AN WEB APPLICATION

FOR TRAVELLO- The Ultimate Tour Booking Platform

By

VYOM RANA

(ID No.:- 20CEUBS046)

A project submitted
In
Partial fulfillment of the requirements
for the degree of
BACHELORE OF TECHNOLOGY
In
Computer Engineering

Internal Guide
Prof. Bhavika Gambhava
Associate Professor
Dept. of Comp. Engg.

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2024

CERTIFICATE

This is to certify that the project work titled

TRAVELLO- THE ULTIMATE TOUR
BOOKING PLATFORM

is the bonafide work of

VYOM RANA (Id No.:- 20CEUBS046)

carried out in the partial fulfillment of the degree of Bachelor of Technology in Computer Engineering at Dharmsinh Desai University in the academic session
December 2023 to April 2024.

Prof. Bhavika Gambhava	*Dr. C.K. Bhensdadia*
Associate Professor	*Head,*
Dept. of Comp. Engg.	*Dept. of Comp. Engg.*

Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University
April 2024

Acknowledgements

I am immensely grateful to all those who supported us throughout this project journey. Their contributions were instrumental in its success. I extend my heartfelt thanks to our institute for providing us with the platform to apply our theoretical knowledge practically. This project has been a significant part of our curriculum, and we are grateful for the opportunity.

Special gratitude goes to our guide, **Prof. Bhavika M. Gambhava**, whose unwavering support and vast knowledge guided us through every step of this project. Her expertise, guidance, and remarkable testing abilities were indispensable in achieving perfection in the tasks. Without her efforts, this project would not have reached its successful conclusion.

I am thankful to the faculty and staff of **Dharmsinh Desai University**, especially **Dr. C. K. Bhensdadia,** for providing the necessary resources and facilities. Their encouragement has motivated me to strive for excellence in this endeavor

I also want to thank everyone who asked insightful questions and helped me navigate through various crises encountered during the project. Your support was invaluable, and I deeply appreciate it.

Theoretical knowledge is of no importance if one doesn’t know the way of its implementation. I am thankful to our institute that provided us an opportunity to apply our theoretical knowledge through the project. I feel obliged in submitting this project as part of our curriculum.

With Sincere Regards,
Vyom Rana,

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Introduction:

The Travello - Tour Booking Website project is aimed at revolutionizing the way travel enthusiasts discover, plan, and book tours worldwide. With a comprehensive suite of functionalities, intuitive user interfaces, and seamless interactions, Travello strives to provide a seamless and enjoyable experience for both users and administrators alike.

1.1 Overview:

In today's fast-paced world, where travel has become an integral part of lifestyle and exploration, the need for a reliable platform that simplifies tour discovery and booking processes has never been more apparent. Travello aims to fill this gap by offering a one-stop solution for travelers to explore diverse tour options, make hassle-free bookings, and share their experiences through reviews and ratings.

Travello adopts a scalable architecture to anticipate the evolving needs of travel agencies, ensuring adaptability to changing business requirements. Additionally, it implements lazy loading for optimized performance. Lazy loading allows Travello's systems to load resources or data only when they are needed, thus enhancing efficiency and reducing unnecessary overhead.

In summary, Travello emerges as a promising solution for travel agencies, offering a powerful suite of features for effective tour management and customer interaction. The abstract provides a concise overview of the project's goals, scope, and limitations, setting the stage for a detailed exploration in the following sections of the project report.

1.2 Objectives and Goals:

Efficient Tour Discovery: Enable users to search and discover tours based on location, group size, and distance, ensuring personalized recommendations tailored to their preferences.

Seamless Booking Experience: Implement a seamless booking process with features such as guest size limits, real-time availability updates, and secure payment integration using Razorpay.

Engaging User Interaction: Increases user engagement through a robust review system, allowing users to share their experiences, rate tours, and provide valuable feedback for future travelers.

Responsive UI Design: Ensure a responsive user interface that adapts to various devices, providing a consistent and optimal user experience across desktops, tablets, and smartphones.

1.3 Scope and Limitations:

The scope of the Travello project encompasses the development of a full-fledged tour booking website with a wide range of functionalities, including tour search, booking management, review submission, and user profile management. While the project aims to deliver a comprehensive solution, certain limitations exist, such as:

Payment Integration: While the platform facilitates booking processes, it may not handle actual payment transactions in the initial release. Integration with payment gateways can be considered as a future enhancement.

Advanced Weather Integration: While users can view future 7-day weather forecasts during booking, more advanced weather-related features may be limited in the initial release.

Enhanced User Management: While basic user registration and login functionalities are implemented, more advanced features such as social login or two-factor authentication may be explored in future iterations.

Chat-Bot Integration: Creating a chatbot to assist users in selecting a tour tailored to their preferences involves designing a conversational interface that guides users through the selection process.

Chapter 2

About The System

2.1 Software Requirement Specification:

2.1.1 Functional Requirements:

R.1: Manage Users

R.1.1: Add User:
Description: New users can create an account by providing necessary information.
Input: Username, Password, Email Address.
Output: User account created.
Next Operation: User verification using link.

R.1.2: Login User:
Description: Existing users can log in to their accounts.
Input: Email Address, Password.
Output: User account logged in.

R.1.3: Update User:
Description: Users can update their profile information such as username, password and Email.
Input: Updated User details.
Output: User Updated if email not already exists.

R.1.4: Logout User:
Description: Users can log out from their accounts.
Input: User selection.
Output: User Profile Logout.

R.1.5: View Users:
Description: Admin can see the list of users and admin can ban user or make other user an admin user.
Input: Admin selection.
Output: List of users.

R.2: Manage Bookings

R.2.1: Add Booking:
Description: Users can easily pick and book the tours they want , they choose where they want to go and when.
Input: Details of user and tour.
Output: Tour Booked Successfully.

R.2.2: View Bookings:
Description: Users and admin can browse and view bookings.
Input: User selection.
Output: User see list of all booking with their details.

R.2.3: Cancel Booking:
Description: Users have the option to cancel their booking if it's scheduled for a future date and the difference between the current day and the booked day is more than one day.
Input: User selection.
Output: Booking is cancel and user see red colored on booking page.

R.3: Manage Reviews

R.3.1: Add Review:
Description: Users can add review for the tour that they have experience the tour.
Input: User rating by selecting stars and small message.
Output: Review added to Tours page.

R.3.2: View Reviews:
Description: Users can see the reviews for the particular tour
Input: User Selection for tour.
Output: See list of reviews for that tour with user name , review message and stars.

R.4: Manage Tour

R.4.1: Add Tour:
Description: Admin user can add tour package so other users can book it.
Input: Tour name, days , description, photo, price ,city.
Output: Tour created successfully.

R.4.2: Update Tour:
Description: Admin can update the existing tours.
Input: Updated Details of the tour.
Output: Details of the tour is updated.

R.4.3: View Tour:
Description: Users can view the tours that are not paused by the admin and search tour by name.
Input: User selection.
Output: All tours with description in a card form.

R.4.4: Delete Tour:
Description: Admin user can delete the tour.
Input: User selection.
Output: Tour is deleted successfully.

R.4.5: Pause Tour:
Description: Admin can pause the tour so no user can book it.
Input: Admin selection.
Output: Tour is paused and users can’t book the tour.

R.4.5: UnPause Tour:
Decription: Admin can unpause the tour so user can book it.
Input: Admin selection.
Output: Tour is unpaused and users can book the tour.

2.1.2 Non-Functional Requirements:

- **Performance**: The system should respond quickly to user interactions, with minimal latency in loading pages, processing bookings, and displaying reviews.
- **Scalability**: The platform should be able to handle increasing numbers of users, tours, and reviews without a significant degradation in performance.
- **Reliability**: The system should be highly reliable, ensuring minimal downtime and maintaining data integrity, especially during peak booking periods.
- **Security**: User data, including personal information and payment details, should be securely stored and transmitted using encryption protocols to prevent unauthorized access or data breaches.
- **Usability**: The user interface should be intuitive and easy to navigate, catering to users of all technical backgrounds and ensuring a pleasant booking and review experience.
- **Accessibility**: The system should be accessible to users with disabilities, complying with accessibility standards such as WCAG (Web Content Accessibility Guidelines) to ensure inclusivity.
- **Compatibility**: The platform should be compatible with a wide range of devices and browsers, ensuring a consistent user experience across different platforms and screen sizes.
- **Maintainability**: The system should be designed with modularity and clean code practices to facilitate easy maintenance, updates, and enhancements over time.
- **Auditability**: The system should maintain logs of user actions, bookings, and reviews for auditing purposes, ensuring accountability and traceability of data changes.

Chapter 3

Analysis

3.1 ER Diagram: The ER diagram represents entities for users, reviews, bookings, and tours. Bookings are associated with users and tours . Tours can have multivalued attributes like reviews. Additionally, there may be weak entities such as Booking, which are dependent on the tour entity and User entity but don't have a unique identifier on their own similarly , Reviews is weak entity dependent on User and Tour.

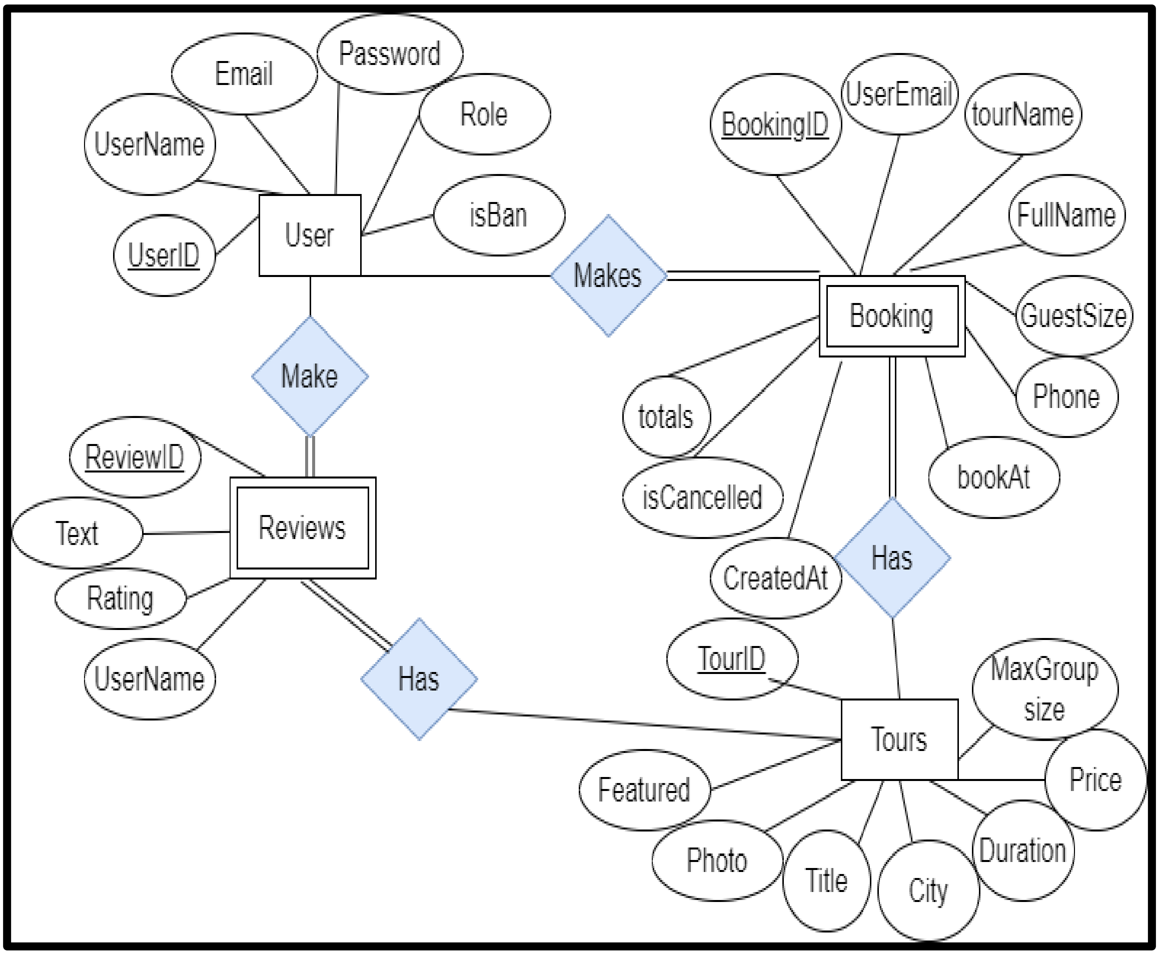
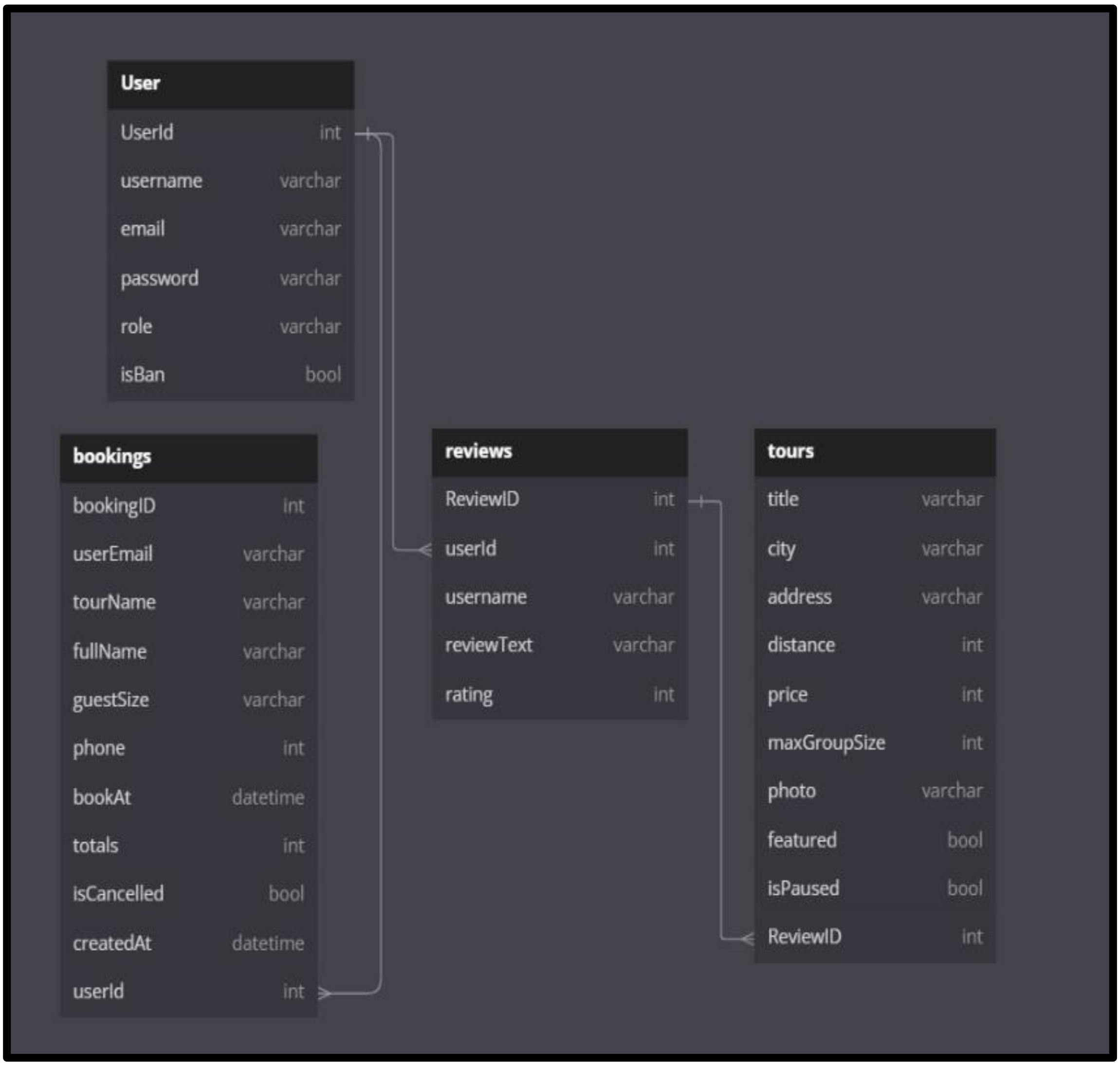


Figure 3.1 ER-Diagram

3.2 Database Schema:- Database Schema consist of fields which each entity has and its relationship. User has one to many relationship with reviews, and bookings.Tour and reviews are related with one to many relationship. Also bookings are related to tour with many to one relationship.



User	
UserId	int
username	varchar
email	varchar
password	varchar
role	varchar
isBan	bool

bookings	
bookingID	int
userEmail	varchar
tourName	varchar
fullName	varchar
guestSize	varchar
phone	int
bookAt	datetime
totals	int
isCancelled	bool
createdAt	datetime
userId	int

reviews	
ReviewID	int
userId	int
username	varchar
reviewText	varchar
rating	int

tours	
title	varchar
city	varchar
address	varchar
distance	int
price	int
maxGroupSize	int
photo	varchar
featured	bool
isPaused	bool
ReviewID	int

Figure 3.2 Database Schema

Data Flow Diagram:

3.3 DFD Level-0: The Level 0 Data Flow Diagram (DFD) for Travello provides a high-level overview of the system's data flow and processing. It illustrates the interactions between external entities such as users and the core system components like user management, Review sharing, and booking management . This diagram serves as a top-level blueprint for understanding the flow of data and functionalities within the Travello application, facilitating system design and development.

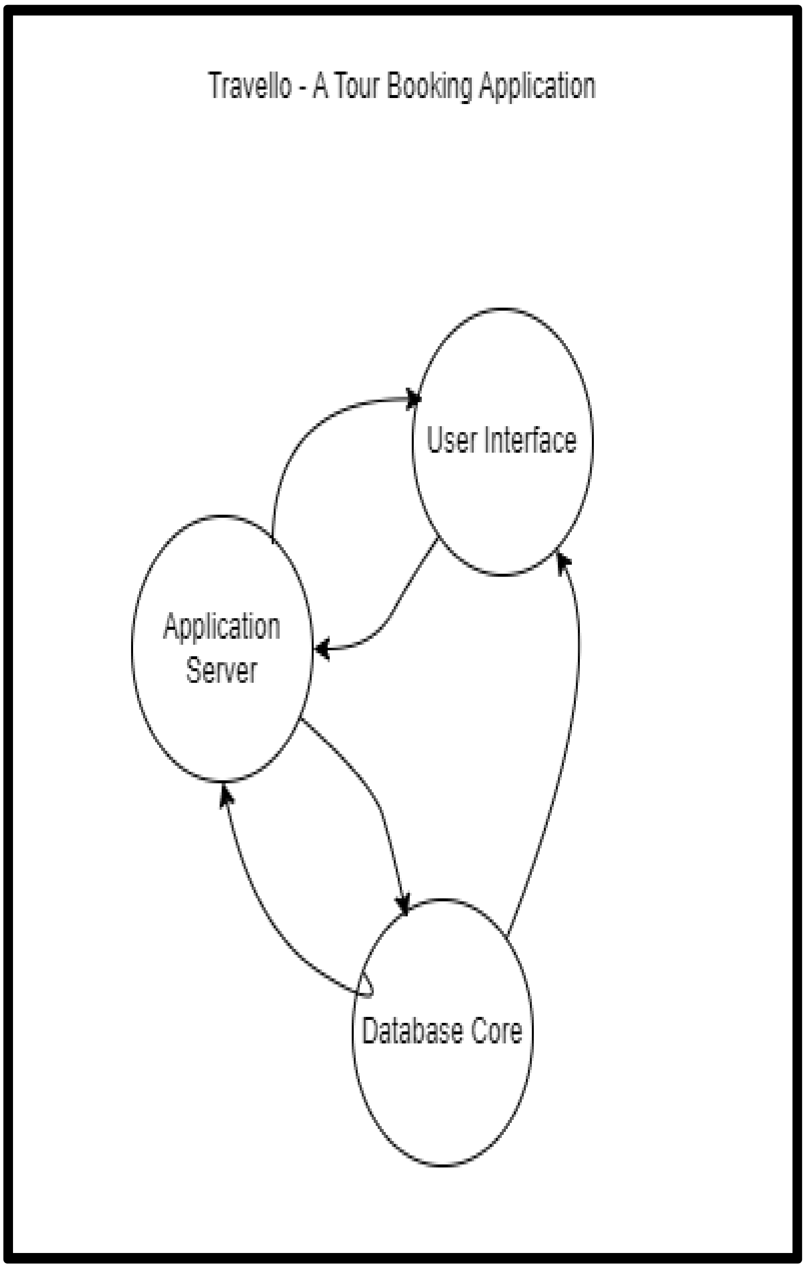


Figure 3.3 DFD Level-0

3.4 DFD Level-1:- The Level 1 Data Flow Diagram (DFD) for Travello offers a more detailed depiction of the system's data flow and processing compared to Level 0. It expands on the core functionalities represented in Level 0, breaking them down into smaller, more manageable processes and subprocesses. This diagram illustrates the interactions between various system components, highlighting the flow of data and control as it moves through the different modules and functions within the Travello application. It serves as a guide for developers to understand the internal workings of the system and how data is processed at a more granular level.

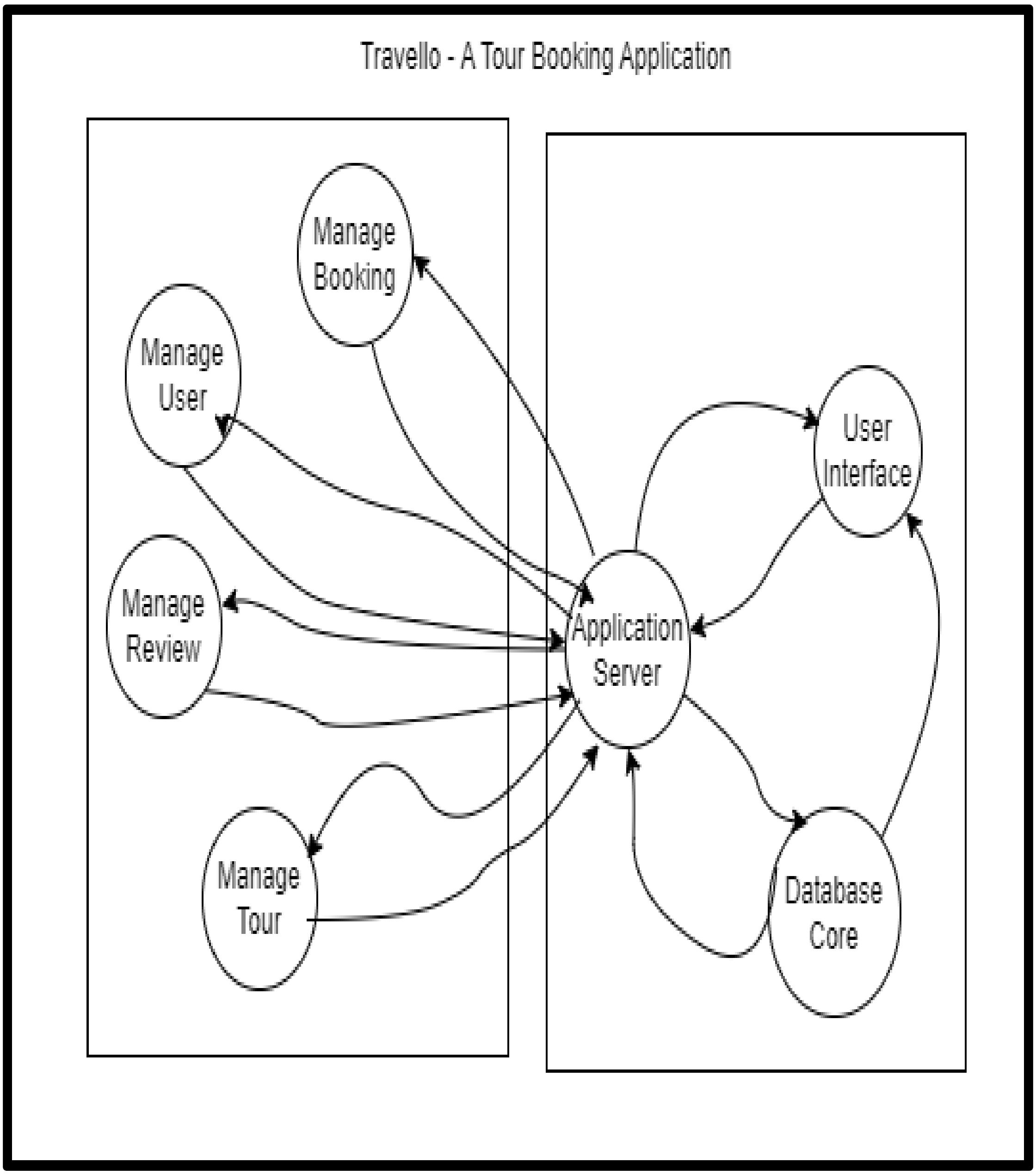


Figure 3.4 DFD Level-1

3.5 DFD Level-2:- The Level 2 Data Flow Diagram (DFD) for Travello tell even deeper into the system's data flow and processing compared to Level 1. It provides a detailed breakdown of the processes and subprocesses identified in Level 1, further refining the interactions between system components and data entities. This diagram offers a closer look at the internal workings of the Travello application, illustrating how data is transformed and manipulated within each individual process. It serves as a comprehensive guide for developers to understand the intricacies of data flow and processing at a highly detailed level within the system.

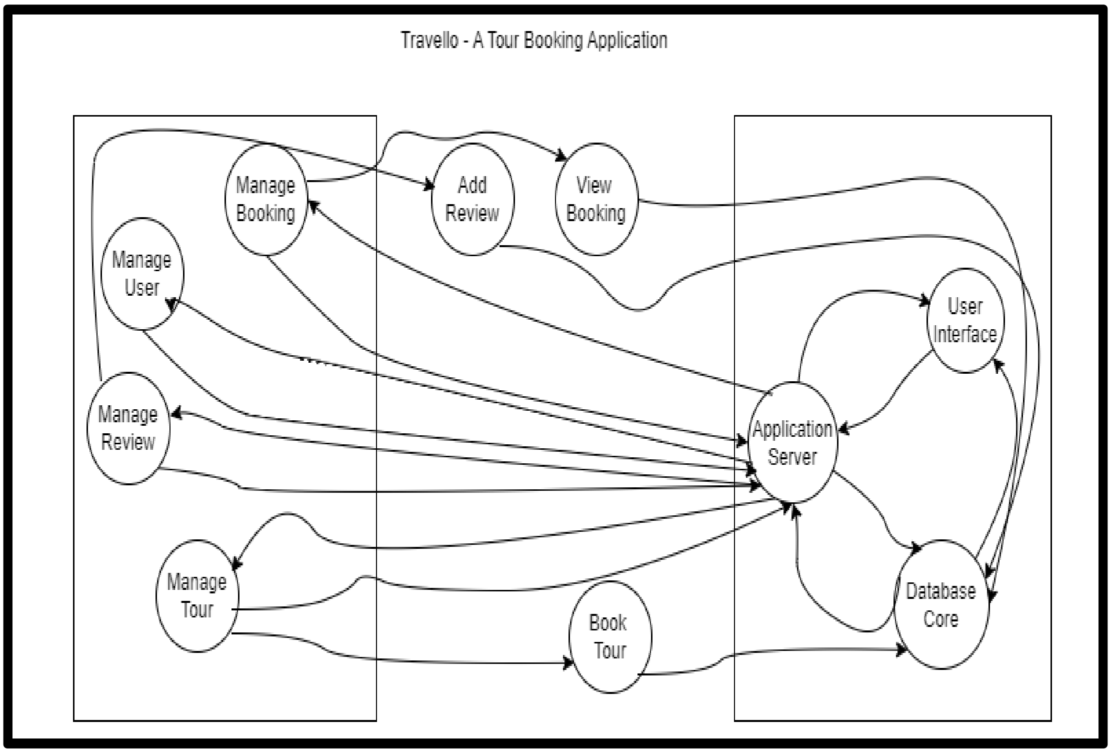


Figure 3.5 DFD Level-2

3.6 Usecase Diagram – Normal USER: The use case diagram illustrates the various interactions between users and the system. It includes actions such as adding a user, logging in, updating user information, and logging out. Users can also interact with tours by viewing, booking, and reviewing them. Additionally, they can view their bookings, cancel them if needed, and search for tours. Furthermore, the system provides the functionality to view weather information and enables users to receive emails for notifications or updates.

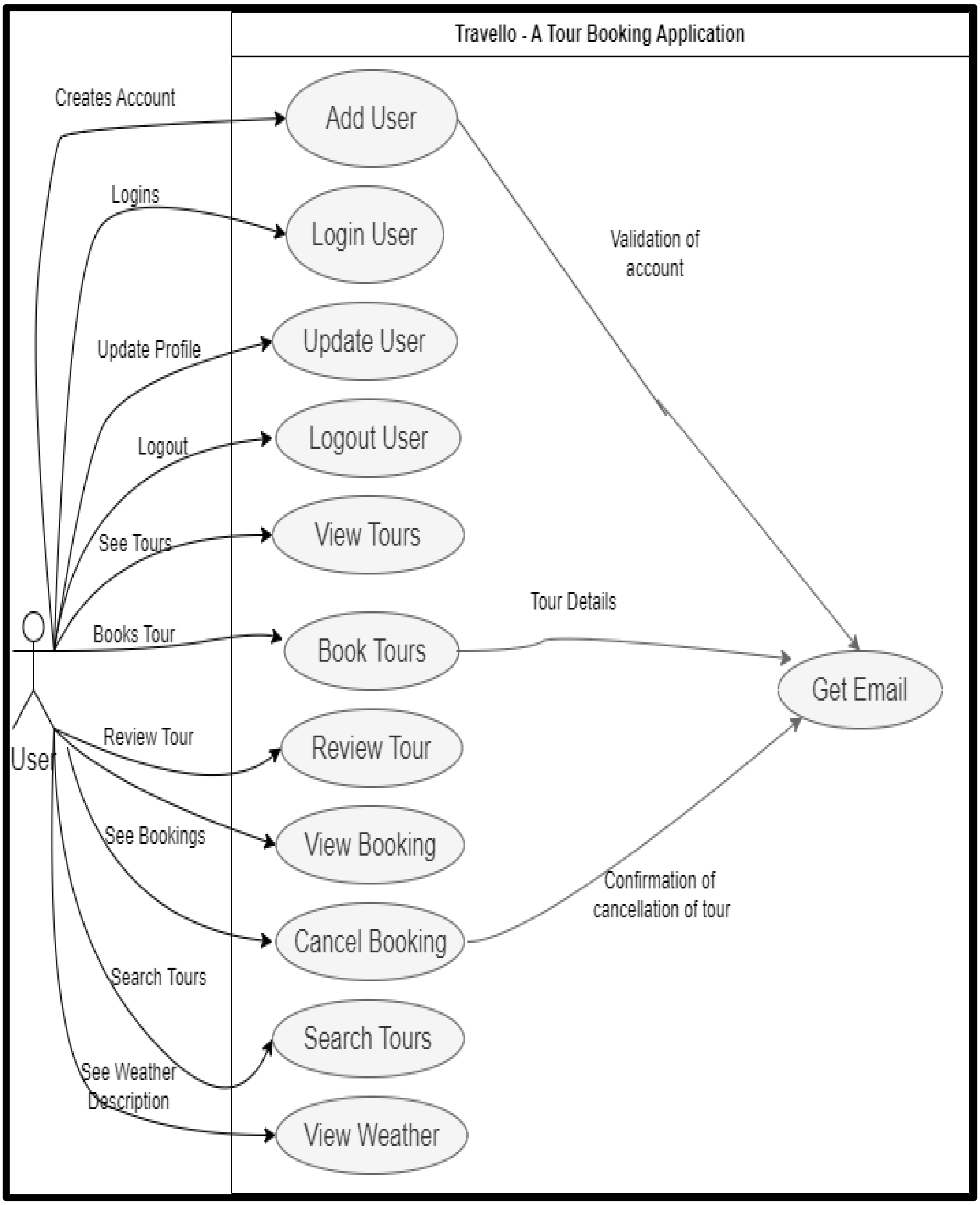


Figure 3.6 Use Case Diagram (Normal User)

3.7 Usecase Diagram – Admin User:- The admin use case diagram outlines key functionalities exclusive to administrators. It includes actions like banning users and promoting them to admin status, alongside standard user operations such as login, logout, and updating user details. Admins can manage tours by adding, updating, and deleting them, as well as pausing and unpausing tours as needed. They also have access to viewing bookings, searching for tours, and ensuring seamless tour management within the system.

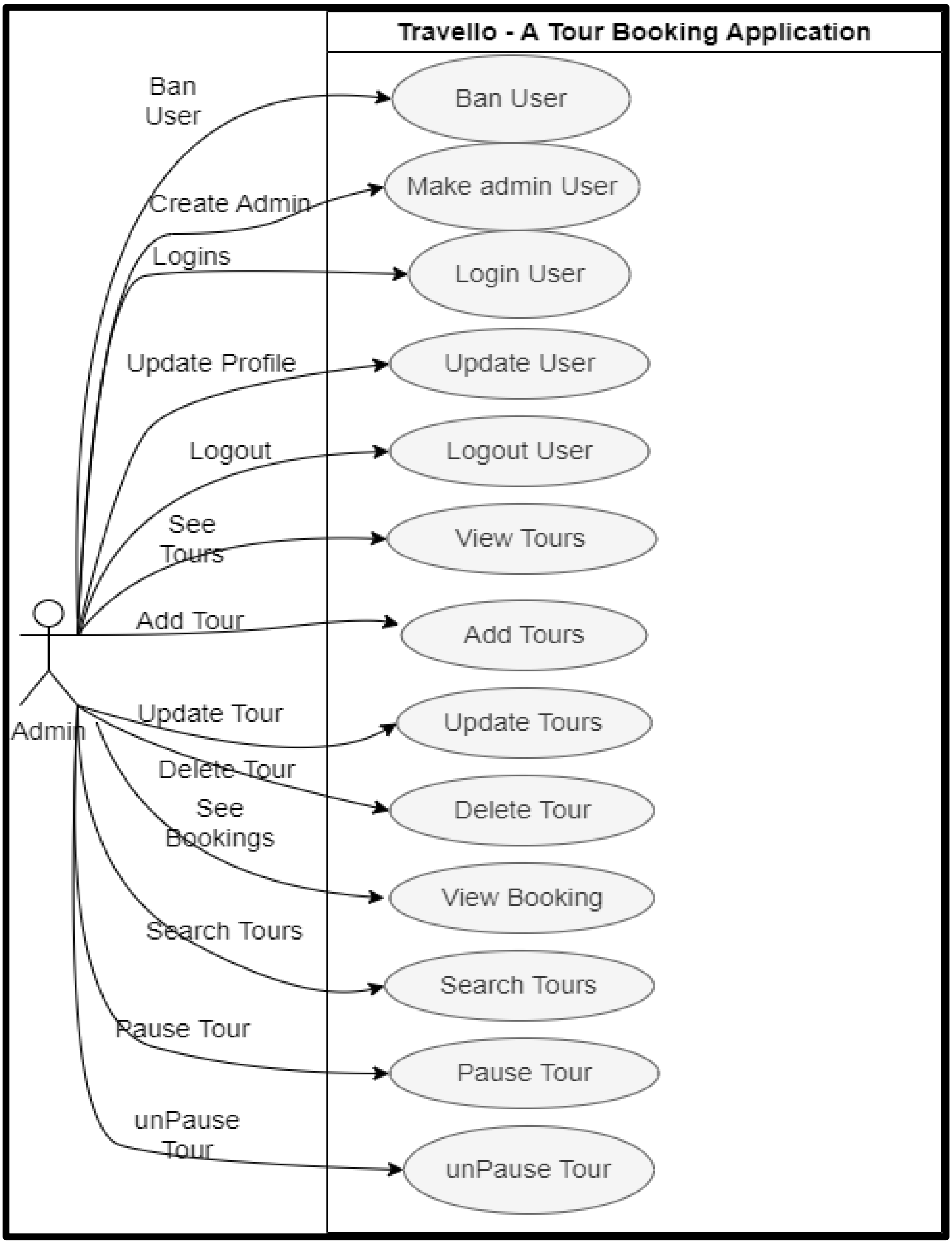


Figure 3.7 Use Case Diagram (Admin User)

3.8 Sequence Diagram – Normal User:- A sequence diagram for a normal user would depict their interaction with the system in a chronological order. It typically starts with the user initiating an action, such as logging in or booking a tour, followed by the system's response or processing of the request.

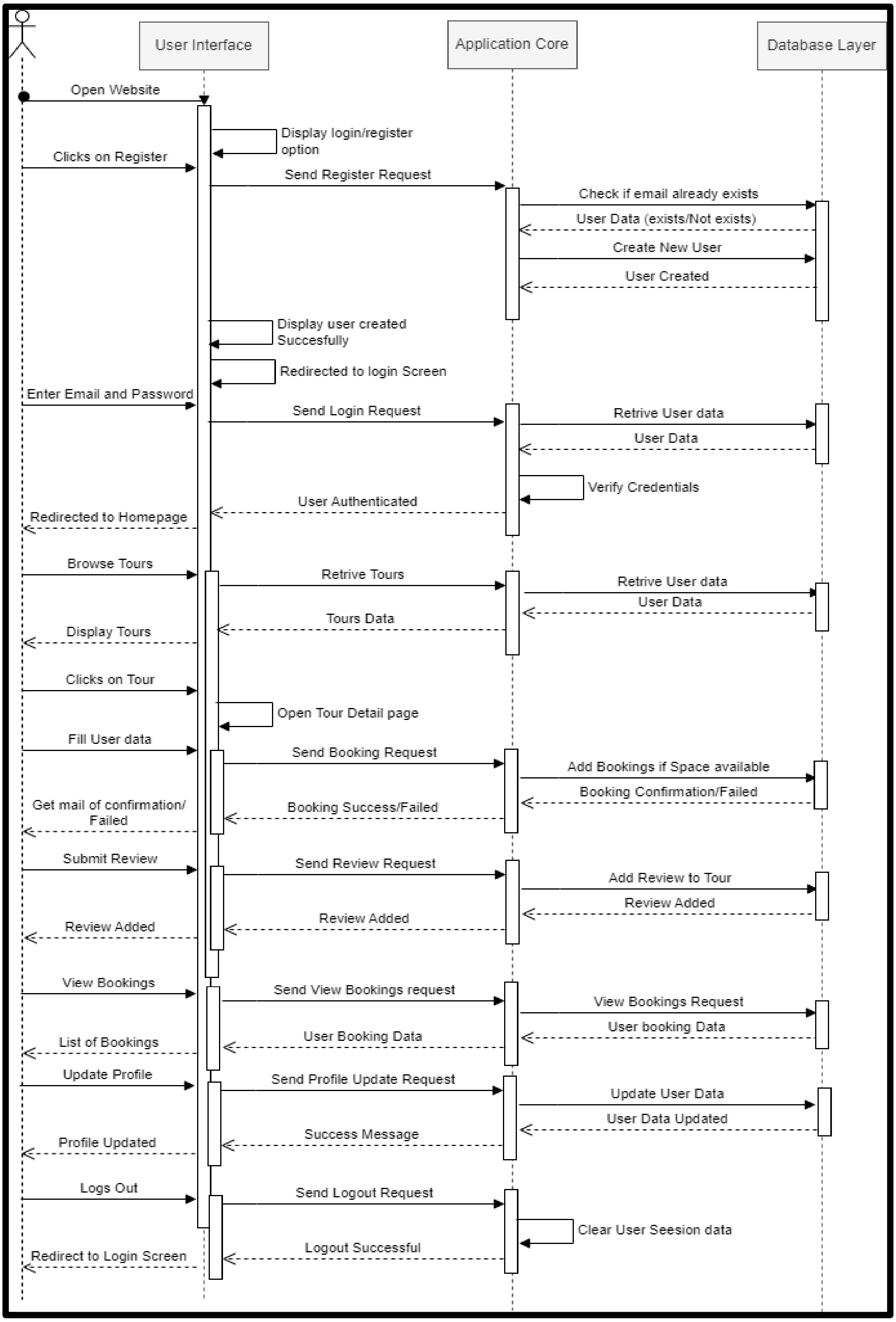


Figure 3.8 Sequence Diagram (Normal User)

3.9 Sequence Diagram –Admin User:- A sequence diagram for an admin user would similarly illustrate their interaction with the system. It begins with the admin initiating an action, such as logging in or managing tours, followed by the system's response or execution of the requested task.

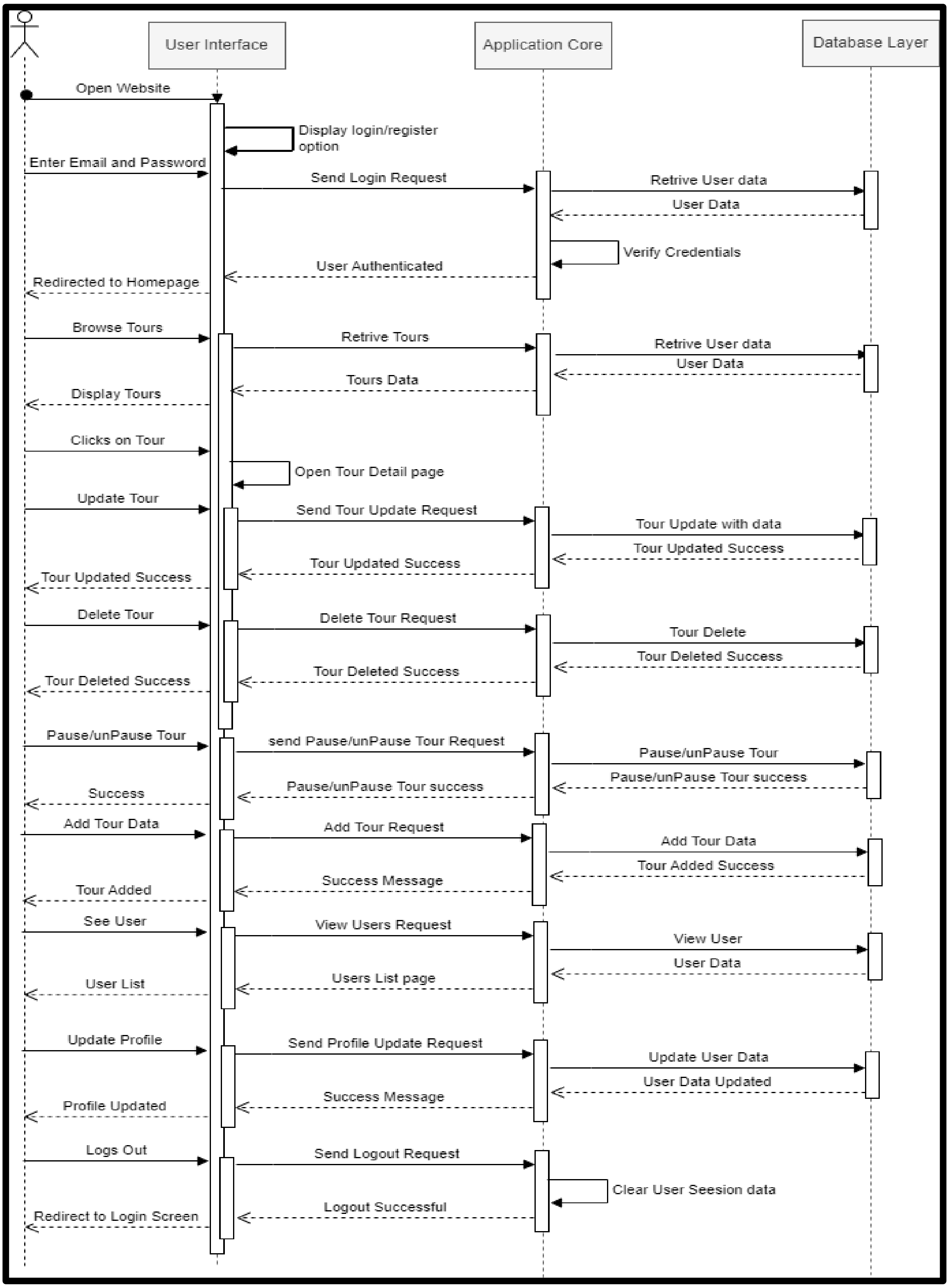


Figure 3.9 Sequence Diagram (Admin User)

3.10 Activity Diagram:- The activity diagram visually represents the flow of activities within the system, illustrating the sequence of actions performed by users or admins and the corresponding system responses.

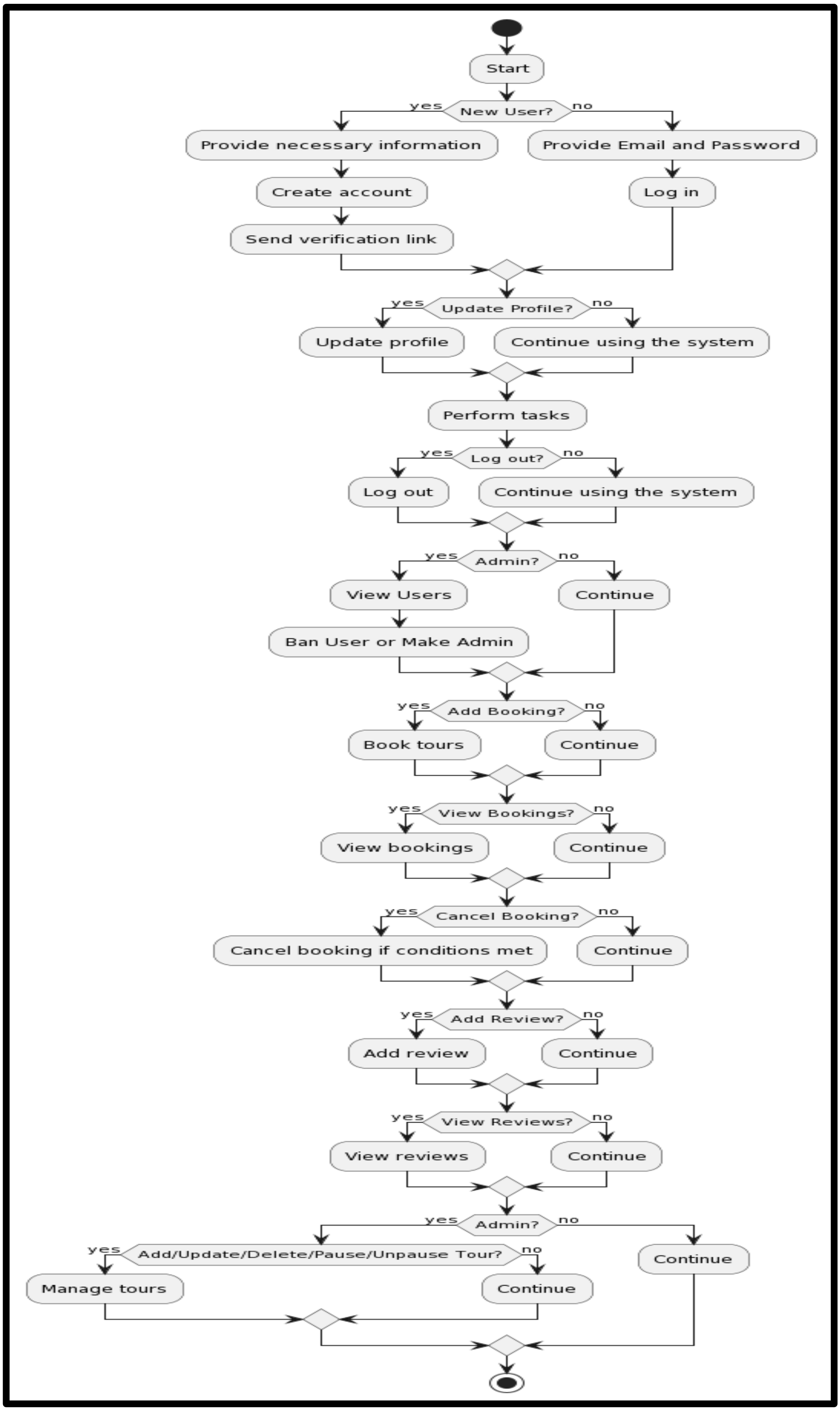


Figure .3.10 Activity Diagram

Chapter 4

Design

4.1 Data Dictionary:

Users

Field Name	Data Type	Size	Description	Example
_id	ObjectId		Unique user identifier	6582e5be605933c7133da5c8
Email	String	255	User's email address	vyom1.raa02@gmail.com
username	String	20	User's username	vyom1
password	String	255	User's hashed password	$2a$10$wQLwK4yweICRu/7.yEP7ceTcUt4XXyDxwiUurdh.XtmdZuni7Mpza
Role	String	255	Is User normal user or Admin	Admin
isBan	Bool	1	Is User banned	False

Table 4.1 User Data Dictionary

Reviews

Field Name	Data Type	Size	Description	Example
_id	ObjectId		Unique user identifier	658e42e4df9dd1a30e8cc837
reviewText	String	255	User's review for tour	user
username	String	20	User's username	nice
Rating	Integer	4	Rating star for Tour	4

Table 4.2 Reviews Data Dictionary

Bookings

Field Name	Data Type	Size	Description	Example
_id	ObjectId		Unique user identifier	65939e0ffd2411b03023d7b8
userID	String	255	User's ID of who book tour.	658eed1a3f4f388f6072e2a2
userEmail	String	20	User's Email	vyom.rana02@gmail.com
Tour	String	255	Booking of Tour	Beautiful Snowy Mountains
Full Name	String	255	Name of User	Vyom Rana
GuestSize	Integer	4	Nos of Guests	2
Phone	String	255	Phone Number of User	8490066177
BookAt	Date	20	Booking Date	2024-05-09
Totals	Integer	4	Total price of Booking.	20000
isCancelled	Bool	1	Is Tour Canceled.	False

Table 4.3 Bookings Data Dictionary

Tours

Field Name	Data Type	Size	Description	Example
_id	ObjectId		Unique user identifier	65939e0ffd2411b03023d7b8
Title	String	255	Tour Name	Srinagar
City	String	255	City name	Srinagar
Distance	Integer	4	Max travelling distance	300
Duration	String	255	Duration of Tour	3N/4D
Price	Integer	4	Price of Tour per Guest	2000
MaxGroup Size	Integer	4	No. persons allowed at a time	5
Description	String	255	Description of Tour	Srinagar Tourism ….
Reviews	Array	-	Reviews of Tour	[‘nice tour’]
Photo	String	255	Is Tour Canceled.	/uploads/1705674363464_Srinanagar.jpg
Featured	Bool	1	Feature tour or not	True
Paused	Bool	1	Paused tour or not	False

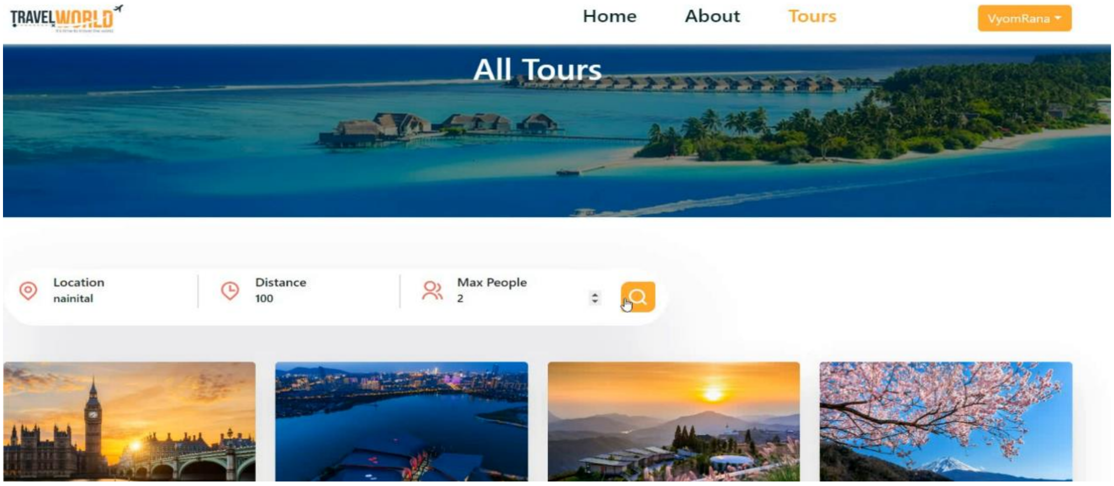
Table 4.4 Tours Data Dictionary

4.2 Validation

- Email for registration should not matched with existing user’s email in Database.
- Password should contain atleast 8 characters, atleast 1upper case, atleast 1 lowercase, atleast 1 number , atleast 1 symbol.
- User Register and can Login only if User has verified the email link.
- User can Review Tour only if user has already booked and experience the Tour.
- User can’t Book Tour which has more guestsize than maxGroupSize.
- User can’t Book overlapping Tour which was previously book by user.
- User can Book Tour in Future date than todays Date.
- User can cancel booking if the difference between today’s Date and Booked Date is more than 1 day.

4.3 Frontend Interface:-

Tours Interface:- This is Search tour interface for searching tours using location, distance and max no. of people.



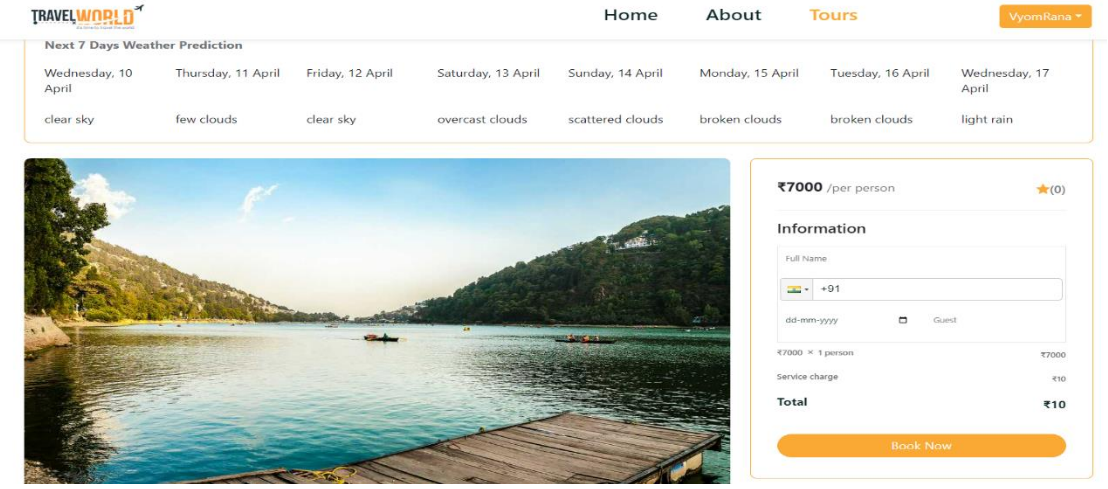
Location
nainital

Distance
100

Max People
2

Figure 4.1 Tour Interface

Tour Detail Interface:- This is Tour Detail page where weather predictions of 7 days and tour description is seen.



Next 7 Days Weather Prediction

Wednesday, 10 April	Thursday, 11 April	Friday, 12 April	Saturday, 13 April	Sunday, 14 April	Monday, 15 April	Tuesday, 16 April	Wednesday, 17 April
clear sky	few clouds	clear sky	overcast clouds	scattered clouds	broken clouds	broken clouds	light rain

₹7000 /per person ★(0)

Information

Full Name

+91

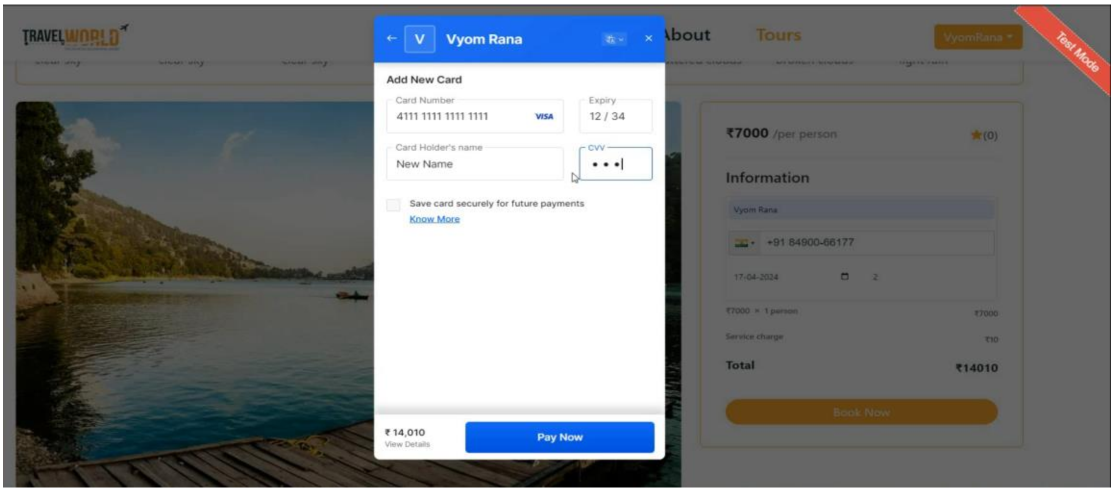
dd-mm-yyyy Guest

₹7000 × 1 person	₹7000
Service charge	₹10
Total	**₹10**

Book Now

Figure 4.2 Tour Detail Interface

Payment Interface:- This is Payment Interface for razorpay where user while booking can add their payment details.



Vyom Rana

Add New Card

Card Number: 4111 1111 1111 1111 VISA

Expiry: 12 / 34

Card Holder's name: New Name

CVV: •••

□ Save card securely for future payments

Know More

₹ 14,010
View Details

Pay Now

₹7000 /per person ★(0)

Information

Vyom Rana

+91 84900-66177

17-04-2024 2

₹7000 × 1 person ₹7000

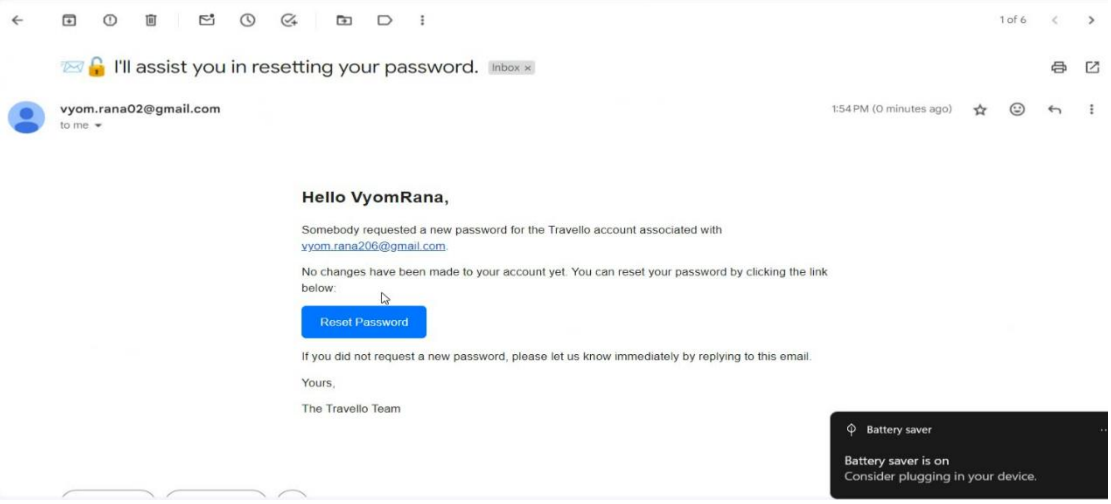
Service charge ₹10

Total ₹14010

Book Now

Figure 4.3 Payment Interface

Reset Password Mail:- This is mail interface when user tries to reset the password.



📨🔓 I'll assist you in resetting your password.

vyom.rana02@gmail.com
to me

Hello VyomRana,

Somebody requested a new password for the Travello account associated with vyom.rana206@gmail.com.

No changes have been made to your account yet. You can reset your password by clicking the link below:

Reset Password

If you did not request a new password, please let us know immediately by replying to this email.

Yours,

The Travello Team

Battery saver

Battery saver is on
Consider plugging in your device.

Figure 4.4 Password Reset Mail

4.4 Application Navigation

- **/login:-** Login page.
- **/register:-** Register Page.
- **/forgot-password:-** Forgot password page.
- / :- Redirected to the /home.
- **/home:** Entry point for users.
- **/tours:** Navigate to Tours page.
- **/tours/:id:** Navigate to particular tour page ie Tour Detail Page.
- **/tour/search:**- Seach page of tours.
- **tour/updateTour:**- Page to Update particular tour.
- **/about:**- About page of Travello App.
- **/addTour:**- Page to add tours.
- **/allUser:-** Page to see all user list.
- **/thank-you:**- Page render if tour booked Succesfully.
- **/updateUser:**- Page to Update user Details.
- **/bookings :**- Page to see list of bookings.
- **/logout:-** User logout and redirected to home page.

Chapter 5

Implementation

5.1 Modules:

User Module:

- Login Account: Easily access the application by logging in with your email and password.
- View Profile: See your profile details, including username, email, and more.
- Update Profile: Keep your information up to date by editing your profile settings.
- Change Password: Enhance security by changing your account password whenever needed.
- View Users: Admins can manage users, view their profiles, and take actions like banning/unbanning.
- Logout: Securely end your session with the application to protect your account.

Review Module:

- Post Review: Share your experiences by posting reviews of tours, including ratings and comments.
- View Reviews: Explore reviews shared by other users to make informed decisions about tours.

Booking Module:

- Add Booking: Easily book a tour by providing your details, booking date, and payment information.
- Cancel Booking: Cancel your booking hassle-free if plans change.
- View Bookings: Review all your bookings in one place, or admins can manage all bookings across users.

Tours Module:

- Add Tours: Admins can enrich the Travello app by adding new tours with details like photos, descriptions, and prices.
- Delete Tour: Remove outdated or irrelevant tours from the system.
- Update Tour: Keep tour information accurate and relevant by updating details as needed.
- Pause/Unpause Tour: Temporarily halt or resume tours based on availability or other factors.
- View Tour: Users can browse and explore all available tours conveniently.

5.2 Implementation Details:

User Module:

Login Account:

- Implement authentication logic using email and password.
- Use bcrypt for securely storing and comparing passwords.
- Utilize sessions or JSON Web Tokens (JWT) for managing user sessions.

View Profile:

- Fetch user details from the database based on the logged-in user's ID.
- Display the profile information on the user interface.

Update Profile:

- Implement an endpoint for updating user profile information.
- Validate user input to prevent security vulnerabilities.
- Update the user's information in the database.

Change Password:

- Provide a form for users to input their old and new passwords.
- Implement validation to ensure the new password meets security requirements.
- Update the user's password in the database after validation.

View Users:

- Create an endpoint accessible only to admin users to retrieve a list of all users.
- Include functionality for banning/unbanning users based on admin actions.

Logout:

- Implement a logout endpoint to destroy the user session.
- Redirect the user to the login page after successful logout.

Review Module:

Post Review:

- Create a form for users to submit their reviews, including ratings and comments.
- Validate user input to ensure the review meets requirements.
- Save the review details in the database associated with the user and the tour.

View Reviews:

- Implement an endpoint to fetch reviews for a specific tour or user.
- Display reviews with ratings and comments on the tour details page.

Booking Module:

Add Booking:

- Develop a booking form for users to input their details, booking date, and payment information.
- Validate and process payment using a payment gateway (e.g., Razorpay).
- Save the booking details in the database associated with the user and tour.

Cancel Booking:

- Implement an endpoint for users to cancel their bookings.
- Validate the cancellation request and update the booking status in the database accordingly.

View Bookings:

- Create endpoints for users and admins to retrieve booking details.
- Display user-specific bookings on the user profile page and all bookings on the admin dashboard.

Tours Module:

Add Tours:

- Develop a form for admins to input tour details, including photos, descriptions, and prices.
- Validate user input to prevent malicious data.
- Save the tour details in the database.

Delete Tour:

- Implement an endpoint for admins to delete tours.
- Ensure proper authorization to prevent unauthorized deletion of tours.

Update Tour:

- Create a form for admins to edit tour details.
- Validate and process the updated information, then save it in the database.

Pause/Unpause Tour:

- Implement endpoints for admins to pause/unpause tours.
- Update the tour status in the database accordingly.

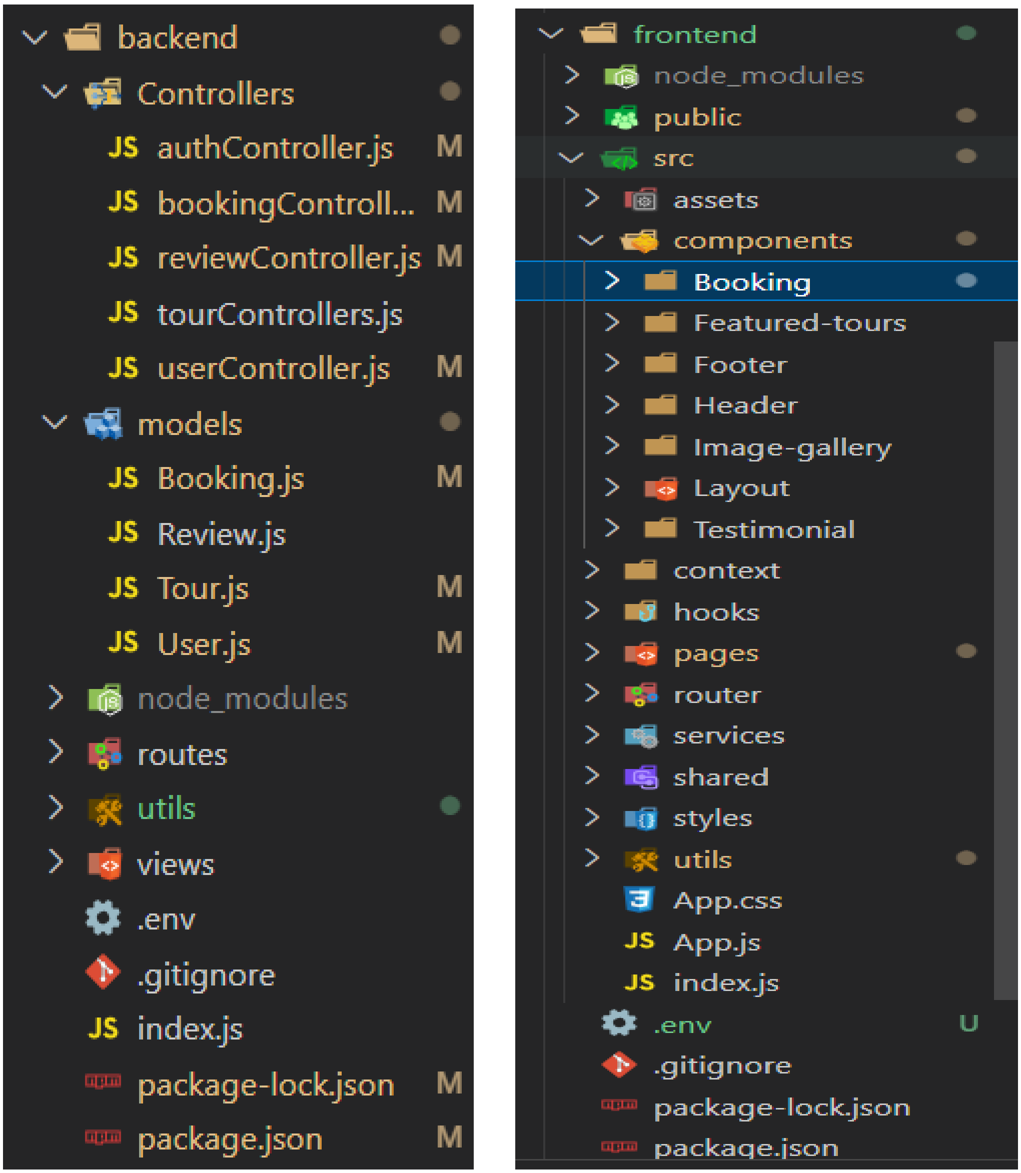
View Tour:

- Implement endpoints to retrieve tour details for users to browse.
- Display tour information on the tour listing page.

5.3 Directory Structure:-

The main modules that are implemented in this website are listed below.

- User Module
- Review Module
- Tour Module
- Booking Module



```
backend
  Controllers
    authController.js        M
    bookingControll...       M
    reviewController.js      M
    tourControllers.js
    userController.js        M
  models
    Booking.js               M
    Review.js
    Tour.js                  M
    User.js                  M
  node_modules
  routes
  utils
  views
  .env
  .gitignore
  index.js
  package-lock.json          M
  package.json               M
```

```
frontend
  node_modules
  public
  src
    assets
    components
      Booking
      Featured-tours
      Footer
      Header
      Image-gallery
      Layout
      Testimonial
    context
    hooks
    pages
    router
    services
    shared
    styles
    utils
    App.css
    App.js
    index.js
  .env                       U
  .gitignore
  package-lock.json
  package.json
```

Figure 5.1 Directory Structure

Chapter 6

Testing

Testing is a process of executing a program with the intent of finding bugs that makes the application fail to meet the expected behavior. Regardless of the development methodology.

Testing plays a critical role for assuring quality and reliability of the software. I have included testing as a part of the development process. The test cases should be designed with maximum possibilities of finding the errors or bugs. Various levels of testing are as follows.

- **Unit testing:** Unit testing tests the functionality of individual units of source code. It is the smallest component of a testable software that works in isolation with other parts of the code. I have done unit testing for various individual components of the source code to uncover errors within the boundary of the application.

- **Integration testing:** Integration testing focuses on the design and construction of the software. Here the individual components that are tested using unit tests are combined and tested as a group. Its primary purpose is to expose the defects associated with the interfacing of modules. It checks if the modules perform the desired functionality when integrated together.

- **System testing:** System testing is performed on a completely integrated system to see if it meets the requirements.

- **Regression testing:** Regression testing aims at verifying the functionality of the software that is previously tested and to which changes are made. It is to ensure the old software still works with new changes.

- **Acceptance testing:** Acceptance testing is conducted to verify if the system compliance the business requirements. Adhering to the levels of testing, Unit testing is performed on individual components of the system ensuring the expected behavior. Later, I integrated various components together and performed Integration testing. Once the integration testing is done, I have performed System testing and ensured the application works as per the requirements. Finally, acceptance testing is performed to check if the client accepts the system.

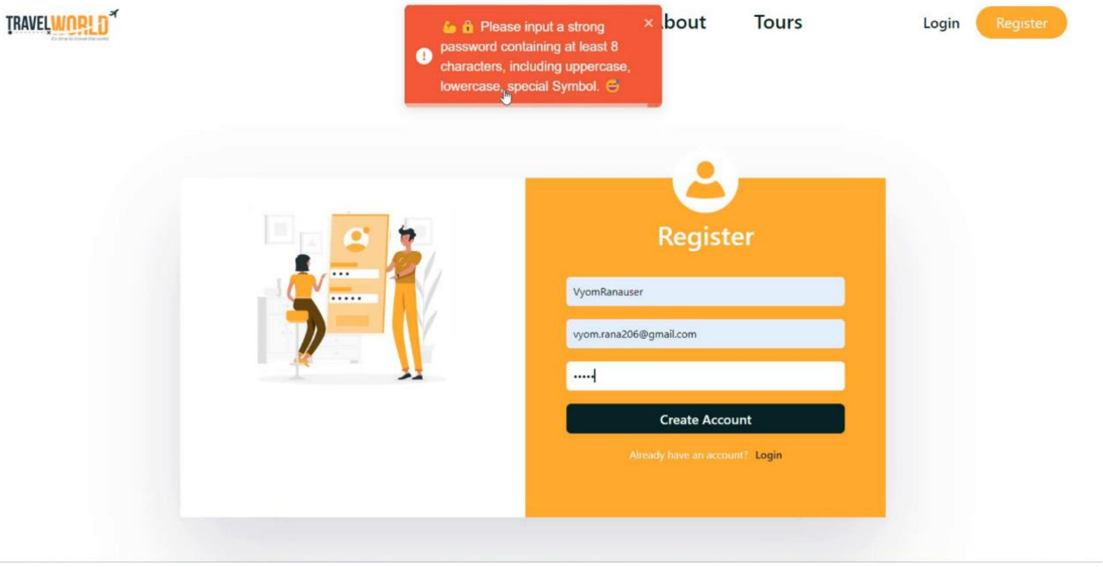
6.1 Manual Testing

_Test Case Id	SRS Id	Test Case Objective	Input Data	Expected Output	Actual Output	Status
TC_01	R.1	Authentication of User Credentials	Username, Password	Success message for valid credentials	Success message for valid credentials	Pass
TC_02	R.1	Authentication of User Credentials	Username, Password	Error message for invalid credentials	Error message for invalid credentials	Pass
TC_03	R.1	Password Validation	Weak Password	Error message indicating password strength	Error message indicating password strength	Pass
TC_04	R.1	Display User Profile Information	User Selection - Home Menu	User's profile information displayed	User's profile information displayed	Pass
TC_05	R.1	Update User Profile Information	User details for the profile	Details Updated successfully	Details Updated successfully	Pass
TC_06	R.1	Forgot Password	2 same passwords	Change password Succcesfully	Change password Succcesfully	Pass
TC_07	R.2	Add review	User Selection and message	Review added success	Review added success	Pass
TC_08	R.2	Add review with no booking	User Selection and message	Error adding review to the tour	Error adding review to the tour	Pass
TC_09	R.3	Add Bookings	User Selection, and data	User booking for tour success	User booking for tour success	Pass
TC_10	R.3	Add overlapping bookings	User Selection, and data	User booking for tour failure	User booking for tour failure	Pass
TC_11	R.3	Add guestsize more bookings	User Selection, and data	User booking for tour failure	User booking for tour failure	Pass
TC_12	R.4	Add Tour	User Selection and data.	Tour added success	Tour added success	Pass
TC_13	R.4	Update Tour	User selection and data	Tour Updated Success	Tour Updated Success	Pass
TC_14	R.4	Delete Tour	User Selection and data	Tour deleted success	Tour deleted success	Pass

Table 6.1 Manual Testing

6.2 Screenshots:

Password Validator:- User has to select password with atleast 1 capital,1 small letter, 1 numeric ,1 symbol and alteast size of password should be 8.



💪 🔒 Please input a strong password containing at least 8 characters, including uppercase, lowercase, special Symbol. 😅

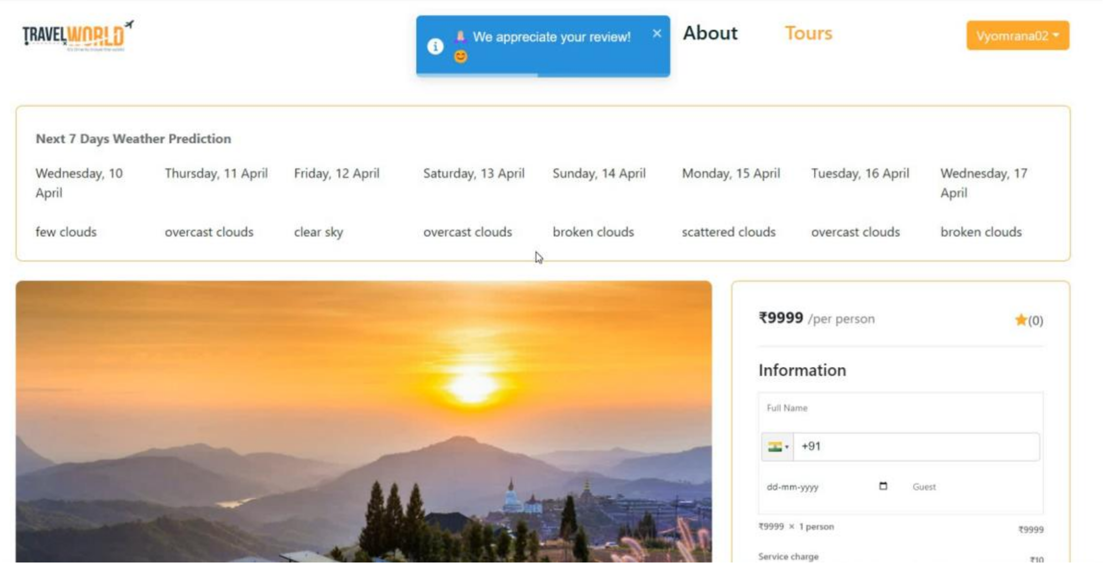
VyomRanauser

vyom.rana206@gmail.com

•••••

Figure 6.1 Password Testing[TC_03] [Pass]

Review Tour with booking:- User can review tour cause already experienced the tour.



👍 We appreciate your review! 😊

Next 7 Days Weather Prediction

Wednesday, 10 April	Thursday, 11 April	Friday, 12 April	Saturday, 13 April	Sunday, 14 April	Monday, 15 April	Tuesday, 16 April	Wednesday, 17 April
few clouds	overcast clouds	clear sky	overcast clouds	broken clouds	scattered clouds	overcast clouds	broken clouds

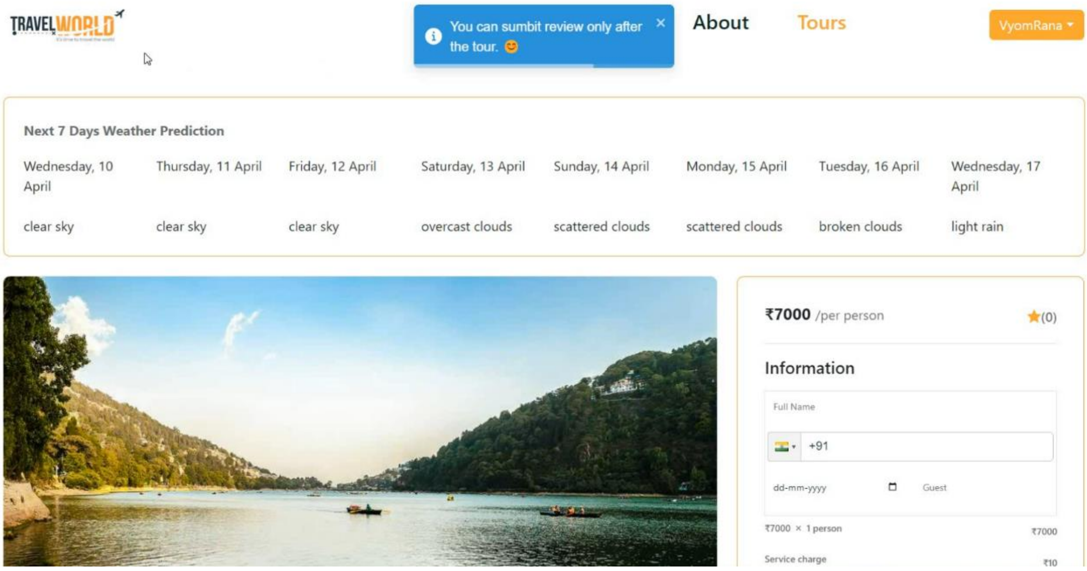
₹9999 /per person ★(0)

₹9999 × 1 person ₹9999

Service charge ₹10

Figure 6.2 Review Testing with booking[TC_07] [Pass]

Review Tour without booking:- User can review tour if user has already experienced the tour.



You can sumbit review only after the tour. 😊

Next 7 Days Weather Prediction

Wednesday, 10 April	Thursday, 11 April	Friday, 12 April	Saturday, 13 April	Sunday, 14 April	Monday, 15 April	Tuesday, 16 April	Wednesday, 17 April
clear sky	clear sky	clear sky	overcast clouds	scattered clouds	scattered clouds	broken clouds	light rain

₹7000 /per person ★(0)

Information

Full Name

+91

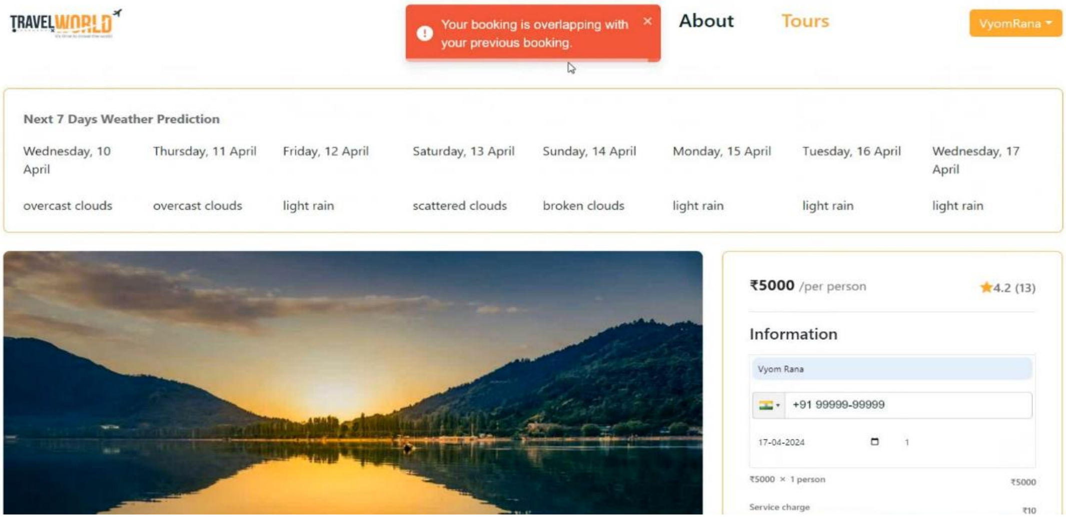
dd-mm-yyyy Guest

₹7000 × 1 person ₹7000

Service charge ₹10

Figure 6.3 Review Testing without booking [TC_08] [Pass]

Overlapping Booking:- User cannot add booking which is overlapping with any of previous booking.



Your booking is overlapping with your previous booking.

Next 7 Days Weather Prediction

Wednesday, 10 April	Thursday, 11 April	Friday, 12 April	Saturday, 13 April	Sunday, 14 April	Monday, 15 April	Tuesday, 16 April	Wednesday, 17 April
overcast clouds	overcast clouds	light rain	scattered clouds	broken clouds	light rain	light rain	light rain

₹5000 /per person ★4.2 (13)

Information

Vyom Rana

+91 99999-99999

17-04-2024 1

₹5000 × 1 person ₹5000

Service charge ₹10

Figure 6.4 Overlapping Booking Testing[TC_10] [Pass]

Chapter 7

Conclusion and Future Extension

7.1 Conclusion:

In the journey of developing the travel management application, I have encountered numerous challenges, explored various design considerations, and implemented intricate functionalities to meet the diverse needs of our users. As we conclude this project, it's essential to reflect on our accomplishments, challenges faced, and the implications of our work.

Accomplishments:

Functional Implementation: I have implemented all modules outlined in the Software Requirement Specification (SRS), including User Management, Review System, Booking Management, and Tour Administration, ensuring that our application fulfills the core requirements of our users.

Robust Security Measures: Security has been a top priority in our application development. I have incorporated industry-standard authentication mechanisms, data validation techniques, and encryption protocols to safeguard user data and ensure secure interactions within the platform.

Scalable Architecture: Our application architecture has been designed with scalability in mind, allowing us to accommodate future growth and handle increased user traffic effectively. By using scalable technologies and best practices, we have built a platform capable of supporting a growing user base and expanding
feature set.

User-Centric Design: User experience (UX) has been at the forefront of our design considerations. I have crafted an pleasing user interface, prioritizing ease of navigation, accessibility, and seamless interaction flows to enhance user satisfaction and engagement.

Challenges Faced: Despite our achievements, we encountered several challenges during the development process:

Complex Business Logic: Implementing business logic and workflow processes, such as tour booking and review management, imposed challenges in terms of system design, data modeling, and code complexity. Balancing functionality and maintainability while adhering to project timelines required careful planning and collaboration.

Integration Complexity: Integrating various third-party services, such as payment gateways and mapping APIs, introduced complexity in system integration and dependency management. Ensuring seamless communication between different system components while maintaining data integrity and security was a significant challenge.

Performance Optimization: As the application grows and user traffic increases, optimizing performance becomes crucial. Identifying and mitigating performance bottlenecks, optimizing database queries were ongoing challenges that required continuous monitoring and optimization efforts.

7.2 Future Extension:

- API Integrations: Connect with flight booking, hotel reservation, and transportation services for seamless travel planning.
- Personalized Recommendations: Use machine learning for tailored tour suggestions and targeted promotions.
- Social Sharing: Enable sharing travel experiences on social media and collaborative trip planning.
- Loyalty Program: Reward users with discounts and perks for repeat bookings.
- Accessibility Features: Ensure accessibility with screen reader compatibility and high contrast modes.
- Multi-language Support: Provide localized interfaces for global users.
- Community Forums: Create a platform for user-generated content and travel advice.

Bibliography

[1] ReactJs official documentation. [Online]
https://legacy.reactjs.org/tutorial/tutorial.html
[2] Node package manager. [Online]
https://www.npmjs.com/
[3] Bootstrap 5.3 official documentation. [Online]
https://getbootstrap.com/docs/5.3/getting-started/introduction/
[4] MongoDB 7.0 official documentation. [Online]
https://www.mongodb.com/
[5] RazorPay official documentation. [Online]
https://dashboard.razorpay.com/
[6] JWT official documentation [Online]
https://jwt.io/
[7] NPM official documentation [Online]
https://www.npmjs.com/
[8] Express JS 4.0 documentation [Online]
https://expressjs.com/en/4x/api.html