### A

**Project Report on**

### CRUISEHUB

### BTech-IT, Sem VI

Prepared By: Ayush Shah (IT-140) Vaishal Shah(IT-146)

Guided By: Prof. Kunal J Sahitya

Dept. of Information Technology



### Department of Information Technology Faculty of technology,

**Dharmsinh Desai University**

**College road, Nadiad- 387001 April, 2023**

## CANDIDATE’S DECLARATION

We declare that 6th semester report entitled “**CRUISEHUB**” is our own work conducted under the supervision of the guide **Prof. Kunal J Sahitya**.

We further declare that to the best of our knowledge the report for B.Tech. VI semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

**Candidate’s Signature: Candidate’s Name: Ayush Shah**

**Student ID: 20ITUOS002**

**Candidate’s Signature:**

**Candidate’s Name: Vaishal Shah**

**Student ID: 20ITUOS026**

**DHARMSINH DESAI UNIVERSITY**

**NADIAD-387001, GUJARAT**



# CERTIFICATE

### This is to certify that the project carried out in the subject of System Design Practice, entitled “CruiseHub” and recorded in this report is a bonafide report of work of

|  |  |  |
| --- | --- | --- |
| **1) Ayush Shah** | **Roll No. IT140** | **ID No: 20ITUOS002** |
| **2) Vaishal Shah** | **Roll No. IT146** | **ID No: 20ITUOS026** |

**of Department of Information Technology, semester VI. They were involved in Project work during academic year 2022 -2023.**

**Prof. Kunal J Sahitya**

(Project Guide),

Department of Information Technology, Faculty of Technology, Dharmsinh Desai University, Nadiad Date:

**Prof. (Dr.) V. K. Dabhi**

Head, Department of Information Technology, Faculty of Technology, Dharmsinh Desai University, Nadiad Date:

**ACKNOWLEDGEMENT**

It is a great pleasure to express our thanks and gratitude to all those who helped us during development this project and to our professors who guided us .This system design project helps us to gain more understanding about technologies we used to create it .We would like to acknowledge who helped us during these project development.

We express deep sense of gratitude towards our project guide Prof. Kunal J Sahitya towards his innovative ideas and support given to us during entire semester for this subject. And it is his sincerity that helps us to do thorough hard work for this project.

We are thankful to those who guided and helped us during this project development journey and we are full of gratitude and pleasure to be part of this project.

with sincere regards, **Ayush Shah**

**Vaishal Shah**

### Table Contents

ACKNOWLEDGMENT 4

[ABSTRACT 7](#_TOC_250049)

[LIST OF FIGURES 8](#_TOC_250048)

[LIST OF TABLES 9](#_TOC_250047)

1. [Introduction 10](#_TOC_250046)
   1. [Purpose 10](#_TOC_250045)
   2. [Document Conventions 10](#_TOC_250044)
   3. [Intended audience and reading suggestions 10](#_TOC_250043)
   4. [Product Scope 10](#_TOC_250042)
   5. [Technology and Literature Review 11](#_TOC_250041)
2. [Project Management 12](#_TOC_250040)
   1. [Feasibility Study 12](#_TOC_250039)
      1. [Technical feasibility 12](#_TOC_250038)
      2. [Time Schedule feasibility 12](#_TOC_250037)
      3. [Operational feasibility 12](#_TOC_250036)
      4. [Implementation feasibility 13](#_TOC_250035)
   2. [Project Planning 13](#_TOC_250034)
      1. [Project Development Approach and Justification 13](#_TOC_250033)
      2. Project Plan 14
      3. Milestones and Deliverables 14
      4. Roles and Responsibilities 14
      5. Group Dependencies 14
3. [System Requirements Study 15](#_TOC_250032)
   1. [Study of Current System 15](#_TOC_250031)
   2. [Problems and Weaknesses of Current System 15](#_TOC_250030)
   3. User Characteristics 15
   4. Hardware and Software Requirements 16
   5. [Constraints 16](#_TOC_250029)
      1. [Regulatory Policies 16](#_TOC_250028)
      2. [Hardware Limitations 16](#_TOC_250027)
      3. [Parallel Operations 16](#_TOC_250026)
      4. Higher Order Language Requirements 16
      5. [Reliability Requirements 17](#_TOC_250025)
      6. Criticality of the Applications 17
      7. [Safety and Security Considerations 17](#_TOC_250024)
   6. [Assumptions and Dependencies 17](#_TOC_250023)
4. [System Analysis 18](#_TOC_250022)
   1. Requirements of New System 18
      1. User Requirements 18
      2. [System Requirements 20](#_TOC_250021)
   2. [Activity Diagram 23](#_TOC_250020)
   3. [Class Diagram 24](#_TOC_250019)
   4. [System Activity (Use Case) 25](#_TOC_250018)
   5. [Sequence Diagram 26](#_TOC_250017)
5. [System Design 27](#_TOC_250016)
   1. [System Architecture Design 27](#_TOC_250015)
      1. [Data Flow Diagrams 27](#_TOC_250014)
      2. [Deployment Diagram 28](#_TOC_250012)
   2. [Input/Output and Interface Design 29](#_TOC_250011)
      1. [State-Chart Diagram 29](#_TOC_250010)
6. [Implementation Planning 30](#_TOC_250008)
   1. Implementation Environment 30
   2. [Program/Modules Specification 31](#_TOC_250007)
   3. Coding Standards 31
      1. Rules for limiting the use of globals 31
      2. [Standard headers for different modules 31](#_TOC_250006)
      3. Naming conventions for global variables, 31

local variables, and constant identifiers

* + 1. Conventions regarding error return values 32

and exception handling mechanisms

1. [Testing 33](#_TOC_250005)
   1. [Testing Plan 33](#_TOC_250004)
   2. [Testing Strategy 33](#_TOC_250003)
   3. [Testing Methods 33](#_TOC_250002)
   4. [Test Cases 34](#_TOC_250001)
2. [User Manual 35](#_TOC_250000)
3. **Limitation and Future Enhancement 42**
   1. Limitation 42
   2. Future Enhancement 42
4. **Conclusion and Discussion 43**
   1. Conclusions 43
   2. Discussion 43
      1. Self-Analysis of Project Viabilities 43
      2. Problem Encountered and Possible Solutions
      3. Summary of Project work 43
5. **References 44**

## ABSTRACT

This website is for online reservation of cruise .This website will help people to browse their tickets online without contacting to an agent .The system provide broad overview about the facilities provided in the cruise. It also provide the review from our customer who visited in the past.

### 

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Page No.** |
| **Fig 2.2.1** | **Iterative Waterfall Model** | **13** |
| **Fig 4.1** | **Activity Diagram** | **23** |
| **Fig 4.2** | **Class Diagram** | **24** |
| **Fig 4.3** | **Use case Diagram** | **25** |
| **Fig 4.4** | **Sequence diagram for authentication** | **26** |
| **Fig 5.1** | **2 level DFD** | **27** |
| **Fig 5.2** | **Deployment Diagram** | **28** |
| **Fig 5.3** | **State-Chart Diagram** | **29** |
| **Fig 8.1 -**  **8.13** | **UI Interfaces** | **35** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Page No.** |
| **Table 2.2.2** | **Project Plan** | **14** |
| **Table 7.4.1** | **User Test Cases** | **34** |

### INTRODUCTION

### PURPOSE

This website is a platform which will help early age start-ups:-

* + - For viewing different types of seats available for travelling
    - For booking/Reservation of seats on particular date and of particular types of seats

### DOCUMENT CONVENTIONS

This document follows MLA format. Bold-faced text has been used to emphasize section and sub-section headings. Italicized text is used to label and recognize diagrams.

### INTENDED AUDIENCE AND READING SUGGESTIONS

* + - Developers
    - Users (who want to reserve seats)
    - Project Managers
    - Testers
    - Documentation Writers

### PRODUCT SCOPE

* + - Secure registration of all the users.
    - Smooth user experience.
    - Storing crucial data of the users in encrypted form. Site supports end to end encryption.
    - Time to Time update of seats to be available is done for each date.

### TECHNOLOGY AND LITERATURE REVIEW

Following technologies will be used for development/management/tracking activities…. 1)Visual Studio Code for developing the software

2) React JS is used for frontend technology and for backend Services we use NodeJs framework ExpressJS and MongoDB as database and use cloud service provided by mongo i.e. MongoDB atlas to store data

### PROJECT MANAGEMENT

### FEASIBILITY STUDY

### Technical Feasibility

Viewing our project from technical point of view (thinking about various tools and technologies being used in developing the system). We have decided that following technologies will be more than enough to develop complete working system (including tech. & tools used for project tracking, monitoring etc. along with development).

For app development: VS Code Frontend: React JS

Backend: Express JS, MongoDB

We are equipped with basic workflows of each tool and tech. and capable to explore further if required. Each of the above technologies is freely available and some of the skills yet to be learnt but it is manageable. From this, it is clear that our project is technically feasible.

### Time Schedule Feasibility

We have planned the steps for completion of our project in given duration. Firstly, we will perform requirement gathering & analysis by start of January 2023. We will prepare SRS document and the GUI design tentatively by January 2023 ending. The diagrams required for the design as well as the database design will be tentatively completed by February 2023.For coding and unit testing 4 weeks and for system and integration testing another 2 weeks will be required. Hence tentatively by the end of March 2023. We will be able to complete the project and ready for its demonstration at starting of April 2023. Being a 2-member team, we will be able to complete our project in the estimated time.

### Operational Feasibility

Operability and management of project is going to be somewhat difficult but it will be quite feasible to develop project remotely as well as good social media platforms to communicate ideas and work regarding the project. Also, we are a team of 2 -persons. So, from organizational point of view, it is sufficient to maintain proper teamwork even remotely. Hence our project is operationally feasible.

### Implementation Feasibility

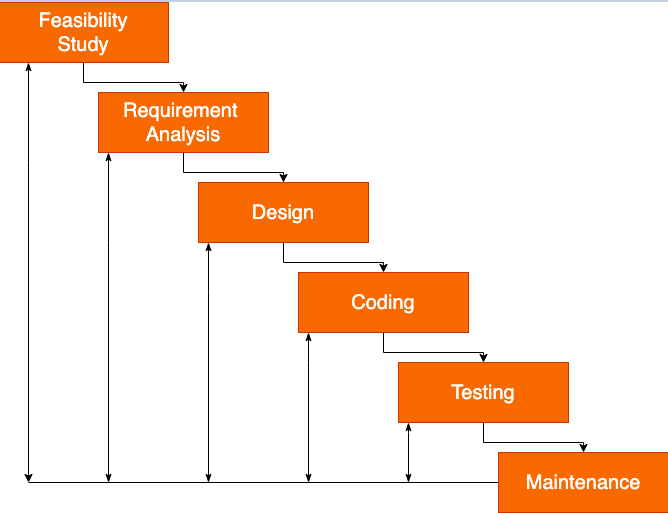
We will be working on developing a web-application for the first time. So, we need to learn the basics of React JS. Also, we need to learn how to connect our project with backend using axios or simply using fetch technique and how to connect Express JS to mongoDB atlas . Since we are well aware of the basics of Development, we just need to learn how to implement it according to our needs which will take around 2 or 3 weeks and be completed before starting implementation.

### PROJECT PLANNING

### Project Development Approach and Justification

The software development approach we used for our project is Iterative Waterfall Model. This includes the stages like:

* + - 1. Feasibility Study
      2. Requirement Analysis
      3. Design
      4. Coding
      5. Testing
      6. Maintenance



### Fig 2.2.1

***Iterative Waterfall Method***

We choose this approach because it gives us the liberty to get back to a stage if it needs some redo. Our entire approach was trial and error and learn as you need. By choosing the iterative waterfall we get the liberty of flexibility and the risk management can be done easily.

### 2.2.1 Project Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task name | Start | Finish | December | January | February | March |
|  |  |  | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 |
| Plan and  Feasibility study | 27/12/22 | 10/01/23 | | | | | | | |  |  |
| Requirements  Gathering | 13/01/23 | 31/01/23 |  | | | | |  |  |
| analysis | 1/02/23 | 7/02/23 |  |  | | | | |  |
| design | 9/02/23 | 23/02/23 |  |  | | | |  |
| coding | 15/02/23 | 7/03/23 |  |  | | | | | | |
| testing | 5/03/23 | 18/03/23 |  |  |  | | | | |

*Table 2.2.2*

### Milestones and Deliverables

The project milestone was to complete it within maximum of 3 months and finally the project was almost completed. The deliverable has mostly covered all topics used in our project scope as per the requirement in project definition.

### Roles and Responsibilities

The role of both the team members was distributed equally for working on this project. Both of us faced many challenges during the development the project and worked together to solve errors and to reach the required solution.

### Group Dependencies

Our Group used many different references individually and even together to progress the Development of Project. Logical analysis was done together. Resource analysis was done on the bases of scope. The project monitored was available for all members to function properly.

### 3. SYSTEM REQUIREMENTS STUDY

### STUDY OF CURRENT SYSTEM

CruiseHub Provide connection of general users to enjoy cruise ride for vacation or weekend times . Current system is not so responsive and server gets very much load which affects the whole user experience.

### PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM

The following Problems exist in current system:

Currently there are some websites that provide this service but they are G2C.They also don’t provide Transparency between investor and startup. And they also charge high processing fees which discourages Startup for use such system.

The above mentioned are some problems which must be kept in mind while building the Application. There are some other problems that concern all the apps in general like data redundancy, keeping secured data & transaction etc.

### USER CHARACTERISTICS (Type of Users Who Are Dealing with The System)

There are 3 types of users in system require:

1)Admin

2)User

1.Admin:

* + Admin handle user data securely
  + Admin handle number of seats available for each date , after some user book some number of sites.
  + Admin handle customers reviews .
  + Admin handles payment/Transaction of user securely using stripe api.
  + Admin Uses services like nodemailer to send email to user after reviewing our service.

1. User
   * User Registered themselves and login
   * Then only , user can able to book seats for particular date.
   * Then user proceed to pay for their reserved seats

### HARDWARE AND SOFTWARE REQUIREMENTS (Minimum Requirements To Run Your System)

There are no such specific hardware requirements other than basic requirements such as a Smartphone , PC with good internet connectivity.

Frontend: react JS

Backend: Express JS , MongoDB

MongoDB atlas a cloud based service for storing data of project online.

Visual Studio Code: Visual Studio Code is the Integrated Development Environment (IDE) for Web app development

### CONSTRAINTS

### Regulatory Policies

* + - * Healthcare Quality Improvement Act
      * Patient Safety and Quality Act
      * Affordable Act

### Hardware Limitations

There is only one limitation of this site , that the device must have internet connectivity.

### Parallel Operations

User must be verified by their user id and password after storing in database . And their passwords are stored in encrypted format in database so that anyone cannot see their credentials.

### Higher Order Languages Requirements

The Higher Order Language used is JavaScript (React JS)for Frontend. And for Backend we have used ExpressJS , MongoDB.

### Reliability Requirements

The site does demand much reliability and it is fully assured that the information about the users should be secured and flow is maintained and accessed according to the rights.

### Criticality of the Application

Establish safe Money transaction and secure for users for storing their data.

Keywords: Responsive Website , good user experience

### Safety and Security Considerations

Our Website has encrypted passwords of users stored , so there is no privacy issue. The data of the User is safe.

### ASSUMPTIONS AND DEPENDENCIES

1. Users have sufficient privileges to access the internet.
2. Internet on Device is running smoothly.

# 4. SYSTEM ANALYSIS

### REQUIREMENTS OF NEW SYSTEM (SRS)

* + 1. **User Requirement**
       1. User 2.Admin
          1. User:

Login: The User can log in to the system using a username and password

View prices of seat

Select number of seats at particular date of particular type

Payment Request: User can see pending payment Request at next page.

Review : User review our service.

* + - * 1. Admin:

Login: Able to login user according to data used by user and which is stored in database

View number of seats booked by user at a particular date

Update seats : Update seats of each type of particular date.

Review: Send email whenever user review our service

Payment: Handle payment securely

### System Requirements

1. Functional Requirements:

### *R1:* Register and Log in

*R1.1 Description and Priority*

This feature creates an account for a new user in the system. This feature enlists the user details in the database. This is important feature so it has high priority.

*R1.2 Stimulus/Response Sequences*

User clicks on the button to initiate registration process . System then prompts the user to fill name, email, password. User enters the fields.

System validates the user’s information and creates a new account for the user. Once a user registers, clicks on the log in button to initiate the log in process. The system prompts the user for email and password and verifies the information. After verification the system displays a home page.

*R1.3 Functional Requirements*

REQ-1: A button for register and log in must be available.

REQ-2: A table in database must be created for all users.

### *R2:* View Prices of different types of seats

*R2.1 Functional Requirements*

REQ-1: List is required to maintain seats at particular date.

### *R3:* Select seats

*R3.1 seat price and availability*

This shows availability of particular seat at particular date.

*R3.2 Stimulus/Response Sequences*

Association can be properly establish between Frontend and Backend.

As whenever user selects some amount of seats then there should be update in backend.

*R3.3 Functional Requirements*

REQ-1: Accept and reject button for Payment

### *R4:* Payment Handling

*R4.1 Description and Priority*

Startup can send Association Request to Investor by providing Investment amount and Request message.

*R4.2 Transaction of amount calculated for user to pay*

By clicking on checkout button page reflecting number of seats of particular type and on particular date is shown and showing calculated amount to be paid.

***R5:* Review**

*R5.2 review page*

Press review option from navbar to send review

*R5.3 Functional Requirements*

REQ-1: For safe negotiation backend Encryption provided by app . Reviews are also stored in database for future uses.

REQ-2: Email is sent whenever user reviewed us.

1. Non-Functional Requirements:

### *R1:* Performance Requirements

Maximum possible quick response to schedule appointment is required, also should

provide fast updating of records. The changes if any made should be reflected automatically in the next screens that is it must be responsive in nature.

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular user as any number of users can access to the system at any time. Also, the connections to the servers will be based on the attributes of the user like his location and server will be working 24\*7 times.

### *R2:* Safety Requirements

The password protected and any managing the entities, schedule the appointments or managing the appointments are done only by privileged users.

### *R3:* Software Quality Attributes

The necessary qualities of software products are:

***R3.1:* Security**

The system is password protected and requires proper authentication for the users to perform operations.

***R3.2:* Maintainability**

The system is to be designed so that it is easily maintained. Also, it should allow incorporating new requirements in any module of system.

### *R3.*3: Reliability

The system will be able to handle multiple appointments. When a Startup associated with Investor database will be updated immediately.

***R3.4:* Portability**

The system will be easily portable on any Android-based system.

### ACTIVITY DIAGRAM

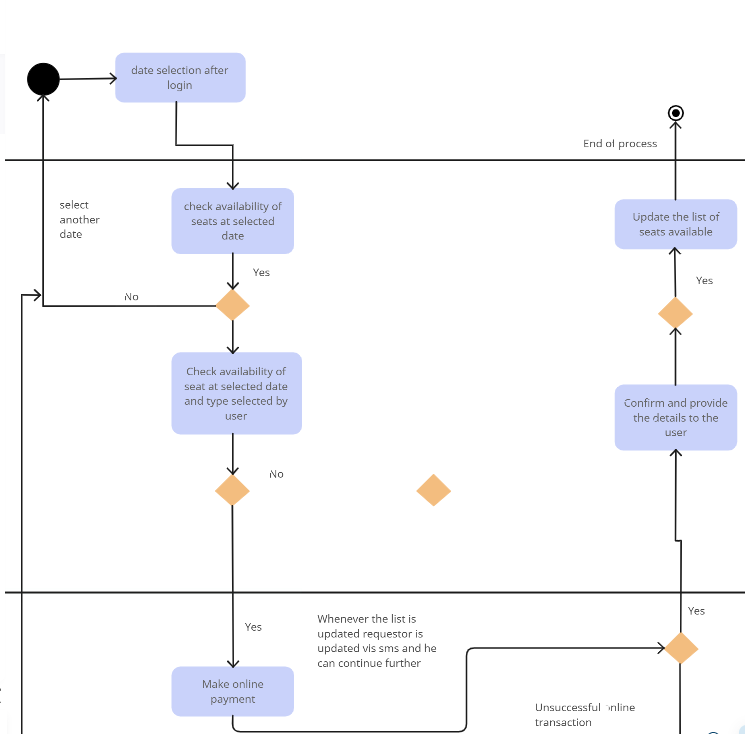
****

Fig 4.1 Activity Diagram for user

### CLASS DIAGRAM

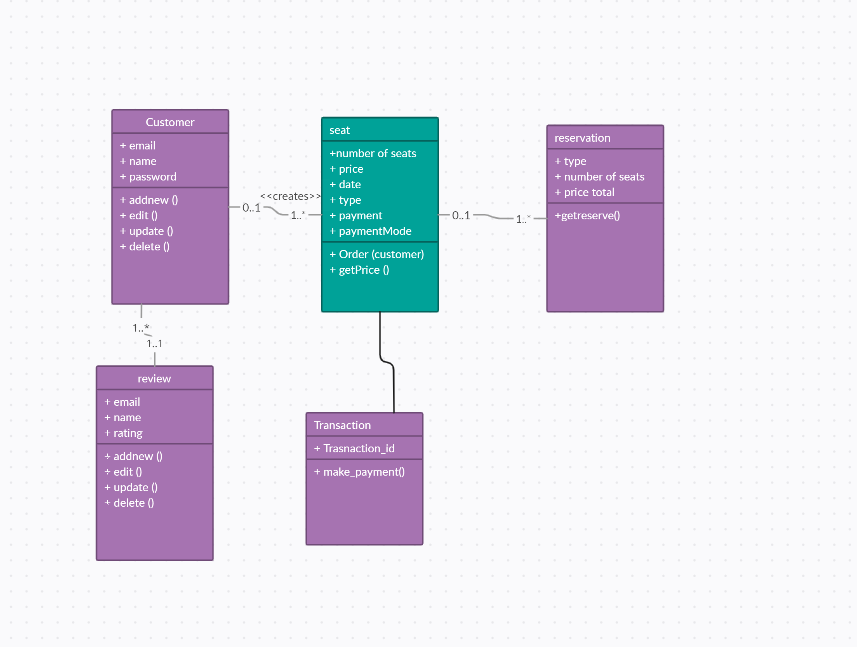
****

Fig 4.2 Class diagram of system

### SYSTEM ACTIVITY (USE CASE)

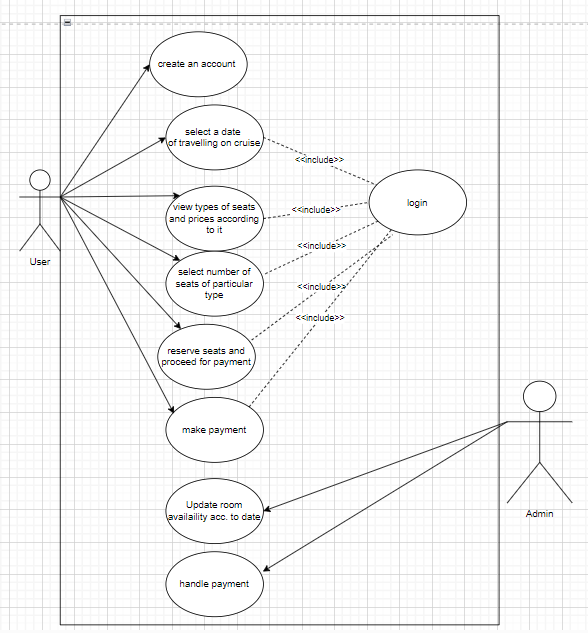
****

Fig 4.3 Use Case diagram for system

### SEQUENCE DIAGRAM

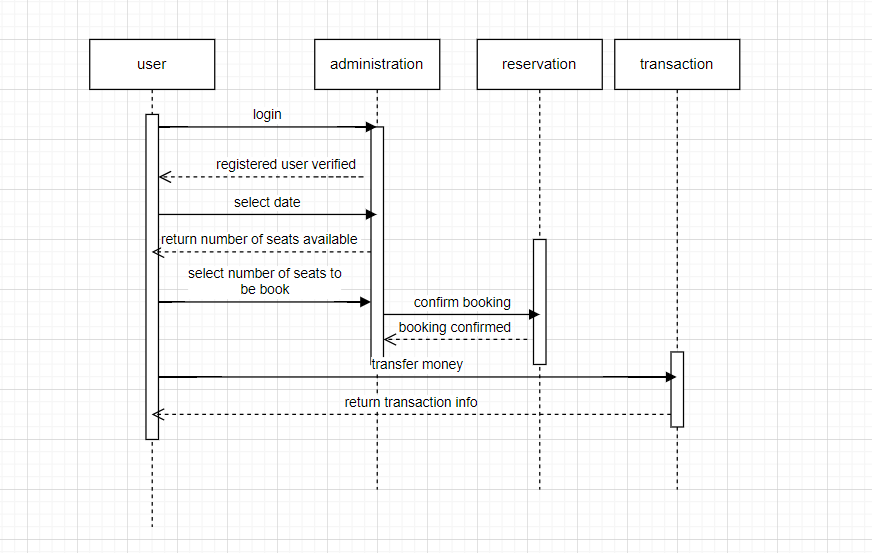
****

Fig 4.4(a) Sequence diagram

# SYSTEM DESIGN

### SYSTEM ARCHITECTURE DESIGN

### Data Flow Diagrams

### 2-Level DFD:

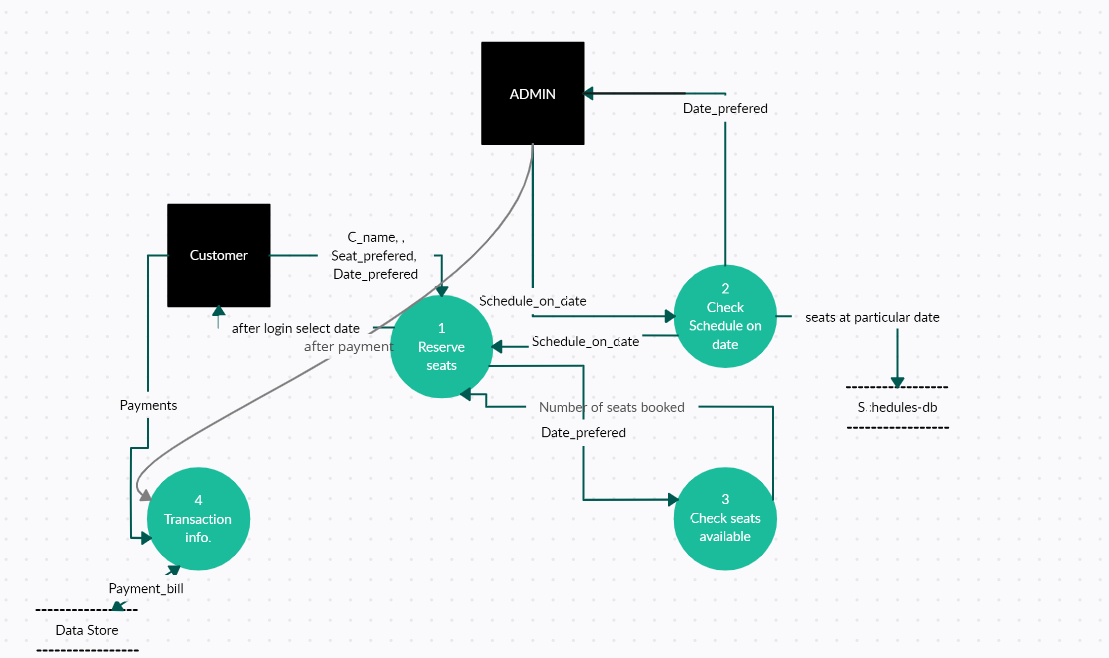
****

Fig 5.1 2 level dfd for system

### Deployment Diagram

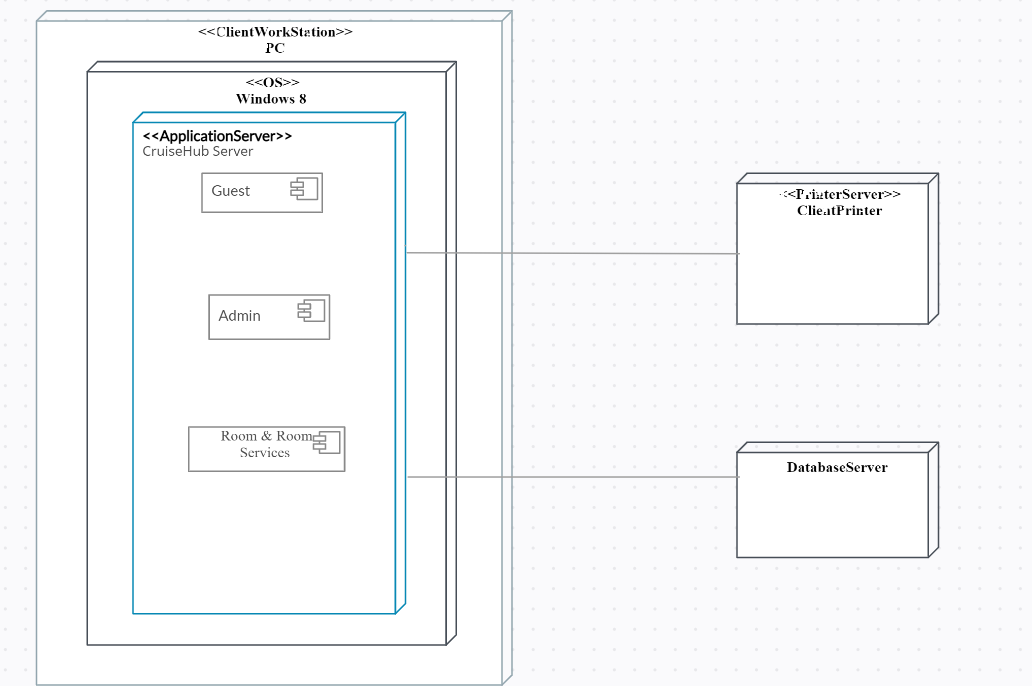
****

Fig 5.3 Deployment diagram for system

### INPUT/OUTPUT AND INTERFACE DESIGN

### State-Chart Diagram

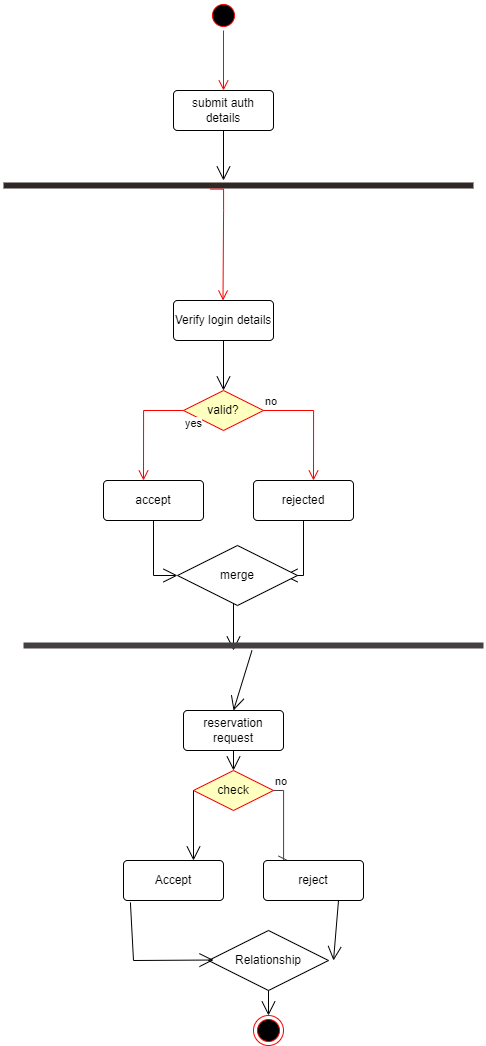
****

Fig 5.4 State-chart diagram for user

# IMPLEMENTATION PLANNING

### IMPLEMENTATION ENVIRONMENT (SINGLE VS MULTIUSER, GUI VS NON-GUI)

For implementation we have used:

Visual Studio Code

Our project is built using Visual Studio Code. Seeing that it is cross platform application we saw fit that Visual studio code provided us with all the required basis for successful implementation of our app. Also, for storing our data we have used Firebase and PostgreSQL which enables our app to always run.

### PROGRAM/MODULES SPECIFICATION

The following Modules are implemented:

* + - User
      * Signup and login
      * View Prices of different types of seats
      * Select date and type of seat to be selected
      * Select number of seats to be booked for particular type and date
      * Proceed to pay
      * Review services
    - Admin
      * Authenticate user
      * Update seat count of particular type for particular date.
      * Handle Payment
      * Store users data after encryption
      * Send mail whenever user reviewed our site service
      * Make payment secure

### 6.2 Coding Standards

* + 1. **Rules for limiting the use of global**:

These rules list what types of data can be declared global and what cannot, with a view to limit the data that needs to be defined with global scope.

### Standard headers for different modules:

The header of different modules should have standard format and information for ease of understanding and maintenance. The following is an example of header format that is being used is:

* + - * Name of the module.
      * Date on which the module was created.
      * Author’s name.
      * Modification history.
      * Synopsis of the module. This is a small writeup about what the module does.
      * Different functions supported in the module with their input/output parameters.
      * Global variables accessed/modified by the module.

### Naming conventions for global variables, local variables, and constant identifiers:

A popular naming convention is that variables are named using mixed case lettering.

Global variable names start with a capital letter (e.g., GlobalData) and local variable.

names start with small letters (e.g., localData). Constant names are formed using capital letters only (e.g., CONSTDATA).

### Conventions regarding error return values and exception handling mechanisms:

The way error conditions are reported by different functions in a program should be standard for example, All functions while encountering an error condition should either return a 0 or 1 consistently, independent of which programmer has written the code.

# TESTING

### TESTING PLAN

The testing technique that is going to be used in the project is White box testing. In White box testing the Tester has knowledge about the internal structure of the code or the program of the software.

### TESTING STRATEGY

The development process repeats this testing subprocess several times for the following phases.

* + 1. Unit Testing.
    2. Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

### TESTING METHODS

Black Box and White Box Testing*:*

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing is likely to be found.

Out of the 2 methods for testing, black box testing and white box testing, we would be using the white box testing as we are aware of the internal functionalities of our application unlike in the black box testing, where we require a 3rd party to test our cases and the internal details are hidden from him.

### TEST CASES

1. **User**

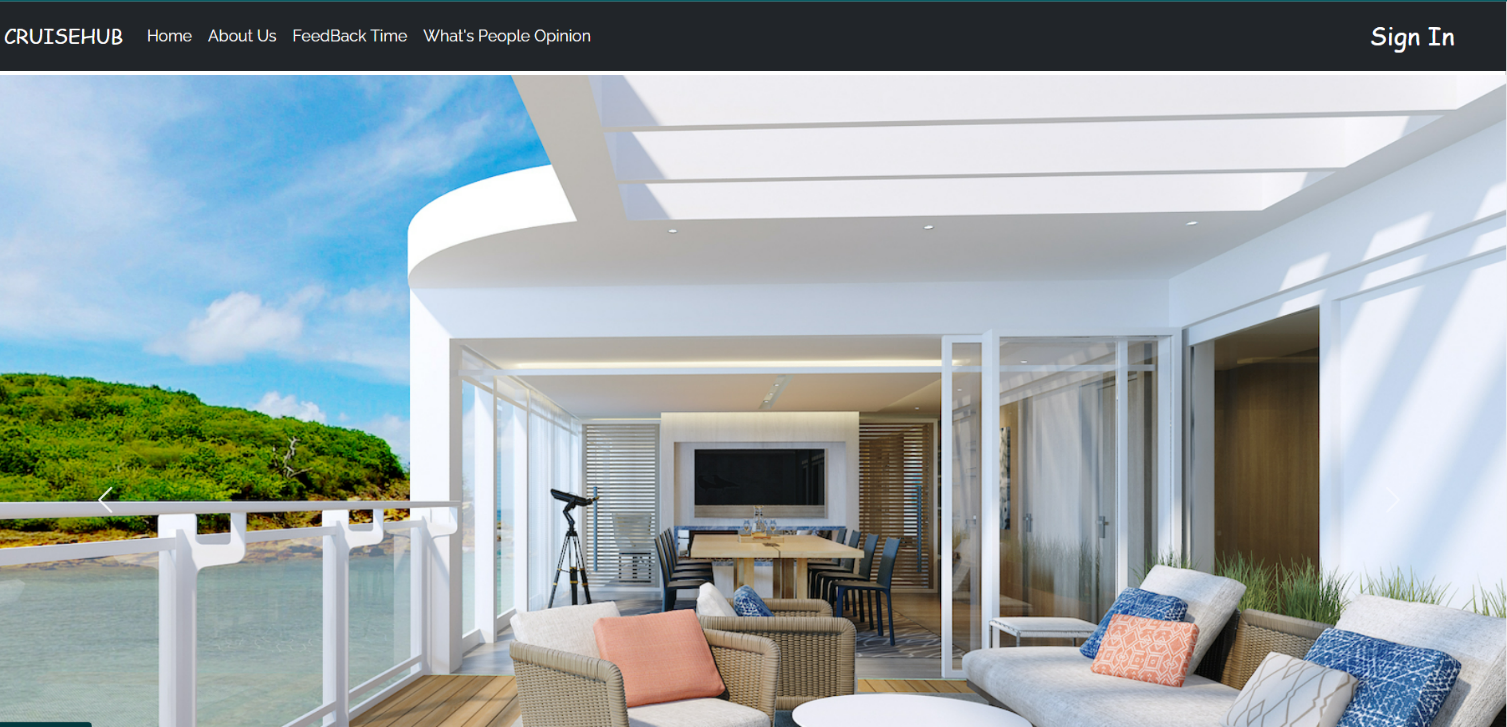
*Table 7.4.1 User Test Cases*

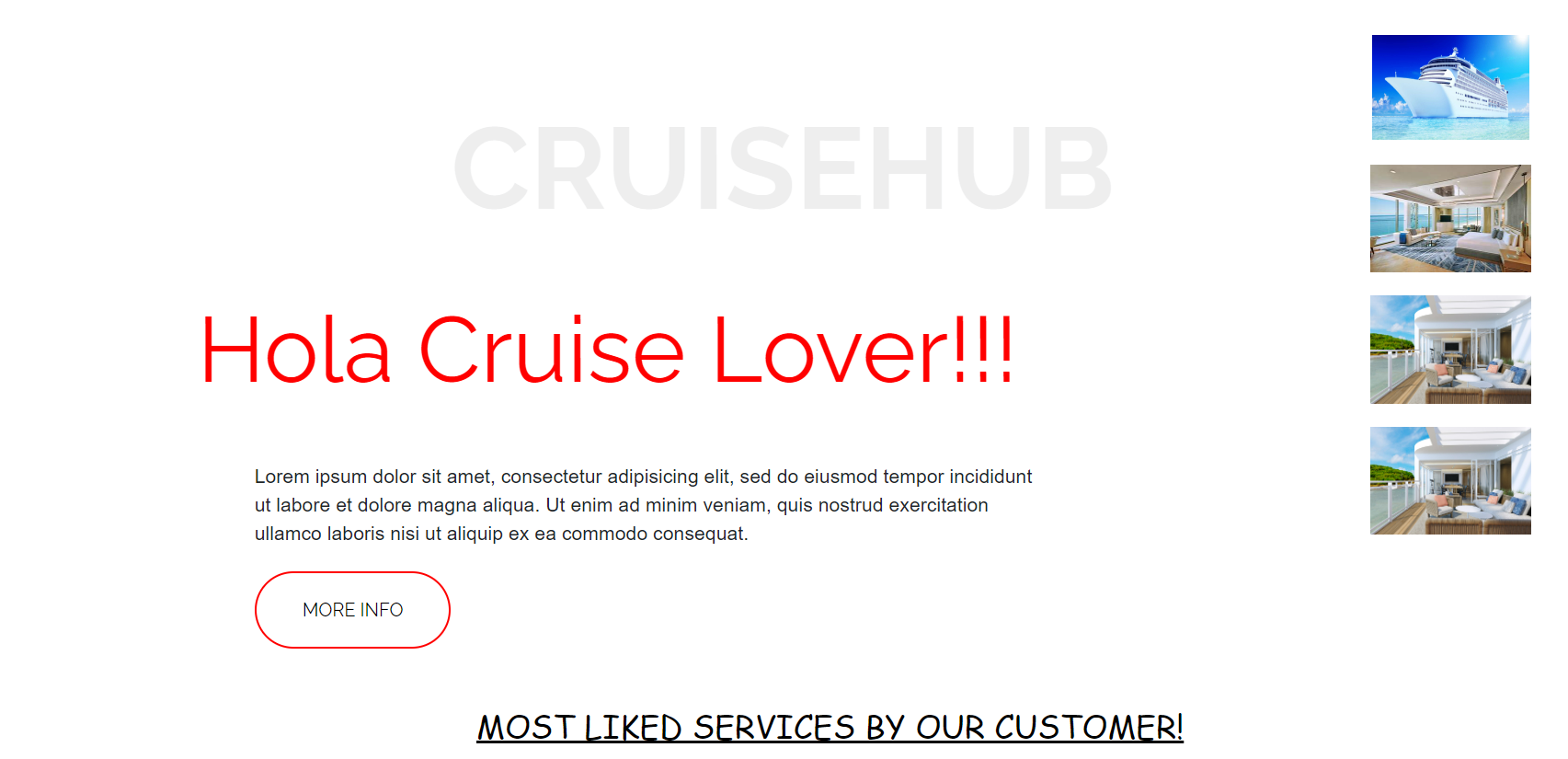
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Test Case** | **Expected Output** | **Actual Output** | **Resu lt** |
| Registration of  Investor or Startup | Validation | Successfully Registered | Successfully Registered | Pass |
| Login of user | Validation | Login Successful | Login Successful | Pass |
| Date selection | View dates | View date s  available | View dates  available | Pass |
| Seat selection | View and select seats of particular type | Selection successfully | Request send Successfully | Pass |
| Proceed to pay page | Calculation of price | Correct price calculated | Correct price is reflected | Pass |
| Enter details for payment | Correct card number and email | Return to homepage reflects | Homepage button option reflects | Pass |
| Review | Enter review and rating | review is stored in database | Email sent successfully | Pass |

# USER MANUAL

User Manuals are manuals that enable the user of a system or application to understand the working of the system and help them to use them efficiently. It is usually written by a technical writer, although user guides are written by programmers, productor project managers, or other technical staff, particularly in smaller companies.

Follow below mentioned steps in order to work with the site:





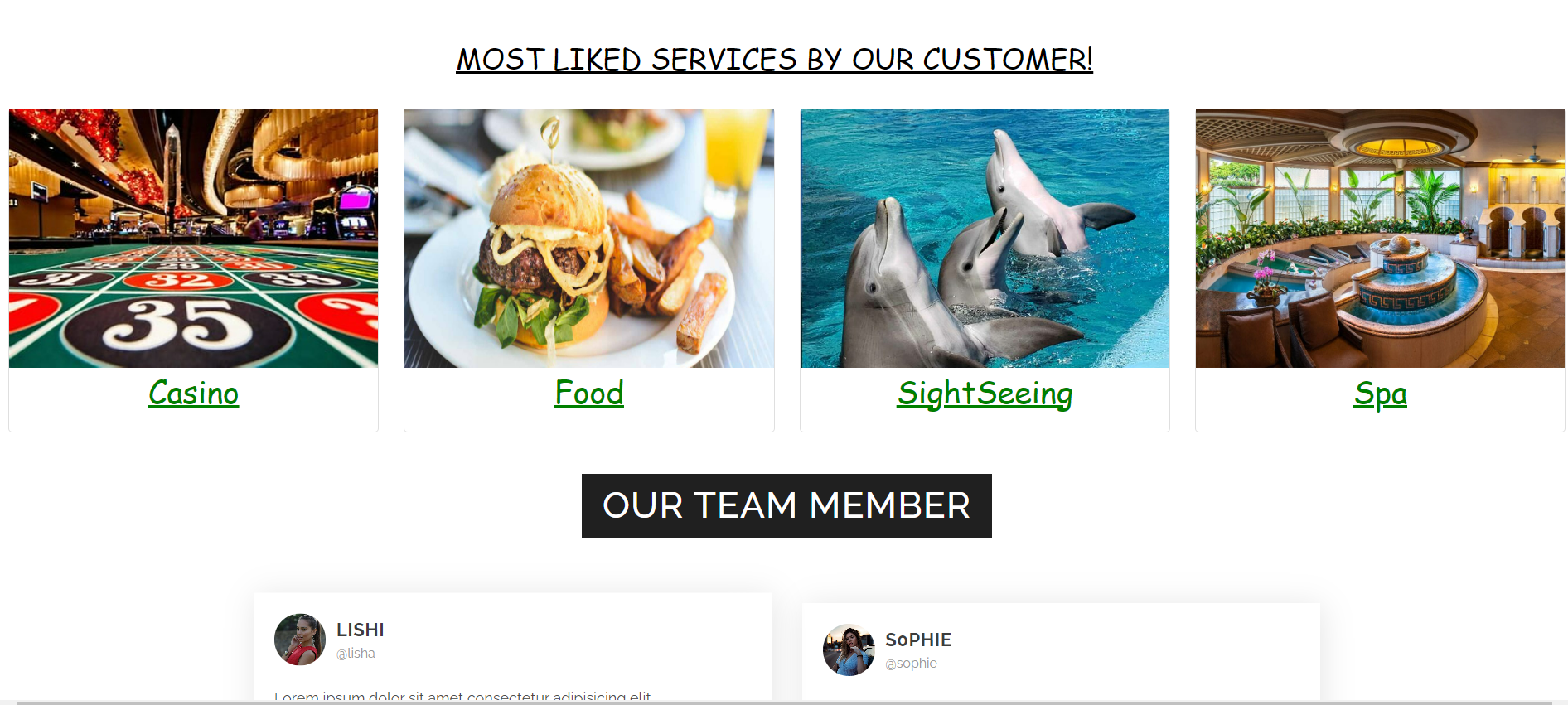


Fig 8.1,8.2,8.3 Home Pages

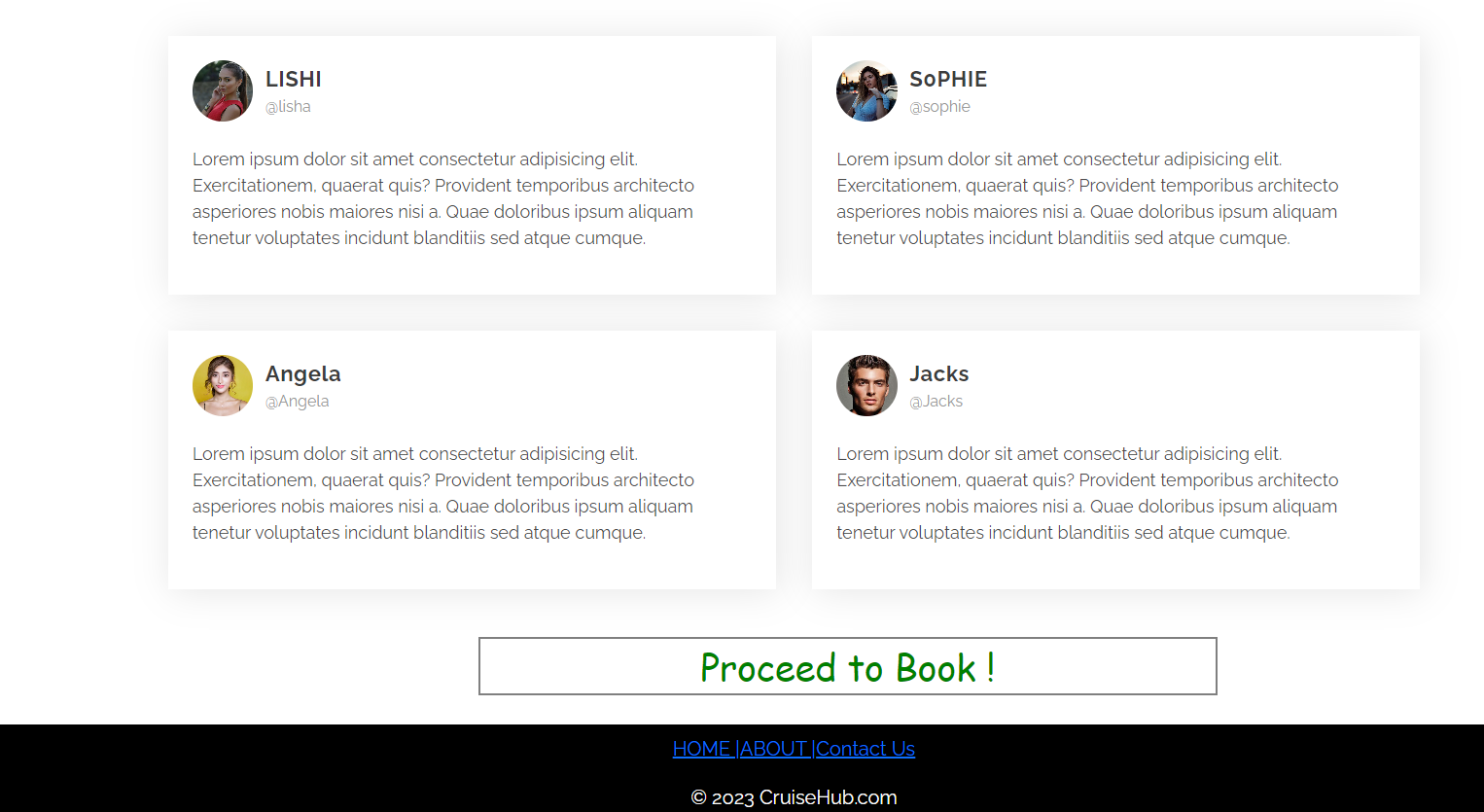
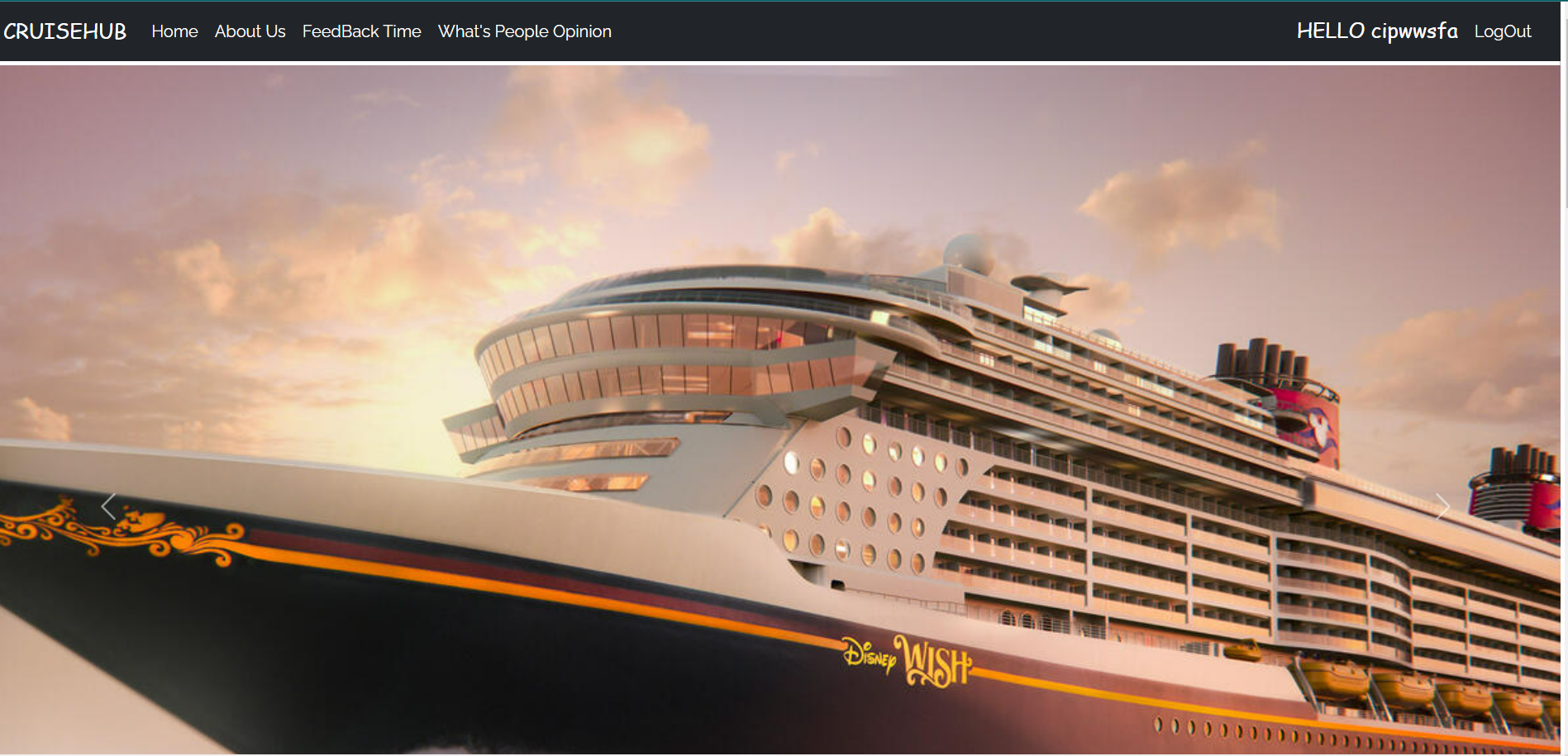


Fig. 8.4 without login



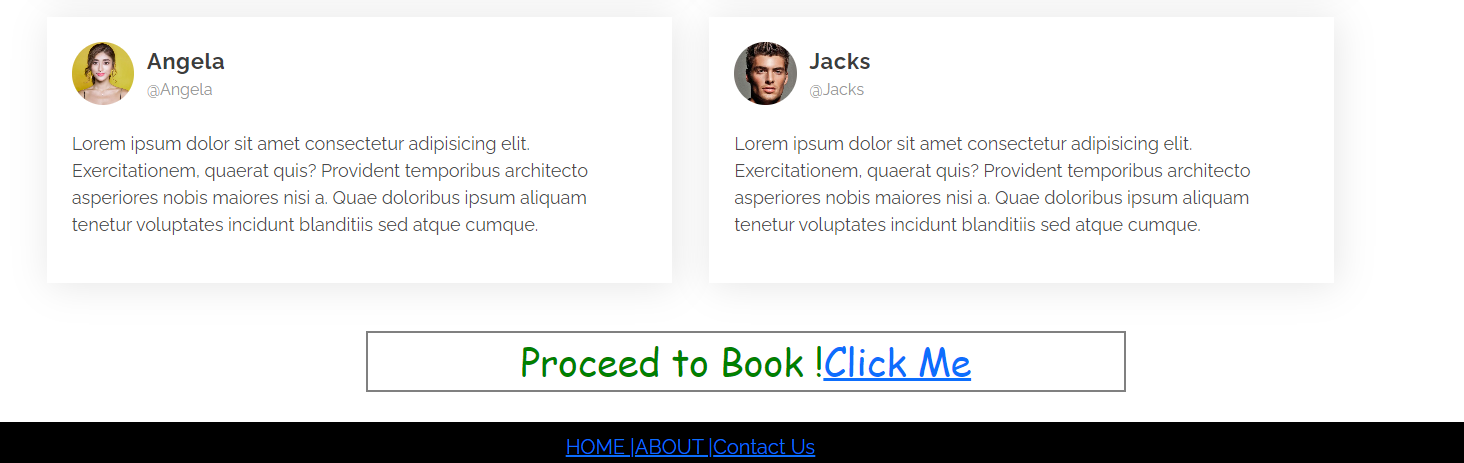


Fig 8.5,8.6 Home page after login

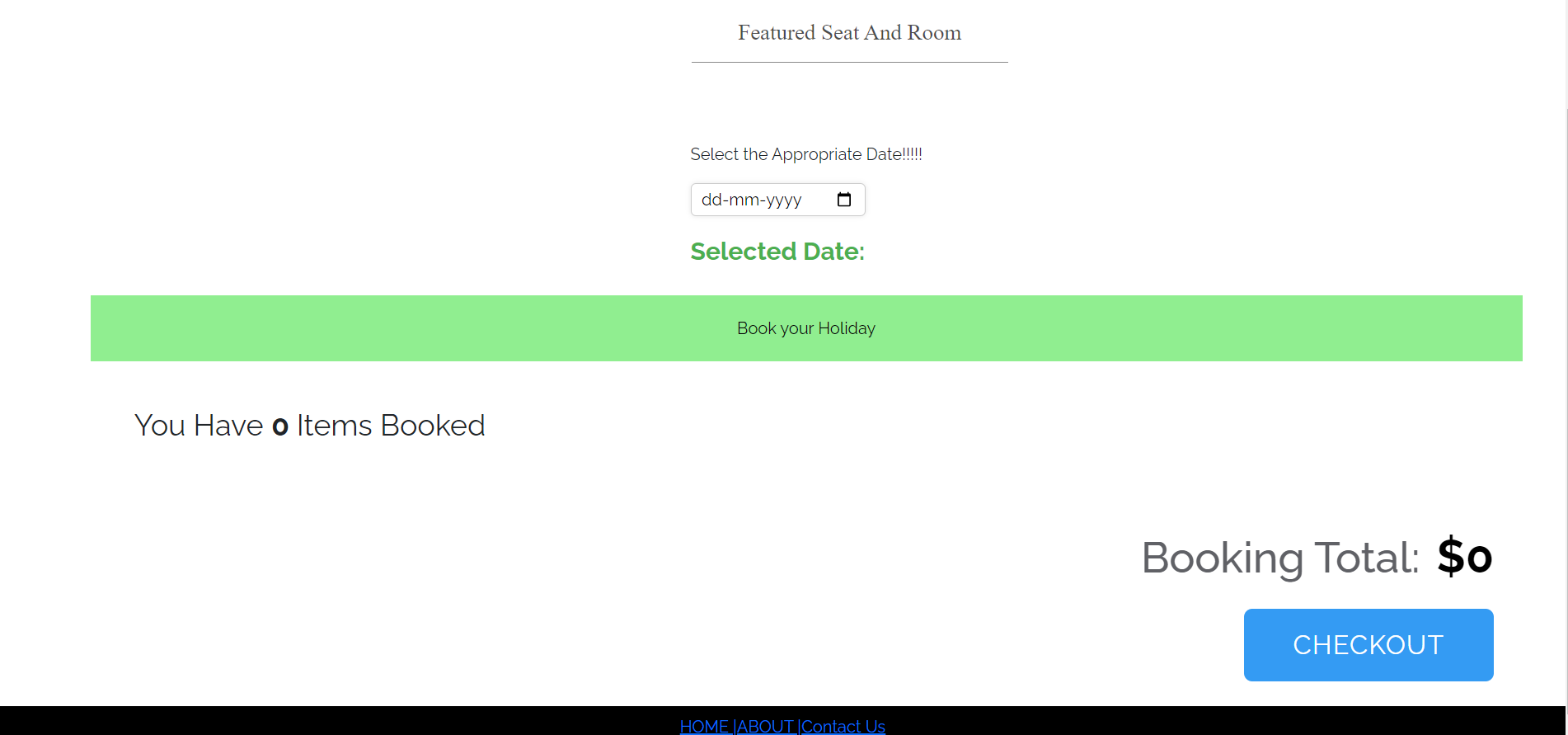


Fig 8.7 date selection

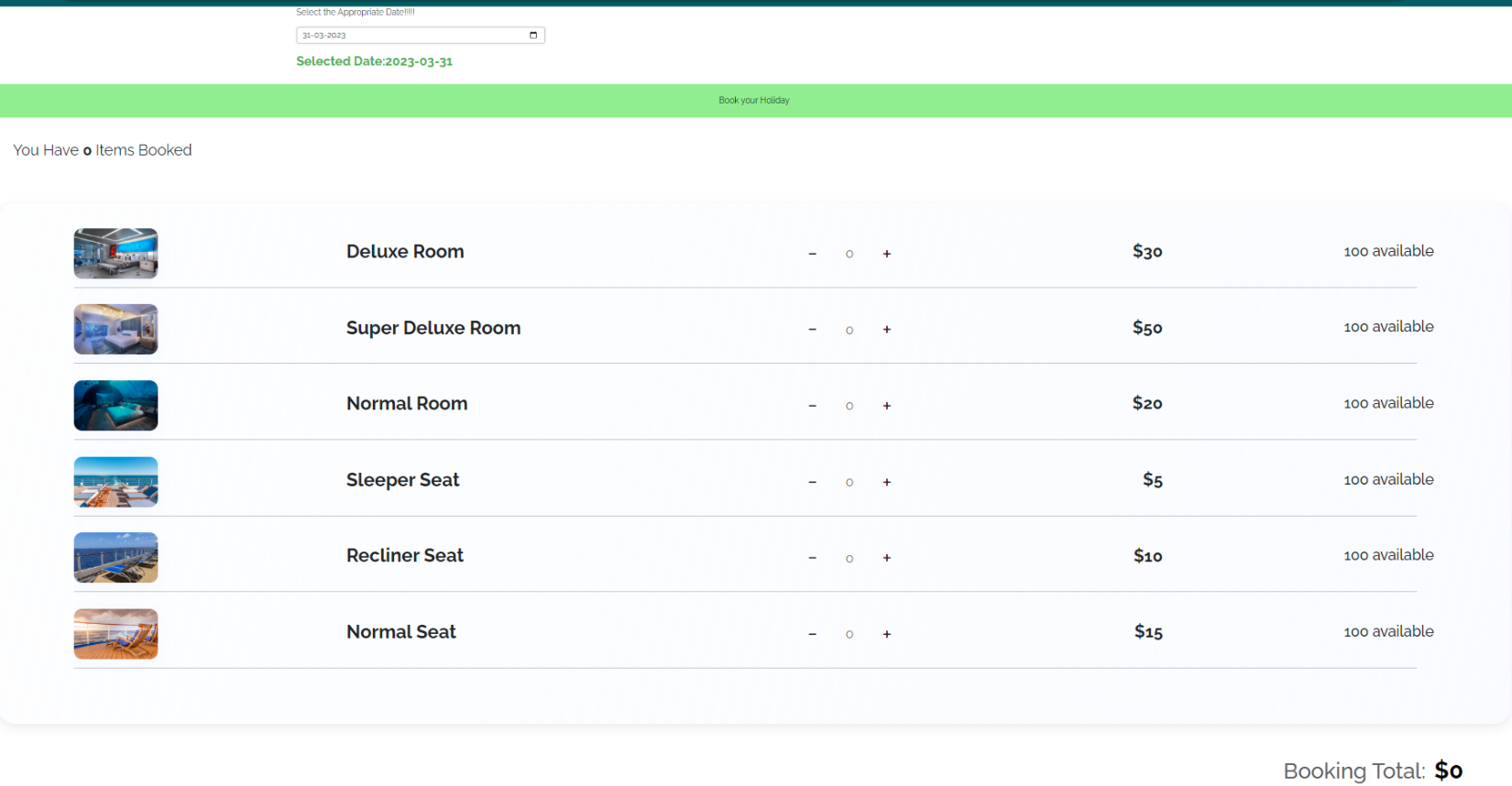


Fig 8.8 seat types and quantity available for each type



Fig 8.9 Checkout page after selecting seats

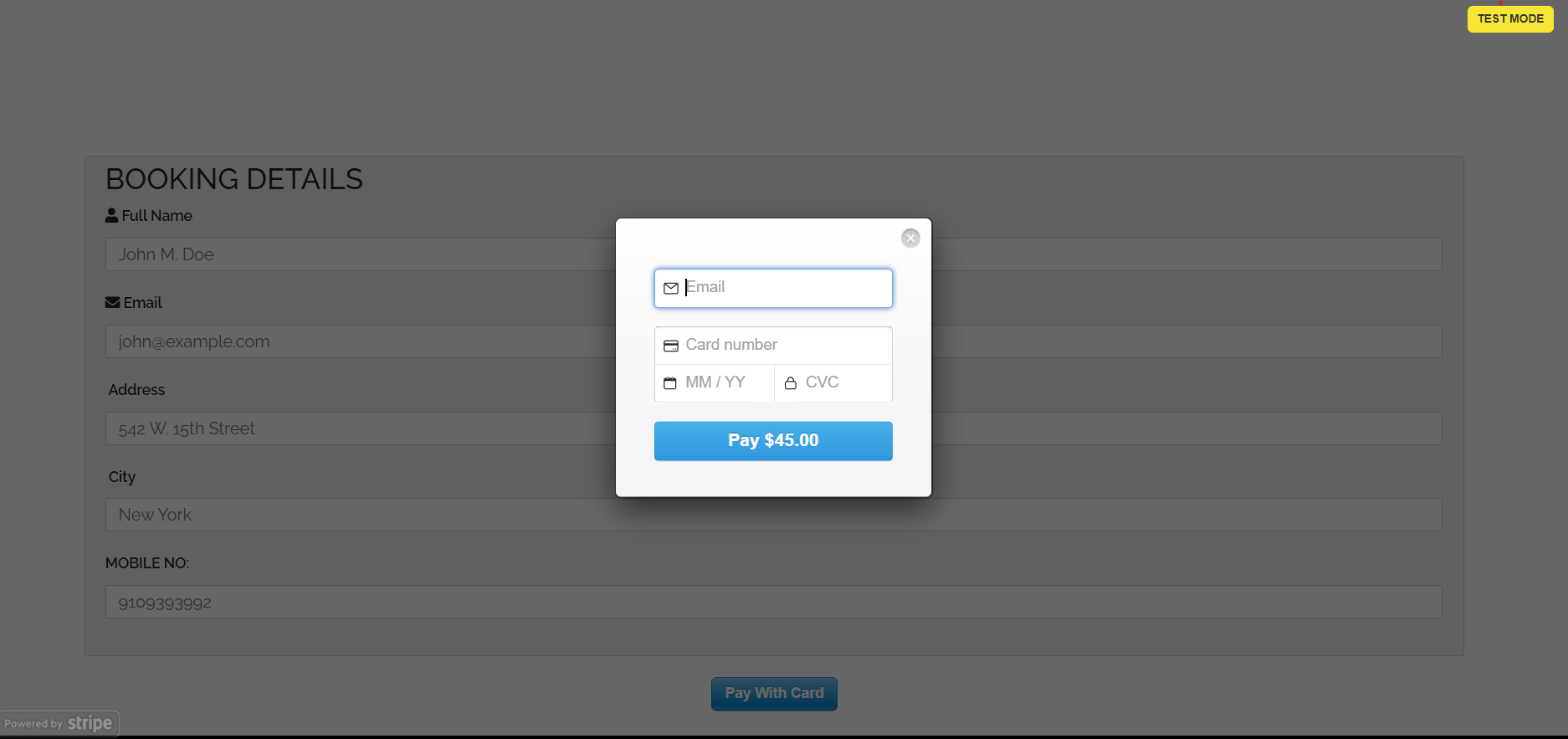


Fig 8.10 Payment page

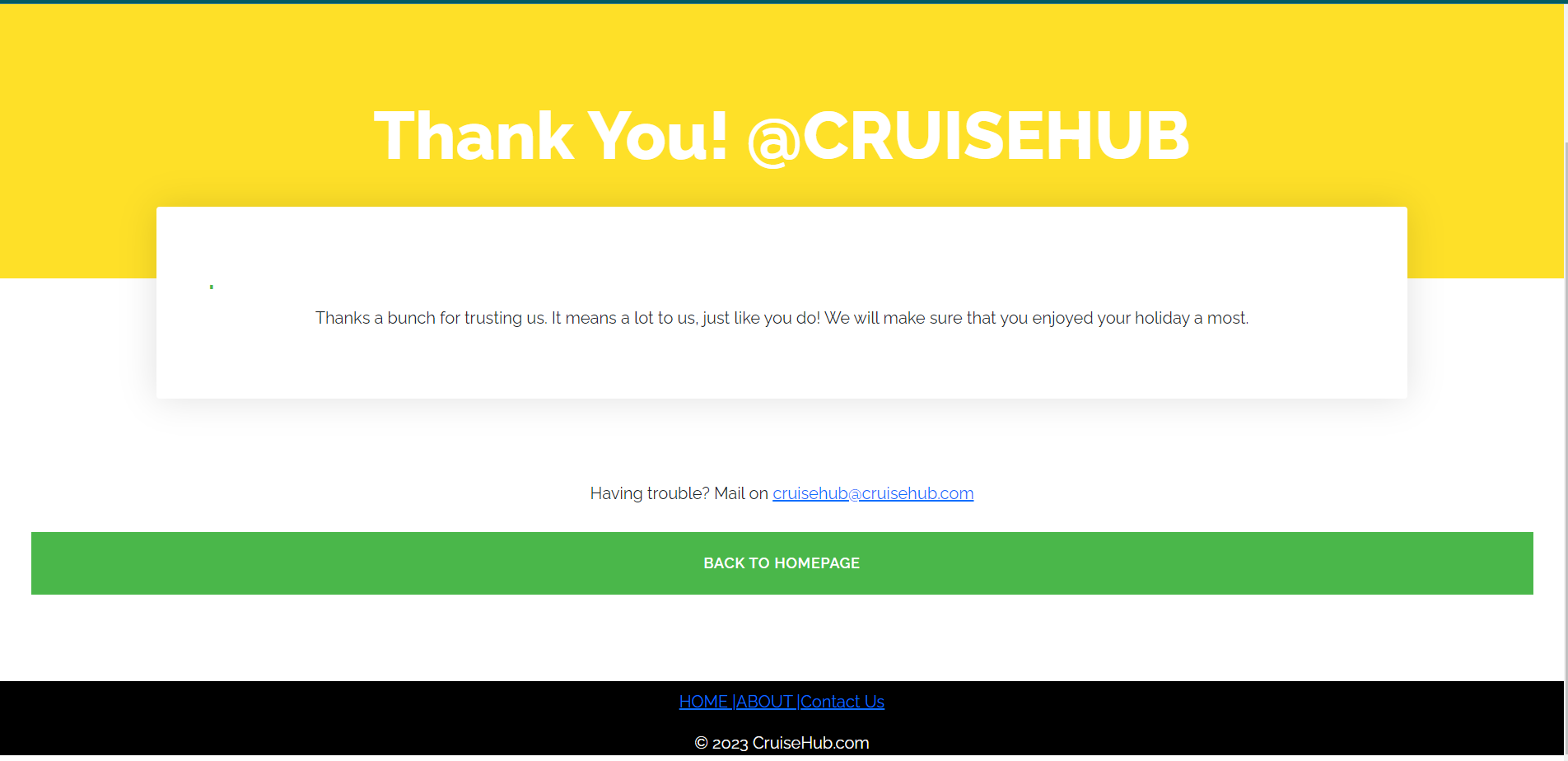


Fig 8.11 Thank you page after payment

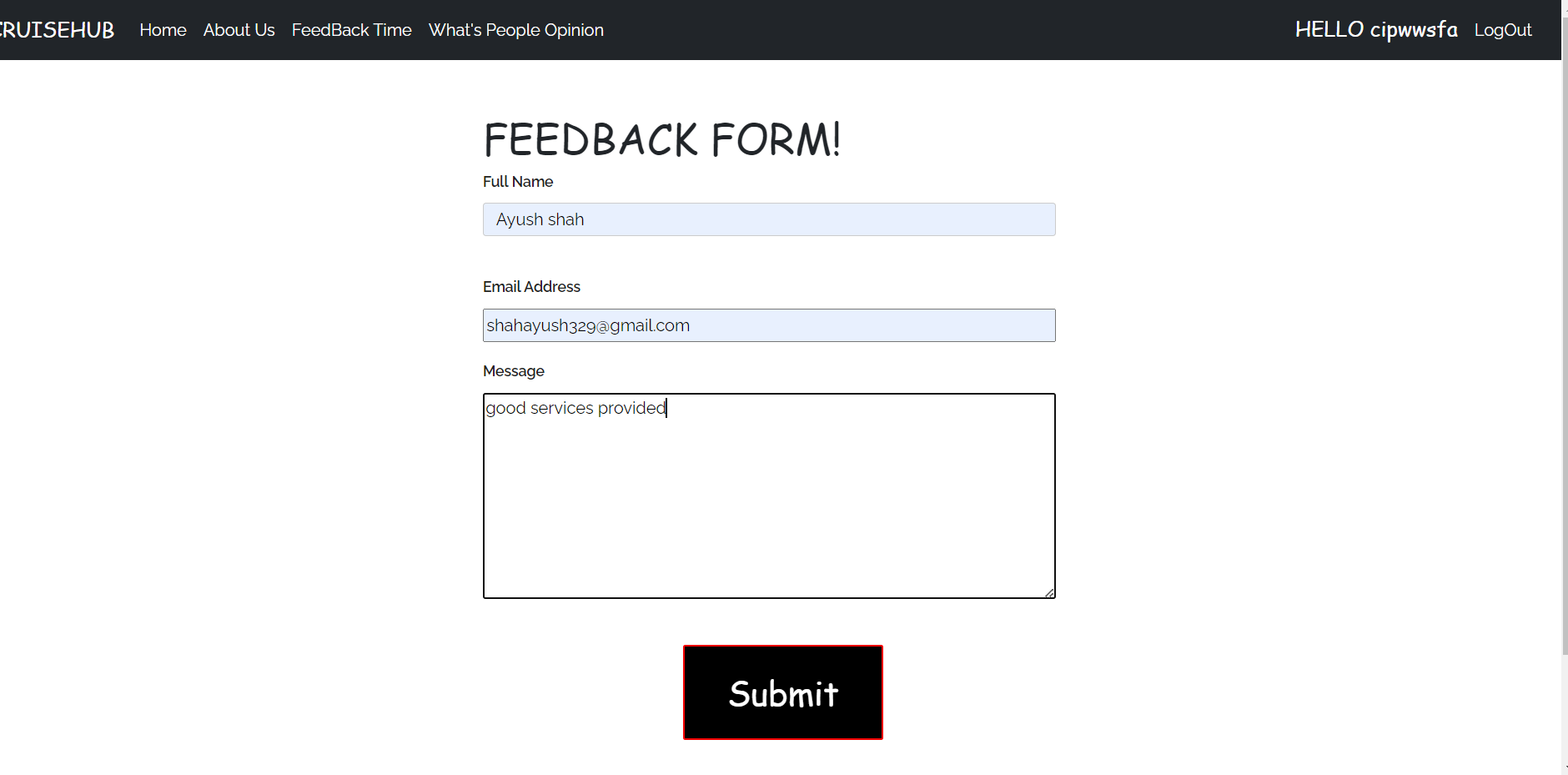


Fig 8.12 Feedback page

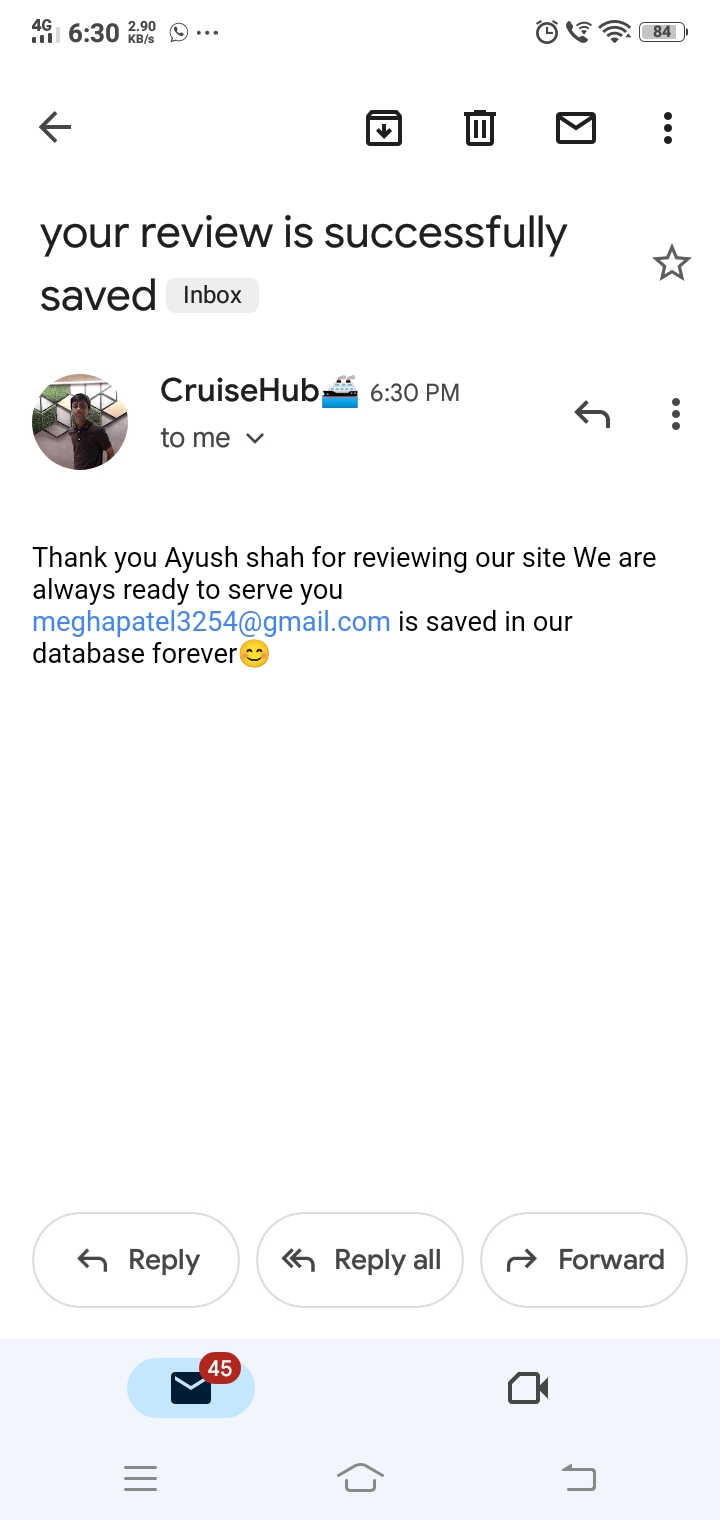


Fig 8.13 email is sent to client after feedback is send

# LIMITATION AND FUTURE ENHANCEMENT

* 1. **LIMITATION**

No Government Certification required for Startup or Investor. There is probability of fake Startup or Investor Registration too. News and Government Schema are depending on Third party API so their may be delay in updating latest News or Schema.

* 1. **FUTURE ENHANCEMENT**
     + More info. of User should be fetched so that daily to daily updates are provided to them through email.
     + Enhance booking algorithm effectively according to dates.
     + Easy and fast processing than now.

# CONCLUSION AND DISCUSSION

* 1. **CONCLUSION**

According to us, this project gave all of us the confidence to believe in ourselves and a great experience of how to work as a team. It also boosted our requirement gathering, system analysis, designing aspects, technical coding as well as time management skills . Also we achieved a great understanding on concepts and technologies like nodemailer and stripe used in backend.

Also, we learned how to work together as a team & collaborate for making ends meet for our app. We also got an insight into how we would have to work in future at a job or startup & how we must contribute for the good of the entity we are working.

* 1. **DISCUSSION**
     1. **Self-Analysis of Project Viabilities**

According to us, this project is absolutely a good start for gaining hands-on experience on projects. It is useful if it is managed according to the goal for which it is made.

* + 1. **Problems Encountered and Possible Solutions**

There are so many problems encountered during this project.

* + - * Technical problems like maintenance of server and database.
      * Untechnical aspects like effective teamwork, better communication of our team members’ individual ideas & combine them for improvement of the application.
    1. **Summary of Project Work**

It is a great achievement to successfully complete the project. The prior knowledge of software engineering has helped immensely in overcoming the various roadblocks. We have done work with pre-planned scheduling related with time constraints and weekly progress in project development. Also, we received guidance from Prof. Kunal J Sahitya at all stages of our project which helped in overall improvement of the project.

It was a excellent experience which taught us many things needed to succeed

further in life during our professional carrier.

* 1. **REFERENCES**
* Node JS documentation: https://www.w3schools.com/nodejs/
* YouTube channel: [https://www.youtube.com/@Pepcoding](https://www.youtube.com/%40HeyFlutter)
* YouTube channel for frontend: <https://www.youtube.com/@ThapaTechnical>