Project

Chologhuri - Resort Booking System

SUPERVISED BY

Abdullah Bin Kashem Bhuiyan

Lecturer

Dept. of Computing and Information System

Daffodil International University

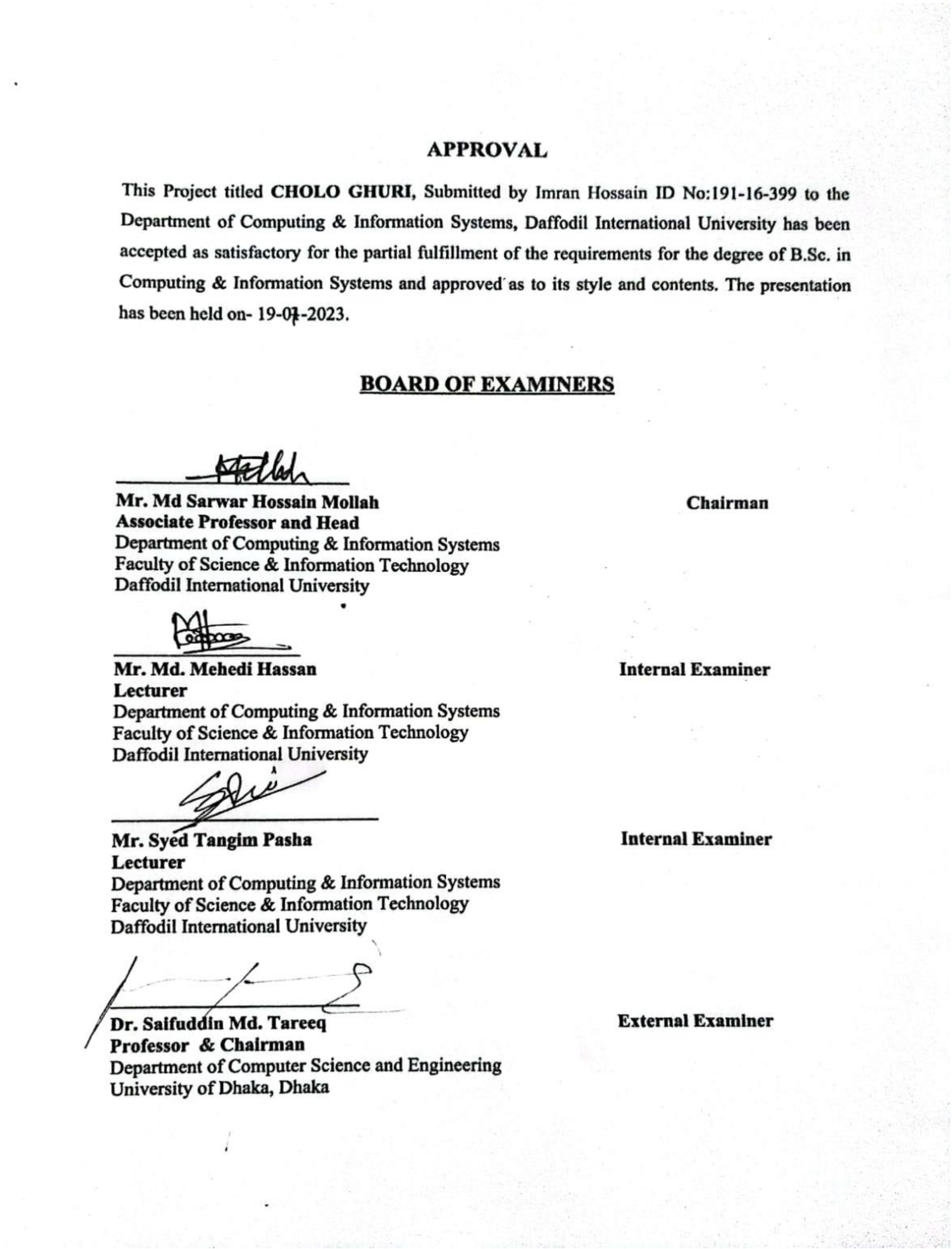
SUBMITTED BY

Name: Imran Hossain

ID: 191-16-399

Dept. Computing and Information System

Daffodil International University



APPROVAL

This Project titled **CHOLO GHURI**, Submitted by Imran Hossain ID No:191-16-399 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 19-07-2023.

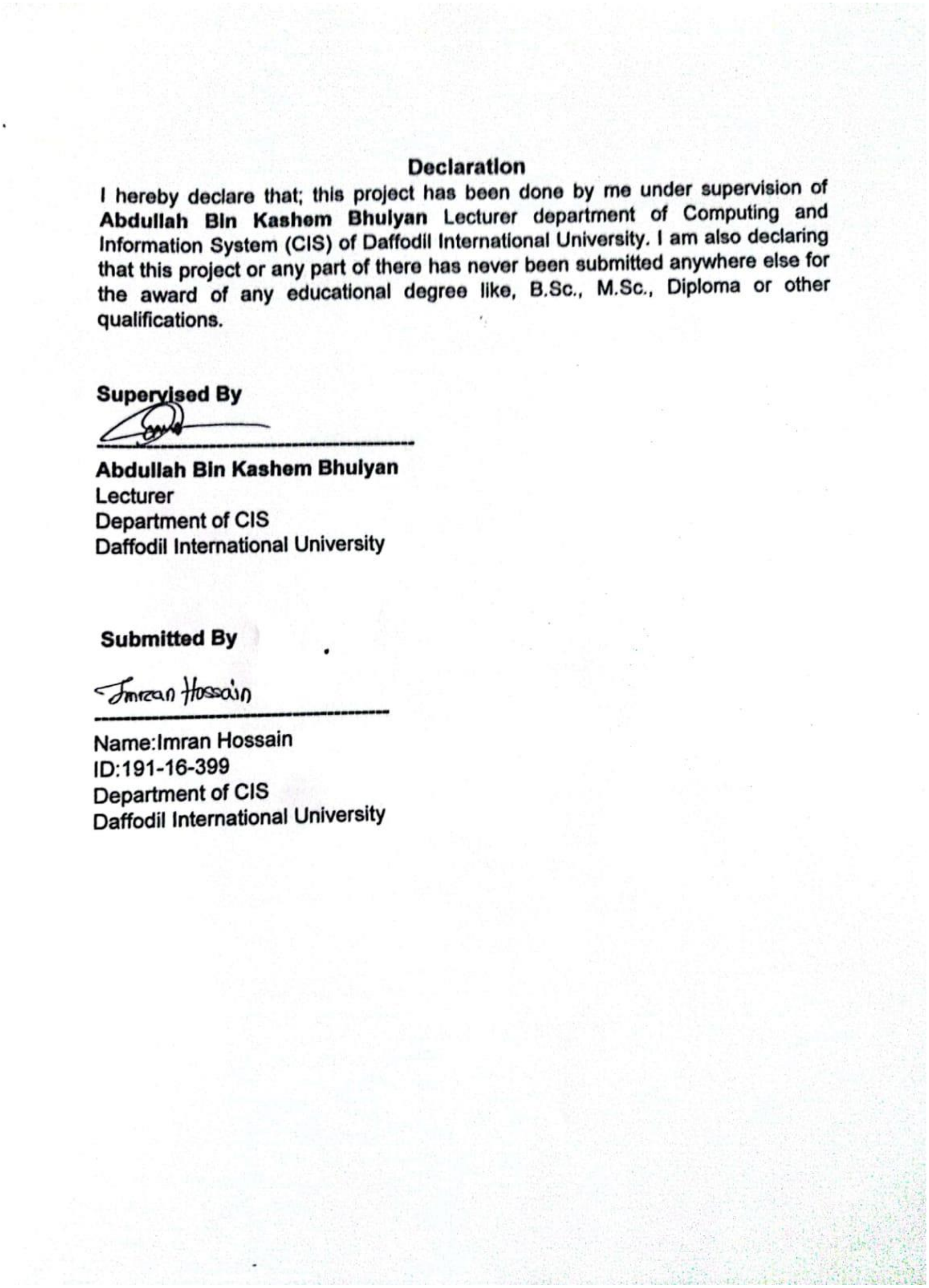
BOARD OF EXAMINERS

Mr. Md Sarwar Hossain Mollah — **Chairman**
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Mr. Md. Mehedi Hassan — **Internal Examiner**
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Mr. Syed Tangim Pasha — **Internal Examiner**
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Dr. Saifuddin Md. Tareeq — **External Examiner**
Professor & Chairman
Department of Computer Science and Engineering
University of Dhaka, Dhaka



Declaration

I hereby declare that; this project has been done by me under supervision of **Abdullah Bin Kashem Bhuiyan** Lecturer department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By

Abdullah Bin Kashem Bhuiyan
Lecturer
Department of CIS
Daffodil International University

Submitted By

Imran Hossain

Name:Imran Hossain
ID:191-16-399
Department of CIS
Daffodil International University

Acknowledgement

I would like to express my gratitude to Abdullah Bin Kashem Bhuiyan Lecturer Department of Computing and Information System. He has been a great inspiration, encourager, mentor, and supporter of me. I'm very grateful to him. Thank you to those who helped me with my internship by providing valuable suggestions. It is with great pleasure and pride that I express my gratitude and deep respect for the Department of Computing and Information System faculty members who took this opportunity.

Last but not least, I would like to express my gratitude to my parents for being a source of great inspiration. They have been a great source of inspiration to me.

Dedication

I would like to dedicate this project to my beloved mother, whose unwavering love, support, and encouragement have been the driving force behind my journey. Her selflessness, sacrifice, and belief in my abilities have shaped me into the person I am today.

Mom, you have been my constant pillar of strength, always standing by my side and cheering me on. Your endless patience, understanding, and guidance have been instrumental in my pursuit of knowledge and passion for computer science.

This project is a testament to your unwavering dedication as a mother, and I am eternally grateful for the values and principles you have instilled in me. Your unwavering faith in my abilities has given me the confidence to push beyond my limits and strive for excellence.

In every line of code, I write, in every feature I implement, and in every success I achieve, I carry your love and support with me. Your belief in my dreams has been the foundation upon which I have built my aspirations.

Executive Summary

Resort Booking System is a web-based application that will use for travellers and resorts owner as well.

First of all, nowadays everything is based online. And the online platform helps everyone to make their life easier. When some traveller wants to go somewhere first they search for a resort/hotel in this area. This Resort Booking system software will help them to find resorts/hotels as their requirement.

And the other hand, the owners of resorts can upload information about their resorts. This will help them to increase their business as well. And there has a super admin who will control the website/application and the relationship between customers/Travelers and owners.

Abstract

The Resort Booking System in Bangladesh is an innovative e-commerce project created to facilitate efficient and convenient resort accommodation booking in Bangladesh. This system's primary objective is to offer both local and foreign tourists an easy platform for searching, comparing, and booking resorts in Bangladesh. Leveraging modern technologies as well as its extensive database of resorts ensures users an effortless booking experience. Travellers can explore a comprehensive selection of resorts - beachfront properties, hillside retreats and urban getaways alike - while accessing comprehensive information regarding facilities, amenities, location and pricing at each. Key features of the Resort Booking System are advanced search functionality, personalized user profiles, secure payment gateways and real-time availability updates. Users can specify dates, locations and preferences to find resorts that best suit them. Reviews from past guests help inform decisions made for both tourists and resort owners; so using this system benefits both. Tourists find it convenient and less time-consuming than using traditional booking methods to discover and reserve accommodations on their terms, while resort owners gain greater visibility with new customers, thus increasing occupancy rates and revenue potential. The project is completed for the company Porch IT and I worked as an intern for them.

Table of Contents

Table of Contents

List of Tables:

Table of Figures:

Chapter 1-Introduction

1.1 Introduction

This document is prepared as part of my internship in the IT Development department at Porch.com during my undergraduate program, Bachelor of Science in Computing and Information Systems at Daffodil International University (DIU). Internships play a crucial role in providing students with practical knowledge and insights related to their academic and career goals. As someone with a keen interest in IT development, my internship at Porch.com offered me a valuable opportunity to gain hands-on experience in the field.

Throughout this documentation, I will outline the development process and key features of a resort booking system. The system allows users to create, update, and delete resorts, as well as book reservations. Additionally, it incorporates email notifications to both users and resort owners upon successful bookings. Furthermore, the system includes functionality to prevent bookings on dates already confirmed, ensuring an efficient and streamlined reservation process.

By documenting my experience and the development process, this report aims to provide insights into the implementation of the resort booking system. It highlights the technologies used, such as Laravel 8, HTML 5, CSS, and Bootstrap, and demonstrates how these technologies contribute to the system's functionality and user experience.

1.2 Document Contents

Chapter 1: Introduction

Briefly describe the industry scope and provide some introductory important key points

Chapter 2: Initial Study

We will summarize here the history of the system proposed, and identify the problem areas. Then, by listing the main reasons, we can discover a possible solution.

Chapter 3: Literature Review

Chapter 4: Methodology

The report contains an analysis and solution of the problems, a review of other solutions and a conclusion.

Chapter 5: Planning

I have presented some development methods that you can use, explained why they are necessary, and selected the best methodology for your project.

Chapter 6: Feasibility

In this section, we have covered different types of feasibility studies. We also demonstrate the cost-benefit of the system proposed.

Chapter 7: Foundation

Discussions on the problem areas, the list of general needs, the recommended solutions and the justifications are recorded.

Chapter 8: Exploration

Display the requirements catalog and the prototype of the new system. Also, demonstrate the activity model and use cases for the old system.

Chapter 9: Engineering

UML diagrams were provided to represent the consumer behavior model and business logic of the project.

Chapter 10: Deployment

Discussion and presentation of some code illustrations for the core modules.

Chapter 11: Testing

Show some examples of functional and non-functional tests, including integration, unit, module, accessibility, security testing, etc.

Chapter 12: Implementation

This section covers all types of implementation methods and includes a training session for the end-user.

Chapter 13: Critical Appraisal and Evaluation

All critical and appraisal evaluations are arranged in this section.

Chapter 14: Lessons Learned

This section focuses on the challenges I've faced and the lessons I learned during project development.

Chapter 15: Conclusion

The project is summarized here, along with my success and experience throughout the project.

Chapter 2- Initial Study

2.1 Background of the project

Web development is a fundamental aspect of Computing and Information Systems, encompassing the creation and enhancement of websites for the Internet (World Wide Web) or an intranet (a private network). The field of web development spans a wide range of activities, from crafting single static pages of plain text to building complex web applications, e-commerce platforms, and social network services.

In today's digital era, having a strong online presence is crucial for businesses and organizations of all sizes. Websites serve as a medium for communication, interaction, and transactional activities, providing a platform to showcase products, services, and information to a global audience. Therefore, the ability to develop and maintain effective websites is essential for businesses to thrive in the digital landscape.

Web development involves a combination of programming languages, design principles, and technologies that enable the creation of visually appealing, functional, and user-friendly websites. It requires expertise in front-end development, which focuses on designing and implementing the user interface and user experience (UI/UX), as well as back-end development, which involves server-side programming and database management.

Throughout this documentation, I will delve into the process of developing a resort booking system, which encompasses various aspects of web development. This includes designing an intuitive user interface, implementing secure authentication and authorization mechanisms, integrating with external APIs for email notifications, and ensuring data integrity through robust database management.

2.2 Scope of the study

This report focuses on the IT development sector of Porch.com, providing insights into the specific aspects of web development undertaken during the internship. While the report primarily revolves around the development of a resort booking system, it also offers a glimpse into the broader IT development landscape at Porch.com.

The study explores the various components of web development, including the development tools, programming languages, frameworks, and APIs utilized in the creation of the resort booking system. By delving into these aspects, this report aims to provide valuable knowledge and understanding for development-related organizations seeking insights into effective web development systems.

The scope of this study encompasses the following areas:

1. Development Tools: The report highlights the tools employed during the development process, such as code editors, version control systems, debugging tools, and project management platforms. These tools play a vital role in streamlining the development workflow and ensuring efficient collaboration within the development team.
2. Development Languages: The study focuses on the programming languages utilized in the creation of the resort booking system. It explores the role of languages such as PHP, HTML, CSS, and JavaScript in implementing the system's functionality and user interface.
3. Frameworks: The report discusses the adoption of Laravel 8, a popular PHP framework, and its significance in accelerating the development process, promoting code reusability, and providing a robust foundation for building the resort booking system.
4. APIs: The study examines the integration of external APIs for email notifications, enabling seamless communication between the system and its users and resort owners. The implementation of these APIs enhances the overall user experience and ensures timely notifications for booking confirmations.

By focusing on these specific areas, this report provides valuable insights and practical knowledge for development-oriented organizations aiming to enhance their web development processes. The documentation of the development experience at Porch.com serves as a reference point for understanding the practical application of web development technologies and methodologies.

2.3 Problem Area

When it comes to resort booking systems in Bangladesh, there are a few notable problem areas that need attention. These issues, which often arise due to human error or technical limitations, can hinder the overall user experience and affect the efficiency of the system. Understanding these problem areas is crucial for improving the resort booking process and providing a seamless experience for travelers.

- Limited Availability Information:
- Inconsistent Pricing:
- Poor User Interface and Navigation:
- Limited Payment Options:
- Insufficient Customer Support:
- Language and Localization Challenges

2.4 Possible Solution

As previously discussed in Bangladesh's resort booking system, Laravel PHP may provide effective solutions. Laravel is a widely popular PHP framework renowned for its elegant syntax, robust features and user friendliness; therefore it may provide effective solutions. Here are a few possible approaches using Laravel:

Implement a Real-Time Update Mechanism: Implementing real-time availability updates is the cornerstone of an accurate resort booking system, Laravel offers features like broadcasting events and websockets that enable instantaneous notifications of room availability changes for instant updates on room vacancy changes.

- Consistent Pricing: Laravel can help ensure consistent pricing by using secure payment gateways or APIs that access real-time pricing information from resort management systems and display prices that reflect actual charges - building trust while eliminating discrepancies.
- User-Friendly Interface and Navigation: Laravel's powerful templating engine, Blade, allows developers to design user-friendly user interfaces using Laravel templating capabilities - creating intuitive yet visually appealing user experiences for resort booking systems that enable easy user navigation through resort booking systems, searching resorts or amenities and booking them seamlessly for users.
- Variety of Payment Solutions: Leveraging Laravel's integration capabilities, developers can integrate multiple local and international payment gateways to provide users with

various payment methods and currencies - providing flexibility and convenience to users while accommodating their preferred methods and currencies.

- Reliable Customer Support: Laravel's extensive ecosystem enables developers to seamlessly incorporate customer support features like live chat, ticketing systems and email notifications into resort booking systems using packages from Laravel or custom development - offering prompt and efficient support throughout the booking process for users.
- Laravel provides robust localization features that enable multilingual support. Developers can utilize Laravel's localization capabilities to translate their resort booking system into different popular languages spoken across Bangladesh ensuring users from varying linguistic backgrounds have equal access and utilize it efficiently for an enhanced overall user experience.

Utilizing Laravel PHP's features and capabilities, Bangladesh's resort booking system can overcome its problems more efficiently. Implementation of real-time availability updates, consistent pricing integration, user-friendly interfaces, diverse payment options, reliable customer support services and language localization can significantly enhance user experiences, making the booking process smooth and pleasurable for travelers.

CHAPTER 3 – LITERATURE REVIEW

The literature review is very important for the successful project research. With this review we can find out the existing system information and compare the work with them to find the updated feature for our current system. So, in this section we will focus on reviewing the existing system and their drawbacks.

3.1 Discussion on the Problem Domain

Resort Booking Systems in Bangladesh aim to enable seamless hotel reservations and management within their borders, drawing tourists both local and foreign alike. As one of Bangladesh's premier tourist spots known for natural beauty and cultural history, Bangladesh welcomes many local and foreign visitors each year; yet their current process for reserving resorts often presents obstacles and inefficiencies which a dedicated booking system could address effectively.

One key issue stemming from this lack of centralized information regarding available resorts and their amenities, pricing and availability creates difficulties for travelers in finding accommodations that suit both their preferences and budget. Furthermore, an unsystematized reservation process causes customer frustration as well as potential loss of business for resort owners.

Manual booking processes often use phone calls or in-person visits as the means for making reservations, which requires considerable time and effort for both customer and resort staff, with increased risks for errors or miscommunications due to limited payment options online and reduced convenience preventing timely bookings being secured promptly. Resort owners face operational hurdles due to an ineffective system for handling reservations, cancellations and modifications; without an efficient centralized solution it becomes challenging to track bookings, allocate rooms efficiently and handle changes or cancellations effectively resulting in overbookings, customer dissatisfaction and revenue losses.

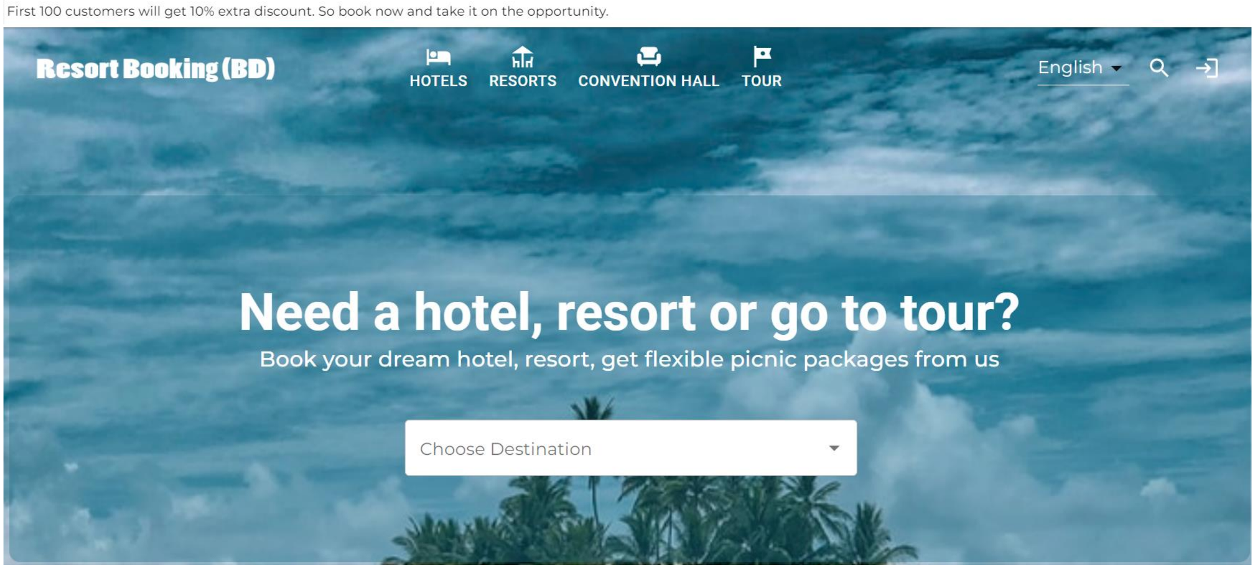
To address these challenges, a Resort Booking System in Bangladesh must aim to offer users with an easily navigable online platform that consolidates information on various resorts throughout Bangladesh. A hassle-free booking experience must allow customers to search available accommodations, view detailed descriptions and compare prices before reserving directly online via payment gateways for hassle-free transactions and reservations.

Additionally, resort owners will require robust backend features from their system such as an effective dashboard to manage bookings, track availability and handle modifications or cancellations as they arise. Automated notifications and reminders will help streamline communication while guaranteeing an exceptional guest experience.

3.2 Comparison of the leading solutions

There are several resort booking solutions in Bangladesh that can meet the needs of different travelers. Compare and explore some of the best options.

ResortBookingBD : ResortBookingBD offers an extensive range of accommodations and resorts in Bangladesh. The platform has an intuitive interface that allows travelers to find and book resorts according to their budget and preferences. Users can also find detailed information on each resort including reviews, photos and amenities. (Resortbookingbd, n.d.)

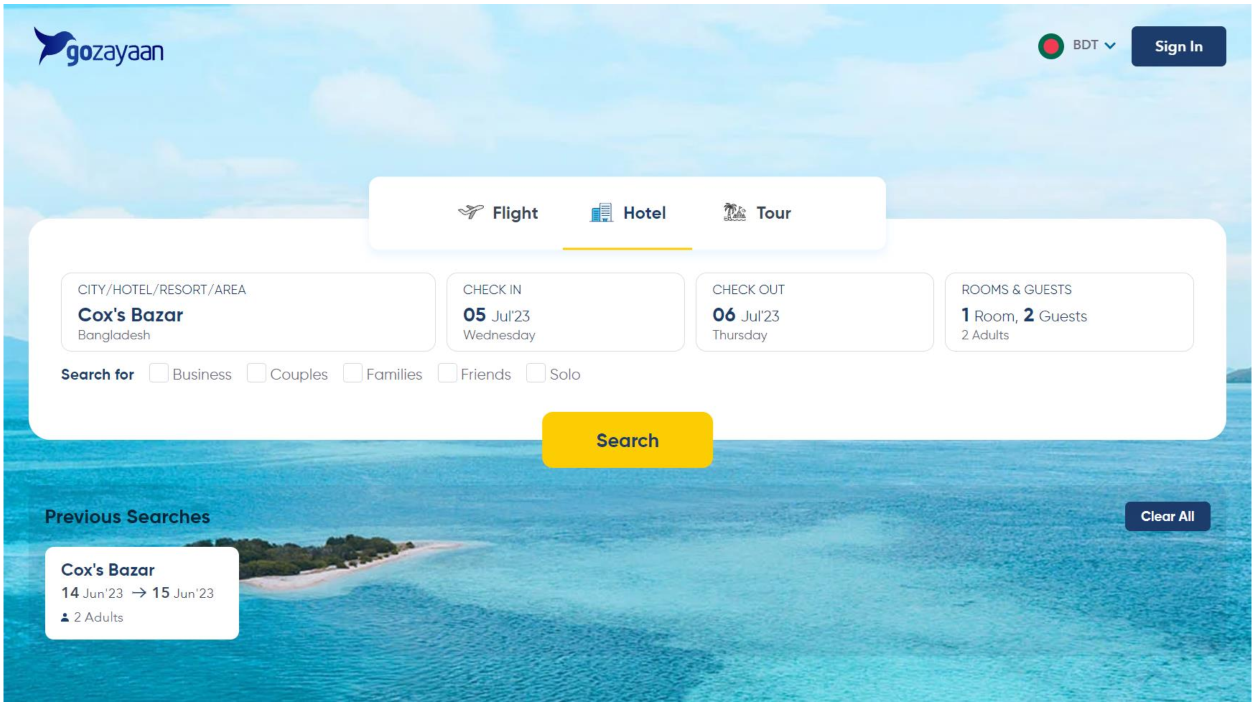


First 100 customers will get 10% extra discount. So book now and take it on the opportunity.

Need a hotel, resort or go to tour?

Book your dream hotel, resort, get flexible picnic packages from us

GoZayaan: GoZayaan another popular resort booking system is available in Bangladesh. The platform offers an extensive selection of resorts including eco-friendly and budget-friendly choices. It has a simple interface that allows travelers to easily search resorts by location, budget, or facilities. GoZayaan provides user reviews and ratings that help users choose the best resort. (Gozayaan, n.d.)



CITY/HOTEL/RESORT/AREA
Cox's Bazar
Bangladesh

CHECK IN
05 Jul'23
Wednesday

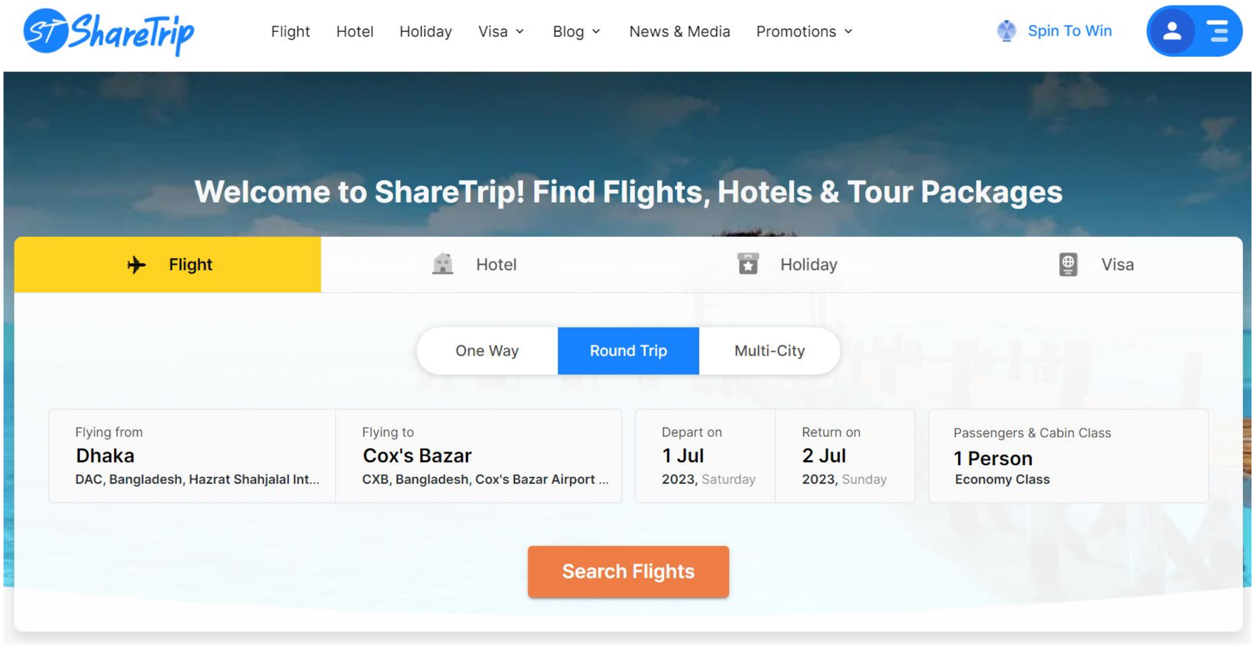
CHECK OUT
06 Jul'23
Thursday

ROOMS & GUESTS
1 Room, 2 Guests
2 Adults

Search for □ Business □ Couples □ Families □ Friends □ Solo

Cox's Bazar
14 Jun'23 → 15 Jun'23
2 Adults

Sharetrip: Sharetrip connects travellers with resorts across Bangladesh. The platform offers an easy booking process, which allows users to search through resorts of all types and book them with ease. Platform provides detailed resort profiles including descriptions, photos, and amenities. (Sharetrip, n.d.) offers both a payment gateway that is secure and customer service to help ensure an easy booking.



Flying from
Dhaka
DAC, Bangladesh, Hazrat Shahjalal Int...

Flying to
Cox's Bazar
CXB, Bangladesh, Cox's Bazar Airport ...

Depart on
1 Jul
2023, Saturday

Return on
2 Jul
2023, Sunday

Passengers & Cabin Class
1 Person
Economy Class

ResortFinderBD: ResortFinderBD offers a simple interface with a wide range of resorts in Bangladesh. The system allows travelers to find resorts by location, date, and amenities. Users can compare prices, and book resorts according to budget. ResortFinderBD provides ratings and reviews from customers to assist users in making informed decisions.

These are the top resort booking platforms in Bangladesh. The platforms offer a variety of resorts and user-friendly interfaces. They also include additional features like customer reviews and secured payment options. The final decision is based on personal preferences, travel needs, and budget. You should explore all platforms to determine which one best meets your requirements for an unforgettable resort booking in Bangladesh.

3.3 Recommended Approach

The remaining systems are not perfect, according to an analysis of the leading solutions. There are some features and functions that have lost the public's demand as well as user satisfaction. Some features are available. Thoughts that are needed to make the system more interactive application. This feature must be present in any system that is proposed:

- Multi-language support and localization: To accommodate users of different languages, the system should be available both in Bengali as well as English.
- Local Payment Methods: By integrating popular payment methods such as bKash Rocket or Nagad into the system, users will be able to make transactions easily.
- Customizable search filters: Users should be able to customize the system's results by filtering them based on their preferences such as location, price, nearby attractions, and amenities.
- Users Reviews and Ratings - By incorporating a system of reviews and ratings for resorts, users will be able to make informed decisions by reading the feedback provided by previous guests. It is important to have this feature in order to build trust locally.
- Local Attractions, Activities and Tourist Spots: Information about local attractions, tourist sites, and specific activities for different resort locations can be helpful to users in planning their itinerary and making the most out of their trip.

- Offer Exclusive Seasonal Packages or Discounts. Working with resorts on exclusive packages and discounts can encourage more bookings, and even motivate users to select certain destinations at specific times.
- Safe and Trustworthy Platform: The implementation of robust security measures and protocols for data protection, along with secure payment gateways, instils confidence amongst users and assures them that personal information and financial details are safe.

Incorporating these features, which are tailored specifically to Bangladesh's context, can improve the experience of the users, encourage local tourism and, ultimately, attract both domestic and foreign travellers.

CHAPTER 4 – METHODOLOGY

4.1 What to use

The most popular software development methods include the spiral model (v-model), rapid application development (RAD), and dynamic systems design methodology. Software development is dominated by the spiral (v) model, rapid application development and dynamic systems design methodology. This article will discuss the pros and cons of three software development methods: rapid application (RAD), Dynamic Systems Development Method (DSDM), or waterfall model. (Whitefiled, n.d.)

4.2 Choosing Methodology

Several factors effects while choosing the correct methodology. User requirement is needed to choose the correct methodology. So, keeping in my mind I have chosen to use the DSDM methodology for this project. DSDM provides very flexible stages in development. Also, both client and user involvement in the development stage makes the customer very satisfied. Another advantage is it is also easy to customize any changes if required. (Agile Business, n.d.)

4.3 Why to use

DSDM offers a wide range of methods for projects. It helps us by reducing project time and also improves the quality of the project. Although, it can be challenging in making changes by collaborating with stakeholders, none the less we this is the best methodology to use in our resort booking system.

4.4 Sections of Methodology

DSDM has many stages and features. The six stages of DSDM for this resort booking system are: DSDM (Dynamic Systems Development Method) software methodology includes six distinct steps designed to facilitate an efficient and cost-effective software project development life cycle.

When followed closely, each stage fosters iterative and incremental development while meeting both user needs and business requirements.

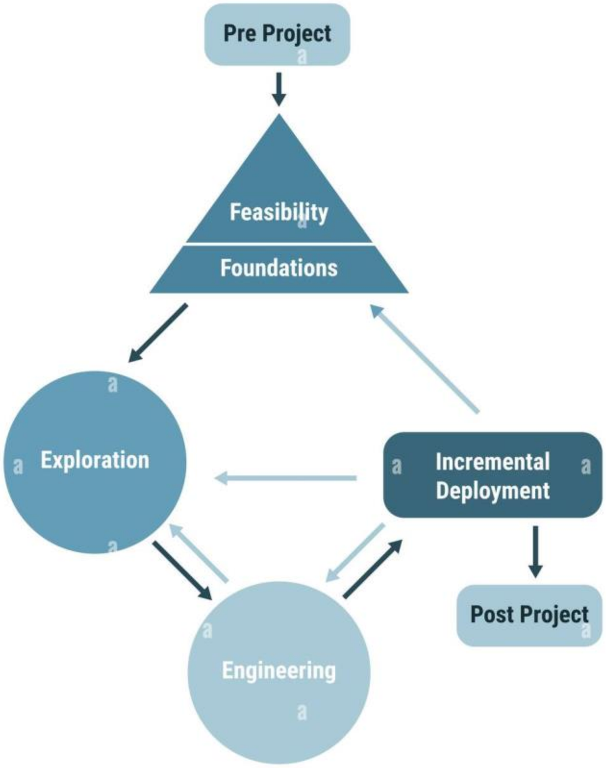


Figure 1: DSDM Process

1. Feasibility and Business Study Stage:

At this point in the project cycle, project team members conduct an exhaustive feasibility and business study assessment on their software project to ascertain its viability and economic viability as well as any associated benefits or risks. Ultimately this stage aims to determine whether to continue with further stages or whether its work should cease altogether.

2. Functional Model Iteration Stage:

At this step in the iterative development process, a project team initiates its iterative approach by designing an initial functional model demonstrating key software functionalities that serves both as an early blueprint and a means to gather user feedback early in the development lifecycle.

Subsequent iterations allow further refining of this functional model so as to meet evolving user and business requirements and needs.

3. Design and Build Iteration Stage:

At this point, attention turns towards translating the functional model into a tangible design and developing software for implementation. Working collaboratively as a project team, members design the architecture, specify system components and develop individual feature or module designs before beginning the build process which transforms designs into functional code through regular iterations that enable continuous feedback to ensure the project aligns with user requirements as well as technical constraints.

4. Implementation Stage:

Once software reaches maturity, project teams begin the next stage: its deployment into production environments and rigorous testing by an outside vendor or internal staff to verify functionality, performance and reliability before user acceptance testing commences to ensure it fulfils user needs and expectations; any identified issues must then be rectified before moving onto subsequent steps of implementation.

5. Deployment Stage:

At this critical juncture in development lifecycle management, project teams carefully deploy software applications or services to their target users or customers. In doing so, they consider factors like user training, data migration, system integration, organizational change management and transition between development to operational use - thus guaranteeing its successful adoption within user communities and communities at large.

6. Post-Project Stage:

The final phase of DSDM software methodology involves reviewing and assessing completed projects. Teams gather to conduct in-depth evaluations on the performance, usability and user satisfaction of software applications developed. They compare this success against their initial business case to determine areas for enhancement; then lessons learned during this stage are captured for use during future software development endeavours for continuous improvement and organizational learning.

Adherence to these six stages ensures an iterative and collaborative software development approach, supporting agility, adaptability and alignment with business objectives and user requirements.

4.4 Implementation Plan

In this stage, the final system is made available for the customer. Here, release principles, project settings, and schedule programs are established for the system. The system should be made available for beta testing. Some of the modules of the project are made available on demand of the user.

CHAPTER 5 – PLANNING

5.1 Project Scope

The Resort Booking System will include features such as user registration and authentication, resort and room information management, online booking and payment processing, reservation modification and cancellation, availability tracking, and reporting.

5.2 Project Planning

This project's objective is to design and deploy an end-to-end resort booking system for Bangladesh's hospitality industry using the Laravel framework, an open-source PHP framework renowned for its reliability. This project plan details key stages, activities and deliverables necessary for successfully developing and deploying such a system.

5.2.1 Work Breakdown Structure

Project breakdown structure provides visual, hierarchical, and deliverable-oriented to manage a project on time. It is very important for a successful project delivery. WBS is mainly used for project's possibility, expense, and schedule estimation. (workbreakdownstructure, n.d.) Not only this we can define project boundaries, measure project completion, identify milestones. In this this resort management system we have visualize the WSB below:

Table 1: Work Breakdown Structure

	Task Name	Duration	Start	End
01	Users should be able to create accounts and log in securely.	7 days	05/02/2023	12/02/2023
02	Resort staff should be able to add, update, and remove resort and room information.	5 days	12/02/2023	17/02/2023
03	Room availability status should be maintained and updated in real time.	7 days	17/02/2023	24/02/2023

04	Customers should be able to search for available resorts and rooms based on their preferences.	9 days	24/02/2023	03/03/2023
05	Customers should be able to make bookings and proceed with secure online payments.	15 days	03/03/2023	18/03/2023
06	The system should generate booking confirmations and receipts.	15 days	18/03/2023	03/04/2023
07	The system should track the availability of rooms in real time and prevent double bookings.	12 days	03/04/2023	15/04/2023
08	Resort staff should be able to update room availability status based on reservations.	10 days	29/04/2023	09/05/2023
09	User authentication should be implemented to ensure secure access to the system.	14 days	09/05/2023	23/05/2023
10	The system should provide reports and analytics on booking trends, occupancy rates, and revenue.	7 days	23/05/2023	30/06/2023
11	The Website should be responsive on both mobile and computer device	12 days	30/06/2023	12/06/2023
12	A proper security system should be implemented	8 days	12/06/2023	20/06/2023
	Total	120 days		

5.2.2 Resource Allocation

Table 2: Resource Allocation

Task Name		Duration	Resource
01	Users should be able to create accounts and log in securely.	7 days	User, Developer
02	Resort staff should be able to add, update, and remove resort and room information.	5 days	Developer, User, Admin
03	Room availability status should be maintained and updated in real time.	7 days	Developer, User, Admin

04	Customers should be able to search for available resorts and rooms based on their preferences.	9 days	Developer, User
05	Customers should be able to make bookings and proceed with secure online payments.	15 days	Developer, User
06	The system should generate booking confirmations and receipts.	15 days	Developer, Analyst
07	The system should track the availability of rooms in real time and prevent double bookings.	12 days	Developer, Analyst
08	Resort staff should be able to update room availability status based on reservations.	10 days	Developer, User
09	User authentication should be implemented to ensure secure access to the system.	14 days	Developer, User, Admin
10	The system should provide reports and analytics on booking trends, occupancy rates, and revenue.	7 days	Developer, Analyst
11	The Website should be responsive on both mobile and computer device	12 days	Developer, Analyst, User, Admin
12	A proper security system should be implemented	8 days	Developer, Analyst, User, Admin
	Total	**120 days**	

5.2.3 Time Boxing

Timeboxing is a project management technique utilized during software development that allocates fixed periods, or "timeboxes," for certain activities or tasks. By adopting timeboxing in software development teams can focus on finishing work within specified deadlines while improving time management practices and increasing productivity. (Clokify, n.d.) With timeboxing being employed in iterations of development iterations, collaboration among team members increases, creating rhythm and cadence as milestones are met and fostering an overall sense of accomplishment when milestones are accomplished. (Simplilearn, n.d.)

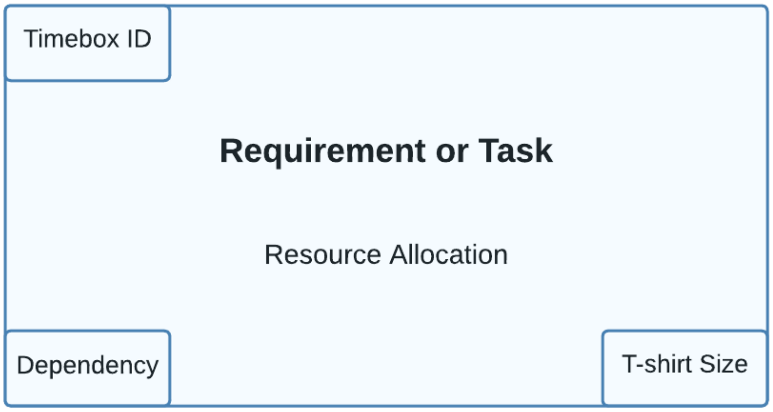
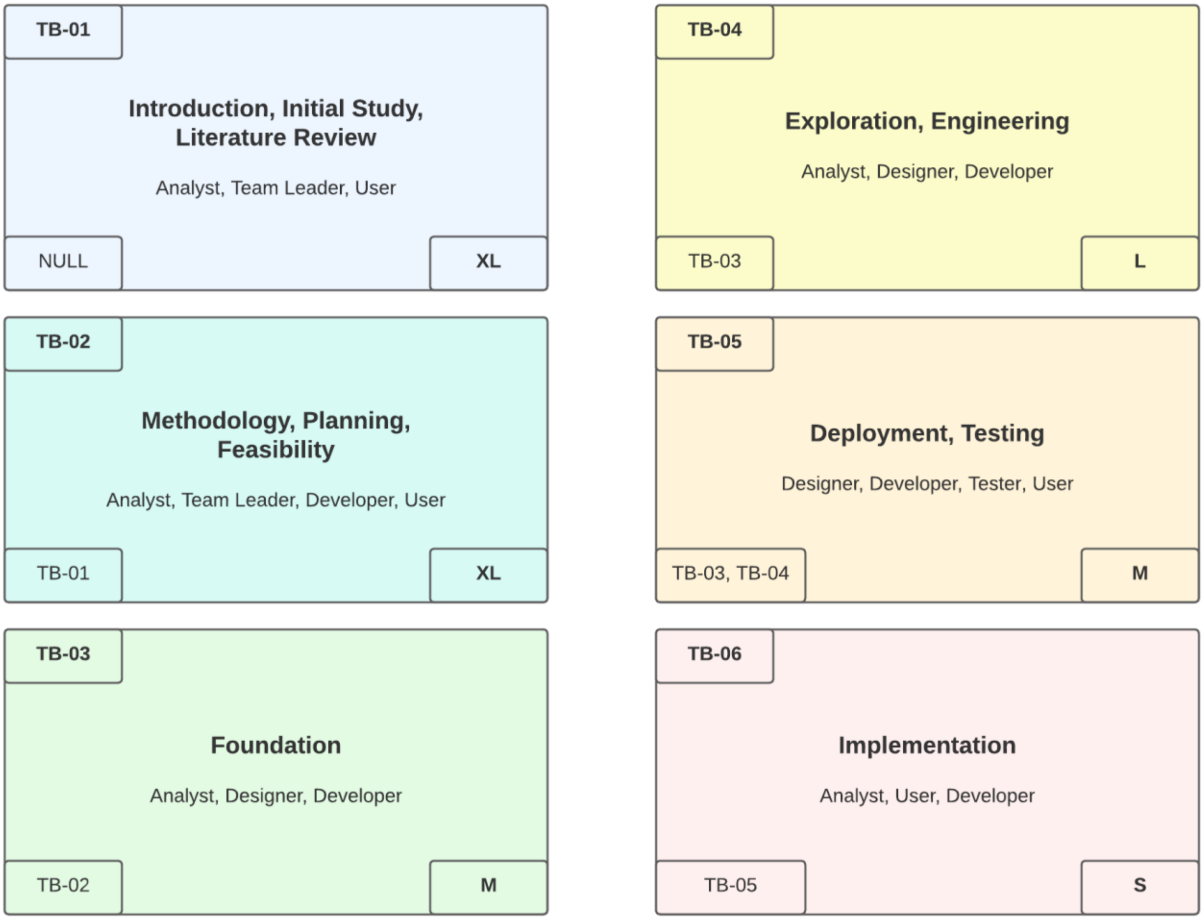


Figure 2: Sample of Timeboxing

Time-Boxing List



Timebox ID	Requirement or Task	Resource Allocation	Dependency	T-shirt Size
TB-01	Introduction, Initial Study, Literature Review	Analyst, Team Leader, User	NULL	XL
TB-02	Methodology, Planning, Feasibility	Analyst, Team Leader, Developer, User	TB-01	XL
TB-03	Foundation	Analyst, Designer, Developer	TB-02	M
TB-04	Exploration, Engineering	Analyst, Designer, Developer	TB-03	L
TB-05	Deployment, Testing	Designer, Developer, Tester, User	TB-03, TB-04	M
TB-06	Implementation	Analyst, User, Developer	TB-05	S

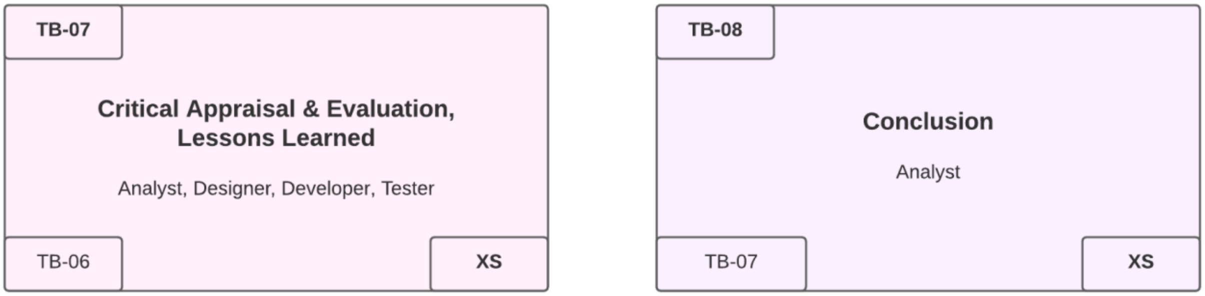

TB-07
Critical Appraisal & Evaluation, Lessons Learned
Analyst, Designer, Developer, Tester
TB-06
XS
TB-08
Conclusion
Analyst
TB-07
XS

Figure 3: Time-Boxing List

Table 3: Timeboxing by task

Time Box	Task_Name	Duration	Resource
TB 1	Users should be able to create accounts and log in securely.	7 days	User, Developer
TB 2	Resort staff should be able to add, update, and remove resort and room information.	5 days	Developer, User, Admin
TB 3	Room availability status should be maintained and updated in real time.	7 days	Developer, User, Admin
TB 4	Customers should be able to search for available resorts and rooms based on their preferences.	9 days	Developer, User
TB 5	Customers should be able to make bookings and proceed with secure online payments.	15 days	Developer, User
TB 6	The system should generate booking confirmations and receipts.	15 days	Developer, Analyst
TB 7	The system should track the availability of rooms in real time and prevent double bookings.	12 days	Developer, Analyst
TB 8	Resort staff should be able to update room availability status based on reservations.	10 days	Developer, User
TB 9	User authentication should be implemented to ensure secure access to the system.	14 days	Developer, User, Admin
TB 10	The system should provide reports and analytics on booking trends, occupancy rates, and revenue.	7 days	Developer, Analyst
TB 11	The Website should be responsive on both mobile and computer device	12 days	Developer, Analyst, User, Admin
TB 12	A proper security system should be implemented	8 days	Developer, Analyst, User, Admin

5.2.4 Gantt Chart

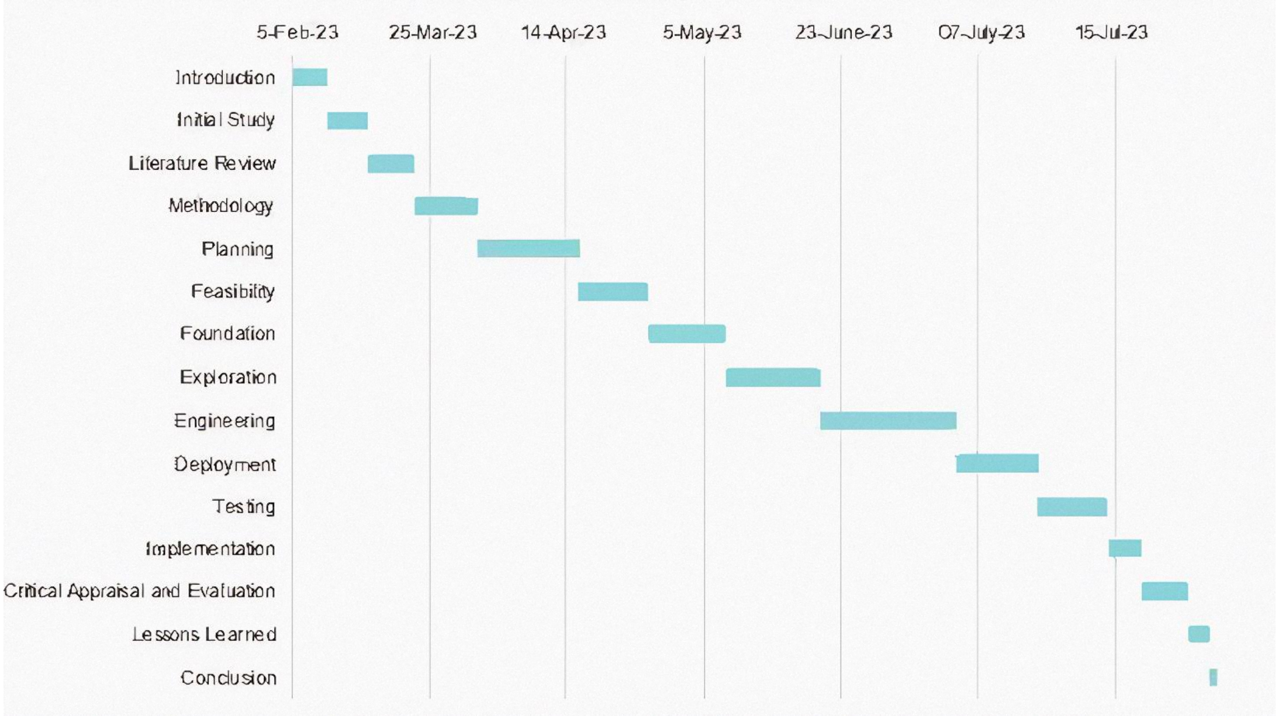


Figure 4: Gantt Chart

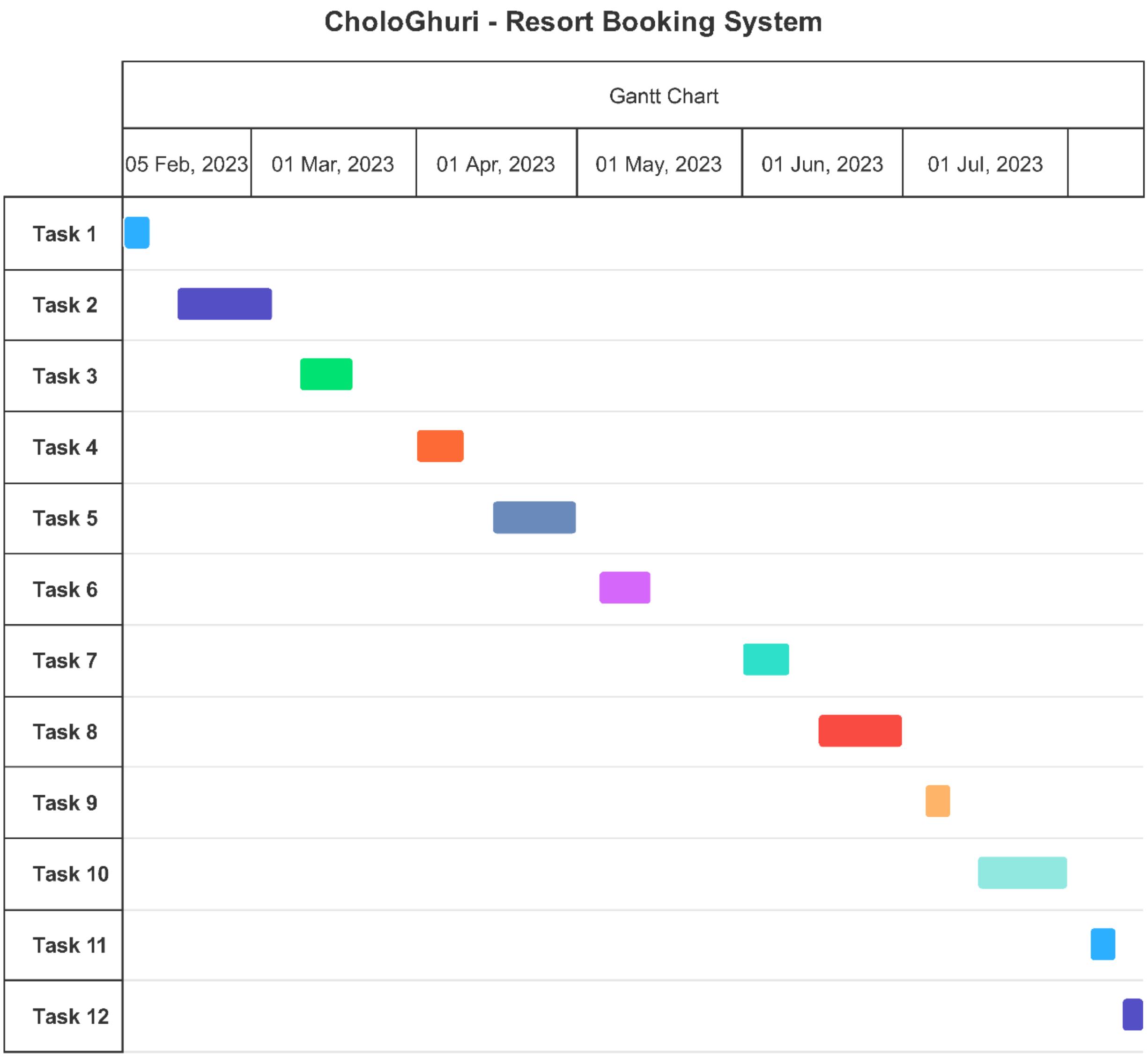


Figure 5: Gantt Chart by Task

CHAPTER 6 – FEASIBILITY

6.1 Operational Feasibility

Operational feasibility is concerned with the usability, ease of use, and required functionalities of the proposed system. In the case of the Resort Booking System, the system has been designed with proper validation and verification mechanisms for user inputs. It offers a user-friendly and intuitive navigation system, including a location-based search option. The system also incorporates an easy request option, making it effortless for users to book resorts. With the provision of valid credentials, users, including the admin, resort owners, and guests, can efficiently access their respective portals and perform their tasks with ease. (osarome, n.d.)

6.2 Technical Feasibility

The technical feasibility of the Resort Booking System is assessed based on the following aspects:

1. Technology Stack: The system is developed using Laravel 8, a widely adopted PHP framework, along with HTML5, CSS, and Bootstrap for front-end development. These technologies ensure compatibility across different web browsers and devices, offering a seamless user experience.
2. Server and Database: The system utilizes an Apache web server and MySQL database server, ensuring a stable and reliable infrastructure. These technologies have been chosen based on their popularity, community support, and compatibility with the selected development stack.
3. Mobile Compatibility: The system is designed to be responsive and compatible with mobile devices, allowing users to access and use the Resort Booking System on their smartphones or tablets. This enhances the convenience and accessibility for users who prefer mobile devices.
4. Security: The system incorporates an access control hierarchy to ensure secure access to sensitive data. Additionally, it follows best practices for data protection and implements necessary security measures to safeguard user information.

Based on the chosen technologies and their compatibility, the Resort Booking System is deemed technically feasible, as it leverages reliable and cost-effective resources.

Hardware

- ✓ ASUS Brand Laptop
- ✓ Intel Core i5 8th gen
- ✓ 1 TB HDD
- ✓ 16 GB DDR4 RAM
- ✓ Wi-fi Router

Software

- ✓ Xampp Server
- ✓ Microsoft Word
- ✓ Brave Browser
- ✓ Google Firebase
- ✓ Windows 10 (Operating System)
- ✓ Visual Studio Code Editor

Database

- ✓ MySQL

Technology

- ✓ Client-Side
- ✓ HTML 5
- ✓ CSS 3
- ✓ JavaScript
- ✓ jQuery 3.6.2
- ✓ Ajax

Server-Side

- ✓ PHP 7.8
- ✓ Laravel 8
- ✓ Apache

6.3 Economic Feasibility

The economic feasibility of the Resort Booking System is assessed. This involves an analysis of the development costs, deployment and maintenance as well as potential returns and benefits.

1. Development Costs:
    - Development Team: The costs associated with the development team, including salaries, benefits, and any outsourcing expenses, are considered.
    - Software and Tools: The expenses for acquiring or licensing development tools, frameworks, and software libraries necessary for building the Resort Booking System are evaluated.
    - Infrastructure: The costs related to setting up the development environment, including hardware, servers, and networking infrastructure, are factored into the analysis.
2. Deployment and Operational Costs:
    - Hosting and Server: The expenses for hosting the Resort Booking System on a server, including monthly hosting fees, domain registration, and SSL certificates, are considered.
    - Maintenance and Updates: The costs associated with maintaining and updating the system, including bug fixes, security patches, and feature enhancements, are evaluated.
    - Support and Training: The expenses related to providing customer support, user training, and system documentation are included in the economic feasibility analysis.

3. Return on Investment:
    - Revenue Generation: The potential revenue streams from the Resort Booking System, such as booking fees, commissions, or subscription charges, are estimated based on market research and industry benchmarks.
    - Cost Savings: The anticipated cost savings resulting from process automation, reduced manual efforts, and improved efficiency are taken into account.

- Market Potential: The size of the target market, competitive analysis, and growth projections are considered to evaluate the revenue-generating potential of the system.

By conducting a comprehensive analysis of the development and operational costs alongside the potential returns, the economic feasibility of the Resort Booking System can be determined. It is essential to ensure that the projected benefits outweigh the costs over a reasonable period, demonstrating a positive return on investment.

Considering the growing demand for online resort bookings and the potential revenue streams associated with the Resort Booking System, it is expected to be economically feasible. However, it is important to regularly review and update the economic feasibility analysis to adapt to changing market conditions and ensure the system's long-term sustainability.

6.2 Cost Benefit Analysis

Cost-Benefit Analysis (CBA) is an assessment technique used to analyze potential costs and benefits associated with any software project or investment, helping determine whether its benefits outweigh costs and whether its financial viability. (investopedia, n.d.) Here is an easy step-by-step process to conducting cost-benefit analyses on software:

Table 4: Cost Benefit Analysis

Serial No.	Equipment Cost	1st Year	2nd Year	3rd Year	4th Year	5th Year	Cost in Total
1	Web-Application	৳1,20,000					৳1,20,000
2	Email, Domain & Hosting Cost	৳10,000	৳15,000	৳15,000	৳15,000	৳15,000	৳75,000

4	Desktop Application	৳50,000					৳1,25,000
5	Maintenance	৳50,000	৳55,000	৳55,000	৳55,000	৳55,000	৳2,75,000
6	Staff Expenses	৳25,000	৳25,000	৳25,000	৳25,000	৳25,000	৳1,25,000
Total		৳4,50,000	৳95,000	৳95,000	৳95,000	৳95,000	**৳6,35,000**

Table 6: Total Cost Estimation for project

Earnings in total:

Sr. No.	Earning Sector	1st Year	2nd Year	3rd Year	4th Year	5th Year	Earning in total
1	Software Selling	৳2,00,000	৳2,50,000	৳3,00,000	৳3,80,000	৳4,50,000	৳16,25,000
Total		৳2,00,000	৳2,55,000	৳3,00,000	৳3,80,000	৳4,50,000	৳16,25,000

Table 7: Earning estimation for project

Analysis of Cost/Benefit shows that by adopting this strategy, an organization will accrue a large sum of Taka each year through revenue generated. Implementation will prove its benefit as income grows steadily year over year proving it as worthwhile investment by any measure.

6.4 Market Research Analysis

While the previous feasibility analysis focused on the technical and economic aspects, market research analysis provides insights into the demand and potential impact of the Resort Booking System. Although the analysis provided here is specific to the previously mentioned project, the Resort Booking System can also be considered in a similar context.

According to market research, the demand for automated systems for managing resort bookings is increasing. With the growing tourism industry and the need for efficient booking processes, the Resort Booking System can serve as a valuable platform for resort owners and guests. The system's ability to automatically send notifications, manage bookings, and provide a centralized authority for oversight makes it an effective solution for the target market.

By offering a user-friendly interface, advanced search options, and streamlined booking processes, the Resort Booking System aims to meet the increasing demand for efficient resort bookings. It has the potential to enhance the overall experience for both resort owners and guests, making it a valuable addition to the market.

CHAPTER 7 – FOUNDATION

7.1 The Problem Area Identification

Software is designed to solve problems that arise in our everyday activities. It is important to understand the problem from the perspective of the users in order to build a successful project. Users play a crucial role in identifying the problem area, because we build software that is user-centric, and the information they provide is the basis for developing a suitable solution. There are many different techniques for requirement elicitation. Below are some of these techniques:

7.1.1 Interview

One-on-one interview refers to one of the primary sources of requirements-gathering technique which is mainly conducted face-to-face between the system analyst and the stakeholders or anyone else that would be involved in the system. We must maintain certain criteria before organizing an interview session. (Tutorials Point, n.d.) These include selecting a well-prepared person to interview who has experience with the new or current system and creating open-ended and closed questions. We have categorized a set of questions for users to the system that we propose.

- What kind of issues user face in searching for resort?
- Is it is cost effective?
- Is it convenient for users in online payment?
- Does the platform need automation?

7.1.2 Questionnaires

Surveys and questionnaires are great ways to gather information quickly by asking questions of users or stakeholders. When creating questionnaires, it is important to organize them by feature or project objectives. They can be short or MCQ questions. Below is a sample question structure,

Table 5: Question for Problem Identification

Question for Problem Identification			
Name		**Age**	
End User		**Gender**	

Question-01	When do you usually book a resort?
Answer	
Question-02	What features would you like to see improved?
Answer	
Question-03	What is your preferred method of receiving price updates?
Answer	
Question-04	Are you having any difficulties with your current circumstances?
Answer	

7.3 Specific Problem Area Identification

When it comes to resort booking systems in Bangladesh, there are a few notable problem areas that need attention. These issues, which often arise due to human error or technical limitations, can hinder the overall user experience and affect the efficiency of the system. Understanding these problem areas is crucial for improving the resort booking process and providing a seamless experience for travelers.

- Limited Availability Information:
- Inconsistent Pricing:
- Poor User Interface and Navigation:
- Limited Payment Options:
- Insufficient Customer Support:
- Language and Localization Challenges

7.4 Possible Solutions

As previously discussed in Bangladesh's resort booking system, Laravel PHP may provide effective solutions. Laravel is a widely popular PHP framework renowned for its elegant syntax, robust features and user friendliness; therefore, it may provide effective solutions. Here are a few possible approaches using Laravel:

Implement a Real-Time Update Mechanism: Implementing real-time availability updates is the cornerstone of an accurate resort booking system, Laravel offers features like broadcasting events and web sockets that enable instantaneous notifications of room availability changes for instant updates on room vacancy changes.

- Consistent Pricing: Laravel can help ensure consistent pricing by using secure payment gateways or APIs that access real-time pricing information from resort management systems and display prices that reflect actual charges - building trust while eliminating discrepancies.
- User-Friendly Interface and Navigation: Laravel's powerful templating engine, Blade, allows developers to design user-friendly user interfaces using Laravel templating capabilities - creating intuitive yet visually appealing user experiences for resort booking systems that enable easy user navigation through resort booking systems, searching resorts or amenities and booking them seamlessly for users.
- Variety of Payment Solutions: Leveraging Laravel's integration capabilities, developers can integrate multiple local and international payment gateways to provide users with various payment methods and currencies - providing flexibility and convenience to users while accommodating their preferred methods and currencies.
- Reliable Customer Support: Laravel's extensive ecosystem enables developers to seamlessly incorporate customer support features like live chat, ticketing systems and email notifications into resort booking systems using packages from Laravel or custom development - offering prompt and efficient support throughout the booking process for users.
- Laravel provides robust localization features that enable multilingual support. Developers can utilize Laravel's localization capabilities to translate their resort booking system into different popular languages spoken across Bangladesh ensuring users from varying linguistic backgrounds have equal access and utilize it efficiently for an enhanced overall user experience.

Utilizing Laravel PHP's features and capabilities, Bangladesh's resort booking system can overcome its problems more efficiently. Implementation of real-time availability updates, consistent pricing integration, user-friendly interfaces, diverse payment options, reliable customer support services and language localization can significantly enhance user experiences, making the booking process smooth and pleasurable for travelers.

7.5 Overall Requirement List

The project requirements can be categorized in two ways. These are:

- Functional
- Non-functional

7.5.1 Functional Requirement

✓ User Registration and Authentication

✓ Resort and Room Management

- Resort staff should be able to add, update, and remove resort and room information.
- Room availability status should be maintained and updated in real-time.

✓ Booking and Payment Processing

- Customers should be able to search for available resorts and rooms based on their preferences.
- Customers should be able to make bookings and proceed with secure online payments.
- The system should generate booking confirmations and receipts.

✓ Reservation Modification and Cancellation

- Customers should have the option to modify or cancel their reservations within a specified time frame.
- The system should handle reservation modifications, cancellations, and refund processing.

✓ Availability Tracking and Management

- The system should track the availability of rooms in real-time and prevent double bookings.
- Resort staff should be able to update room availability status based on reservations.

- ✓ Reporting and Analytics
  - The system should provide reports and analytics on booking trends, occupancy rates, and revenue.
  - Reports should be generated in a user-friendly format and accessible to authorized users.

7.5.2 Non-Functional Requirements

- ✓ Usability
  - The user interface should be intuitive, visually appealing, and easy to navigate.
  - Response times for search, booking, and payment processing should be fast and efficient.
- ✓ Security
  - User authentication and authorization should be implemented to ensure data security.
  - Payment processing should be handled securely, following industry best practices.
- ✓ Performance
  - The system should be able to handle concurrent user requests without significant performance degradation.
  - Response times should be optimized to ensure a smooth user experience.
- ✓ Constraints
  - The Resort Booking System will be developed using PHP as the programming language.
  - The system will utilize a MySQL database for data storage and retrieval.
  - The application will be deployed on a Linux server environment.

7.6 Technology to be implemented

Web technology is the ability to connect two or more computers over a network using an Internet connection. There are many different activities involved in the implementation of the system. There are also several options available for technology. A client-server or web-server program would work well.

7.6.1 Web Application

Web browsers are software programs that allow users to carry out specific tasks stored on servers. A fixed domain on the Internet allows the user to easily connect with the server and interact with commercials by filling out online forms, making bookings, purchasing, etc. This is part of the client/server system that allows multiple computers to be connected.

Web application features

- ✓ Flexibility in the remainder of the task
- ✓ You can reach us at all times and from any location
- ✓ There are no inconsistencies as everyone has the same version.
- ✓ You need an Internet connection to use the system
- ✓ Maintenance does not require any installation.
- ✓ Regulation for various devices

7.7 Recommendation and Justification

It is essential that the proposed Resort Booking System be accessible 24/7 throughout the world. The system should work on any internet-capable devices, such as a smartphone, tablet, laptop or desktop. It must also support multiple users simultaneously, making the web application a recommended solution. Client-server technology is unable to achieve this.

CHAPTER 8 – EXPLORATION

8.1 Full System Use Case Diagram

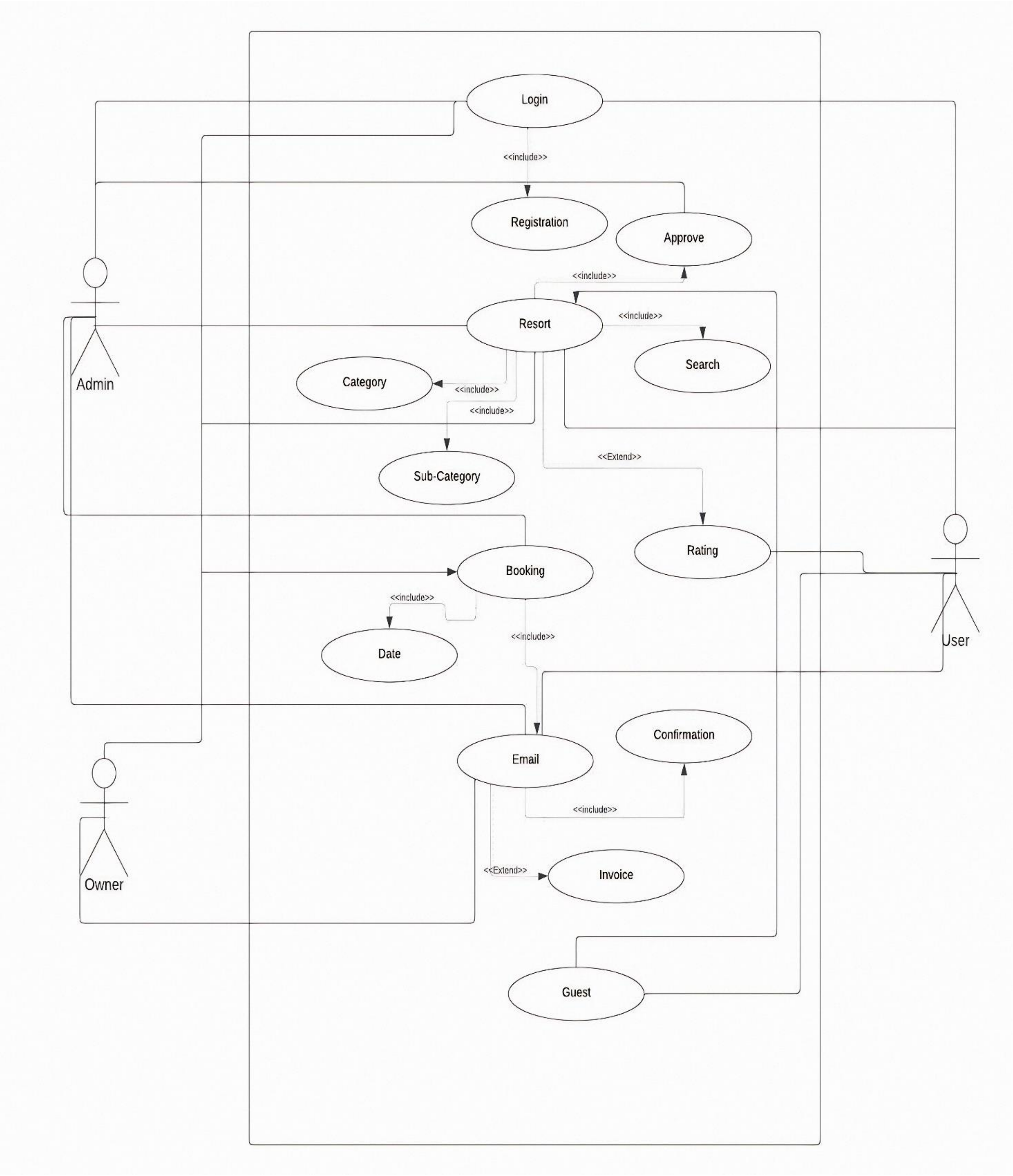


Figure 6: Use Case Diagram

8.2 ER Diagram

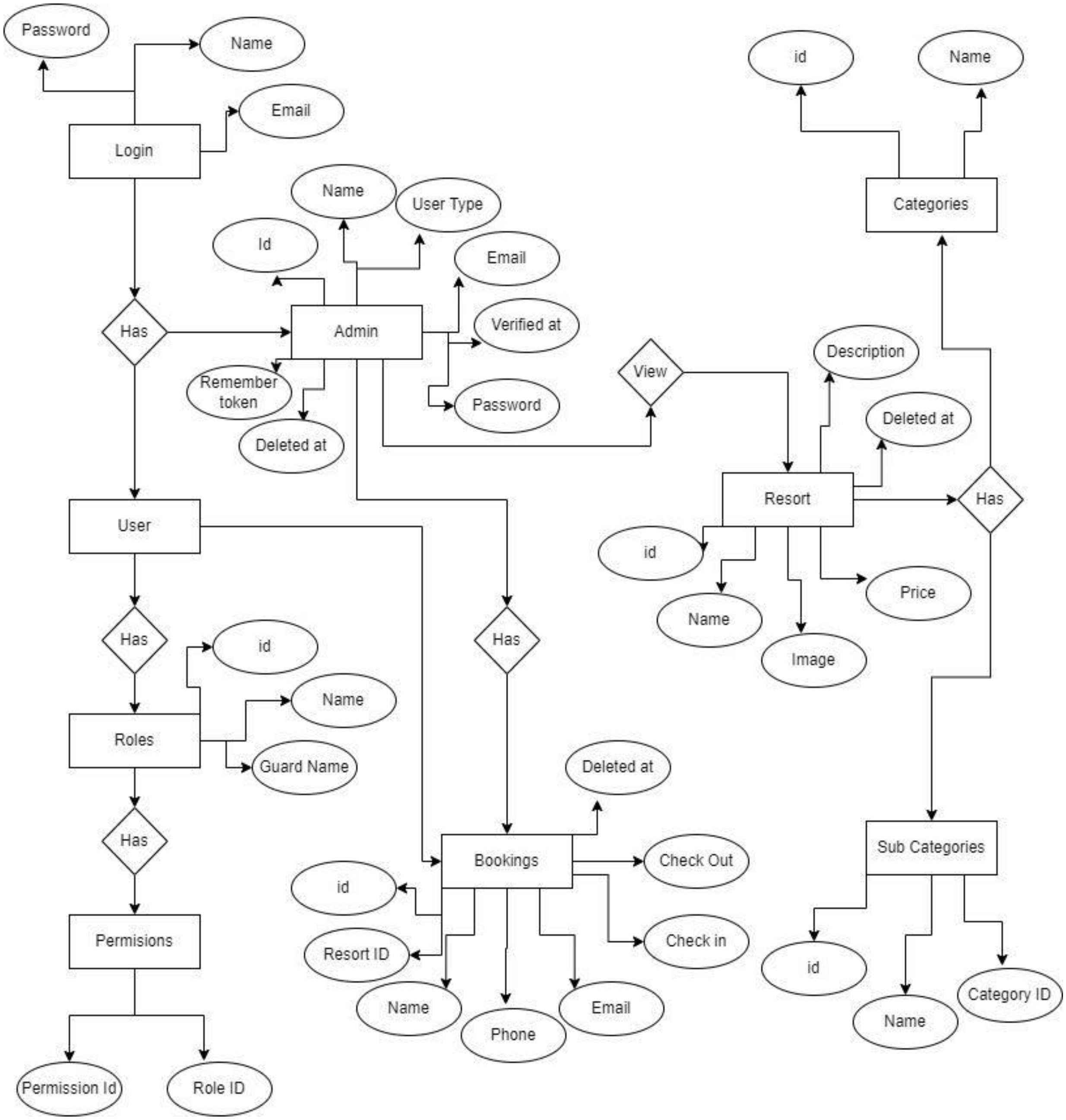


Figure 7: ER Diagram

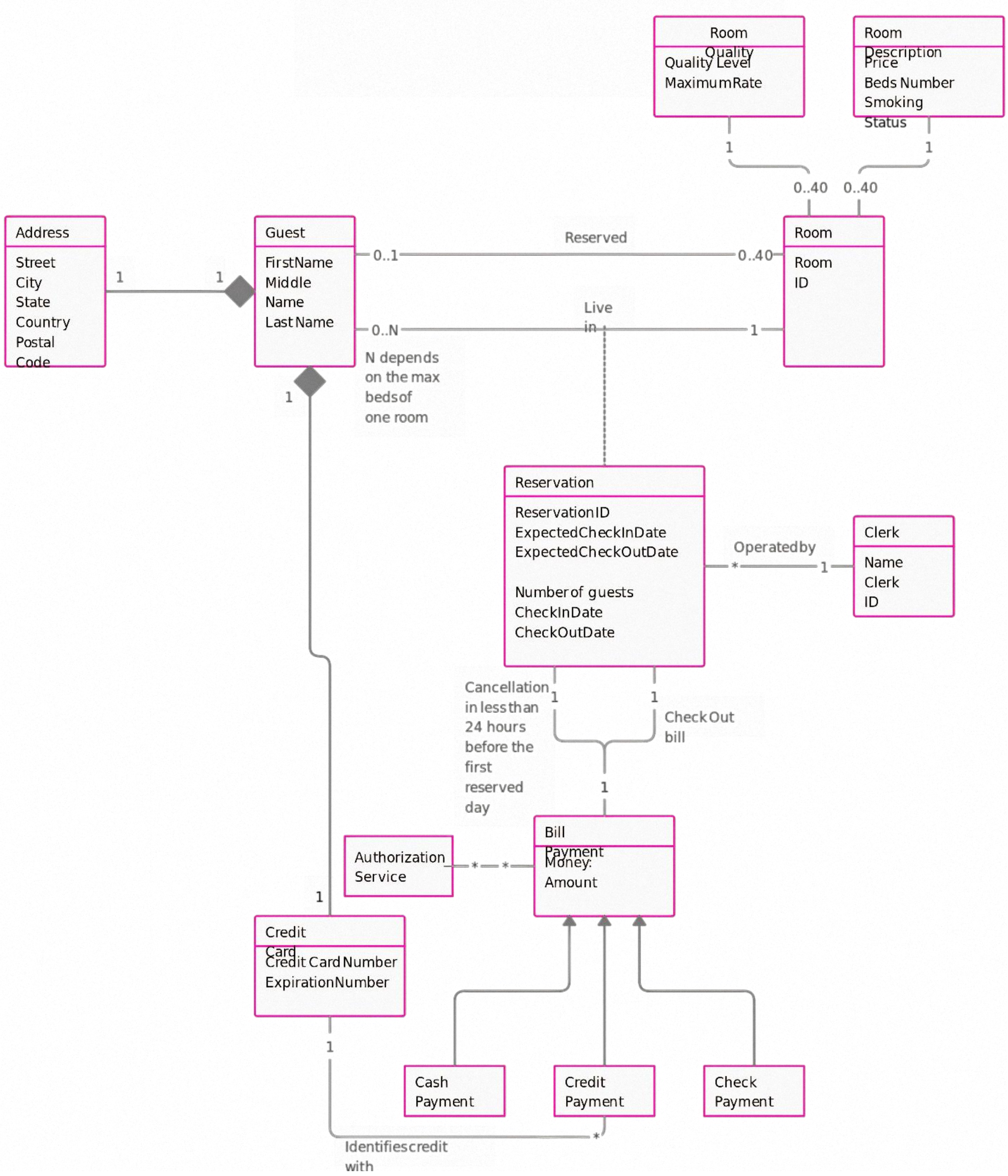


Figure 8: Class Diagram

8.3 Activity Diagram of Admin

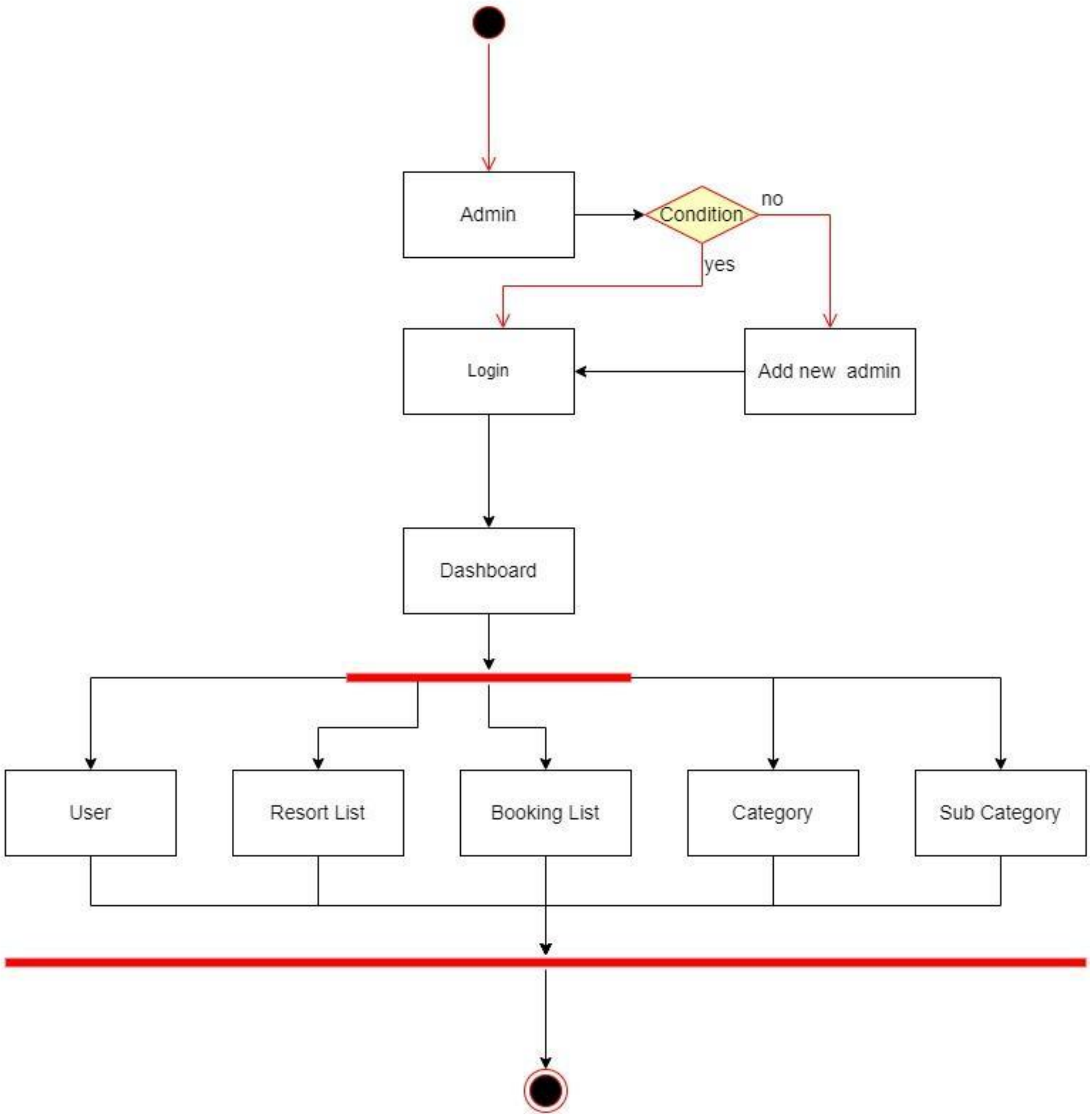


Figure 9: Activity Diagram (Admin)

8.4 Activity Diagram of User

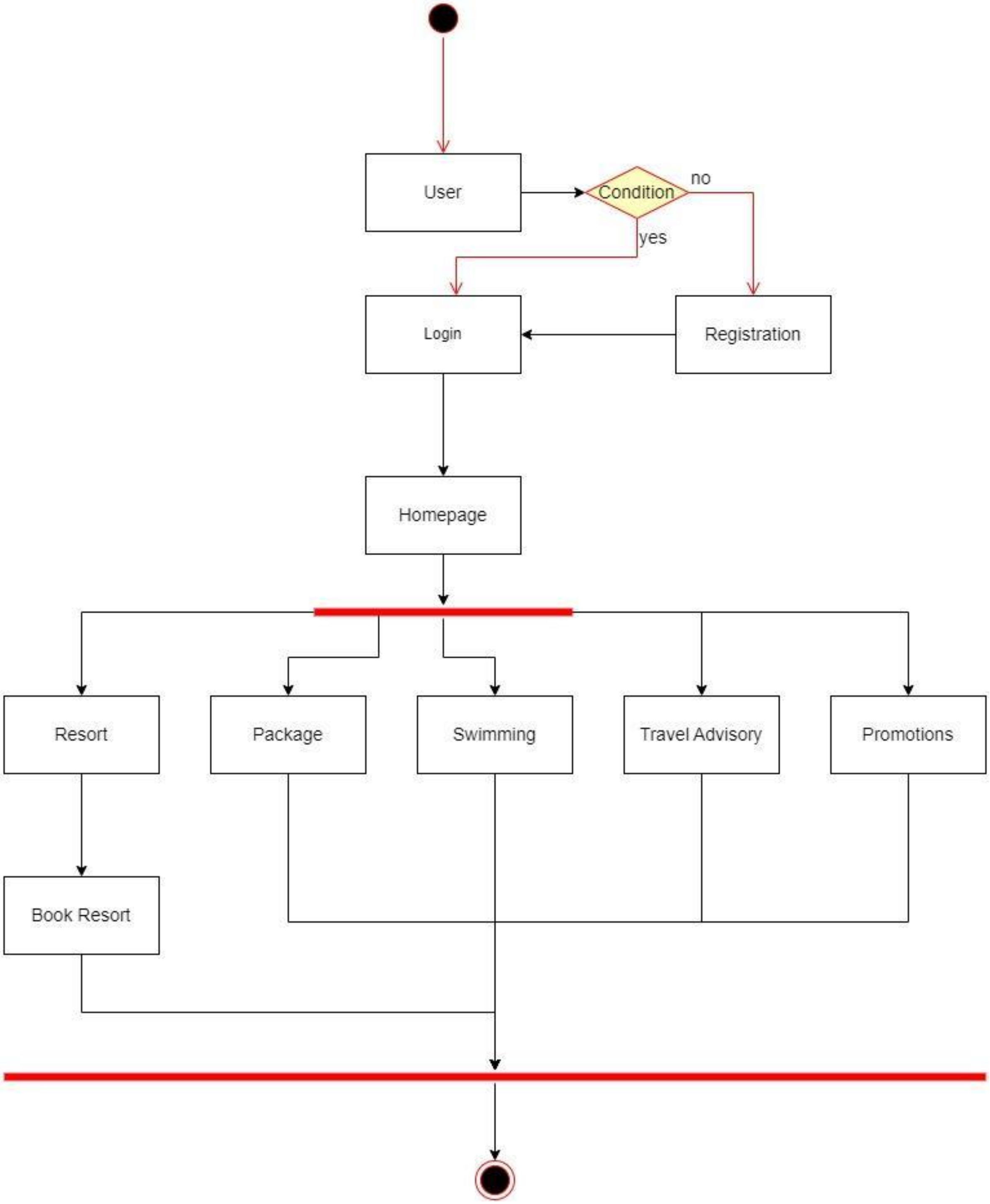


Figure 10: Activity Diagram (User)

Activity Diagram of Owner

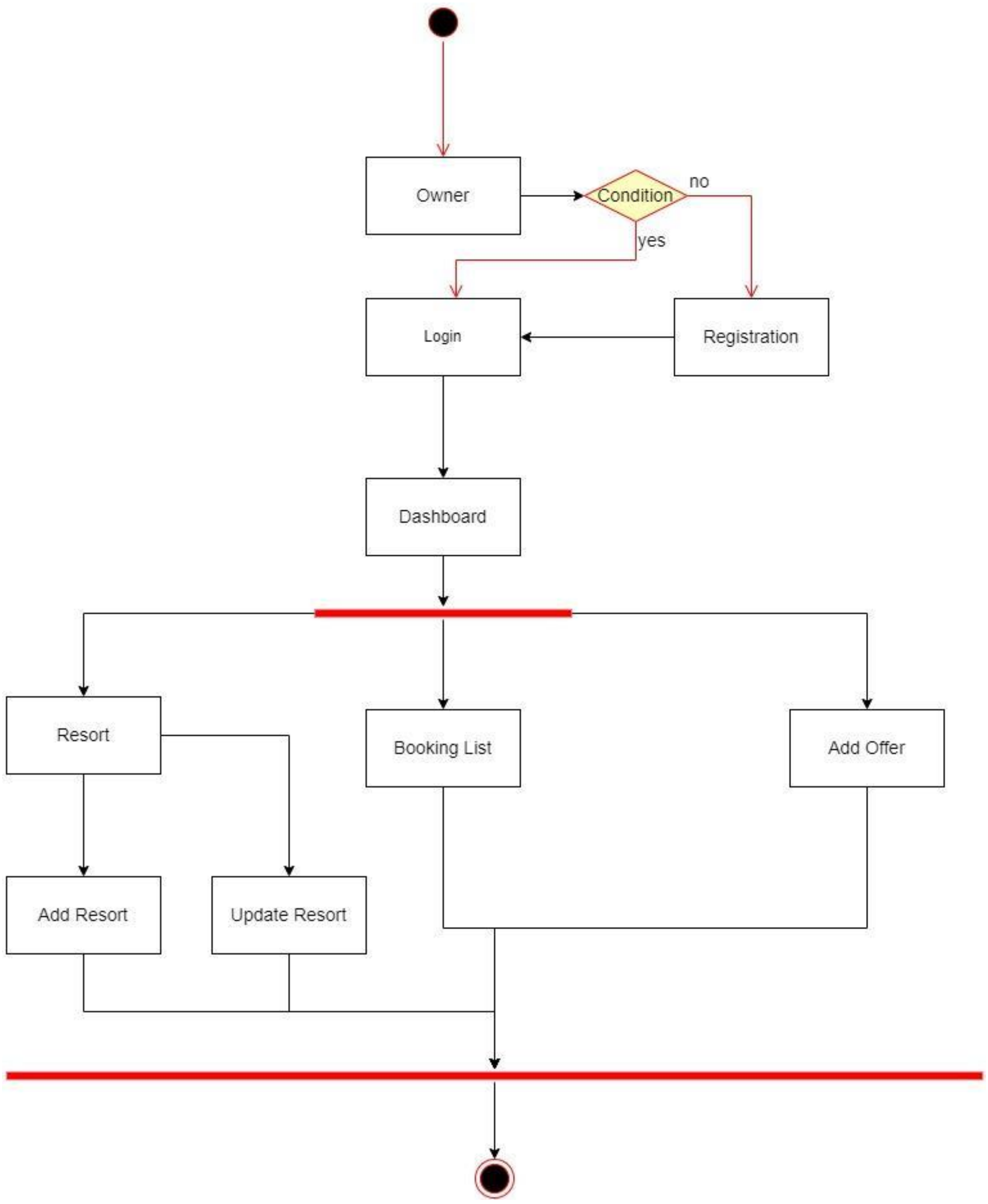


Figure 11: Activity Diagram (Owner)

8.5 Sequence Diagram of Admin

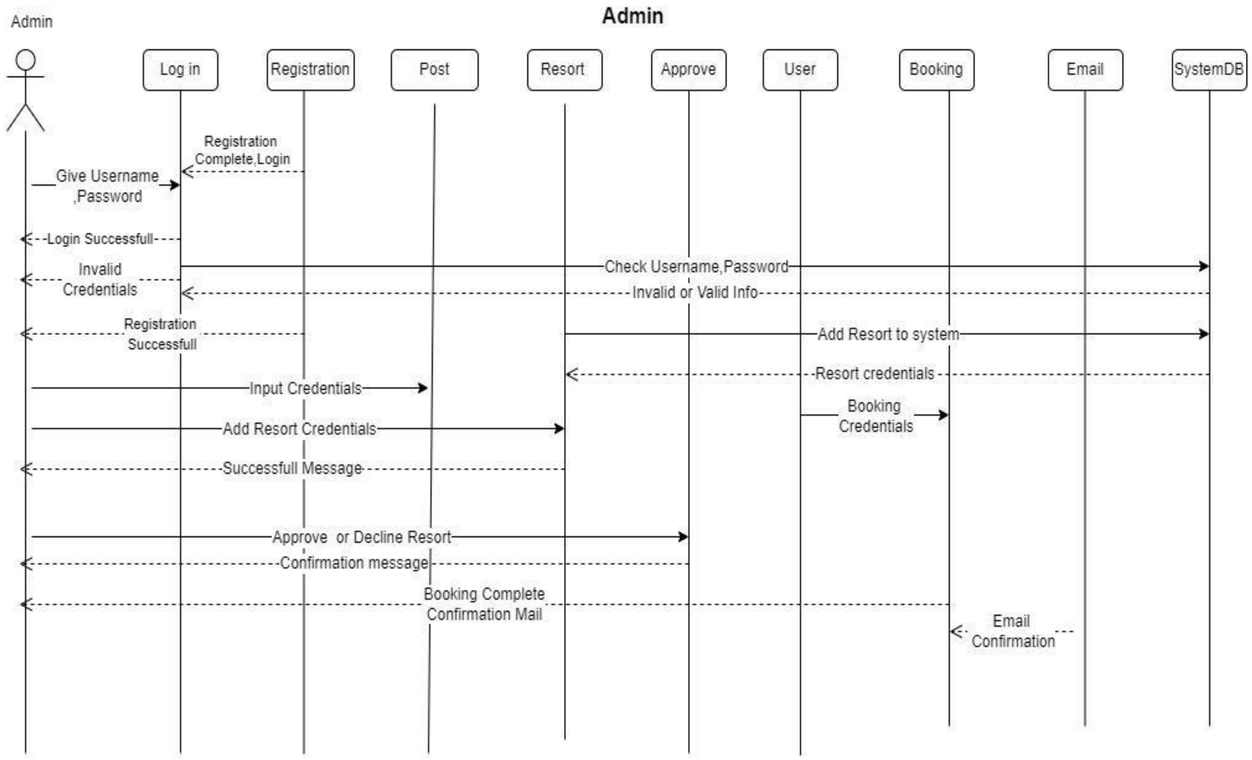


Figure 12: Sequence Diagram (Admin)

8.6 Sequence Diagram of Owner

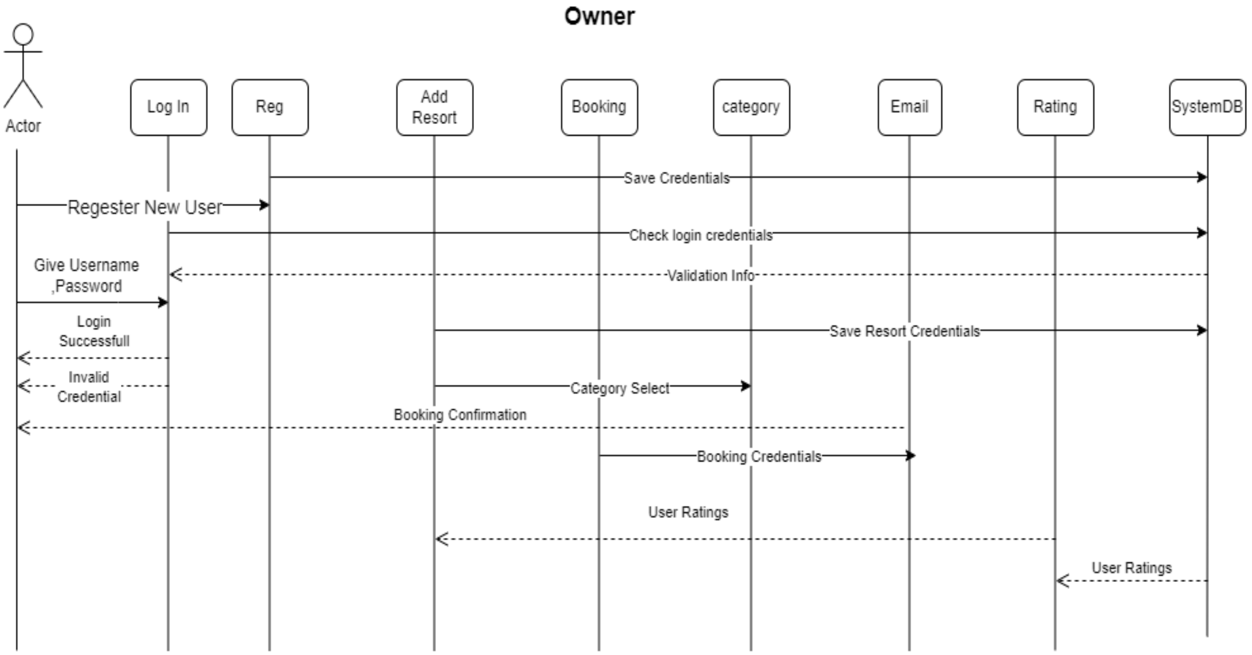


Figure 13: Sequence Diagram (Owner)

8.7 Sequence Diagram of User

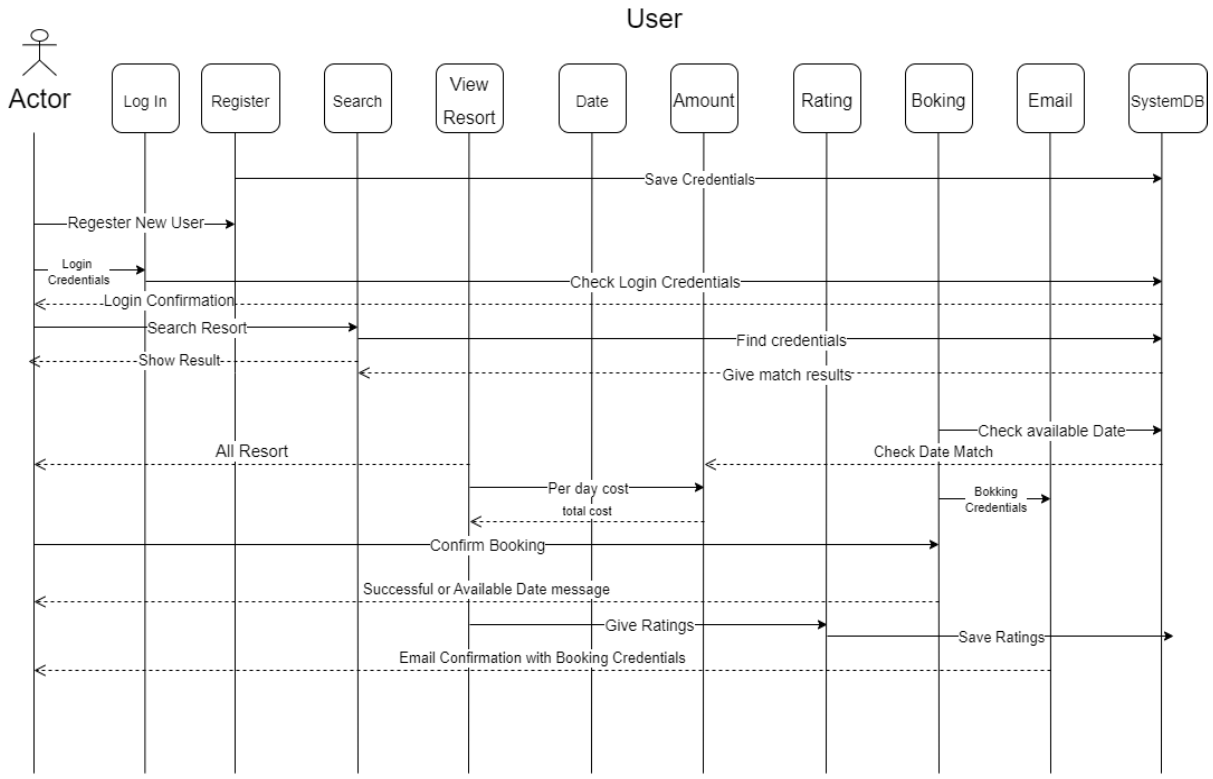


Figure 14: Sequence Diagram (Owner)

CHAPTER 9 – ENGINEERING

9.1 New System Modules

Many modules in the Resort Booking System are designed according to specific business needs. In this section, I will show you some of the core modules that were developed to explain how the system interacts after an action is taken by a user.

Table 6: Module for the login system

Serial No.	Action	Interaction
1	After completion of registration, the user will navigate to Login URL.	The system will get the user to the login page and it will display a form.
2	They form will be used to submit credentials for login.	The system checks if the credential of the user is valid or not.
3	The User will submit the login button.	If the credentials of the user match the database, the user will be redirected to the dashboard, else an error notice will be displayed.

Table 7: Module for managing resort

Serial No.	Action	Interaction

1	After login to the dashboard, the user creates on Manage resort	The system will get the user to the manage resort page and it will display a list of resort exits.
2	They user gets the select the resort to manage.	The system patch data from the database and show all available data.
3	The User adds or changes any required data.	If the data are according to policy then the system saves it.

Table 8: Module for Booking resort

Serial No.	Action	Interaction
1	After the user in the home page, they click the book resort page,	The system will redirect the user to the booking resort page.
2	The user enters the name of the resort or the location of it.	The system patch data from the database and show all available resort according to the search keyword.
3	The User selects the resort and enters the required information.	If the data are according to policy then the system saves it.

9.2 Use Case of Cholo Ghuri

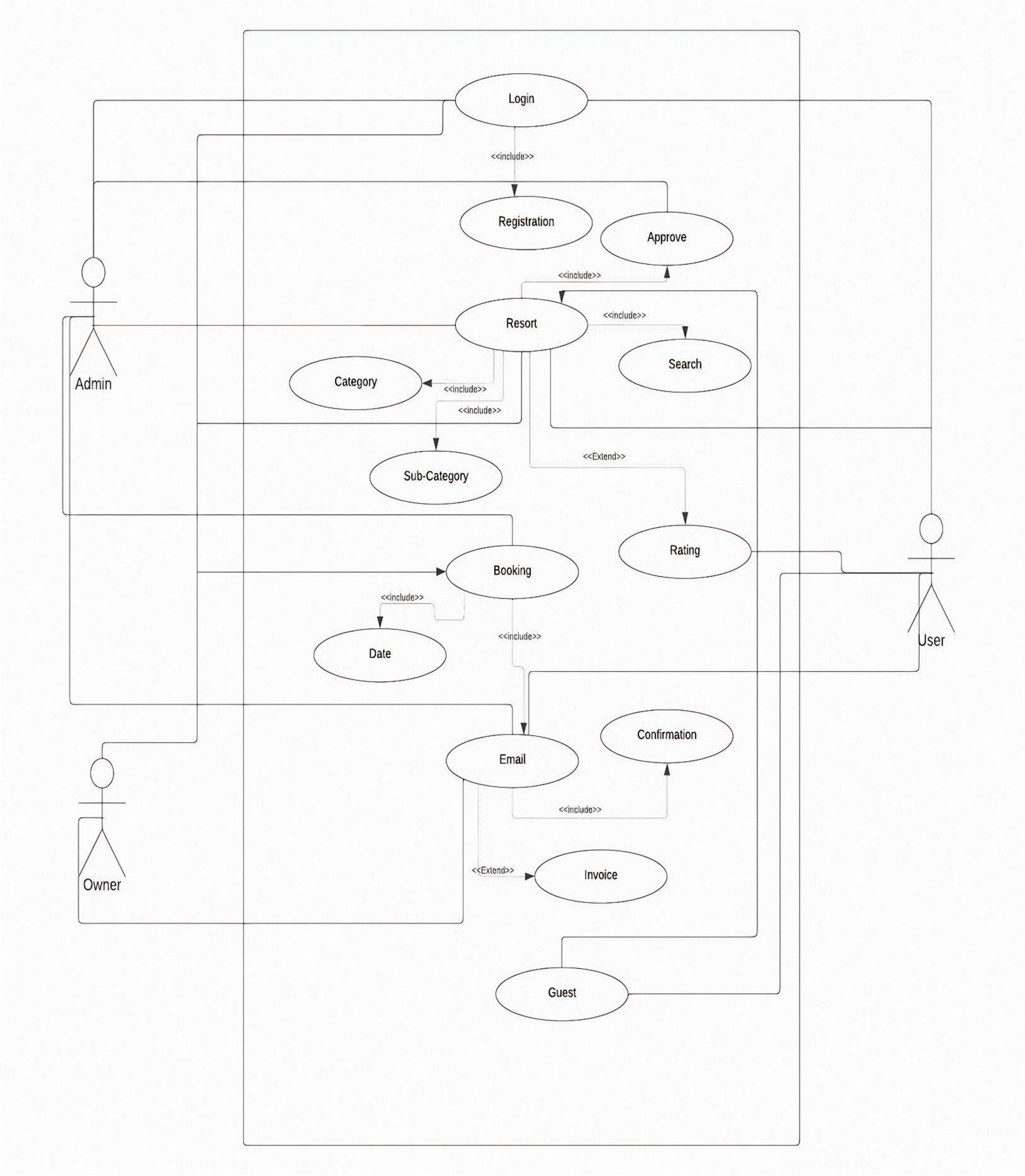


Figure 15: Use Case of Cholo Ghuri

9.8 System Interface Prototype

Hero Section

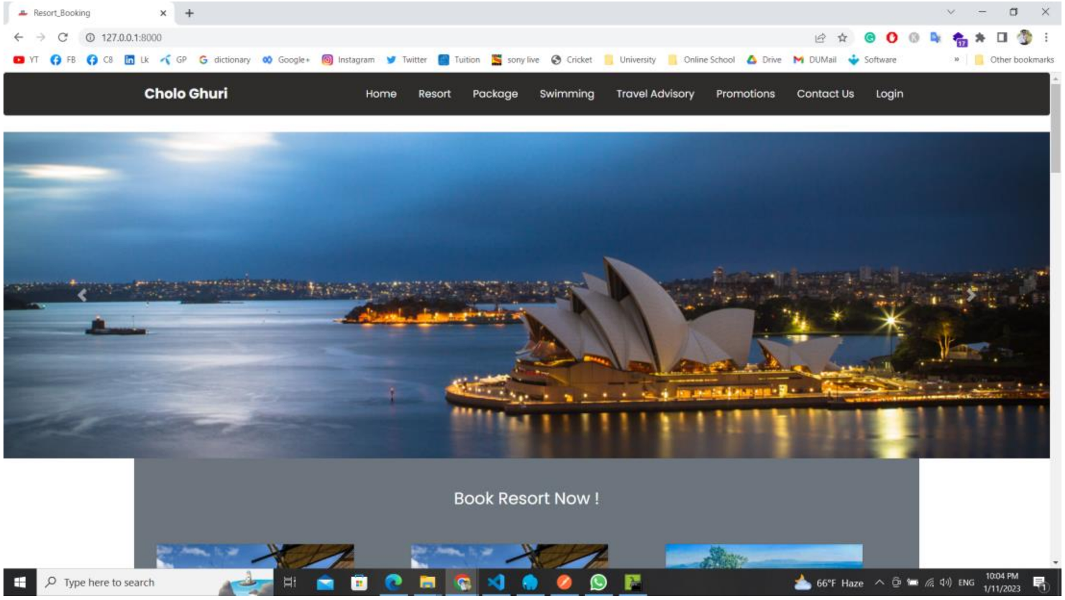
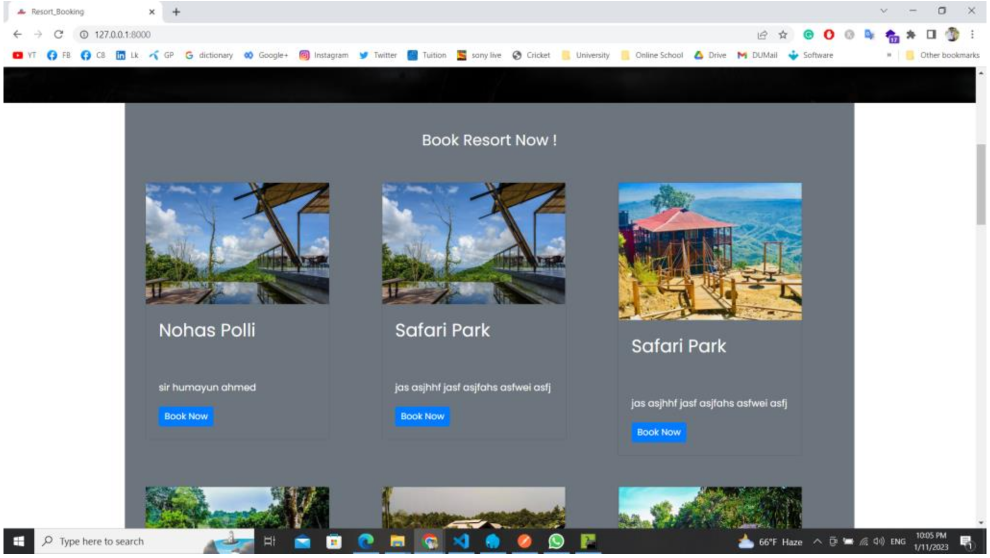


Figure 16: Hero Section

Homepage



Book Resort Now !

Nohas Polli

sir humayun ahmed

Book Now

Safari Park

jas asjhhf jasf asjfahs asfwei asfj

Book Now

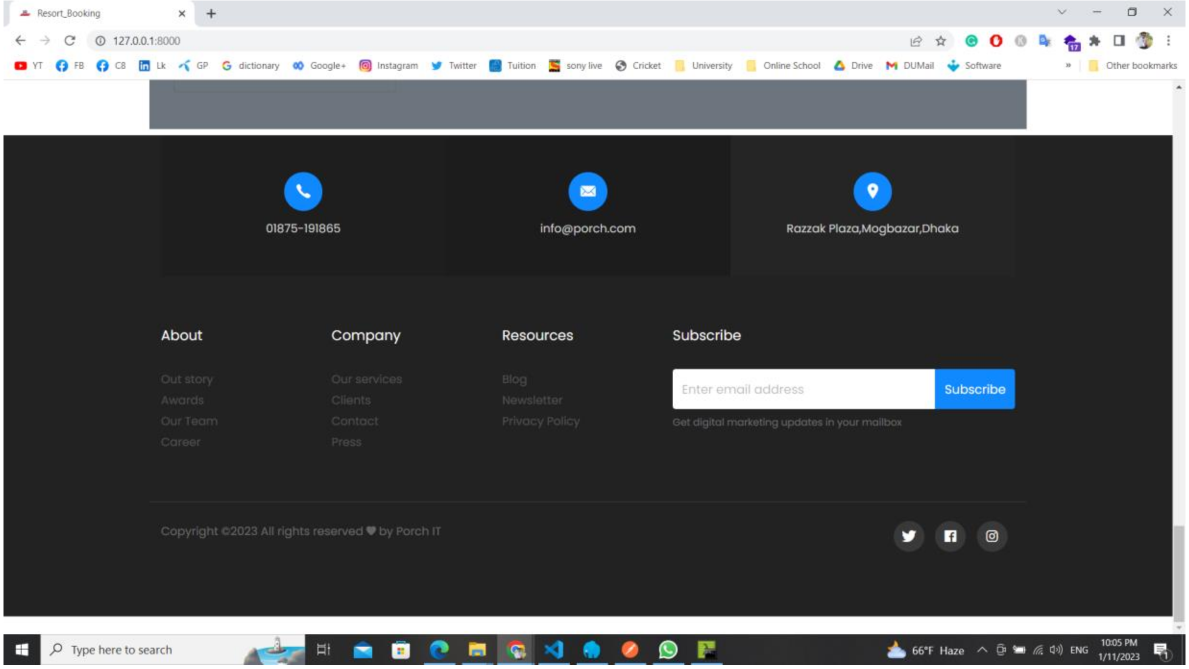
Safari Park

jas asjhhf jasf asjfahs asfwei asfj

Book Now

Figure 17: Homepage

Footer



01875-191865

info@porch.com

Razzak Plaza,Mogbazar,Dhaka

About

Our story
Awards
Our Team
Career

Company

Our services
Clients
Contact
Press

Resources

Blog
Newsletter
Privacy Policy

Subscribe

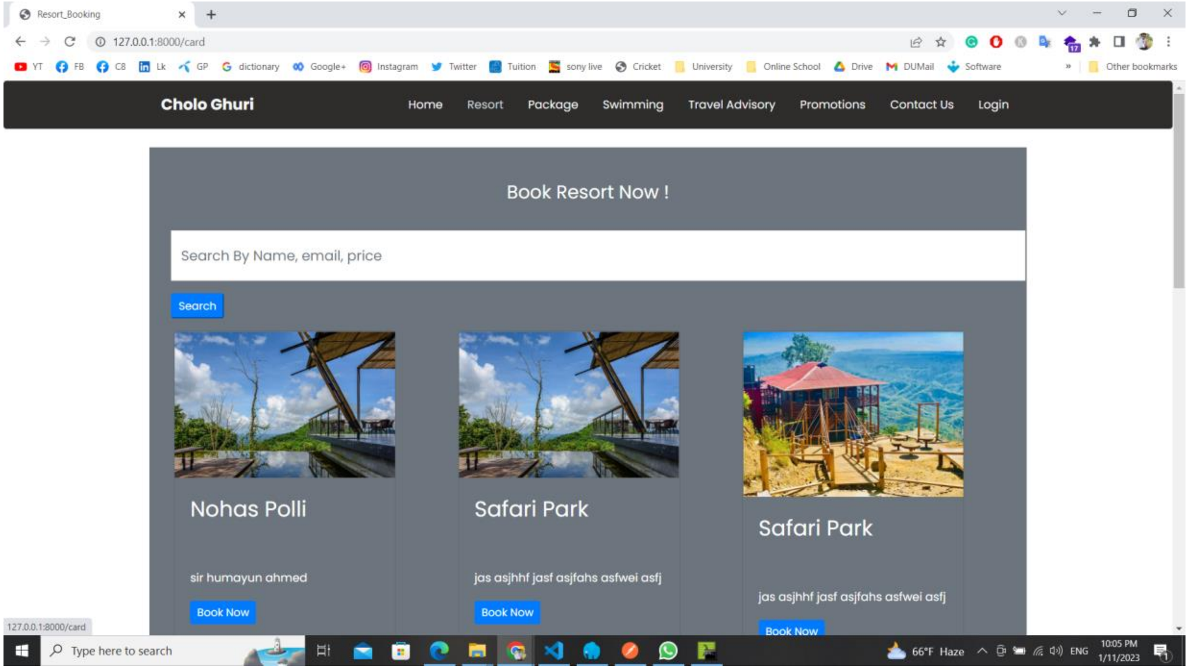
Enter email address Subscribe

Get digital marketing updates in your mailbox

Copyright ©2023 All rights reserved ♥ by Porch IT

Figure 18: Footer

Resort Page



Book Resort Now !

Search By Name, email, price

Nohas Polli

sir humayun ahmed

Safari Park

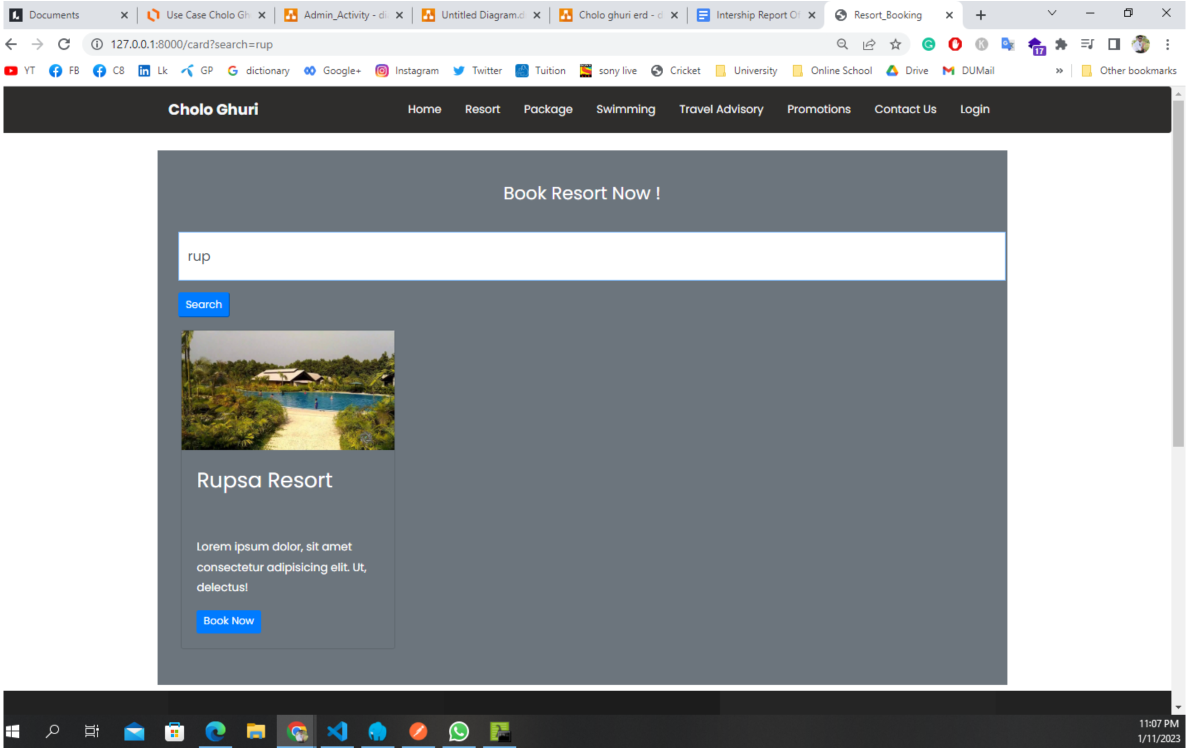
jas asjhhf jasf asjfahs asfwei asfj

Safari Park

jas asjhhf jasf asjfahs asfwei asfj

Figure 19: Resort Page

Resort Search Result



Book Resort Now !

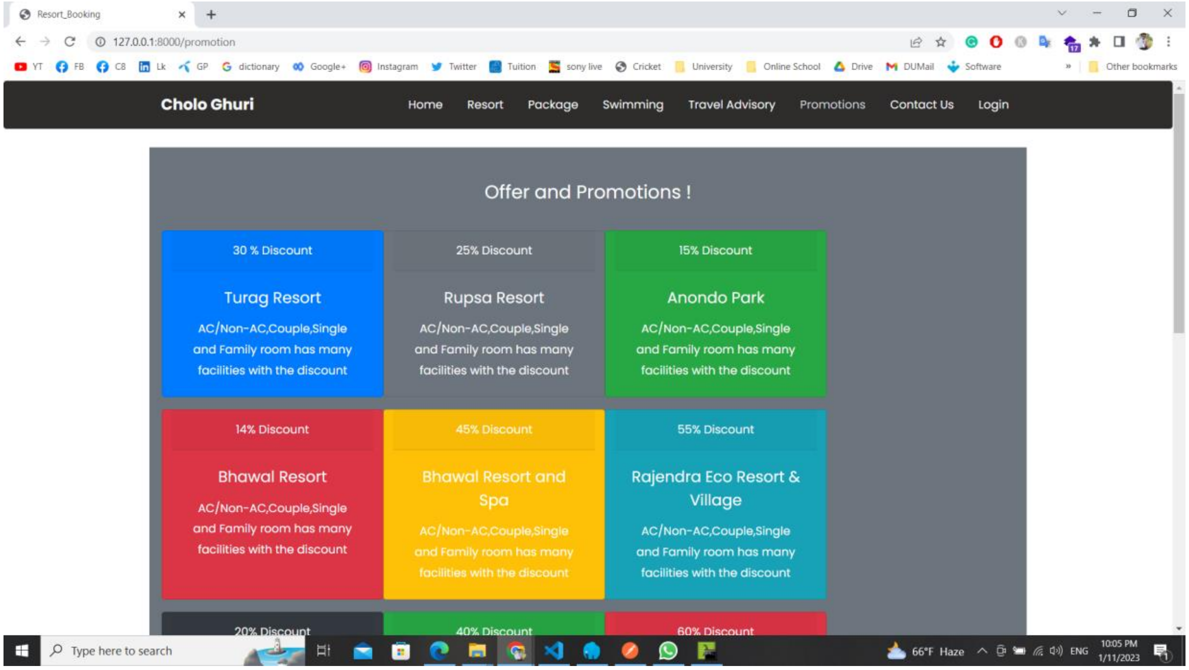
rup

Rupsa Resort

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Ut, delectus!

Figure 20: Resort Search Result

Promotions Page



Offer and Promotions !

30 % Discount

Turag Resort

AC/Non-AC,Couple,Single and Family room has many facilities with the discount

25% Discount

Rupsa Resort

AC/Non-AC,Couple,Single and Family room has many facilities with the discount

15% Discount

Anondo Park

AC/Non-AC,Couple,Single and Family room has many facilities with the discount

14% Discount

Bhawal Resort

AC/Non-AC,Couple,Single and Family room has many facilities with the discount

45% Discount

Bhawal Resort and Spa

AC/Non-AC,Couple,Single and Family room has many facilities with the discount

55% Discount

Rajendra Eco Resort & Village

AC/Non-AC,Couple,Single and Family room has many facilities with the discount

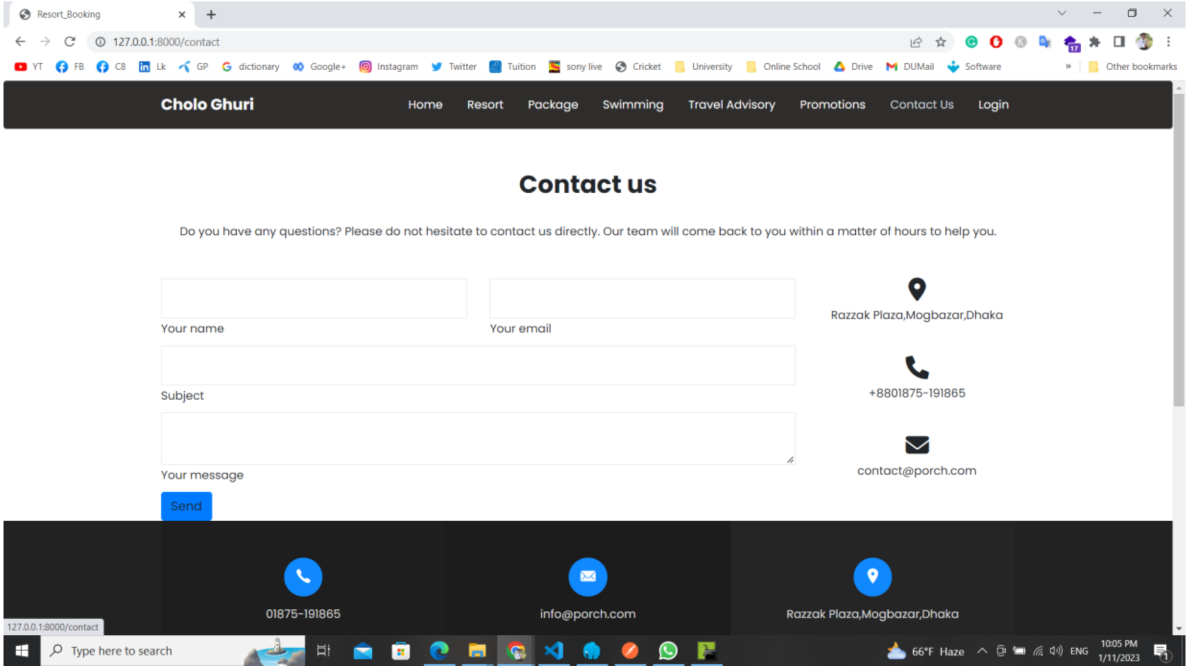
20% Discount

40% Discount

60% Discount

Figure 21: Promotions Page

Contact Us Page



Contact us

Do you have any questions? Please do not hesitate to contact us directly. Our team will come back to you within a matter of hours to help you.

Your name

Your email

Subject

Your message

Razzak Plaza,Mogbazar,Dhaka

+8801875-191865

contact@porch.com

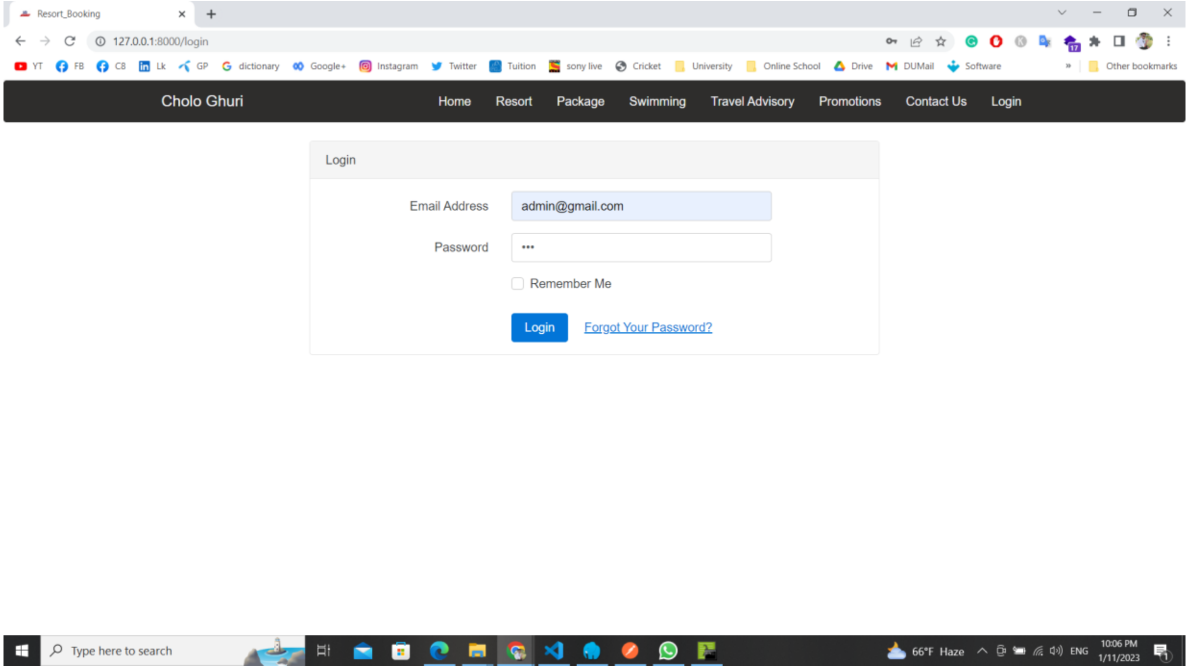
01875-191865

info@porch.com

Razzak Plaza,Mogbazar,Dhaka

Figure 22: Contact Us Page

Login Page



Login

Email Address admin@gmail.com

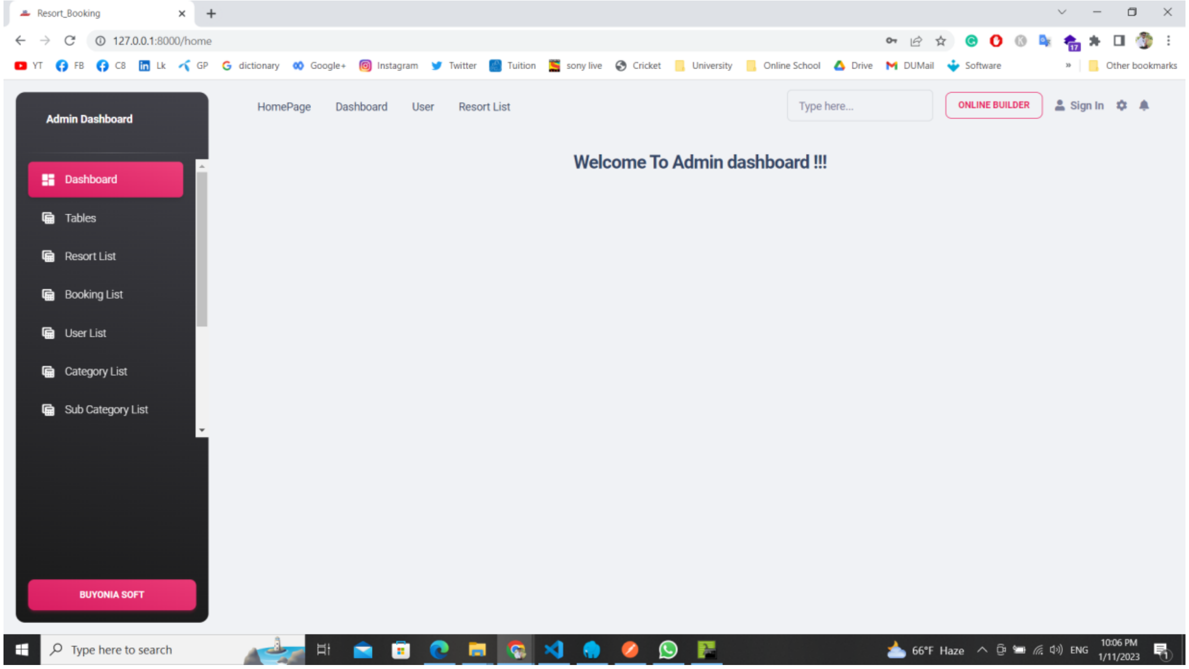
Password •••

□ Remember Me

Login Forgot Your Password?

Figure 23: Login Page

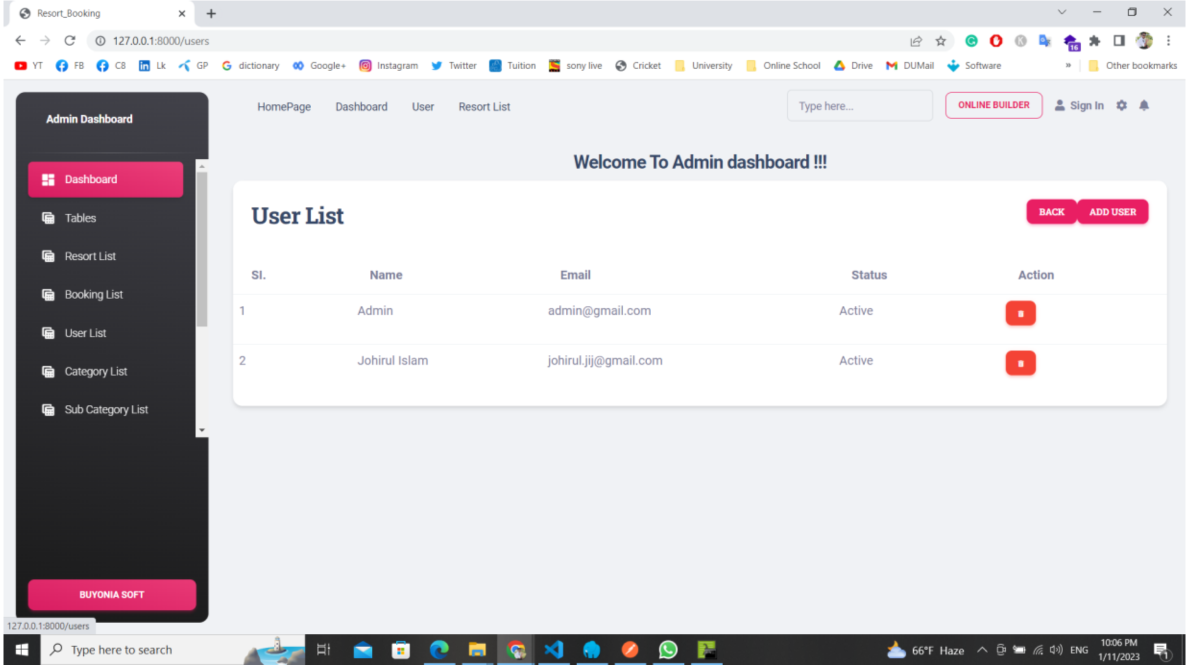
Admin Dashboard



Welcome To Admin dashboard !!!

Figure 24: Admin Dashboard

User List



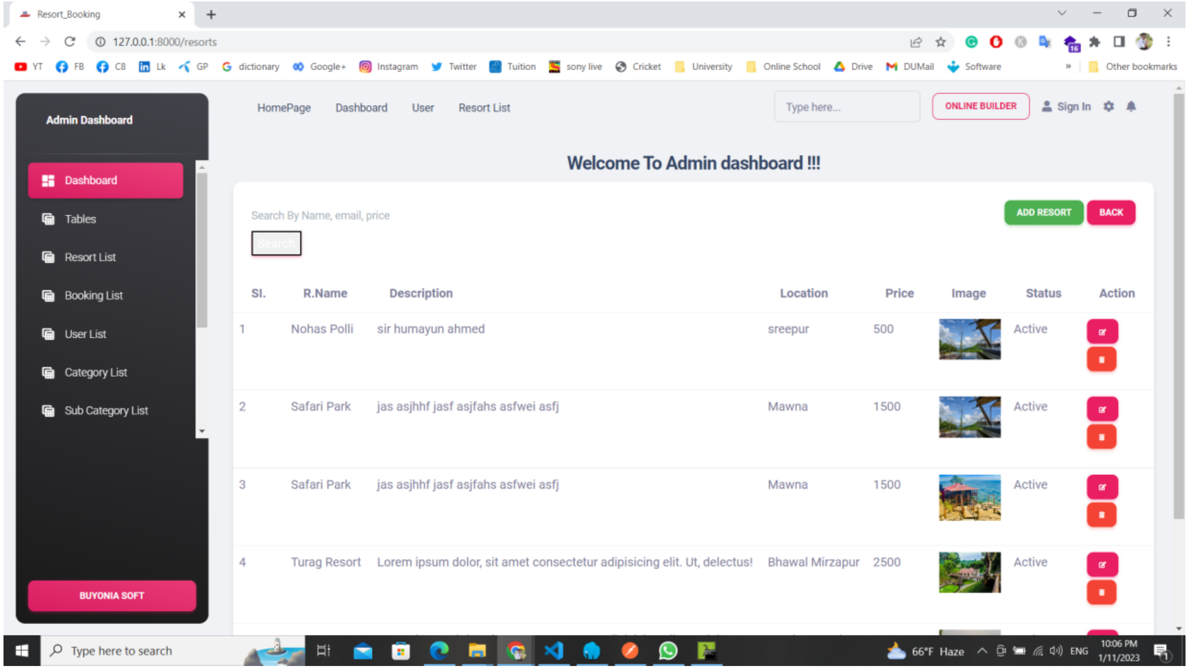
Welcome To Admin dashboard !!!

User List

Sl.	Name	Email	Status	Action
1	Admin	admin@gmail.com	Active	▪
2	Johirul Islam	johirul.jij@gmail.com	Active	▪

Figure 25: User List

Resort List

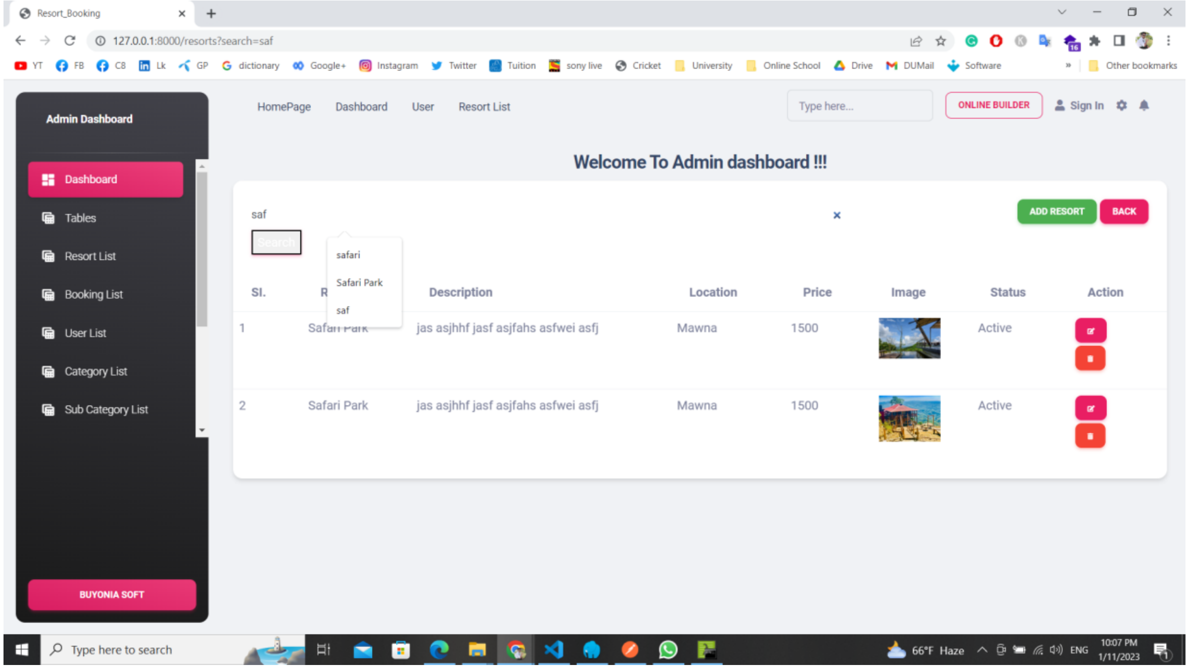


Welcome To Admin dashboard !!!

Sl.	R.Name	Description	Location	Price	Image	Status	Action
1	Nohas Polli	sir humayun ahmed	sreepur	500		Active	
2	Safari Park	jas asjhhf jasf asjfahs asfwei asfj	Mawna	1500		Active	
3	Safari Park	jas asjhhf jasf asjfahs asfwei asfj	Mawna	1500		Active	
4	Turag Resort	Lorem ipsum dolor, sit amet consectetur adipisicing elit. Ut, delectus!	Bhawal Mirzapur	2500		Active	

Figure 26: Resort List

Resort Search

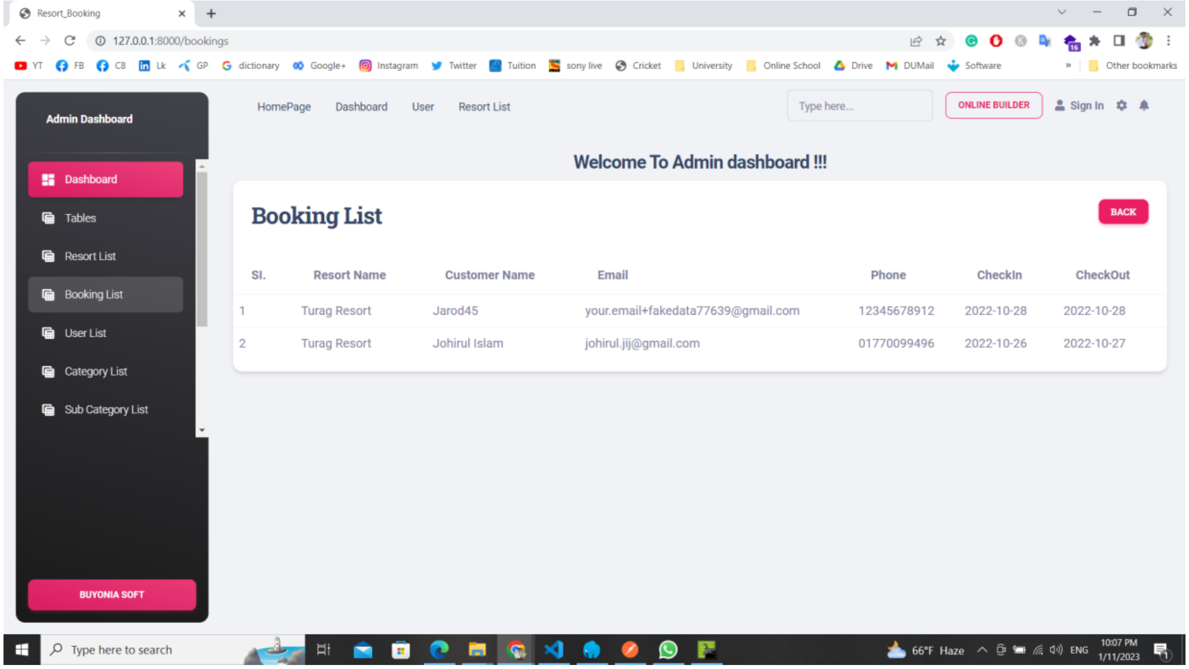


Welcome To Admin dashboard !!!

Sl.	R.Name	Description	Location	Price	Image	Status	Action
1	Safari Park	jas asjhhf jasf asjfahs asfwei asfj	Mawna	1500		Active	
2	Safari Park	jas asjhhf jasf asjfahs asfwei asfj	Mawna	1500		Active	

Figure 27: Resort Search

Booking List



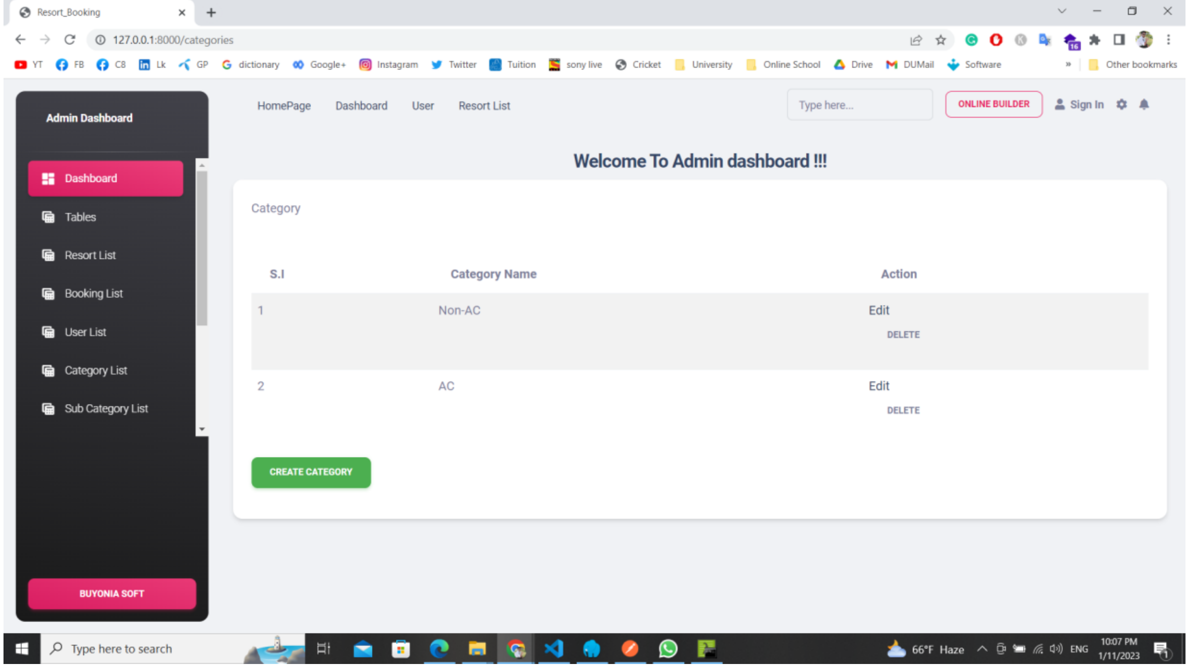
Welcome To Admin dashboard !!!

Booking List

SI.	Resort Name	Customer Name	Email	Phone	CheckIn	CheckOut
1	Turag Resort	Jarod45	your.email+fakedata77639@gmail.com	12345678912	2022-10-28	2022-10-28
2	Turag Resort	Johirul Islam	johirul.jij@gmail.com	01770099496	2022-10-26	2022-10-27

Figure 28: Booking List

Category List



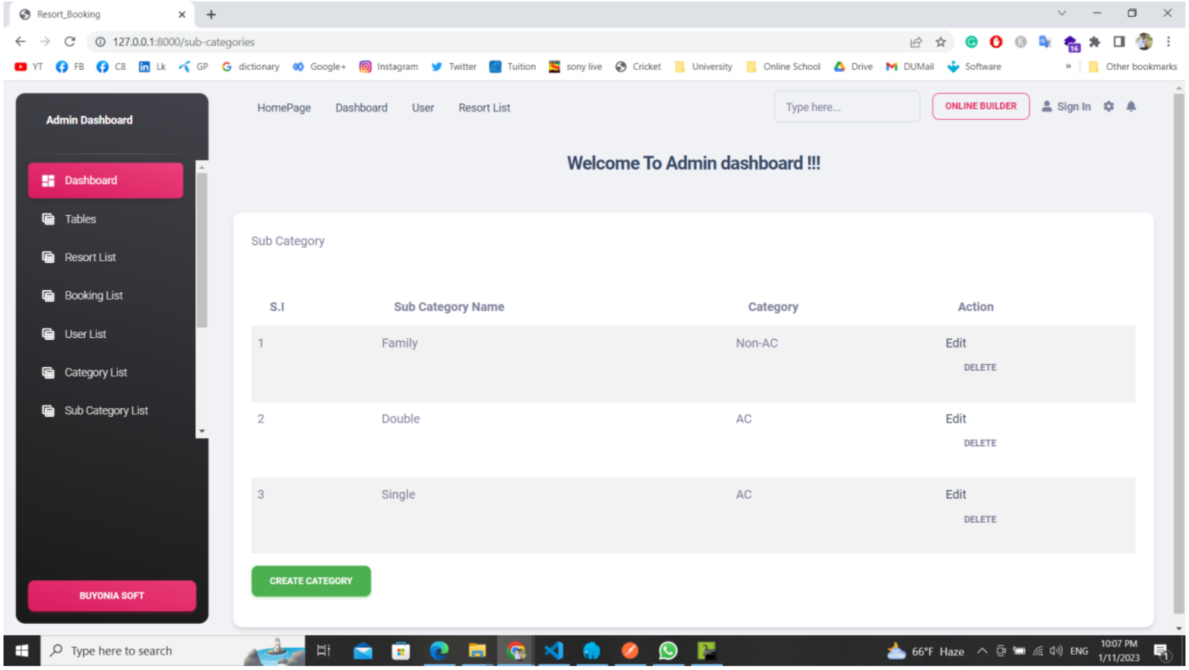
Welcome To Admin dashboard !!!

Category

S.I	Category Name	Action
1	Non-AC	Edit DELETE
2	AC	Edit DELETE

Figure 29: Category List

SubcategoryList



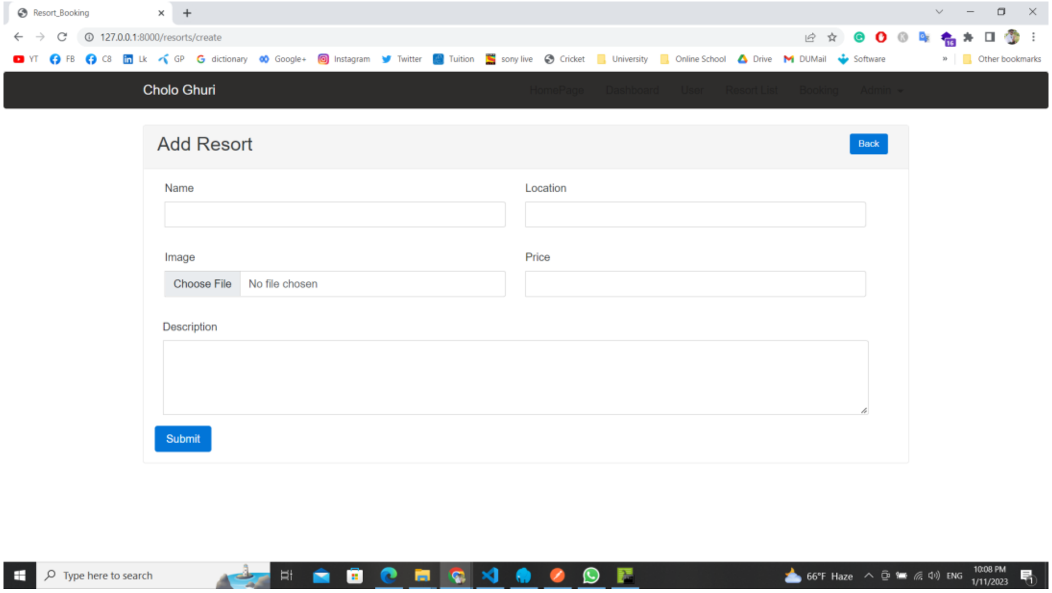
Welcome To Admin dashboard !!!

Sub Category

S.I	Sub Category Name	Category	Action
1	Family	Non-AC	Edit DELETE
2	Double	AC	Edit DELETE
3	Single	AC	Edit DELETE

Figure 30: SubcategoryList

Add Resort



Add Resort

Name

Location

Image

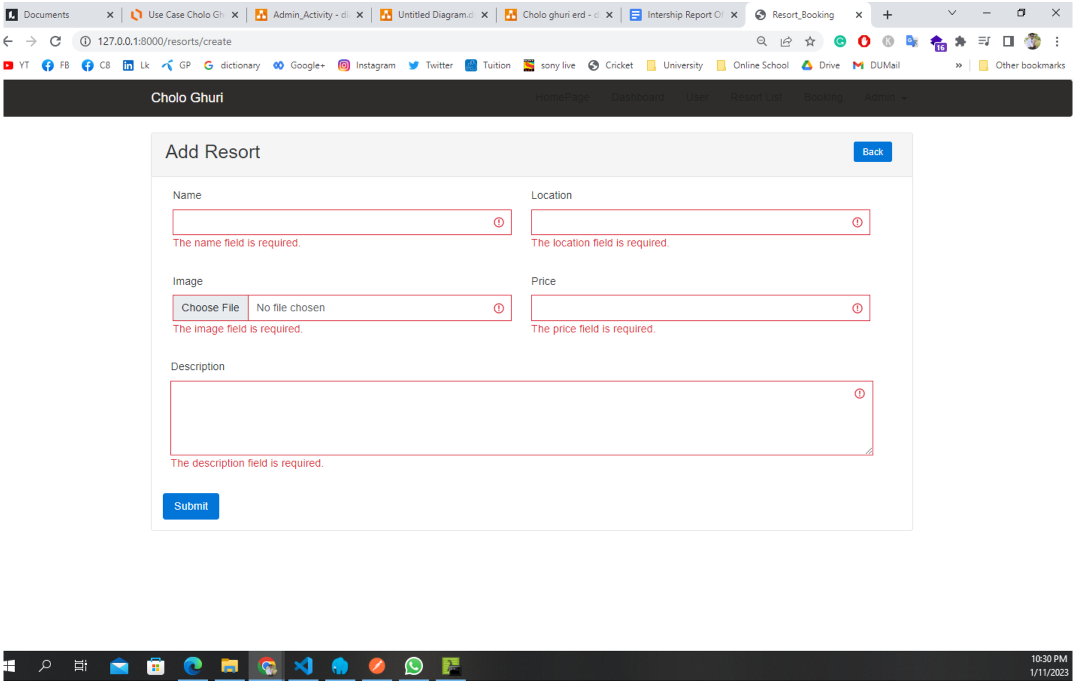
Choose File No file chosen

Price

Description

Figure 31: Add Resort

Add Resort Validation



Add Resort

Name

The name field is required.

Location

The location field is required.

Image

No file chosen

The image field is required.

Price

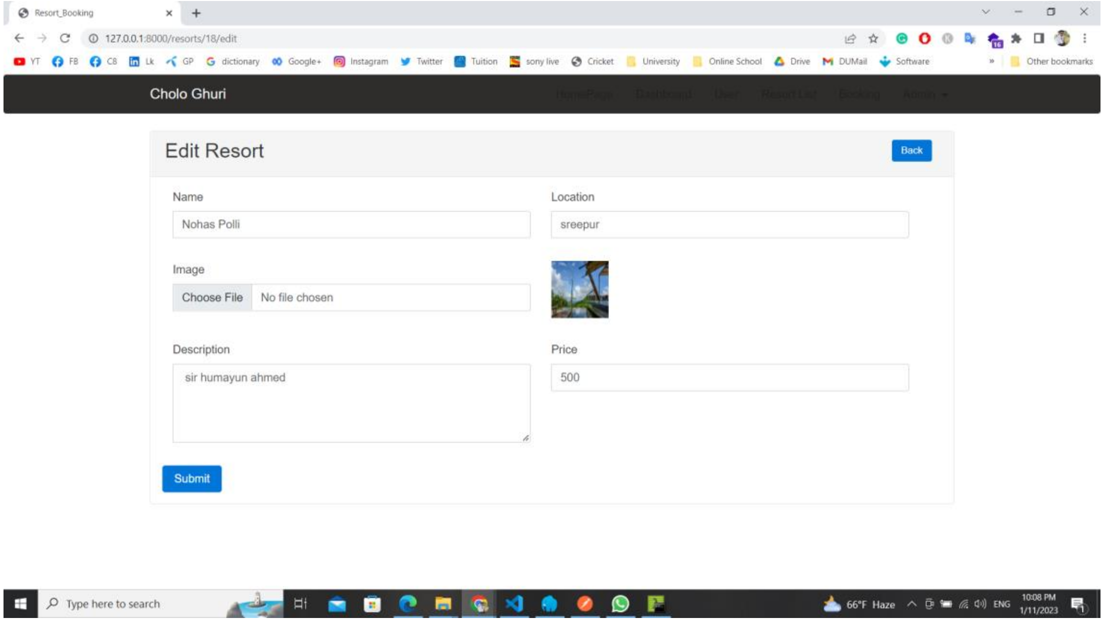
The price field is required.

Description

The description field is required.

Figure 32: Add Resort Validation

Edit Resort



Edit Resort

Name: Nohas Polli

Location: sreepur

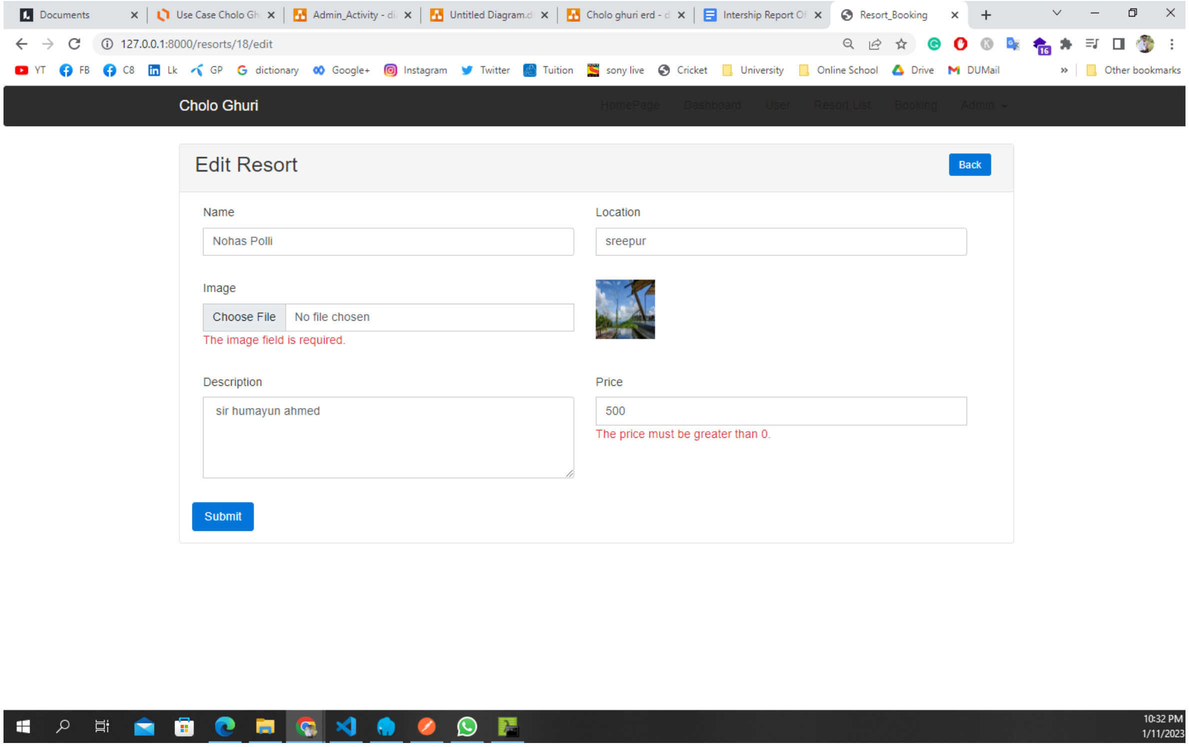
Image: No file chosen

Description: sir humayun ahmed

Price: 500

Figure 33: Edit Resort

Edit Resort Validation



Edit Resort

Back

Name: Nohas Polli

Location: sreepur

Image: Choose File No file chosen

The image field is required.

Description: sir humayun ahmed

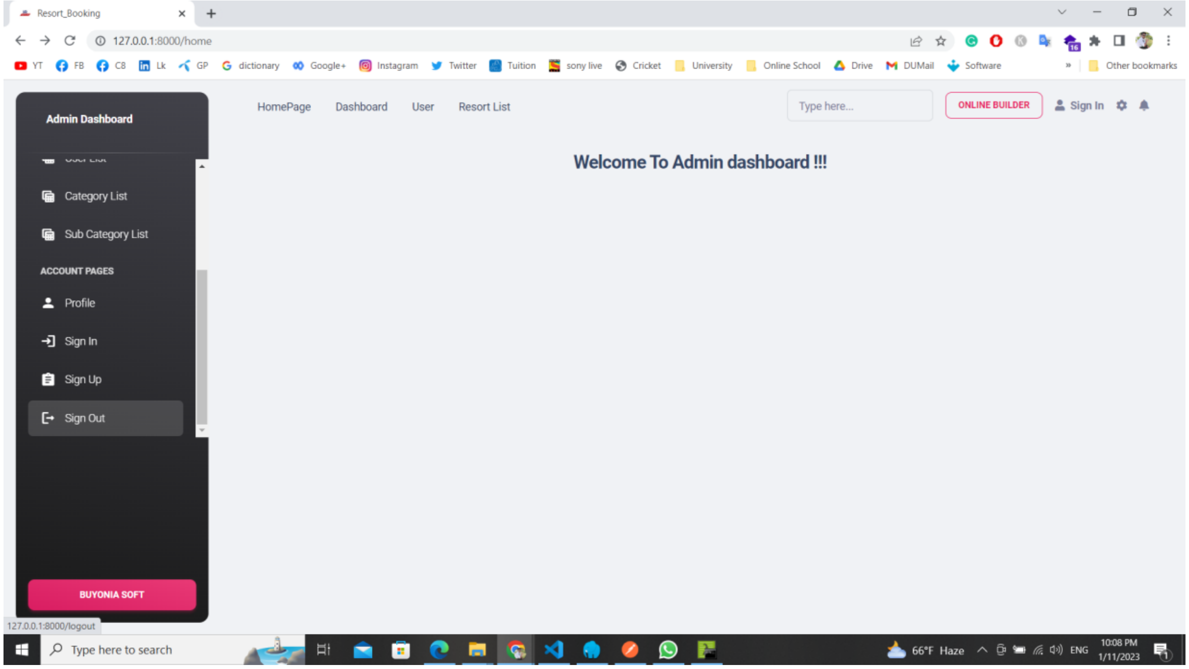
Price: 500

The price must be greater than 0.

Submit

Figure 34: Edit Resort Validation

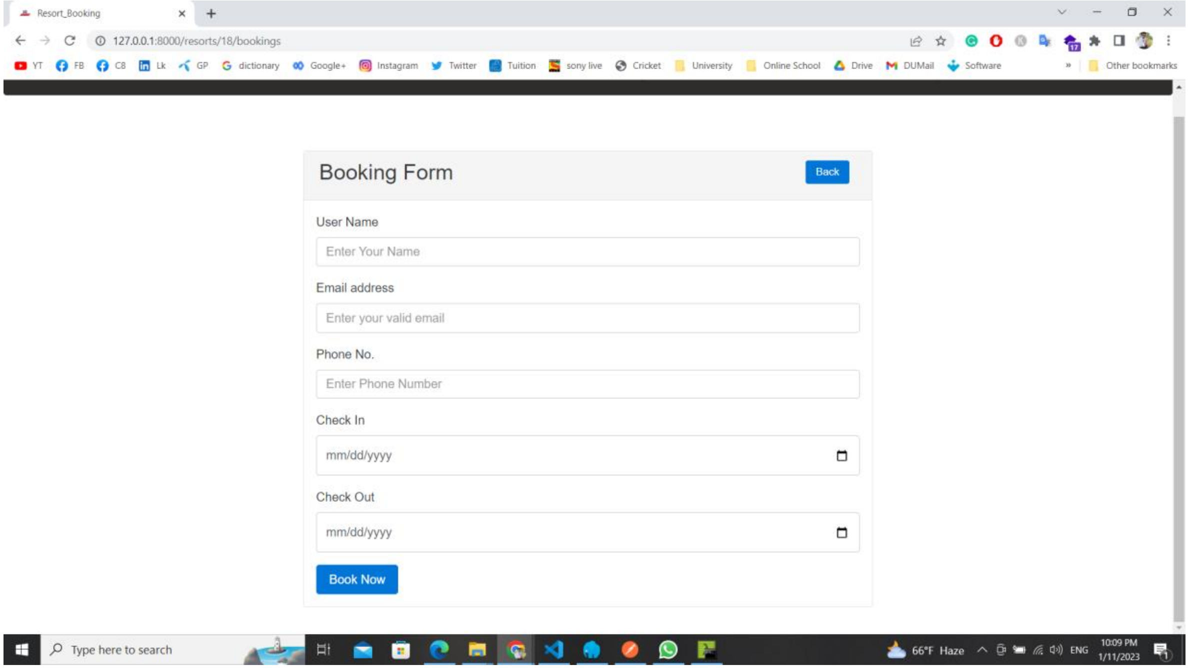
Side Bar



Welcome To Admin dashboard !!!

Figure 35: Side Bar

Booking Form



Booking Form

Back

User Name

Enter Your Name

Email address

Enter your valid email

Phone No.

Enter Phone Number

Check In

mm/dd/yyyy

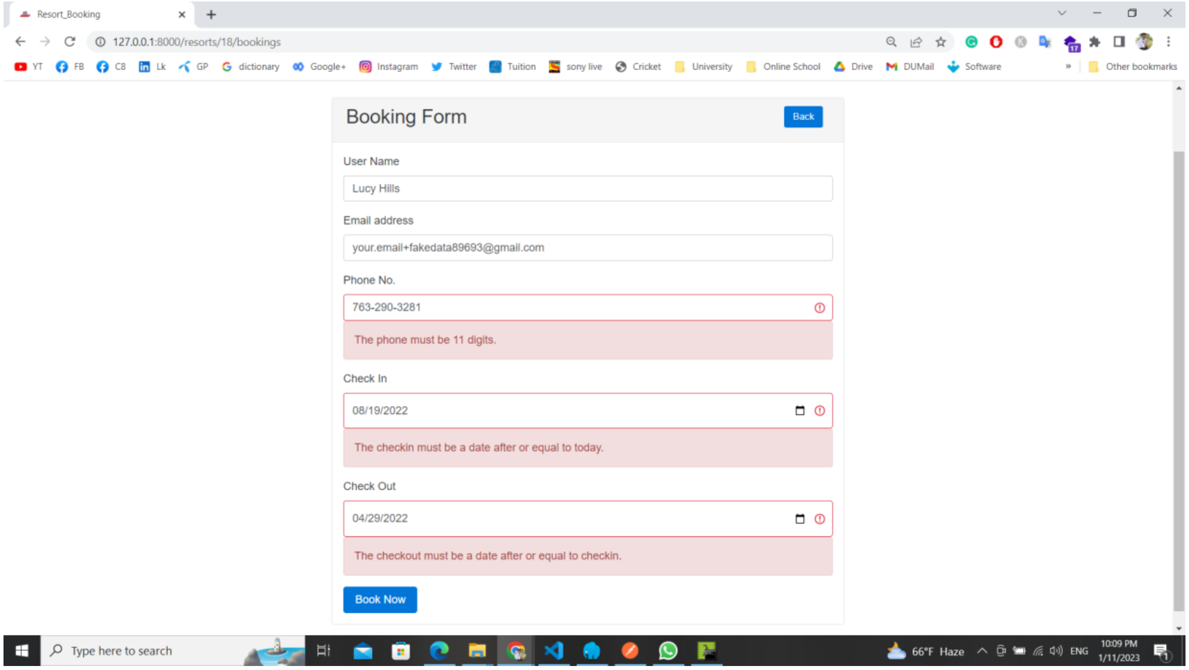
Check Out

mm/dd/yyyy

Book Now

Figure 36: Booking Form

Booking Form Validation



Booking Form

Back

User Name

Lucy Hills

Email address

your.email+fakedata89693@gmail.com

Phone No.

763-290-3281

The phone must be 11 digits.

Check In

08/19/2022

The checkin must be a date after or equal to today.

Check Out

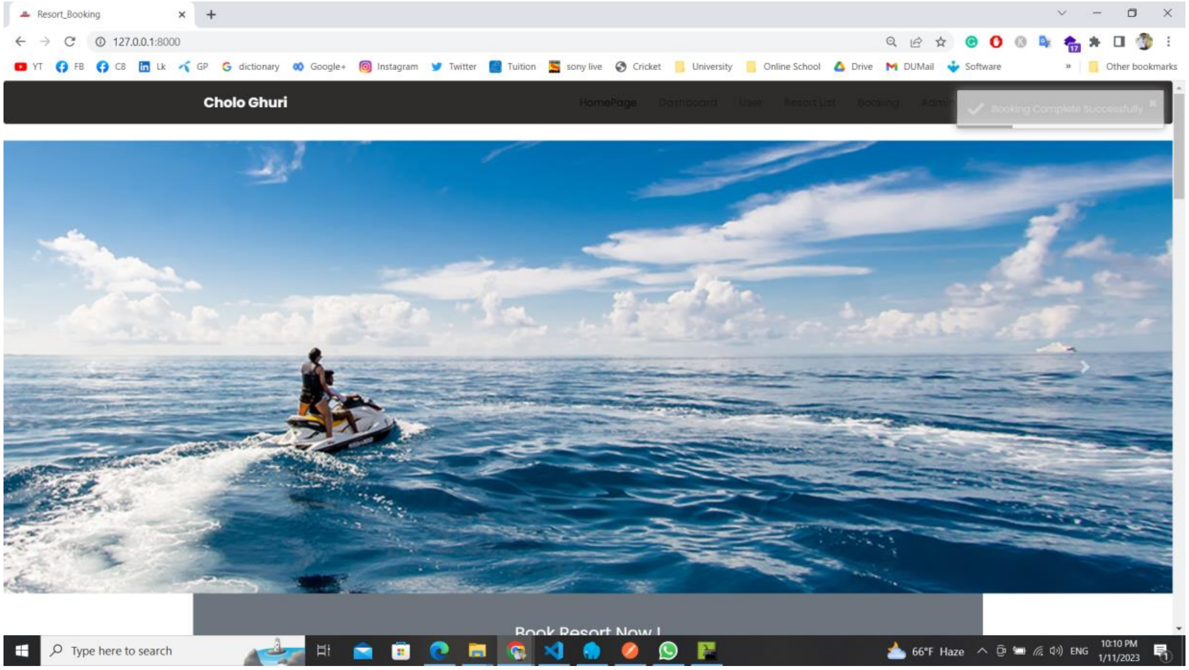
04/29/2022

The checkout must be a date after or equal to checkin.

Book Now

Figure 37: Booking Form Validation

Booking Confirm

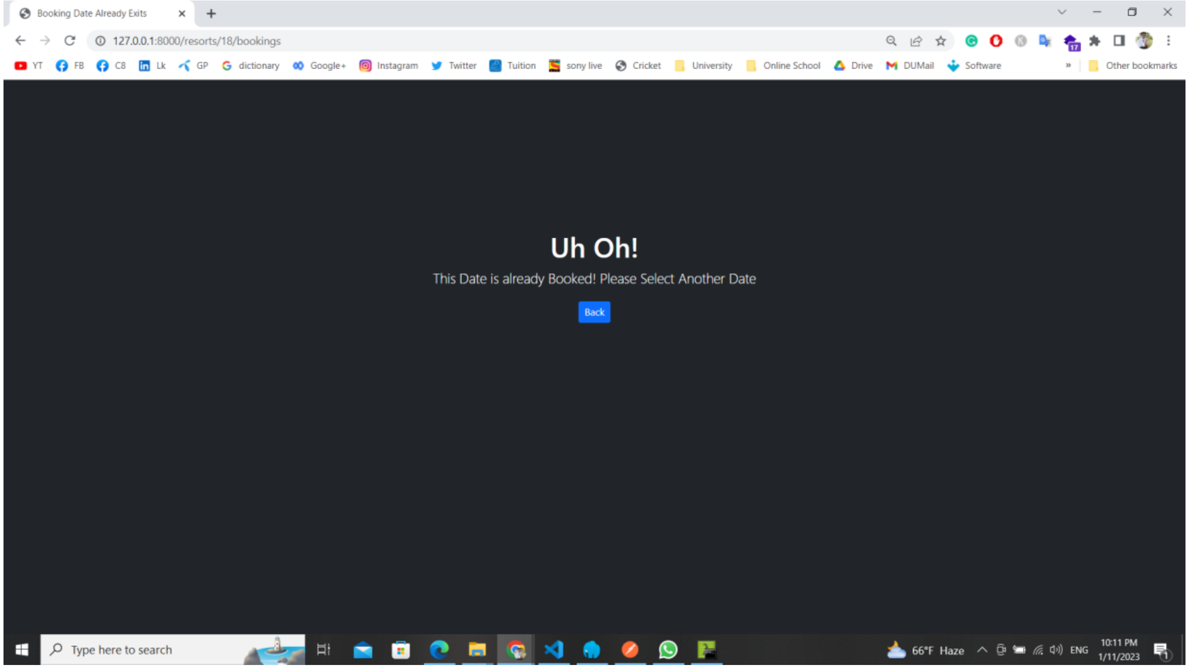


Booking Complete Successfully

Book Resort Now !

Figure 38: Booking Confirm

Booking Date

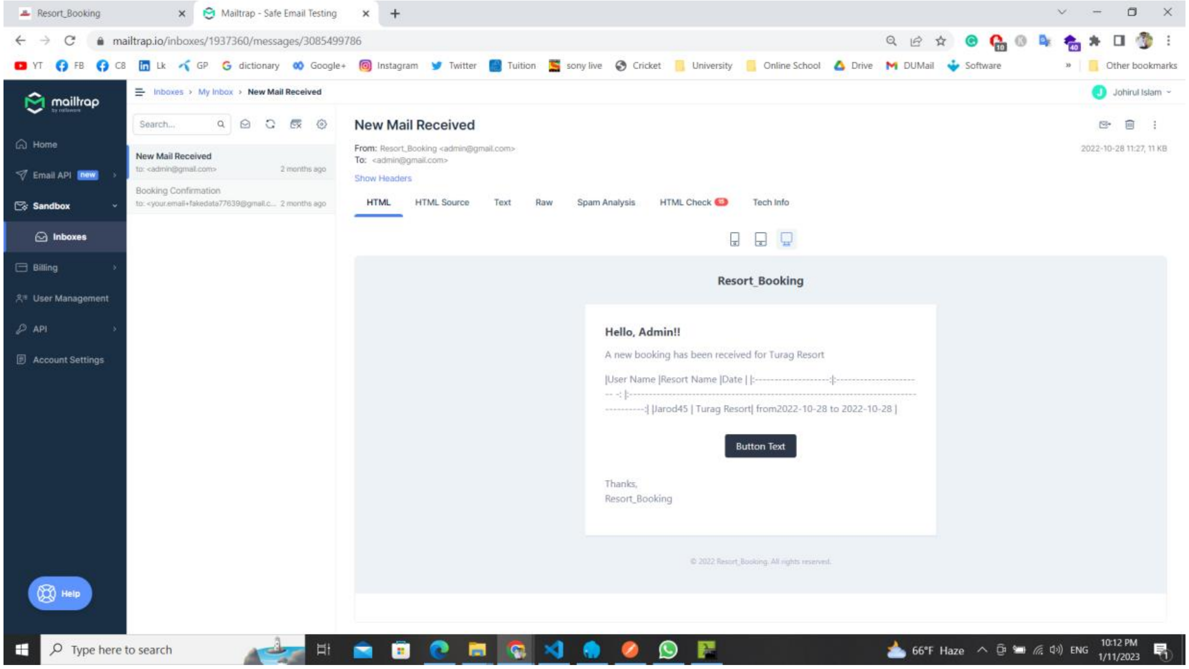

Booking Date Already Exits
127.0.0.1:8000/resorts/18/bookings
Back

Uh Oh!

This Date is already Booked! Please Select Another Date

Figure 39: Booking Date

Confirmation Mail to Admin



New Mail Received
to: <admin@gmail.com> 2 months ago

Booking Confirmation
to: <your.email+fakedata77639@gmail.c... 2 months ago

New Mail Received

From: Resort_Booking <admin@gmail.com>
To: <admin@gmail.com>

2022-10-28 11:27, 11 KB

Resort_Booking

Hello, Admin!!

A new booking has been received for Turag Resort

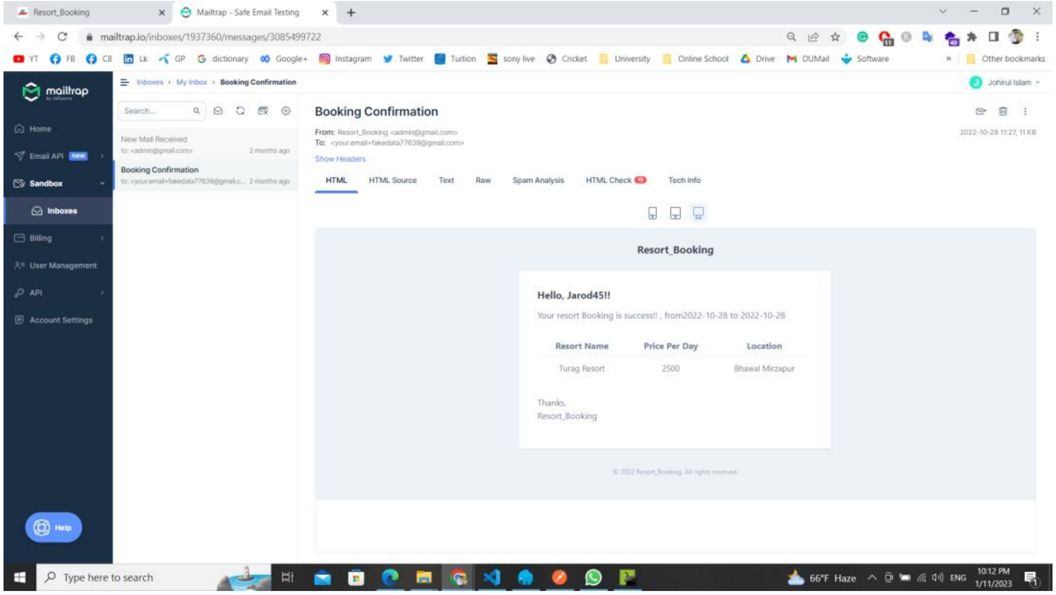
|User Name |Resort Name |Date | |:-------------------:|:--------------------- -- -:|:-- ----------:| |Jarod45 | Turag Resort| from2022-10-28 to 2022-10-28 |

Thanks,
Resort_Booking

© 2022 Resort_Booking. All rights reserved.

Figure 40: Confirmation Mail to Admin

Confirmation Mail to User



Booking Confirmation

From: Resort_Booking <admin@gmail.com>
To: <your.email+fakedata77639@gmail.com>

2022-10-28 11:27, 11 KB

Resort_Booking

Hello, Jarod45!!

Your resort Booking is success!! , from2022-10-28 to 2022-10-28

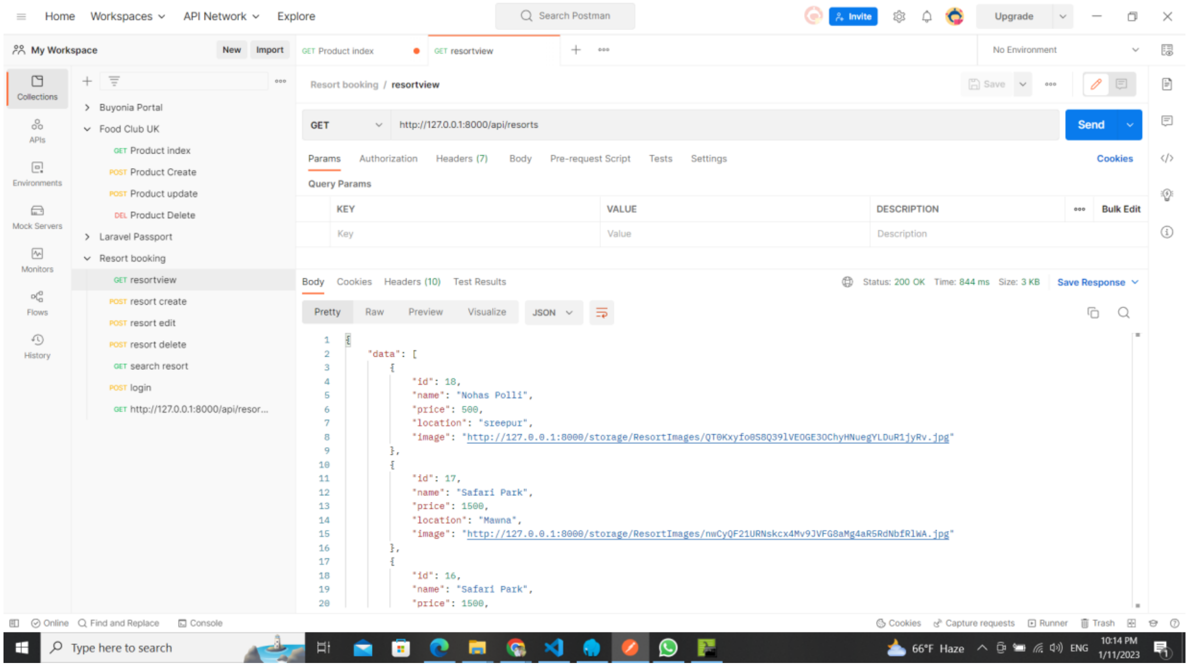
Resort Name	Price Per Day	Location
Turag Resort	2500	Bhawal Mirzapur

Thanks,
Resort_Booking

© 2022 Resort_Booking. All rights reserved.

Figure 41: Confirmation Mail to User

API Resort List



Resort booking / resortview

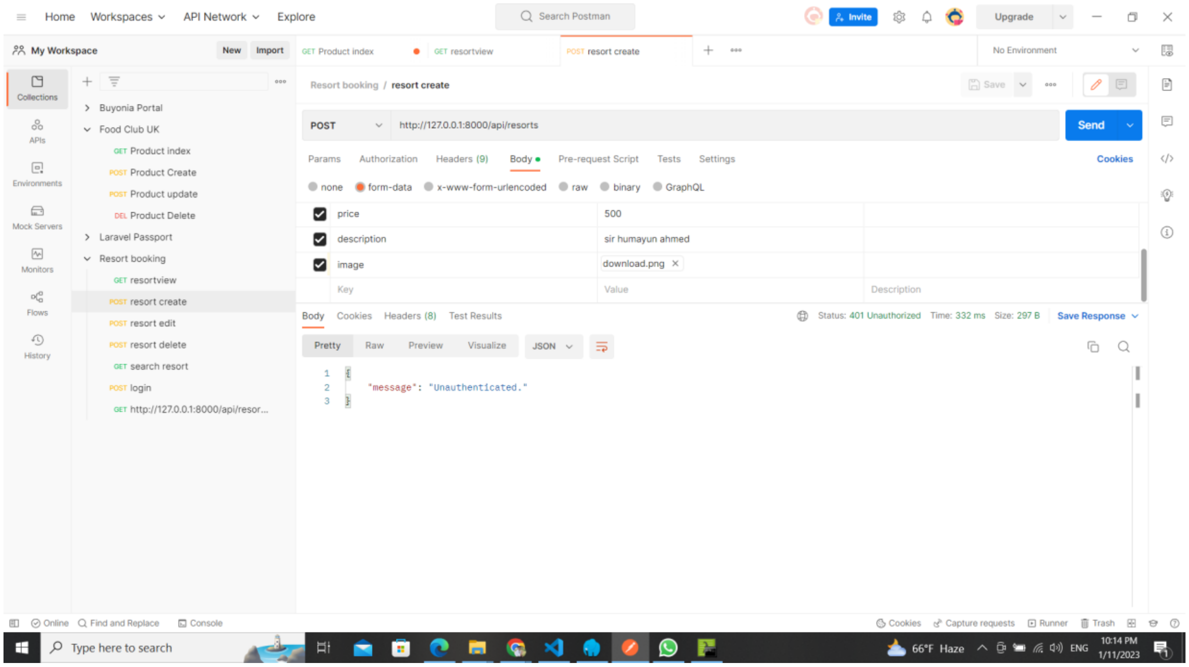
GET http://127.0.0.1:8000/api/resorts

Status: 200 OK Time: 844 ms Size: 3 KB

```
{
    "data": [
        {
            "id": 18,
            "name": "Nohas Polli",
            "price": 500,
            "location": "sreepur",
            "image": "http://127.0.0.1:8000/storage/ResortImages/QT0Kxyfo0S8Q39lVEOGE3OChyHNuegYLDuR1jyRv.jpg"
        },
        {
            "id": 17,
            "name": "Safari Park",
            "price": 1500,
            "location": "Mawna",
            "image": "http://127.0.0.1:8000/storage/ResortImages/nwCyQF21URNskcx4Mv9JVFG8aMg4aR5RdNbfRlWA.jpg"
        },
        {
            "id": 16,
            "name": "Safari Park",
            "price": 1500,
```

Figure 42: API Resort List

Resort Create



Resort booking / resort create

POST http://127.0.0.1:8000/api/resorts

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

☑	price	500	
☑	description	sir humayun ahmed	
☑	image	download.png ×	
	Key	Value	Description

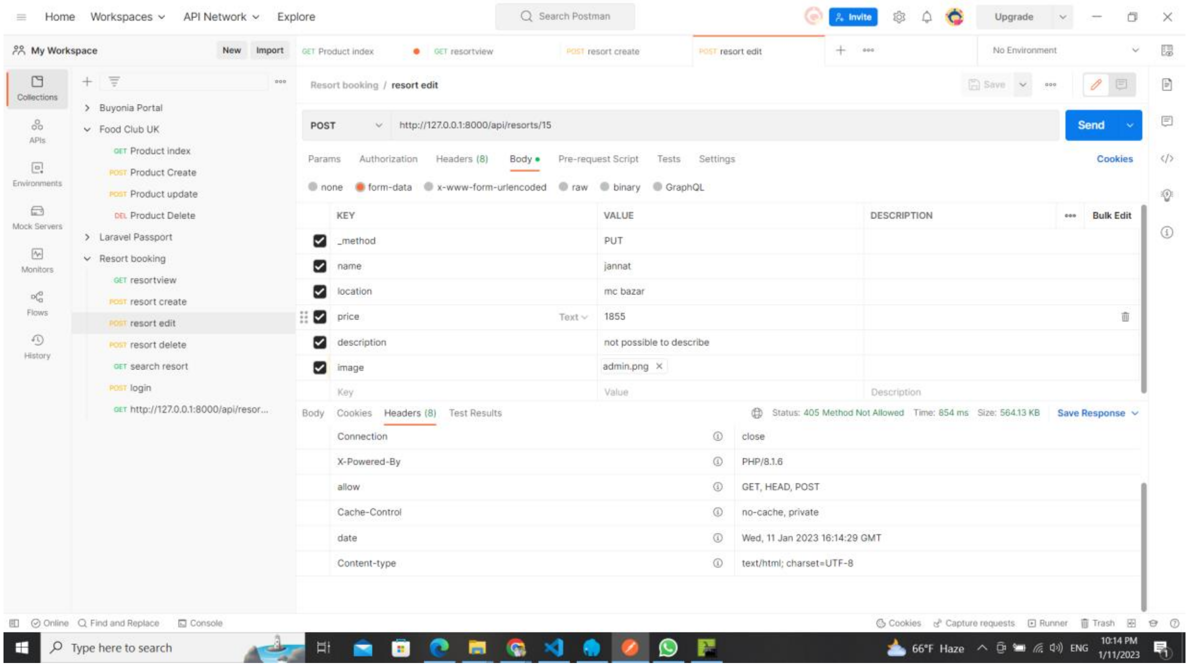
Body Cookies Headers (8) Test Results — Status: 401 Unauthorized Time: 332 ms Size: 297 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "message": "Unauthenticated."
}
```

Figure 43: Resort Create

Resort Edit



Resort booking / resort edit

POST http://127.0.0.1:8000/api/resorts/15

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

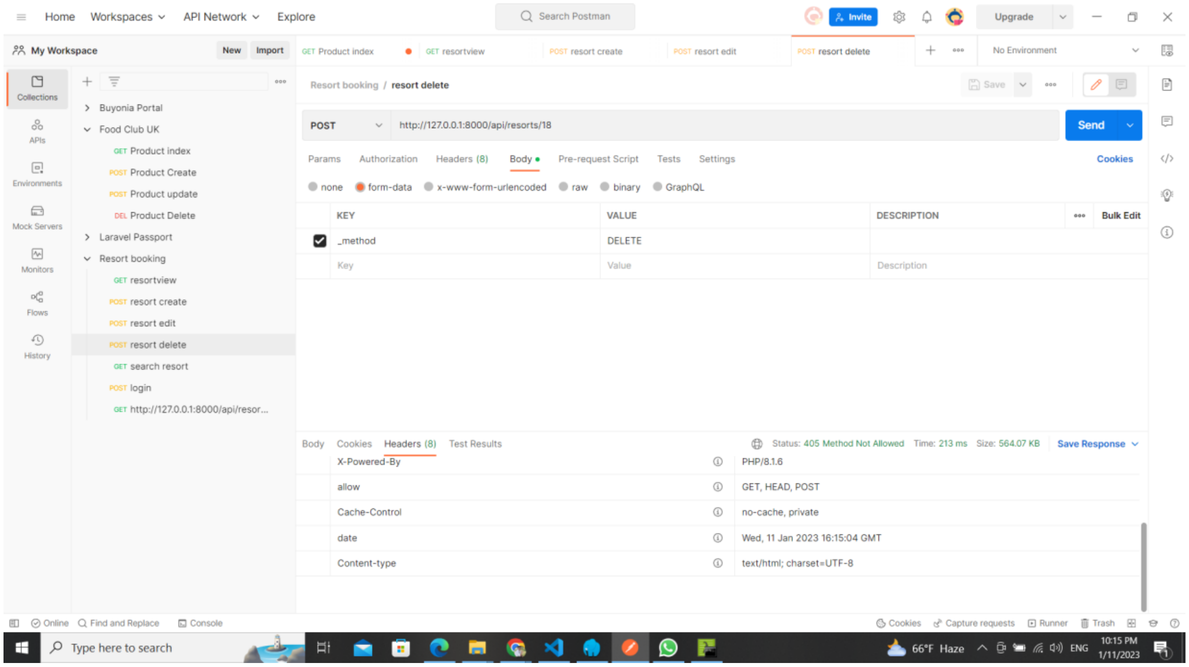
○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	_method	PUT	
☑	name	jannat	
☑	location	mc bazar	
☑	price	1855	
☑	description	not possible to describe	
☑	image	admin.png ×	
	Key	Value	Description

Body Cookies **Headers (8)** Test Results — Status: 405 Method Not Allowed Time: 854 ms Size: 564.13 KB Save Response

Connection	close
X-Powered-By	PHP/8.1.6
allow	GET, HEAD, POST
Cache-Control	no-cache, private
date	Wed, 11 Jan 2023 16:14:29 GMT
Content-type	text/html; charset=UTF-8

Resort Delete



Resort booking / resort delete

POST http://127.0.0.1:8000/api/resorts/18

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	_method	DELETE	
	Key	Value	Description

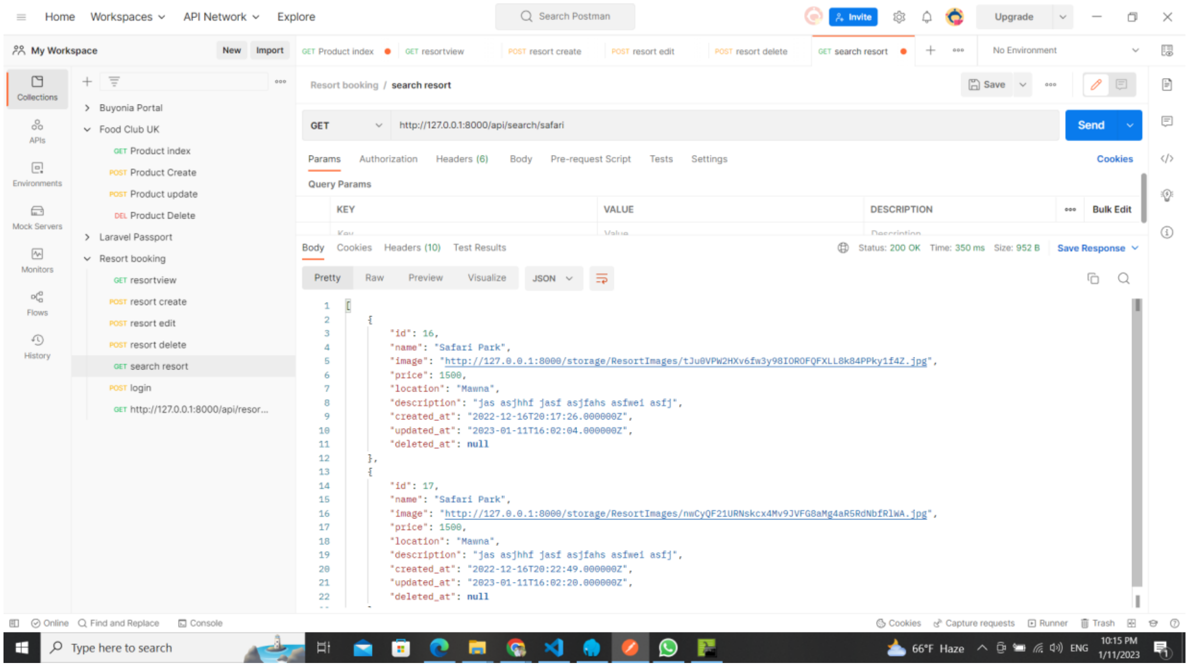
Body Cookies Headers (8) Test Results

Status: 405 Method Not Allowed Time: 213 ms Size: 564.07 KB

X-Powered-By	PHP/8.1.6
allow	GET, HEAD, POST
Cache-Control	no-cache, private
date	Wed, 11 Jan 2023 16:15:04 GMT
Content-type	text/html; charset=UTF-8

Type here to search 66°F Haze ENG 10:15 PM 1/11/2023

Resort Search



Resort booking / search resort

GET http://127.0.0.1:8000/api/search/safari

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

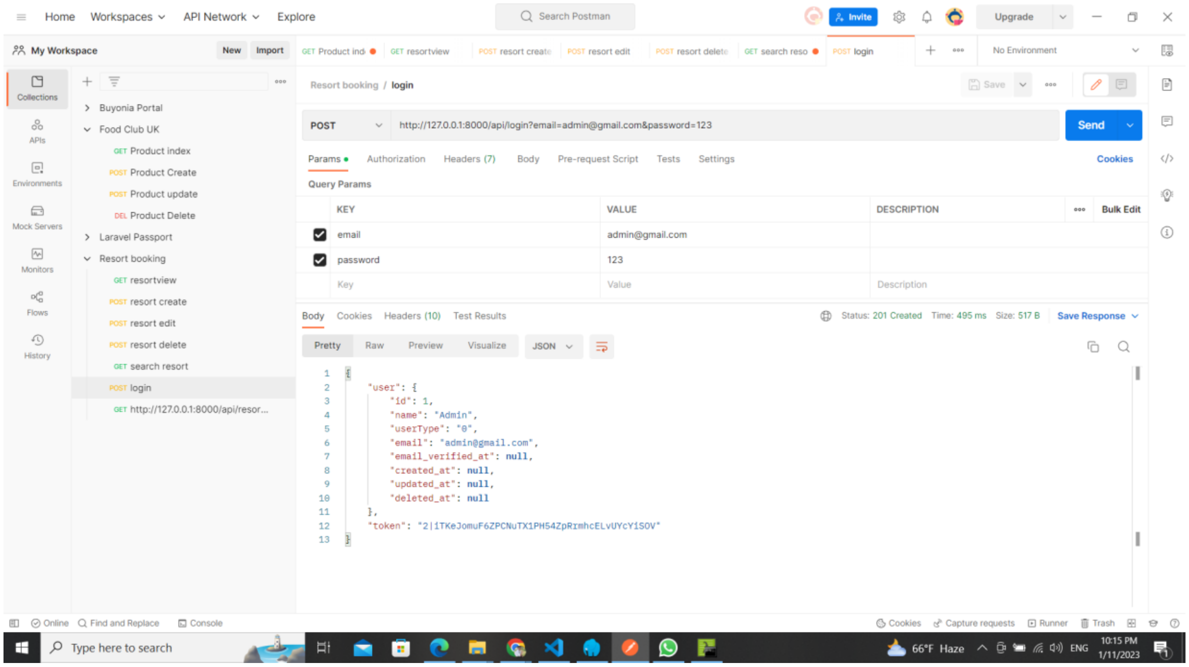
Body Cookies Headers (10) Test Results

Status: 200 OK Time: 350 ms Size: 952 B

```
[
    {
        "id": 16,
        "name": "Safari Park",
        "image": "http://127.0.0.1:8000/storage/ResortImages/tJu0VPW2HXv6fw3y98IOROFQFXLL8k84PPky1f4Z.jpg",
        "price": 1500,
        "location": "Mawna",
        "description": "jas asjhhf jasf asjfahs asfwei asfj",
        "created_at": "2022-12-16T20:17:26.000000Z",
        "updated_at": "2023-01-11T16:02:04.000000Z",
        "deleted_at": null
    },
    {
        "id": 17,
        "name": "Safari Park",
        "image": "http://127.0.0.1:8000/storage/ResortImages/nwCyQF21URNskcx4Mv9JVFG8aMg4aR5RdNbfRlWA.jpg",
        "price": 1500,
        "location": "Mawna",
        "description": "jas asjhhf jasf asjfahs asfwei asfj",
        "created_at": "2022-12-16T20:22:49.000000Z",
        "updated_at": "2023-01-11T16:02:20.000000Z",
        "deleted_at": null
```

Type here to search 66°F Haze ENG 10:15 PM 1/11/2023

Login



Resort booking / login

POST http://127.0.0.1:8000/api/login?email=admin@gmail.com&password=123

Query Params

	KEY	VALUE	DESCRIPTION
☑	email	admin@gmail.com	
☑	password	123	
	Key	Value	Description

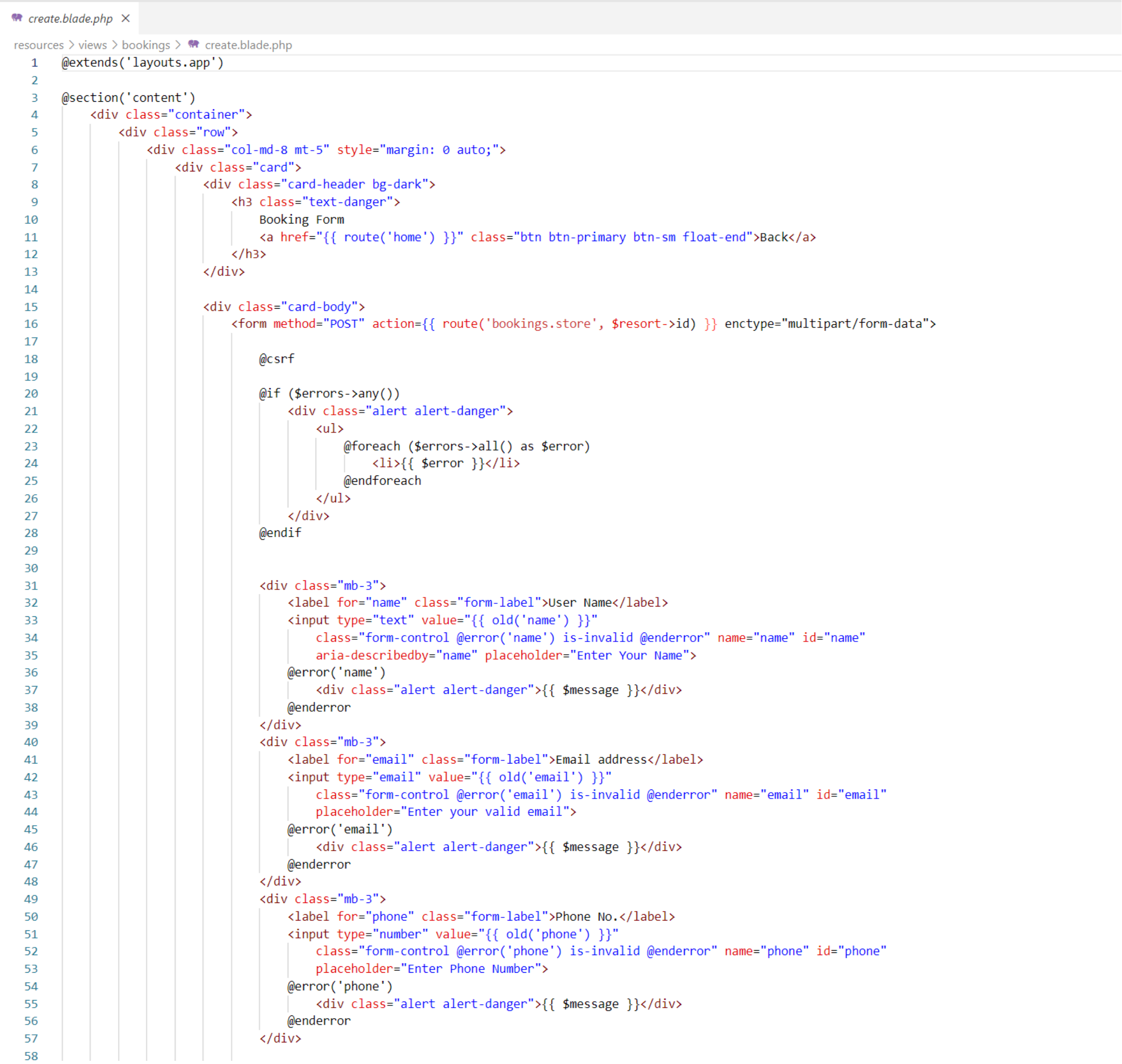
Status: 201 Created Time: 495 ms Size: 517 B

```
{
    "user": {
        "id": 1,
        "name": "Admin",
        "userType": "0",
        "email": "admin@gmail.com",
        "email_verified_at": null,
        "created_at": null,
        "updated_at": null,
        "deleted_at": null
    },
    "token": "2|1TKeJomuF6ZPCNuTX1PH54ZpRrmhcELvUYcY1SOV"
}
```

CHAPTER 10 – DEPLOYMENT

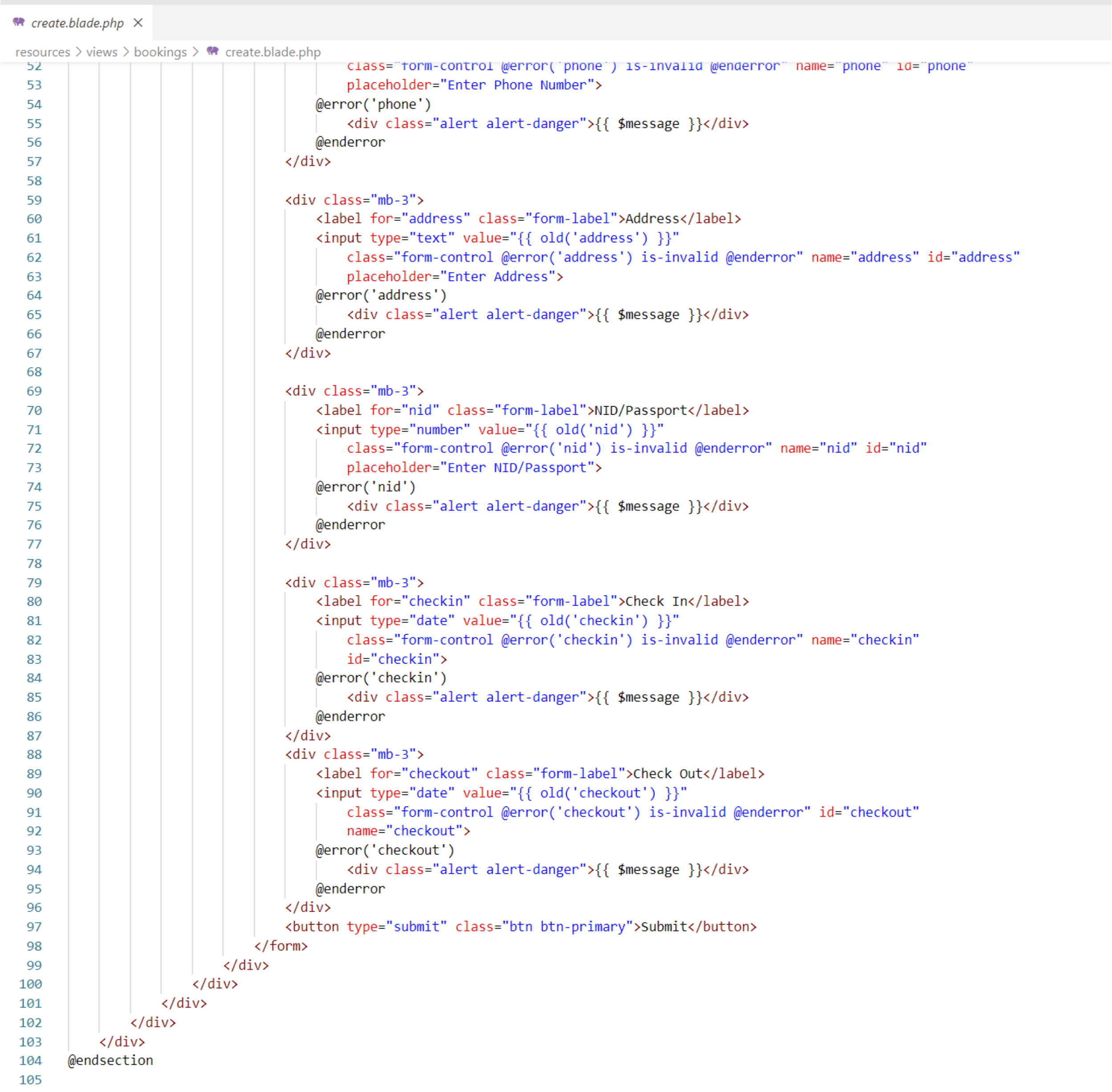
10.1 Core Module Coding Sample

For this system, front-end, a number of technologies were implemented. These include HTML5, CSS3, Bootstrap (5.0.0), Modern JavaScript (3.6.1) and jQuery (3.6.1). The backend architecture was built using MySQL, PHP 7.8, and Laravel 8. Below are some coding examples:



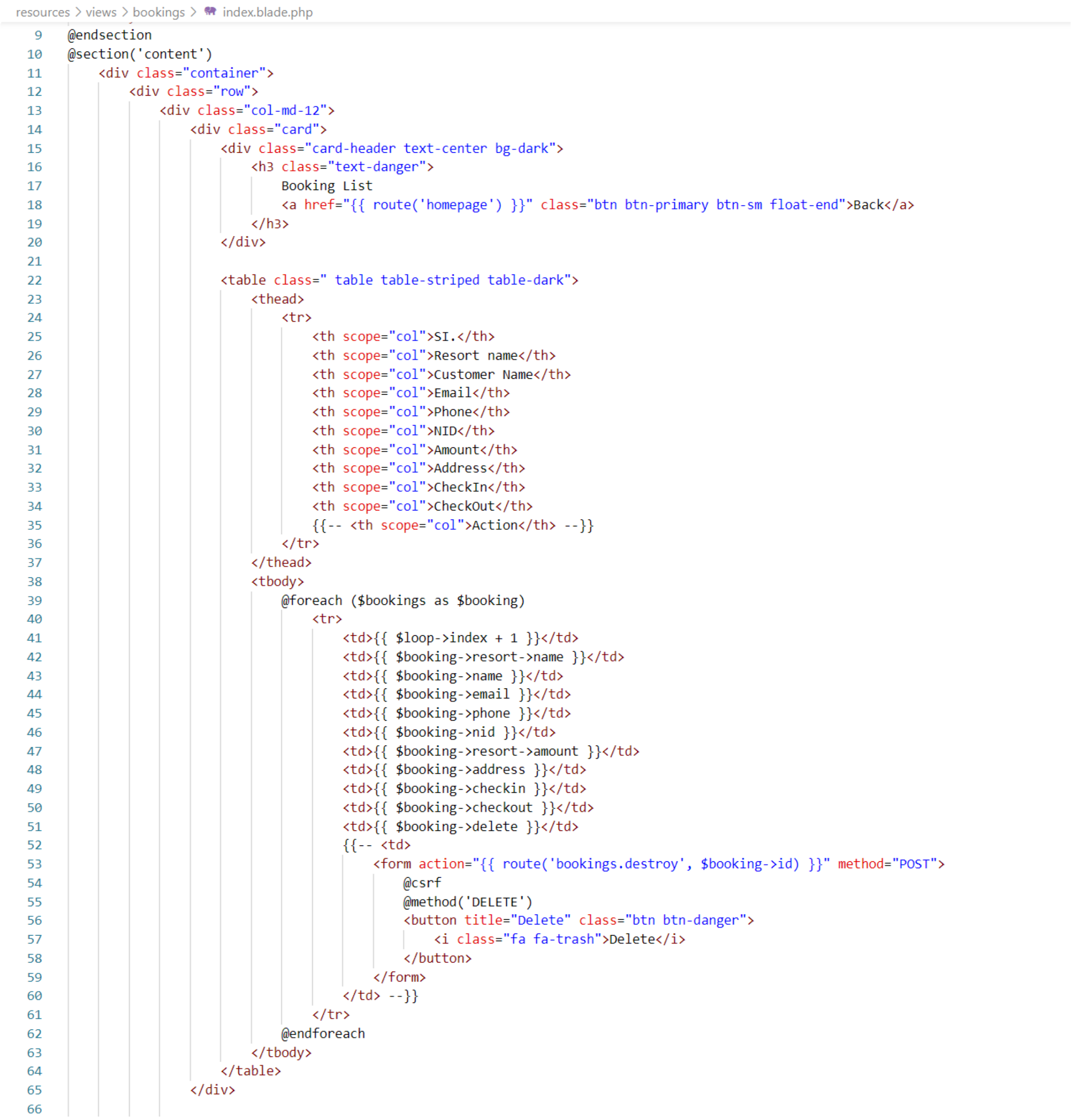
```
@extends('layouts.app')

@section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-8 mt-5" style="margin: 0 auto;">
                <div class="card">
                    <div class="card-header bg-dark">
                        <h3 class="text-danger">
                            Booking Form
                            <a href="{{ route('home') }}" class="btn btn-primary btn-sm float-end">Back</a>
                        </h3>
                    </div>

                    <div class="card-body">
                        <form method="POST" action={{ route('bookings.store', $resort->id) }} enctype="multipart/form-data">

                            @csrf

                            @if ($errors->any())
                                <div class="alert alert-danger">
                                    <ul>
                                        @foreach ($errors->all() as $error)
                                            <li>{{ $error }}</li>
                                        @endforeach
                                    </ul>
                                </div>
                            @endif


                            <div class="mb-3">
                                <label for="name" class="form-label">User Name</label>
                                <input type="text" value="{{ old('name') }}"
                                    class="form-control @error('name') is-invalid @enderror" name="name" id="name"
                                    aria-describedby="name" placeholder="Enter Your Name">
                                @error('name')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>
                            <div class="mb-3">
                                <label for="email" class="form-label">Email address</label>
                                <input type="email" value="{{ old('email') }}"
                                    class="form-control @error('email') is-invalid @enderror" name="email" id="email"
                                    placeholder="Enter your valid email">
                                @error('email')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>
                            <div class="mb-3">
                                <label for="phone" class="form-label">Phone No.</label>
                                <input type="number" value="{{ old('phone') }}"
                                    class="form-control @error('phone') is-invalid @enderror" name="phone" id="phone"
                                    placeholder="Enter Phone Number">
                                @error('phone')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>

```

Figure 44: Sample code of Create Booking



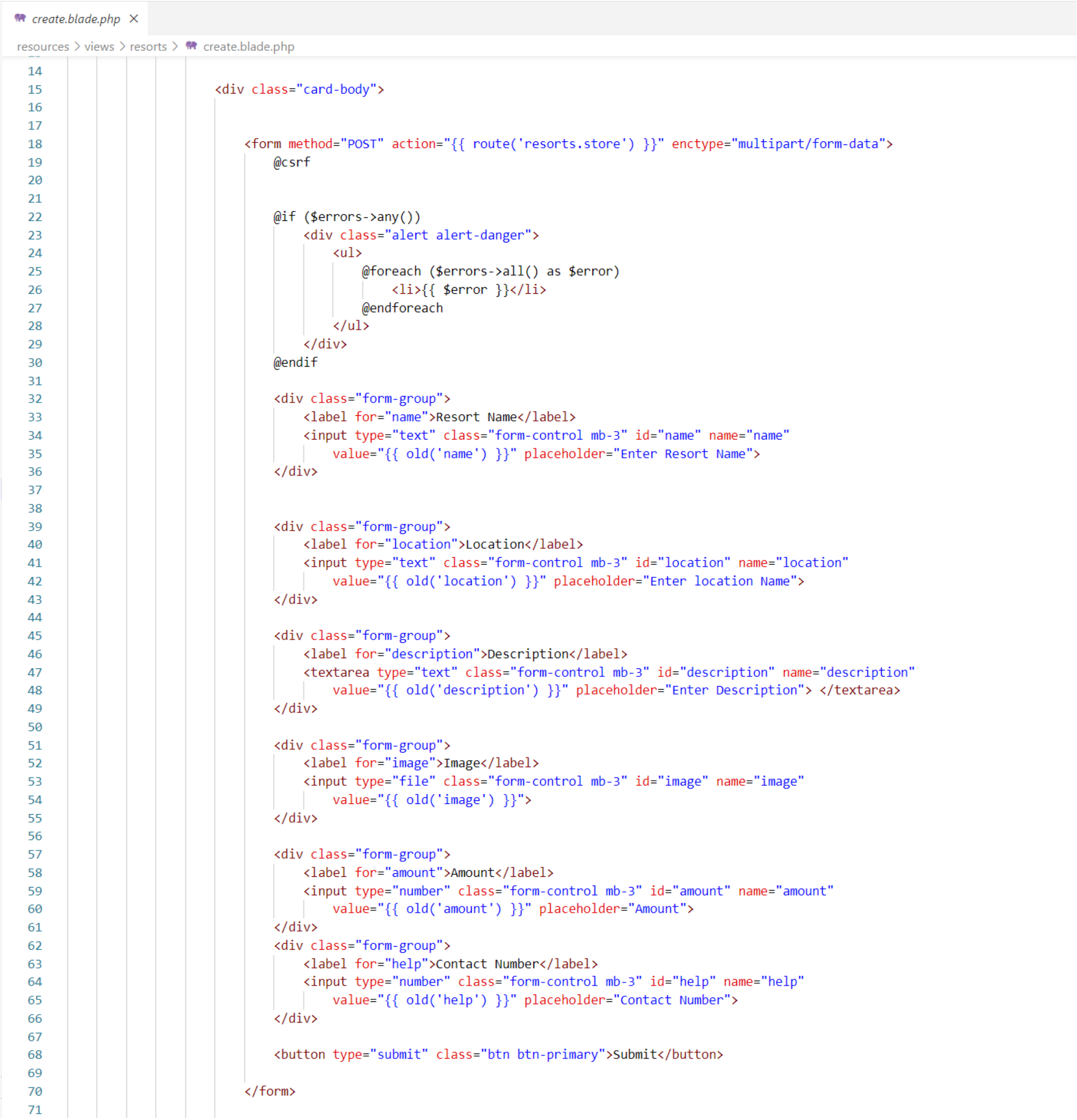
```
                                    class="form-control @error('phone') is-invalid @enderror" name="phone" id="phone"
                                    placeholder="Enter Phone Number">
                                @error('phone')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>

                            <div class="mb-3">
                                <label for="address" class="form-label">Address</label>
                                <input type="text" value="{{ old('address') }}"
                                    class="form-control @error('address') is-invalid @enderror" name="address" id="address"
                                    placeholder="Enter Address">
                                @error('address')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>

                            <div class="mb-3">
                                <label for="nid" class="form-label">NID/Passport</label>
                                <input type="number" value="{{ old('nid') }}"
                                    class="form-control @error('nid') is-invalid @enderror" name="nid" id="nid"
                                    placeholder="Enter NID/Passport">
                                @error('nid')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>

                            <div class="mb-3">
                                <label for="checkin" class="form-label">Check In</label>
                                <input type="date" value="{{ old('checkin') }}"
                                    class="form-control @error('checkin') is-invalid @enderror" name="checkin"
                                    id="checkin">
                                @error('checkin')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>
                            <div class="mb-3">
                                <label for="checkout" class="form-label">Check Out</label>
                                <input type="date" value="{{ old('checkout') }}"
                                    class="form-control @error('checkout') is-invalid @enderror" id="checkout"
                                    name="checkout">
                                @error('checkout')
                                    <div class="alert alert-danger">{{ $message }}</div>
                                @enderror
                            </div>
                            <button type="submit" class="btn btn-primary">Submit</button>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection
```

Figure 45: Create Booking



```
@endsection
@section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="card">
                    <div class="card-header text-center bg-dark">
                        <h3 class="text-danger">
                            Booking List
                            <a href="{{ route('homepage') }}" class="btn btn-primary btn-sm float-end">Back</a>
                        </h3>
                    </div>

                    <table class=" table table-striped table-dark">
                        <thead>
                            <tr>
                                <th scope="col">SI.</th>
                                <th scope="col">Resort name</th>
                                <th scope="col">Customer Name</th>
                                <th scope="col">Email</th>
                                <th scope="col">Phone</th>
                                <th scope="col">NID</th>
                                <th scope="col">Amount</th>
                                <th scope="col">Address</th>
                                <th scope="col">CheckIn</th>
                                <th scope="col">CheckOut</th>
                                {{-- <th scope="col">Action</th> --}}
                            </tr>
                        </thead>
                        <tbody>
                            @foreach ($bookings as $booking)
                                <tr>
                                    <td>{{ $loop->index + 1 }}</td>
                                    <td>{{ $booking->resort->name }}</td>
                                    <td>{{ $booking->name }}</td>
                                    <td>{{ $booking->email }}</td>
                                    <td>{{ $booking->phone }}</td>
                                    <td>{{ $booking->nid }}</td>
                                    <td>{{ $booking->resort->amount }}</td>
                                    <td>{{ $booking->address }}</td>
                                    <td>{{ $booking->checkin }}</td>
                                    <td>{{ $booking->checkout }}</td>
                                    <td>{{ $booking->delete }}</td>
                                    {{-- <td>
                                        <form action="{{ route('bookings.destroy', $booking->id) }}" method="POST">
                                            @csrf
                                            @method('DELETE')
                                            <button title="Delete" class="btn btn-danger">
                                                <i class="fa fa-trash">Delete</i>
                                            </button>
                                        </form>
                                    </td> --}}
                                </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>

```

Figure 46Customer Booking



```

                    <div class="card-body">


                        <form method="POST" action="{{ route('resorts.store') }}" enctype="multipart/form-data">
                            @csrf


                            @if ($errors->any())
                                <div class="alert alert-danger">
                                    <ul>
                                        @foreach ($errors->all() as $error)
                                            <li>{{ $error }}</li>
                                        @endforeach
                                    </ul>
                                </div>
                            @endif

                            <div class="form-group">
                                <label for="name">Resort Name</label>
                                <input type="text" class="form-control mb-3" id="name" name="name"
                                    value="{{ old('name') }}" placeholder="Enter Resort Name">
                            </div>


                            <div class="form-group">
                                <label for="location">Location</label>
                                <input type="text" class="form-control mb-3" id="location" name="location"
                                    value="{{ old('location') }}" placeholder="Enter location Name">
                            </div>

                            <div class="form-group">
                                <label for="description">Description</label>
                                <textarea type="text" class="form-control mb-3" id="description" name="description"
                                    value="{{ old('description') }}" placeholder="Enter Description"> </textarea>
                            </div>

                            <div class="form-group">
                                <label for="image">Image</label>
                                <input type="file" class="form-control mb-3" id="image" name="image"
                                    value="{{ old('image') }}">
                            </div>

                            <div class="form-group">
                                <label for="amount">Amount</label>
                                <input type="number" class="form-control mb-3" id="amount" name="amount"
                                    value="{{ old('amount') }}" placeholder="Amount">
                            </div>
                            <div class="form-group">
                                <label for="help">Contact Number</label>
                                <input type="number" class="form-control mb-3" id="help" name="help"
                                    value="{{ old('help') }}" placeholder="Contact Number">
                            </div>

                            <button type="submit" class="btn btn-primary">Submit</button>

                        </form>

```

Figure 47: Create new resort

10.2 Possible problem breakdown

Resort Booking Systems in Bangladesh's resort industry play an essential role, enabling customers to easily make reservations. Unfortunately, like any software application built using Laravel may face unique obstacles; therefore, this breakdown will outline any issues developers might experience while creating or maintaining one for Bangladesh market use. Below We have listed down some of these possible breakdown:

- Localization and Language Support:
- Integrating secure and reputable payment gateways
- Resort Availability and Pricing Management:
- Integration With Third-Party APIs:
- User Experience and Responsive Design
- Performance and Scalability:
- Security and Data Protection:

Localization and Language Support:

A major challenge facing Bangladesh lies in offering localization and language support for users across its vast landmass, especially due to a diverse population. Implementing translation processes while guaranteeing localized date, time, and currency formats may prove to be complex processes.

Integrating secure and reputable payment gateways that support popular local payment methods can be challenging in Bangladesh, which boasts its own set of preferred options such as bKash, Rocket and Nagad. Developers must carefully research these payment gateways so as to integrate them seamlessly while meeting security standards while managing transactions effectively and seamlessly.

Resort Availability and Pricing Management:

Resort availability and pricing management is a critical feature of the booking system, involving real-time updates of the availability of rooms, amenities, and packages across time zones. Coordinating updates while simultaneously assuring accurate price calculations across various seasons, promotions, and discounts can be complex undertakings that require careful execution to achieve success.

Integration With Third-Party APIs:

In order to enhance user experiences, Resort Booking Systems often need to integrate with various third-party APIs from various third parties, including weather forecast APIs that give users accurate weather information that helps inform booking decisions. But these integrations present their own set of challenges: finding reliable APIs at appropriate rate limits while remaining compatible as time progresses - these challenges must all be taken into consideration before final decisions on any integration can be made.

User Experience and Responsive Design:

A booking system must offer users a user-friendly and visually attractive user interface for it to function seamlessly across different devices such as desktop computers, smartphones and tablets. Developers should focus on responsive design to ensure the system functions across these various screen sizes seamlessly without interruption - from desktop computers through smartphones to tablets and more requiring responsive designs aimed at desktops, smartphones and tablets requiring constant testing for optimal usability requiring rigorous optimization efforts requiring ongoing testing & optimization activities to achieve balance among aesthetics usability performance needs balancing aesthetics usability performance requires regular optimization activities requiring extensive testing & optimization processes and requires regular updating & optimization efforts as it requires rigorous optimization efforts as it requires thorough testing/optimization processes as it requires extensive testing & optimization cycles before beginning with any new system or product roll out process, taking into consideration any modifications which need be applied during its design stage or any given device lifecycle phase or device life cycle cycle starts lifecycle begins life cycle begins its life cycle cycle begins again and continues on this basis until further optimization techniques take hold and are required.

Performance and Scalability:

With more users and bookings coming through the system, scalability becomes ever more of a top concern to ensure optimal performance and keep up with increased traffic loads. An efficient database design, caching strategies, and optimizing queries all play key roles in reaching this goal; while load testing to identify any performance bottlenecks may help guarantee peak loads such as those found during holiday seasons or special events are handled seamlessly.

Security and Data Protection:

A Resort Booking System places great emphasis on data protection as it deals with personal and payment details as well as reservation data. Implementation of robust authentication mechanisms, encryption of sensitive data and adherence to industry standards such as PCI-DSS are vital elements in providing maximum protection of this sensitive data. Regular audits as well as proactive vulnerability scanning should take place to safeguard potential vulnerabilities and minimize their effects.

10.3 Prioritization while developing

Prioritizing requirements is crucial during the design and construction of the system. According to the MoSCoW standards, I have already prioritized my list of requirements. You'll see the list of requirements that will be used to develop the Cholo Ghuri Project.

Table 9: MoSCoW prioritization

SL	Task Name
1	Users should be able to create accounts and log in securely.
2	Resort staff should be able to add, update, and remove resort and room information.
3	Room availability status should be maintained and updated in real-time.
4	Customers should be able to search for available resorts and rooms based on their preferences.
5	Customers should be able to make bookings and proceed with secure online payments.
6	The system should generate booking confirmations and receipts.
7	The system should track the availability of rooms in real time and prevent double bookings.
8	Resort staff should be able to update room availability status based on reservations.
9	User authentication should be implemented to ensure secure access to the system.
10	The system should provide reports and analytics on booking trends, occupancy rates, and revenue.

CHAPTER 11 – TESTING

11.1 Test Plan Acceptance

This document is comprehensive and describes all the necessary information to complete software testing. It includes the strategy for the tests, the objectives, the schedule, the estimations, the deliverables as well as the resources. The user must acknowledge the plan during the analysis phase of the project. This certifies that the testing materials and criteria for approval are appropriate. Two main types of testing exist.

- ➢ Functional Testing
- ➢ Non-functional testing

11.1.1 Functional Testing

Functional testing allows QAs to determine whether a software product is conforming with the predefined requirements. Black-box approaches are used, where the tester is unaware of system logic. Functional testing is concerned with whether or not the system works as expected. Functional tests are divided into three categories.

- ➢ **Unit Testing**
- ✓ Validation form input fields
- ✓ List of mobile applications with their status
- ✓ List users according to an app
- ➢ **Module Testing**

- ✓ Submission of login forms without providing information

- ✓ You can't log in using invalid information.

- ✓ Filling out the form correctly.

- ➢ **Integrity Testing**

- ✓ Login to the system with your credentials

- ✓ Details of the administrator can be edited and added.

- ➢ **Acceptance testing**
  - ✓ Check if user satisfaction level

11.1.2 Non-functional testing

This is a form of testing software that checks the aspects other than functionality, such as performance, usability and reliability. This optimizes how software is installed, configured, executed, and monitored. There are many different types of non-functional tests, such as

- ✓ Security Testing
- ✓ Accessibility Testing
- ✓ Usability Testing.

Test Case 11.2

The test case has a particular purpose in software testing, like evaluating the program's path or checking that an expectation is met. The test case is composed of restrictions on input, plans, testing techniques, and processes. Focused testing is built on test cases rather than random testing. You can construct test cases to cover the software being tested.

Unit Test – test case

Table 10 Unit Test – test case

Test Case Name	Unit Test		
Test Class			
Test Description			
Source of Data	**Test Steps**	**Expected Result**	**Actual Result**

Module Test – test case

Table 11 Module Test – test case

Test Case Name	Module Test		
Test Class			
Test Description			
Source of Data	**Test Steps**	**Expected Result**	**Actual Result**

Integration Test – test case

Table 12 Integration Test – test case

Test Case Name	Integration Test		
Test Class			
Test Description			
Source of Data	**Test Steps**	**Expected Result**	**Actual Result**

11.3 Unit Testing

Unit Test 1

CHAPTER 12 – IMPLEMENTATION

12.1 Training

The user must receive training on the new software when it is launched. Training is, therefore, essential to ensure maximum productivity and to limit the number of new employees. It also improves satisfaction and morale. There are many ways to train computer users after deploying software. When developing an efficient plan, it is important to consider the seriousness of the business. This table outlines the Cholo Ghuri training procedures.

Table 13 Admin training

Sr. No.	User	Training Scope	Time Allocation	Comment
1	Admin	Successfully create roles, permissions and staff. Also, successfully create apps, spots, match types, credentials for the app, highlights, series popular, alerts, subscribers, and notifications. What information can an administrator edit? Admin can update information he provided when registering with the system. The admin instructed the user on which field is needed to create the function.	5 hours	Admin knows all the features and functions.

12.2 Big Bang

Big bang acceptance has the fundamental benefit that software recommended does not have to be compatible with any other methods. It simplifies the design and implementation of the system. This is particularly useful in corporations with multiple incompatible systems. Comparing acquisition styles, big bangs are the most costly, even though they have fewer built-in learning opportunities and require more planning. The systematic approval of large-scale projects will clarify this preparation. By quickly deactivating the previous system, and activating the new one.

The test will be completed much faster with this approach. The data transfer may cause destruction or unpredictability, as the old system could have been corporate.

12.3 Scaling

Porch IT's developer team lead was responsible for scaling the project. They did not give any information or a report to the software developer trainee about scaling.

12.4 Load Balancing

Load balancing, or equalization is the process of balancing out consumer influence and distributing network traffic across several servers. Load balancing or load equalization is the process of distributing network traffic to a group of backend servers in order to counter the influence and impact from its consumers. The load balancer will transfer traffic if a server goes down to other servers. The load balancer sends automatic requests when a server is added to the group. This allows the system to continue working. In order to achieve a load balance that is optimal, optimum techniques should be developed.

Chapter 13 – Critical Appraisal and Evaluation

13.1 Objective Could be Met

This below functionality was primarily listed but could not finished:

- Provide reports and analytics on booking trends.
- Notify the user about price changes.
- Subscription System for users

Objective/goals – 1

Admin information is required during program installation. Only the administrator can access the login and registration features of the system, as well as the login credentials. The administrator can log in with the credentials provided when the program was installed. The administrator is sent directly to the dashboard with all the reports if the credentials entered are valid. The system will be up and running in a matter of seconds.

13.2 How much better could have been done

- **Notification management:** Notification trough the mail and phone would made the user more updated about any price changes.
- **Price Recommendation:** If there were any recommendation system of oncoming price changes it would be good

CHAPTER 14 – LESSONS LEARNED

14.1 Pre-Project-Review-closing

Cholo Ghuri is mainly an Resort Booking System. Our company follows a pre-defined process before starting any new projects. This includes gathering client requirements, creating a system plan, choosing an architecture, and then naming the project. The system was developed using Laravel, a PHP framework. I also transcribed the documentation to match the project.

14.2 What I have learned

Embarking on my internship program at Porch IT as a total fresher in this environment initially made me feel a bit apprehensive. However, I have always been eager to learn new things and embrace challenges. Despite the initial nervousness, I gradually gained self-confidence and adapted to the daily work routine. Over the course of the internship, I acquired valuable knowledge and skills that have significantly contributed to my professional growth.

One of the most important aspects of my acquired knowledge is the practical experience of working on a significant project and collaborating with other team members. Initially, the scale of the project and the prospect of teamwork seemed overwhelming, but I embraced the challenges and became more confident with each passing day. Through effective communication, coordination, and cooperation with my colleagues, I learned how to contribute to a larger project and work towards common goals.

Furthermore, the internship program exposed me to a real-world work environment, providing me with firsthand experience of how tasks are managed and executed in a professional setting. I gained insights into the importance of meeting deadlines, managing priorities, and maintaining clear communication channels. These skills are crucial in ensuring the smooth progress of projects and achieving optimal outcomes.

Additionally, the internship allowed me to enhance my technical skills in web development. I had the opportunity to work with new technologies, such as PHP, WordPress, Laravel, and REST API, which expanded my knowledge and proficiency in these areas. The hands-on experience of using development tools like VS Code, GitHub Desktop, and Trello further refined my technical skills and familiarized me with industry-standard practices.

Moreover, the internship program enabled me to develop essential interpersonal skills. By working in a formal environment and interacting with colleagues from diverse backgrounds, I improved my communication abilities and learned the importance of effective teamwork. Collaborating with supervisors and senior managers also provided valuable insights into professional conduct and relationship-building within an organization.

Overall, my internship at Porch IT has been a transformative experience, equipping me with practical knowledge, technical skills, and interpersonal abilities. It has given me the confidence to tackle challenges, adapt to new environments, and continuously learn and grow in my chosen field. I am grateful for the opportunity to acquire such valuable knowledge and skills during this internship program.

14.3 The problems I have faced

The internship at Porch IT Solutions posed several constraints and challenges that I had to overcome during my time as a web development intern. These challenges included:

1. Finding a Suitable Internship Program: It was initially challenging to find an internship program in the IT field that aligned with my career goals and provided the necessary technical experience. However, the opportunity to work at Porch IT as a web developer intern proved to be an excellent fit for me.
2. Transitioning from a Fresh Graduate to an Intern: As a recent graduate, transitioning into an internship program and adapting to the professional environment presented its own set of difficulties. I had to quickly adjust to the expectations and responsibilities of the role, while also proving myself as a valuable member of the team.
3. Continuity of the Internship Program: Maintaining the continuity of the internship program as an employee posed challenges due to various issues and boundaries. However, I persevered and found ways to navigate these obstacles, ensuring that I could continue gaining valuable experience and contributing to the organization.
4. Daily Task Assignments and Learning New Technologies: Each day during the internship, I was assigned new tasks that required me to work with technologies and concepts that I had little to no prior knowledge about. This presented a steep learning curve, but I embraced the challenges

and actively sought resources and guidance to overcome them. Through dedication and persistence, I was able to adapt and grow my skills in unfamiliar areas.

5. Problem-solving and Adaptability: The internship provided numerous opportunities to encounter and solve problems related to web development. I had to quickly adapt to these challenges, leveraging my problem-solving skills and utilizing available resources such as blogs, YouTube, Google, and Stack Overflow to find solutions and overcome any obstacles I faced.

Overall, despite the constraints and challenges I encountered during the internship, I embraced the learning opportunities presented to me and demonstrated resilience in adapting to new technologies and problem-solving. This experience has helped shape my professional growth and prepared me for future endeavors in the IT field.

14.4 What Solution Occurred

During my internship at Porch IT Solutions, I utilized various tools and technologies to support my work in web development. Some of the key tools and technologies I used include:

1. Visual Studio Code: Visual Studio Code is a powerful code editor that offers a wide range of features including debugging, task execution, and version control integration. It provides a user-friendly interface and allows developers to customize themes, use code snippets, and configure language support. Visual Studio Code is highly extensible and supports a variety of programming languages.
2. GitHub Desktop: GitHub Desktop is a desktop application that simplifies version control and collaboration using Git and GitHub. With GitHub Desktop, I was able to create new local repositories, track changes, commit code, and push/publish changes to GitHub. It provided a seamless workflow for managing code and collaborating with teammates. Additionally, I could easily create branches and merge changes to keep the codebase organized.
3. Trello: Trello is a popular visual work management tool that enables teams to plan, organize, and track their tasks and projects collaboratively. It provides a visual interface with boards, lists, and cards, allowing users to create, prioritize, and assign tasks. Trello

helped me stay organized and keep track of my project tasks, deadlines, and progress. It facilitated efficient communication and task management within the team.

These tools and technologies played a crucial role in enhancing my productivity, streamlining the development process, and facilitating effective collaboration with team members. By leveraging these tools, I was able to effectively manage my codebase, track project progress, and stay organized throughout the internship period.

CHAPTER 15 – CONCLUSION

15.1 Summary of the project

Cholo Ghuri is a Resort Booking System that aims to provide easy online booking solutions for customer. The system primary goal is to provide the user available resort, guide them towards checking and managing them. In addition to it, Admin and staff can easily manage users and resort data. User can easily access their booking and modify if necceasry. The system is built with Laravel 8 and php, alongside MySQL database.

15.2 Goal of the project

The purpose of the Resort Booking System is to provide a user-friendly and efficient platform for customers to book and manage their reservations at a resort. The system aims to streamline the booking process, enhance customer satisfaction, and improve the resort's overall operational efficiency.

- User registration and authentication
- Resort and room information management
- Online booking and payment processing
- Reservation modification and cancellation
- Availability tracking and management
- Reporting and analytics

15.3 Success of the project

Maintaining high-quality work is paramount for the reputation and success of any company. As an intern at Porch IT, I recognized the significance of delivering work of exceptional quality. I consistently strived to produce outputs that met or exceeded the company's standards. By prioritizing attention to detail, thoroughness, and precision, I contributed to upholding and enhancing the company's reputation.

15.4 What I have done in the documentation

From the very beginning, I was required to follow the structure of academic documentation when doing the documentation. There are diagrams in this document that show the various features. The tables and information in this document were all prepared with MS Word. All figures and diagrams, however, were made using third party libraries to conduct time boxing and numerous illustrations as well as analyze data.

15.5 Value of the Project

We are committed to delivering the best industry practices coupled with emerging practical and dependable technological solutions. Our team is specialized in every aspect of software development. Our values reflect in

1. Developing lasting beneficial relationships with our clients
2. Confidentiality concerning work
3. Learning and Growing atmosphere
4. Integrity expressed
5. Independent and Reliable business objectives

15.6 My Experience

I had the opportunity to gain practical experience and understand the working processes of the organization. As a novice in the professional environment, my primary responsibility was to familiarize myself with the team's operations and actively participate in various meetings and discussions.

In the initial phase of my internship, I received two weeks of training. During this period, I focused on learning PHP fundamentals, Laravel project setup, and mastering object-oriented programming concepts. With dedication and focus, I quickly grasped these concepts and gained a solid foundation.

Following the training, I was assigned to work on several WordPress projects, approximately 10 to 12 in total. My role involved customizing premium WordPress themes to meet the specific requirements of users. This experience allowed me to enhance my skills in WordPress development and customization.

After a couple of months, I transitioned to the web development team, which was primarily focused on building dynamic websites. At the time, our team was working on an e-commerce website. As a member of the team, I actively contributed to the website's development tasks. My responsibilities included working on both the backend and frontend aspects of the website.

On the backend, I handled tasks related to website development, ensuring smooth functionality and efficient data management. Additionally, I took on responsibilities related to frontend design, ensuring that the website was responsive and optimized for various devices.

Throughout my internship, whenever I encountered errors or faced challenges, I proactively sought assistance from various resources such as blogs, YouTube tutorials, Google searches, and Stack Overflow. This approach allowed me to overcome obstacles and further develop my problem-solving skills.

Overall, the workload during my internship at Porch.com provided me with valuable hands-on experience in software development, allowing me to enhance my technical skills and gain practical knowledge in web development.

REFERENCES

[1] *Agile Business*. (n.d.). Retrieved from https://www.agilebusiness.org/business-agility/what-is-dsdm.html#:~:text=What%20is%20DSDM%3F-,DSDM%20is%20an%20Agile%20method%20that%20focuses%20on%20the%20full,new%20iterative%20way%20of%20working.

[2] Clokify. (n.d.). Retrieved from https://clockify.me/timeboxing

[3] Gozayaan. (n.d.). Retrieved from gozayaan.com

[4] investopedia. (n.d.). Retrieved from https://www.investopedia.com/terms/c/cost-benefitanalysis.asp

[5] osarome. (n.d.). Retrieved from https://osarome.blogspot.com/2011/10/1-technical-feasibility-2-operational.html

[6] Resortbookingbd. (n.d.). Retrieved from https://resortbookingbd.com/

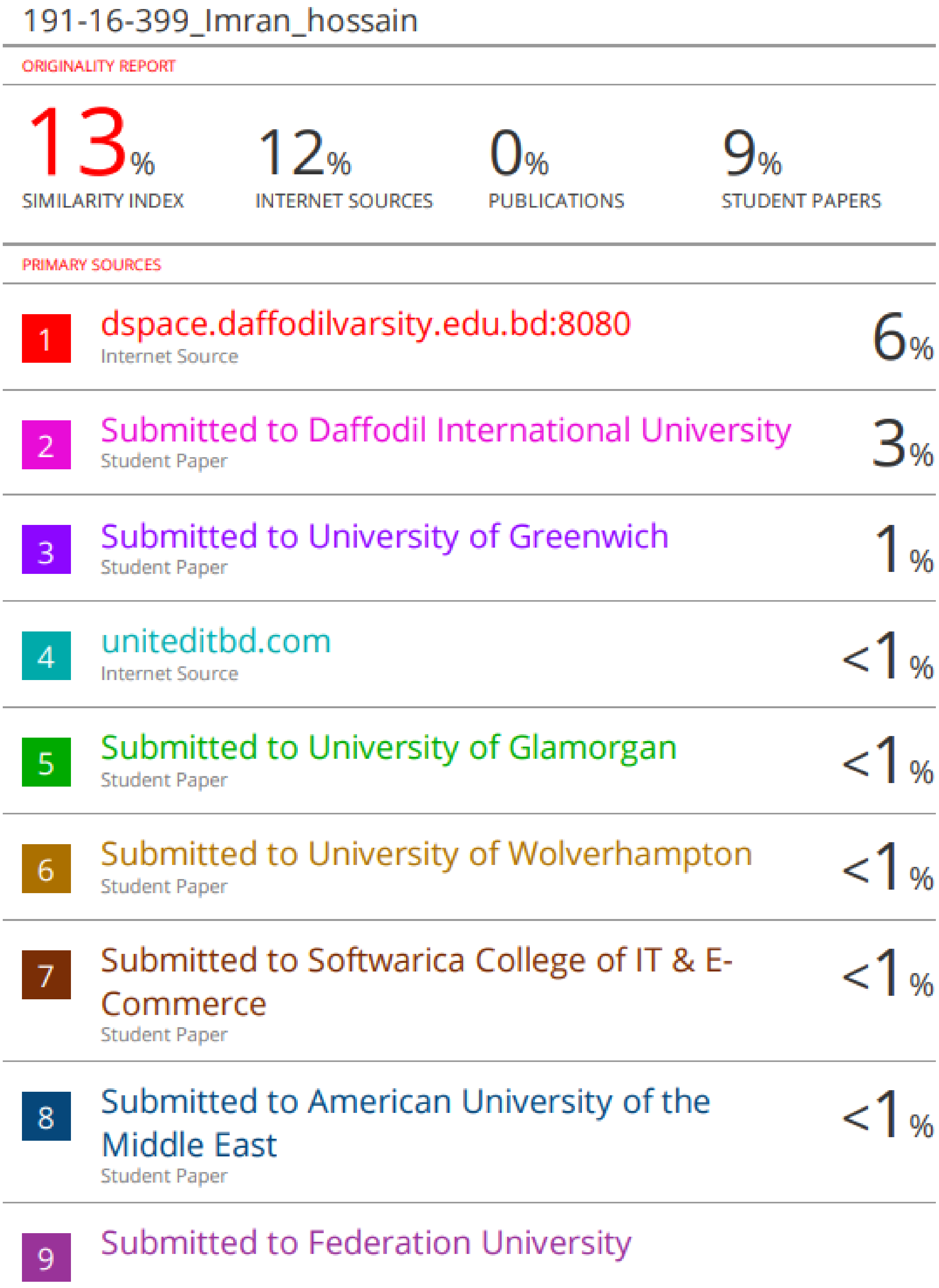
[7] Sharetrip. (n.d.). Retrieved from sharetrip.com

[8] Simplilearn. (n.d.). Retrieved from https://www.simplilearn.com/time-boxing-in-agile-practices-article#:~:text=A%20timebox%20is%20a%20set,releasable%20update%20to%20the%20software.

[9] Tutorials Point. (n.d.). Retrieved from https://www.tutorialspoint.com/techniques-for-requirements-gathering#:~:text=Documentation%20analysis-,Interviews,stages%20of%20project%20life%20cycle

[10]Whitefiled. (n.d.). Retrieved from https://mark-whitfield.com/prince2-project-templates-download/dsdm-dynamic-systems-development-method/

[11]workbreakdownstructure. (n.d.). Retrieved from https://www.workbreakdownstructure.com/



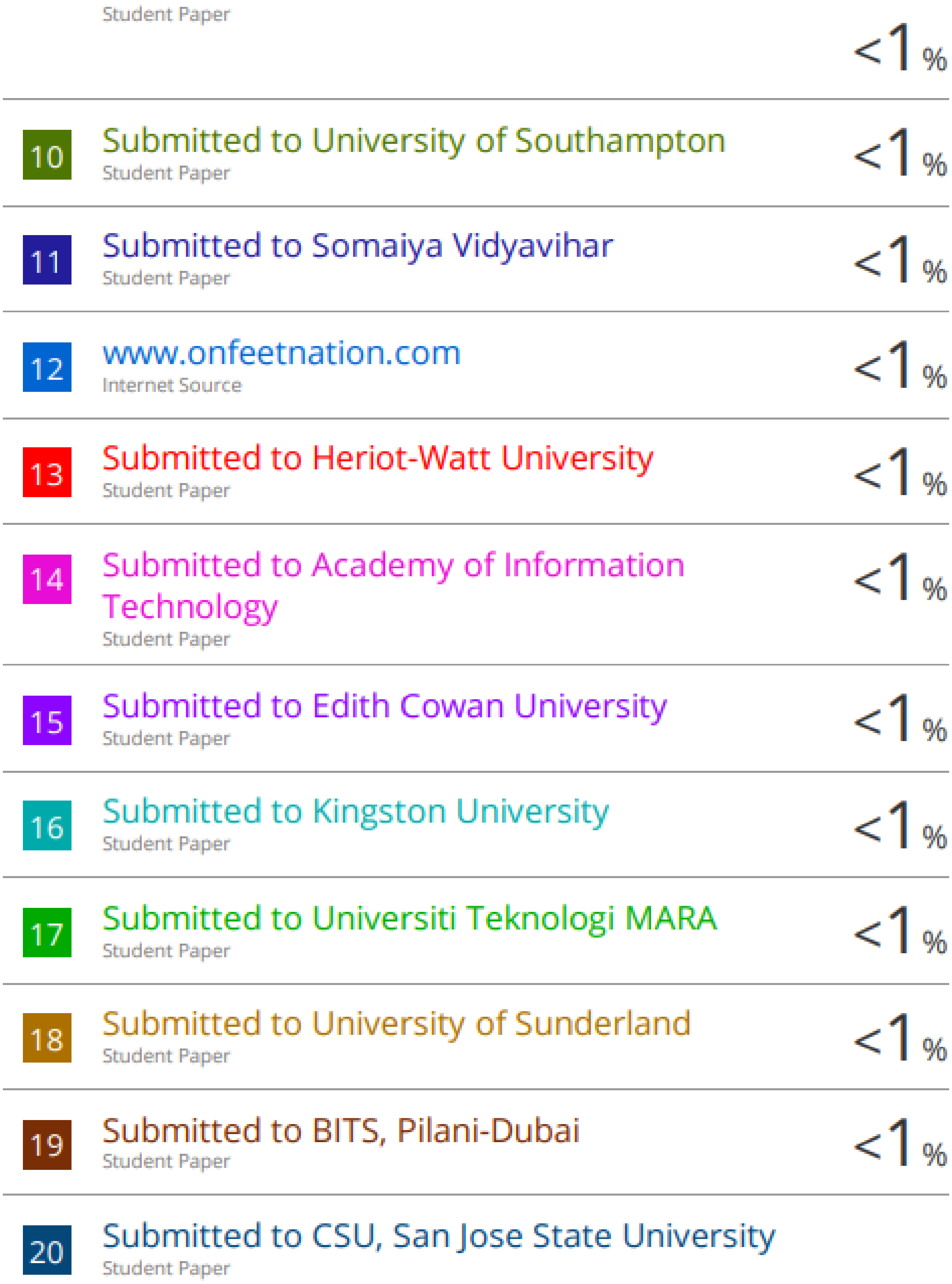
191-16-399_Imran_hossain

ORIGINALITY REPORT

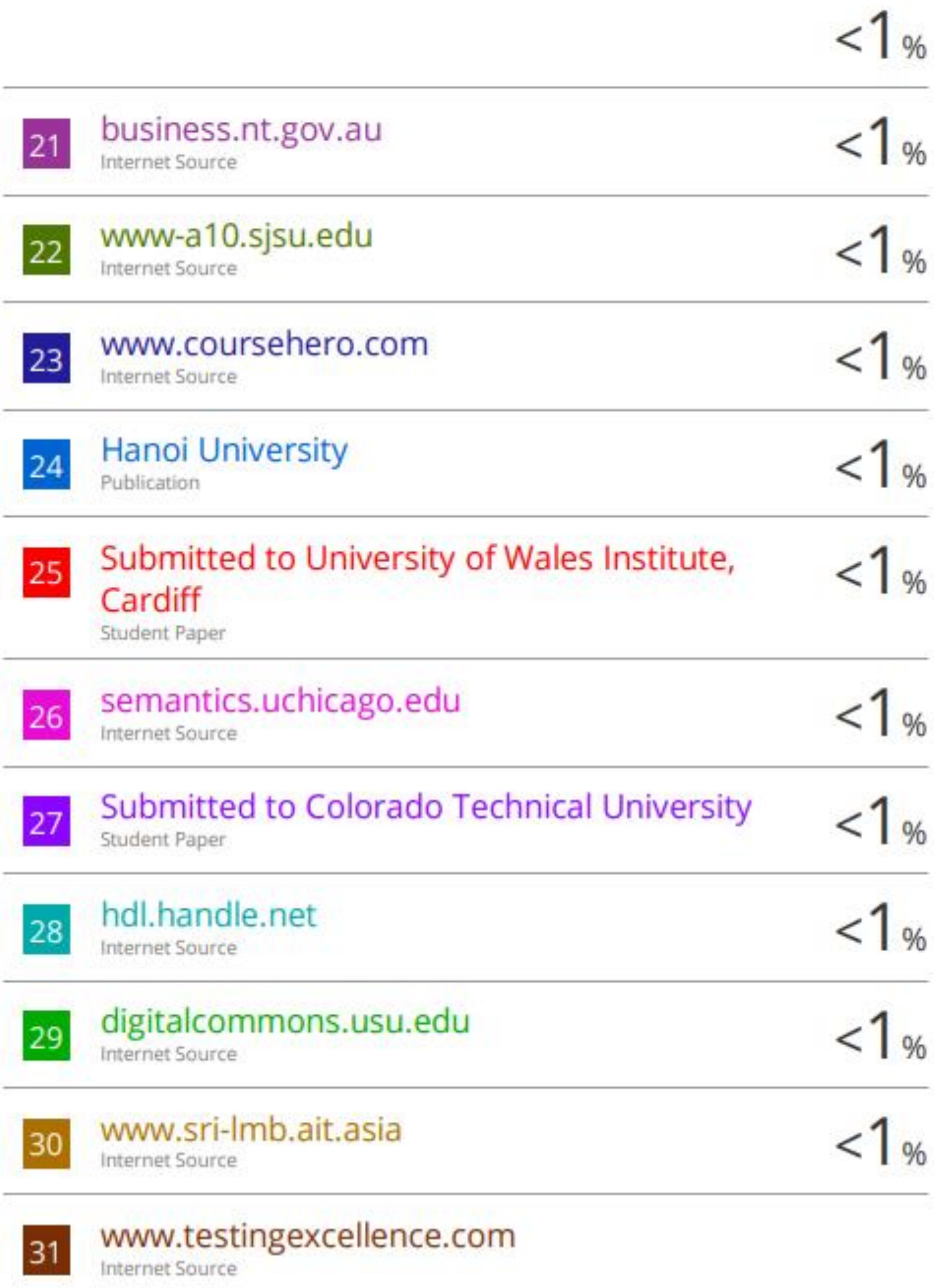
13%	12%	0%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	6%
2	Submitted to Daffodil International University Student Paper	3%
3	Submitted to University of Greenwich Student Paper	1 %
4	uniteditbd.com Internet Source	<1 %
5	Submitted to University of Glamorgan Student Paper	<1 %
6	Submitted to University of Wolverhampton Student Paper	<1 %
7	Submitted to Softwarica College of IT & E-Commerce Student Paper	<1 %
8	Submitted to American University of the Middle East Student Paper	<1 %
9	Submitted to Federation University	



	Student Paper	<1 %
10	Submitted to University of Southampton Student Paper	<1 %
11	Submitted to Somaiya Vidyavihar Student Paper	<1 %
12	www.onfeetnation.com Internet Source	<1 %
13	Submitted to Heriot-Watt University Student Paper	<1 %
14	Submitted to Academy of Information Technology Student Paper	<1 %
15	Submitted to Edith Cowan University Student Paper	<1 %
16	Submitted to Kingston University Student Paper	<1 %
17	Submitted to Universiti Teknologi MARA Student Paper	<1 %
18	Submitted to University of Sunderland Student Paper	<1 %
19	Submitted to BITS, Pilani-Dubai Student Paper	<1 %
20	Submitted to CSU, San Jose State University Student Paper	



		<1%
21	business.nt.gov.au Internet Source	<1%
22	www-a10.sjsu.edu Internet Source	<1%
23	www.coursehero.com Internet Source	<1%
24	Hanoi University Publication	<1%
25	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
26	semantics.uchicago.edu Internet Source	<1%
27	Submitted to Colorado Technical University Student Paper	<1%
28	hdl.handle.net Internet Source	<1%
29	digitalcommons.usu.edu Internet Source	<1%
30	www.sri-lmb.ait.asia Internet Source	<1%
31	www.testingexcellence.com Internet Source	