DREAM VACATION

Project Report Submitted By

NAMITHA T. S

Reg. No: AJC19MCA014

In Partial fulfillment for the Award of the Degree Of

MASTER OF COMPUTER APPLICATIONS
(MCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2019-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, "**DREAM VACATION"** is the bonafide work of **NAMITHA T. S (Reg.No:AJC19MCA014)** in partial fulfillment of the requirements for the award of the Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2019-22.

Ms. Meera Rose Mathew
Internal Guide

Mr. Binumon Joseph
coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

External Examinar

DECLARATION

I hereby declare that the project report **"DREAM VACATION"** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2019-2022.

Date: **NAMITHA T.S**

KANJIRAPPALLY **Reg. No: AJC19MCA014**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mr. Binumon Joseph** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide**, Ms. Meera Rose Mathew** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

NAMITHA T.S

ABSTRACT

The Tourism industry has an immense growth in past few years. Technology has made life easier for the customers and also for the businesses. Customers enjoy availing various services on one clickwhile businesses enjoy growth and popularity they get due to the internet. Online resort booking system is the best platform, as it adds many benefits and you can easily book the resort or room through the internet. It discovers the more information about a resort which is situated in a particulararea and you can also select a resort according to your demands and choice. A resort reservation system enables guests to schedule dates and length of stay, room selection, extras, and payment all in one place. We can't ignore the fact that booking a resort through the agent is the time-consuming process. Customers can make their bookings via computers or smartphones while taking care of theironline security in order to protect their privacy. This service is also very beneficial for the last minutetraveling plans. Online booking for resort rooms require the details about the pricing of each room. The prices of rooms may vary due to the certain factors like the view, size of the room, the interior of the room and many other factors. This service provides the excellence whole word option to the customers. Online booking services must be highly user friendly and informative. The customers must take good care of their online security in order to protect their privacy and financial details. The proposed system is a website in which user can book good and affordable resorts as well as individualscan manage their resort efficiently. We will also provide users to give customization comments to customize their resort features they can view the booking resort details, payment details etc.

CONTENT

List of Abbreviations

MVC	-	Model View Controller
IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

"DREAM VACATION" is a online web portal for atomization of resort booking system and to create a scope for visiting tourists from different geographic locations. For every resort there will be different branches at different location so managing all these resorts information will be easy through this online application. This applicationwill help to improve services for tourists and also improve revenue source for resorts management. In existing system manual work is used where tourists should contact through phone and know details about availability of rooms and locations to view at that area. As the usage of internet had increased developing web portal will be helpful and atomized

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can book good and affordable resorts as well as individuals can manage their resort efficiently. We will also provide users to give customization comments to customize their resort features, they can view the booking resort details, payment details etc. The system includes 3 modules and 3 users, They are:

- Admin
- Users
- Payment Gateway

Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin must have a login in to this system. They can approve resort booking, view complaints, give responses, approve resort, room details provided by manager. Login/Registration ,Add room, Delete room, Details adding, Payment details etc…

Customer Module

Customer can reserve and they can choose their rooms and details and do secure online payment. Customer can also find resorts nearby them and can add feedbacks and complaints to them. Customer can perform functionalities like,Registration/Login and view/manage profile/change password,View resorts, searches based on availability, resort name, cost, room type etc. Sort resorts by discount, price: high to low/low to highetc. Add customization comments based on the requirement and post feedbacks as publicaly. View/Download booking summary and recipite,Add Review, Rating and PostComplaints for the booked resorts.Payment with Paytm Gateway.

Payment Gateway

Laravel Stripe has built a reputation of being easy-to-use for both travelers and room owners. Because of great compatibility with Laravel, Stripe has become the go-to payment gateway for booking system. With a host of features and simple Stripe payment gateway integration in Laravel, many development agencies have opted for Stripe as the default payment gateway option for all online booking, they develop for their clients.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. Thissystem is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Customer can register and they can place their room. Each customer can create their own profile .The proposed system rectify the drawbacks of the present system. It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can book resorts by viewing the profile details, achievements etc.

DRAWBACKS OF EXISTING SYSTEM

- No publicity for the resort details
- One way and face-to-face communication
- Human effort is needed.
- Customers do not get a chance of identify good resorts.

2.3 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth. On such consideration the system is proposed. In our proposed system there is an admin who can view and manage all the booked people and customers. It allowscustomers to make their booking and do their transactions by using online payment method .Users of this proposed system are admin, customer. The aim of proposed systemis to develop a system of improved facilities. The system provides propersecurity and reduces the manual work. Online booking advancements have been so drastic that it has evolved to be a part of our life. Today customer doesn't drive down to some bookings fora room but preferably check over the internet for bill, offers, reviews and book online. The existing system working procedure is like customers create their booking and find their customers via Facebook, Instagram or through any other social networking sites. Similarly customers do not get a chance to find good rooms or resorts from the existing method.

2.4 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost allconfigurations. It has got following features:

Better security:

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we areusing the secured databases for maintaining the documents.

Ensure data accuracy:

The proposed system eliminates the manual errors whileentering the details of the users during the registration.

Better service:

The resorts can mention their services so we can easy to understand and never any complains based on the services. The everyday updation can help to latest offer and discounts easily.

ChatBox system:

tawk.to is a live chat software designed to help businesses communicate with clients and website visitors to deliver customer support. Key features include canned responses, offline forms, website visitor tracking, customizable branding, file sharing, screen sharing and chat history.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development. The cost of project, DREAM VACATION was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project wasdeveloped in a low cost, as it is completely developed using open source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP framework LARAVEL in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using LARAVEL in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 8GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.DREAM VACATION, GUI is simple so that users can easily use it. DREAM VACATION is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 8 GB

Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML,CSS

Backend - MYSQL

Client on PC - Windows 10

Technologies used - JS, HTML5, AJAX, J Query, CSS,SCSS

3.3 SOFTWARE DESCRIPTION

3.3.1 LARAVEL

Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which helps in creating a web application. The web application thus designed is more structured and pragmatic.

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails. Laravel has a very rich set of features which will boost the speed of web development.

If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks.Composer is a tool which includes all the dependencies and libraries. It allows a user to create a project with respect to the mentioned framework.

The second step after we learned about installation is to learn basic concept of MVC. MVC stands for Model, View and Controller. As a beginner you need to understand how request/response cycle works.

Features Of LARAVEL:

- Routing controllers.
- Configuration management.
- Testability.
- Authentication and authorization of users.
- Modularity.
- ORM (Object Relational Mapper) features.
- Provides a template engine.
- Building schemas.
- E-mailing facilities.

The Laravel Framework follows **MVC architecture**. MVC is an architectural design pattern that helps to develop web applications faster.

MVC stands for **Model-View-Controller**.

- **Model (M)**–A model handles data used by the web application.
- **View (V)**–A view helps to display data to the user.
- **Controller (C)**–A controller interacts with the model to create data for the view.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

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

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**
  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**
  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm thatis used. The system design develops the architectural detail required to build a system orproduct. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. Forexample, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UMLdiagrams. UML has a direct relation with object oriented analysis and design. After

some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation forth modeling of real-world objects and systems.

System objectives can include planning overall requirements, validatinga hardware design, testing and debugging a software product under development, creatingan online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include itemordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

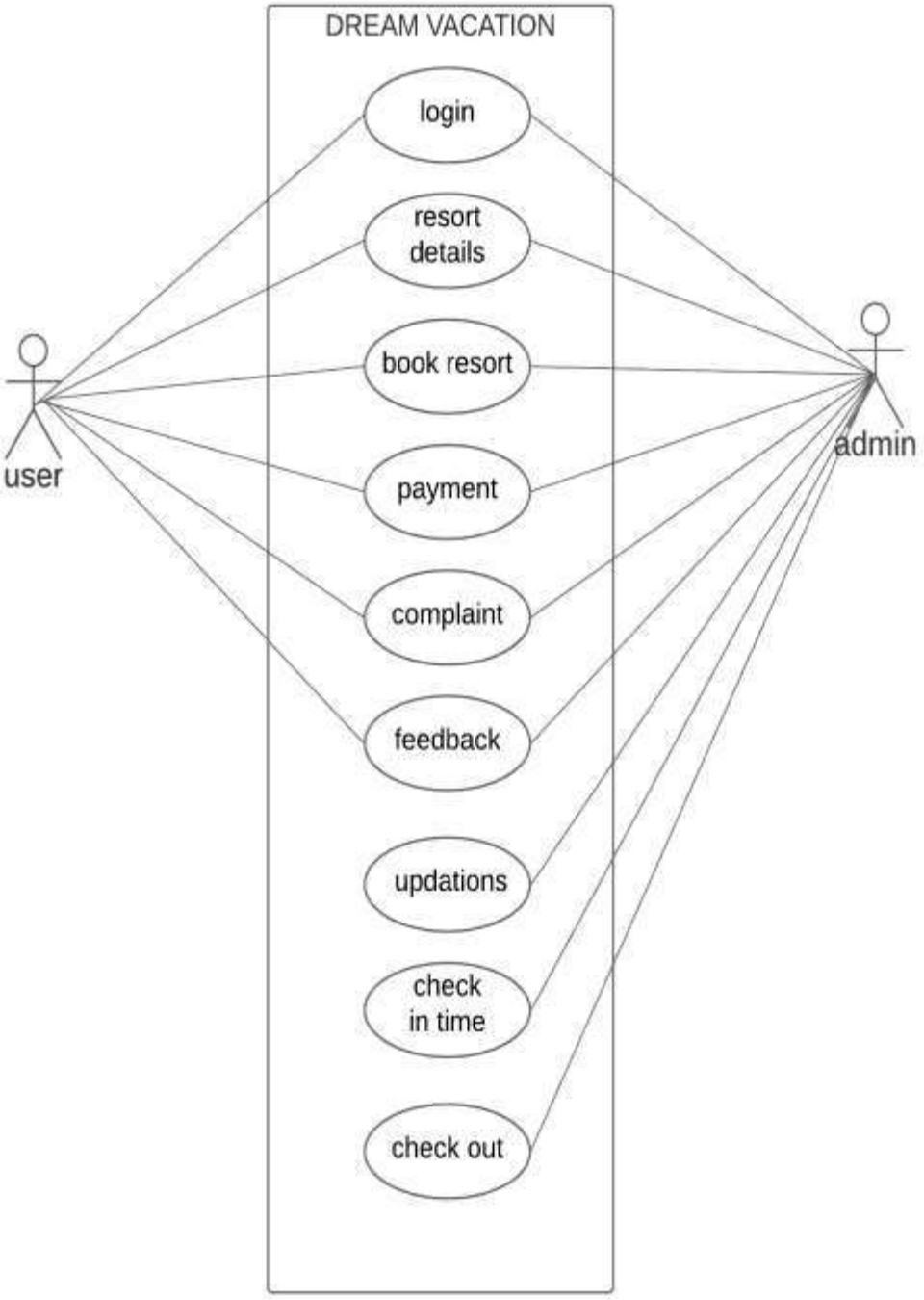
- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their

roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

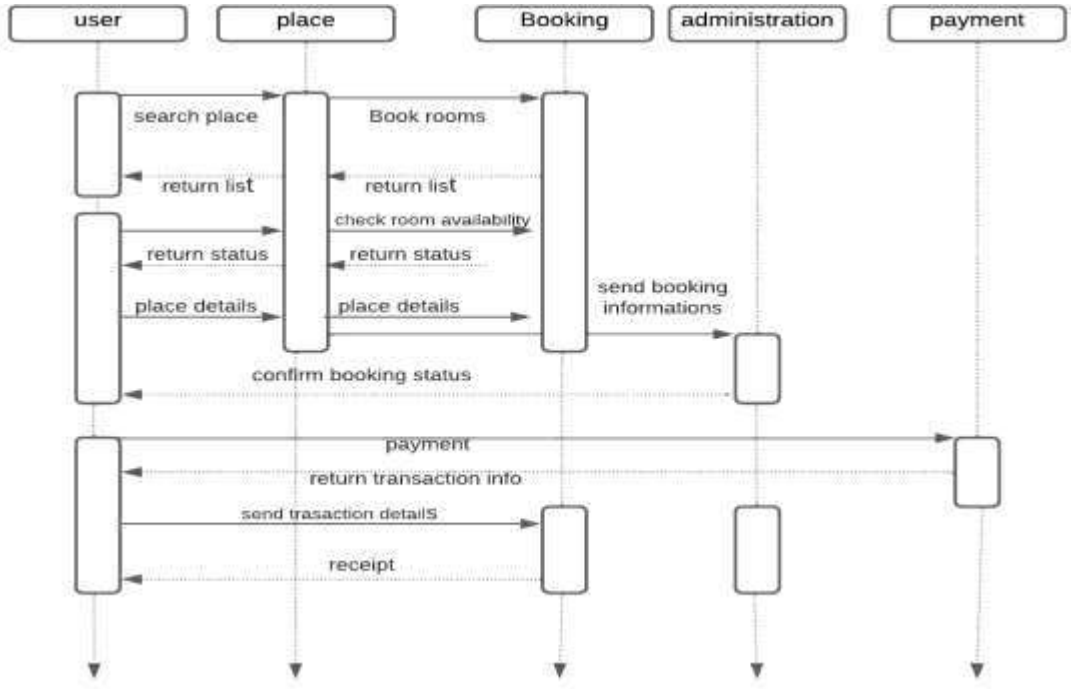

DREAM VACATION
login
resort details
book resort
payment
complaint
feedback
updations
check in time
check out
user
admin

4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

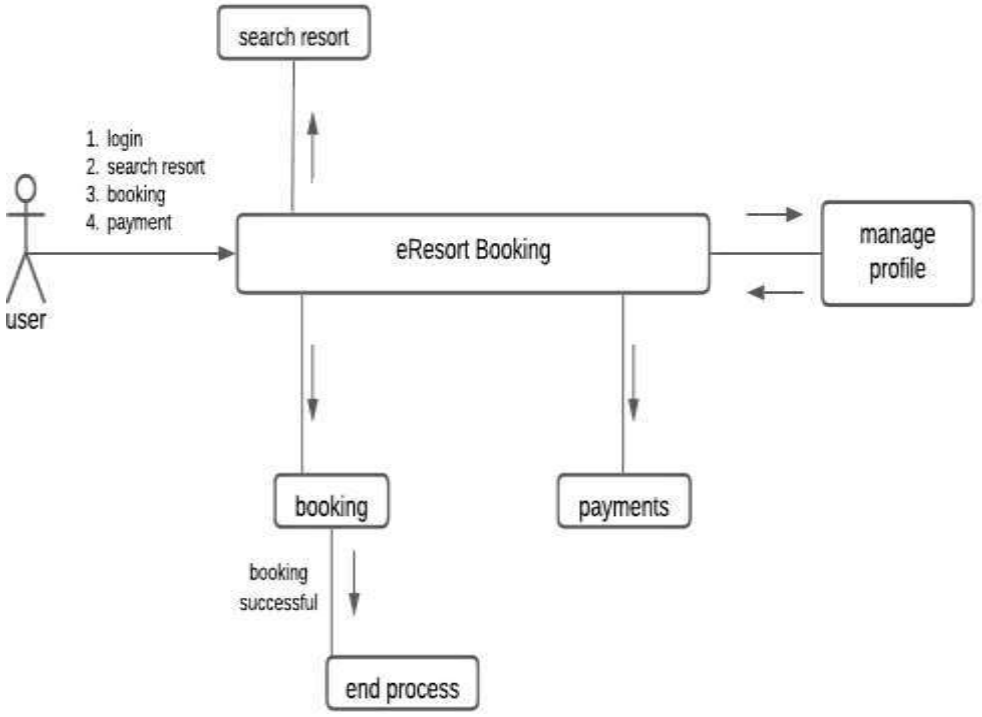
Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.



4.2.3 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship betweenthe objects in a system. Both the sequence and the collaboration diagrams represent thesame information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object- oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecturein the system.

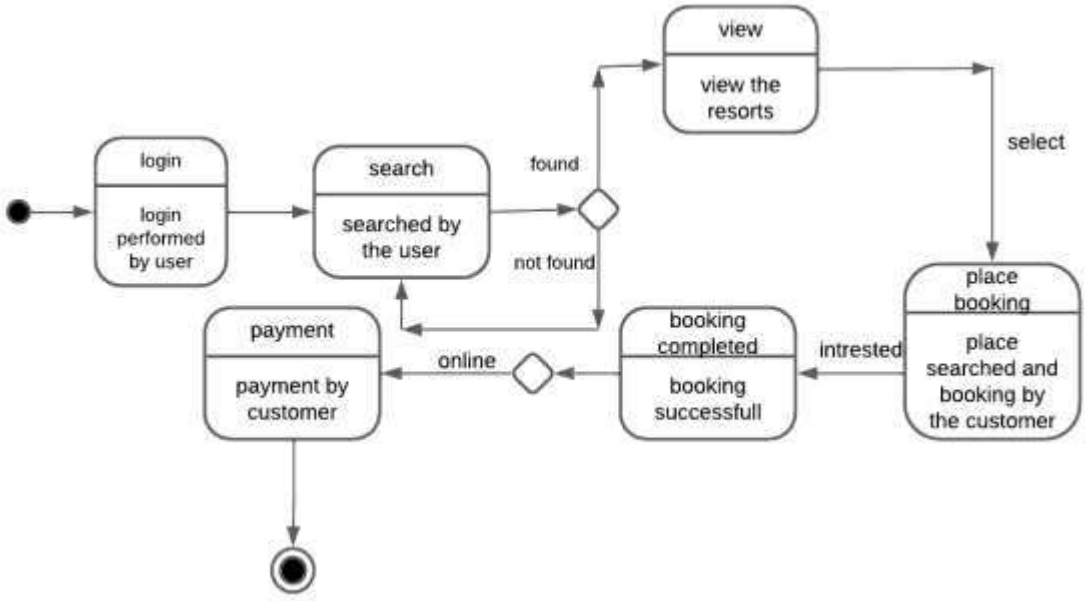


4.2.4 STATE CHART DIAGRAM

It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as

a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system. Following are the main purposes of using Statechart diagrams –

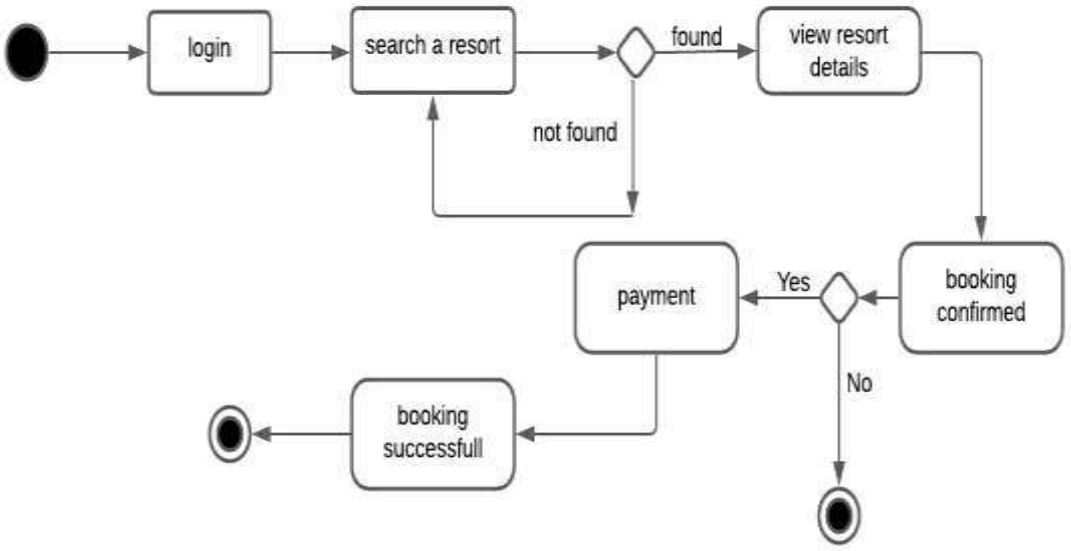
- To model the dynamic aspect of a system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object



4.2.5 ACTIVITY DIAGRAM

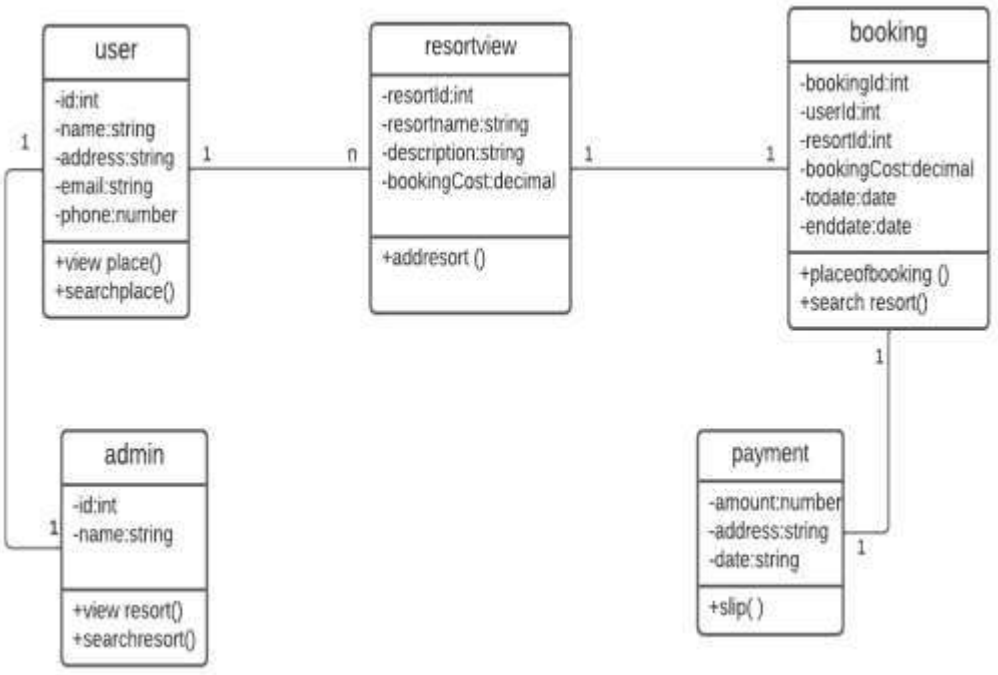
Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or

concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to draw the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart.



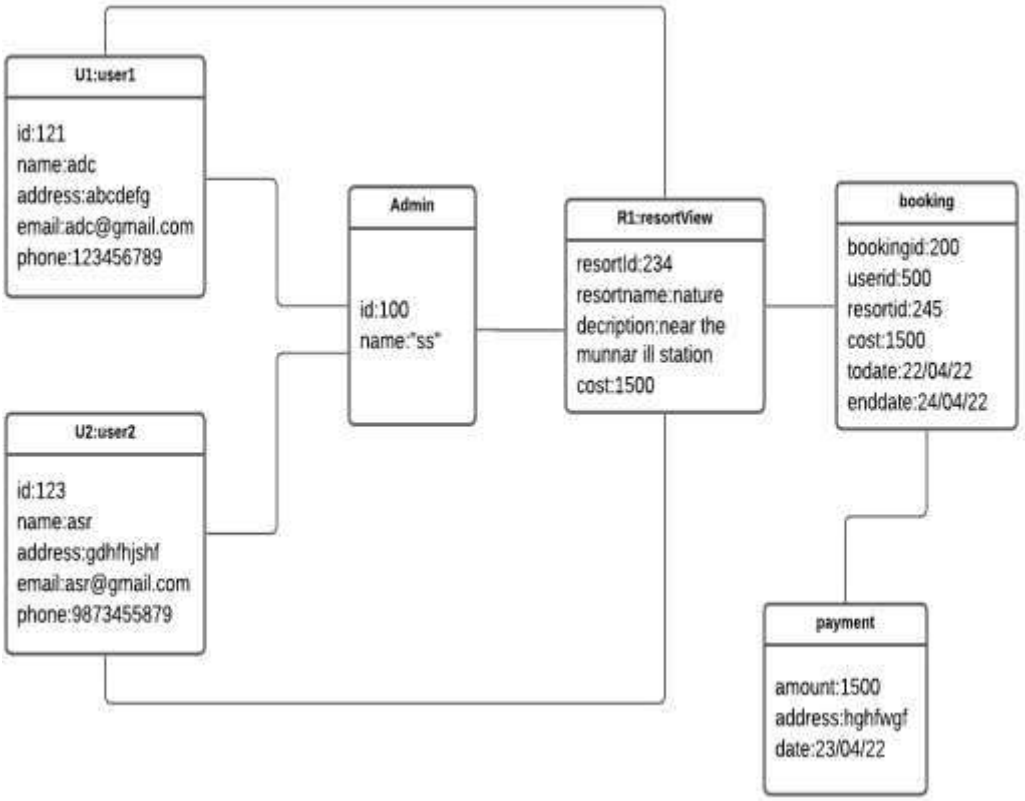
4.2.6 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



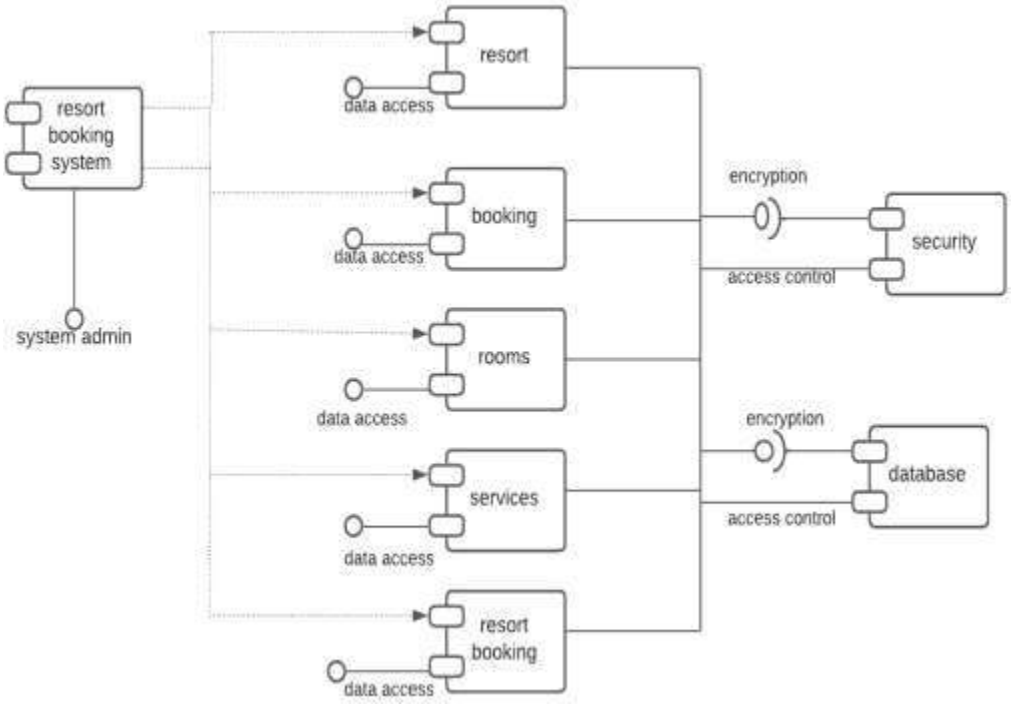
4.2.7 OBJECT DIAGRAM

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.



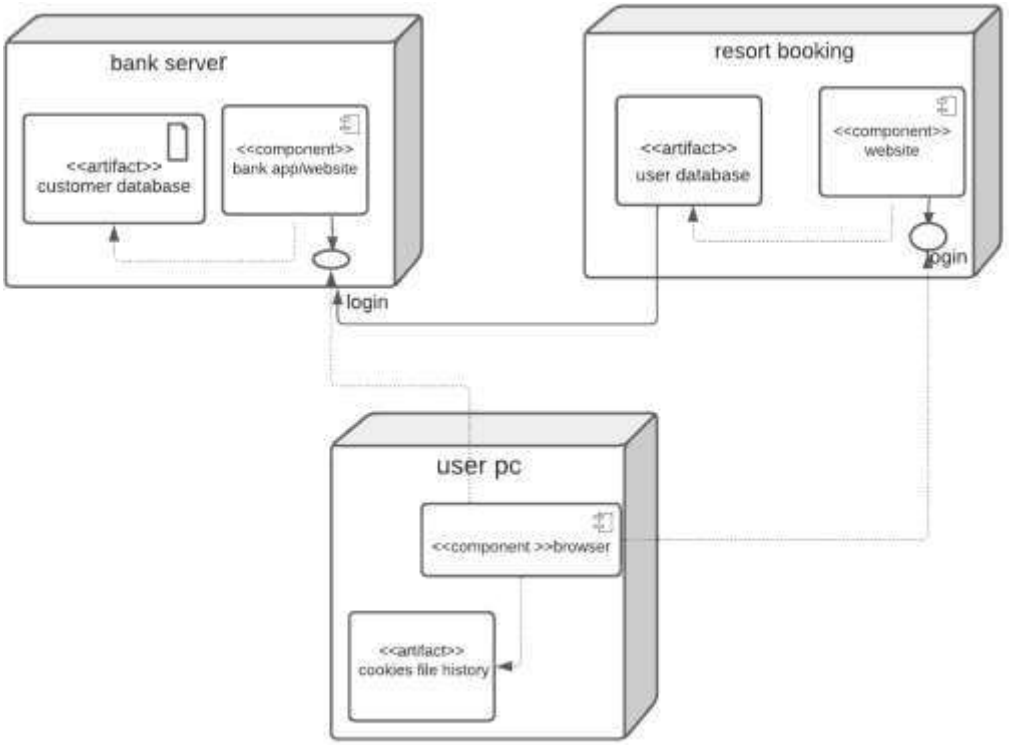
4.2.8 COMPONENT DIAGRAM

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used tovisualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be desribed as a staticimplementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannotrepresent the entire system but a collection of diagrams is used to represent the whole.



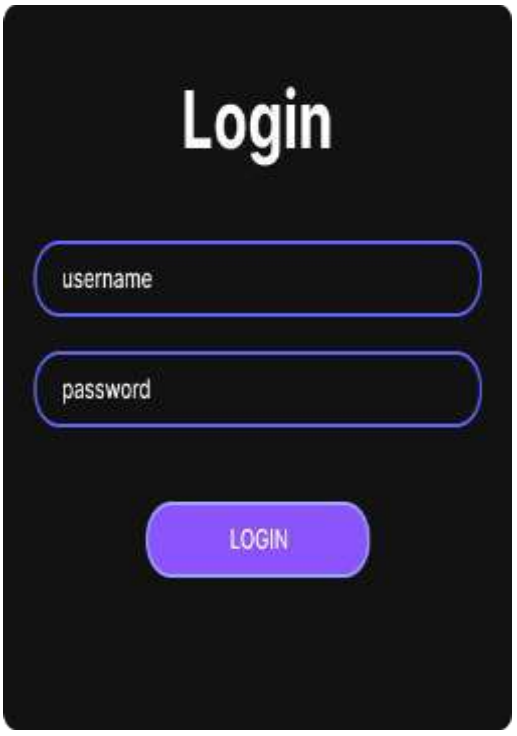
4.2.9 DEPLOYMENT DIAGRAM

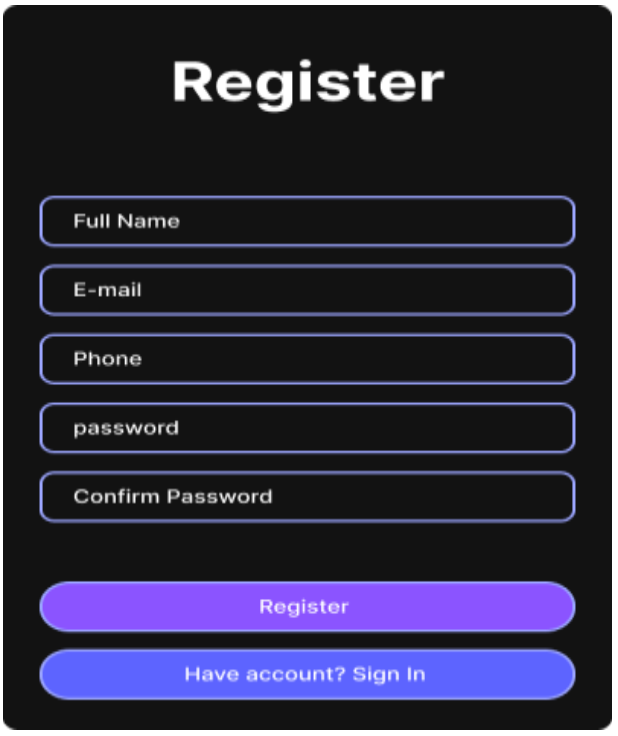
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.


bank server
<<artifact>>
customer database
<<component>>
bank app/website
login
resort booking
<<artifact>>
user database
<<component>>
website
login
user pc
<<component >>browser
<<artifact>>
cookies file history

4.3 USER INTERFACE DESIGN USING FIGMA

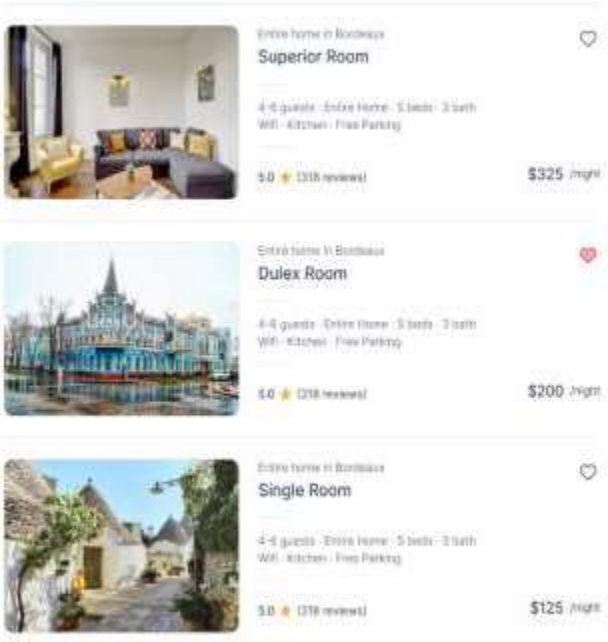
4.3.1-LOGIN/REGISTRATION





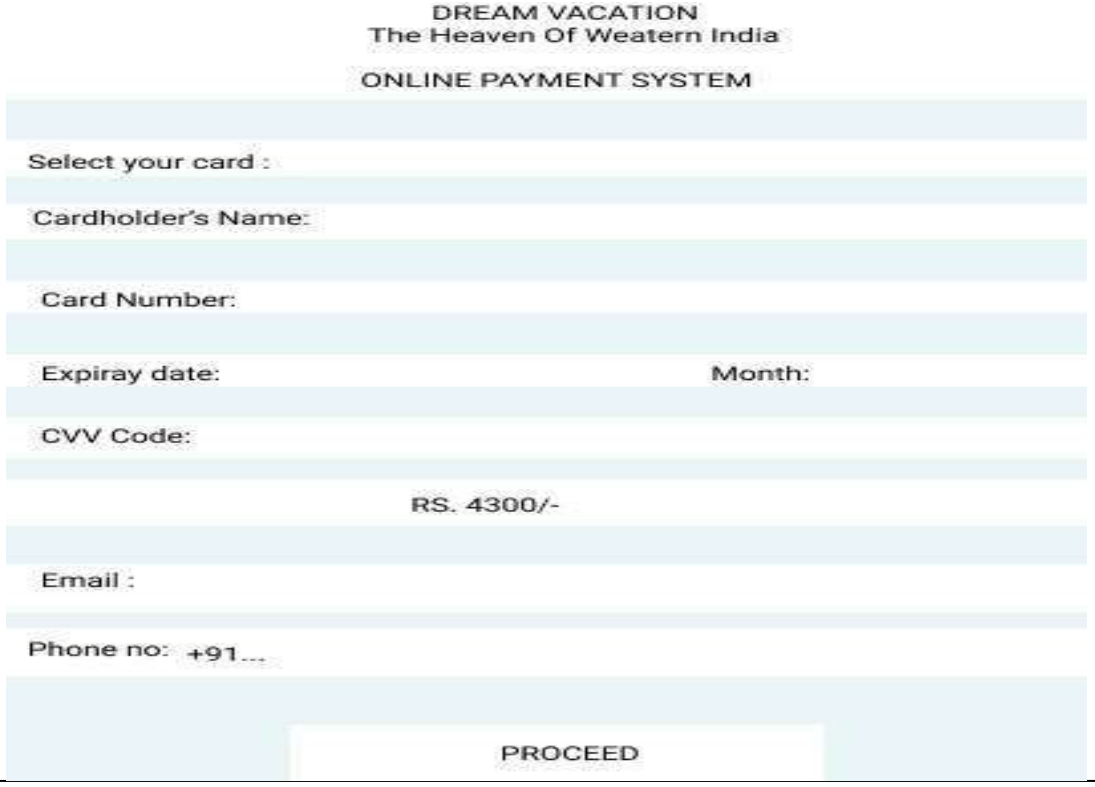
:

4.3.3 USER VIEW



4.3.4 RESERVATION

4.3.5 PAYMENT



4.4 TABLE DESIGN

4.4.1 LOGIN

Primary key: login_id

SL NO	Field Name	Data Type	Constraints	size	Description
1	lid	varchar	Primary key	30	Login id
2	Password	Varchar	Not null	8	Password

4.4.2 REGISTRATION

Primary key: Registration id, Login id

SL NO	Field name	Data type	constraints	Size	Description
1	id	Varchar	Primary key	30	Registration id
2	name	Varchar	Not null	30	Name
3	lid	Varchar	Primary key	10	Login id
4	Mail	Varchar	Not null	30	email

4.4.3 BOOKING DETAILS

Primary key:booking_id
Foreign key:resort_id,lid

SL NO	Field name	Data type	constraint	size	Description
1	booking_ id	varchar	Primary key	30	Room Booking id
2	resort_id	Varchar	Foreign key	30	Resort id
3	lid	varchar	Foreign key	30	User login id
4	acnonac	Varchar	Not null	10	AC or Non AC
5	name	Varchar	Not null	30	Name
6	address	Varchar	Not null	30	Address
7	gender	Varchar	Not null	10	Gender
8	email	Varchar	Not null	30	Email
9	phone	Int	Not null	10	Phone
10	Native _ place	Varchar	Not null	30	Native place
11	checkin	Date	Not null	10	Check IN
12	checkout	Date	Not null	10	Check OUT
13	resort _type	varchar	Not null	30	Resort type

4.4.4 RESORT DETAILS

Primary key:resort_id
Foreign key:room_id

Sl no	Field name	Data type	constraint	size	description
1	resort_ id	varchar	Primary key	30	Resort id
2	room _id	varchar	Foreign key	30	Room id
3	resort _name	varchar	Not null	30	Resort name
4	Rate	varchar	Not null	30	Rate
5	facilities	varchar	Not null	30	Facilities of resort
6	features	varchar	Not null	30	Features of resort
7	Image	file	Not null		Image of resort

4.4.5 ADMIN

Primary key:manager_id

Sl no	Field	Data type	constraint	size	description
1	mngid	varchar	Primary key	30	Manager id
2	Name	varchar	Not null	30	Name
3	address	varchar	Not null	30	Address
4	place	varchar	Not null	30	Place
5	pin	int	Not null	10	Pin number
6	rdetails	varchar	Not null	30	Room details
7	rname	varchar	Not null	30	Resort name
8	rplace	varchar	Not null	30	Resort place
9	checkin	date	Not null	10	Check IN
10	chechkout	date	Not null	10	Check OUT

4.4.6 PAYMENT

Primary key: payment id

SL NO	Field name	Data type	constraint	size	Description
1	Pay _id	int	Primary key	30	Payment id
2	amount	Int	Not null	10	Total amount
3	Name _card	Varchar	Not null	30	Name in card
4	Card _num	Varchar	Not null	30	Card number
5	Exp _date	date	Not null	10	Expirer date
6	cvv	Number	Not null	10	CVV
7	Card_name	varchar	Not null	30	Card Name

4.5 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.The database design is a twolevel process. In the first step, user requirements are gathered together anda database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is calledPhysical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in thedatabase is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.5.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigneda unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.Every value in a relation is atomic, that is not decomposable.

4.5.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one.It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records fromthe same table. A foreign key is a column in a table that uniquely identifies record from adifferent table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of theprimary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal formif and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this wedecompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that doesnot depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

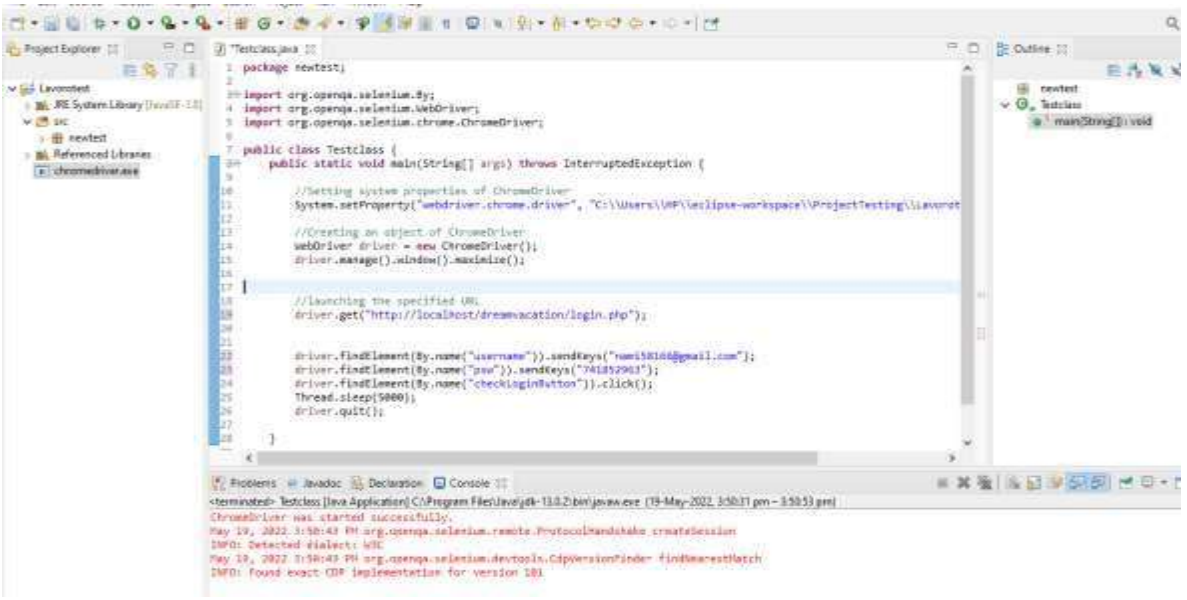
Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Test Case 1

Project Name: Dream Vacation					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By:Namitha T S**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 19-05-2022		
Module Name: Login Screen			**Test Executed By : Ms.Meera Rose Mathew**		
Test Title : Verify login with validemail and password			**Test Execution Date:** 19-05-2022		
Description: Test the Login Page					
Pre-Condition : User has valid email id and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: nami58166@gmail.com	User should beable to Login	User Logged inand navigated to Subadmin Dashboard with records	Pass
3	Provide Valid Password	Password: 741852963			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: user@gmail.Com Password: User1234	User shouldnot be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

CODE

```
packagenewtest;
import org.openqa.selenium.By;
importorg.openqa.selenium.WebDriver;
importorg.openqa.selenium.chrome.ChromeDriver;
publicclassTestclass{
publicstaticvoidmain(String[]args)throwsInterruptedException{

//SettingsystempropertiesofChromeDriver
System.setProperty("webdriver.chrome.driver","C:\\Users\\Lenovo\\ecl
ipse-workspace\\ProjectTesting\\Lavorotest\\chromedriver.exe");

//CreatinganobjectofChromeDriver
WebDriverdriver=newChromeDriver();
driver.manage().window().maximize();
//launchingthespecifiedURL
driver.get("http://localhost/dreamvacation/login.blade.php");
driver.findElement(By.name("username")).sendKeys("nami58166@gmail.com");
driver.findElement(By.name("psw")).sendKeys("741852963");
driver.findElement(By.name("Button")).click();
Thread.sleep(5000);
driver.quit();}}
```



Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite.

Test Case 2

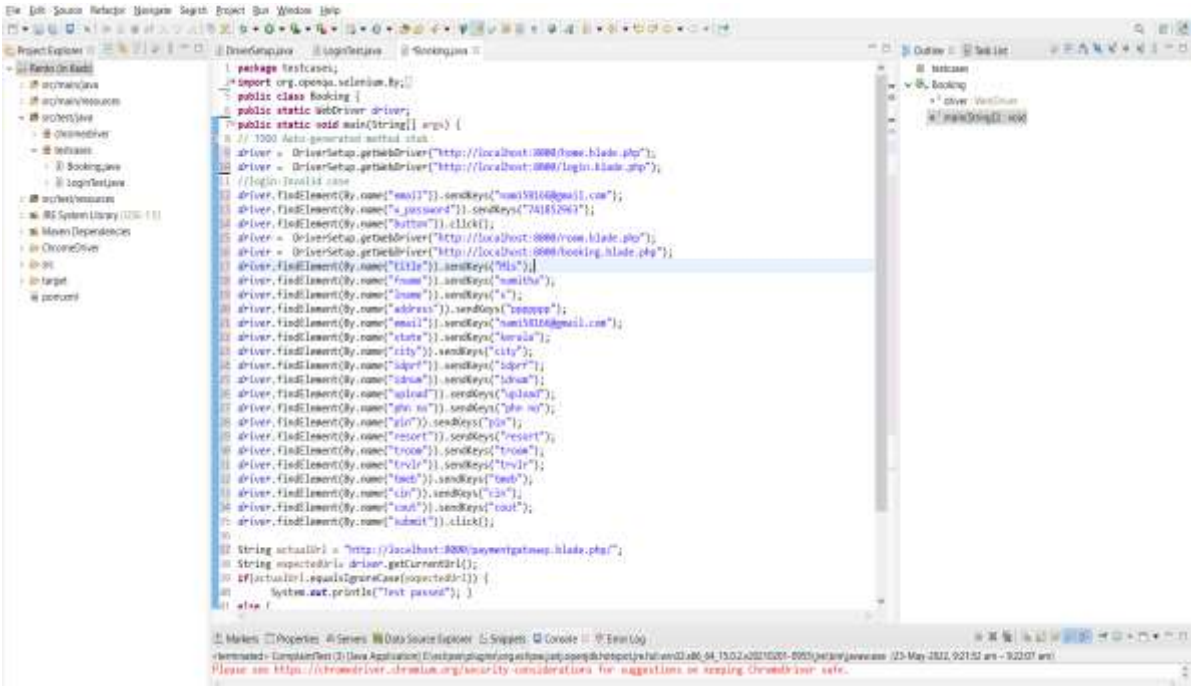
Project Name: Dream Vacation					
Booking Test Case					
Test Case ID: Fun_1			**Test Designed By:Namitha T S**		
Test Priority(Low/Medium/High): High			**Test Designed Date:** 19-05-2022		
Module Name: Booking Screen			**Test Executed By : Ms.Meera Rose Mathew**		
Test Title : Verify room booking details			**Test Execution Date:** 19-05-2022		
Description: Test the booking Page					
Pre-Condition :Must be valid user					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status(Pass/Fail)**
1	Navigation toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: nami58166@gmail.com	User should beable	User Logged inand navigated to Subadmin	Pass

3	Provide Valid Password	Password: 741852963	to Login	Dashboard with records	
4	Click on Sign In button				
5	Click on Book now button	Fname:namitha Lname:s Email:nami58166@gmail.com Address:pppp State:kerala Pin:686573 Resort:sunshin Idnum:1254678936 Idprf:PAN Troom:single Tmeb:1 Trlvr:solo Cin:19/5/22 Cout:20/5/22	User shouldnot be able to Book	User for Room Booking	Pass
6	Provide Null phone number	Phone no: Enter phone no:			
7	Click on Submit button				
Post-Condition: User is validated with database and successfully login into account.The Account session details are logged in database					

Roombook.java

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;
public class Booking {
public static WebDriver driver;
public static void main(String[] args) {
// TODO Auto-generated method stub
driver =  DriverSetup.getWebDriver("http://localhost:8000/home.blade.php");
driver =  DriverSetup.getWebDriver("http://localhost:8000/login.blade.php");
```

```
driver.findElement(By.name("email")).sendKeys("nami58166@gmail.com");
driver.findElement(By.name("u_password")).sendKeys("741852963");
driver.findElement(By.name("button")).click();
driver =  DriverSetup.getWebDriver("http://localhost:8000/room.blade.php");
driver =  DriverSetup.getWebDriver("http://localhost:8000/booking.blade.php");
driver.findElement(By.name("title")).sendKeys("Mis");
driver.findElement(By.name("fname")).sendKeys("namitha");
driver.findElement(By.name("lname")).sendKeys("s");
driver.findElement(By.name("address")).sendKeys("ppppppp");
driver.findElement(By.name("email")).sendKeys("nami58166@gmail.com");
driver.findElement(By.name("state")).sendKeys("kerala");
driver.findElement(By.name("city")).sendKeys("city");
driver.findElement(By.name("idprf")).sendKeys("idprf");
driver.findElement(By.name("idnum")).sendKeys("idnum");
driver.findElement(By.name("upload")).sendKeys("upload");
driver.findElement(By.name("phn no")).sendKeys("phn no");
driver.findElement(By.name("pin")).sendKeys("pin");
driver.findElement(By.name("resort")).sendKeys("resort");
driver.findElement(By.name("troom")).sendKeys("troom");
driver.findElement(By.name("trvlr")).sendKeys("trvlr");
driver.findElement(By.name("tmeb")).sendKeys("tmeb");
driver.findElement(By.name("cin")).sendKeys("cin");
driver.findElement(By.name("cout")).sendKeys("cout");
driver.findElement(By.name("submit")).click();
String actualUrl = "http://localhost:8000/paymentgateway.blade.php/";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
 System.out.println("Test passed"); }
else {
System.out.println("Test failed"); }
driver.quit();
}}
```



5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.Black Box testing method focuses on the functional requirements of the software. That is,Black Box testing enables the software engineer to derive sets of input conditions that willfully exercise all functional requirements for a program.Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need.The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays avital role in the system testing. After preparing the test data, the system under study is testedusing that test data. While testing the system by which test data errors are again uncoveredand corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is theprocess of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will bethe system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to

ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In this digital era, people like to have everything in one click. Dream Vacation aims to convert existing manual reservation system into hassle free online reservation system. This project is user friendly in all perspectives. Both, admin and customers can easily access the data. It also provide payment gateway to complete booking and also an instant chatbox service provided to solve customer queries. .All these services make dream vacation one of the best in industry. Online booking services must be highly user friendly and informative.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the more payment should be done in online mode.
- Customers can able to do advanced search options
- Customers can able to view resort owners details and reserve the rooms.
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- https://app.diagrams laravel
- https://learn2torials.com/a/laravel8-beginner-tutorial
- https://www.tutorialspoint.com/learn-the-basics-of-laravel-8/index.asp l

CHAPTER 9

APPENDIX

9.1 Sample Code

9.1.1 Room Book.Blade.Php

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
   <meta name="description" content="">
   <meta name="author" content="">
   <link rel="preconnect" href="https://fonts.gstatic.com">
   <link

   <title>DREAM VACATION</title>

   <!-- Bootstrap core CSS -->
   <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">


   <!-- Additional CSS Files -->
   <link rel="stylesheet" href="assets/css/fontawesome.css">
   <link rel="stylesheet" href="assets/css/templatemo-plot-listing.css">
   <link rel="stylesheet" href="assets/css/animated.css">
   <link rel="stylesheet" href="assets/css/owl.css">

    <!--booking css file -->
   <link rel="stylesheet" href="css/app.css">
   <link rel="stylesheet" href="css/bootstrap.css">
   <link rel="stylesheet" href="css/chocolat.css">
   <link rel="stylesheet" href="css/common.css">
   <link rel="stylesheet" href="css/easy-responsive-tabs.css">
   <link rel="stylesheet" href="css/flexslider.css">
   <link rel="stylesheet" href="css/font-awesome.css">
   <link rel="stylesheet" href="css/jquery-ui.css">
   <link rel="stylesheet" href="css/style.css">
   <link rel="stylesheet" href="css/swipebox.css">
  <!-- date Validation-->
   <script src="https://code.jquery.com/jquery-2.2.4.js"></script>
```

```
<script src="https://code.jquery.com/ui/1.13.1/jquery-ui.js"></script>
   <link href=https://code.jquery.com/ui/1.13.0/themes/smoothness/jquery-
 ui.cssrel="stylesheet"/>

   <script>
      $(document).ready(function
         (){minDate = new Date();
         $("#checkIn").datepicker(
            {showAnim: 'drop' ,
            minDate : minDate,
            numberOfMonth: 1,
            dateFormat: 'yy/mm/dd',

         });
      });
   </script>
   <script>
      $(document).ready(function
         (){minDate = new Date();
         $("#checkOut").datepicker(
            {showAnim: 'drop' ,
            minDate : minDate,
            numberOfMonth: 1,
            dateFormat: 'yy/mm/dd',

         });
      });
   </script>

  </head>

<body>

  <!-- ***** Preloader Start ***** -->
  <div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
      <span class="dot"></span>
      <div class="dots">
        <span></span>
        <span></span>
        <span></span>
      </div>
    </div>
  </div>
```

```
<!-- ***** Header Area Start ***** -->
<header class="header-area header-sticky wow slideInDown"
data-wow-duration="0.75s" data-wow-delay="0s">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <nav class="main-nav">

          <!-- ***** Menu Start ***** -->
          <ul class="nav">
            <li><a href="/">Home</a></li>
            <li><a href="/booking" class="active">Booking</a></li>
            <li><a href="/room">Rooms</a></li>
            <li><a href="/contact">Contact Us</a></li>
            <li><a href="/gallery"></a></li>

          </ul>
          <a class='menu-trigger'>
              <span>Menu</span>
          </a>
          <!-- ***** Menu End ***** -->
        </nav>
      </div>
    </div>
  </div>
</header>
<!-- ***** Header Area End ***** -->

<div class="page-heading">
  <div class="container">
    <div class="row">
      <div class="col-lg-8">
        <div class="top-text header-text">
          <h6>Dreams with Simple Tabs</h6>
          <h2>Book your Dream House and Happyeee</h2>
        </div>
      </div>
    </div>
  </div>
</div>

<form action="{{ url('useradd') }}" method="POST">
@method('post')
@csrf
```

```
<div id="page-wrapper" >
        <div id="page-inner">
                <div class="row">
            <div class="col-md-12">
              <h1 class="page-header">
                RESERVATION <small></small>
              </h1>
            </div>
          </div>


        <div class="row">

          <div class="col-md-5 col-sm-5">
            <div class="panel panel-primary">
              <div class="panel-heading">
                PERSONAL INFORMATION
              </div>
              <div class="panel-body">
                                    <form name="form" method="post">
                <div class="form-group">
                        <label>Title*</label>
                        <select name="title" class="form-control" required >
                            <option value selected ></option>
                          <option value="Dr.">Dr.</option>
                          <option value="Mis.">Mis.</option>
                          <option value="Mr.">Mr.</option>
                          <option value="Mrs.">Mrs.</option>

                        </select>
                    </div>
                                            <div class="form-group">
                        <label>First Name*</label>
                        <input name="fname" class="form-control" required>

                     </div>
                                             <div class="form-group">
                        <label>Last Name*</label>
                        <input name="lname" class="form-control" required>

                     </div>
                     <div class="form-group">
                        <label>Address*</label>
                    <input name="address" type="address" class="form-control" required>
```

```
                    </div>
                                        <div class="form-group">
                        <label>Email*</label>
                      <input name="email" type="email" class="form-control" required>

                    </div>
                                        <div class="form-group">
                        <label>Nationality*</label>
                        <label class="radio-inline">
                            <input type="radio" name="nation"  value="Indian"
                checked="">Indian
                        </label>
                        <label class="radio-inline">
                    <input type="radio" name="nation"  value="Non Indian ">Non Indian
                        </label>

                    </div>
                    <div class="form-group">
                        <label>State*</label>
                        <select name="state"  class="form-control" require
                                <option value selected >Kerala</option>




                        </select>
                    </div>
                    <div class="form-group">
                        <label>City*</label>
                      <input name="city" type="city" class="form-control" required>
                    </div>


                                                <?php

$countries = array("Aadhar","PAN","Voters ID");

?>
<div class="form-group">
        <label>ID Proof*</label>
        <select name="idprf" class="form-control" required>
          <option value selected ></option>
          <?php
          foreach($countries as $key => $value):
          echo '<option value="'.$value.'">'.$value.'</option>'; //close your tags!!
```

```
endforeach;
            ?>
          </select>
      </div>
        <div class="form-group">
          <label>ID Number*</label>
          <input name="idnum" type ="text" class="form-control" required>



    </div>

        <div class="form-group">
          <label>Upload your Id*</label>
        <input type="file" name="file">
     </div>

       <div class="form-group">
          <label>Phone Number*</label>
          <input name="phone" type ="text" class="form-control" required>

        </div>
        <div class="form-group">
          <label>Pin*</label>
          <input name="pin" type="pin" class="form-control" required>
        </div>
          </div>
          </div>
          </div>
          <div class="row">
<div class="col-md-6 col-sm-6">
<div class="panel panel-primary">
<div class="panel-heading">
RESERVATION INFORMATION
                  </div>
                  <div class="panel-body">

                  <div class="form-group">
                              <label>Which resort you want?*</label>
                              <select name="resort"  class="form-control" required>
   <option value selected ></option>
  <option value="Sunshine paradise">SUNSHINE PARADISE-1800/day</option>
 <option value="Amazing party villa">AMAZING PARTY VILLA-3000/day</option>
 <option value="Apple vally resort">APPLE VALLY RESORT-3000/day</option>
```

```
<option value="Holiday resort">HOLIDAY RESORT-1199/day</option>
 <option value="Green vally resort">GREEN VALLY RESORT-3000/day</option>
<option value="Birds paradise resort">BIRDS PARADISE RESORT-2400/day</option>
 <option value="Vismaya">VISMAYA-3000/day</option>

                        </select>
                    </div>
<div class="form-group">
                        <label>Type Of Room *</label>
                        <select name="troom"  class="form-control" required>


  <option value selected ></option>
 <option value="Superior Room">SUPERIOR ROOM</option>
 <option value="Deluxe Room">DELUXE ROOM</option>

<option value="Guest House">GUEST HOUSE</option>


<option value="Single Room">SINGLE ROOM</option>
                        </select>
                    </div>
                    <div class="form-group">
                        <label>Which of these are u?*</label>
                        <select name="trvlr" class="form-control" required>
          <option value selected ></option>
                            <option value="Solo">Solo traveler</option>
                            <option value="Family">Family travelers</option>
           <option value="Couple">Couple/Pair</option>
                            <option value="Business">Business traveler</option>
                        </select>
                    </div>
                                            <div class="form-group">
                        <label>No.of members *</label>
                        <select name="tmeb" class="form-control" required>
                <option value selected ></option>
                            <option value="1">1</option>
                            <option value="2">2</option>

          <option value="3">3</option>

          <option value="4">4</option>
                        </select>
                    </div>
```

```
<div class="form-group">
                        <label>Check-In</label>


  <input name="cin" id="checkIn" type ="text" class="form-control">

                </div>
                                        <div class="form-group">
                        <label>Check-Out</label>
            <input name="cout" id="checkOut" type ="text" class="form-control">

                </div>
            </div>
        </div>


        <div class="col-md-12 col-sm-12">
          <div class="well">
            <h4>HUMAN VERIFICATION</h4>
 <p>Type Below this code <?php $Random_code=rand(); echo$Random_code; ?>

</p><br />

<p>Enter the random code<br /></p>
<input  type="text" name="code1" title="random code" />
<input type="hidden" name="code" value="<?php echo $Random_code; ?>" />
<input type="submit" name="submit" class="btn btn-primary">
                <?php
                                        if(isset($_POST['submit']))
                                        {
                                        $code1=$_POST['code1'];
                                        $code=$_POST['code'];
                                        if($code1!="$code")
                                        {
                                        $msg="Invalide code";
                                        }
                                        else
                                        {


  $con=mysqli_connect("127.0.0.1","root","","vacations");
  $check="SELECT * FROM roombook WHERE email = '$_POST[email]'";
        $rs = mysqli_query($con,$check);
```

```
if($data[0] > 1) {
                                                    echo "<script
type='text/javascript'> alert('User Already in Exists')</script>";}



        else
                                                {
                                $new ="Not Conform";
  $newUser="INSERT INTO `roombook`(`Title`, `FName`, `LName`,`address`, `Email`,
`National`,`state`,`city`,`idprf`, `idnum`,`image`,`Phone`,`pin`,`Resort`,
                            `troom`,`trvlr`, `Tmeb`, `cin`, `cout`,`stat`,`nodays`)


 VALUES ('$_POST[title]','$_POST[fname]','$_POST[lname]','$_POST[address]',
'$_POST[email]',
'$_POST[nation]','$_POST[state]','$_POST[city]','$_POST[idprf]','$_POST[idnum]',
'$_POST[file]','$_POST[phone]','$_POST[pin]','$_POST[resort]',

'$_POST[troom]','$_POST[trvlr]','$_POST[tmeb]','$_POST[cin]','$_POST[cout]','$new',
datediff('$_POST[cout]','$_POST[cin]'))";
                                                    if
(mysqli_query($con,$newUser))
                                                    {
                                                        echo "<script
type='text/javascript'> alert('Your Booking application has been sent')</script>";

                                                    }
                                                    else
                                                    {
                                                    echo "<script
type='text/javascript'> alert('Error adding user in database')</script>";

                                                    }
                                                }

                                    $msg="Your code is correct";
                                    }
                                    }
                                    ?>
                                </form>

            </div>
        </div>
```

```
</div>
</div> <!-- /.
        PAGE
        INNER


          </form>

 <footer>
  <div class="container">
   <div class="row">
    <div class="col-lg-4">
     <div class="about">
      <div class="logo">
       <img src="assets/images/black-logo.png" alt="">

      </div>

     </div>
    </div>
    <div class="col-lg-4">
     <div class="helpful-links">
      <h4>Helpful Links</h4>
      <div class="row">
       <div class="col-lg-6">
        <ul>
         <li><a href="#">Categories</a></li>
         <li><a href="#">Reviews</a></li>
         <li><a href="#">Listing</a></li>
         <li><a href="#">Contact Us</a></li>
        </ul>
       </div>
       <div class="col-lg-6">
        <ul>
         <li><a href="#">About Us</a></li>
         <li><a href="#">Awards</a></li>
         <li><a href="#">Useful Sites</a></li>
         <li><a href="#">Privacy Policy</a></li>
        </ul>
       </div>
      </div>
     </div>
     <div class="contact
```

```
          <div class="row">
            <div class="col-lg-6">


        <a href="#">010-020-0340</a>
            </div>
            <div class="col-lg-6">
              <a href="#">090-080-0760</a>
            </div>
           </div>
          </div>
         </div>

       </div>
     </div>
   </footer>
    <!-- Scripts -->
    <!-- <script src="vendor/jquery/jquery.min.js"></script> -->
   <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
   <script src="assets/js/owl-carousel.js"></script>
   <script src="assets/js/animation.js"></script>
   <script src="assets/js/imagesloaded.js"></script>
   <script src="assets/js/custom.js"></script>
 </body>
 </html>
```

9.1.2 WEB.PHP

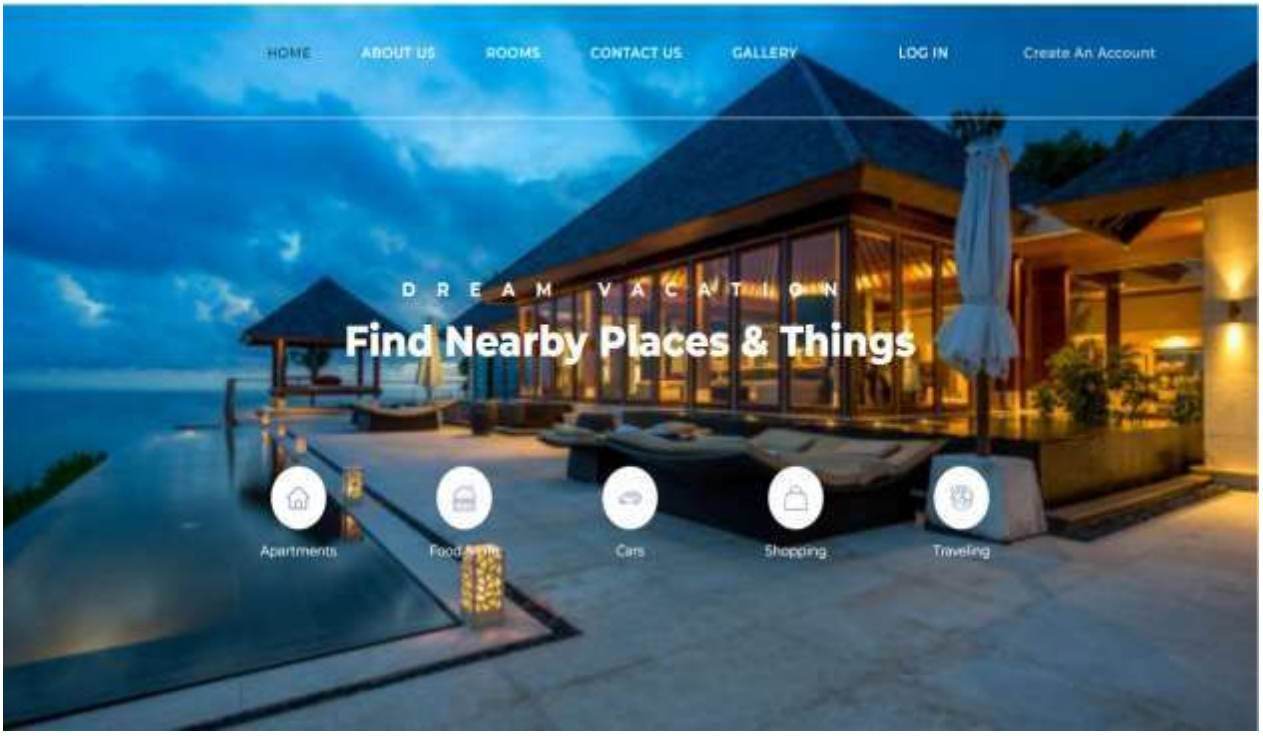
```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\HomeController;
use App\Http\Controllers\BookingController;
use App\Http\Controllers\RoomController;
Route::get('/',function(){return view('welcome');});
Route::middleware(['auth:sanctum', 'verified'])->get('/dashboard', function () {
    return view('dashboard');})->name('dashboard');
Route::get('/redirect',[HomeController::class,'redirect']);
Route::get('/booking',[BookingController::class,'booking']);
Route::post('/useradd',[RoomController::class,'useradd'])->name('useradd');
Route::get('/showbooking',[RoomController::class,'showbooking']);
```

9.1.3 BOOKING CONTROLLER

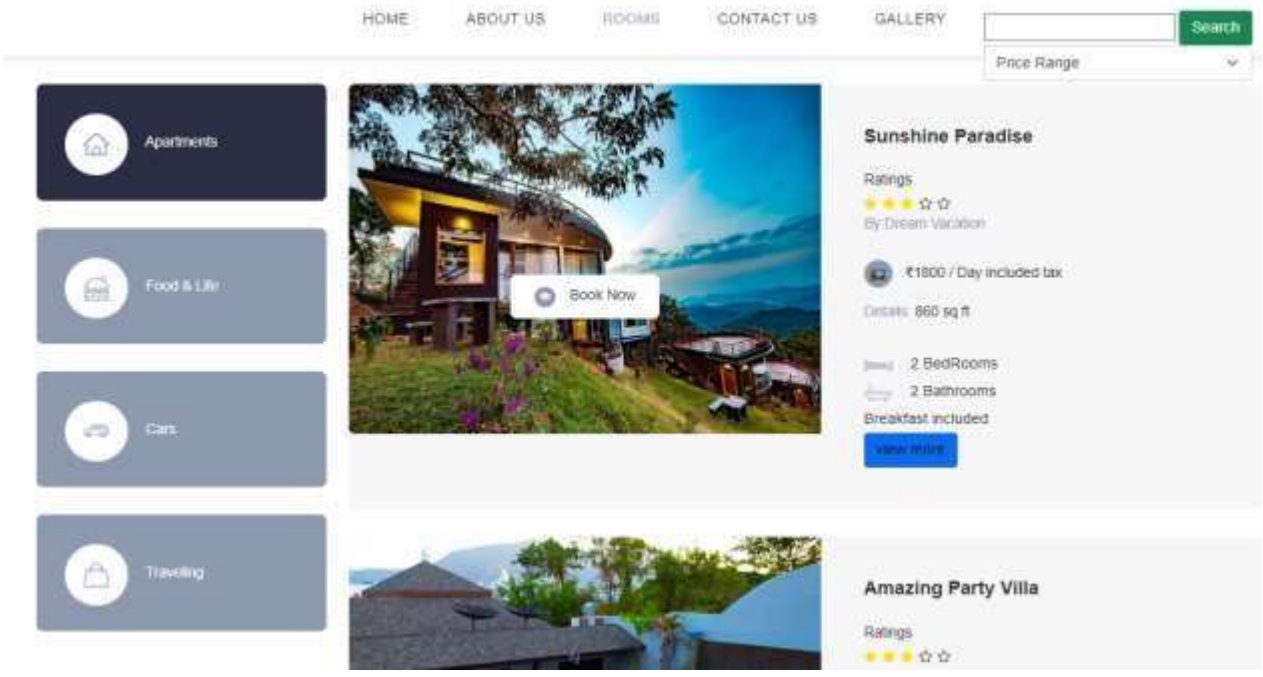
```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Model\Test;

class BookingController extends Controller

{
public function booking()

{

return view('user.booking');

}

public function view1()

{
return view('user.view1');
}
}
```

9.2 Screen Shots

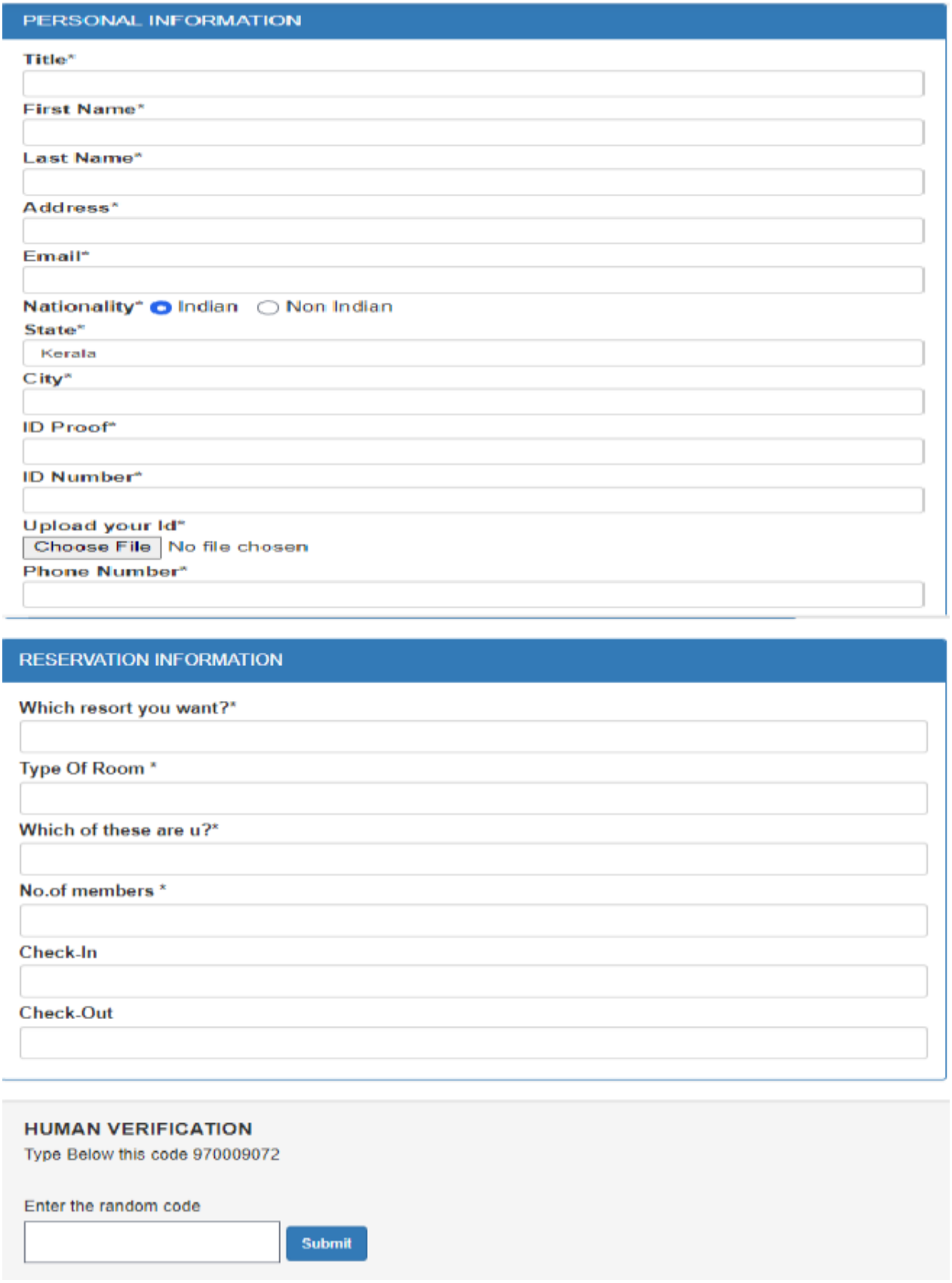
9.2.1 HOMEPAGE



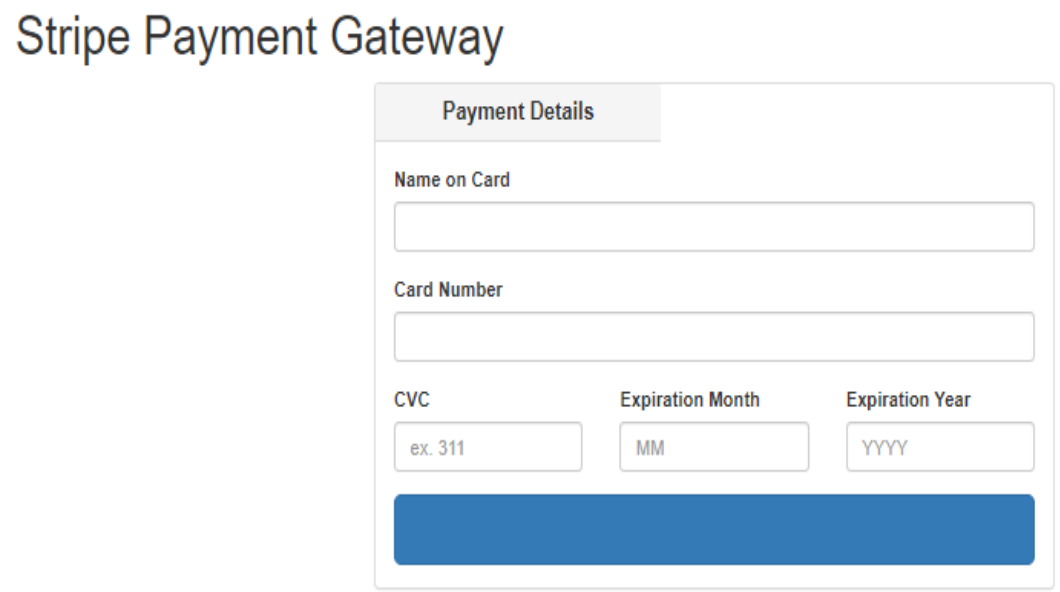
9.2.2 USER VIEW PAGE



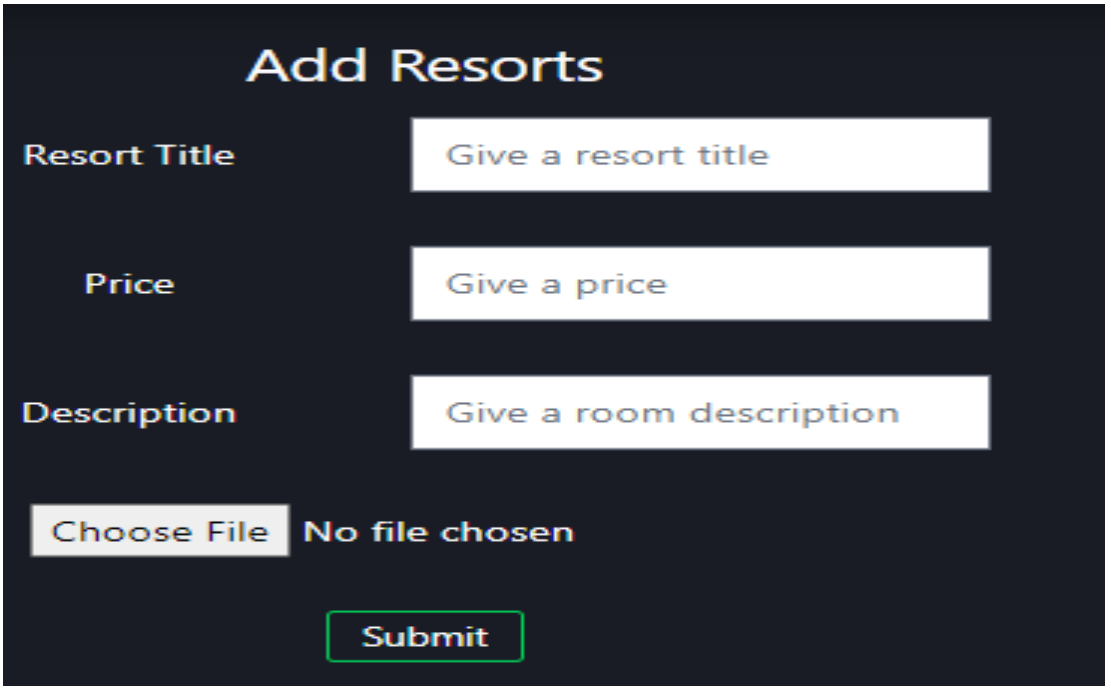
9.2.3 BOOKING PAGE



9.2.4 PAYMENT PAGE



9.2.4 ADMIN ROOM ADDING, DELETE, UPDATE



Title	Description	Price	Image	Update	Delete
Sunshine Paradise	2 BedRooms	1800		Update	Delete
Amazing Party Villa	2 BedRooms	3700		Update	Delete
Apple Vally Resort	2 BedRooms	3000		Update	Delete
Holiday Resort	14 Rooms	1199		Update	Delete