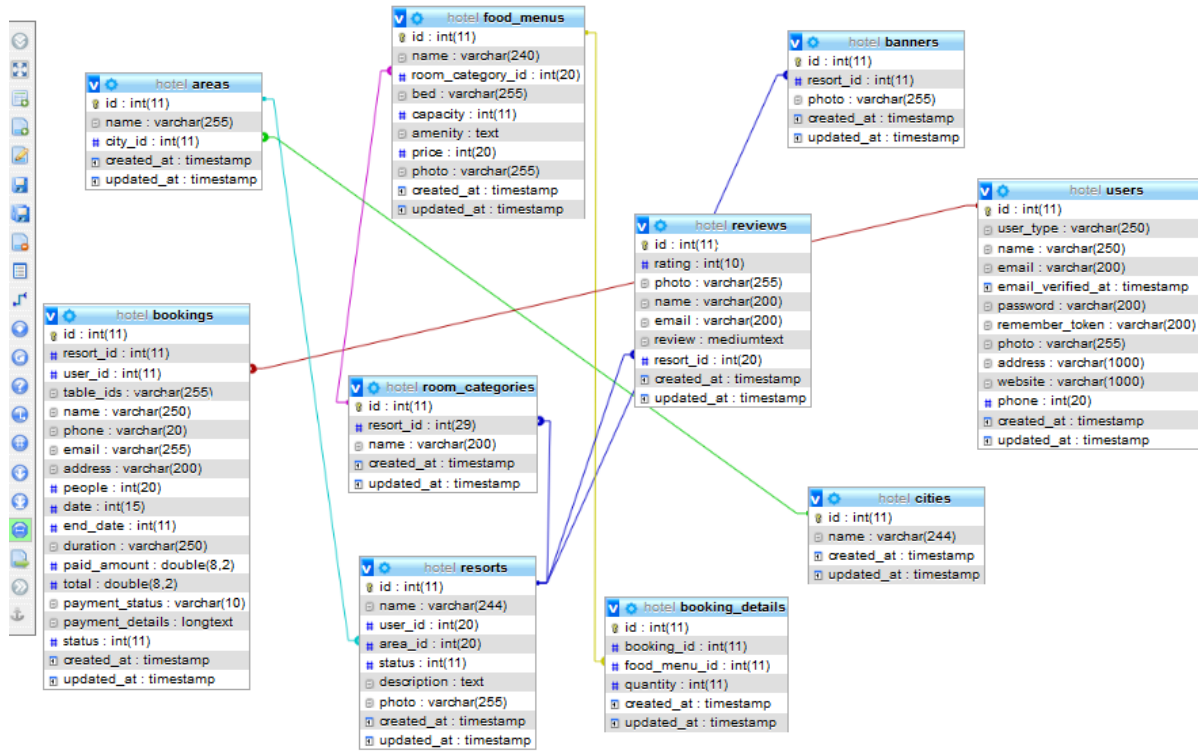


Figure 4.4.1: Database relationship

Table Name	Engine	Character Set	Collation	Row Count	Size
areas	InnoDB	utf32_bin		3	32 KiB
banners	InnoDB	utf8_unicode_ci		10	32 KiB
bookings	InnoDB	utf32_bin		11	80 KiB
booking_details	InnoDB	utf32_bin		11	48 KiB
cities	InnoDB	utf32_bin		3	16 KiB
food_menus	InnoDB	utf32_bin		13	32 KiB
resorts	InnoDB	utf32_bin		8	48 KiB
reviews	InnoDB	utf32_bin		3	32 KiB
room_categories	InnoDB	utf32_bin		10	32 KiB
users	InnoDB	utf32_bin		11	16 KiB
10 tables	Sum			83	368 KiB

Figure 4.4.2: Database table

4.5 Implementation Requirements

These are following things and technology that we need to implement this project. Here is the list below:

- Internet Browser
- Xampp Server
- Text Editor (ex: Atoms, sublime Text)
- PHP 7.2
- Laravel Framework
- HTML 5
- CSS
- jQuery 3.0+
- Bootstrap 4

4.6 Hardware Requirement

- Operating System (Windows, Linux, Mac OS, Android)
- Ram 1 GB or Higher
- HDD 256 GB
- Modem or Broadband (Must have good speed)

4.7 Software Requirement

- Internet Browser (Google chrome v60 or above)
- Internet Browser (Firefox v50 or above)

4.8 Summary

This system will give a very effective and detail information about resorts. It will be the best web application for anyone.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

We will discuss how we execute this undertaking. As it is an online venture so it will incorporate front-end and back-end usage. All the usage will appear with a few figures.

5.2 Implementation of database

This part will demonstrate how we execute our database. This will cover how we make association our database with how it responds with the clients. Additionally, the connections execution process.

```
DB_CONNECTION=mysql
DB_HOST="localhost"
DB_PORT="3306"
DB_DATABASE="hotel"
DB_USERNAME="root"
DB_PASSWORD=""
```

Figure 5.1: Database Connection

Presently here is the figure of database association process. It is extremely simple to interface database as we have worked in Laravel framework. The full venture is finished by Laravel framework. As we have utilized simple establishment process for the client so we utilized pre-characterized strategy for getting to the database name, username and secret word. At that point after get every one of the information from administrator every one of the information are utilized to set up secure database association with item arranged organization.

Presently after the database association usage we will discuss how every one of the tables are executed. In reality, we made every one of the tables legitimately utilizing xampp server physically. At that point we just moved the table. In the wake of finishing movement there is no other errand to do. At that point just need is do coding for information inclusion.

Here is the figure-5.2 for migration table.

```
$ php artisan make:model Review -m
```

Figure 5.2: Command for migration table

In the wake of completing this we physically set the essential key, outside key by entering the localhost xampp server. Furthermore, this is the all out procedure for table creation and set the properties.

Now we are going to talk about various types of query like insert, delete, update, retrieve. In Laravel framework it is very easy to do. Some screenshots of such coding is given below:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Booking;
use App\BookingDetail;
use App\Table;
use Session;
use Auth;
use Nexmo;

class BookingController extends Controller
{
    public function store(Request $request)
    {
        $booking = new Booking;
        $booking->resort_id = $request->resort_id;
        $booking->name = $request->name;
        $booking->phone = $request->phone;
        $booking->email = $request->email;
        $booking->address = $request->address;
        $booking->people = $request->people;
        $booking->date = Session::get('checkin');
        $booking->end_date = Session::get('checkout');
        //$booking->duration = $request->duration;
        //$booking->time = $request->time;
        if(Auth::check()){
            $booking->user_id = Auth::user()->id;
        }
        $booking->save();
    }
}
```

Figure 5.3: Query for insert data into database

“save()” is used to insert the data in specific table.


```
public function destroy($id)
{
    $city = City::find($id);
    $city->delete();

    return back();
}
```

Figure 5.3: Query of delete data

```
63 public function edit($id)
64 {
65     $city = City::find($id);
66     return view('systemadmin.city.edit', compact('city'));
67 }
68
```

Figure 5.3: Query of edit data from Database

```
5 public function index()
6 {
7     $cities = City::all();
8     return view('systemadmin.city.index', compact('cities'));
9 }
10
11 /**
```

Figure 5.3: Query of retrieve data

In the above figures we can see “retrieve query” and this is used to retrieve data from the database.

We can likewise observe "redirect ()- >route('dashboard')" which is utilized to take the client to dashboard. This is really taking us to our dashboard catalog. Course is utilized to send or coordinate along a predetermined course. For better comprehension here is given some code of course:

```

14 Route::get('/', function () {
15     return view('user.home');
16 }->name('user');
17
18 Route::post('/restaurant_list', 'HomeController@show_restaurants')->name('restaurant_list');
19
20 Route::get('/search', 'HomeController@search_restaurants')->name('restaurant_search');
21
22 Route::get('/restaurant_details/{id}', 'HomeController@show_restaurant_details')->name('restaurant_details');
23
24 Route::get('/filter', 'HomeController@filter')->name('filter');
25
26 Route::post('/addToCart', 'CartController@addToCart')->name('addToCart');
27
28 Route::post('/updateQuantity', 'CartController@updateQuantity')->name('updateQuantity');
29 Route::post('/removeFromCart', 'CartController@removeFromCart')->name('removeFromCart');
30 Route::post('/checkout', 'CheckoutController@checkout')->name('checkout');
31
32 Route::get('/order/{id}', 'CheckoutController@show_order')->name('order');
33
34 // SSLCOMMERZ Start
35 Route::get('/sslcommerz/pay', 'PublicSslCommerzPaymentController@index');
36 Route::POST('/sslcommerz/success', 'PublicSslCommerzPaymentController@success');
37 Route::POST('/sslcommerz/fail', 'PublicSslCommerzPaymentController@fail');
38 Route::POST('/sslcommerz/cancel', 'PublicSslCommerzPaymentController@cancel');
39 Route::POST('/sslcommerz/ipn', 'PublicSslCommerzPaymentController@ipn');
40 //SSLCOMMERZ END

```

Figure 5.4: Route query

Furthermore, the most critical thing is implementation of connection for every single page. What's more, for that we simply need to go the controller and afterward make a function based on our craving activity (ex: insert, update, edit, delete and so on). In that work we consider the model of table from where the information will be utilized. Furthermore, the function will return want an incentive to the page through course.

5.3 Implementation of booking system and time configuration

Presently we are going to discuss booking system and time design. For setting up and execution of booking system and time arrangement what code is composed is given underneath from figure-5.8 to figure-5.12:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Booking;
use App\BookingDetail;
use App\Table;
use Session;
use Auth;
use Nexmo;

class BookingController extends Controller
{
    public function store(Request $request)
    {
        $booking = new Booking;
        $booking->resort_id = $request->resort_id;
        $booking->name = $request->name;
        $booking->phone = $request->phone;
        $booking->email = $request->email;
        $booking->address = $request->address;
        $booking->people = $request->people;
        $booking->date = Session::get('checkin');
        $booking->end_date = Session::get('checkout');
        //$booking->duration = $request->duration;
        //$booking->time = $request->time;
        if(Auth::check()){
            $booking->user_id = Auth::user()->id;
        }
        $booking->save();
    }
}
```

Figure 5.5:Booking setup

```

<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Auth;
use Hash;
use App\Resort;
use Session;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        // $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Http\Response
     */

    public function datechoose(Request $request){
        Session::put('checkin', strtotime($request->start));
        Session::put('checkout', strtotime($request->end));

        return back();
    }
}

```

Figure 5.6: Home controller

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Redirect;
use Session;
use Illuminate\Routing\UrlGenerator;
use App\Http\Controllers;
use App\Order;
use App\BusinessSetting;
use App\Seller;
use App\Http\Controllers\CheckoutController;
use App\Http\Controllers\CommissionController;
session_start();

class PublicSslCommerzPaymentController extends Controller
{
    public function index(Request $request)
    {
        # Here you have to receive all the order data to initiate the payment.
        # Lets your order transaction informations are saving in a table called "orders"
        # In orders table order uniq identity is "order_id", "order_status" field contain status of the transaction, "grand_to

        $total = 0;
        foreach (Session::get('cart') as $key => $item) {
            $total = $total + $item['price'] * $item['quantity'];
        }

        $post_data = array();
        $post_data['total_amount'] = ($total*10)/100; # You cant not pay less than 10
        $post_data['currency'] = "BDT";
        $post_data['tran_id'] = substr(md5($request->session()->get('booking_id')), 0, 10); // tran_id must be unique
    }
}

```

Figure 5.6: Payment Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Resort;
use App\Admin;
use App\Table;
use App\TimeConfig;
use Auth;

class TimeConfigController extends Controller
{
    public function __construct()
    {
        //$this->middleware('auth');
    }

    public function index(){
        return view('admin.times.index');
    }

    public function showTimeConfig(){

```

Figure 5.7: Calculate TimeConfig

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Http\Controllers\BookingController;
use App\Booking;
use App\Http\Controllers\PublicSslCommerzPaymentController;
use Session;
use Mail;
use App\Mail\EmailManager;

class CheckoutController extends Controller
{
    public function show_order($id)
    {
        return view('user.order',compact('id'));
    }

    public function checkout(Request $request)
    {
        $bookingcontroller = new BookingController;
        $bookingcontroller->store($request);
        $sslcommerz = new PublicSslCommerzPaymentController;
        return $sslcommerz->index($request);
    }

    public function checkout_done($booking_id, $payment_details)
    {
        $total = 0;
        foreach (Session::get('cart') as $key => $item) {
            $total = $total + $item['price'] * $item['quantity'];
        }
    }

```

Figure 5.8: Checkout Submission

After completing this user can Book resort and specific room and maintain time configuration.

5.4 Testing Implementation

As this is an online endeavor that is the reason testing is particularly required for this structure before send the application. Cause it will check the general execution and can get the data that application data and yield estimation. On the other hand, a test can give the info that how much time it can make sense of how to run and can give unmistakable execution. There are such countless testing that should be feasible for application. For this application we have done some testing. They are given here:

5.4.1 Unit Testing

Unit testing is used for when the coding is executing then the designer checked for his own special basic mistakes. Thusly, for our circumstance we have checked when we coding and endeavored to find the hidden errors. It happens when we coding and after that test immediately.

5.4.2 Integration Testing

We have right off the bat done the front-end plan. At that point we have incorporated all the back-end code. At that point we have watched that if any sorts of design is broken or not. And furthermore, watched that all the interface works splendidly or not.

5.4.3 Functional Testing

All the limit that we have decided for our application from the start. We have finished all the fundamental essential of this undertaking ought to be. What's more, all of the limits are attempted autonomously. Since limits are the major indispensable part for a system. In case work does not work honestly, by then the structure will be regard less.

5.4.4 System Testing

System testing estimates that if the application is dealing with another system condition. As our application is completely online so we have checked every one of the programs and other working system and furthermore on various pc.

5.4.5 Usability Testing

The ease of use testing estimates the client association advertisement how well disposed the UI is. Along these lines, to do that we have testing on a few clients and get their experience on the UI and the amount they can explore effectively.

5.5 Test Result

We finished a couple of sorts of test only for a couple of reasons. In the wake of doing all the test there are some result. In light of these results we can get insist about our structure and its extraordinary side and horrendous side. Results are given bellow:

5.5.1 Result of Unit Testing

The full assignment relies upon coding. There are thousands line of code what we did here. In this titanic proportion of line coding it is absolutely possible to mess up or bungles. For finding this kind of stumbles we did unit testing. Moreover, when we did unit testing we got some bungle in our coding and that was our achievement of unit testing. In the wake of getting the bumbles and slip we got another chance to fix them. Moreover, we did accordingly. We enlightened them immediately then no further issue was found.

5.5.2 Result of Integration Testing

Incorporating testing was for no good reason and reasons are what code we formed for front-end and back-end design, are those lines of code working absolutely. Since the two plans are a ton of basic factor for any kind of endeavor. Accept we made an amazingly appealing front-end plan yet there are a couple of mistakes in back-end program and front-end and back-end arrangement isn't planning, in that time what will be said by customers. Thusly, to avoid this kind of issue we did blend test. In addition, there was no slip-up found in the midst of this test.

5.5.3 Result of Functional Testing

Capacity testing was a ton of goal. Since we did incredible front-end and back-end plan anyway we mixed up in utilitarian coding. Consequently, what is the estimation of hard and fast coding. It is silly. Pragmatic testing was the best way to deal with understand that

is every limit working authentically or not. Individually we checked every limit. In the midst of this testing we got some horrendous direct for marking in. when we got to marked in the accompanying system was taking us to confused course. What's more, moreover, got some horrendous practices in setting profile page when we move pictures and displayed that. There was some reasonable slip-up. Also, after that we fixed them. Another front-end site the administrators there was a hold issue on the program. This issue was crushed also by clearing progressively prepared archives and data new one. Along these lines, we have completed our functional testing.

5.5.4 Result of System Testing

It was additionally important to test since we know huge numbers of the general population use windows working system. Be that as it may, this does not mean no one uses others working system like MAC, Linux and so on. Along these lines, we additionally taken care of them. Along these lines, we did system testing. Furthermore, in the wake of testing what we see is our application has effectively kept running on others system and condition. So there was no blunder found.

5.6 Summary

In this part we finished some critical execution and testing. Here we found out about execution of database in Laravel structure. We made sense of it here in light of the fact that without database execution demonstrating it is intense for one to acknowledge and comprehend the program in Laravel structure. By then we comprehends Booking the executives and time setup use which is the most stand-out component of our errand. Here we exhibited the full code. Since here we worked with various thing, for instance, course figuring which is so difficult to appreciate without seeing the utilization. By then for security reason we did various sorts of testing what are fundamental for an endeavor. By then we further more showed the result and discussed it.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

We have attempted to cover every one of the things for client to manage booking in a resort. We have additionally make this report as a genuine programming improvement like feel. Presently we need to share what we have looked in doing this task and what we have gained from it.

Amid the absolute starting point period of our undertaking we have confronted different sort of issues and got opportunity to know numerous new things by experimentation premise. This voyage was an extraordinary accomplishment for us since we have tremendous thought regarding an exceptionally advanced and easy to use and vigorous system like Laravel. At first, we have begun our task utilizing Laravel outline work what was absolutely new for us. We took the difficulties lastly carried out our responsibility with some great supplements from our regarded resources. We made them intrigue experience as well. Amid our short voyage we got understanding of php, jquery, bootstrap and Laravel.

Most critical thing is that we have adapted genuine programming. We likewise have figured out how to form programming learning into advancement part. There was a critical part subterranean insect that was database plan. We got a decent thought how to build up a database as per the database model and styles.

6.2 Scope for Further Development

Because of the compelled proportion of time we couldn't add up to our goal. If we got sufficient opportunity, we had intention to incorporate some more features in our system. In future we have plan to extend all of the things referenced beforehand.

6.3 Limitation

There are some of limitation in our project, because of time limit. As we are beginner in Laravel we couldn't add some function, but we try our best to set up our project user friendly. Our design is not so attractive like other resort management websites. We had intension to include some more highlights in our system In future.

REFERENCES

- [1] Xampp Server available at: <https://www.apachefriends.org/download.html>, June, 2019
- [2] Sublime text editor available at: <https://www.sublimetext.com>, June 2019
- [3] Laravel available: <https://laravel.com/docs>, June 2019
- [4] HTML available at: <https://www.w3schools.com/htm>, June 2019
- [5] CSS available at: [https://www.w3schools.com/css/default .asp](https://www.w3schools.com/css/default.asp), June 2019
- [6] Bootstrap available at: <https://getbootstrap>, June 2019
- [7] jQuery available at: <https://stackoverflow.com>. , July 2019
- [8] Laravel resource available at: <https://stackoverflow>, July 2019

Plagiarism Report

