Final Year Project Report
Full Unit – Final Report

Tourism Guide APP

Thabiso Seleke 201902015

A report submitted in part fulfilment of the degree of
BSc (Hons) in Computer Science
Supervisor: T.Taukong

Department of Computer Science
Royal Holloway, University of London

Declaration

This assignment has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Student Name: Thabiso Nathaniel Seleke

Date of Submission: 03/05/2023

Signature: T.S

Acknowledgment

For the project to be an absolute success, I had received an immense support from a multitude of people to whom I owe gratitude to. First, I am very grateful to the Almighty God whose love, peace and mercy led me to complete our both my project and succeed in my studies. Secondly, I thank my Co-examiner Mr Ngau for the support rendered during this course of the research and implementation of the project and who have granted all their time, encouragement to pursue this project.

Lastly but not least, sincere appreciation goes to my supervisor Ms Taukobong for the guidance she has given me through the completion of this project report. I also extend my sincere gratitude to our academic colleagues at the University who have encouraged us during the period of the study

Table of Contents

Chapter 1: Introduction

Tourism is an important industry for many countries, generating significant economic benefits and providing employment opportunities. With the increasing popularity of smartphones and mobile devices, there is a growing demand for tourism guide apps that can help visitors explore new destinations and make the most of their travel experience. The proposed project is aimed at developing a Tourism Guide App that can provide users with useful information about popular tourist attractions, local destinations, hotels, and other amenities.

Problem Statement:

Despite the significant contribution of the tourism industry to the Botswana economy, there is a lack of a centralized platform for tourists to access reliable and up-to-date information about tourist attractions, accommodations, events, and activities. Additionally, travel agents struggle to promote their services and reach a wider audience. This leads to disconnect between travelers and travel agents, resulting in missed opportunities and a less-than-optimal experience for both parties. Therefore, there is a need for a tourism guide app that provides comprehensive information about Botswana's tourist attractions and serves as a platform for travel agents to showcase their services, thus promoting tourism in the country.

Motivation:

The Tourism Guide App is designed to address these challenges and provide users with a convenient and user-friendly platform for accessing information about local tourist destinations. By providing accurate and up-to-date information about local attractions, destinations, hotels, and other amenities, the app can help visitors make informed decisions about their travel plans and make the most of their time in the region.

Proposed Solution:

The proposed solution is a Tourism Guide App that is designed to provide users with a comprehensive guide to local tourist attractions and amenities. The app will feature an intuitive user interface that is easy to navigate, and will provide users with access to a wide range of information about local attractions, including maps, reviews, and ratings.

Project Objectives:

The primary objective of the project is to develop a Tourism Guide App that can provide users with accurate and up-to-date information about local tourist attractions, destinations, hotels, and other amenities. In addition, the project aims to develop an app that is user-friendly and accessible to users with varying levels of technical proficiency.

Project Scope:

The scope of the project will include the development of a Tourism Guide App for web platforms. The app will be designed to provide users with access to a wide range of information about local tourist attractions, including maps, reviews, and ratings. The app will also feature a user-friendly interface that is easy to navigate, and will be accessible to users with varying levels of technical proficiency. The project will also include a booking feature which user can book services. Below are two types of users:

Travel Agent

- Insert destinations
- Delete destinations
- edit profile

Traveler

- Register and login profile
- Search for car destinations
- Edit profile

Table 1.0 WORK PLAN

Task	**Activity**	**Duration**	**Dependencies**
T1	Submission of project proposal	1 day	
T2	Approval of the project proposal	3 days	1
T3	Allocation of supervisors	3 days	2
T4	Project planning	5 days	1
T5	Requirements elicitation and data gathering	7 days	4
T6	Analysis	5 days	5
T7	Software design	1 day	4,5
T8	Program design	12 days	7,6
T9	Updating computation	29 days	4,5,6,7,8
T10	Program Implementation	5 days	7,8
T11	Maintenance	3 days	10
T12	Actual cost	10 days	10
T13	Report writing	3 days	1
T14	Finalize documentation	1 day	1,2,3,4,5,6,7,8,9,10,11,12,13
T15	Presentation	1 day	14

Gantt Chart

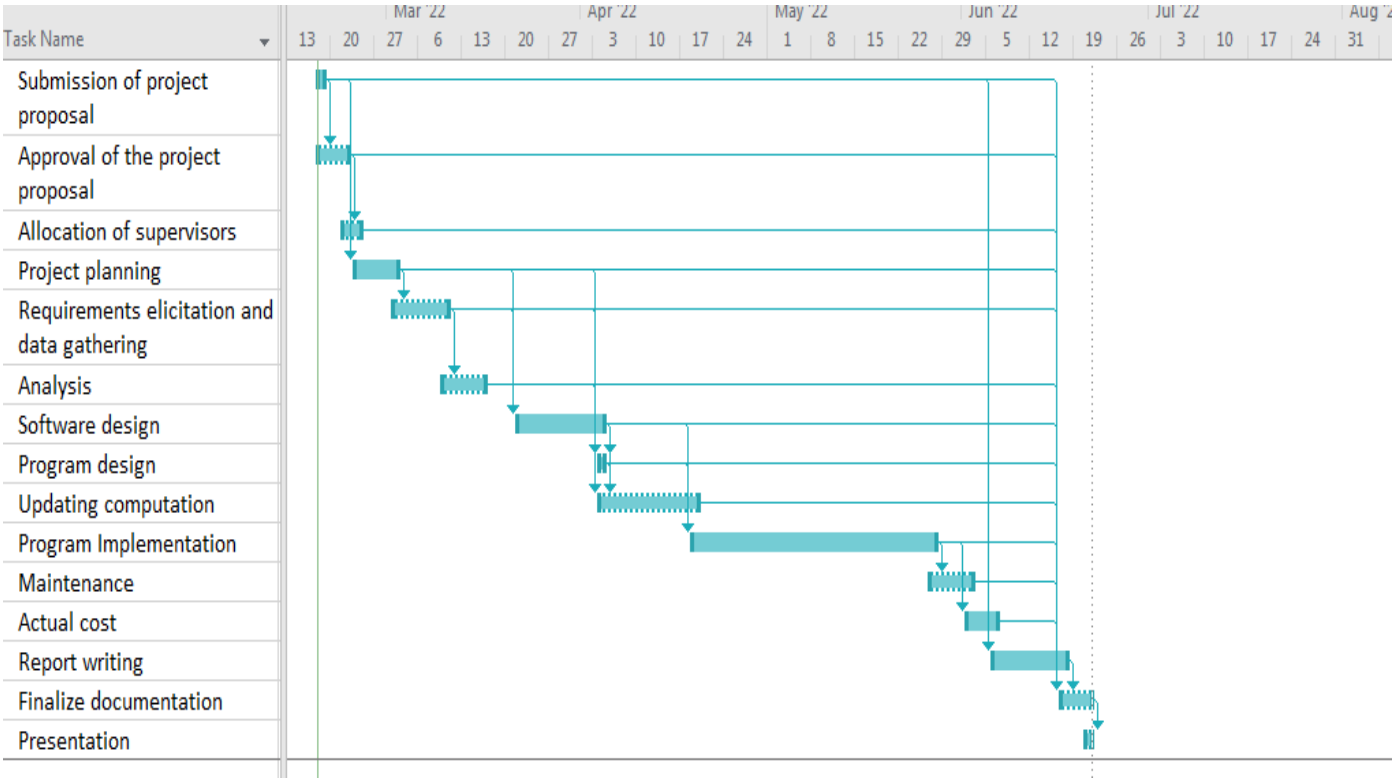


Figure 1.0

Resources

SOFTWARE REQUIREMENT

Table 1.1 shows the software requirements for the proposed system.

Table 1.1 Development software requirement

Software	**Description / Purpose**
XAMPP	Server to run localhost
Visual Studio Code	Use to code the program of the project, especially connection application to the database
PhpMyAdmin	To manage MySQL database.
Microsoft Word 365	Documentation of application

MySQL	For system database
Bootstrap	Framework to format the system
Lucid chart	To design Context Diagram, Data Flow Diagram and Entity Relation Diagram
Google Chrome/Microsoft Edge	To view webpages and website of the system

HARDWARE REQUIREMENT

Table 1.2 shows the hardware requirement for the proposed system.

Table 1.2 Development hardware requirement

Hardware	**Description / Purpose**
Laptop	hp laptop
Processor	Intel ® Core ™ i3 – 5005U CPU @ 2.00 GHz 2.00
Random Access Memory (RAM)	4.00 GB
Operating system	Windows 11
System type	64-bit Operating System

Methodology:

The development of the Tourism Guide App will follow an Agile Software Development approach. This approach is characterized by iterative development and continuous feedback, which allows for flexibility in the development process and the ability to respond to changing requirements and feedback from users.

Data Gathering Methods and Techniques: To gather data for the development of the Tourism Guide App, the following methods and techniques will be used:

1. Literature Search: A literature search will be conducted to gather information on the current state of the tourism industry in Botswana, and to identify the features and functionality required for a successful tourism guide app. Sources will include academic journals, industry reports, and online resources(cited in the referencing).
2. Interviews: Interviews will be conducted with travel agents and tourists to gather information on their needs and requirements for a tourism guide app. The interviews will be structured to ensure consistency and will be conducted in-person or via online video conferencing.
3. Web searching: Web searching will be used to gather information on local tourist attractions, destinations, and accommodations. This will involve visiting websites, social media platforms, and other online resources.

Software Engineering Approach: The following methods and techniques will be used for systems design, development, and testing of the Tourism Guide App:

1. Object-Oriented Analysis and Design (OOAD): OOAD will be used to identify and define the system requirements and to design the system architecture. This will involve creating use cases, class diagrams, and sequence diagrams.
2. Prototyping: Prototyping will be used to create a working prototype of the app, which will be used to gather feedback from users and stakeholders. The prototype will be developed using HTML, CSS, and JavaScript.
3. Relational Data Modelling: Relational data modelling will be used to design the database schema and to ensure that the system can efficiently store and retrieve data.
4. Object-Oriented Programming (OOP): OOP will be used to develop the system using NodeJS and JavaScript. OOP enables the creation of reusable code, which reduces development time and ensures code consistency.
5. Functional Testing: Functional testing will be used to test the app's functionality and ensure that it meets the system requirements. This will involve testing individual features and the app as a whole, using manual and automated testing techniques.

Chapter 2: Literature Review

Introduction

The purpose of this literature review is to explore the background research conducted to develop the Tourism Guide Web App. This literature review will provide an overview of the current state of the art, major tools/techniques in use, likely directions for advancement in the field, and how the project fits into the picture. This review will also include background information about an organization used as a case study for the project's use.

Web-Based Technologies

Web-based technologies are a crucial aspect of the Tourism Guide Web App development process. The use of web-based technologies such as HTML, CSS, and JavaScript has enabled the creation of dynamic and interactive web pages. The use of web-based technologies has also facilitated the development of responsive web designs, which can adjust to various device screens. According to Tsai et al. (2019), the development of web-based technologies has resulted in the proliferation of web applications, which have become an essential part of modern-day life.

Current State of the Art: The development of tourism guide web applications has been largely driven by advancements in web-based technologies, including cloud computing, mobile computing, and social media (Alzahrani et al., 2019). These technologies have enabled the creation of more sophisticated and user-friendly applications that can be accessed from anywhere at any time. Many tourism guide web applications now incorporate features such as location-based services, personalized recommendations, and real-time updates to enhance the user experience.

Major Tools and Techniques: The development of tourism guide web applications typically involves the use of a range of tools and techniques, including software development frameworks, programming languages, and databases (Abdel-Basset et al., 2020). Some of the most popular frameworks for web application development include React, Angular, and Vue. These frameworks provide developers with a set of tools and libraries to create dynamic, responsive, and scalable web applications. In terms of programming languages, JavaScript and Python are widely used for

front-end and back-end development, respectively. Databases such as MySQL, PostgreSQL, and MongoDB are commonly used for data storage and management (Muharraqi et al., 2019).

Distributed Systems

Distributed systems are a critical component of the Tourism Guide Web App development process. The use of distributed systems has enabled the creation of scalable web applications that can handle large amounts of traffic. According to Tanenbaum and Steen (2017), distributed systems have become an essential part of modern web applications, as they can facilitate the development of highly available and fault-tolerant systems.

Case Study: Airbnb

Airbnb is an online marketplace that enables people to rent out their properties to travelers. The company's success has been attributed to its user-friendly platform and its ability to provide personalized services to its users. The company uses a range of web-based technologies such as HTML, CSS, and JavaScript to create a dynamic and interactive platform. The company also uses AI technologies such as machine learning to make personalized recommendations to its users. Finally, the company's use of distributed systems has enabled it to handle large amounts of traffic and provide highly available services to its users.

Chapter 3: SYSTEM INVESTIGATION & ANALYSIS

3.1 Functional/Processing Requirement Analysis:

The Tourism Guide App is designed to provide users with a comprehensive guide to local tourist attractions and amenities. The app will have the following functional requirements:

Functional Requirements:

- User Authentication: The app should provide a secure and user-friendly login and registration process for users, including the ability to sign in through social media accounts.

- Tourist Attractions Information: The app should provide detailed information about local tourist attractions, including historical sites, natural wonders, cultural events, and other points of interest. Users should be able to search and filter attractions based on their preferences and receive recommendations based on their location.

- Accommodations Information: The app should provide information about hotels, resorts, and other accommodations in Botswana, including room rates, availability, and user

reviews. Users should be able to search and filter accommodations based on their preferences and book directly through the app..

- Transportation Information: The app should integrate with local transportation services, including car rental agencies, taxis, and public transportation, making it easy for users to navigate the region. Users should be able to search and book transportation options directly through the app.

- Travel Packages: The app should allow travel agents and tour operators to create and promote customized travel packages to users, including tours, transportation, accommodations, and other services. Users should be able to book travel packages directly through the app.

Non-Functional Requirements:

- User Interface: The app should have an intuitive and user-friendly interface that is easy to navigate and understand, providing a seamless and efficient user experience.

- Performance: The app should be fast and responsive, with minimal loading times and downtime.

- Security: The app should provide a secure environment for user data and transactions, including encryption and authentication measures.

- Compatibility: The app should be compatible with a range of mobile devices and operating systems, including iOS and Android.

- Scalability: The app should be designed to accommodate future growth and expansion, with the ability to handle increasing numbers of users and transactions.

This system design specification document outlines the technical details for the development of a Tourism Guide App. The app will be designed to provide users with comprehensive information about popular tourist attractions, local destinations, hotels, and other amenities in Botswana. The app will also serve as a platform for travel agents and tour operators to promote their services and offer customized travel packages to users.

3.2 Data Analysis:

The Tourism Guide App will require a database to store information about tourist attractions, accommodations, destinations, and user information. The database will be designed using MySQL, and the following data modelling and specification techniques will be used:

1. Entity-Relationship Diagrams (ERD): ERDs will be used to model the relationships between the different entities in the database, including users, tourist attractions,

accommodations, and destinations. This will help to ensure that the database is well-structured and organized.

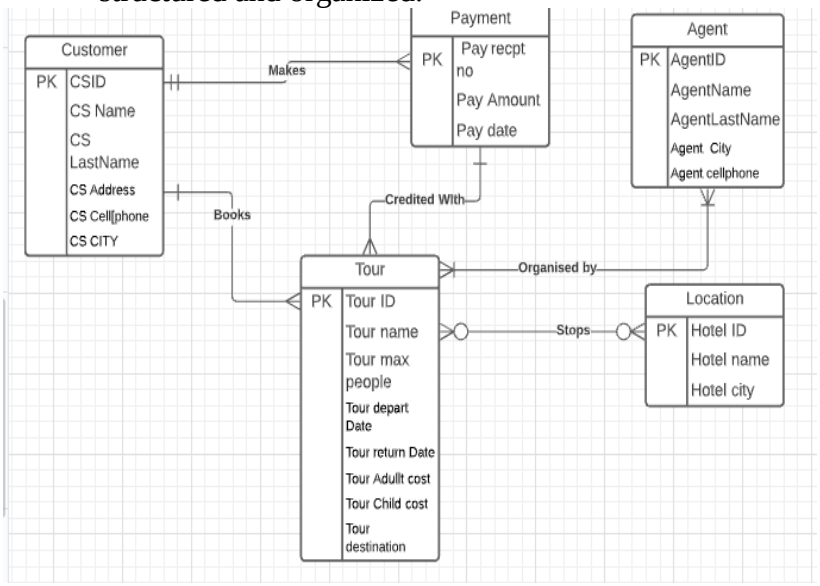


Figure 1.1

The following are the entities that will be included in the database:

1. Users: This entity will store information about registered users, including their names, email addresses, and passwords.
2. Tourist Attractions: This entity will store information about local tourist attractions, including the attraction name, location, description, photos, reviews, and ratings.
3. Accommodations: This entity will store information about local accommodations, including the accommodation name, location, description, photos, reviews, and ratings.
4. Destinations: This entity will store information about local Destinations, including the Destinations name, location, description, photos, reviews, and ratings.
5. Travel Agents: This entity will store information about travel agents, including their names, contact information.

The following are the relationships between the entities in the database:

1. Users can leave reviews and ratings for tourist attractions, accommodations, and destinations.
2. Tourist attractions, accommodations, and destinations can have multiple reviews and ratings.
3. Travel agents can be associated with multiple tourist attractions, accommodations, and destinations.

3.3 Input Requirement Analysis:

The Tourism Guide App will require various inputs to support the functionalities and tasks of the system. The following are the inputs that will be required:

1. User Information: Users will be required to input their personal information, including their names, email addresses, and passwords, to create an account and access the features of the app.

2. Search Queries: Users will be able to search for tourist attractions, accommodations, and destinations using keywords and location-based searches.

3. Travel Agent Information: Travel agents will be required to input their names, contact information, and services offered to be included in the app's directory.
4. Travel Agent: Travel Agent users will be able to access the system to manage and update information in the database, including tourist attractions, accommodations, and destinations.
5. Booking Information: Travel users will be able to make bookings for accommodations and destinations through the app, which will require inputting their personal and payment information.
6. User Preferences: Users will be able to specify their preferences for tourist attractions, accommodations, and destinations, such as their preferred cuisine or activity type.

3.4 Output Requirement Analysis:

The Tourism Guide App will generate various outputs to support the functionalities and tasks of the system. The following are the outputs that will be generated:

Tourist Attraction Information: The app will provide users with detailed information about tourist attractions, including descriptions, photos, reviews, ratings, location on a map, and operating hours.

Accommodation Information: The app will provide users with detailed information about accommodations, including descriptions, photos, reviews, ratings, location on a map, and booking options.

Destination Information: The app will provide users with detailed information about destinations, including descriptions, photos, reviews, ratings, location on a map, and menu options.

User Profile Information: The app will display users' profile information, including their name, profile picture, and saved preferences.

Booking Confirmation: The app will generate a booking confirmation message for users who book accommodations or destinations through the app, including booking details and a confirmation number.

3.5 Interface Requirement Analysis:

The interface of the Tourism Guide App will be designed to be intuitive, user-friendly, and visually appealing. The following are the interface requirements and considerations for the app:

1. User Analysis: The app will have two main types of users: tourists and travel agents. Tourists will use the app to plan their travel itineraries, explore tourist attractions, book accommodations and destinations. Travel agents will use the app to promote their services and reach a wider audience.
2. User-Task Analysis: The app will support a wide range of user tasks, including searching for tourist attractions, accommodations, and destinations, booking accommodations and destinations, exploring maps, writing reviews and ratings, and managing user profiles.
3. Functional Considerations/Issues: The app will be designed to be accessible and usable by users with different levels of digital literacy. The interface will be optimized for mobile devices, with a clean and simple design that allows for easy navigation and interaction.

3.6 Dynamic model specification of the system

To specify the dynamic model of the system, a combination of formal and informal language specifications will be utilized, with an emphasis on formal specifications. The following diagrams will be used to show the behavior and composition of the system to realize the functional requirements:

1. Use Case Diagram: A use case diagram will be used to identify all the actors and their interactions with the system. This will help to define the scope of the system and ensure that all functional requirements are considered.

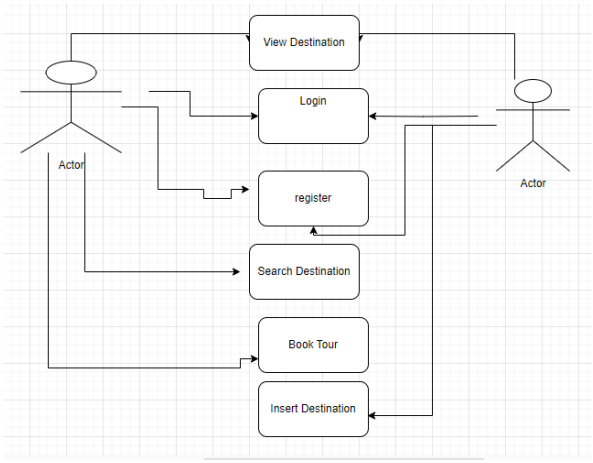


Figure 1.2

In this figure 2.0 the actor on the left is the traveler and the actor on the right is the travel agent

USE CASE 1:Search for a Tourist Attraction
Actor: Traveler
Description: This use case describes the process of searching for a tourist attraction.
Flow of events: 1. User enters a search query for a tourist attraction. 2. System displays a list of matching tourist attractions. 3. User selects a tourist attraction from the list. 4. System displays detailed information about the selected tourist attraction.

USE CASE 2:Create an Account
Actor: Traveler and Travel Agent
Description: This use case describes the process of creating a user account.
Flow of events: 1. User clicks on the "Create an Account" button. 2. System displays the registration form. 3. User fills out the registration form with the required information. 4. User submits the registration form. 5. System creates a new user account and sends a confirmation email to the user

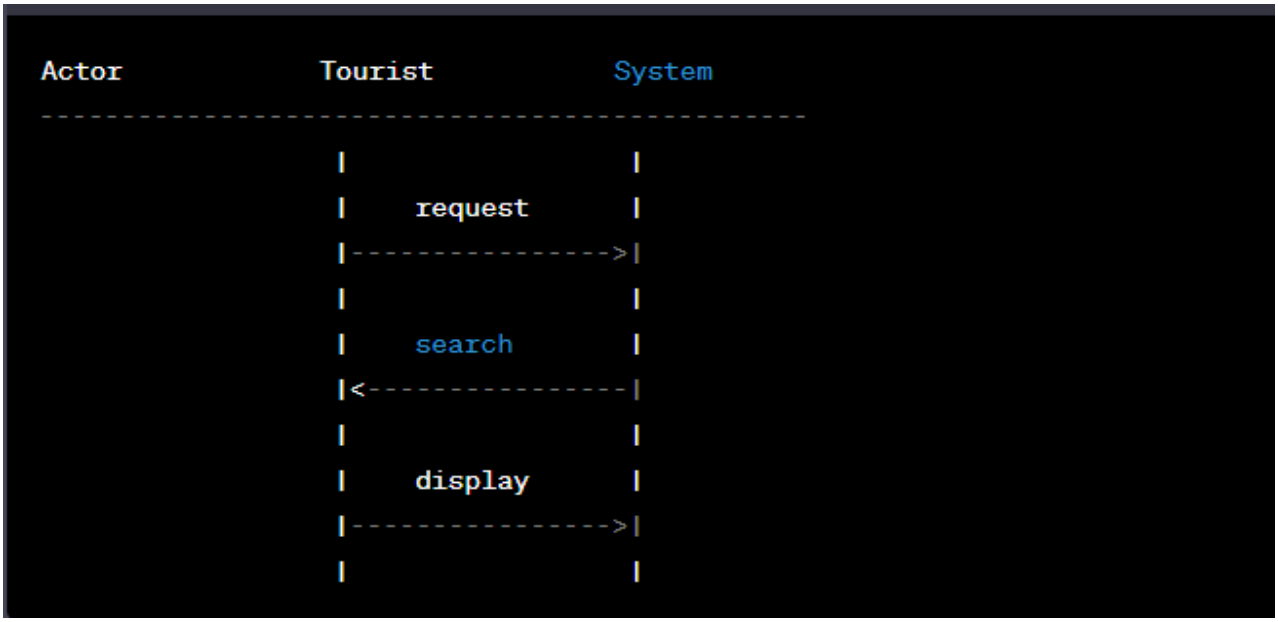
USE CASE 3:View Tourist Attraction Details
Actor: Traveler
Description: This use case describes the process of viewing detailed information about a tourist attraction.
Flow of events: 1. User selects a tourist attraction from a list. 2. System displays detailed information about the selected tourist attraction.

USE CASE 4:Add Review of Tourist Attraction
Actor: traveler
Description: This use case describes the process of adding a review of a tourist attraction.
Flow of events: 1. User selects a tourist attraction from a list. 2. System displays detailed information about the selected tourist attraction. 3. User clicks on the "Add Review" button. 4. System displays the review form. 5. User fills out the review form with the required information. 6. User submits the review form. 7. System adds the review to the tourist attraction's reviews section.

The Tourism Guide App is the system being modeled, which includes various use cases such as view tourist attractions, search for tourist attractions and hotels, view hotel details, promote services, and add/edit events.

2. Sequence Diagrams: Sequence diagrams will be used to describe the interaction between the actors and the system for each use case. This will help to show the dynamic behavior of the system and ensure that the functional requirements are being met.

Sequence diagram for the "Search for Tourist Attraction" use case:



In this diagram, the Tourist actor initiates the use case by requesting a search for tourist attractions. The system receives the request and performs a search, returning the results to the Tourist for display.

Chapter 4: DESIGN

4.1 Functional design specification

The Tourism Guide App is designed to provide tourists with information about tourist attractions, accommodations, and dining options in a given location. The app allows users to search for specific attractions, accommodations, or dining options based on their location or preferences.
To achieve this functionality, we have designed the following processing tasks:

1. Search: Allows users to search for tourist attractions, accommodations based on their location or preferences. This processing task is accomplished using a search algorithm that takes user input and compares it to a database of tourist attractions, accommodations.
2. View Details: Allows users to view details about tourist attractions, accommodations, or dining options. This processing task is accomplished using an object model that retrieves the details of the selected attraction, accommodation, or dining option from the database.
3. Insert destination: Allows users to add tourist attractions. This processing task is accomplished using a structure chart that adds the selected attraction in the database.
4. Remove from destination: Allows users to remove tourist attractions. This processing task is accomplished using a structure chart that removes the selected attraction in the database.

4.2 Interface Design specification

Interface Design Specification:
The interface design of the Tourism Guide App is intended to be user-friendly and easy to navigate. The interface is designed to be visually appealing to attract users and encourage them to engage with the app. The following are the specifications for the interface design:

1. Home Screen: The home screen will have a simple design with a welcoming image of the city and the name of the city. The screen will provide three options to users: explore, search,

and favorites. The Explore button will lead to a page where users can browse the attractions and activities in the city. The Search button will lead to a page where users can search for specific attractions or activities. The Favorites button will lead to a page where users can see their saved favorite attractions or activities.

2. Gallery Screen: The Gallery screen will have a list of categories such as historical places, natural attractions, cultural activities, and others. Users can select a category to explore further, and the app will display a list of attractions or activities in that category. Users can tap on an item to see more information about it.
3. Search Screen: The search screen will have a search bar where users can type in the name of a specific attraction or activity. The app will display a list of matching results, and users can tap on an item to see more information about it.

4.3 Input design specification

In the input design specification, all the forms and input screens used in the system should be described in detail. This includes specifying the types of input fields used, the layout of the screen or form, and any data validation or error handling processes that are in place.

For the tourism guide app, the input design specification includes the following:

- A search form allowing users to search for tourist attractions by name.
- A registration form allowing new users to create an account
- A login form allowing registered users to log in to the system
- A form allowing users to add new tourist attractions to the system, including fields for name, location, description, and images
- A form allowing users to update their account information, including fields for name, email, and password.

4.4 Output design specification

The output design specification for the Tourism Guide App is as follows:

1. Map and Location Information: The app will display maps of tourist destinations, along with location-specific information such as addresses, contact details, and opening hours.
2. Recommendations and Reviews: The app will provide recommendations and reviews for tourist destinations based on the user's preferences and previous searches. It will also allow users to leave their own reviews and ratings.
3. Itinerary Planning: The app will allow users to plan their itineraries, including creating a list of places they want to visit, scheduling their visits, and setting reminders.
4. Transportation Information: The app will provide information on transportation options such as train services, car rentals, and bus services.

4.5 Database/file and data structures design specification

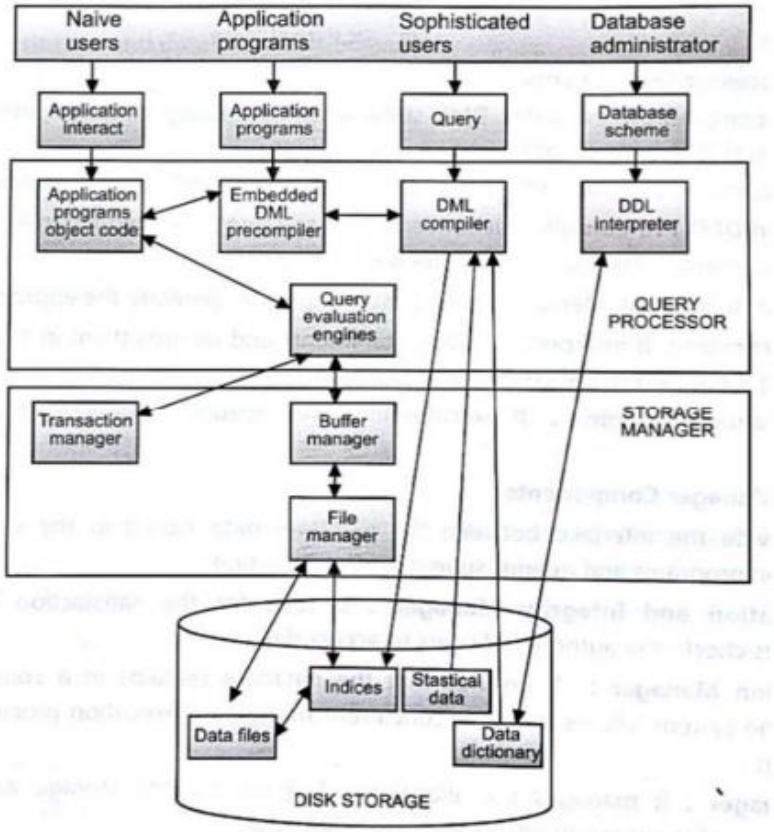


Figure 1.3

A relational database was produced.

User

FIELD	**TYPE**	**LENGTH**	**PRIMARYKEY**
Id	Int	11	Yes
UserName	varchar	100	No
Password	varchar	100	No

Table 2.1

Booking

FIELD	**TYPE**	**LENGTH**	**PRIMARYKEY**
Destination	varchar	100	No
Check in date	date		No
Check out	Date		No

Table 2.2

Tours

FIELD	TYPE	LENGTH	PRIMARYKEY
id	int	100	Yes
name	varhar	128	No
duration	varchar	128	No
date	Date		No
dateEnd	Date		No
cost	int		No
amount	int		No
bookingdate	Date		No
discount	int		No
valid	varchar	128	No
transport	varchar	128	No

Table 2.3

Chapter 5: IMPLEMENTATION AND EVALUATION

5.1Implementation Environment

The Tourism Guide App was developed using the following software and hardware:

Software:

Integrated Development Environment (IDE) - a software application that provides developers with a comprehensive set of tools for developing and testing software such a visual studio code.

Node.js- JavaScript runtime environment that allows developers to build server-side applications.

JavaScript- a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else

MySQL- a tool used to manage databases and servers

Hardware:

Development Environment - high-performance computers with adequate processing power, memory, and storage to support the development and testing of the app.

Testing Environment - mobile devices and other hardware needed for testing the app on various platforms and configurations.

Server and Hosting Environment - servers and hosting infrastructure needed to support the app and its deployment to a production environment.

PROGRAMMING LANGUAGE

- Nodejs
- JavaScript
- SQL

5.2Documentation:

DOCUMENTATION
Documentation will entail weekly dairy sheets for the work done and the user manual for the Tourism Guide System.

USER MANUAL
Tourism Guide System comes with the user manual. The user manual will help users on how the system works and perform all the functionalities that it provides. Refer to appendix B for the full user manual.

5.3System Testing:

Test Plan for Tourism Guide Web App
Introduction

This section describes the test plan for the Tourism Guide Web App. The purpose of the testing is to ensure that the web app is functional, user-friendly, and meets the requirements outlined in the specifications.
Test Approach

The testing approach for this project will be black box testing. This approach will be used to test the functionality of the web app from the perspective of an end user, without knowledge of the internal workings of the system.

Test Plan Summary

The following test cases will be executed:

- Test Case 1: User registration
- Test Case 2: User login
- Test Case 3: Search for tourist attractions
- Test Case 4: Display tourist attraction details
- Test Case 5: Add a tourist attraction
- Test Case 6: Edit a tourist attraction
- Test Case 7: Delete a tourist attraction
- Test Case 8: Logout

4. Test Data

The following test data will be used:

- Test data for user registration: username, email, password, confirm password
- Test data for user login: username, password
- Test data for searching tourist attractions: keyword(s)
- Test data for displaying tourist attraction details: tourist attraction ID
- Test data for adding a tourist attraction: name, description, address, latitude, longitude, image
- Test data for editing a tourist attraction: tourist attraction ID, name, description, address, latitude, longitude, image
- Test data for deleting a tourist attraction: tourist attraction ID

5. Test Results

The following test results were obtained:
Test Case 1: User registration

- Result: Passed

Test Case 2: User login

- Result: Passed

Test Case 3: Search for tourist attractions

- Result: Passed

Test Case 4: Display tourist attraction details

- Result: Passed

Test Case 5: Add a tourist attraction

- Result: Passed

Test Case 6: Edit a tourist attraction

- Result: Failed

Test Case 7: Delete a tourist attraction

- Result: Passed

Test Case 8: Logout

- Result: Passed

6. Conclusion

All test cases but the editing of tourist attractions were executed successfully and passed. The Tourism Guide Web App is functional and meets most requirements outlined in the specifications

5.4System Evaluation:

System evaluation is a critical aspect of software development. It involves assessing the effectiveness and efficiency of the system, including its design, functionality, and user experience. The evaluation is typically carried out after the development phase is completed and before the system is deployed to users.
Through the period I was given to complete this project, I have managed to archive objectives

Listed below;

- Login in user
- Register user
- Edit user profiles

- Add destination
- Delete destination
- Search destination
- Book destination
- Logout of session
- Responsive view

However there we a lot of functionality which I did not complete such as:

- Validation of dates when inserting a destination
- A proper package system were travel agents shows case there companies and different packages they have
- Editing the destinations
- Adding a payment gateway
- Making the dashboard personalized to the user
- Adding Reviews and Rating

The project objectives have been met to some extent as travelers can view information about the travel destination however it could use more work in adding a custom dashboard, having more functionality like writing a review, adding your own pictures to the web app and better ui design.

Future work:

I will be going on to implement the following in the future:

- Improving the user interface for consistency
- Implement a review page
- Adding a payment gateway for booking
- Editing destinations page
- Adding a page for users to insert images

CONCLUSION

In conclusion, we haven't successfully designed and implemented a tourism guide web application that meets the specified requirements. The application allows users to search for tourist destinations, view their user details, and insert new destination. The interface is user-friendly and responsive

During the implementation, we used HTML, CSS, JavaScript, Nodejs, and MySQL to create the application. The testing approach used was black box testing, which was effective in identifying bugs and errors in the system.

Based on our evaluation, we can conclude that the system meets some of the functional and non-functional requirements specified in the system specifications. The system is responsive, secure, and reliable.

As for further work, the application could be enhanced by adding more features such as user reviews and ratings, weather forecasts, and a booking system for hotels and transportation. Additionally, the system could be optimized for faster performance and made more accessible for users with disabilities.

Overall, this project has been a valuable experience and has taught me a lot including new frameworks and time scheduling.

REFERENCES

- Aaker, D. A. (1996). Building strong brands. Simon and Schuster.
- Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. ACM SIGMOD Record, 22(2), 207-216..
- Argyris, C., & Schön, D. A. (1996). Organizational learning II: Theory, method, and practice. Addison-Wesley Publishing Company.
- Buhalis, D. (1998). Strategic use of information technologies in the tourism industry. Tourism management, 19(5), 409-421.
- Buhalis, D., & Law, R. (2008). Progress in information technology and tourism management: 20 years on and 10 years after the Internet—The state of eTourism research. Tourism Management, 29(4), 609-623.
- Chang, J. H., & Chen, S. W. (2008). The impact of online reviews on hotel booking intentions and perception of trust. Tourism management, 29(4), 622-640.
- Fodness, D. (1994). Measuring tourist motivation. Annals of Tourism Research, 21(3), 555-581.
- Gartner, W. C. (1986). Image formation process. Journal of Travel & Tourism Marketing, 2(2-3), 191-215.
- Gretzel, U. (2011). Intelligent tourism: Technology-enabled experiences. In Information and Communication Technologies in Tourism 2011 (pp. 3-15). Springer.

1. S. S. Kim, S. Lee, and K. W. Lee, "Mobile Application for Smart Tourism: Enhancing Tourists' Experiences," Journal of Travel & Tourism Marketing, vol. 31, no. 8, pp. 1016-1031, 2014.
2. C. Sigala, "Smart tourism ecosystems: Definition, structure, and dynamics," Journal of Destination Marketing & Management, vol. 5, no. 3, pp. 177-180, 2016.
3. J. S. Chen, H. Huang, and T. J. Petrick, "Tourism Experience Design with Smart Technology," Journal of Travel Research, vol. 54, no. 4, pp. 433-445, 2015.
4. N. H. Kim and D. K. Lee, "Smart Tourism Information System for Small and Medium-sized Enterprises," Journal of Hospitality and Tourism Technology, vol. 11, no. 3, pp. 333-348, 2020.

5. M. A. Zaidi, S. A. R. Abu Bakar, and N. H. Zulkifli, "Mobile Tourism Guide Application with Augmented Reality Features," Journal of Telecommunication, Electronic and Computer Engineering, vol. 9, no. 3-9, pp. 71-76, 2017.
6. R. A. Barile and S. M. Sthapit, "Smart tourism and cultural heritage in the age of mobile devices," Journal of Tourism Futures, vol. 3, no. 2, pp. 162-175, 2017.
7. L. M. Lee, C. M. Tan, and K. S. Tan, "Developing a Mobile Tourist Guide for Kuala Lumpur City," Procedia - Social and Behavioral Sciences, vol. 105, pp. 413-420, 2013.
8. S. M. Sthapit and R. A. Barile, "Smart Tourism and Destination Management with Mobile Applications," Information Technology & Tourism, vol. 19, no. 1, pp. 29-47, 2017.
9. J. Kim and M. Gretzel, "Smart Tourism Destinations: Enhancing Tourism Experience Through Personalisation of Mobile Services," Journal of Destination Marketing & Management, vol. 8, pp. 271-280, 2018.
10. A. G. Benitez, R. García, and A. Sánchez, "Mobile Application for Tourist Information and Support: A Case Study in Seville," Procedia Computer Science, vol. 130, pp. 415-422, 2018.

APPENDIX A : LOG-BOOK

Student Name & ID:	Thabiso Seleke 201902015
Supervisor Name:	Ms Taukobong
Week Ending:	17 March 2022

Student Self Evaluation:

Feedback on proposal

Report on Work done this week:

We looked at website to cite from

Assigning of tasks to be done by next meeting:

I was told to expand my objectives since they were shortly before the next meeting

Supervisor Evaluation: Supervisor's observation on progress: (e.g. general observation of progress and estimated progress measures based on previously assigned tasks)

Student's Signature:T N Seleke	**Date: 17 / 03/ 2021**
Supervisor's Signature:_____	**Date: _____**

Student Name & ID:	Thabiso Seleke 201902015
Supervisor Name:	Ms Taukobong
Week Ending:	22 April 2022

Student Self Evaluation:

Report on Work done this week:

Show a demo with php
Assigning of tasks to be done by next meeting:

I was told to implement using another language

Supervisor Evaluation: Supervisor's observation on progress: (e.g. general observation of progress and estimated progress measures based on previously assigned tasks)

Student's Signature: T N Seleke	**Date: 22/ 04/ 2021**
Supervisor's Signature:_____	**Date: _____**

Student Name & ID:	Thabiso Seleke 201902015
Supervisor Name:	Ms Taukobong
Week Ending:	28 April 2022

Student Self Evaluation:

Report on Work done this week:

Implemented project using node js
Assigning of tasks to be done by next meeting:

Fixing my ui design

Supervisor Evaluation: Supervisor's observation on progress: (e.g. general observation of progress and estimated progress measures based on previously assigned tasks)

Student's Signature: T N Seleke	**Date: 28/ 04/ 2021**
Supervisor's Signature:_____	**Date: _____**

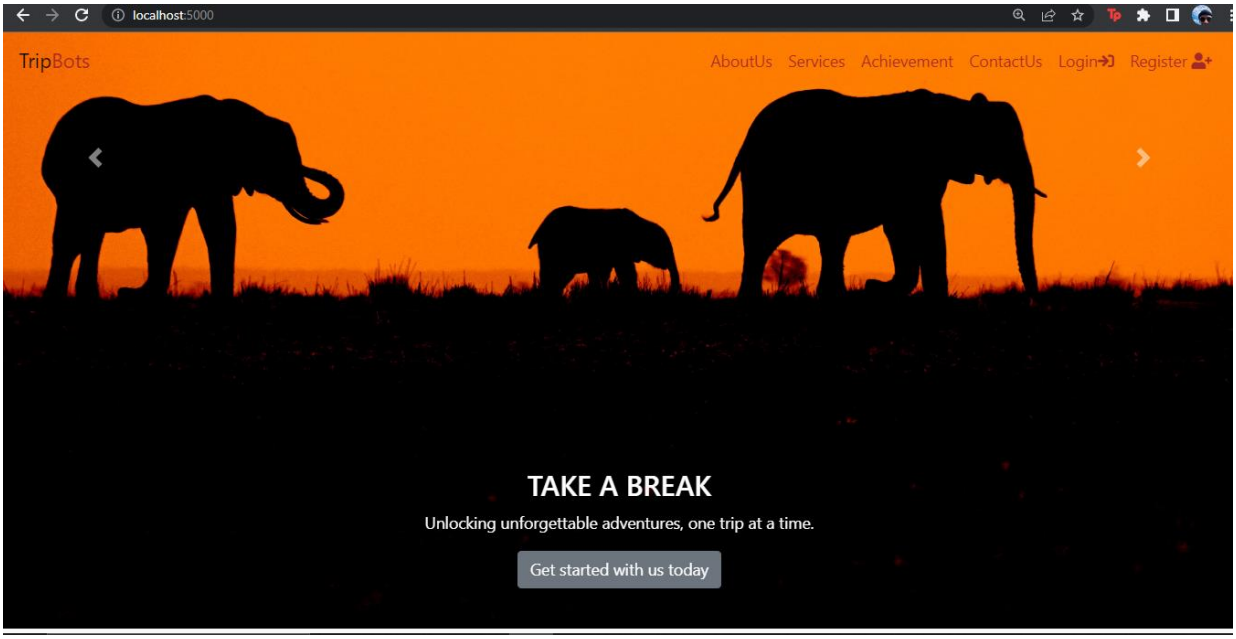
APPENDIX B: USER MANUAL

I**ntroduction**

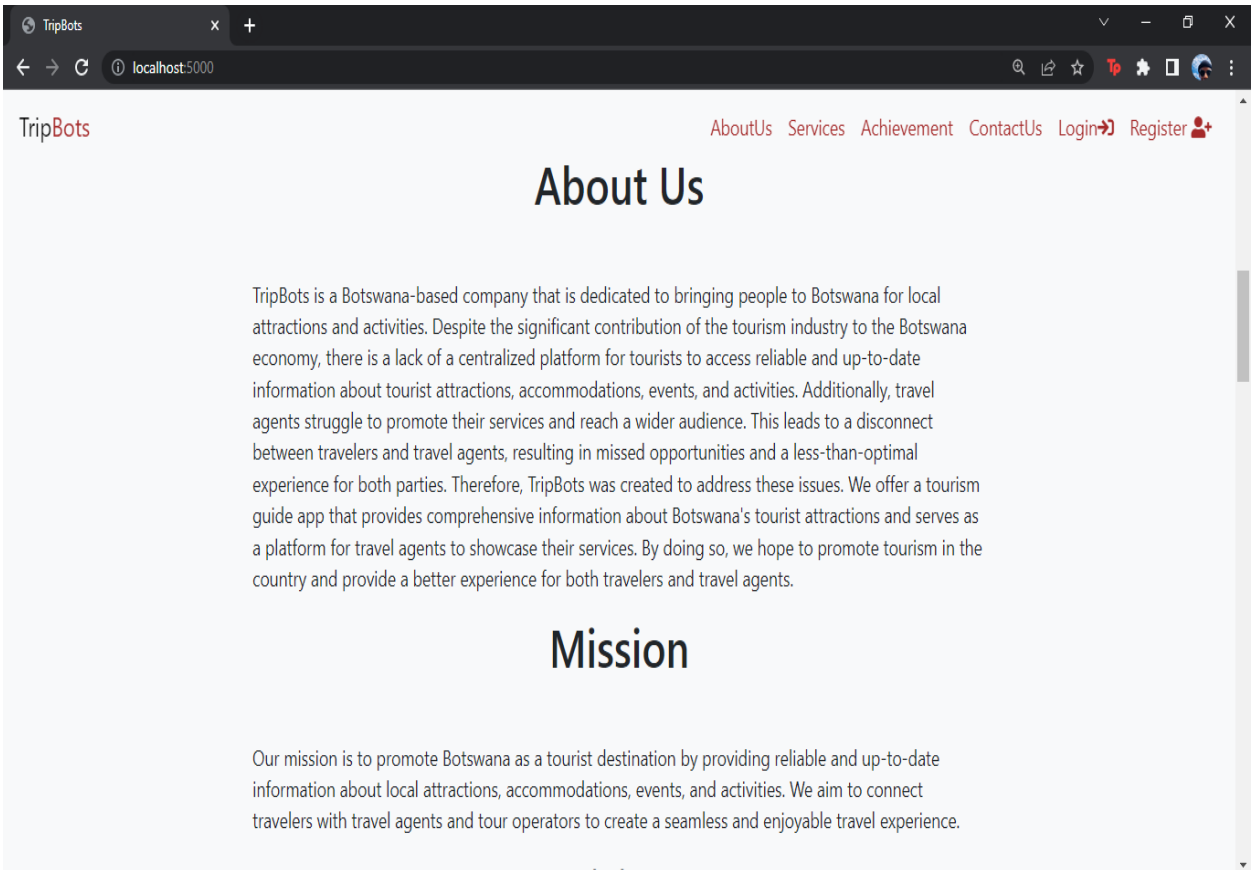
Tourism Guide system is intended to enable users to book a package for trips

User manual for Traveler and Travel Agent

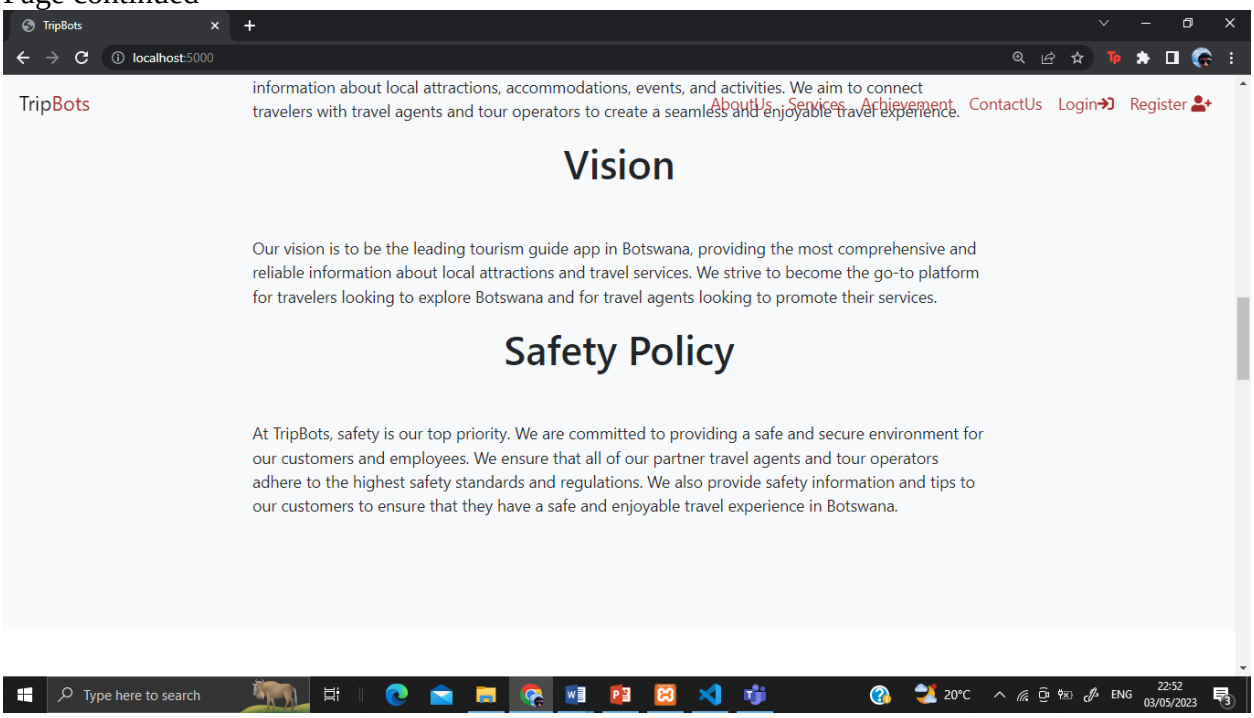
Welcome/Home page for Traveler and Travel Agent



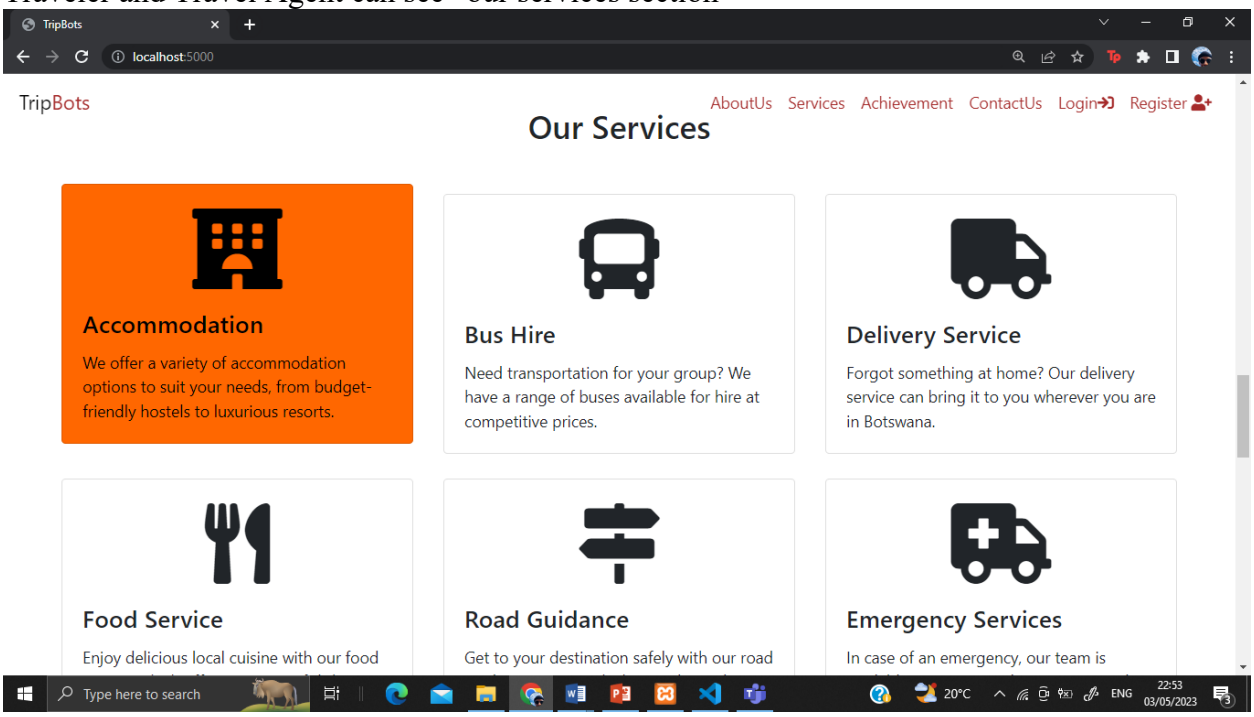
Traveler and Travel Agent is able to scroll down and see our about, mission an vision section



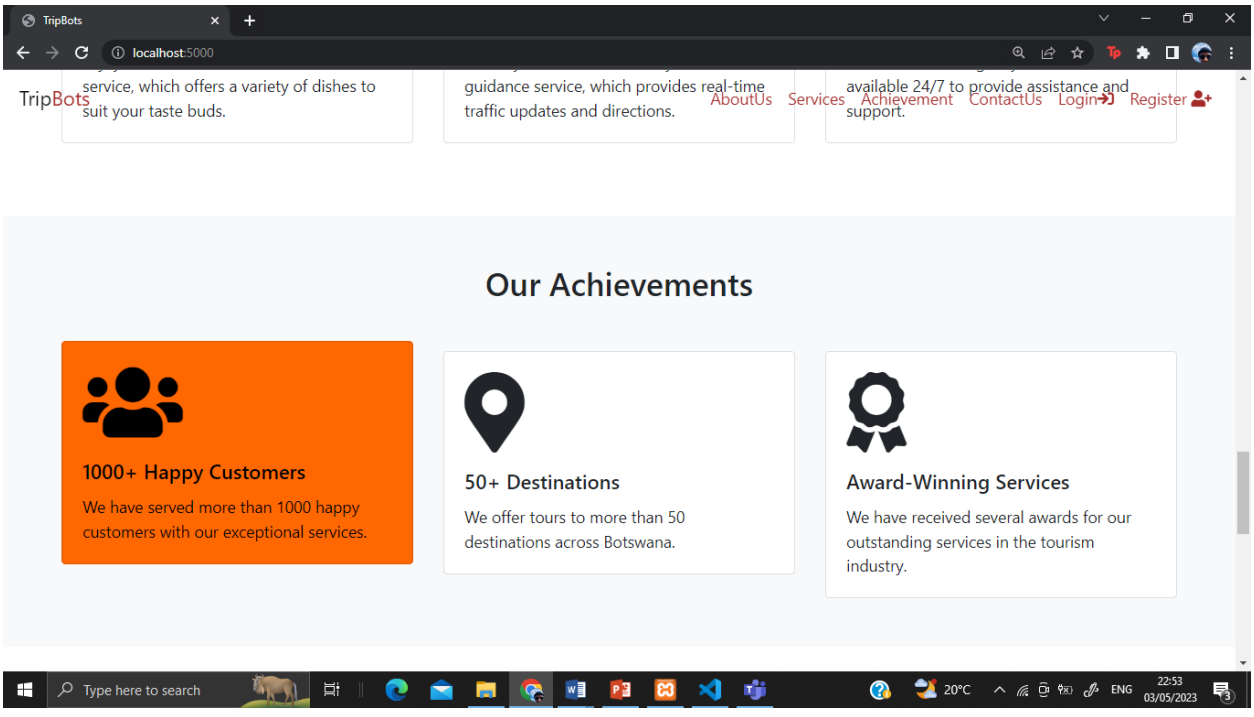
Page continued



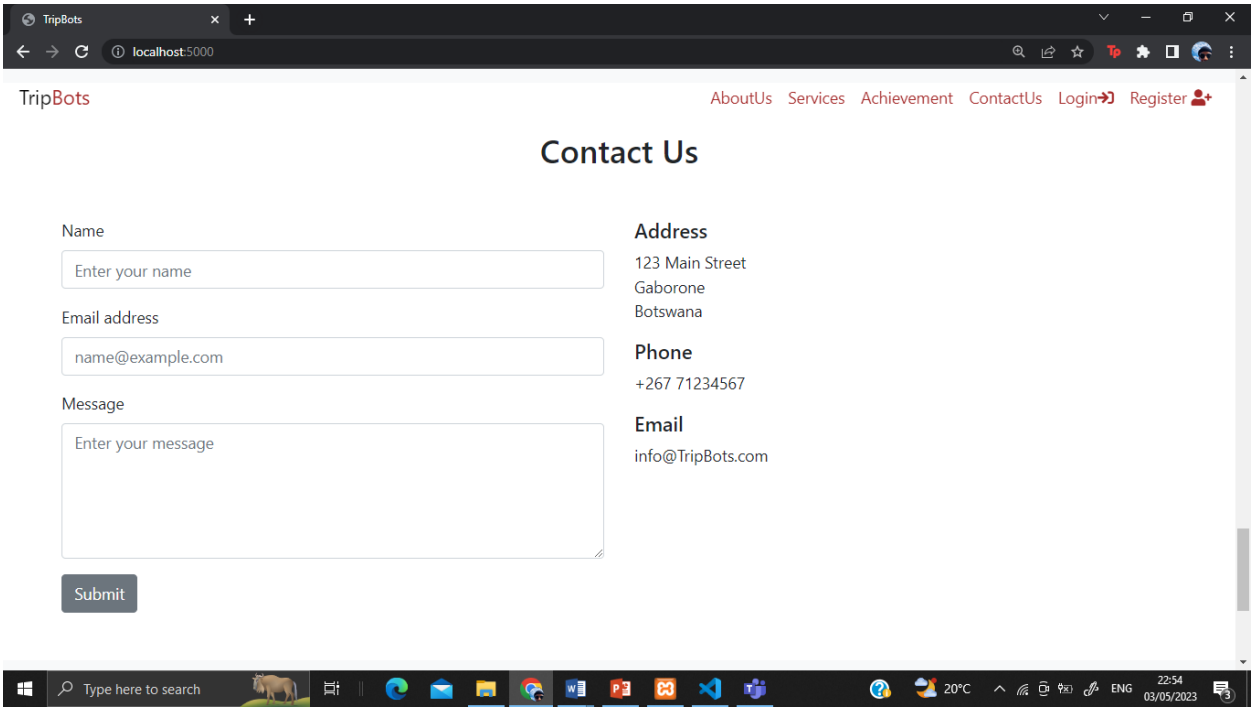
Traveler and Travel Agent can see “our services section



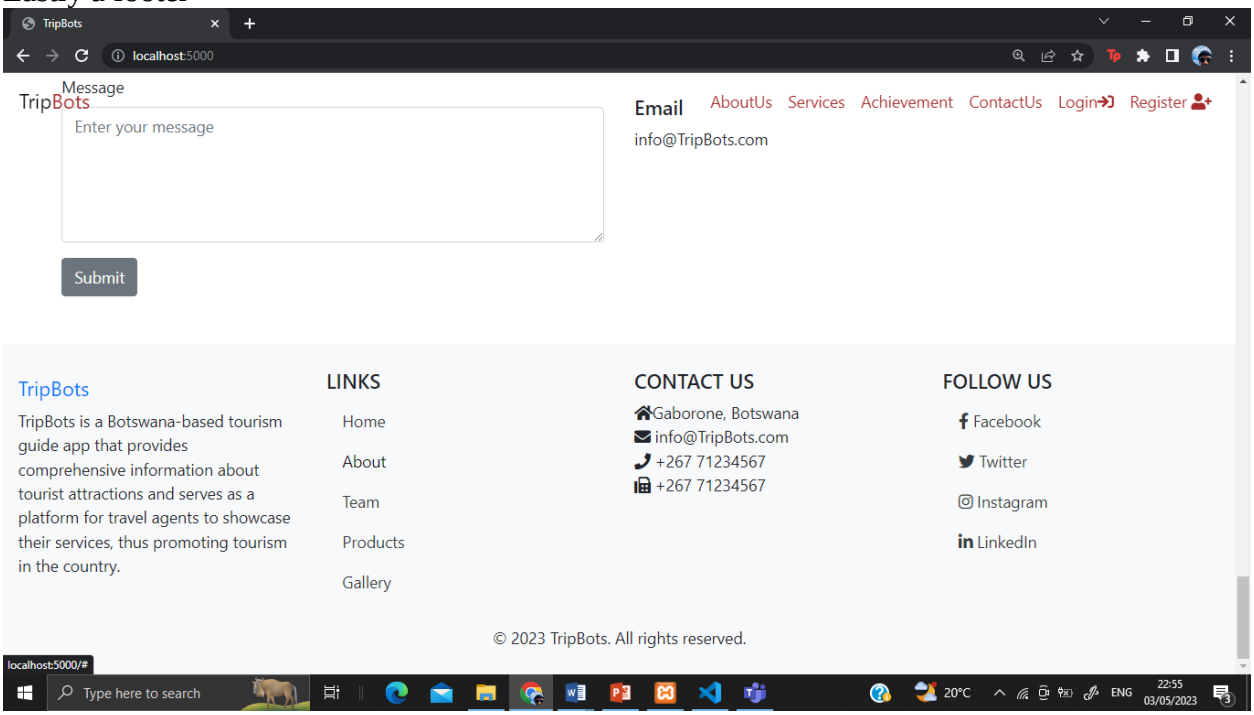
Next an achievements section



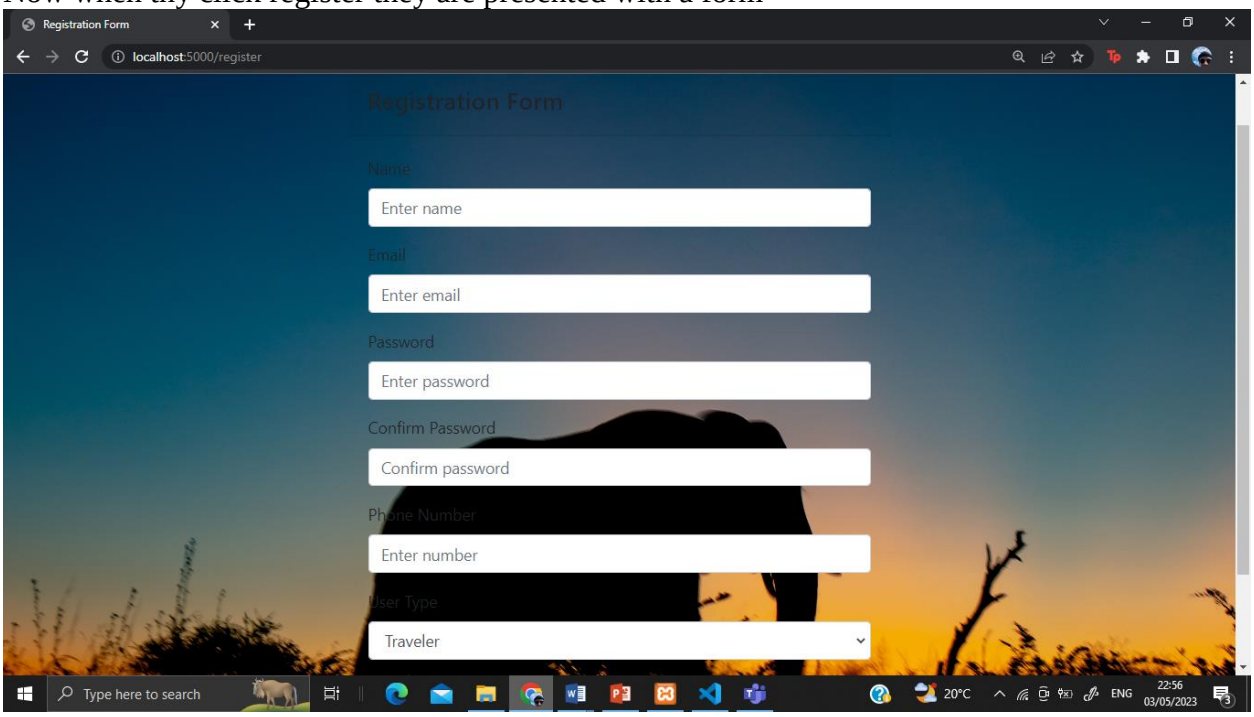
Next a contact us section



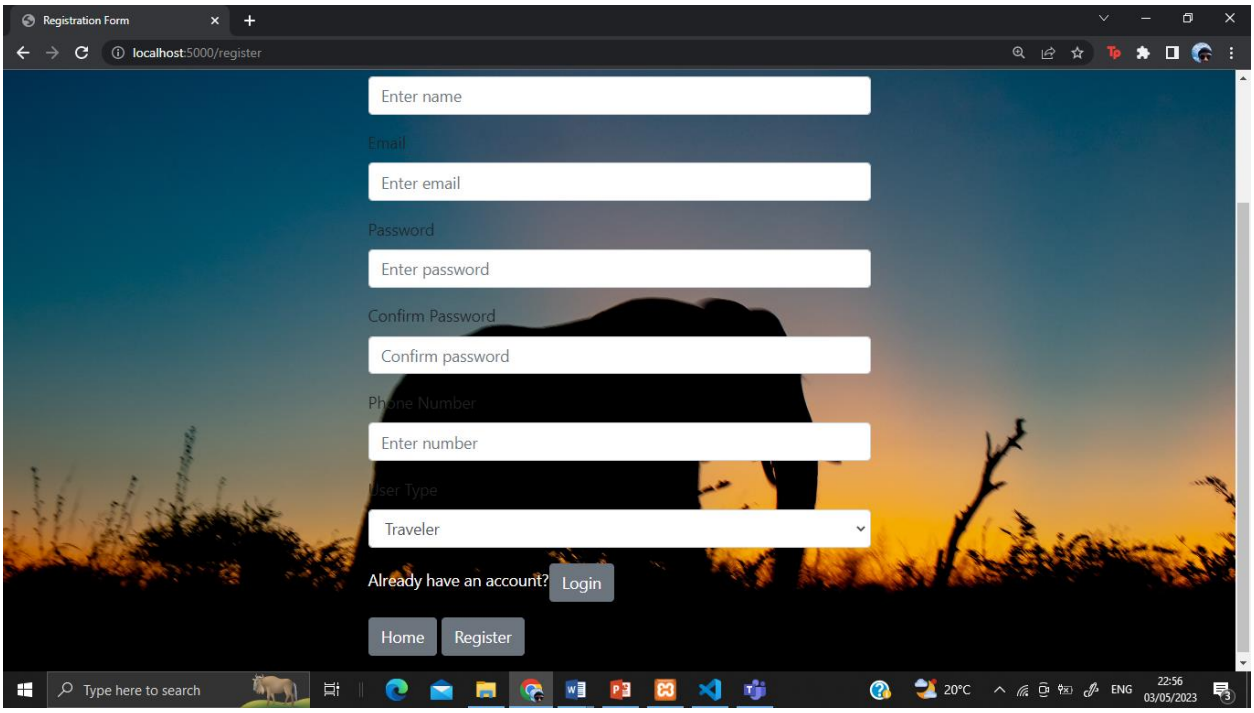
Lastly a footer



Now when thy click register they are presented with a form

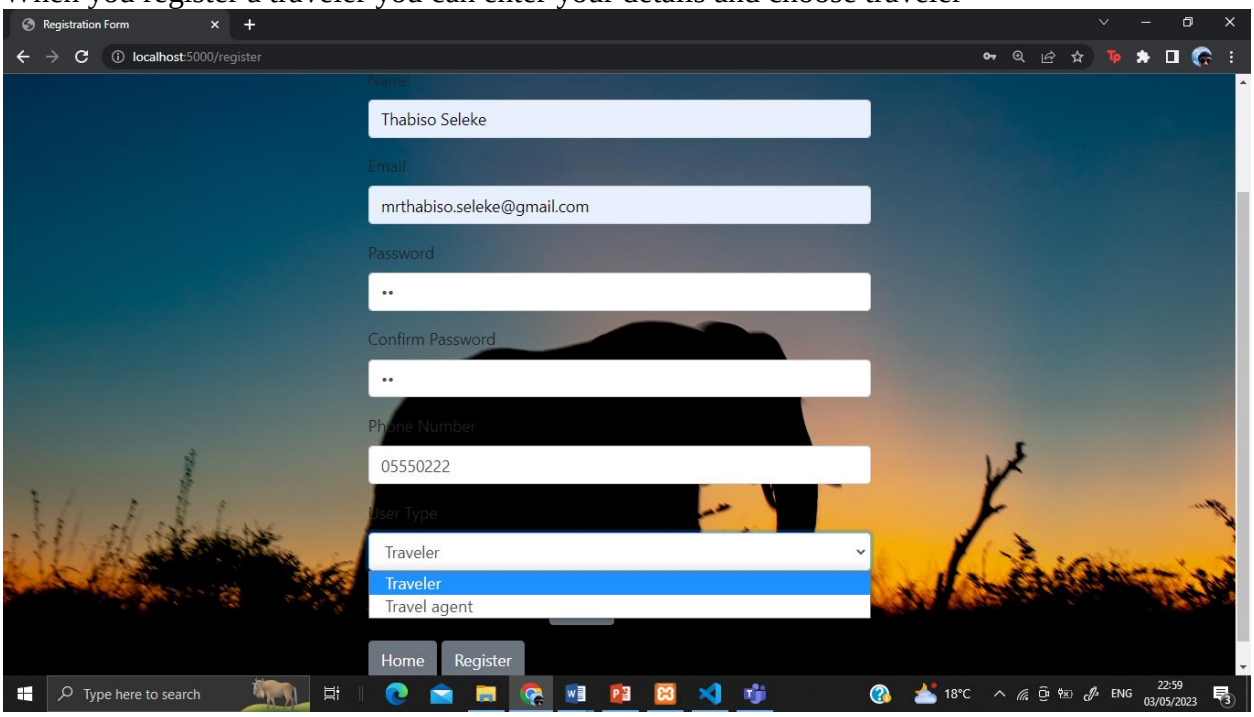


Form continued

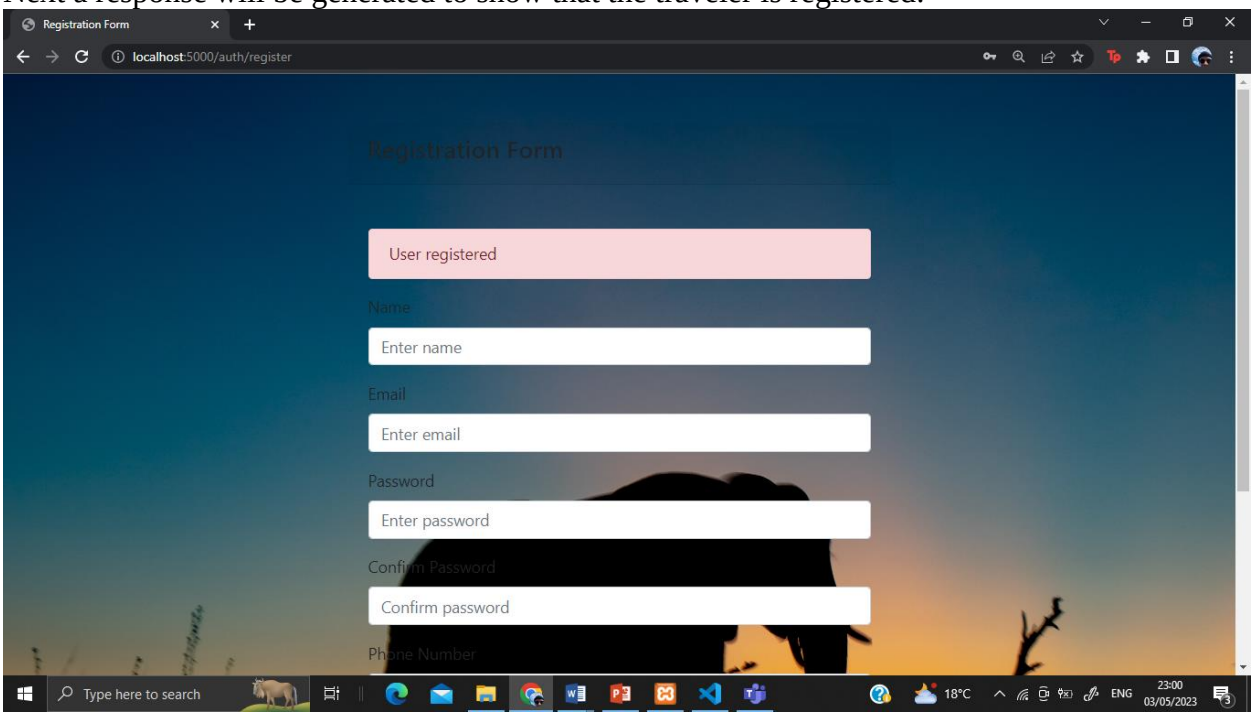


Now I will focus on the user being a traveler:

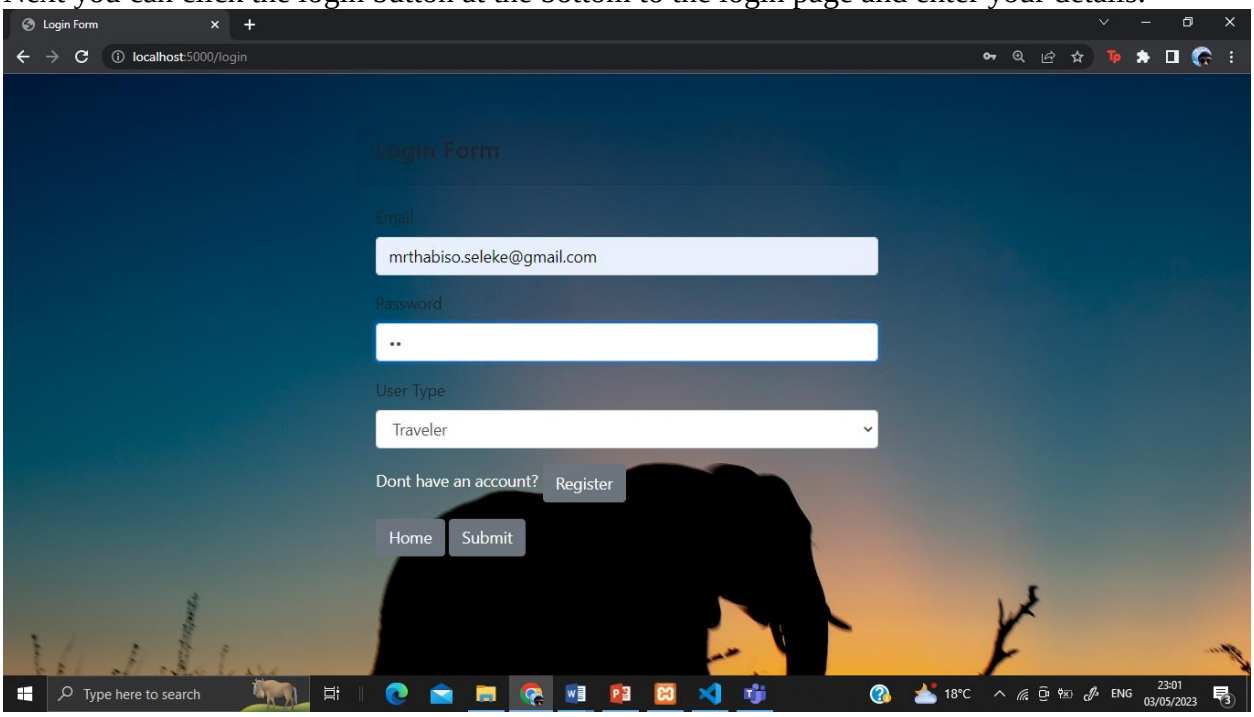
When you register a traveler you can enter your details and choose traveler



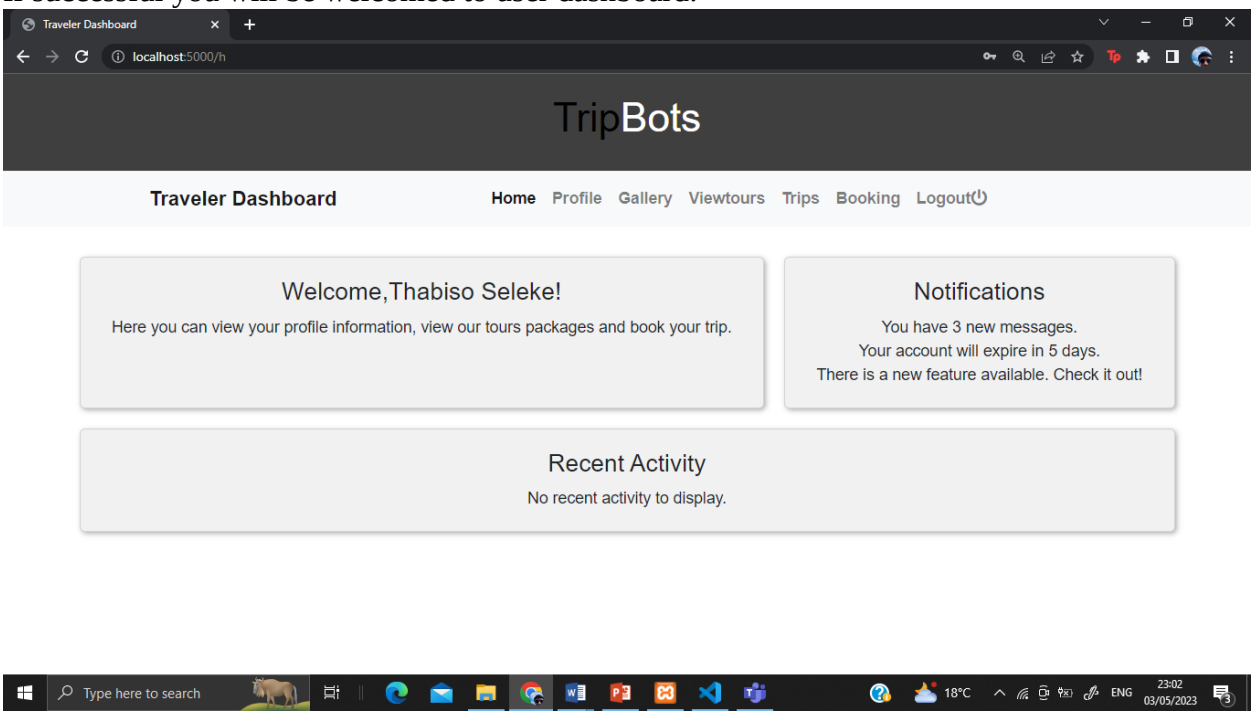
Next a response will be generated to show that the traveler is registered:



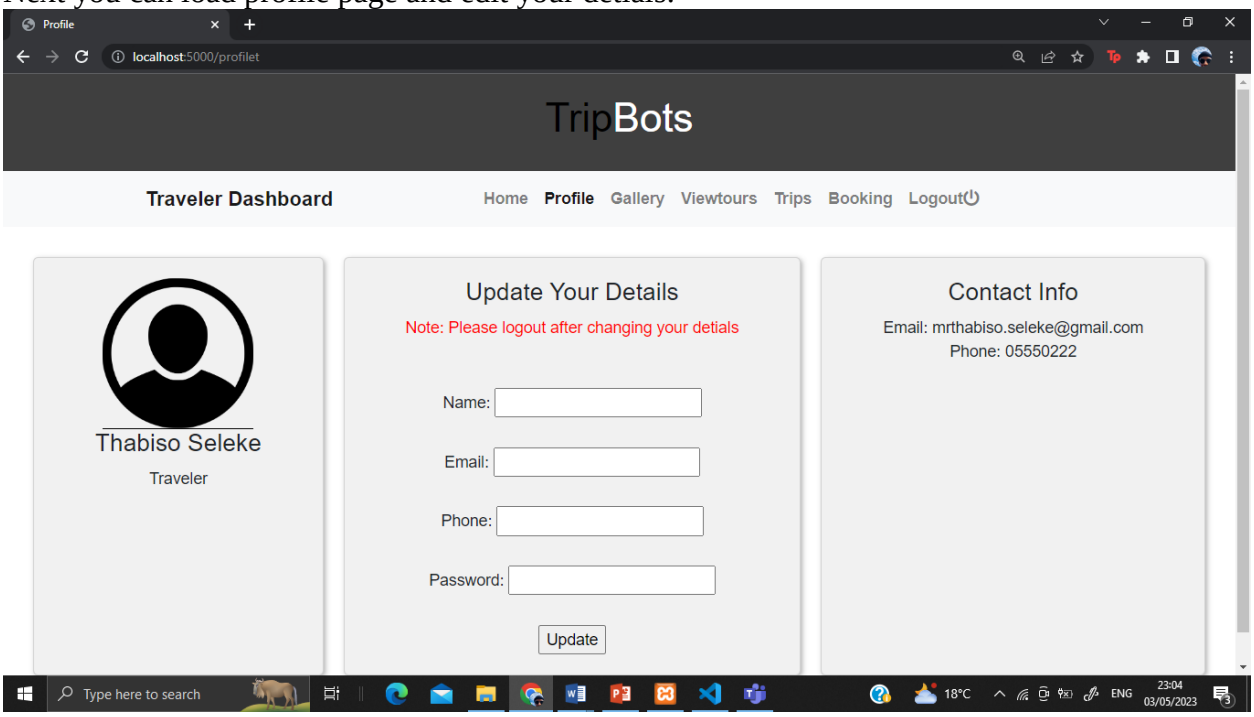
Next you can click the login button at the bottom to the login page and enter your details:



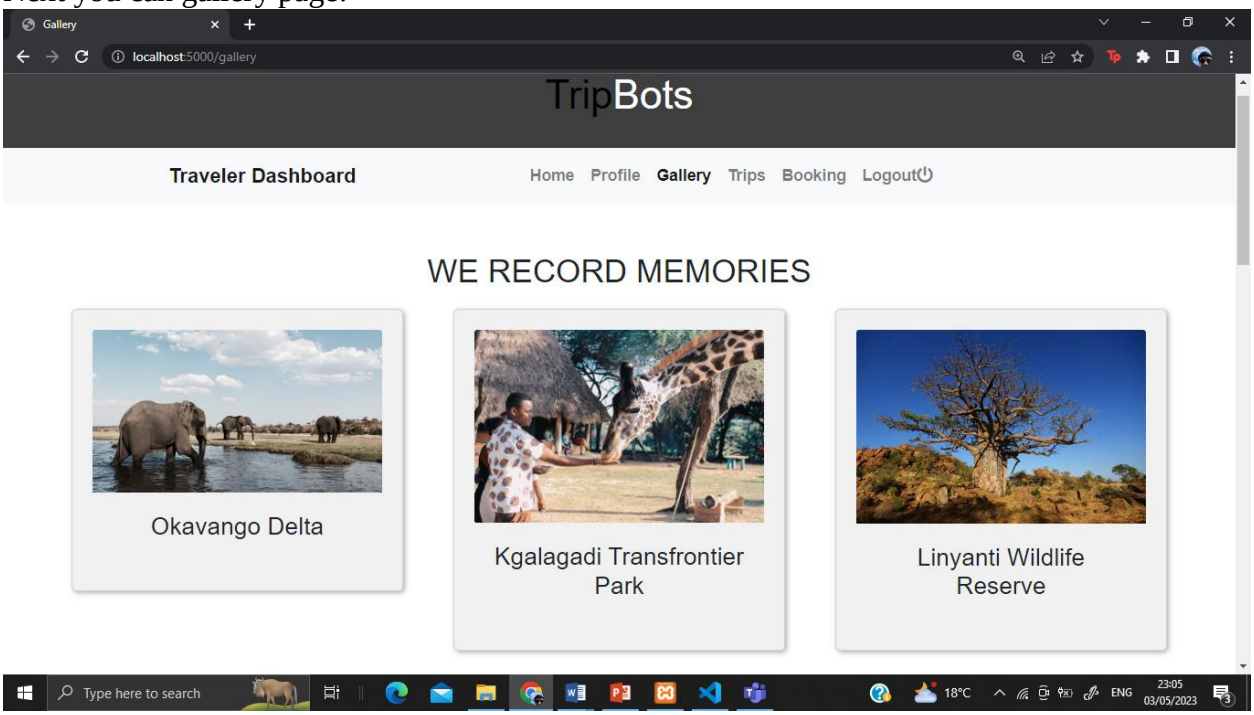
If successful you will be welcomed to user dashboard:



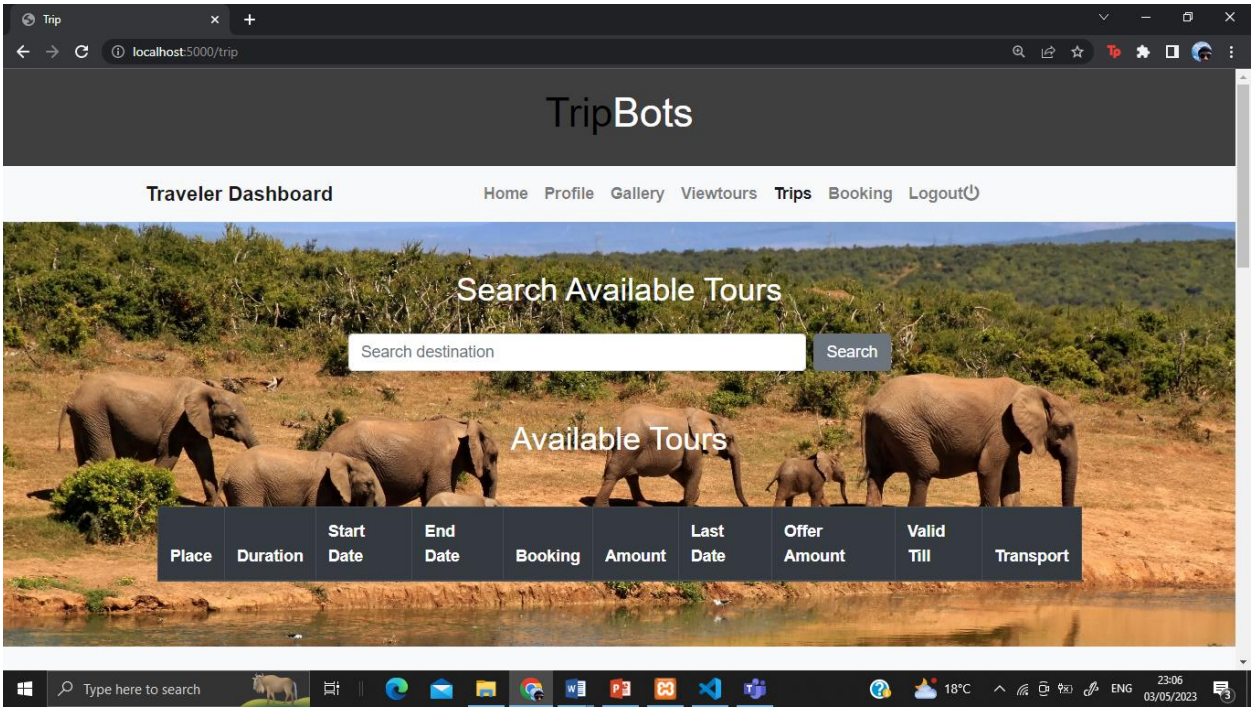
Next you can load profile page and edit your detials:



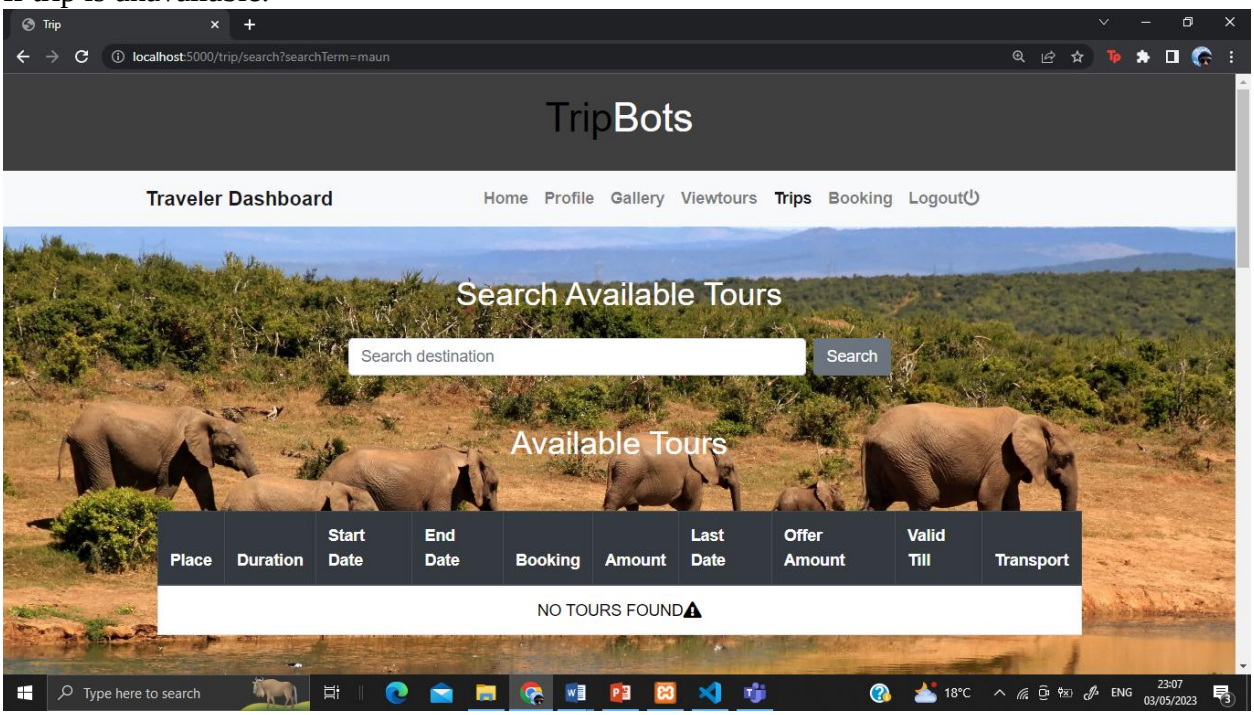
Next you can gallery page:



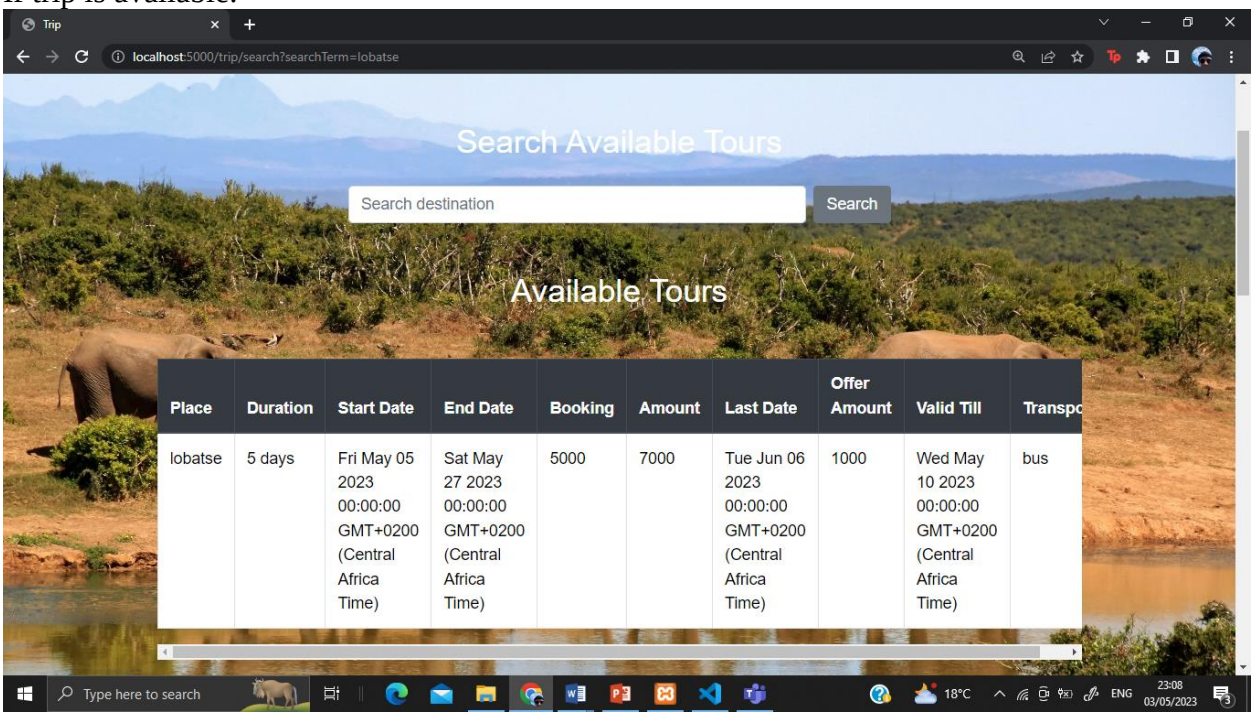
Next you can go to the search and search page:



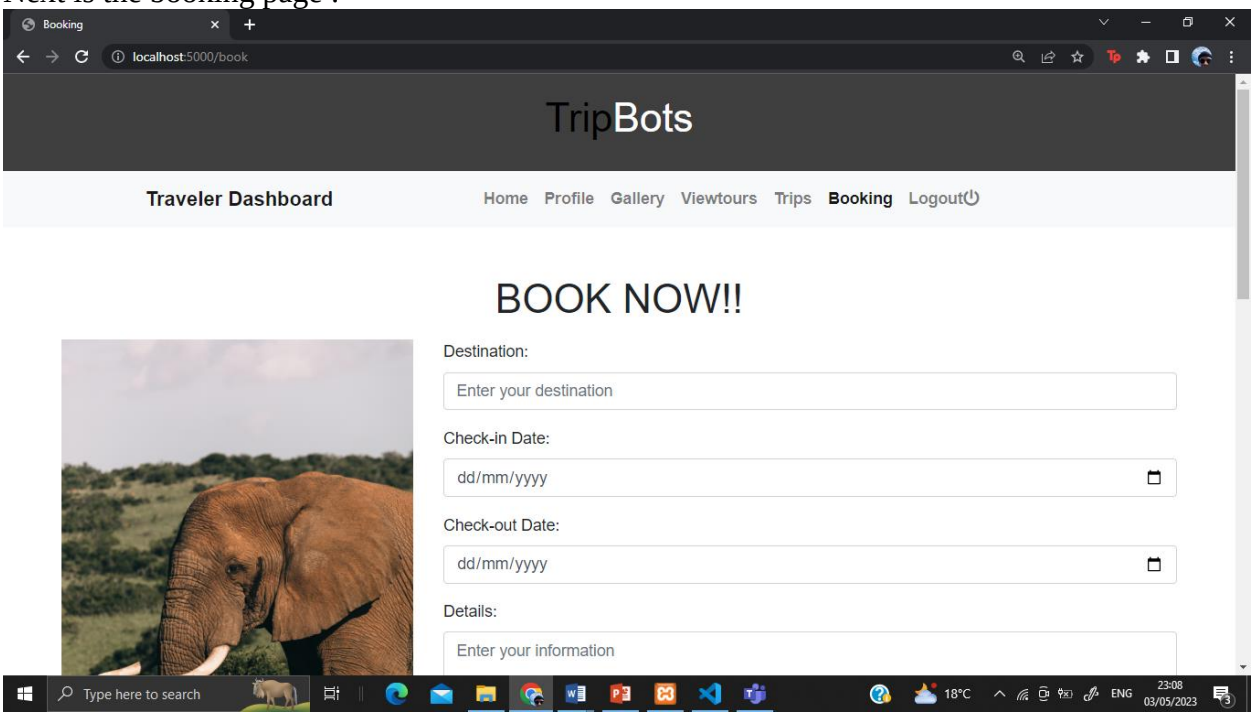
If trip is unavailable:



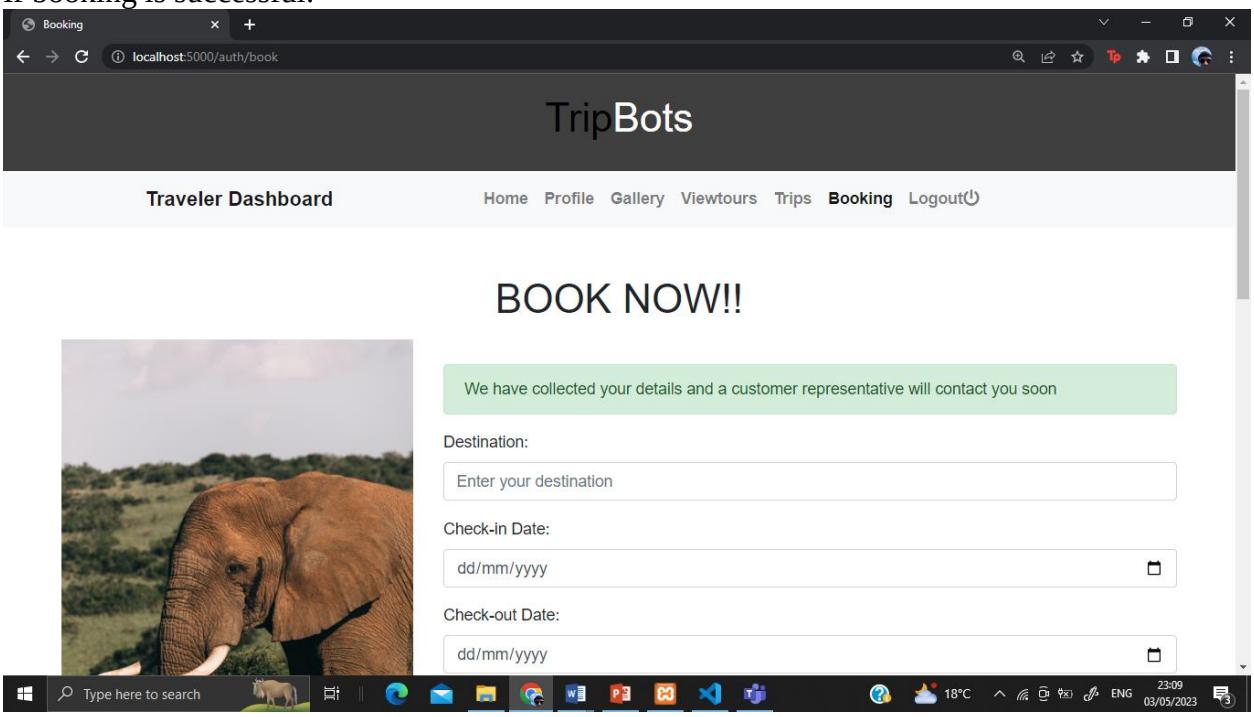
If trip is available:



Next is the booking page :

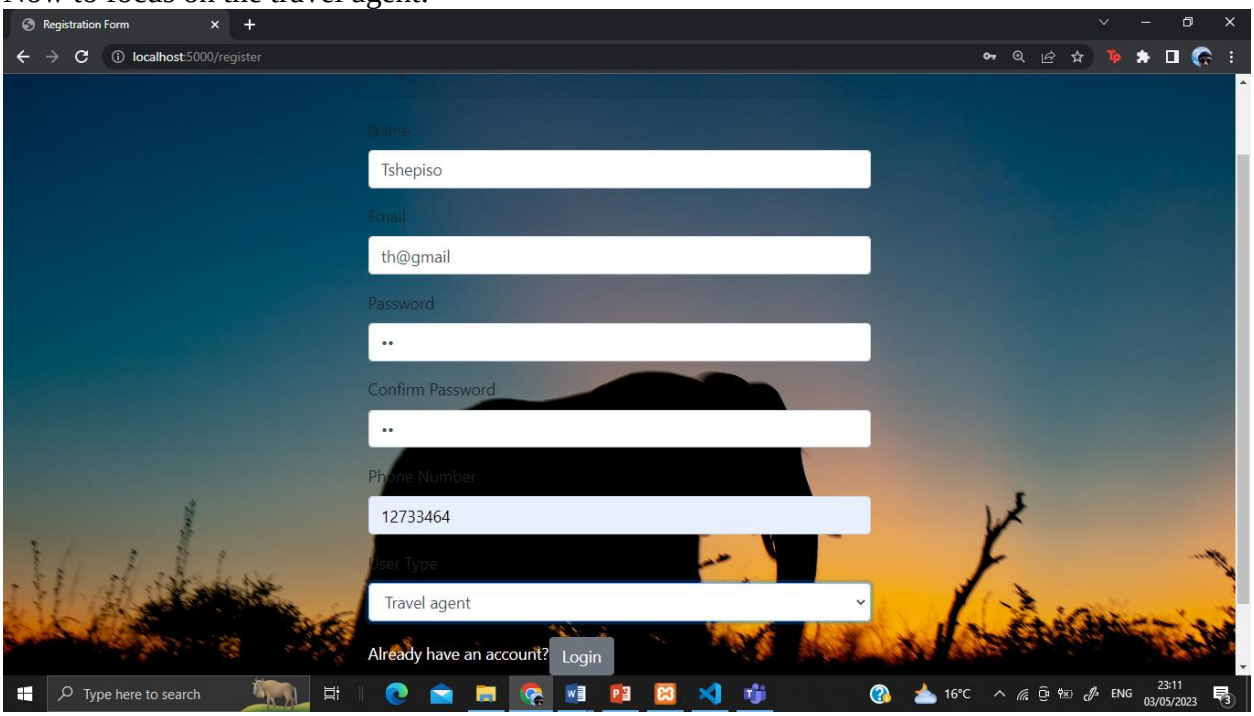


If booking is successful:

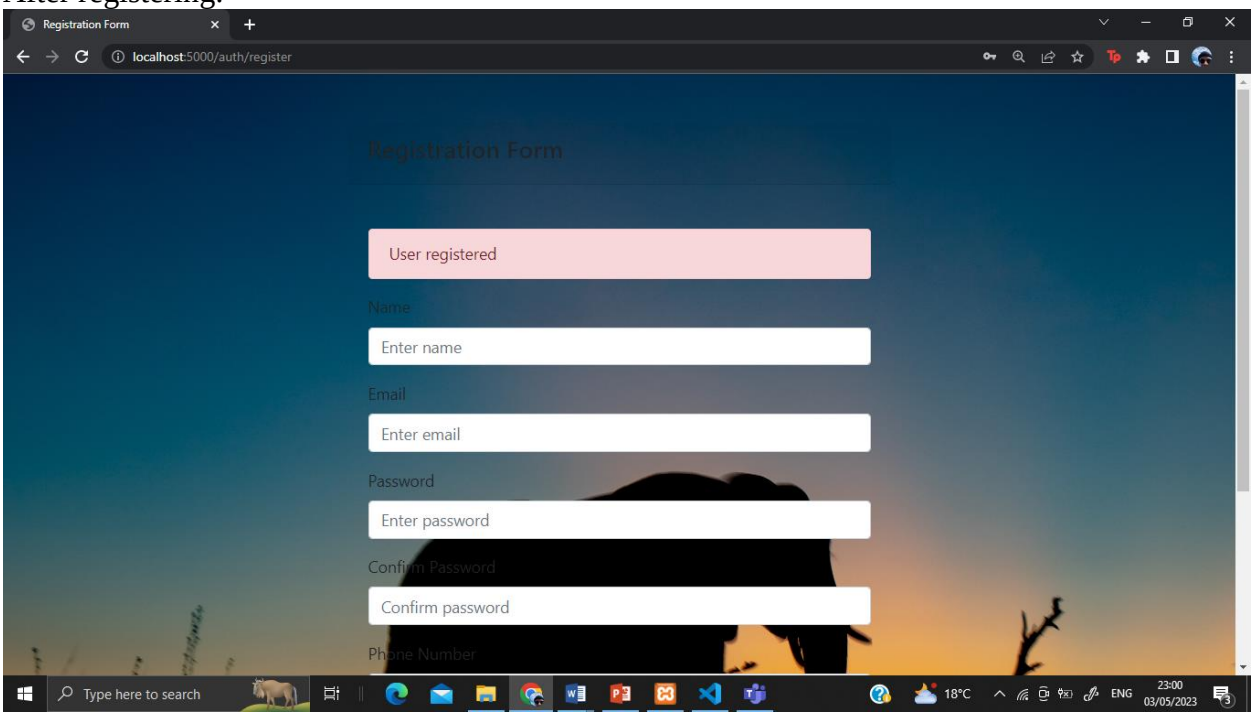


Final the logout takes you to the home page

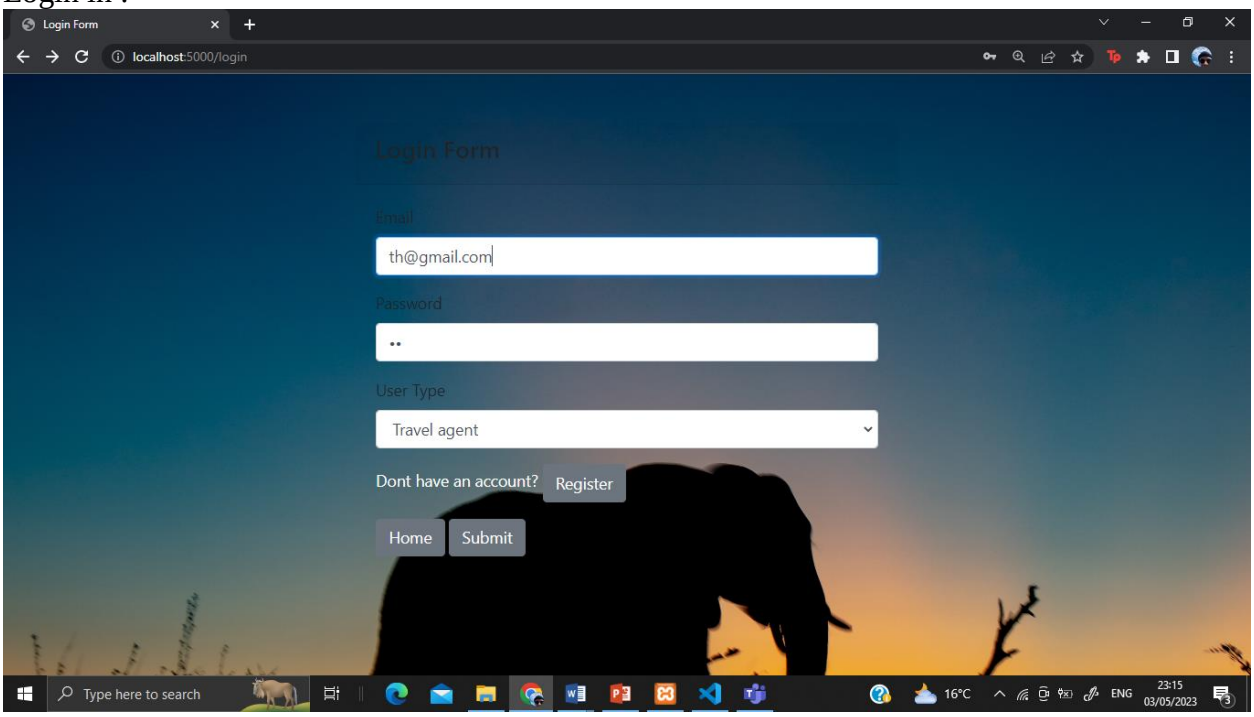
Now to focus on the travel agent:



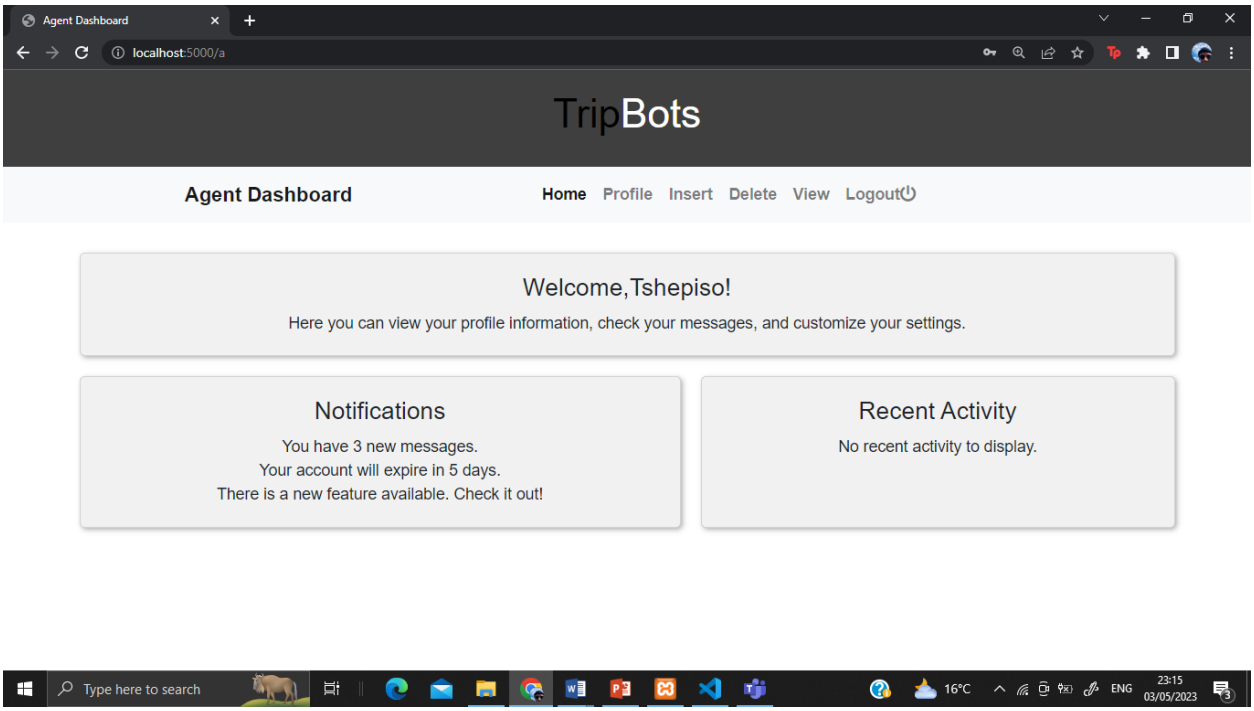
After registering:



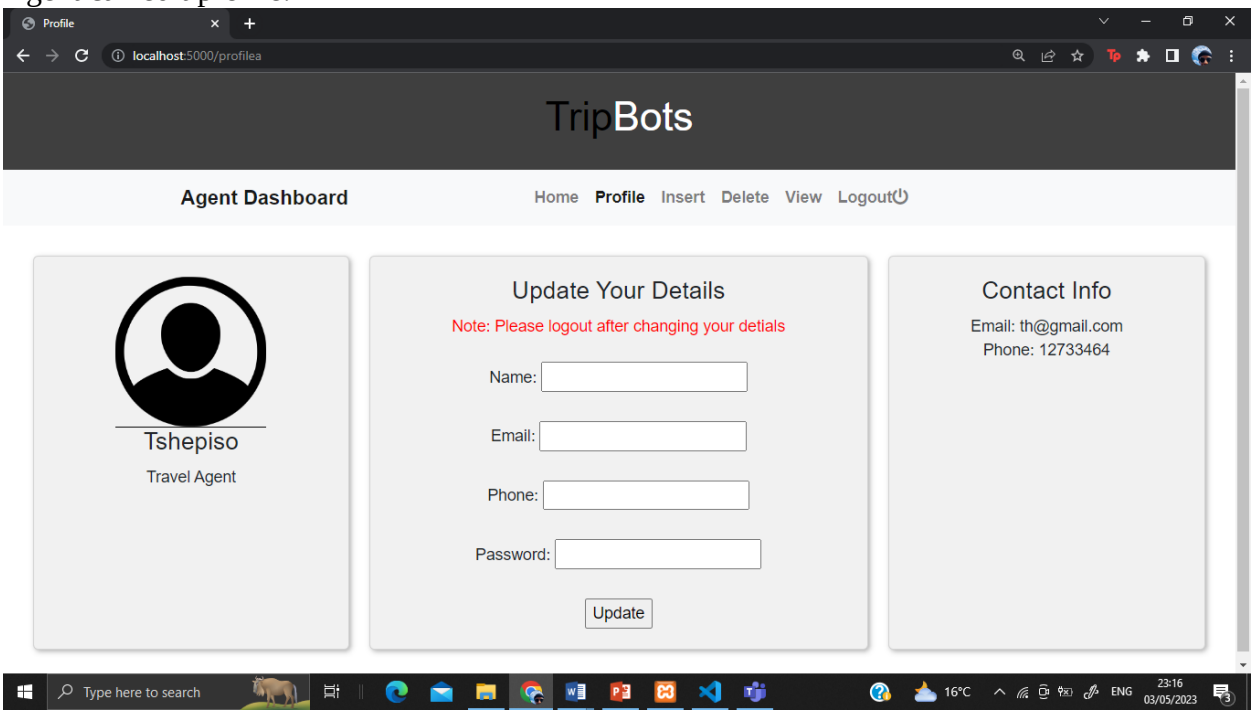
Login in :



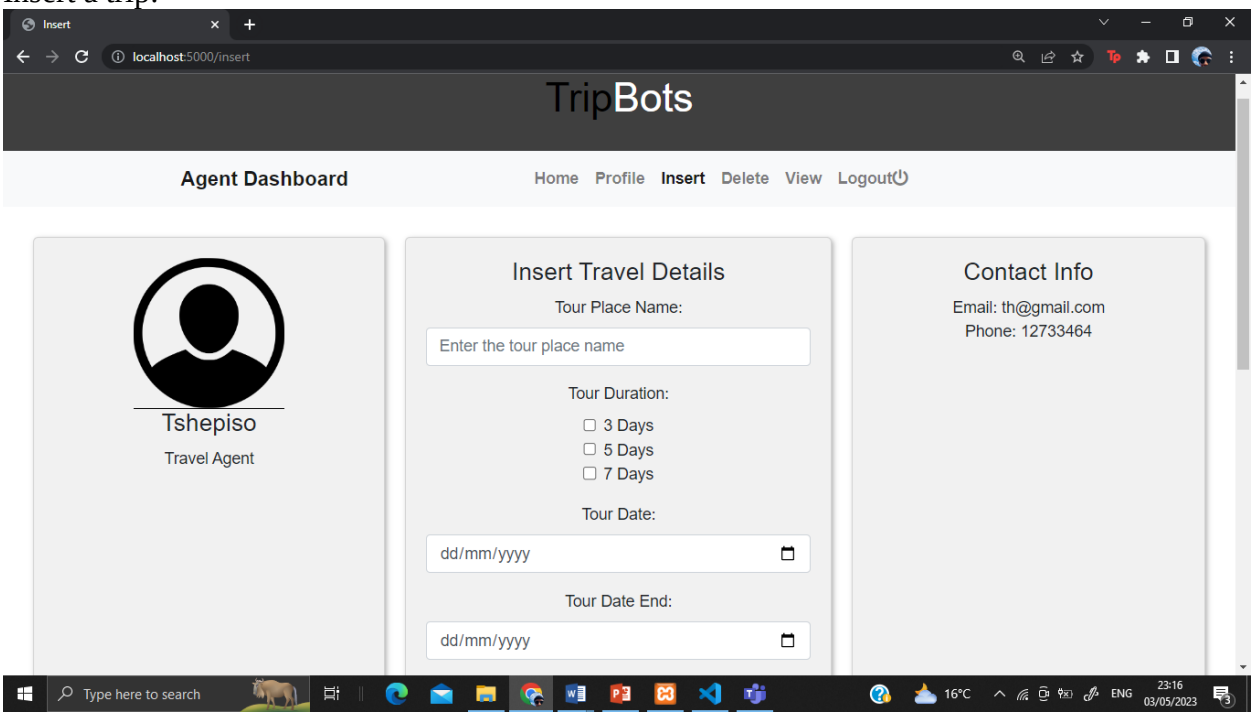
Agent dashboard:



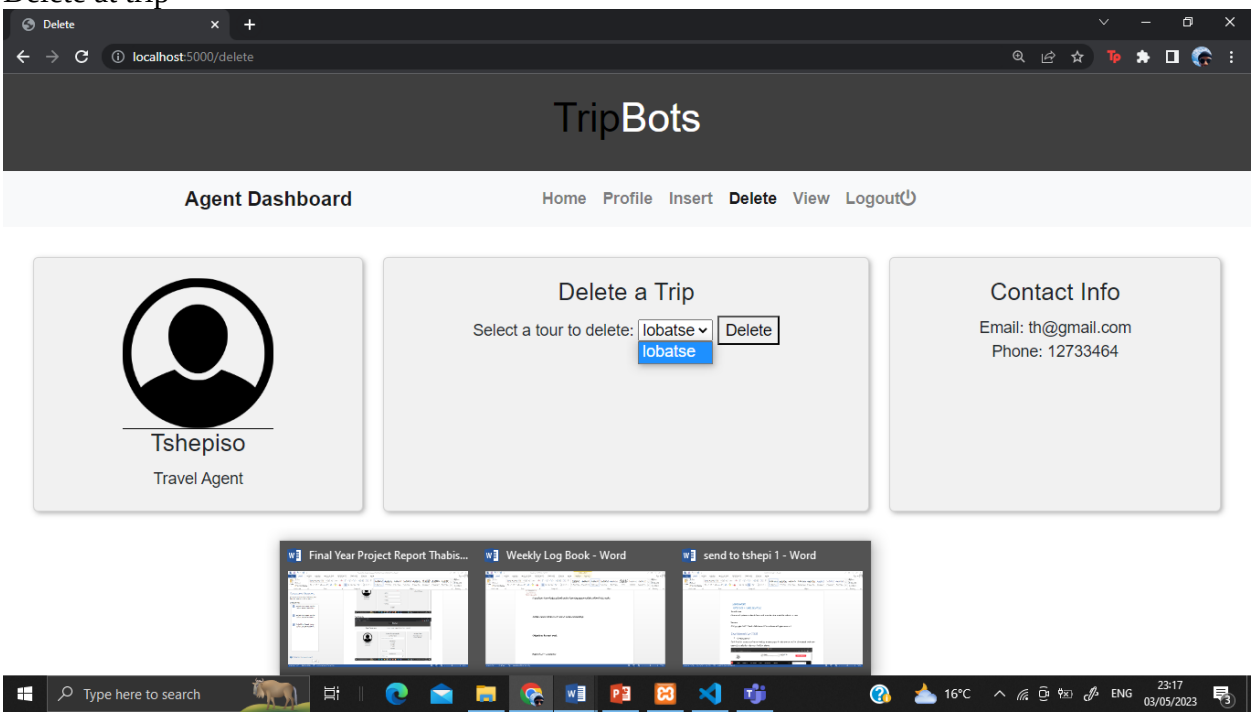
Agent can edit profile:



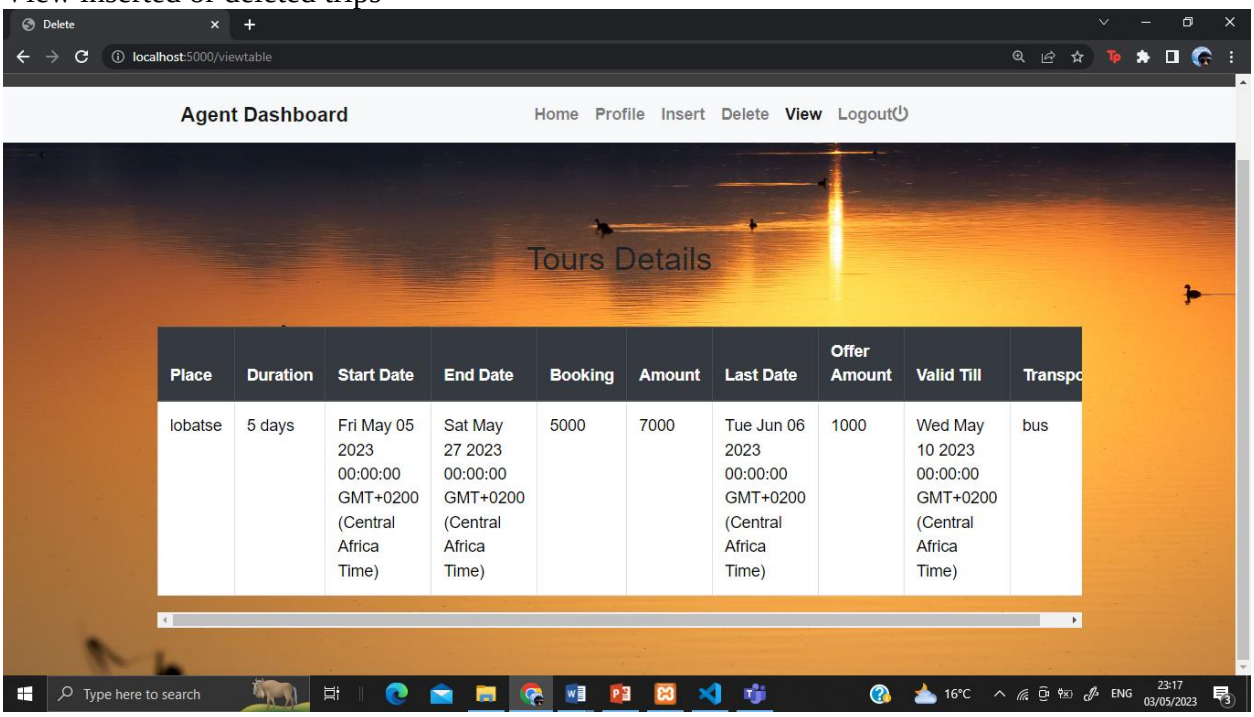
Insert a trip:



Delete at trip



View inserted or deleted trips



Finally the logout takes you to the home page

APPENDIX C: SOURCE CODE

Controllers auth.js:

```
const mysql = require("mysql");
const jwt = require("jsonwebtoken");
const bcrypt = require("bcryptjs");
const {promisify} = require("util");
const { error } = require("console");

const db= mysql.createConnection({
    host: process.env.DATABASE_HOST,
    user:process.env.DATABASE_USER,
    password:process.env.DATABASE_PASSWORD,
    database:process.env.DATABASE,
});

exports.login = async (req, res) => {
    try {
```

```
      const { email, password, role } = req.body;
      if (!email || !password || !role) {
        return res.status(400).render('login', {
          message: 'Please provide an email, password, and role'
        });
      }
      db.query('SELECT * FROM users WHERE email = ?', [email], async (error,
results) => {
        console.log(results);
        if (
          !results ||
          !(await bcrypt.compare(password, results[0].password)) ||
          results[0].role !== role
        ) {
          res.status(401).render("login", {
            message: "Email, password, or role is incorrect",
          });
        } else {
          const id = results[0].id;
          const token = jwt.sign({ id }, process.env.JWT_SECRET, {
            expiresIn: process.env.JWT_EXPIRES_IN
          });

          console.log("The token is: " + token);

          const cookieOptions = {
            expires: new Date(
              Date.now() + process.env.JWT_COOKIE_EXPIRES * 24 * 60 * 60 * 1000
            ),
            httpOnly: true
          }
          res.cookie('jwt', token, cookieOptions);
          console.log(req.cookies);
        // Redirect based on role
        if (role === "Travel agent") {
          res.status(200).redirect("/a");
        } else if (role === "Traveler") {
          res.status(200).redirect("/h");
        } else {
          // If the role is neither "Travel agent" nor "Traveler"
          res.status(401).render("login", {
            message: "Role is incorrect",
          });
        }
      }
```

```
    })
  } catch (error) {
      console.log(error);
    }
  }

exports.register = async (req, res) => {
    console.log(req.body);

    const { name, email, password, confirm_password, number, role } = req.body;

    db.query('SELECT email FROM users WHERE email = ?', [email], async (error,
results) => {
        if (error) {
            console.log(error);
        }

        if (results.length > 0) {
            return res.render('register', {
                message: 'That email is already in use'
            });
        } else if (password !== confirm_password) {
            return res.render('register', {
                message: 'Passwords do not match'
            });
        }

        let hashedPassword = await bcrypt.hash(password, 8);
        console.log(hashedPassword);

        db.query('INSERT INTO users SET ?', {
            name: name,
            email: email,
            password: hashedPassword,
            number: number,
            role: role
        }, (error, results) => {
            if (error) {
                console.log(error);
            } else {
                console.log(results);
                return res.render('register', {
                    message: 'User registered'
                });
            }
```

```
        });
    });
};

exports.isLoggedIn = async (req, res, next) => {
  let decoded = null;
  if (req.cookies && req.cookies.jwt) { // check if req.cookies is defined and
has the jwt property
    console.log("JWT token present in cookie");
    try {
      decoded = await promisify(jwt.verify)(req.cookies.jwt,
process.env.JWT_SECRET);
      console.log("JWT token verified");
      db.query('SELECT * FROM users WHERE id = ?', [decoded.id], (error, result)
=> {
        if (error) {
          console.log("Error while querying the database for user information:",
error);
          return next(error);
        }
        if (!result || result.length === 0) { // check if a user is found for the
decoded JWT token
          console.log("No user found for the decoded JWT token:", decoded);
          return next(new Error('User not found')); // return an error if no user
is found
        }
        console.log("User information retrieved from the database:", result[0]);
        req.user = result[0];
        return next();
      });
    } catch (error) {
      console.log("Error while verifying the JWT token:", error);
      return next();
    }
  } else {
    console.log("JWT token not present in cookie");
    next();
  }
};

exports.updateUserDetails = async (req, res) => {
  try {
    const userId = req.body.userId;

    const { name, email, number, password } = req.body;
```

```
    // Check that all required fields are provided
    if (!name || !email || !number || !password) {
      return res.status(400).render('profilet', {
        message: 'Please provide all requirements fields'
      });
    }

    // Hash the password
    const hashedPassword = await bcrypt.hash(password, 10);

    // Update the user details in the database
    const updateUserQuery = 'UPDATE users SET name=?, email=?, number=?,
password=? WHERE id=?';
    db.query(updateUserQuery, [name, email, number, hashedPassword, userId],
(error, results) => {
      if (error) {
        console.log(error);
        return res.status(500).render('profilet', {
          message: 'Something went wrong'
        });
      } else {
        return res.status(200).render('profilet', {
          message: 'Successfully updated'
        });
      }
    });
  } catch (error) {
    console.log(error);
  }
};

exports.updateUserDetailsA = async (req, res) => {
  try {
    const userId = req.body.userId;

    const { name, email, number, password } = req.body;
    // Check that all required fields are provided
    if (!name || !email || !number || !password) {
      return res.status(400).render('profilea', {
        message: 'Please provide all requirements fields'
      });
    }
    // Hash the password
    const hashedPassword = await bcrypt.hash(password, 10);
    // Update the user details in the database
```

```
    const updateUserQuery = 'UPDATE users SET name=?, email=?, number=?,
password=? WHERE id=?';
    db.query(updateUserQuery, [name, email, number, hashedPassword, userId],
(error, results) => {
      if (error) {
        console.log(error);
        return res.status(500).render('profilea', {
          message: 'Something went wrong'
        });
      } else {
        return res.status(200).render('profilea', {
          message: 'Successfully updated'
        });
      }
    });
  } catch (error) {
    console.log(error);
  }
};

exports.input = async (req, res) => {
  // Get the form data from the request body
  const { tourPlace, tourDuration, tourDate, tourDateEnd, tourCost, amount,
lastDate, offerAmount, validTill, transport } = req.body;
  // Prepare the SQL query to insert the form data into the tours table
  const insertQuery = `INSERT INTO tours (name, duration, date, dateEnd, cost,
amount, bookingdate, discount, valid, transport) VALUES (?, ?, ?, ?, ?, ?, ?, ?,
?, ?)`;
  // Execute the SQL query with the form data
  db.query(
    insertQuery,
    [tourPlace, tourDuration, tourDate, tourDateEnd, tourCost, amount, lastDate,
offerAmount, validTill, transport],
    (error, result) => {
      if (error) {
        console.log(error);
        return res.status(500).render('insert', {
          message: 'Something went wrong'
        });
      } else {
        console.log('Form data inserted into MySQL database:', result);
        return res.status(200).render('insert', {
          message: 'Successfully inserted'
        });
      }
```

```
    });
};

exports.viewTable = (req, res) => {
  const sql = `SELECT * FROM tours`;
  db.query(sql, (err, result) => {
    if (err) throw err;
    res.render('viewtable', {
      user: req.user,
      tours: result
    });
  });
};

exports.viewTablet = (req, res) => {
  const sql = `SELECT * FROM tours`;
  db.query(sql, (err, result) => {
    if (err) throw err;
    res.render('viewtablet', {
      user: req.user,
      tours: result
    });
  });
};

exports.deleteTour = (req, res) => {

  const sql = `SELECT * FROM tours`;
  db.query(sql, (err, result) => {
    if (err) throw err;
    res.render('delete', {
      user: req.user,
      tours: result
    });
  });

  const tourName = req.body.tourName;

  const deleteQuery = "DELETE FROM tours WHERE name = ?";

  db.query(deleteQuery, [tourName], (err, result) => {
    if (err) {
      console.log(err);
      return res.status(500).render('delete', {
        message: 'Something went wrong'
```

```
      });
    } else {
      console.log(`Tour ${tourName} deleted from database`);
      return res.status(200).render('delete', {
        message: 'Successfully deleted'
      });
    }
  });
};

exports.searchTours = (req, res) => {
  let searchTerm = req.query.searchTerm;
  let sql = `SELECT * FROM tours`;

  if (searchTerm) {
    sql += ` WHERE name LIKE '%${searchTerm}%'`;
  }

  db.query(sql, (err, result) => {
    if (err) throw err;
    res.render('trip', {
      user: req.user,
      tours: result,
      searchTerm: searchTerm
    });
  });
};

exports.books = (req, res) => {
  const { destination, checkin, checkout, comments } = req.body;

  db.query('INSERT INTO booking SET ?', {
      destination: destination,
      checkin: checkin,
      checkout: checkout,
      comments: comments
  }, (error, results) => {
      if (error) {
          console.log(error);
          return res.render('book', {
              message: 'An error occurred while saving your details, please try
again later'
          });
      } else {
          console.log(results);
```

```
            return res.render('book', {
                message: 'We have collected your details and a customer
representative will contact you soon'
            });
        }
    });
};


exports.logout = async (req,res) =>{
    res.cookie('jwt', 'logout', {
        expires: new Date(Date.now() + 2 * 1000),
        httpOnly: true
    });
    res.status(200).redirect('/');
}
```

Book.hbs

```
<!DOCTYPE html>
<html>
<head>
    <title>Booking</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
```

```
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/h">Traveler Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="/h">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/profilet">Profile</a>
                </li>
                <li class="nav-item">
```

```
                    <a class="nav-link" href="/gallery">Gallery</a>
                </li>
                 <li class="nav-item">
                    <a class="nav-link" href="/viewtablet">Viewtours</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/trip">Trips</a>
                </li>
                <li class="nav-item active">
                    <a class="nav-link" href="/book">Booking</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
</ul>
</div>
</nav>
<h1 class="mt-5 mb-3 text-center">BOOK NOW!!</h1>
<div class="container">
    <div class="row">
        <!-- picture section -->
        <div class="col-md-4">
            <img src="/b.jpg" class="img-fluid" alt="placeholder">
        </div>
        <!-- form section -->
        <div class="col-md-8">
            {{#if message}}
            <p class="alert alert-success mt-4">{{ message }}</p>
            {{/if}}
            <form action="/auth/book" method="post">
                <div class="form-group">
                    <label for="destination">Destination:</label>
                    <input type="text" class="form-control" id="destination"
name="destination" placeholder="Enter your destination" required>
                </div>
                <div class="form-group">
                    <label for="checkin">Check-in Date:</label>
                    <input type="date" class="form-control" id="checkin"
name="checkin" required>
                </div>
                <div class="form-group">
                    <label for="checkout">Check-out Date:</label>
                    <input type="date" class="form-control" id="checkout"
name="checkout" required>
```

```
                </div>
                <div class="form-group">
                  <label for="comments">Details:</label>
                  <textarea class="form-control" id="comments" name="comments"
rows="3" placeholder="Enter your information"></textarea>
                </div>
                <button type="submit" class="btn btn-secondary">Book Now</button>
            </form>
        </div>
    </div>
</div>
  <!-- Term and Condition section -->
  <div class="py-5 bg-light" id="accessories">
    <div class="container my-5">
      <div class="row">
        <div class="col-md-8 mx-auto">
        <h1 class="text-center mb-5"><i class="fa fa-shield" aria-
hidden="true"></i></h1>
          <h1 class="text-center mb-5">Terms and Conditions</h1>
          <h2>Booking Policy</h2>
          <p>All bookings made through our website are subject to availability
and confirmation. A booking is considered confirmed only when you receive a
confirmation email from us. We reserve the right to cancel any booking if we are
unable to confirm availability.</p>
          <h2>Cancellation Policy</h2>
          <p>If you wish to cancel your booking, please notify us as soon as
possible. Our cancellation policy varies depending on the type of booking and the
time of cancellation. Please refer to your booking confirmation email for more
information. We reserve the right to charge a cancellation fee in some cases.</p>
          <h2>Refund Policy</h2>
          <p>If we cancel your booking for any reason, we will offer you a full
refund. If you cancel your booking within the allowed timeframe and are entitled
to a refund, we will process the refund as soon as possible. Please note that
refunds may take several business days to appear on your account depending on
your bank or credit card issuer.</p>
        </div>
      </div>
    </div>
    </div>


<!-- Add jQuery and Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></scri
pt>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Routes pages.js:

```
const express = require('express');
const authController = require('../controllers/auth');
const router = express.Router();


router.get('/',(req,res) => {
    res.render('index');
});

router.get('/a',authController.isLoggedIn,(req,res) => {
    console.log(req.user); // log the user object
    res.render('homeagent', {
        user: req.user
    });
});

router.get('/h',authController.isLoggedIn,(req,res) => {
    console.log(req.user); // log the user object
    res.render('hometraveler', {
        user: req.user
    });
});

router.get('/register',(req,res) => {
    res.render('register');
});

router.get('/login',(req,res) => {
    res.render('login');
});

router.get('/profilea', authController.isLoggedIn, (req,res) => {
```

```
    res.render('profilea', {
        user: req.user
    });
});

router.get('/profilet', authController.isLoggedIn, (req,res) => {
    res.render('profilet', {
        user: req.user
    });
});

router.get('/gallery', authController.isLoggedIn, (req,res) => {
    res.render('gallery', {
        user: req.user
    });
});

router.get('/delete', authController.isLoggedIn,authController.deleteTour,
(req,res) => {
    res.render('delete', {
        user: req.user,
        tours: result
    });
});

router.get('/insert', authController.isLoggedIn, (req,res) => {
    res.render('insert', {
        user: req.user
    });
});

router.get('/book', authController.isLoggedIn, (req,res) => {
    res.render('book', {
        user: req.user
    });
});

router.get('/trip', authController.isLoggedIn, (req,res) => {
    res.render('trip', {
        user: req.user
    });
});

router.get('/trip/search', authController.isLoggedIn, authController.searchTours,
(req, res) => {
```

```
    res.render('trip', {
      user: req.user,
      tours: req.tours,
      searchTerm: req.query.searchTerm
    });
  });


router.get('/viewtable', authController.isLoggedIn, authController.viewTable,
(req, res) => {
    res.render('viewtable', {
      user: req.user
    });
  });

  router.get('/viewtablet', authController.isLoggedIn, authController.viewTablet,
(req, res) => {
    res.render('viewtablet', {
      user: req.user
    });
  });


module.exports = router;
```

Routes auth.js :

```
const express = require('express');
const authController = require ('../controllers/auth');
const router = express.Router();

router.post('/register',authController.register);

router.post('/profilet',authController.updateUserDetails);

router.post('/profilea',authController.updateUserDetailsA);

router.post('/insert',authController.input);

router.post('/delete',authController.deleteTour);

router.post('/login',authController.login);

router.get('/logout',authController.logout);
```

```
router.post('/book',authController.books);


module.exports = router;
```

profila.hbs

```
<!DOCTYPE html>
<html>
<head>
    <title>Profile</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
```

```
            background-color: #333;
            color: #fff;
        }
        section {
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            align-items: stretch;
        }
        .card {
            background-color: #f1f1f1;
            border-radius: 5px;
            box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
            flex-grow: 1;
            margin: 10px;
            padding: 20px;
            text-align: center;
        }
        .card h2 {
            font-size: 24px;
            margin-top: 0;
        }
        .card p {
            font-size: 16px;
            margin-bottom: 0;
        }
        .profile-img {

            border-radius: 50%;
            object-fit: cover;
        }
        .avatar-ctn {
    text-align: center;
}

.avatar {
    width: 150px;
    height: 150px;
    border-bottom: 1px solid rgba(0,0,0,0.125);
}
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
```

```
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/h">Traveler Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="/h">Home</a>
                </li>
                <li class="nav-item active">
                    <a class="nav-link" href="/profilet">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/gallery">Gallery</a>
                </li>
                 <li class="nav-item">
                    <a class="nav-link" href="/viewtablet">Viewtours</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/trip">Trips</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/book">Booking</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
</ul>
</div>
</nav>
<section>
  <div class="card">
  <div class="avatar-ctn">
    <img src="/avt.png" class="avatar" alt="user profile picture" />
    </div>
    <h2>{{user.name}}</h2>
    <p>Traveler</p>
  </div>
  <div class="card">
    <h2>Update Your Details</h2>
```

```
        <p style="color:red;">Note: Please logout after changing your
detials</p><br>
    {{#if message}}
    <p class = "alert alert-danger mt-4">{{message}}</p>
    {{/if}}
    <br>
<form action="/auth/profilet" method="POST">
<input type="hidden" id="id" name="userId" value="{{user.id}}">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" ><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" ><br><br>
  <label for="phone">Phone:</label>
  <input type="tel" id="number" name="number"><br><br>
  <label for="password">Password:</label>
  <input type="password" id="password" name="password"><br><br>
  <input type="submit" value="Update">
</form>
  </div>
  <div class="card">
    <h2>Contact Info</h2>
    <p>Email: {{user.email}}</p>
    <p>Phone: {{user.number}}</p>
  </div>
</section>
<!-- Add jQuery library -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

profilt.hbs

```
<!DOCTYPE html>
<html>
<head>
    <title>Profile</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
```

```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
        section {
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            align-items: stretch;
        }
        .card {
            background-color: #f1f1f1;
            border-radius: 5px;
            box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
            flex-grow: 1;
```

```
            margin: 10px;
            padding: 20px;
            text-align: center;
        }
        .card h2 {
            font-size: 24px;
            margin-top: 0;
        }
        .card p {
            font-size: 16px;
            margin-bottom: 0;
        }
        .profile-img {

            border-radius: 50%;
            object-fit: cover;
        }
        .avatar-ctn {
    text-align: center;
}

.avatar {
    width: 150px;
    height: 150px;
    border-bottom: 1px solid rgba(0,0,0,0.125);
}
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/a">Agent Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item ">
                    <a class="nav-link" href="/a">Home</a>
                </li>
                <li class="nav-item active">
```

```
                    <a class="nav-link" href="/profilea">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/insert">Insert</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/delete">Delete</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/viewtable">View</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
            </ul>
        </div>
    </nav>
<section>
  <div class="card">
  <div class="avatar-ctn">
    <img src="/avt.png" class="avatar" alt="user profile picture" />
    </div>
    <h2>{{user.name}}</h2>
    <p>Travel Agent</p>
  </div>
  <div class="card">
    <h2>Update Your Details</h2>
    <p style="color:red;">Note: Please logout after changing your detials</p><br>
    {{#if message}}
    <p class = "alert alert-danger mt-4">{{message}}</p>
    {{/if}}
<form action="/auth/profilea" method="POST">
<input type="hidden" id="id" name="userId" value="{{user.id}}">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" ><br><br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" ><br><br>
  <label for="phone">Phone:</label>
  <input type="tel" id="number" name="number"><br><br>
  <label for="password">Password:</label>
  <input type="password" id="password" name="password"><br><br>
  <input type="submit" value="Update">
</form>
  </div>
```

```
  <div class="card">
    <h2>Contact Info</h2>
    <p>Email: {{user.email}}</p>
    <p>Phone: {{user.number}}</p>
  </div>
</section>
<!-- Add jQuery library -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>

</body>
</html>
```

Delete.hbs

```
<!DOCTYPE html>
<html>
<head>
    <title>Delete</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
```

```
        nav {
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
        section {
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            align-items: stretch;
        }
        .card {
            background-color: #f1f1f1;
            border-radius: 5px;
            box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
            flex-grow: 1;
            margin: 10px;
            padding: 20px;
            text-align: center;
        }
        .card h2 {
            font-size: 24px;
            margin-top: 0;
        }
        .card p {
            font-size: 16px;
            margin-bottom: 0;
        }
        .profile-img {

            border-radius: 50%;
            object-fit: cover;
```

```
        }
        .avatar-ctn {
    text-align: center;
}

.avatar {
    width: 150px;
    height: 150px;
    border-bottom: 1px solid rgba(0,0,0.125);
}
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/a">Agent Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item ">
                    <a class="nav-link" href="/a">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/profilea">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/insert">Insert</a>
                </li>
                <li class="nav-item active">
                    <a class="nav-link" href="/delete">Delete</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/viewtable">View</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
            </ul>
```

```
        </div>
    </nav>
<section>
  <div class="card">
  <div class="avatar-ctn">
    <img src="/avt.png" class="avatar" alt="user profile picture" />
    </div>
    <h2>{{user.name}}</h2>
    <p>Travel Agent</p>
  </div>
  <div class="card">
    <h2>Delete a Trip</h2>
<form action="auth/delete" method="POST">
  <label for="tourName">Select a tour to delete:</label>
  <select name="tourName" id="tourName">
    {{#each tours}}
    <option value="{{this.name}}">{{this.name}}</option>
    {{/each}}
  </select>
  <button type="submit">Delete</button>
</form>

  </div>
  <div class="card">
    <h2>Contact Info</h2>
    <p>Email: {{user.email}}</p>
    <p>Phone: {{user.number}}</p>
  </div>
</section>
<!-- Add jQuery library -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Gallery.hbs

```
<!DOCTYPE html>
```

```
<html>
<head>
    <title>Delete</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
        section {
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            align-items: stretch;
```

```
        }
        .card {
            background-color: #f1f1f1;
            border-radius: 5px;
            box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
            flex-grow: 1;
            margin: 10px;
            padding: 20px;
            text-align: center;
        }
        .card h2 {
            font-size: 24px;
            margin-top: 0;
        }
        .card p {
            font-size: 16px;
            margin-bottom: 0;
        }
        .profile-img {

            border-radius: 50%;
            object-fit: cover;
        }
        .avatar-ctn {
    text-align: center;
}

.avatar {
    width: 150px;
    height: 150px;
    border-bottom: 1px solid rgba(0,0,0,0.125);
}
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/a">Agent Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
```

```
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item ">
                    <a class="nav-link" href="/a">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/profilea">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/insert">Insert</a>
                </li>
                <li class="nav-item active">
                    <a class="nav-link" href="/delete">Delete</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/viewtable">View</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
            </ul>
        </div>
    </nav>
<section>
  <div class="card">
  <div class="avatar-ctn">
    <img src="/avt.png" class="avatar" alt="user profile picture" />
    </div>
    <h2>{{user.name}}</h2>
    <p>Travel Agent</p>
  </div>
  <div class="card">
    <h2>Delete a Trip</h2>
<form action="auth/delete" method="POST">
  <label for="tourName">Select a tour to delete:</label>
  <select name="tourName" id="tourName">
    {{#each tours}}
    <option value="{{this.name}}">{{this.name}}</option>
    {{/each}}
  </select>
  <button type="submit">Delete</button>
</form>

  </div>
```

```
  <div class="card">
    <h2>Contact Info</h2>
    <p>Email: {{user.email}}</p>
    <p>Phone: {{user.number}}</p>
  </div>
</section>
<!-- Add jQuery library -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Hometrav

```
<!DOCTYPE html>
<html>
<head>
    <title>Agent Dashboard</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
```

```
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
        section {
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            align-items: stretch;
        }
        .card {
            background-color: #f1f1f1;
            border-radius: 5px;
            box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
            flex-grow: 1;
            margin: 10px;
            padding: 20px;
            text-align: center;
        }
        .card h2 {
            font-size: 24px;
            margin-top: 0;
        }
        .card p {
            font-size: 16px;
            margin-bottom: 0;
        }
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
```

```
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/a">Agent Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item active">
                    <a class="nav-link" href="/a">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/profilea">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/insert">Insert</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/delete">Delete</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/viewtable">View</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
            </ul>
        </div>
    </nav>
    <section class="container">
        <div class="card">
            <h2>Welcome,{{user.name}}!</h2>
            <p>Here you can view your profile information, check your messages,
and customize your settings.</p>
        </div>
        <div class="card">
            <h2>Notifications</h2>
                <p>You have 3 new messages.</p>
        <p>Your account will expire in 5 days.</p>
        <p>There is a new feature available. Check it out!</p>

    </div>
```

```
    <div class="card">
        <h2>Recent Activity</h2>
        <p>No recent activity to display.</p>
    </div>
</section>
<!-- Add jQuery and Bootstrap JS -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"><
/script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Homeagent.hbs

```
<!DOCTYPE html>
<html>
<head>
    <title>Traveler Dashboard</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
            background-color: #f1f1f1;
```

```
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
        section {
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            align-items: stretch;
        }
        .card {
            background-color: #f1f1f1;
            border-radius: 5px;
            box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
            flex-grow: 1;
            margin: 10px;
            padding: 20px;
            text-align: center;
        }
        .card h2 {
            font-size: 24px;
            margin-top: 0;
        }
        .card p {
            font-size: 16px;
            margin-bottom: 0;
        }
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
    </header>
```

```
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/h">Traveler Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item active">
                    <a class="nav-link" href="/h">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/profilet">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/gallery">Gallery</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/viewtablet">Viewtours</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/trip">Trips</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/book">Booking</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
            </ul>
    </nav>
    <section class="container">
        <div class="card">
            <h2>Welcome,{{user.name}}!</h2>
            <p>Here you can view your profile information, view our tours
packages and book your trip.</p>
        </div>
        <div class="card">
            <h2>Notifications</h2>
                <p>You have 3 new messages.</p>
        <p>Your account will expire in 5 days.</p>
        <p>There is a new feature available. Check it out!</p>
```

```
    </div>
    <div class="card">
        <h2>Recent Activity</h2>
        <p>No recent activity to display.</p>
    </div>
</section>
<!-- Add jQuery and Bootstrap JS -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"><
/script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Login.hbs

```
<!DOCTYPE html>
<html>
<head>
    <title>Login Form</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
        <style>
  body {
    background-image: url('/back.jpg');
    background-repeat: no-repeat;
    background-size: cover;
  }
    .card {
        background-color: transparent;
        border: none;
    }

</style>
</head>
<body>
```

```
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-md-6">
                <div class="card">
                    <div class="card-header">
                        <h4>Login Form</h4>
                    </div>
                    <div class="card-body">
                                <form action="/auth/login" method="POST"
onsubmit="return validateForm()">
                                                 {{#if message}}
                                                <p class = "alert alert-danger
mt-4">{{message}}</p>
                                                {{/if}}
                                    <div class="form-group">
                                        <label for="email">Email</label>
                                        <input type="email" class="form-control"
id="email" name="email" placeholder="Enter email">
                                        <span id="emailError"
style="color:red"></span>
                                    </div>
                                    <div class="form-group">
                                        <label for="password">Password</label>
                                        <input type="password" class="form-
control" id="password" name="password" placeholder="Enter password">
                                        <span id="passwordError"
style="color:red"></span>
                                    </div>
                                    <div class="form-group">
                                        <label for="user-type">User Type</label>
                                        <select class="form-control" id="role"
name="role">
                                            <option>Traveler</option>
                                            <option>Travel agent</option>
                                        </select>
                                        <span id="roleError"
style="color:red"></span>
                                    </div>
                                    <div class="form-group">
                                   <p style="color:white;">Dont have an
account?   <button type="button" class="btn btn-secondary"><a href="/register"
style="color:white;">Register</a></button></p>
                                    </div>
                                     <button type="button" class="btn btn-
secondary"><a href="/"style="color:white;">Home</a></button>
```

```
                                    <button type="submit" class="btn btn-
secondary">Submit</button>
                                </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
<script>
function validateForm() {
    var email = document.getElementById("email").value;
    var password = document.getElementById("password").value;
    var role = document.getElementById("role").value;
    var emailError = document.getElementById("emailError");
    var passwordError = document.getElementById("passwordError");
    var roleError = document.getElementById("roleError");

    // Email validation
    if (email == "") {
        emailError.innerHTML = "Please enter an email";
        return false;
    } else {
        emailError.innerHTML = "";
    }

    // Password validation
    if (password == "") {
        passwordError.innerHTML = "Please enter a password";
        return false;
    } else {
        passwordError.innerHTML = "";
    }

    // Role validation
    if (role == "") {
        roleError.innerHTML = "Please select a role";
        return false;
    } else {
        roleError.innerHTML = "";
    }

    return true;
}
</script>
    <!-- Add Bootstrap JS -->
```

```
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"><
/script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Register

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration Form</title>
  <!-- Add Bootstrap CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
  body {
    background-image: url('/back.jpg');
    background-repeat: no-repeat;
    background-size: cover;
  }
    .card {
    background-color: transparent;
    border: none;
  }

</style>
</head>
<body>

  <div class="container mt-5">
    <div class="row justify-content-center">
      <div class="col-md-6">
        <div class="card">
          <div class="card-header">
            <h4>Registration Form</h4>
          </div>
```

```
          <div class="card-body">
            <form id="registration-form" action="/auth/register" method="POST">
                        {{#if message}}
                        <p class = "alert alert-danger mt-4">{{message}}</p>
                          {{/if}}
              <div class="form-group">
                <label for="name">Name</label>
                <input type="text" class="form-control" id="name" name="name"
placeholder="Enter name">
              </div>
              <div class="form-group">
                <label for="email">Email</label>
                <input type="email" class="form-control" id="email" name="email"
placeholder="Enter email">
              </div>
              <div class="form-group">
                <label for="password">Password</label>
                <input type="password" class="form-control" id="password"
name="password" placeholder="Enter password">
              </div>
              <div class="form-group">
                <label for="confirm_password">Confirm Password</label>
                <input type="password" class="form-control" id="confirm_password"
name="confirm_password" placeholder="Confirm password">
              </div>
              <div class="form-group">
                <label for="number">Phone Number</label>
                <input type="text" class="form-control" id="number" name="number"
placeholder="Enter number">
              </div>
              <div class="form-group">
                <label for="role">User Type</label>
                <select class="form-control" id="role" name="role" >
                  <option>Traveler</option>
                  <option>Travel agent</option>
                </select>
              </div>
                <div class="form-group">
                <p style="color:white;">Already have an account?<button
type="button" class="btn btn-secondary"> <a href="/login"
style="color:white;">Login</a></button></p>
                </div>
              <button type="button" class="btn btn-secondary"><a
href="/"style="color:white;">Home</a></button>
              <button type="submit" class="btn btn-secondary">Register</button>
```

```
            </form>
          </div>
        </div>
      </div>
    </div>
  <!-- Valiation of form -->
<script>
  const form = document.getElementById('registration-form');
  const nameInput = document.getElementById('name');
  const emailInput = document.getElementById('email');
  const passwordInput = document.getElementById('password');
  const confirmPasswordInput = document.getElementById('confirm_password');
  const numberInput = document.getElementById('number');
  const roleInput = document.getElementById('role');

  form.addEventListener('submit', (event) => {
    // Prevent form submission
    event.preventDefault();

    // Validate name
    if (nameInput.value === '') {
      alert('Please enter your name');
      nameInput.focus();
      return false;
    }

    // Validate email
    const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    if (!emailRegex.test(emailInput.value)) {
      alert('Please enter a valid email address');
      emailInput.focus();
      return false;
    }

    // Validate password
    if (passwordInput.value === '') {
      alert('Please enter a password');
      passwordInput.focus();
      return false;
    }

    // Validate confirm password
    if (confirmPasswordInput.value === '') {
      alert('Please confirm your password');
      confirmPasswordInput.focus();
```

```
      return false;
    }

    if (passwordInput.value !== confirmPasswordInput.value) {
      alert('Passwords do not match');
      confirmPasswordInput.focus();
      return false;
    }

    // Validate number
    if (numberInput.value === '') {
      alert('Please enter your phone number');
      numberInput.focus();
      return false;
    }

    // Validate role
    if (roleInput.value === '') {
      alert('Please select a user type');
      roleInput.focus();
      return false;
    }

    // If all validation passes, submit the form
    form.submit();
  });
</script>


  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.3/dist/umd/popper.min.js"
        integrity="sha384-
HAFz/XZD/H7VcdJl0u8bHm0fQ1A7v9/MWgxbl8Mktp/Oz2Lnd4x3qWU6wHJpvyua"
        crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Trip

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>Trip</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Add Bootstrap CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
  <style type="text/css">
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
    header {
      background-color: #3f3f3f;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
    nav {
      background-color: #f1f1f1;
      display: flex;
      flex-wrap: wrap;
      padding: 10px;
    }
    nav a {
      color: #333;
      padding: 10px;
      text-decoration: none;
      font-weight: bold;
      flex-grow: 1;
      text-align: center;
    }
    nav a:hover {
      background-color: #333;
      color: #fff;
    }
      .bg-image {
    background-image: url('/bk2.jpg');
    background-size: cover;
    background-position: center center;
    background-repeat: no-repeat;
  }
```

```
  </style>
</head>
<body>
  <header>
    <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
  </header>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="/h">Traveler Dashboard</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="/h">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/profilet">Profile</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/gallery">Gallery</a>
        </li>
       <li class="nav-item">
          <a class="nav-link" href="/viewtablet">Viewtours</a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="/trip">Trips</a>
        </li>
                <li class="nav-item ">
          <a class="nav-link" href="/book">Booking</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/auth/logout">Logout<i class="fa fa-power-
off" aria-hidden="true"></i></a>
        </li>
</ul>
</div>
</nav>
<!-- Search box section -->
<section class="py-5 bg-image">
  <div class="container">
    <div class="row">
      <div class="col-md-6 mx-auto text-center">
```

```
        <h2  style="color: white;">Search Available Tours</h2>
        <form action="/trip/search" method="GET" class="d-flex justify-content-
center mt-4">
          <input type="text" name="searchTerm" placeholder="Search destination"
class="form-control mr-2">
          <button type="submit" class="btn btn-secondary">Search</button>
        </form>
      </div>
    </div>
    <h2 class="mt-5 mb-3 text-center" style="color: white;">Available Tours</h2>
    <div class="row mt-5">
      <div class="col-md-10 mx-auto">
        <div class="table-responsive">
          <table class="table table-bordered">
            <thead class="thead-dark">
              <tr>
                <th scope="col">Place</th>
                <th scope="col">Duration</th>
                <th scope="col">Start Date</th>
                <th scope="col">End Date</th>
                <th scope="col">Booking</th>
                <th scope="col">Amount</th>
                <th scope="col">Last Date</th>
                <th scope="col">Offer Amount</th>
                <th scope="col">Valid Till</th>
                <th scope="col">Transport</th>
              </tr>
            </thead>
            <tbody>
              {{#if tours.length}}
                {{#each tours}}
                  <tr style="background-color: white; color: black;">
                    <td>{{name}}</td>
                    <td>{{duration}}</td>
                    <td>{{date}}</td>
                    <td>{{dateEnd}}</td>
                    <td>{{cost}}</td>
                    <td>{{amount}}</td>
                    <td>{{bookingdate}}</td>
                    <td>{{discount}}</td>
                    <td>{{valid}}</td>
                    <td>{{transport}}</td>
                  </tr>
                {{/each}}
              {{else if searchTerm}}
```

```
                <tr style="background-color: white; color: black;">
                  <td class="text-center" colspan="11">NO TOURS FOUND<i class="fa
fa-exclamation-triangle" aria-hidden="true"></i></td>
                </tr>
              {{/if}}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
</section>
<!-- Essential Travel Accessories section -->
<div class="py-5 bg-light" id="accessories">
  <div class="container my-5">
    <div class="row">
      <div class="col-md-8 mx-auto">
    <h1 class="text-center mb-5"><i class="fa fa-suitcase" aria-
hidden="true"></i></h1>
        <h1 class="text-center mb-5">Essential Travel Accessories</h1>
        <p>When packing for a trip, it's important to have the right tools and
accessories to ensure a smooth and enjoyable travel experience. Here are some
essential travel accessories that you should consider bringing:</p>
        <ul>
          <li>Passport</li>
          <li>Travel pillow</li>
          <li>Noise-cancelling headphones</li>
          <li>Portable charger</li>
          <li>Universal adapter</li>
          <li>Reusable water bottle</li>
          <li>Lightweight backpack</li>
        </ul>
        <p>Here's a more detailed list of some of the essential travel
accessories:</p>
        <dl>
          <dt>Passport</dt>
          <dd>Your passport is the most important document you'll need when
traveling internationally. Make sure you have a valid passport that won't expire
for at least six months.</dd>
          <dt>Travel pillow</dt>
          <dd>A travel pillow can help you get some rest during long flights or
train rides. Look for one that is compact and easy to pack.</dd>
          <dt>Noise-cancelling headphones</dt>
```

```
          <dd>Noise-cancelling headphones can help you block out unwanted noise
and distractions, making your travel experience more comfortable and
relaxing.</dd>
          <dt>Portable charger</dt>
          <dd>A portable charger can help you keep your devices charged on the
go, so you don't have to worry about running out of battery.</dd>
          <dt>Universal adapter</dt>
          <dd>A universal adapter can help you stay connected and charge your
devices in foreign countries. Make sure to check the voltage and plug type of
your destination before purchasing one.</dd>
          <dt>Reusable water bottle</dt>
          <dd>A reusable water bottle can help you save money and reduce waste by
refilling it instead of buying bottled water. Look for one that is durable and
leak-proof.</dd>
          <dt>Lightweight backpack</dt>
          <dd>A lightweight backpack can be a versatile and convenient way to
carry your essentials while exploring your destination. Look for one that is
comfortable to wear and has enough pockets and compartments to keep your
belongings organized.</dd>
        </dl>
      </div>
    </div>
  </div>
</div>
<!-- Add jQuery library -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

View table

```
<!DOCTYPE html>
<html>
<head>
    <title>Delete</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Add Bootstrap CSS -->
```

```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
    <style type="text/css">
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
        header {
            background-color: #3f3f3f;
            color: #fff;
            padding: 20px;
            text-align: center;
        }
        nav {
            background-color: #f1f1f1;
            display: flex;
            flex-wrap: wrap;
            padding: 10px;
        }
        nav a {
            color: #333;
            padding: 10px;
            text-decoration: none;
            font-weight: bold;
            flex-grow: 1;
            text-align: center;
        }
        nav a:hover {
            background-color: #333;
            color: #fff;
        }
          .bg-image {
    background-image: url('/bk3.jpg');
    background-size: cover;
    background-position: center center;
    background-repeat: no-repeat;
  }
    </style>
</head>
<body>
    <header>
        <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
```

```
    </header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="/a">Agent Dashboard</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item ">
                    <a class="nav-link" href="/a">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/profilea">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/insert">Insert</a>
                </li>
                <li class="nav-item ">
                    <a class="nav-link" href="/delete">Delete</a>
                </li>
                <li class="nav-item active">
                    <a class="nav-link" href="/viewtable">View</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/auth/logout">Logout<i class="fa
fa-power-off" aria-hidden="true"></i></a>
                </li>
            </ul>
        </div>
    </nav>
<!-- table section -->
<section class="py-5 bg-image">
  <div class="container">
      <h2 class="mt-5 mb-3 text-center">Tours Details</h2>
    <div class="row mt-5">
      <div class="col-md-10 mx-auto">
        <div class="table-responsive">
          <table class="table table-bordered">
            <thead class="thead-dark">
              <tr>
                <th scope="col">Place</th>
                <th scope="col">Duration</th>
                <th scope="col">Start Date</th>
```

```
                <th scope="col">End Date</th>
                <th scope="col">Booking</th>
                <th scope="col">Amount</th>
                <th scope="col">Last Date</th>
                <th scope="col">Offer Amount</th>
                <th scope="col">Valid Till</th>
                <th scope="col">Transport</th>
              </tr>
            </thead>
            <tbody>
            {{#each tours}}
              <tr style="background-color: white; color: black;">
                <td>{{this.name}}</td>
                <td>{{this.duration}}</td>
                <td>{{this.date}}</td>
                <td>{{this.dateEnd}}</td>
                <td>{{this.cost}}</td>
                <td>{{this.amount}}</td>
                <td>{{this.bookingdate}}</td>
                <td>{{this.discount}}</td>
                <td>{{this.valid}}</td>
                <td>{{this.transport}}</td>
              </tr>
              {{/each}}
            </tbody>
          </table>
        </div>
      </div>
    </div>
    </div>
</section>
<!-- Add jQuery library -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Insert.hbs

```
<!DOCTYPE html>
<html>
<head>
  <title>Insert</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Add Bootstrap CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css">
  <style type="text/css">
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
    header {
      background-color: #3f3f3f;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
    nav {
      background-color: #f1f1f1;
      display: flex;
      flex-wrap: wrap;
      padding: 10px;
    }
    nav a {
      color: #333;
      padding: 10px;
      text-decoration: none;
      font-weight: bold;
      flex-grow: 1;
      text-align: center;
    }
    nav a:hover {
      background-color: #333;
      color: #fff;
    }
    section {
```

```
      padding: 20px;
      display: flex;
      flex-wrap: wrap;
      align-items: stretch;
    }
    .card {
      background-color: #f1f1f1;
      border-radius: 5px;
      box-shadow: 2px 2px 5px rgba(0,0,0,0.3);
      flex-grow: 1;
      margin: 10px;
      padding: 20px;
      text-align: center;
    }
    .card h2 {
      font-size: 24px;
      margin-top: 0;
    }
    .card p {
      font-size: 16px;
      margin-bottom: 0;
    }
    .profile-img {

      border-radius: 50%;
      object-fit: cover;
    }
    .avatar-ctn {
    text-align: center;
}

.avatar {
    width: 150px;
  height: 150px;
    border-bottom: 1px solid rgba(0,0,0,0.125);
}
  </style>
</head>
<body>
  <header>
    <h1 style="color:black;">Trip<span style="color:white">Bots</span></h1>
  </header>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="/a">Agent Dashboard</a>
```

```
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item ">
          <a class="nav-link" href="/a">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/profilea">Profile</a>
        </li>
        <li class="nav-item active">
          <a class="nav-link" href="/Insert">Insert</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/delete">Delete</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/viewtable">View</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/auth/logout">Logout<i class="fa fa-power-
off" aria-hidden="true"></i></a>
        </li>
      </ul>
    </div>
  </nav>
<section>
  <div class="card">
  <div class="avatar-ctn">
    <img src="/avt.png" class="avatar" alt="user profile picture" />
  </div>
    <h2>{{user.name}}</h2>
    <p>Travel Agent</p>
  </div>
   <div class="card">
    <h2>Insert Travel Details</h2>
   {{#if message}}
  <p class = "alert alert-danger mt-4">{{message}}</p>
  {{/if}}
<form id="trip-form" action="/auth/insert" method="POST">
  <div class="form-group">
    <label for="tour-place">Tour Place Name:</label>
```

```
    <input type="text" class="form-control" id="tour-place" name="tourPlace"
placeholder="Enter the tour place name" required>
  </div>
  <div class="form-group">
    <label for="tour-duration">Tour Duration:</label>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" id="tour-duration-3"
name="tourDuration" value="3 days">
      <label class="form-check-label" for="tour-duration-3">
        3 Days
      </label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" id="tour-duration-5"
name="tourDuration" value="5 days">
      <label class="form-check-label" for="tour-duration-5">
        5 Days
      </label>
    </div>
    <div class="form-check">
      <input class="form-check-input" type="checkbox" id="tourDuration-7"
name="tour-duration" value="7 days">
      <label class="form-check-label" for="tour-duration-7">
        7 Days
      </label>
    </div>
  </div>
  <div class="form-group">
    <label for="tour-date">Tour Date:</label>
    <input type="date" class="form-control" id="tour-date" name="tourDate"
required>
  </div>
  <div class="form-group">
    <label for="tour-date-end">Tour Date End:</label>
    <input type="date" class="form-control" id="tour-date-end" name="tourDateEnd"
required>
  </div>
  <div class="form-group">
    <label for="tour-cost">Tour Cost:</label>
    <input type="number" class="form-control" id="tour-cost" name="tourCost"
placeholder="Enter the tour cost" required>
  </div>
  <div class="form-group">
    <label for="amount">Amount with VAT:</label>
```

```
    <input type="number" class="form-control" id="amount" name="amount"
placeholder="Enter the amount" required>
  </div>
  <div class="form-group">
    <label for="last-date">Last Booking Date:</label>
    <input type="date" class="form-control" id="last-date" name="lastDate"
required>
  </div>
  <div class="form-group">
    <label for="offer-amount">Discount Offer Amount:</label>
    <input type="number" class="form-control" id="offer-amount"
name="offerAmount" placeholder="Enter the offer amount">
  </div>
  <div class="form-group">
    <label for="valid-till">Valid Till:</label>
    <input type="date" class="form-control" id="valid-till" name="validTill"
placeholder="Enter the valid till date">
  </div>
  <div class="form-group">
    <label for="transport">Transport:</label>
    <div class="form-check">
      <input class="form-check-input transport-option" type="checkbox"
id="transport-car" name="transport" value="car">
      <label class="form-check-label" for="transport-car">
        Car
      </label>
    </div>
    <div class="form-check">
      <input class="form-check-input transport-option" type="checkbox"
id="transport-train" name="transport" value="train">
      <label class="form-check-label" for="transport-train">
        Train
      </label>
    </div>
    <div class="form-check">
      <input class="form-check-input transport-option" type="checkbox"
id="transport-bus" name="transport" value="bus">
      <label class="form-check-label" for="transport-bus">
        Bus
      </label>
    </div>
  </div>
        <button type="submit" class="btn btn-secondary">Submit</button>
      </form>
  </div>
```

```
    <div class="card">
    <h2>Contact Info</h2>
    <p>Email: {{user.email}}</p>
    <p>Phone: {{user.number}}</p>
  </div>
</section>
<!-- form validation -->
<!-- form validation
<script>
const form = document.getElementById('trip-form');
const tourPlace = document.getElementById('tour-place');
const tourDuration = document.querySelectorAll('input[name="tour-
duration"]:checked');
const tourDate = document.getElementById('tour-date');
const tourDateEnd = document.getElementById('tour-date-end');
const tourCost = document.getElementById('tour-cost');
const amount = document.getElementById('amount');
const lastDate = document.getElementById('last-date');
const transportOptions = document.querySelectorAll('.transport-option');

form.addEventListener('submit', (event) => {
  let isFormValid = true;

  if (!tourPlace.value) {
    isFormValid = false;
    alert('Please enter the tour place name');
  }

  if (tourDuration.length === 0) {
    isFormValid = false;
    alert('Please select a tour duration');
  }

  if (!tourDate.value) {
    isFormValid = false;
    alert('Please enter the tour start date');
  }

  if (!tourDateEnd.value) {
    isFormValid = false;
    alert('Please enter the tour end date');
  } else if (new Date(tourDateEnd.value) < new Date(tourDate.value)) {
    isFormValid = false;
    alert('Tour end date must be after tour start date');
  }
```

```
  if (!tourCost.value) {
    isFormValid = false;
    alert('Please enter the tour cost');
  } else if (tourCost.value <= 0) {
    isFormValid = false;
    alert('Tour cost must be greater than zero');
  }

  if (!amount.value) {
    isFormValid = false;
    alert('Please enter the amount');
  } else if (amount.value <= 0) {
    isFormValid = false;
    alert('Amount must be greater than zero');
  }

  if (!lastDate.value) {
    isFormValid = false;
    alert('Please enter the last booking date');
  } else if (new Date(lastDate.value) < new Date()) {
    isFormValid = false;
    alert('Last booking date must be in the future');
  }

  let transportSelected = false;
  for (let i = 0; i < transportOptions.length; i++) {
    if (transportOptions[i].checked) {
      transportSelected = true;
      break;
    }
  }
  if (!transportSelected) {
    isFormValid = false;
    alert('Please select at least one mode of transport');
  }

  if (!isFormValid) {
    event.preventDefault();
  }
});

</script>
-->
<!-- Add jQuery library -->
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Add Bootstrap JS -->
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></scrip
t>

</body>
</html>
```

App.js

```
const express = require("express");
const path = require("path");
const mysql = require("mysql");
const dotenv = require("dotenv");
const cookieParser = require('cookie-parser');

dotenv.config ({ path: "./.env" });

const app = express();

const publicDirectory = path.join(__dirname, "./public");
app.use(express.static(publicDirectory));

// Parse - URL- encoded bodies(as sent by HTML forms)
app.use (express.urlencoded({extended: false}));

// Parse - JSON bodies (as sent by API clients)
app.use(express.json());

// cookies
app.use(cookieParser());

// Database connection
const db = mysql.createConnection({
  host: process.env.DATABASE_HOST,
  user: process.env.DATABASE_USER,
  password: process.env.DATABASE_PASSWORD,
  database: process.env.DATABASE,
});

db.connect((error) => {
  if (error) {
```

```
    console.log(error);
  } else {
    console.log("MYSQL Connected..");
   // db.query("DESCRIBE users", (error, result) => {
    //  if (error) {
    //    console.log(error);
    //  } else {
    //    console.log(result);
    //  }
   // });
  }
});


app.set("view engine", "hbs");

// Define Routes
app.use('/', require('./routes/pages'));
app.use('/auth', require('./routes/auth'));



app.listen(5000, () => {
  console.log("Server started on port 5000");
});
```

.env

```
   DATABASE = fp_login
DATABASE_HOST = Localhost
DATABASE_USER = root
DATABASE_PASSWORD =
JWT_SECRET = mysupersecretpassword
JWT_EXPIRES_IN = 90d
JWT_COOKIE_EXPIRES = 90
```