Design and Development of Venue Booking System

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

Daffodil International University

Design and Development of Venue Booking System

BY

FarzanaHossainSurovi

ID: 161-15-649

LutfunnaharLutfa

ID: 161-15-675

AND

NilufarYasminTamanna

ID: 161-15-687

Supervised By

Naznin Sultana

Assistant Professor

Department of CSE

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

December 2019

APPROVAL

This Project titled "**DESIGN AND DEVELOPMENT OF Venue Booking System**"submitted by FarzanaHossainSurovi ID No:161-15-649,Lutfunnanar Lutfa ID No:161-15-675 and NilufarYasminTamanna ID No:161-15-687 to the department of Computer Science and Engineering, Daffodil International University has accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 10 December 2019.

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Naznin Sultana, Assistant Professor, 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:

[illegible]

____

Naznin Sultana
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Surovi

____

FarzanaHossainSurovi
ID:161-15-649
Department of CSE
Daffodil International University

Lutfa

____

LutfunnaharLutfa
ID:161-15-675
Department of CSE
Daffodil International University

Tamanna

____

NilufarYasminTamanna
ID:161-15-687
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we distinct to Almighty for His divine sanction which assists us to complete the project of final year successfully.

We are really grateful to our honorable teacher **Naznin Sultana**, Assistant professor of Department of CSE Daffodil International University, Dhaka. Her broad knowledge and passionate concern in the field of "Web Design and Development" helps us for completing the project. Her fabulouspersistence, direction, constant inspiration, continuous &supervision ,determination of value , instruction, seeing many inferior draft and mend them every time have formed it enforceable to finish the project.

We would like to express our sincere gratefulness to Head, Department of CSE, for his kind support to finish our project.

We would like to admire our all course companion in DIU, who took part in this discuss while fulfilling the course activity.

Lastly, We must admit with due respect the continuous maintenance andpatienceof our parents.

ABSTRACT

As we are living in a period of science and technology so our most of the daily task are controlled by technology day by day. The exercise of web based systems are progressing day by day. Now people are more curious to set up technology in all area of their life. In this generation computer and internet is everywhere available. Every people can do a lot of things at home easily (ex: online shopping ,gathering information, booking any food, place etc).Because of traffic jam and massive of going in venues people like online booking more and base on this identification our full project is built. By using this system people or users can see different kind and categories of venues according to their own choice. They can book venues without bargaining and without facing any kind of loss. User can register in the system ,can give review about venues. Admin can mange the whole system also can update system by using admin panel.

TABLE OF CONTENTS

CONTENTS **PAGE NO**

CONTENTS **PAGE NO**

LIST OF FIGURES

FIGURES **PAGE NO**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays we are living in period of modern science and technology. As a result many of our daily activities are dependent on web systems and so on.

Today whatever our essentiality is at first, we search it on the Google through internet for searching anything. Then we gather the all information what we want. We don’t like to misuse our valuable time .

Booking is a very necessary work as well as people like to do booking places or venues. But the problem is for any kind of booking we need to go to venues which is a very time killing activity as well as it requires many energy and patience that’s the reason of getting bored easily and feeling tired.

Upon this we have taken step to solve this kind of problem by developing a web based booking system ‘Venue Booking System’ for all people.

We tried our level best to built all drift what are faced by people at booking time. It will help people to find out the best venues according to their choice, according to their budgets and it will also help them by giving all necessary information about their desired venue.

1.2 Motivation

To avoid misuse of time and to get a booking friendly and fresh environment without facing the problem of traffic jam on road and to ensure a budget friendly booking we searched for a web application over internet. We found some booking websites but according to remark it looks they are demanding a very high price comparing to venue price. So we definite to form a web application to assist people by knowing to make a bookingamicable and budget friendlyambitand also for providing venues and places without facing the problem of bargaining.

Then we discussed it with our honorable teacher NazninSultana(Assistant Professor, DIU-UC). Listening our idea about the project, she appreciates us to do this project and gave direction about how we can start the work.

1.3 Objectives

We decided to develop our web application about online venue booking so there are some objectives. Some of the targets of objectives that can be earned by our system:

- Building a user friendly web application.
- Finding out all possible requirements for designing the system.
- Finding necessary resources.
- Passing control to the users to maintain their profile.
- Passing control to the users to give review about venue.
- Passing control to the users to update their booking confirmation.
- Displaying organized information of venues, pictures and prices.
- Creating a pleasant admin interface.
- Testing the application for bugs and other errors.
- Doing necessary code and building up the application

1.4 Expected outcome

The possible outcomes which are expected to produce by the system are:

- Making a good designed and decorated web application
- It will very user friendly and easy to use.

- Getting reviews about the venues from users.
- User can visit website without registration.
- For booking any venues users must need to be registered.
- Registered user can recover their forgotten password.
- Admin panel can be accessible only by the registered admin.
- Admin can change their passwords.
- Admin can manage venue details.

1.5 Report Layout

The report format that is used to make this report on web application of online venue booking system is given below:

- All the topics are abounding with related information.
- All topics are divided with some details thus it become easy to understand.
- Required font and size is used to furnish all the contents of the report.
- All specific margins and spaces are used to format the content.
- Required booking are applied to get all the contents easily.

1.6 Summary

Technology is more faster than everything so after appearing this modern age we do not need to depend on anyone.Booking is very necessary for our everyday life and people also like to do booking places. This system will take all responsibility to give the best idea to users for booking.

CHAPTER 2

BACKGROUND

2.1 Introduction

The main reason behind making this web application is it will make booking system easier, reduce the pain of going places or venues and it will give a user friendly and very comfortable environment for booking. So we decided to make one which will be more users friendly for users.

2.2 Related work

There are also some related systems similar in our web system. We tried to do something impure and particular in our system. There are also some of related system but we try our best to develop our system many different ways. May some systems are more tempting but not so similar. In comparison with those systems our system is primarily not too much glamorous.

Here are some example of booking system and those websites features and drawbacks.

Website link:

- https://www.booking.com[1]
- www.bookingbuddy.com[2]
- www.venuebookingz.com[3]

2.3 Figures of related work

Figure 2.3.1 shows an online booking system named as booking.com[1].

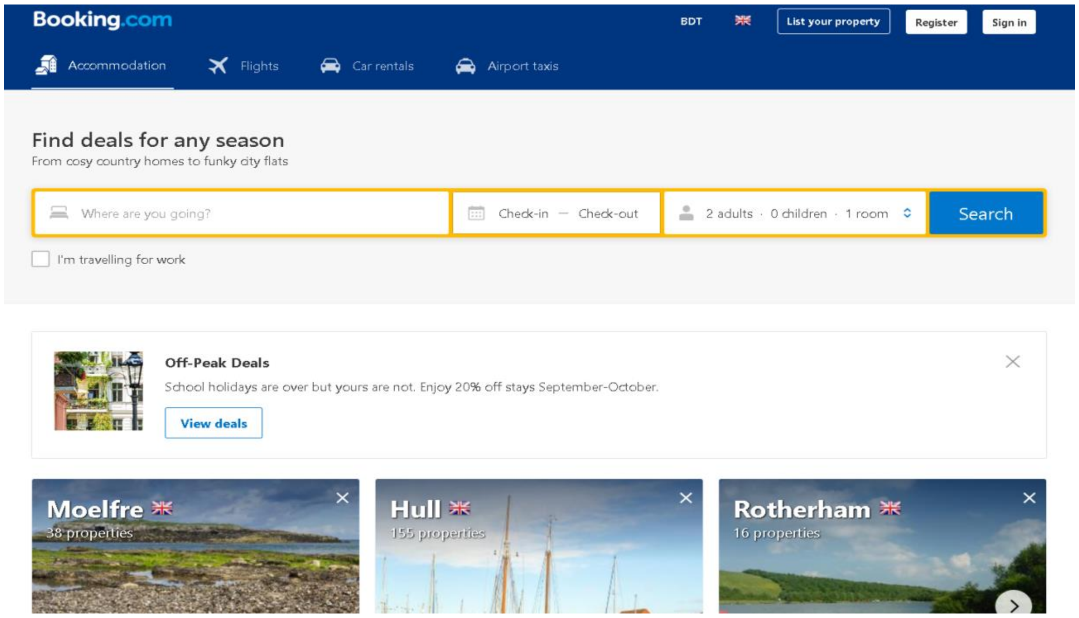
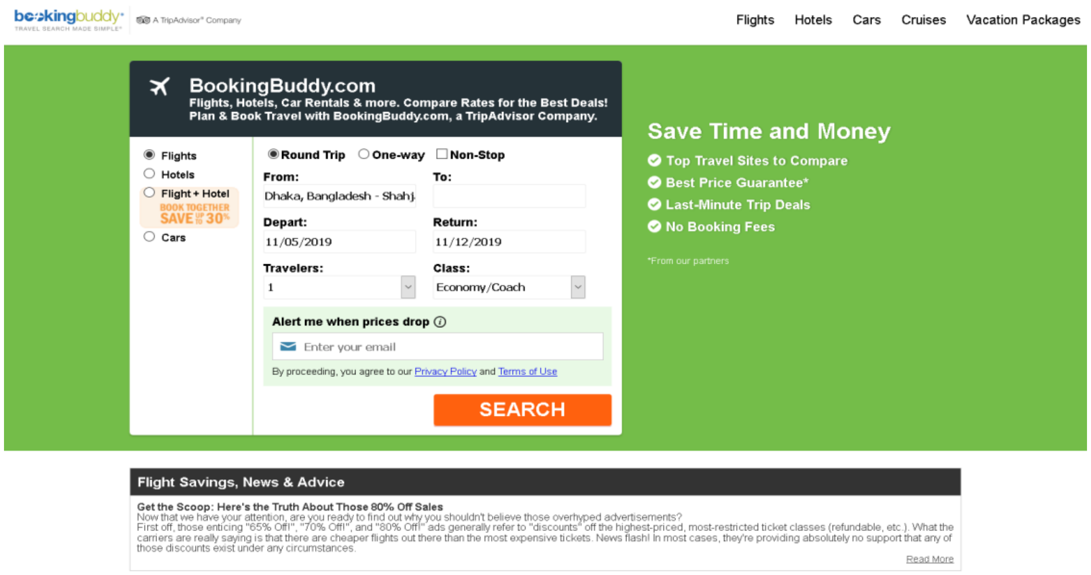


Figure 2.3.1:Related site1

Figure 2.3.2 shows another related site of online booking system named as bookingbuddy.com[2].



[● Flights] Hotels Flight + Hotel Cars

[● Round Trip] One-way Non-Stop

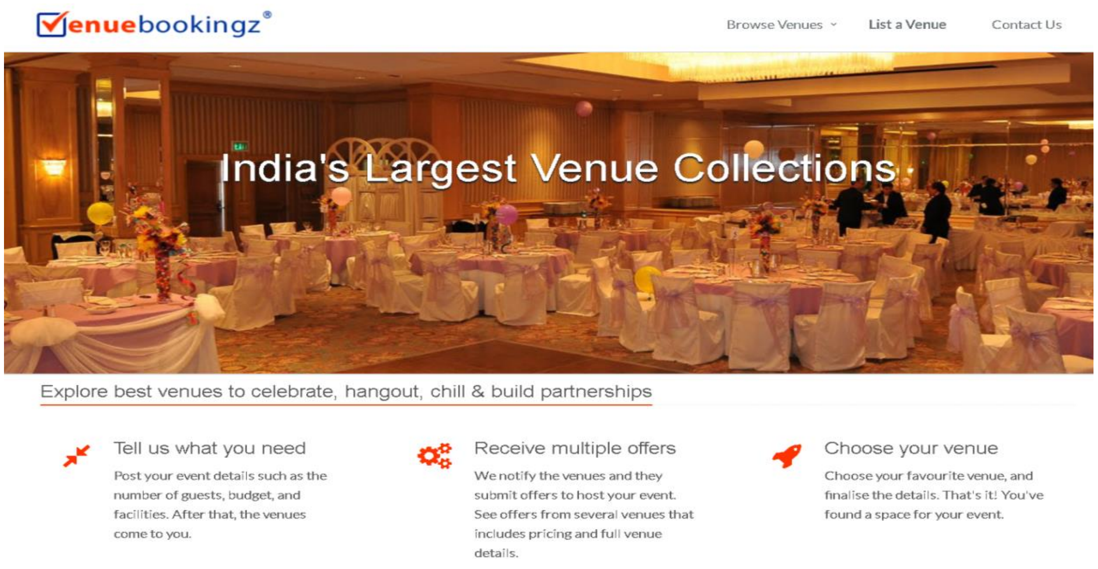
From: Dhaka, Bangladesh - Shahj	To:
Depart: 11/05/2019	Return: 11/12/2019
Travelers: 1	Class: Economy/Coach

By proceeding, you agree to our Privacy Policy and Terms of Use

Get the Scoop: Here's the Truth About Those 80% Off Sales
Now that we have your attention, are you ready to find out why you shouldn't believe those overhyped advertisements?
First off, those enticing "65% Off!", "70% Off!", and "80% Off!" ads generally refer to "discounts" off the highest-priced, most-restricted ticket classes (refundable, etc.). What the carriers are really saying is that there are cheaper flights out there than the most expensive tickets. News flash! In most cases, they're providing absolutely no support that any of those discounts exist under any circumstances.

Figure2.3.2: Related site 2

Figure 2.3.3 shows another related system in the field of online booking system named as venuebookingz.com[3].



Explore best venues to celebrate, hangout, chill & build partnerships

Tell us what you need
Post your event details such as the number of guests, budget, and facilities. After that, the venues come to you.

Receive multiple offers
We notify the venues and they submit offers to host your event. See offers from several venues that includes pricing and full venue details.

Choose your venue
Choose your favourite venue, and finalise the details. That's it! You've found a space for your event.

Figure 2.3.3: Related site3

2.4 Comparative Studies

We can see that there are some same content related online booking systems like our system but obviously there are some distinction between our system and those systems. First all that website don't incorporate all decisive information what users need.Then in the remaining system user has no idea of which events are going on and when they are taking place. Even though the users publish the ads through flexi posts etc. User who need to book for the events they need to visit thevenue.In The Proposed System the user need not go anywhere he can just get all information about events in a single click. The user needs to register need login there he can find the list of events which are added by the user and user can book by checking the availability. Then user needs to wait for the response and if admin confirms then user can take part in the event and can book venue easily.

2.5 Challenges

In developing this booking system we have falled so many demanding situations that allowto make users revel in higher and make the ability formative to the user. A few challenges are too hard to satisfy the purpose and some are absolutely new to locate the solution like these all

- Establishment of cleaning all function: It’s the ultimate part of any system. If the user don’t understand the way to use the system and why they will use it? That’s why we pretended to make an easy and essential manner to build this system for user.
- Making relation to the tables in database: Making relationship between tables of database is a very complex and also a hard process. We effort to do our paramount and normalized all tables of database for our system.
- Front page Processing: This system is an included internet site , and it’s a hard process to keep an internet site with a backend dashboard. We maintained it for a higher user recreate in.
- Admin’s dashboard: Making a dashboard for admin from where admin can manage the whole system is a very complex task and we tried our level best to execute it entirely.

2.6 Summary

Developing of any system it is very necessary to analyze the related work of the system. Vastly good and suitable analyzing no work can get it’s completeness and can’t also reach the goal. So after long time analyzing we decided to develop it and we settled our goal which will remove the inexistent of existing systems in our website.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 UML Diagram

In this division we are going to prepense about UML diagram. There are generous UML diagrams. Here we are going to review four of them. By using this two we are going to review our system.

3.1.1 Use Case Diagram

Use case diagram symbolize all actor whose are pertinent to the system and then show them how they can combat in the system.

Figure-3.1.1 shows the use case diagram of Venue Booking System.

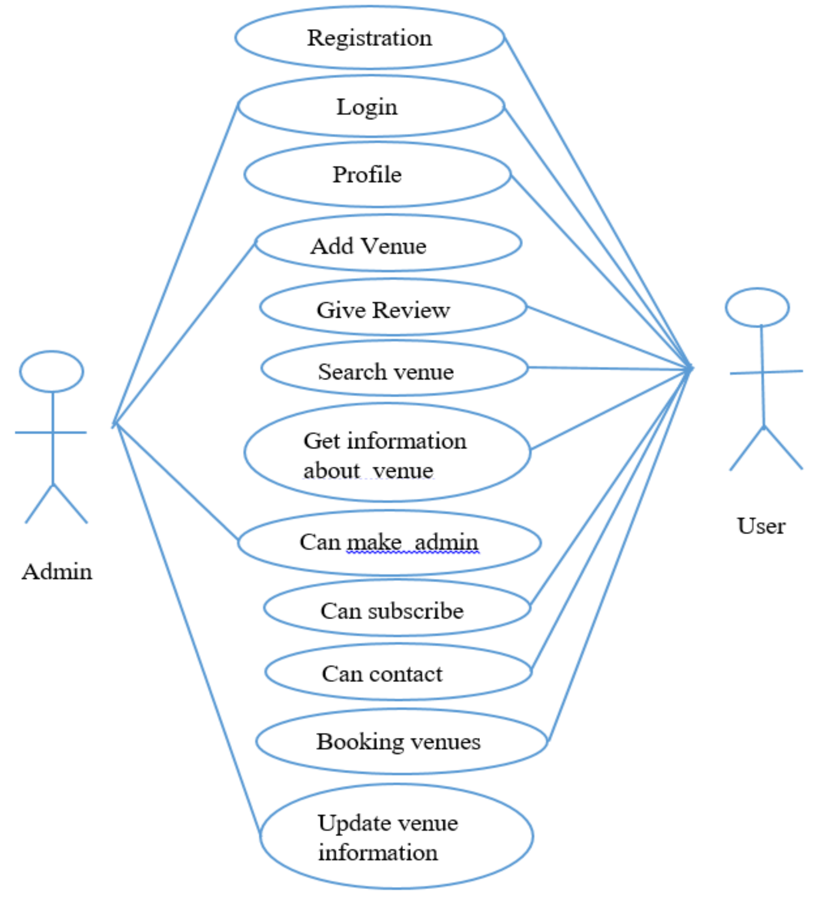


Figure 3.1.1: Use Case of Venue Booking System

3.1.2 Description of Use Case Model

In the grander use case diagram we observed that there are two actors in our system and they are (1) Admin and (2) User. In this system admin can make anyone admin according to his or her choice. Admin can add venues and manage also can update the venue evidence. User can visit the site without login and also can give search venues and can contact to the system. But if user hungriness to booking any venue then they must need to be a registered user. This table 3.1.2(a) shows the login process of the system.

Table 3.1.2(a): Use Case of Login

Cases	Details
Actors	Admin, Users.
Pre-conditions	-Registration is obligatory.
Post-conditions	Confirming successful login actors will go to their anticipated dashboard.
Basic Flows	Actors will login with their login credentials .Afterwards they will reach in their desired dashboard for continuation of their further work.

This table 3.1.2(b) shows the process of adding product in the system.

Table 3.1.2(b): Use case of adding venue

Cases	Details
Actors	Admin.
Pre-condition	-Login into the system. -Enter into dashboard. -Go to insert product option.
Post-condition	After going to insert product option admin can easily insert product and also can manage them.
Basic Flow	Admin will insert products name image price and others details information.

This table 3.1.2(c) shows the process of giving review about venues.

Table 3.1.2(c): Use case of giving review

Cases	**Details**
Actors	Users.
Pre-condition	No compulsion of login both registered and unregistered user can accord review.
Post-condition	Actors can also see others review about venues.
Basic Flow	When any actorcome to visit the website then they can view the venue details and can give review.

This table 3.1.2(d) shows the process of booking venues.

Table 3.1.2(d): Use case of booking venues

Cases	**Details**
Actors	Users.
Pre-condition	Actors can booking venue but if they wants to booking any venue then must need to be logged in.
Post-condition	After successfully booking venue user will able to see the booking status.
Basic Flow	-Firstly need to fill up the booking form. -Click on submit and after it venue will be booked. -get notification in email. -pay booking amount.

3.1.3 System Flow Diagram

Here we are added our flow diagram in figure-3.1.3. Our system has two types of main actors. They are the admin and the users. For user there is no need of being registered for visiting the site, searching venues, viewing venue details, giving review about venues. But if the user wants to booking any venue then he/she must need to be logged in by going through registration process. Registered users can manage their profile and able to see booking history. Now for getting access of admin panel must need to be registered. After successfully log in the admin will redirect to admin dashboard from where admin can perform the task of changing password, booking management, user management, inserting venue and managing venue. Admin can also visit site as a user after logout.

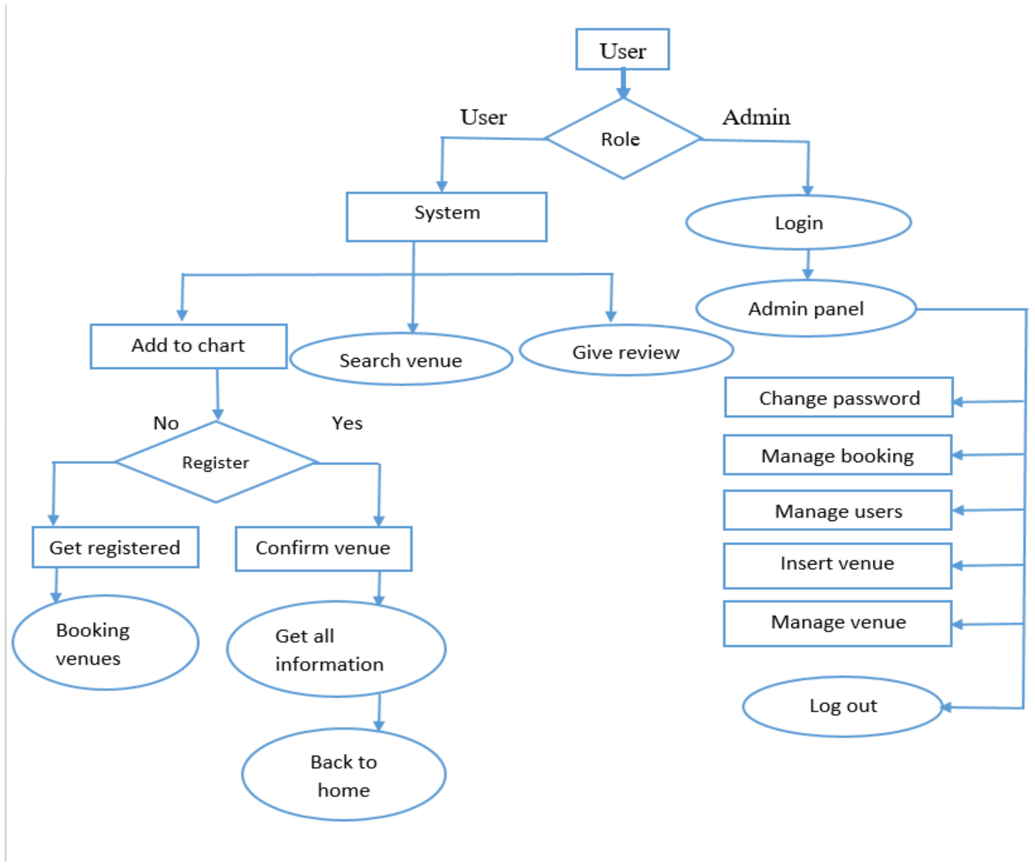


Figure 3.1.3: System flow diagram

3.1.4 Entity Relationship Diagram

The figure 3.1.4 shows the ER diagram of this venue booking system.

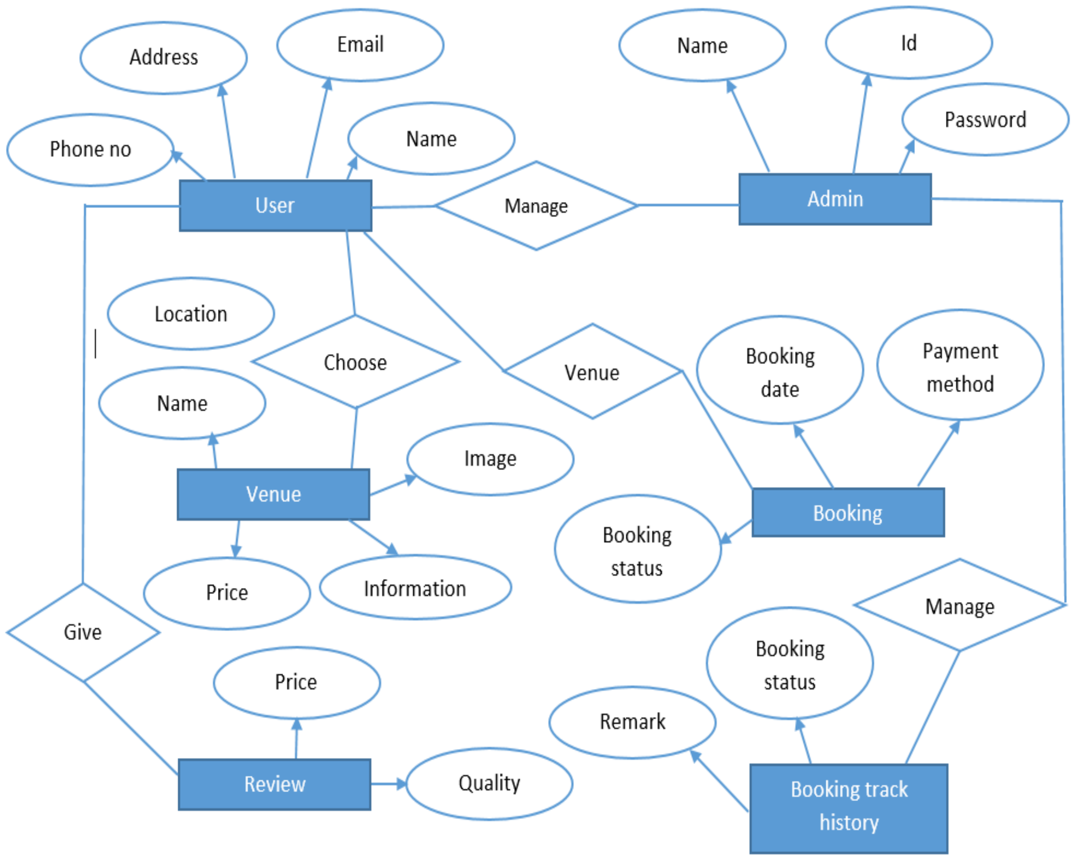


Figure 3.1.4: E-R Diagram

3.1.5 UML Class Diagram

The figure 3.1.5 shows the UML class diagram of venue booking system.

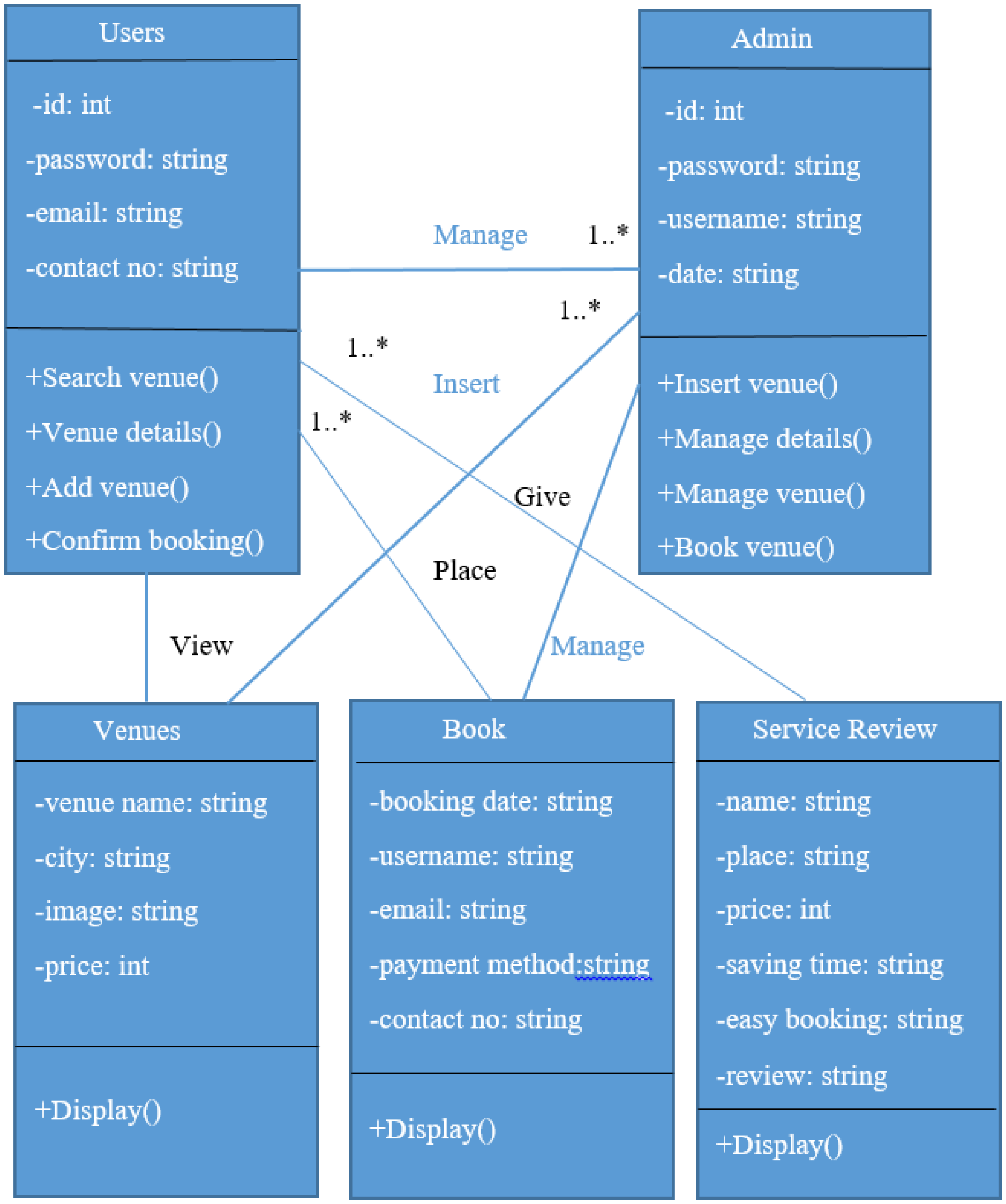


Figure 3.1.5:UML Class Diagram

3.2 Design Requirements

We have to bring some design to form this system more approachable and expectable to user. So in aforesaid system the designs are used in frontend and the backend design.

For realizing the frontend design we have the responsibility to drawing the layouts into paper first so that we can find how the website will look agnate. For the front end design we HTML[8], CSS[7], PHP, and also used bootstrap framework for design faster and easier, bootstrap js[5] and some jQuery[6] for completing the layout. Then we need to design the project as the documentation intended and perfection coding.

For backend design we have to need the database structure and the knowledge on relational database. There is also duty to dividend the relationship among the tables in the database and also responsibility to keep in mind excess things. So that we have urgency of attained a few normalization. For backend coding we tried to follow the object-oriented coding mark.

So that's all the design requirements that needed for this project.

3.3 Summary

In aforesaid chapter we givenall about the full project take in use case diagram, system flow diagram, ER diagram and UML class diagram. We also represent them courteously.

CHAPTER 4

SYSTEM DETAILS

4.1 Introduction

This chapter is for the brief discussion about the aforesaid system. After comprehensively reading this chapter individual will attain at least 70% explanation about our application. Aforesaid chapter we will discussion about our system trait, front-end design, back-end database and implementation object.

4.2 Application Feature

We have used various features in our system. Basically we are constrain to display our features list and after we will depicture them. So, the features of this system are-

- Visit website- Any user either registered or non registered can visit this website.
- Search venue- An option is yielded to search commodity.
- Category selection- Users can select category for finding related venues.
- See venue details- User can see venue details by viewing the details of the venue.
- Give venue review- User can give review about venues and can see others review.
- Registration- User have urgency to be registered so that they can booking venues and compose own profile.
- Login- For booking venues login is compulsory.
- Venue booking- Users can do all the aforesaid activity but whenever they wished to booking venue then they have to be registered in the system.
- Update profile- Registered user can update and manage their profiles.
- Filter venues- User can filter venues to find out exact venue which they want.

4.2.1 Visit Website

Any user either registered or non registered user can visit this website. Visiting need no registration or login process.

4.2.2 Search Venue

This system provides an option for manual searching. User can search venue manually to get the accurate venue they want.

4.2.3 See venue details

User can glimpse the details info of any venue by searching the venue list.

4.2.4 Give venue review

Users can give review about the venue quality, price and others things and user can also see the review of other peoples.

4.2.5 Registration

User need to go through the registration process for booking any venue and also for making personal profile.

4.2.6 Login

Registered user can login into the system for booking venues and will be able to observe the list of all venues.

4.2.7 Booking venue

After log in into the system user can place order for booking the particular venue.

4.2.8 Update profile

Registered user can update their profile by logging into the system.

4.2.9 See all venues

Registered user can view the booking history of previous in list view.

4.3 Front End Design

This figure 4.3.1 shows the home page of the venue booking system.

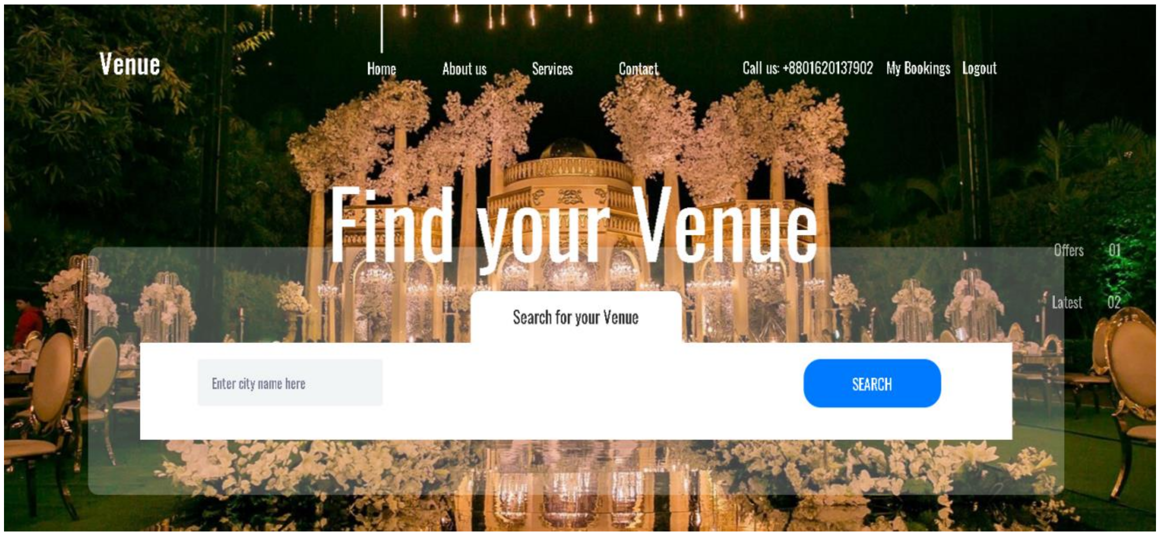
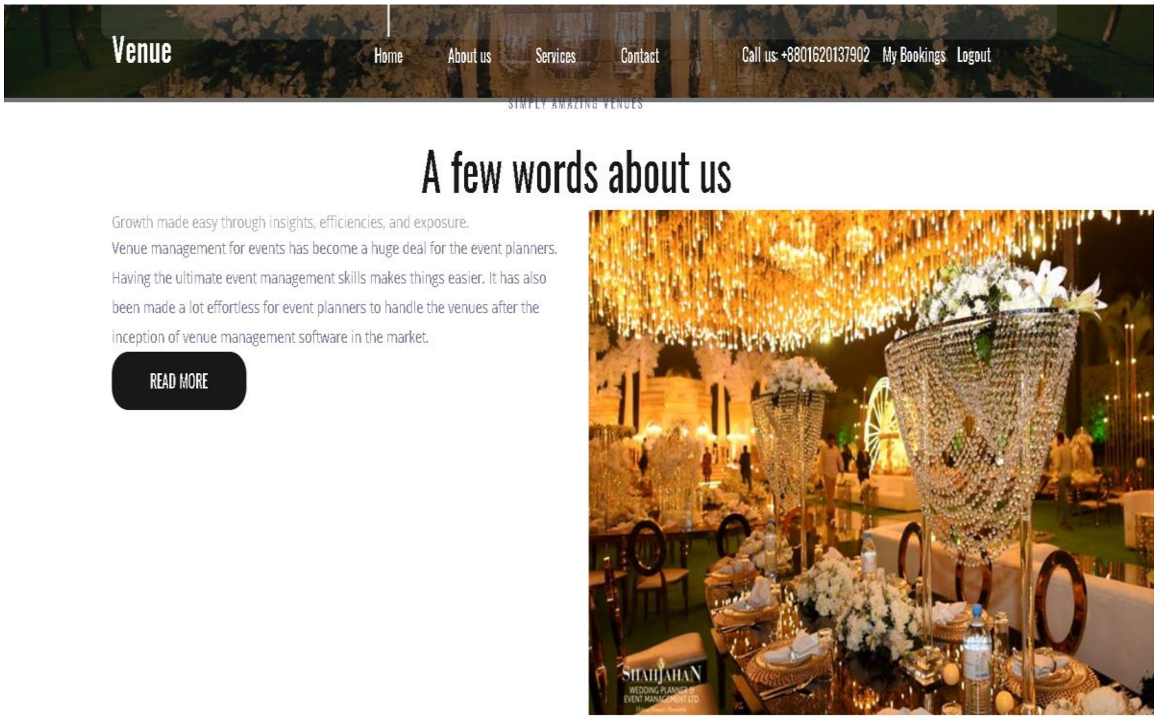


Figure4.3.1: Home page

This figure 4.3.2 shows the about us page of the system.

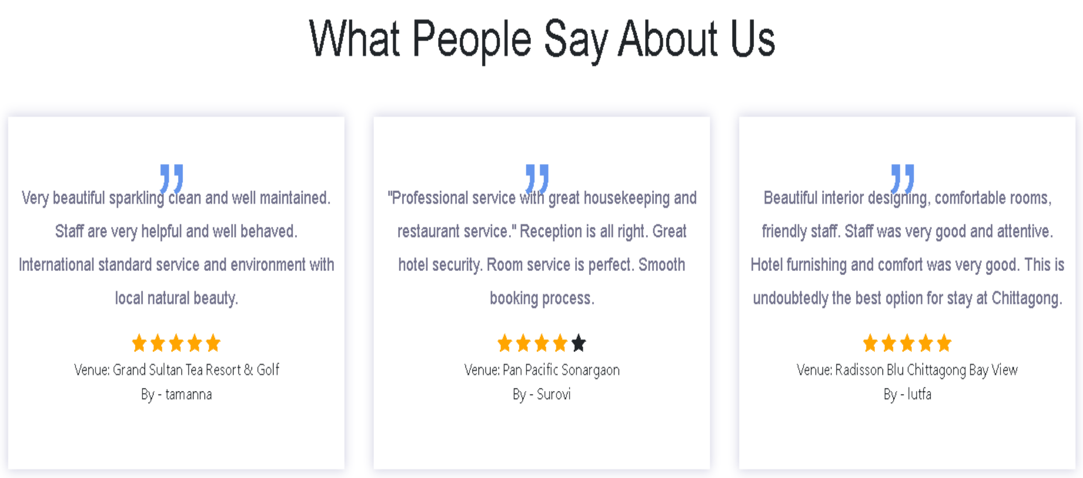


A few words about us

Growth made easy through insights, efficiencies, and exposure.
Venue management for events has become a huge deal for the event planners. Having the ultimate event management skills makes things easier. It has also been made a lot effortless for event planners to handle the venues after the inception of venue management software in the market.

Figure 4.3.2: About us.

This figure 4.3.3 shows the venue review page of this website.



What People Say About Us

Very beautiful sparkling clean and well maintained. Staff are very helpful and well behaved. International standard service and environment with local natural beauty.

★★★★★

Venue: Grand Sultan Tea Resort & Golf

By - tamanna

"Professional service with great housekeeping and restaurant service." Reception is all right. Great hotel security. Room service is perfect. Smooth booking process.

★★★★★

Venue: Pan Pacific Sonargaon

By - Surovi

Beautiful interior designing, comfortable rooms, friendly staff. Staff was very good and attentive. Hotel furnishing and comfort was very good. This is undoubtedly the best option for stay at Chittagong.

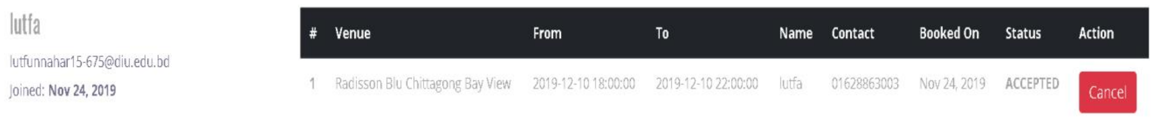
★★★★★

Venue: Radisson Blu Chittagong Bay View

By - lutfa

Figure 4.3.3: Venue Review

This figure 4.3.4 shows the my booking status option of this system.



lutfa

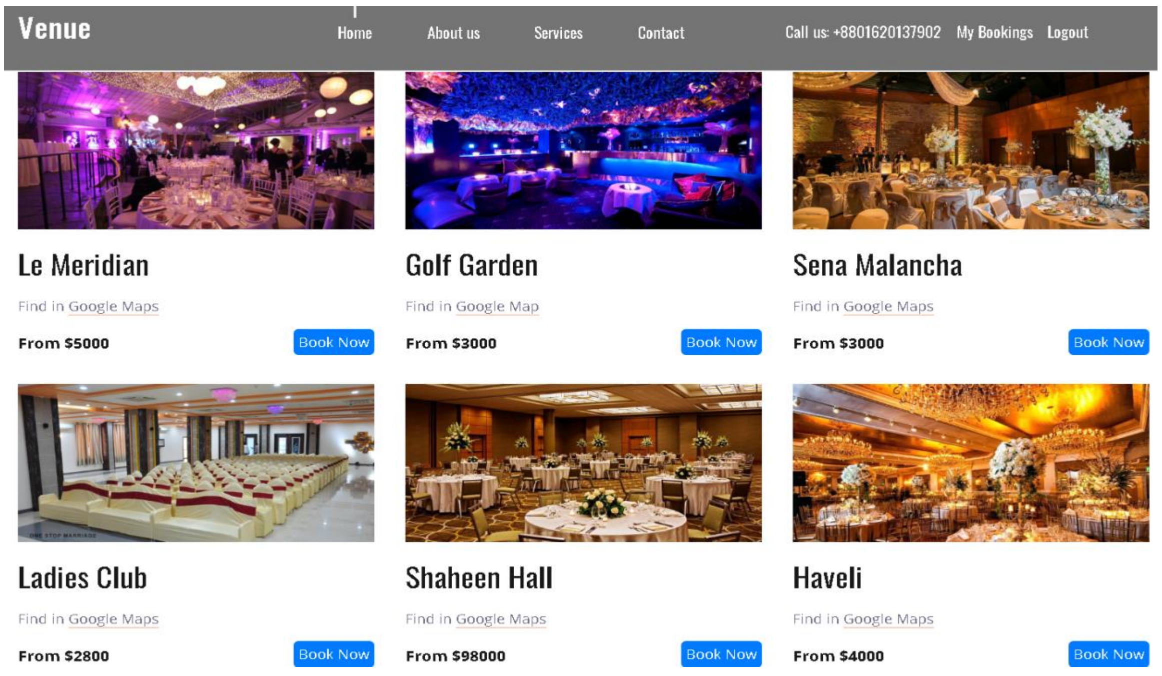
lutfunnahar15-675@diu.edu.bd

Joined: **Nov 24, 2019**

#	Venue	From	To	Name	Contact	Booked On	Status	Action
1	Radisson Blu Chittagong Bay View	2019-12-10 18:00:00	2019-12-10 22:00:00	lutfa	01628863003	Nov 24, 2019	ACCEPTED	Cancel

Figure 4.3.4: My booking status

This figure 4.3.5 shows the venue details page of the system.



Le Meridian
Find in Google Maps
From $5000

Golf Garden
Find in Google Map
From $3000

Sena Malancha
Find in Google Maps
From $3000

Ladies Club
Find in Google Maps
From $2800

Shaheen Hall
Find in Google Maps
From $98000

Haveli
Find in Google Maps
From $4000

Figure 4.3.5: Venue details Page

This figure 4.3.6 shows the login page of the system.

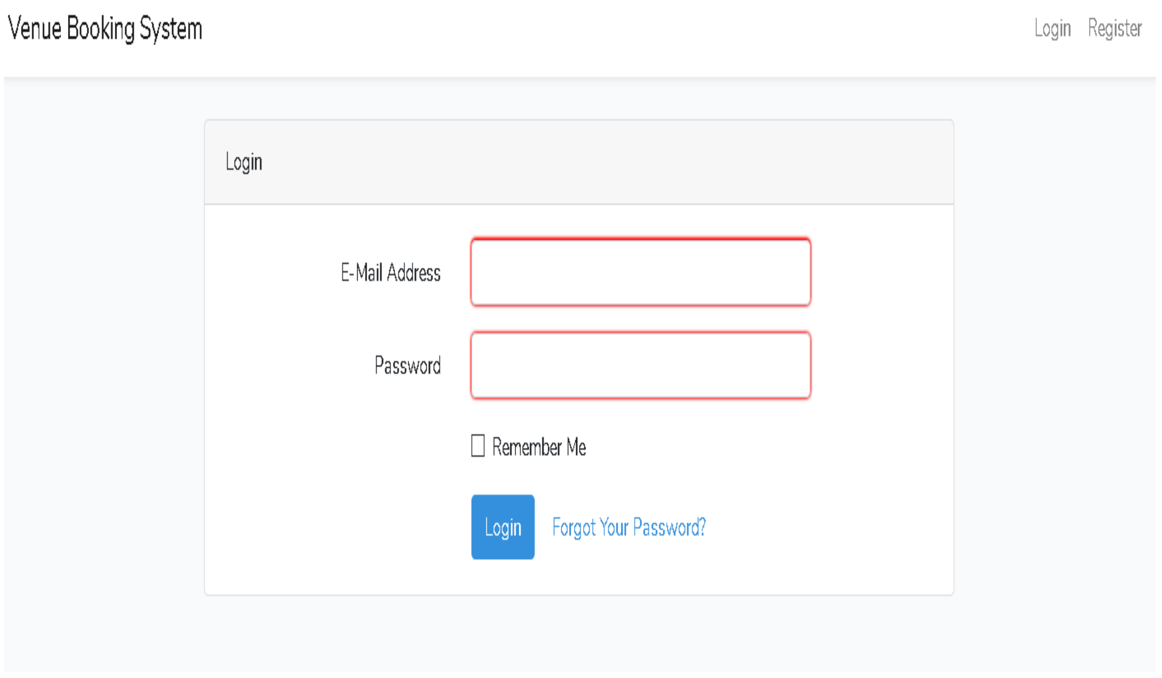


Figure 4.3.6: User Login page

This figure 4.3.7 shows the admin login page of the system.

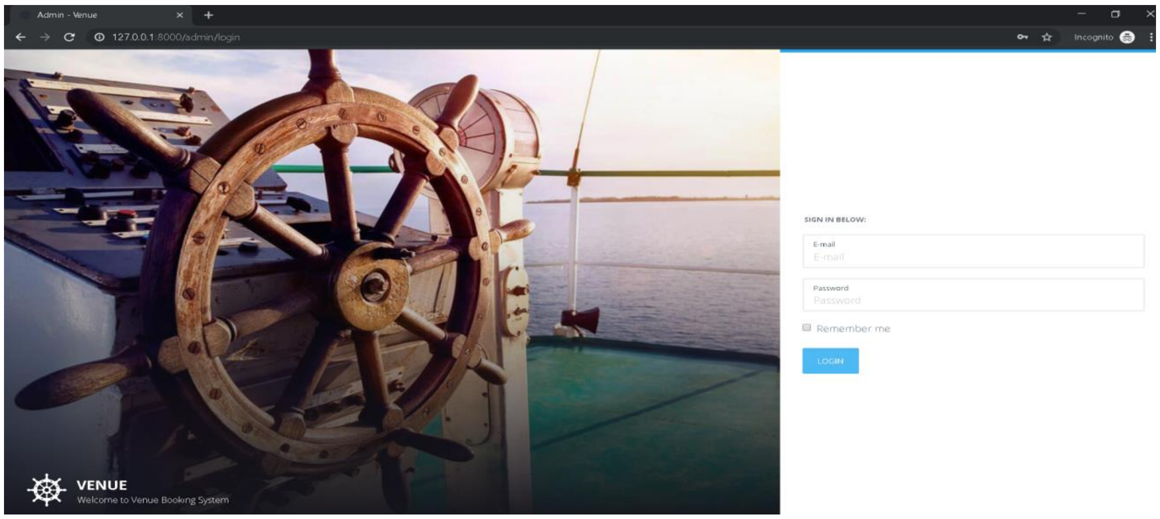
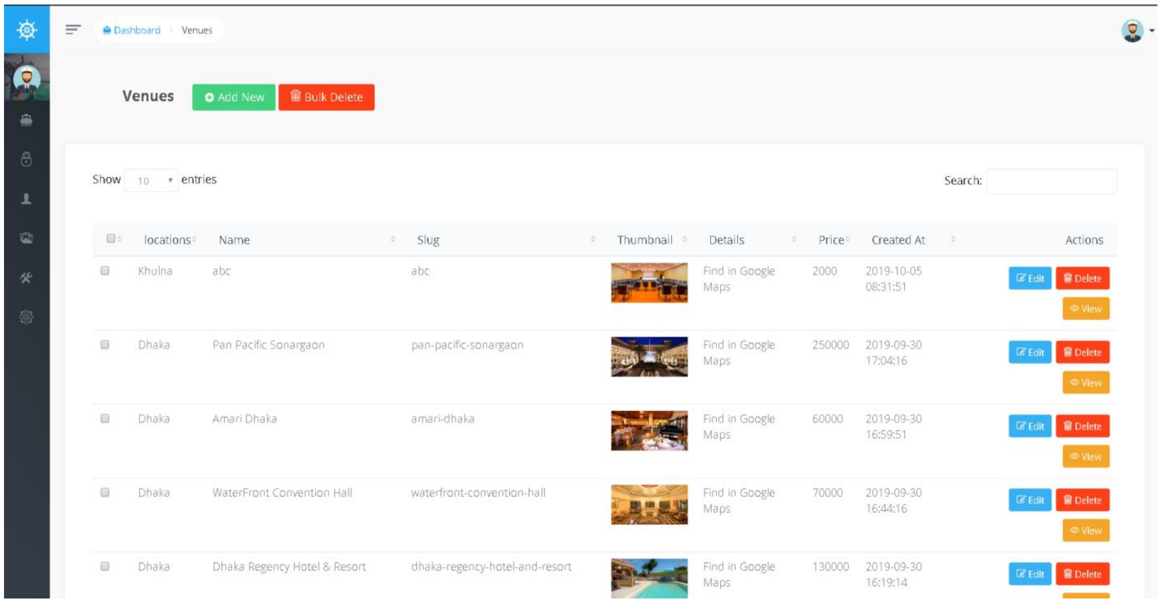


Figure 4.3.7: Admin login page

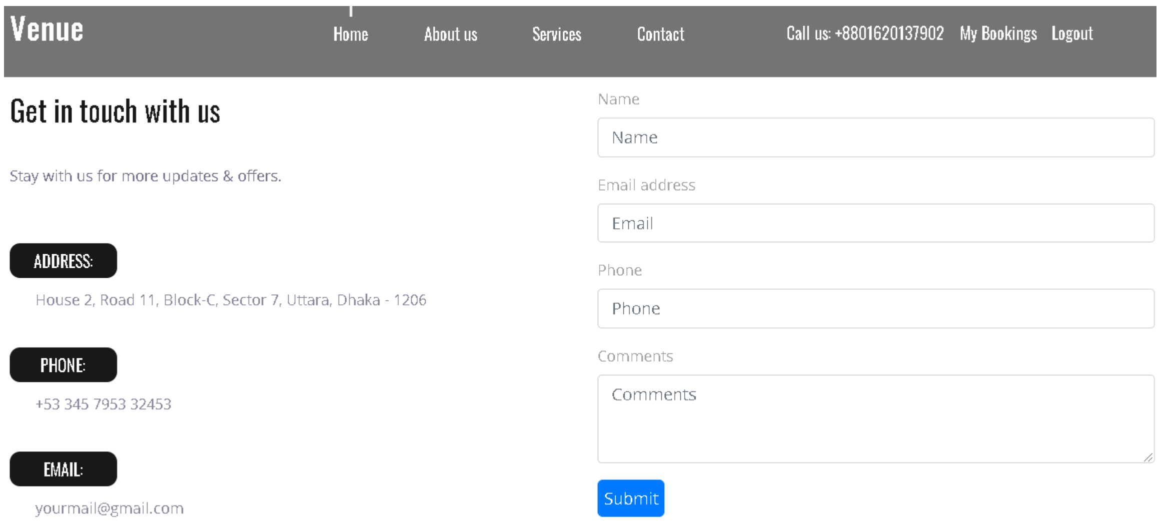
This figure 4.3.8 shows theVenues insertion page accessed by admin.



	locations	Name	Slug	Thumbnail	Details	Price	Created At	Actions
	Khulna	abc	abc		Find in Google Maps	2000	2019-10-05 08:31:51	Edit Delete View
	Dhaka	Pan Pacific Sonargaon	pan-pacific-sonargaon		Find in Google Maps	250000	2019-09-30 17:04:16	Edit Delete View
	Dhaka	Amari Dhaka	amari-dhaka		Find in Google Maps	60000	2019-09-30 16:59:51	Edit Delete View
	Dhaka	WaterFront Convention Hall	waterfront-convention-hall		Find in Google Maps	70000	2019-09-30 16:44:16	Edit Delete View
	Dhaka	Dhaka Regency Hotel & Resort	dhaka-regency-hotel-and-resort		Find in Google Maps	130000	2019-09-30 16:19:14	Edit Delete

Figure 4.3.8: Venues insertion

This figure 4.3.9shows the contact page of this system.



Get in touch with us

Stay with us for more updates & offers.

ADDRESS:

House 2, Road 11, Block-C, Sector 7, Uttara, Dhaka - 1206

PHONE:

+53 345 7953 32453

EMAIL:

yourmail@gmail.com

Name: Name

Email address: Email

Phone: Phone

Comments: Comments

Figure 4.3.9: Contact page

4.4 Backend Design

This figure 4.4.1 spectacles the backend database of the aforesaid system.

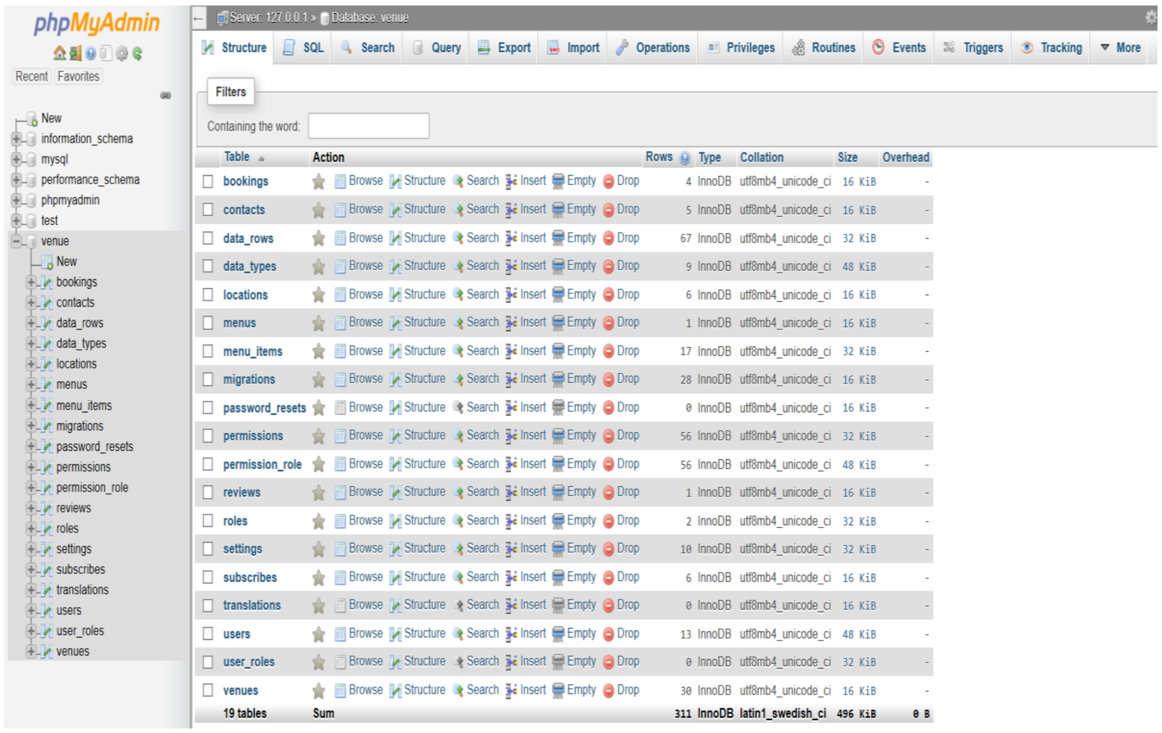


Table	Action	Rows	Type	Collation	Size	Overhead
bookings	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_unicode_ci	16 KiB	-
contacts	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_unicode_ci	16 KiB	-
data_rows	Browse Structure Search Insert Empty Drop	67	InnoDB	utf8mb4_unicode_ci	32 KiB	-
data_types	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_unicode_ci	48 KiB	-
locations	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16 KiB	-
menus	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
menu_items	Browse Structure Search Insert Empty Drop	17	InnoDB	utf8mb4_unicode_ci	32 KiB	-
migrations	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_unicode_ci	16 KiB	-
password_resets	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
permissions	Browse Structure Search Insert Empty Drop	56	InnoDB	utf8mb4_unicode_ci	32 KiB	-
permission_role	Browse Structure Search Insert Empty Drop	56	InnoDB	utf8mb4_unicode_ci	48 KiB	-
reviews	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	16 KiB	-
roles	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32 KiB	-
settings	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_unicode_ci	32 KiB	-
subscribes	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_unicode_ci	16 KiB	-
translations	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
users	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_unicode_ci	48 KiB	-
user_roles	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32 KiB	-
venues	Browse Structure Search Insert Empty Drop	30	InnoDB	utf8mb4_unicode_ci	16 KiB	-
19 tables	Sum	311	InnoDB	latin1_swedish_ci	496 KiB	0 B

Figure 4.4.1: Back-end database

4.5 Implementation Requirements

The succeeding stuffs and technology that we have obligation to implement in this project are:

- Text Editor(ex: notepad++,Visual Studio)
- Xampp Server[4]
- Bootstrap3[5]
- PHP7
- jQuery3.0+[6]
- CSS3[7]
- HTML5[8]
- Internet Browser.

4.6 Hardware Requirements

- Operating System(Ex: Window)
- Ram1GB or Higher
- HDD 256GB
- Modem or Broadband (Must have good speed).

4.7 Software Requirements

- Operating System(Ex: Window)
- Ram4GB
- HDD 1024GB
- Modem or Broadband (Must have good speed).

CHAPTER 5

IMPLEMENTATION AND TESTING

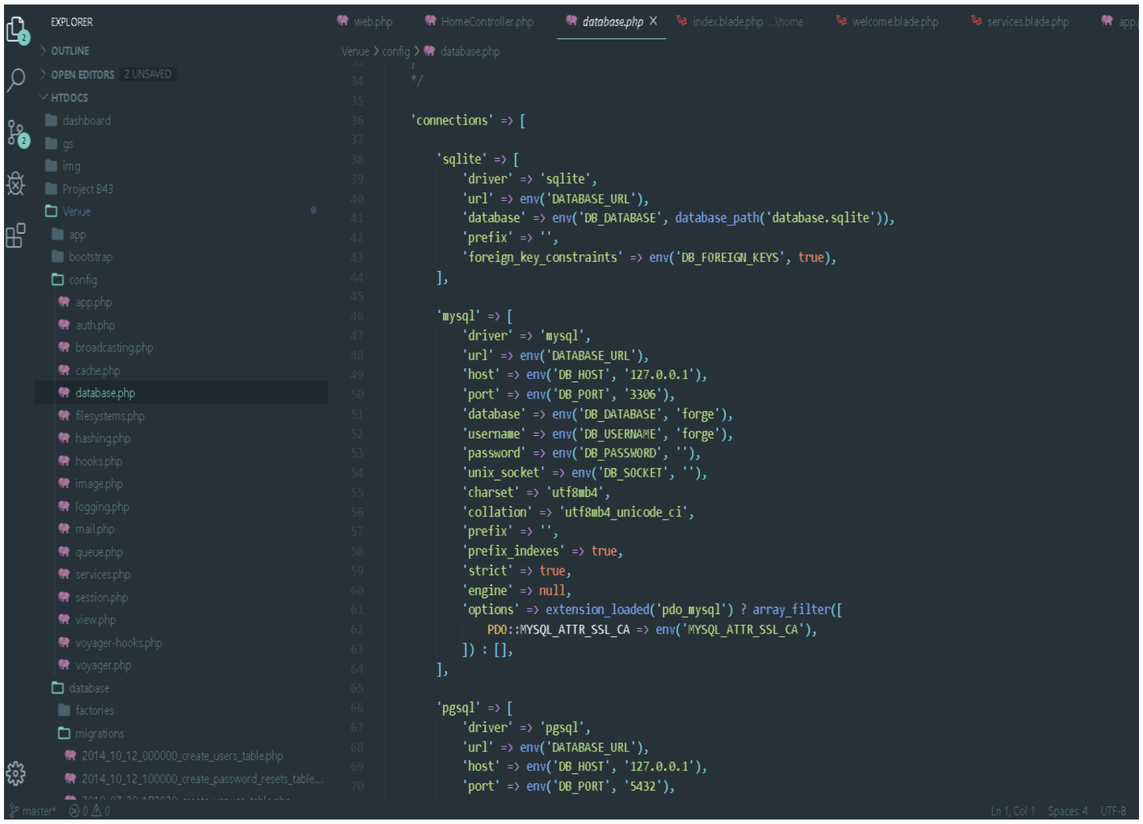
5.1 Introduction

Forthwith in this chapter we will confabulate about how we implemented aforesaid system. As it's a web-based project so its construct front-end and back-end implementations. All the implementation will confabulate with respective pictures in this chapter.

5.2 Implementation of Database

This part will show how we implement our database. This will cover how we make connection with our database and how it reacts with the users.

This figure 5.2.1 shows the database connection process.



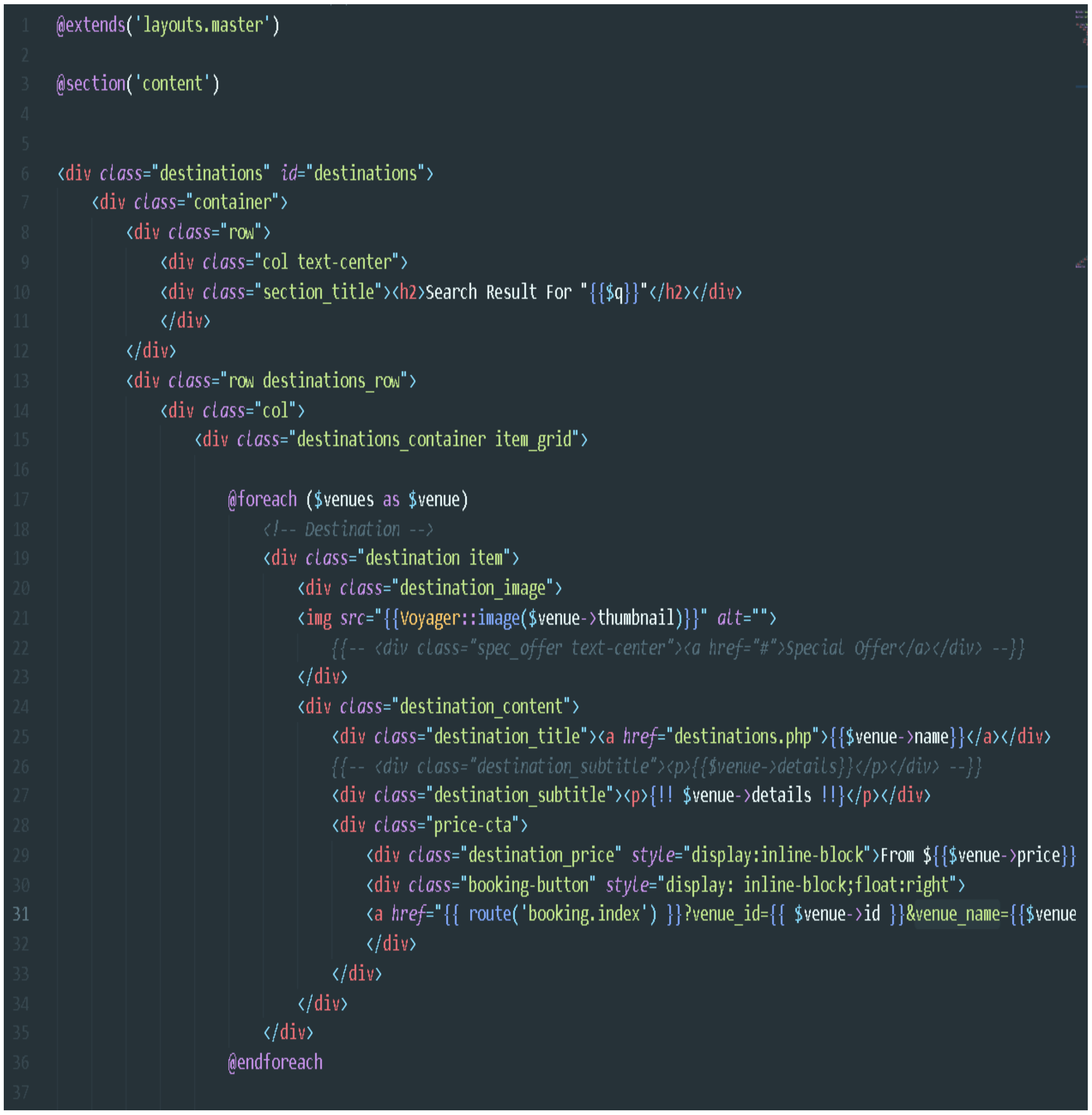
```
    */

    'connections' => [

        'sqlite' => [
            'driver' => 'sqlite',
            'url' => env('DATABASE_URL'),
            'database' => env('DB_DATABASE', database_path('database.sqlite')),
            'prefix' => '',
            'foreign_key_constraints' => env('DB_FOREIGN_KEYS', true),
        ],

        'mysql' => [
            'driver' => 'mysql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '3306'),
            'database' => env('DB_DATABASE', 'forge'),
            'username' => env('DB_USERNAME', 'forge'),
            'password' => env('DB_PASSWORD', ''),
            'unix_socket' => env('DB_SOCKET', ''),
            'charset' => 'utf8mb4',
            'collation' => 'utf8mb4_unicode_ci',
            'prefix' => '',
            'prefix_indexes' => true,
            'strict' => true,
            'engine' => null,
            'options' => extension_loaded('pdo_mysql') ? array_filter([
                PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
            ]) : [],
        ],

        'pgsql' => [
            'driver' => 'pgsql',
            'url' => env('DATABASE_URL'),
            'host' => env('DB_HOST', '127.0.0.1'),
            'port' => env('DB_PORT', '5432'),
```

Figure 5.2.1: Database connection

It is very easy to connect database as we have used easy installation process for the user so we used pre-defined method for accessing the database name, username and password.

Posterior composing the connection with database we construct our tables in the database and inserted data on them.

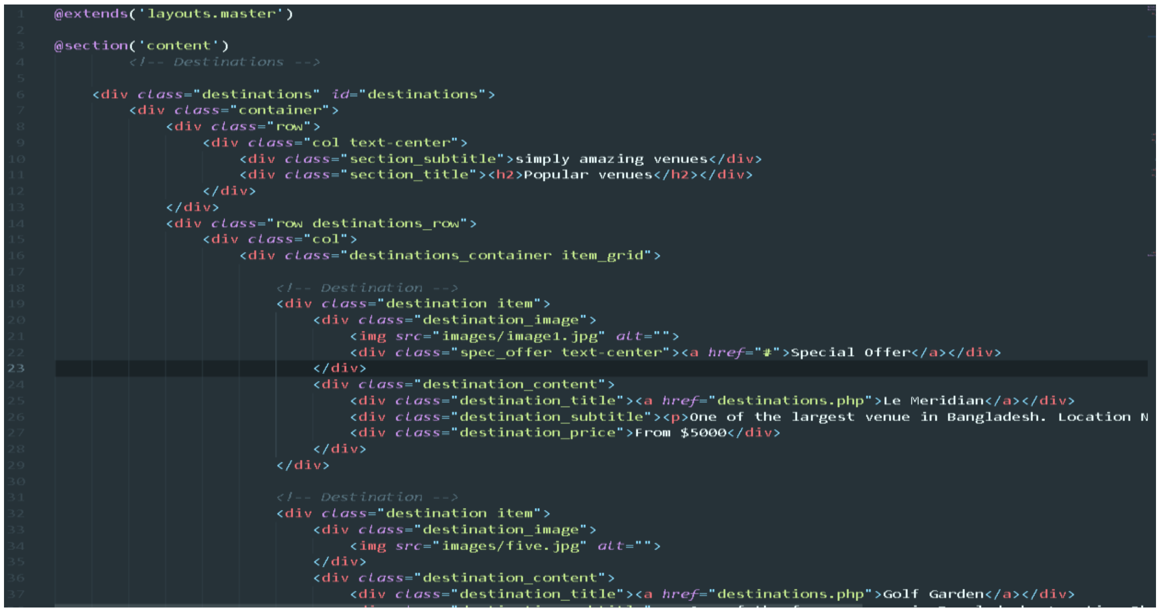
Firstly we inserted data in the search result. This action is done by user for searching venue. This figure 5.2.2 shows the code for search result in the system.



```
@extends('layouts.master')

@section('content')


<div class="destinations" id="destinations">
    <div class="container">
        <div class="row">
            <div class="col text-center">
            <div class="section_title"><h2>Search Result For "{{$q}}"</h2></div>
            </div>
        </div>
        <div class="row destinations_row">
            <div class="col">
                <div class="destinations_container item_grid">

                    @foreach ($venues as $venue)
                        <!-- Destination -->
                        <div class="destination item">
                            <div class="destination_image">
                            <img src="{{Voyager::image($venue->thumbnail)}}" alt="">
                                {{-- <div class="spec_offer text-center"><a href="#">Special Offer</a></div> --}}
                            </div>
                            <div class="destination_content">
                                <div class="destination_title"><a href="destinations.php">{{$venue->name}}</a></div>
                                {{-- <div class="destination_subtitle"><p>{{$venue->details}}</p></div> --}}
                                <div class="destination_subtitle"><p>{!! $venue->details !!}</p></div>
                                <div class="price-cta">
                                    <div class="destination_price" style="display:inline-block">From ${{$venue->price}}
                                    <div class="booking-button" style="display: inline-block;float:right">
                                    <a href="{{ route('booking.index') }}?venue_id={{ $venue->id }}&venue_name={{$venue
                                    </div>
                                </div>
                            </div>
                        </div>
                    @endforeach

```

Figure 5.2.2: Query for Search Result

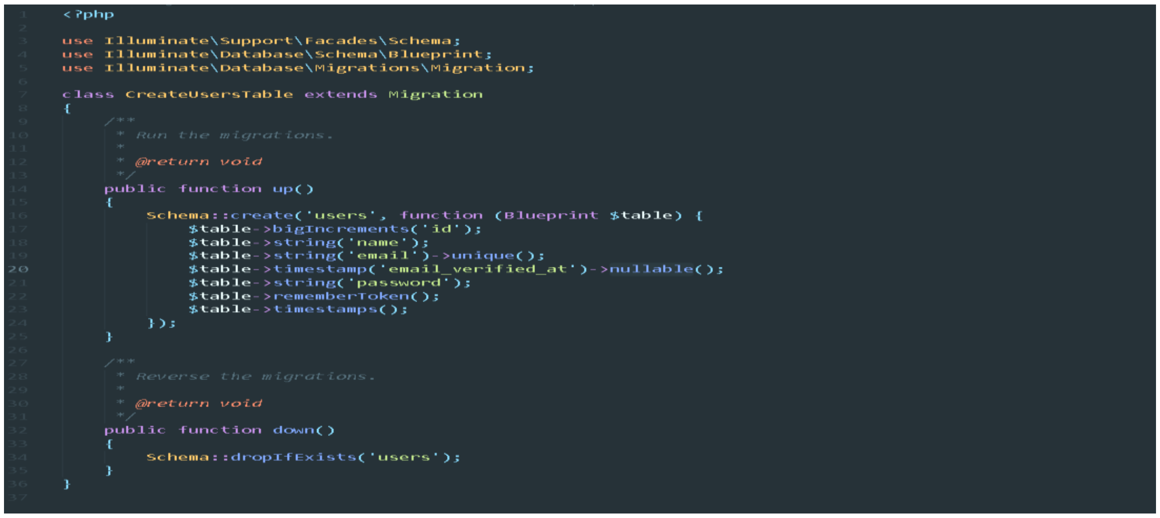
This figure 5.2.3 shows the query for managing home page.



```
@extends('layouts.master')

@section('content')
        <!-- Destinations -->

    <div class="destinations" id="destinations">
        <div class="container">
            <div class="row">
                <div class="col text-center">
                    <div class="section_subtitle">simply amazing venues</div>
                    <div class="section_title"><h2>Popular venues</h2></div>
                </div>
            </div>
            <div class="row destinations_row">
                <div class="col">
                    <div class="destinations_container item_grid">

                        <!-- Destination -->
                        <div class="destination item">
                            <div class="destination_image">
                                <img src="images/image1.jpg" alt="">
                                <div class="spec_offer text-center"><a href="#">Special Offer</a></div>
                            </div>
                            <div class="destination_content">
                                <div class="destination_title"><a href="destinations.php">Le Meridian</a></div>
                                <div class="destination_subtitle"><p>One of the largest venue in Bangladesh. Location N
                                <div class="destination_price">From $5000</div>
                            </div>
                        </div>

                        <!-- Destination -->
                        <div class="destination item">
                            <div class="destination_image">
                                <img src="images/five.jpg" alt="">
                            </div>
                            <div class="destination_content">
                                <div class="destination_title"><a href="destinations.php">Golf Garden</a></div>
```

Figure 5.2.3: Query for home page management

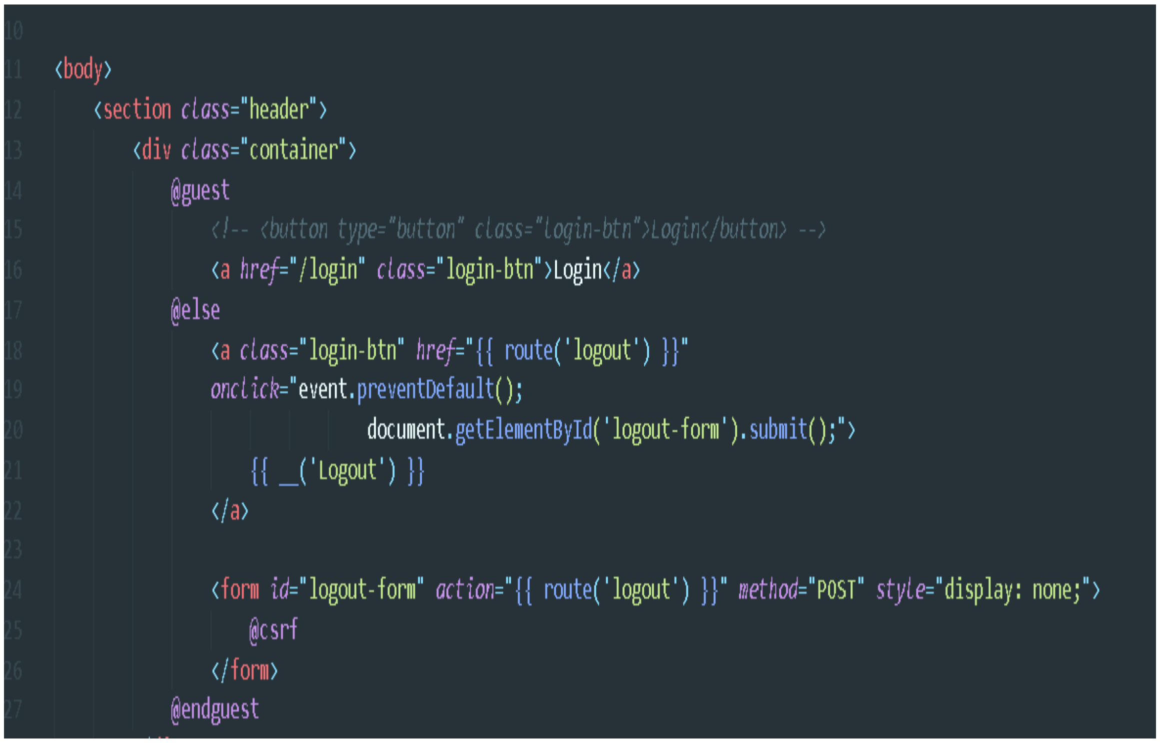
After that we have made the table for storing the registration details of our users. This figure 5.2.4 shows the query for user registration in the system.



```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Figure 5.2.4: Query for user registration

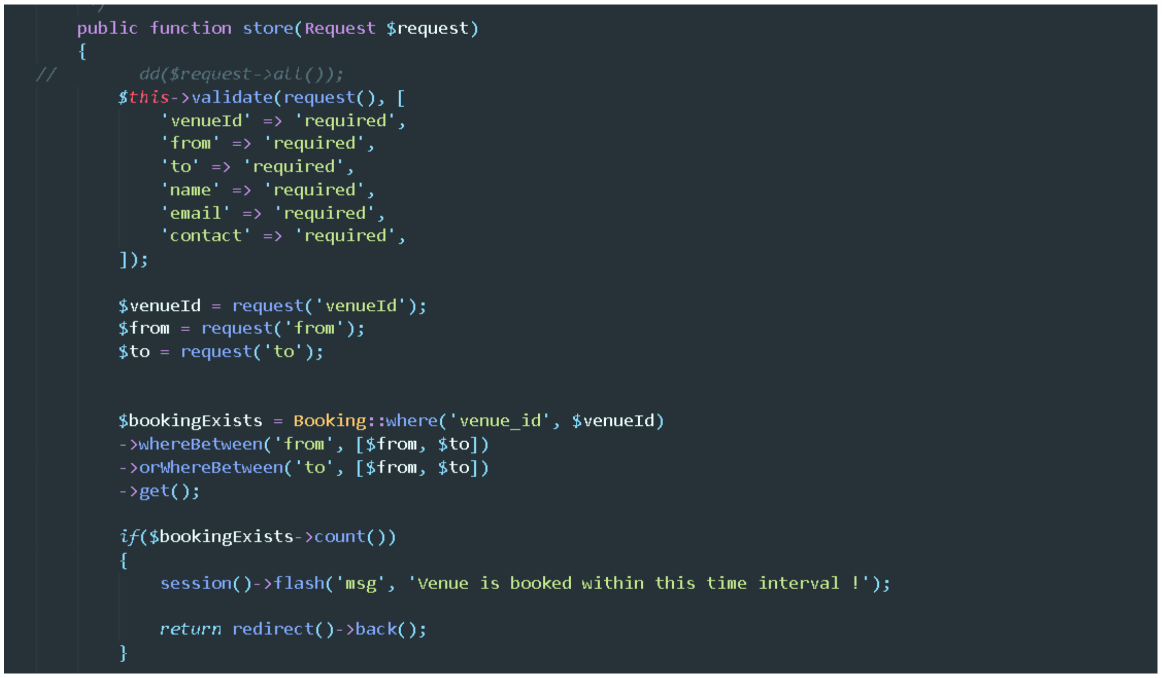
This figure 5.2.5 shows the query for users login into the system.



```
<body>
    <section class="header">
        <div class="container">
            @guest
                <!-- <button type="button" class="login-btn">Login</button> -->
                <a href="/login" class="login-btn">Login</a>
            @else
                <a class="login-btn" href="{{ route('logout') }}"
                onclick="event.preventDefault();
                                document.getElementById('logout-form').submit();">
                    {{ __('Logout') }}
                </a>

                <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
                    @csrf
                </form>
            @endguest
```

Figure 5.2.5: Query for users login

This figure 5.2.6 shows the query forbooking form into the system.



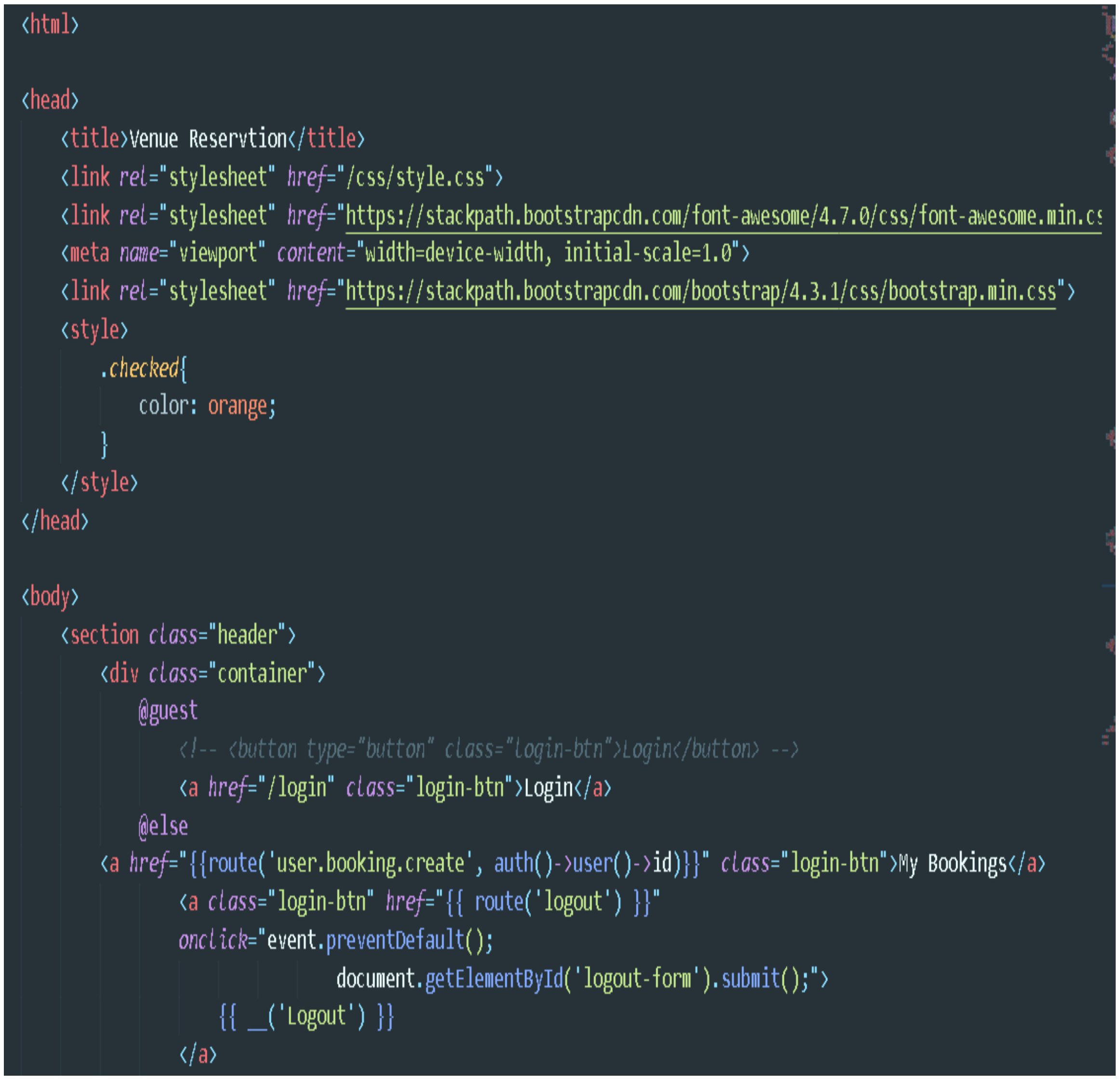
```
    public function store(Request $request)
    {
//          dd($request->all());
        $this->validate(request(), [
            'venueId' => 'required',
            'from' => 'required',
            'to' => 'required',
            'name' => 'required',
            'email' => 'required',
            'contact' => 'required',
        ]);

        $venueId = request('venueId');
        $from = request('from');
        $to = request('to');


        $bookingExists = Booking::where('venue_id', $venueId)
        ->whereBetween('from', [$from, $to])
        ->orWhereBetween('to', [$from, $to])
        ->get();

        if($bookingExists->count())
        {
            session()->flash('msg', 'Venue is booked within this time interval !');

            return redirect()->back();
        }
```

Figure 5.2.6: Query for booking form

5.3 Implementation of Front-end Design

Our utility is fully internet based. So there are copious workings with the front quit layout. Reason the user are watching the front give up and make their interchange with the utility. So for our internet based challenges we've got used html[8], CSS[7], bootstrap[5], jQuery[6] for the front quit accomplishment.

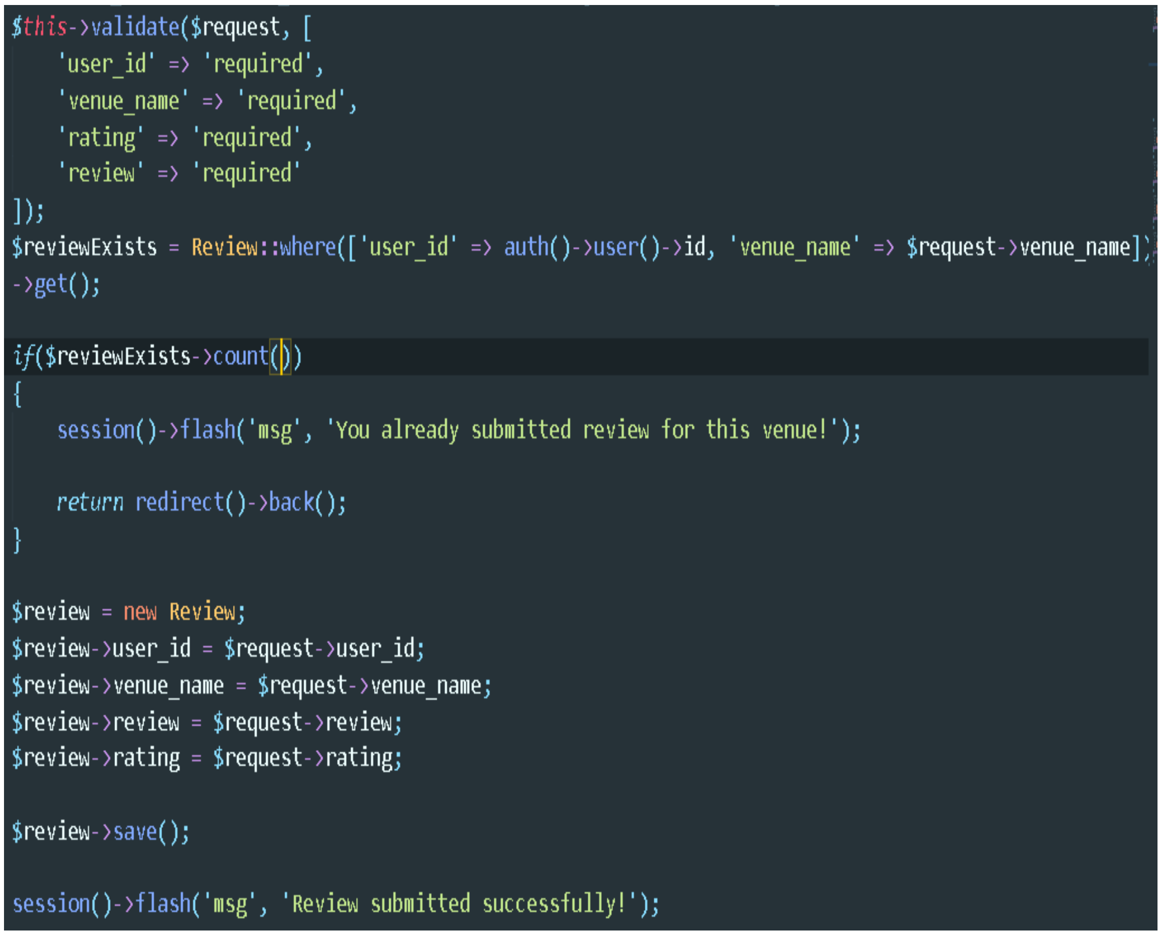
Let's take a stare how we boot our bootstrap and our initial of the web page. So, here the figure 5.3.1 shows the initialization of front end and the linkage to bootstrap.



```
<html>

<head>
    <title>Venue Reservtion</title>
    <link rel="stylesheet" href="/css/style.css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.cs
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <style>
        .checked{
            color: orange;
        }
    </style>
</head>

<body>
    <section class="header">
        <div class="container">
            @guest
                <!-- <button type="button" class="login-btn">Login</button> -->
                <a href="/login" class="login-btn">Login</a>
            @else
        <a href="{{route('user.booking.create', auth()->user()->id)}}" class="login-btn">My Bookings</a>
                <a class="login-btn" href="{{ route('logout') }}"
                onclick="event.preventDefault();
                                document.getElementById('logout-form').submit();">
                    {{ __('Logout') }}
                </a>
```

Figure 5.3.1: Initialization of front page

Then we added an review in our system. This figure 5.3.2 shows the review code.



```
$this->validate($request, [
    'user_id' => 'required',
    'venue_name' => 'required',
    'rating' => 'required',
    'review' => 'required'
]);
$reviewExists = Review::where(['user_id' => auth()->user()->id, 'venue_name' => $request->venue_name])
->get();

if($reviewExists->count())
{
    session()->flash('msg', 'You already submitted review for this venue!');

    return redirect()->back();
}

$review = new Review;
$review->user_id = $request->user_id;
$review->venue_name = $request->venue_name;
$review->review = $request->review;
$review->rating = $request->rating;

$review->save();

session()->flash('msg', 'Review submitted successfully!');
```

Figure 5.3.2: Code for review page

5.4 Testing Implementation

As this is a web based project that's why testing is very much needed for this system before set up the application. Because it will measure the overall performance and can get the knowledge that application input and output capacity. A test can deal the negation that how much time can police to run and can give careful pursuance. There are also so many testing that can be done for application. For this application we have done some testing and that testing are given below:

5.4.1 Unit Testing

Unit testing is used for when the coding is implementing then the developer checked for his own primary mistakes. So, in our case we have checked when we were coding and tried to find out the initial errors. It happens when we were coding and then test instantly.

5.4.2 Integration Testing

We have firstly done the front-end design then we have included all the back-end code. Then we have tartaned that if any kind of layout is broken or not. And also checked that all interface works perfectly or not.

5.4.3 Functional Testing

All the function that we have selected for our application initially then we have finished all the initial requirement of this project which need to be. And all the functions are tested individually. Because functions are the main important part for a system. If function does not work properly then the system will be insignificant.

5.4.4 System Testing

System testing measures that if the application is working on other system environment. As our application is fully web based so we have checked all the browsers and other operating system and also connected on many different devices .

5.4.5 Usability Testing

The usability testing measures the user dealings and how friendly is the UI. So, to do that we have testing on several user and get their experience on the UI and how much they can navigate that easily.

5.5 Test Results

We did several kinds of test only for some several purpose and after doing all the test there are some result. On the basis of these results we can get confirm about our system and it's advantage and disadvantage. The results are given below:

5.5.1 Result of Unit Testing

The full project is based on coding. There are thousands line of code what we did in our project. In this huge amount of line coding completely possible to do wrong or mistakes. For finding out this kind of mistakes we did unit testing. We find some errors and mistakes we got a second chance to fix them. We solved them immediately.

5.5.2 Result of Integration Testing

Integration testing was for a few trepidation are what code we wrote for front-end and back-end design, are those setting down of code working perfectly. Because both designs are too much important factor for any kind of project. Suppose we made a very good looking front-end design but there are several mistakes in back-end program and front-end and back-end design is not matching and in that time what will be said by users. So, to avoid that kind of problem we did some integration test. And there was no error found during that test.

5.5.3 Result of Functional Testing

Function testing was too much important. Because we did good front-end and back-end design but we did mistake in functional coding. So, what is the value of total coding. It is useless. Functional testing was the way to know that is every function working properly or not. One by one we checked every function. During this testing we got some bad behavior for logging and setting up profile page we face some problems. There was some logical error. And then we fixed them. Another front end site management there was a cache problem on the browser. This problem was overcome also by removing older files and inputting new one. Thus, we have completed our functional testing.

5.5.4 Result of System Testing

It was also very necessary to test because we know many of the people use windows operating system. So, we did system testing. And subsequently testing what we scrutinize is our application has successfully run on auxiliary systems and environment. So, there was no error found.

5.5.5 Result of Usability Testing

So this test has run on different type people. So that most of them get this system easily. But some of them cannot get the dashboard button is clickable. But after some minutes using they can found it and can use it without any aggravate.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and conclusion

We have attempted to parade all of the things and contents of our project. We have also make this report as a real-life software development like feel. Instantly we desire to share what we have enclosedby doing this project and what we novicefrom it.

By terminatingthis project we envisageimproper imparity of problems and acquire to admire many leadingpossessions. In our incitement we graboppressive idea about bootstrap framework. Because, at first , we started it using row html and CSS then we were inspired to do it in bootstrap framework what was totally new for us. Then we started to learn bootstrap and many things was happened at that learning time. We faced some problem what gave us lot of pain. We have some interesting experience also. By doing this project we have gained a lot of knowledge about PHP, jQuery, HTML, CSS and bootstrap.

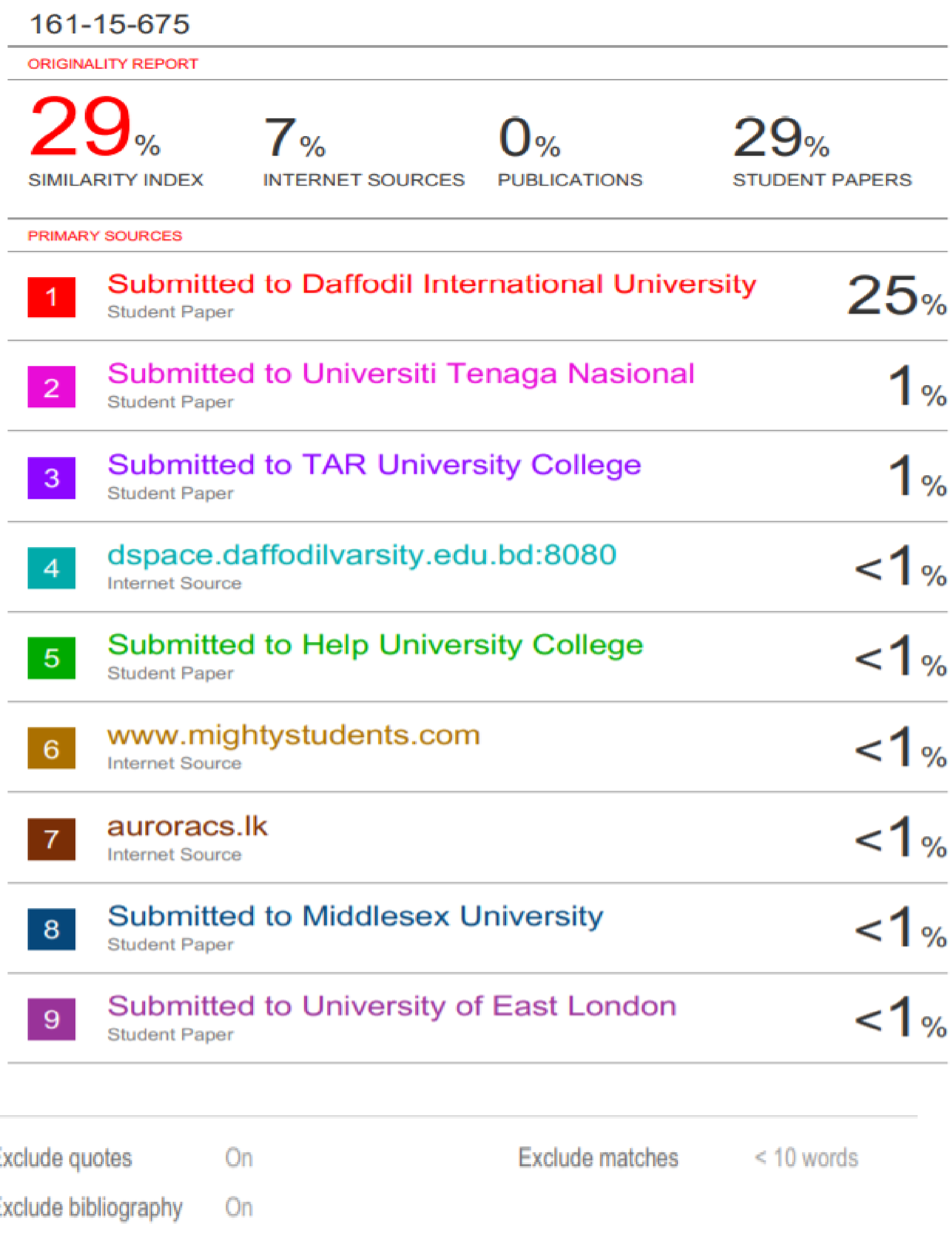
Ultimateineluctable thing is we have learnt lawfulinspiration programming. We also novicehow to prolixity programming knowledge into development sector. There was an important part and that was database design. We got a very nice idea about how to develop a database in consistency with the systems specification.

6.2 Scope for Further Development

As we have bounded quantity of time we could not entire our intent. We can specify part of intelligence in our system. We can particularize some significant offers for our prohibitive customers. We can specify some more options in our admin panel.

APPENDIX

This figure shows the plagiarism checking dispatch for the aggregate report:



161-15-675

ORIGINALITY REPORT

29%	7%	0%	29%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	25%
2	Submitted to Universiti Tenaga Nasional Student Paper	1%
3	Submitted to TAR University College Student Paper	1%
4	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
5	Submitted to Help University College Student Paper	<1%
6	www.mightystudents.com Internet Source	<1%
7	auroracs.lk Internet Source	<1%
8	Submitted to Middlesex University Student Paper	<1%
9	Submitted to University of East London Student Paper	<1%

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

Figure: Plagiarism checking report

REFERENCES

[1]Booking available at: https://www.booking.com[last accessed on 16-11-2019 at 2.00pm]

[2]Bookingbuddy available at: www.bookingbuddy.com [last accessed on 16-11-2019 at 3.00pm]

[3] Venuebookingz available at: www.venuebookingz.com [last accessed on 16-11-2019 at 5.00pm]

[4]Xampp server available at: https://www.apachefriends.org[last accessed on 18-11-2019 at 9.30pm]

[5] Bootstrap available at; https://getbootstrap.com/ [last accessed on 18-11-2019 at 10.00pm]

[6] jQuery available at: https://stackoverflow.com [last accessed on 20-11-2019 at 7.30pm]

[7] CSS available at: https://www.w3schools.com/css [last accessed on 20-11-2019 at 9.00pm]

[8] HTML available at: https://www.w3schools.com/html [last accessed on 22-11-2019 at10.20pm]