Explore Hub

Mini Project Report

Submitted by

Kiran Kurian Philip

Reg. No.: AJC23MCA-2041

In Partial fulfilment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS (MCA)

AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS

KANJIRAPPALLY

[Approved by AICTE, accredited by NAAC, Accredited by NBA.
Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2024-2025

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, “**EXPLORE HUB”** is the bonafide work of **KIRAN KURIAN PHILIP (Regno: AJC23MCA-2041)** in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications under **Amal Jyothi College of Engineering Autonomous, Kanjirappally** during the year 2024-25.

Ms. Nimmy Franics
Internal Guide

Mr. Binumon Joseph
Coordinator

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

DECLARATION

I hereby declare that the project report **“EXPLORE HUB”** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from Amal Jyothi College of Engineering Autonomous during the academic year 2024-2025.

Date:	**KIRAN KURIAN PHILIP**
KANJIRAPPALLY	**Reg: AJC23MCA-2041**

ACKNOWLEDGEMENT

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

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

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

KIRAN KURIAN PHILIP

ABSTRACT

Explore Hub, built with Django, connects travel enthusiasts and facilitates seamless travel planning and group formation. Travel agencies can list their packages, which must be admin-approved. Users can create travel groups based on destinations and dates, with each group limited to 10 members, and use real-time chat for planning. Future phases will include personalized itineraries, and machine learning for personalized recommendations, dynamic suggestions, advanced search with filter options and popularity predictions. Roles for Local Guides and Event Organizers will also be added to enhance the travel experience.

Users Functions

- Admin:
    - Approve Travel Agency Accounts: Admins can approve or reject travel agency accounts, controlling which agencies can list their packages on the platform.
    - Manage Listings and Users: Admins can oversee and manage all travel packages, user accounts, and travel groups.
- Travel agencies:
    - Create and Manage Accounts: Agencies can sign up and create profiles, which need admin approval before activation.
    - List Travel Packages: Once approved, agencies can list travel packages, providing details such as destination, itinerary, cost, etc.
    - Update and Manage Listings: Agencies can edit or delete their travel packages as needed.
- Regular users:
    - Sign Up and Profile Management: Users can sign up, log in, and manage their profiles.
    - Browse and Search Travel Packages: Users can view and search for travel packages listed by various travel agencies.
    - Create Travel Groups: Users can create travel groups by selecting a destination, setting preferred travel dates, and providing a group description.
    - Join Travel Groups: Users can browse available travel groups and join those that match their interests and travel plans.
    - Group Communication: Users can engage in real-time chat with other members of their travel group to discuss plans and coordinate activities.
    - Manage Group Membership: Users can leave groups they no longer wish to participate in.

CONTENT

List of Abbreviations

- AJAX - Asynchronous JavaScript and XML
- CSS - Cascading Style Sheets
- DBMS - Database Management System
- Django - A high-level Python web framework
- HTML - Hypertext Markup Language
- JS - JavaScript
- MySQL - My Structured Query Language
- PK - Primary Key

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Explore Hub is a web-based platform designed to facilitate seamless interaction between travel enthusiasts and travel agencies. Users can explore available travel packages, create and join group trips, and easily book their preferred travel options.

The increasing trend of travel enthusiasts seeking flexible group travel experiences and customized travel plans motivated the development of Explore Hub. The platform aims to simplify travel planning and bookings while offering a collaborative way for users to form travel groups.

The platform is designed for travellers who wish to book unique travel packages, form travel groups, and explore new destinations, as well as for travel agencies that want to showcase their packages to a broader audience.

1.2 PROJECT SPECIFICATION

Explore Hub is a web-based platform designed to simplify the process of discovering and booking travel packages while fostering collaborative travel planning through group features. The system offers role-based access, allowing both Regular Users (travellers) and Travel Agents to interact with the platform according to their specific needs.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The System Study phase of Explore Hub involves a thorough analysis of the current system, identifying its limitations and understanding user needs. This phase is essential for evaluating how the new system will improve upon existing solutions, streamlining the travel experience for users and travel agencies.

By examining both the existing system and the proposed system, this study provides a foundation for the platform's design and development. It highlights the inefficiencies of current travel booking processes and demonstrates how Explore Hub addresses these challenges with an intuitive, feature-rich solution.

2.2 EXISTING SYSTEM

2.2.1 NATURAL SYSTEM STUDIED

The existing travel system often operates as a natural system, relying heavily on informal processes and human interactions for trip planning and booking. This method involves significant manual research, where individuals browse various travel blogs, forums, and review sites, leading to information overload. The booking process is often fragmented, as travellers must contact multiple service providers directly to arrange accommodations, transportation, and activities, creating opportunities for miscommunication and confusion. Coordinating group travel can be particularly chaotic, as discussions take place across various messaging platforms, making it challenging to track preferences and commitments. Additionally, this system lacks transparency, with hidden costs and unclear terms often leading to dissatisfaction. Explore Hub aims to address these challenges by providing a cohesive platform for users to explore and book travel options seamlessly.

2.2.2 DESIGNED SYSTEM STUDIED

The proposed system, Explore Hub, is designed to revolutionize the travel booking experience by integrating various functionalities into a cohesive digital platform. This user-centric system enables travellers to seamlessly explore and book a diverse range of travel packages offered by approved travel agencies. With features like group travel planning, users can create or join groups based on shared interests, facilitating collaborative trip organization and cost-sharing. Explore Hub aims to deliver a comprehensive and satisfying travel experience that caters to both individual travellers and travel agencies.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Fragmented Information: Users must visit multiple sites for travel information, leading to confusion.
- Inefficient Booking: Many systems rely on manual processes, causing delays and errors in reservations.
- Limited Group Travel Options: Few platforms allow users to create or join travel groups, missing collaborative opportunities.
- Poor User Experience: Outdated interfaces make it hard for users to navigate and find what they need.
- Lack of Support: Users struggle to get timely help due to inadequate communication channels.
- Hidden Fees: Pricing is often unclear, leading to mistrust and dissatisfaction among travellers.
- Chaotic Group Planning: Organizing trips with friends or family can be messy, as discussions occur across different platforms.

2.4 PROPOSED SYSTEM

The proposed system, Explore Hub, is an innovative travel platform designed to streamline the entire travel experience for users and travel agencies alike. This user-friendly system allows travellers to easily explore and book a wide variety of travel packages from approved agencies, ensuring a diverse selection of options. One of its standout features is the ability for users to create or join travel groups, facilitating collaborative trip planning and shared experiences. The platform boasts an intuitive interface that simplifies package comparisons, making it easy for users to find the best deals. With secure payment integration through Razorpay, users can complete transactions confidently, receiving immediate PDF invoices for their bookings. Additionally, travel agents benefit from a dedicated dashboard that enables them to efficiently manage their packages, offers, and bookings. Explore Hub aims to provide a comprehensive and enjoyable travel planning experience, fostering connections between travellers and travel agencies.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- User-Friendly Interface: Easy navigation makes it simple for users to find and book travel packages.
- Diverse Travel Options: Users can explore a wide range of travel packages from different agencies in one place.
- Group Travel Feature: Users can create or join travel groups, making it easier to plan trips together.

- Secure Payments: Integrated payment options ensure safe and reliable transactions.
- Instant Invoices: Users receive immediate PDF invoices after booking, enhancing transparency.
- Agent Dashboard: Travel agents can easily manage their packages, offers, and bookings in one convenient location.
- Streamlined Booking Process: The system simplifies the booking process, reducing errors and delays.
- Enhanced Communication: Built-in support channels facilitate better communication between travellers and agents.
- Improved Transparency: Clear pricing and terms help build trust and satisfaction among users.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

A detailed feasibility study was conducted to assess the practicality and success of the Explore Hub project from various perspectives, including economic, technical, and behavioral factors. This ensures that the project can be effectively developed and maintained.

3.1.1 Economical Feasibility

The economic feasibility evaluates the cost-effectiveness of the project. Since Explore Hub uses Django, an open-source framework, along with MySQL for the database and Razorpay for payment processing, the development costs are minimized. These technologies do not require licensing fees, and they offer high flexibility. Additionally, the cloud hosting costs are manageable. The system has the potential to generate revenue through features like paid group memberships or commissions on travel bookings, making it economically feasible for both short-term development and long-term maintenance.

3.1.2 Technical Feasibility

Technical feasibility assesses whether the system can be built with the available technology. Explore Hub is developed using the Django framework, which is known for its scalability, security, and built-in support for user authentication and data management. The database is managed with MySQL, which ensures efficient data handling for travel packages, users, and bookings. Django also integrates easily with the Razorpay payment gateway for secure transactions. The development team's familiarity with these technologies ensures that there are no significant technical obstacles, making the project technically feasible.

3.1.3 Behavioural Feasibility

Behavioural feasibility focuses on user and stakeholder acceptance. Explore Hub provides a user-friendly platform for both travellers and travel agencies. The ability to create travel groups, manage bookings, and communicate within groups adds value for users, while agencies benefit from the ability to list and manage packages. The platform's seamless integration with secure payment options (like Razorpay) builds trust among users. Given these features, it is likely to gain acceptance from the target audience, ensuring high adoption rates.

3.1.4 Feasibility Study Questionnaire

1. How do you currently promote your travel packages?

A: I mostly use social media and personal networks to promote my travel packages.

2. What challenges do you face in managing user inquiries about travel packages?

A: It's tough to handle all the inquiries because they usually come from different places like social media, email, and phone calls. It’s time-consuming to answer the same questions repeatedly.

3. How do you handle pricing and updates to your packages?

A: I manually update prices across different channels, and it’s a bit of a hassle to ensure everything is updated. If there’s a sudden change in availability or price, I have to go to multiple platforms to update it.

4. How do you currently manage payments for travel bookings?

A: Payments are done manually, usually via bank transfers or third-party apps.

5. Do you provide a service where users can form a group of their own and plan a trip themselves?

A: No, I don’t personally provide such a service. However, I think it’s a great idea. If users could form their own groups and plan trips together, it would give them more flexibility and save me the effort of organizing everything.

6. What difficulties do you face in communicating with customers before trips?

A: It’s hard to ensure that all customers get the necessary information before the trip, especially when I have to send the same details to multiple people through different channels.

7. How do you keep track of which travel packages are more popular?

A: Right now, I rely on how many inquiries or bookings I get to figure out which packages are doing well. I don’t have a formal way to track this.

8. What are the challenges in handling customer inquiries or changes after a booking is made?

A: It can get complicated when customers want to make changes after booking. They usually reach out by email or phone, and it’s easy to lose track of things.

9. How do you handle the itinerary and trip details for each travel package?

A: I usually provide the itinerary via email or WhatsApp once the booking is confirmed.

10. What features would make it easier for you to manage your travel agency on the platform?

A: A dashboard where I can manage all my packages, view bookings, handle payments, and communicate with customers would be ideal. If I could get automated notifications for bookings and have the ability to update package details in real time, that would save a lot of effort.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- intel i3 or above
RAM	- 4 GB
Hard disk	- 512 GB

3.2.2 Software Specification

Front End	-	HTML5, CSS, JavaScript
Back End	-	Django (Python)
Database	-	MySQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, CSS, Django, Razorpay

3.3 SOFTWARE DESCRIPTION

3.3.1 Django

Django is the primary backend framework used for building Explore Hub. It is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Django provides built-in tools for user authentication, database management, and security, making it an ideal choice for building scalable web applications. The framework also offers a robust admin interface, which simplifies the management of travel packages, user groups, and bookings. Django's Model-View-Template (MVT) architecture ensures separation of concerns, making the project easier to maintain and extend.

3.3.2 MySQL

MySQL is the relational database management system used to store and manage the project's data. It handles all data related to users, travel packages, groups, and bookings. MySQL's ability to efficiently manage large datasets makes it suitable for the Explore Hub platform, which involves a variety of user interactions, from searching for packages to joining groups and booking trips.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is an important phase in the software development life cycle (SDLC) and involves translating the requirements identified in the system analysis phase into a detailed design that can be implemented by developers.

The system design phase typically involves creating a detailed technical design document that outlines the overall architecture of the system and the individual components that make up the system. This document should include detailed descriptions of the data structures, algorithms, and interfaces that will be used in the system.

The goal of the system design phase is to create a blueprint for the development team that clearly defines how the system will be implemented and how it will interact with other systems and users. It should also identify any potential technical challenges or limitations that may need to be addressed during the development process.

Overall, the system design phase is critical to the success of a software development project as it lays the foundation for the development team to build and test the system in a structured and organized manner. It also provides a clear roadmap for the project stakeholders to follow and can help ensure that the final product meets the requirements and expectations of the users.

4.2 UML DIAGRAM

Unified Modeling Language (UML) is a graphical language used to visualize, specify, construct, and document the artifacts of a software system. UML diagrams are used to represent various aspects of a system's design, such as its structure, behavior, and interactions.

4.2.1 USE CASE DIAGRAM

A use case diagram is a type of behavioural diagram that describes the interactions between a system and its users or other systems. It helps identify the different use cases or scenarios in which the system will be used.

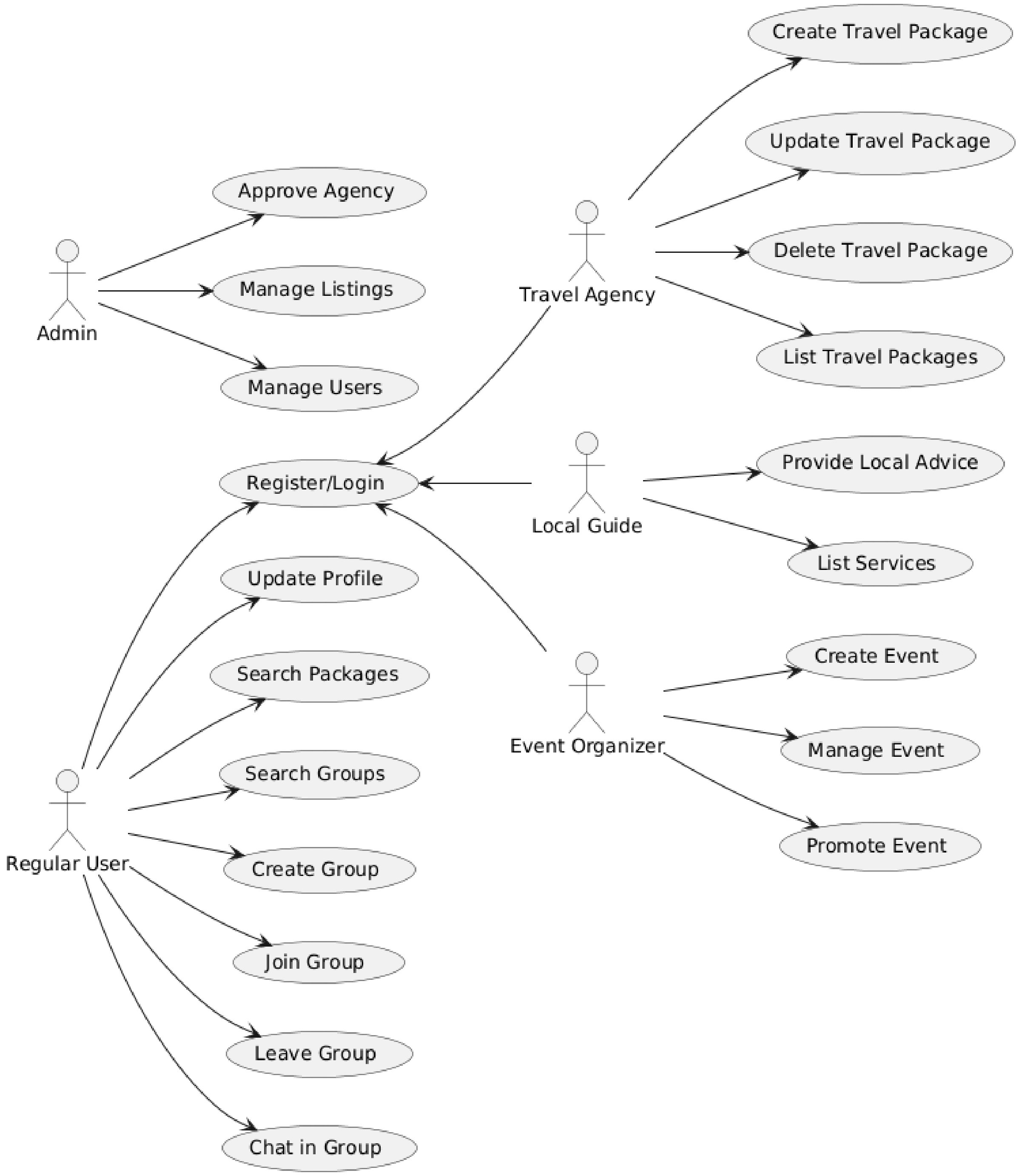


Fig 4.2. 1: Use Case Diagram

4.2.2 Sequence Diagram

A sequence diagram is a type of behavioural diagram that shows the interactions between objects or components in a system over time. It is used to model the dynamic behaviour of the system and can help identify any potential issues or errors.

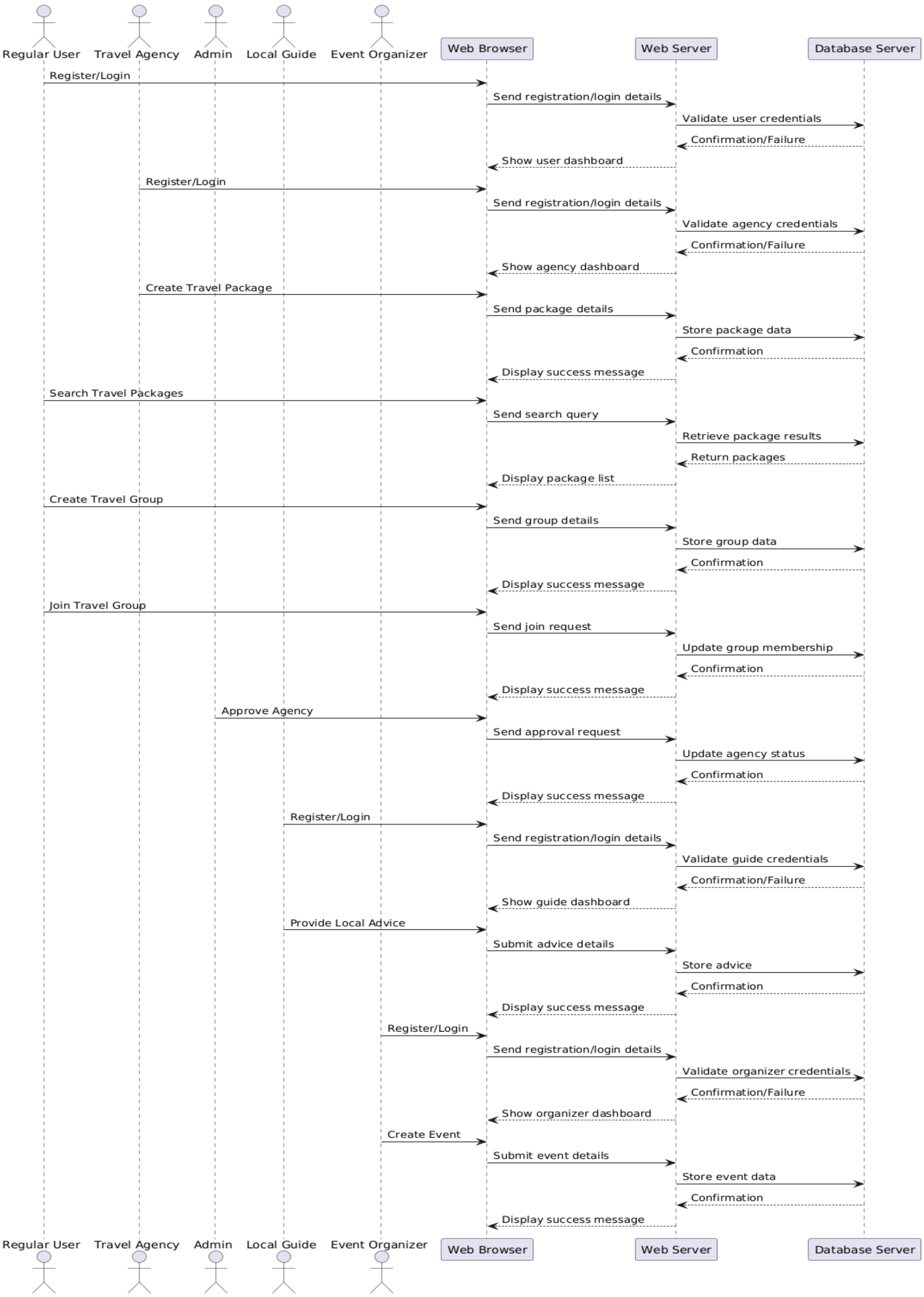


Fig 4.2. 2: Sequence Diagram

4.2.3 State Chart Diagram

A state chart diagram, also known as a state machine diagram or state transition diagram, is a type of behavioural diagram in the Unified Modelling Language (UML) used to model the behaviour of an object or system. It depicts the different states that an object can be in and the transitions between those states.

State chart diagrams consist of three main elements: states, transitions, and events. States represent the different conditions or modes in which an object or system can exist. Transitions represent the movement from one state to another, and events trigger those transitions.

In a state chart diagram, states are represented by circles or rounded rectangles, and transitions are represented by arrows. Events are usually represented by small circles or rectangles placed along the arrows. The diagram also includes labels that describe the states and transitions.

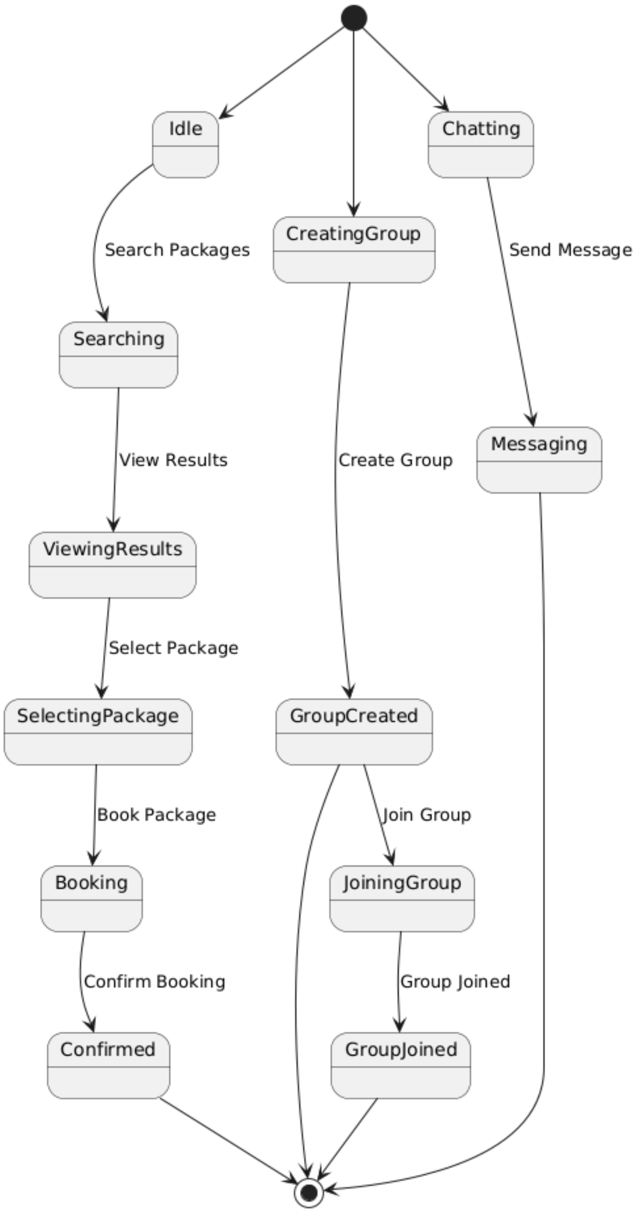


Fig 4.2. 3: State Chart Diagram

4.2.4 Activity Diagram

An activity diagram is a type of behavioural diagram that shows the flow of activities or processes in a system. It is used to model the workflows and can help identify any potential bottlenecks or inefficiencies.

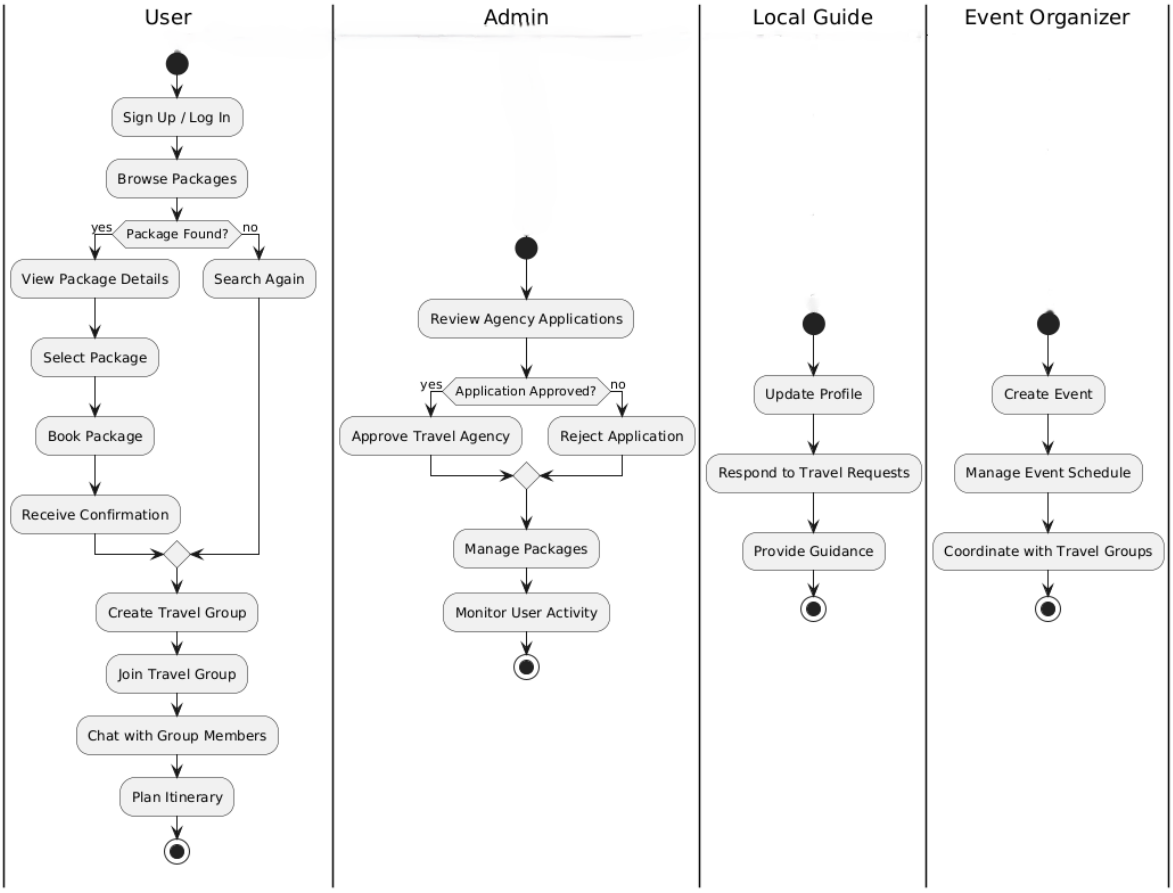


Fig 4.2. 4: Activity Diagra

4.2.5 Class Diagram

A class diagram is a type of structural diagram that shows the classes and their relationships in a system. It represents the static structure of the system and can help identify the objects, attributes, and methods required for the system.

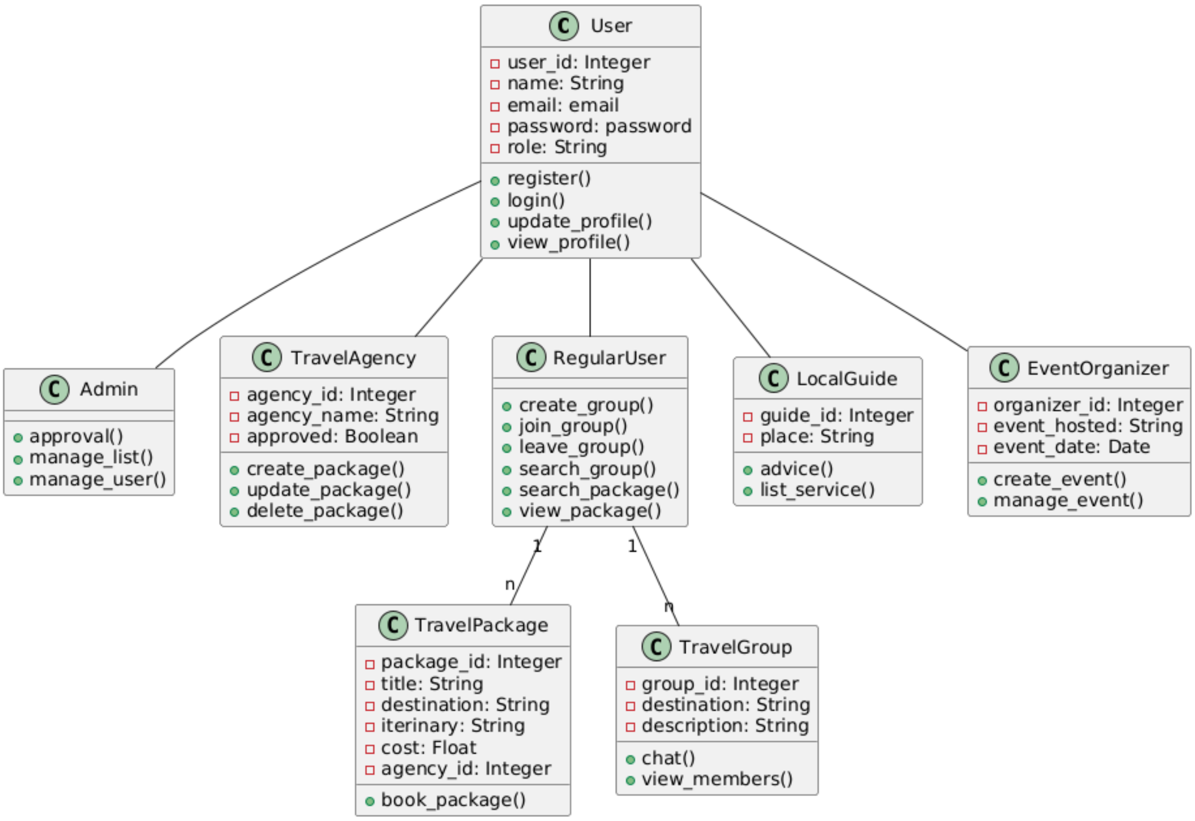


Fig 4.2. 5: Class Diagram

4.2.6 Object Diagram

An object diagram is a type of UML diagram that shows a snapshot of the objects and their relationships in a system at a particular point in time. It is a graphical representation of the instances of classes in a system, along with their attributes and associations.

Object diagrams are used to provide a detailed view of a specific portion of a system, often to help understand or debug a particular issue or scenario.

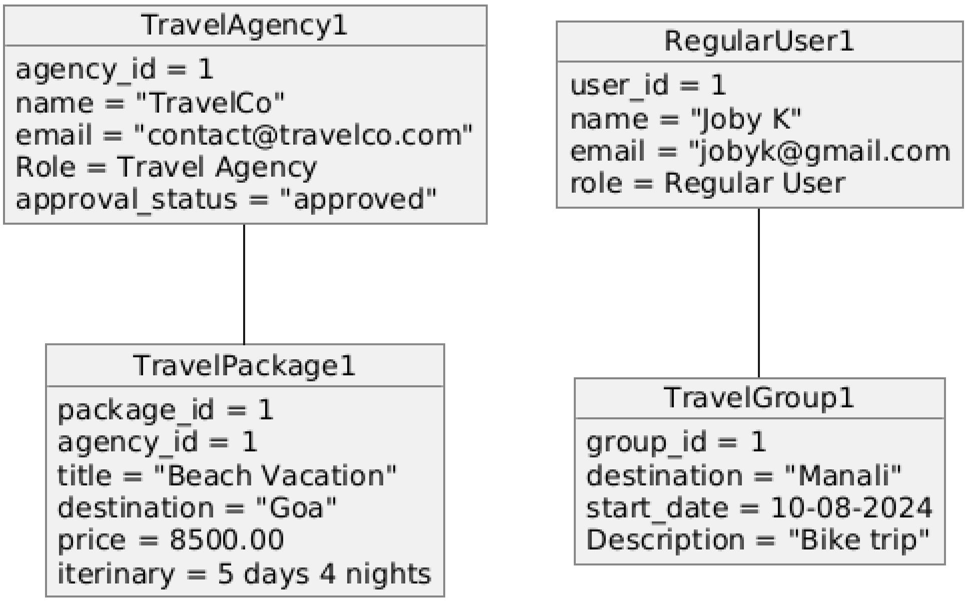


Fig 4.2. 6: Object Diagram

4.2.7 Component Diagram

A component diagram is a type of diagram that shows the physical components of a system and how they are organized and connected. A component is a modular unit that encapsulates some functionality or behaviour of a system. Components can be software modules, libraries, frameworks, executables, or hardware devices. A component diagram helps to visualize the structure and dependencies of the components in a system, as well as the interfaces they provide and require. A component diagram can also show the deployment of the components on different nodes or platforms. A component diagram is useful for designing, documenting, and understanding complex systems that involve multiple components with different responsibilities and interactions.

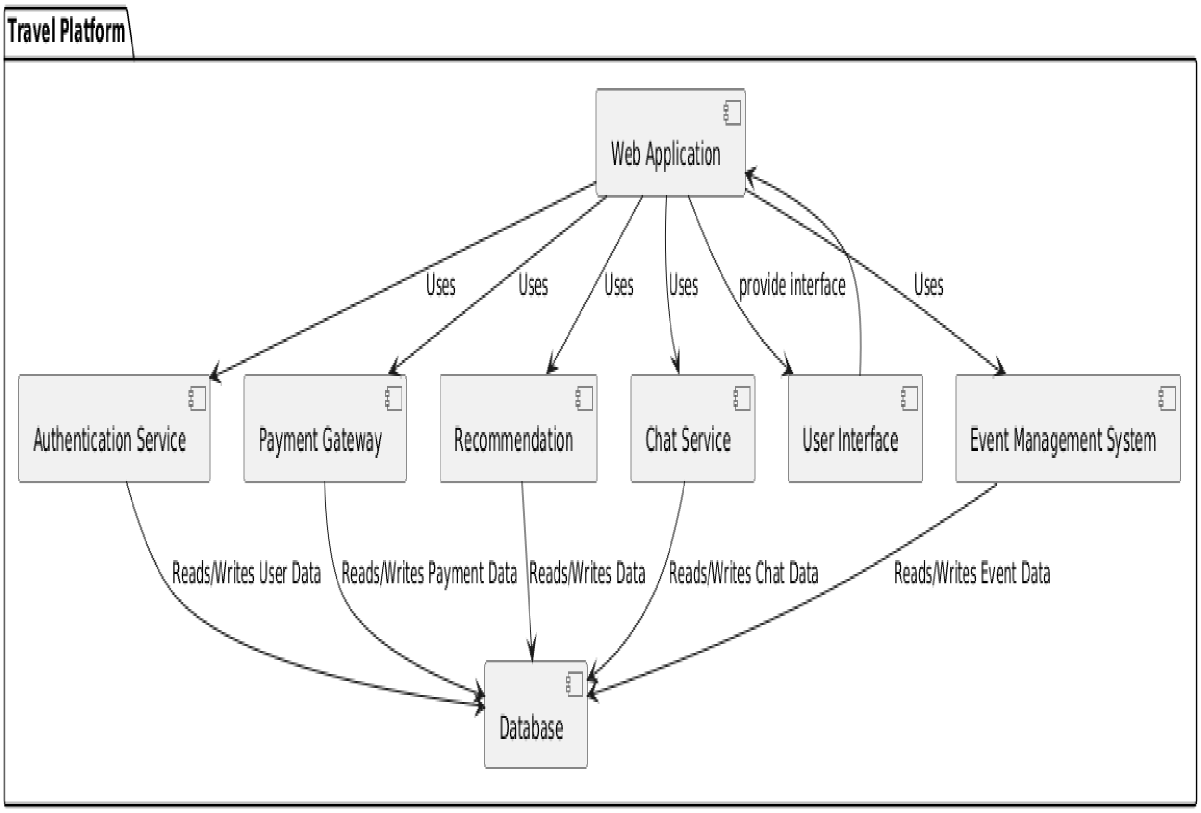


Fig 4.2. 7: Component Diagram

4.2.8 Deployment Diagram

A deployment diagram is a type of diagram that shows the physical arrangement of the components of a software system. It depicts how the software artifacts, such as executable files, libraries, and databases, are deployed on the hardware nodes, such as servers, workstations, and devices. A deployment diagram also shows the communication links between the nodes, such as network connections, protocols, and bandwidth.

A deployment diagram is useful for modelling the physical aspects of a software system, such as the distribution, scalability, performance, and security. It can also show the configuration and dependencies of the components and nodes. A deployment diagram can be used to document the existing system architecture or to design a new one.

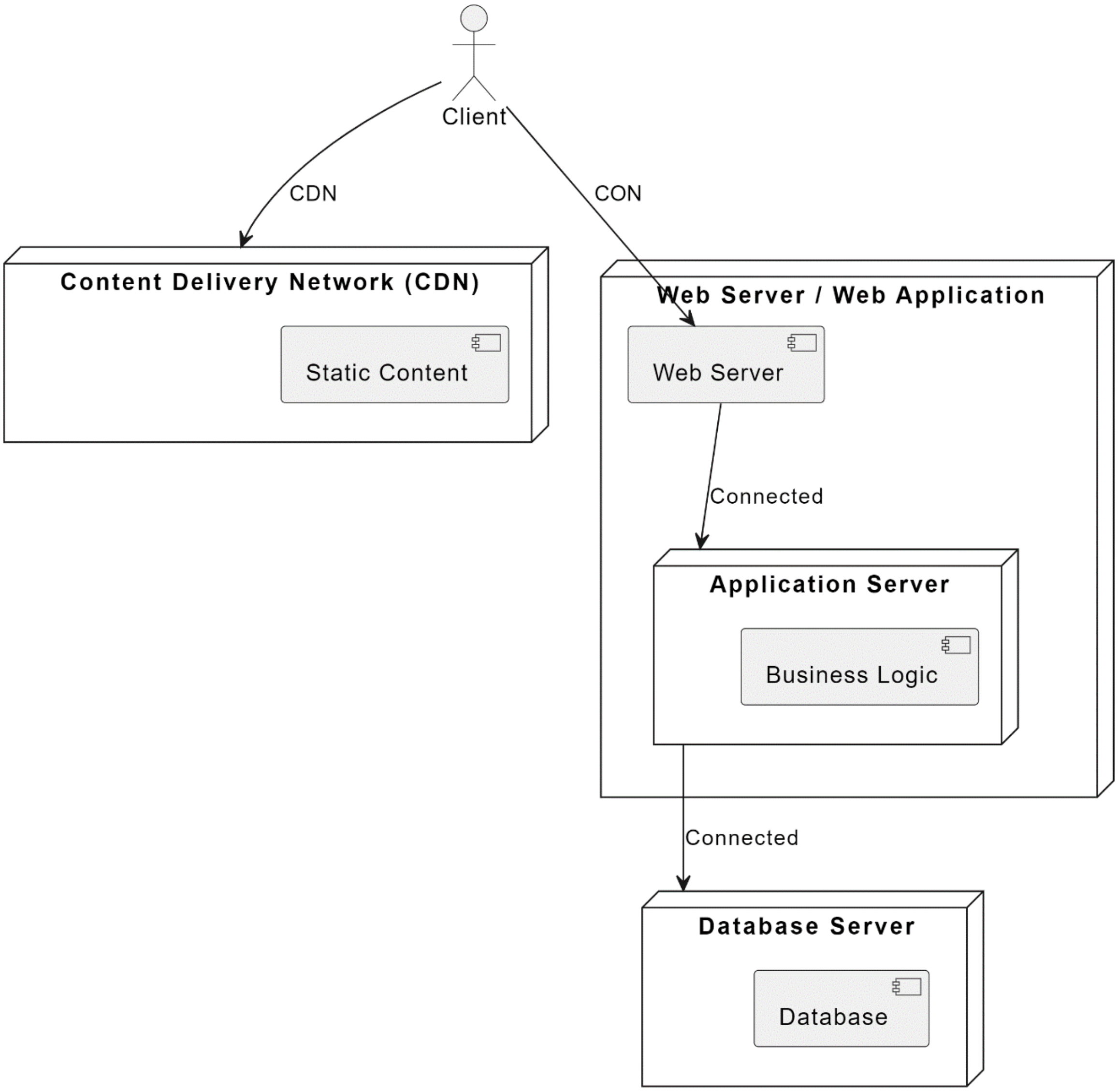


Fig 4.2. 8: Deployment Diagram

4.2.9 Collaboration Diagram

A collaboration diagram is a type of diagram that shows the interactions between objects in a system. It focuses on the roles and responsibilities of each object and how they collaborate to achieve a common goal. A collaboration diagram is also known as a communication diagram or an interaction diagram.

A collaboration diagram consists of objects, links, and messages. Objects are represented by rectangles with the object name and optionally the class name. Links are represented by solid lines that connect objects. They indicate the relationships or associations between objects. Messages are represented by arrows that show the direction and sequence of communication between objects. They can have labels that indicate the name, parameters, and return value of the message.

A collaboration diagram can be used to model different scenarios or use cases in a system. It can help to visualize the dynamic behaviour and interactions of objects, as well as their roles and responsibilities.

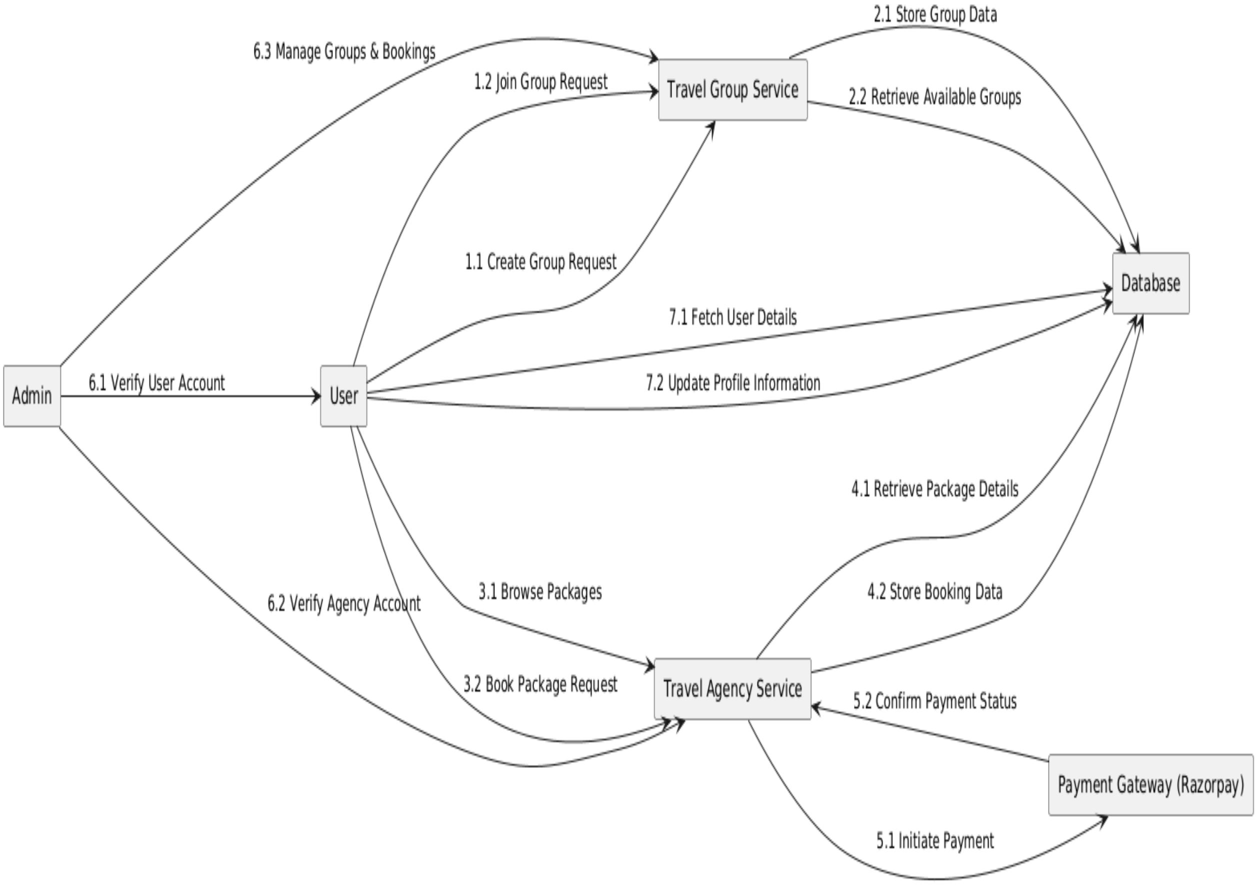


Fig 4.2. 9: Collaboration Diagram

4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: Login form

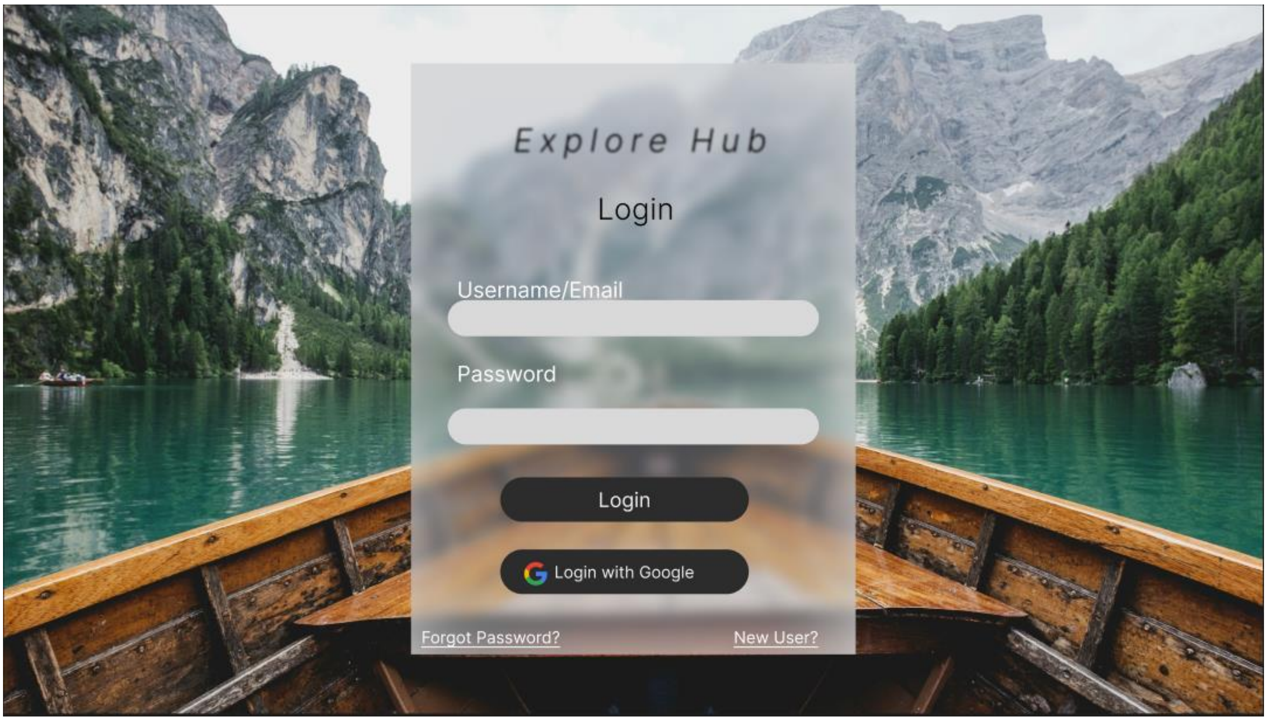
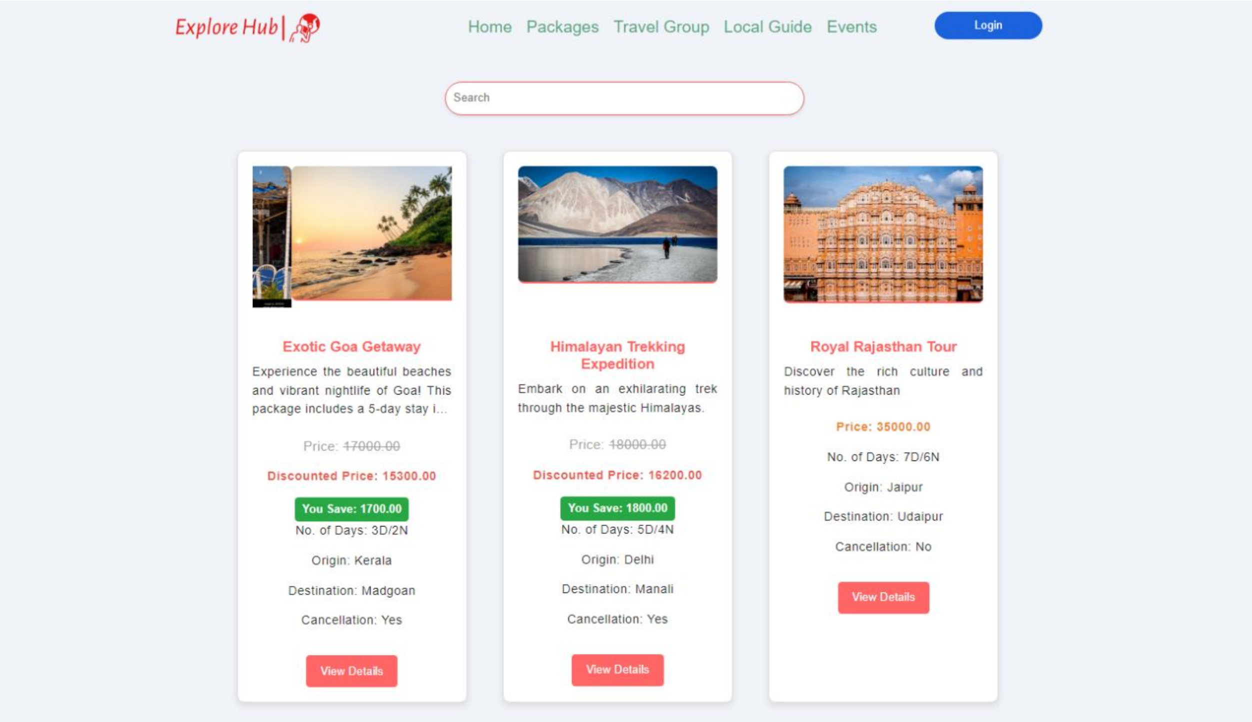


Fig 4.3. 1: Login Form

Form name: Package listing



Exotic Goa Getaway

Experience the beautiful beaches and vibrant nightlife of Goa! This package includes a 5-day stay i...

Price: ~~17000.00~~

Discounted Price: 15300.00

You Save: 1700.00

No. of Days: 3D/2N

Origin: Kerala

Destination: Madgoan

Cancellation: Yes

View Details

Himalayan Trekking Expedition

Embark on an exhilarating trek through the majestic Himalayas.

Price: ~~18000.00~~

Discounted Price: 16200.00

You Save: 1800.00

No. of Days: 5D/4N

Origin: Delhi

Destination: Manali

Cancellation: Yes

View Details

Royal Rajasthan Tour

Discover the rich culture and history of Rajasthan

Price: 35000.00

No. of Days: 7D/6N

Origin: Jaipur

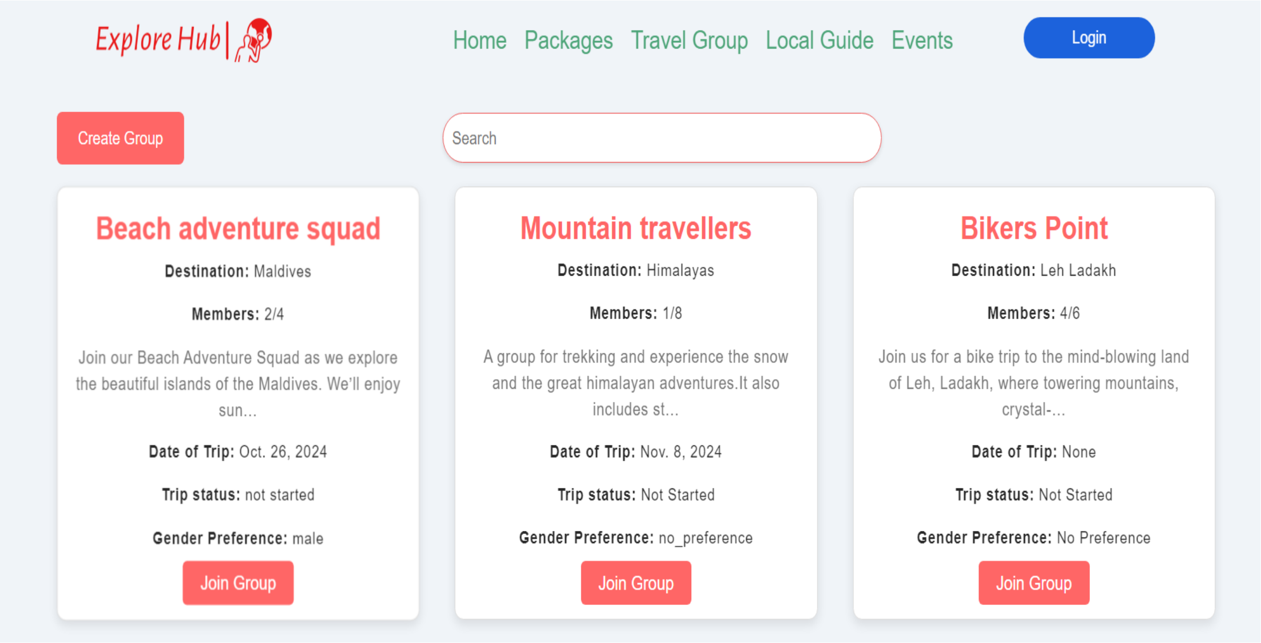
Destination: Udaipur

Cancellation: No

View Details

Fig 4.3. 2: Package Listing

Form name: Group Listing



Beach adventure squad

Destination: Maldives

Members: 2/4

Join our Beach Adventure Squad as we explore the beautiful islands of the Maldives. We'll enjoy sun...

Date of Trip: Oct. 26, 2024

Trip status: not started

Gender Preference: male

Join Group

Mountain travellers

Destination: Himalayas

Members: 1/8

A group for trekking and experience the snow and the great himalayan adventures.It also includes st...

Date of Trip: Nov. 8, 2024

Trip status: Not Started

Gender Preference: no_preference

Join Group

Bikers Point

Destination: Leh Ladakh

Members: 4/6

Join us for a bike trip to the mind-blowing land of Leh, Ladakh, where towering mountains, crystal-...

Date of Trip: None

Trip status: Not Started

Gender Preference: No Preference

Join Group

Fig 4.3. 3: Group Listing

4.4 DATABASE DESIGN

4.4.1 Relational Database Management System (RDBMS)

RDBMS stands for Relational Database Management System. It is a type of database management system that is based on the relational model, which organizes data into one or more tables (or "relations") of columns and rows.

In an RDBMS, data is organized into tables, with each table representing a specific entity or object in the system. Each row in a table represents a single instance of that entity, and each column represents a specific attribute or characteristic of the entity. The relationships between tables are defined through the use of keys, which are columns that are used to identify unique instances of an entity in the system.

RDBMSs are widely used in many different applications and industries, as they provide a reliable, scalable, and flexible way to store and manage large amounts of data. Some popular RDBMSs include MySQL, Oracle Database, Microsoft SQL Server, and PostgreSQL.

4.4.2 Normalization

Normalization is the process of organizing data in a relational database to reduce redundancy and improve data integrity. The goal of normalization is to ensure that each piece of data is stored in only one place and that it is easy to maintain and update.

Normalization is achieved by applying a series of rules to the database design, called normal forms. The most commonly used normal forms are first normal form (1NF), second normal form (2NF), third normal form (3NF), and so on.

Normalization can improve the efficiency and performance of a database, as well as make it easier to maintain and update. However, excessive normalization can also lead to increased complexity and decreased performance, so it is important to strike a balance between normalization and practical considerations for the particular database and its intended use.

4.5 TABLE DESIGN

1.Tbl_users

Primary key: user_id

Foreign key: Nil

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	User_id	Int(11)	PK	Id of the users login
2	Password	Varchar(128)	Not null	Password of the user logging in
3	Last_login	Datetime(6)	Not null	Date of the last login
4	Is_superuser	Tinyint(1)	Not null	Check wether the user is admin or not
5	Username	Varchar(150)	Not null	Username used for login
6	First_name	Varchar(150)	Not null	Name of the user logging in
7	Email	Varchar(254)	Not null	Email id of the user
8	is_active	Tinyint(1)	Not null	Checks wether the user is active or not, also wether the user is blocked or not
9	date_joined	Datetime(6)	Not null	Date and time when the user is Joined
10	phone_number	Varchar(15)	Not null	Phone number of the user
11	Role	Varchar(30)	Not null	Determines the role of the user

2. TravelAgency

Primary key: agency_id

Foreign key: Nil

No:	Field name	Datatype(Size)	Key Constraints	Description of the field
1	Agency_id	Int(11)	PK	Id of the agency registered
2	Name	Varchar(255)	Not null	Name of the travel agency registered
3	Username	Varchar(255)	Not null	Username of the travel agency
4	Password	Varchar(255)	Not null	Password of the travel agency
5	Contact	Varchar(10)	Not null	Phone number of the travel agency

6	Email	Varchar(254)	Not null	Email id of the travel agency
7	Documents	Varchar(100)	Not null	Proof documents submitted for the verification
8	Approved	Tinyint(1)	Not null	Boolean value to check wether the travel agency is approved or not by the admin
9	Agreement	Tinyint(1)	Not null	Boolean value to check wether the travel agency had approved the agreement.

3. TravelPackage

Primary key: package_id

Foreign Key: agency_id referencing TravelAgency

No	Field name	Datatype (Size)	Key Constraints	Description of the field
1	package_id	Int (11)	PK	Id of the package
2	Title	Varchar (200)	Not null	Title of the package
3	Description	Longtext	Not null	Description about the package
4	Price	Decimal	Not null	Price of the package
5	Agency_id	Int (11)	Foreign key referencing TravelAgency	Denotes the agency which the package is listed
6	Destination	Varchar (200)	Not null	Denotes the destination of the package
7	Origin	Varchar (200)	Not null	Denotes the origin of the package
8	Cancellation	Tinyint (1)	Not null	Boolean field to determine wether the package offers cancellation or not
9	Duration	Varchar (10)	Not null	Denotes the duration of the package
10	is_archived	Tinyint (1)	Not null	Boolean field to determine wether the package is archived or not
11	discount_percentage	Decimal	Not null	Determines wether the package offeres discount and what percentage

12	Is_active	Tinyint (1)	Not null	Boolean field to determine wether the package is active or not
13	itinerary	Longtext	Not null	Describes the itinerary of the package

4. PackageImage

Primary Key: image_id

Foreign Key: travel_package_id referencing TravelPackage

No	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Image_d	Bigint (20)	PK	Id of the field
2	Image	Varchar (100)	Not null	Image of the package
3	travel_package_id	Int (11)	Foreign Key	Field linking the travel package

5. TravelGroup

Primary Key: group_id

Foreign Key: creator_id referencing tbl_user

No	Field name	Datatype (Size)	Key Constraints	Description of the field
1	group_id	Int	PK	id of the group
2	name	Varchar (200)	Not null	Name of the travel group
3	Destination	Varchar (200)	Not null	Destination of the preffered travel group
4	max_members	Integer	Not null	Denotes the maximum number of the members to join the group
5	creator_id	int	Foreign key referencing user table	Links the user who creted the group
6	gender	Varchar(20)	Not null	Denotes the prefered gender of the members of the group
7	Trip_date	Datetime	Not null	Denotes the date of the trip prefered

8	Trip_status	varchar(20)	Not null	Dentoes wether the trip had started or not

6. Booking

Primary key: tbl_booking_id

Foreign Key: package_id referencing TravelPackage, user_id referencing tbl_user

No	Field name	Datatype(Size)	Key Constraints	Description of the field
1	Tbl_booking_id	Bigint(20)	PK	Denotes the id of the booking made
2	Booking_date	Datetime	Not null	Denotes the timestamp when the booking is done
3	Total_amount	Decimal	Not null	Denotes the total price for which the package is booked
4	Is_confirmed	Tinyint(1)	Not null	Boolean value to check wether the booking is confirmed or not
5	Number_of_people	Int(10)	Not null	Denotes the number of people
6	Booking_id	Varchar(200)	Not null	Denotes the booking id
7	Is_cancelled	Tinyint(1)	Not null	Denotes wether the booking is cancelled or not
8	Cancellation	Tinyint(1)	Not null	Boolean field to denote wether cancellation is available for the particular package booked
9	Payment_status	Varchar(20)	Not null	Denotes whether the package payment is done or not
10	Package_id	Int(11)	Foreign key referencing TravelPackage	Denotes the package that is booked
11	Transaction_id	Varchar(100)	Not null	Denotes transaction id of the package booked
12	Payment_method	Varchar(50)	Not null	Denotes which payment method is used for paying
13	Payment_date	Datetime	Not null	Denotes the timestamp when the pakcage is booked

14	User_id	Int(11)	Foreign key referencing tbl_user	Dentoes the user who booked the package
15	Trip_date	Date	Not null	Denotes the date of the package booked
16	Refund_amount	Decimal	Not null	The amount if refund is inititated

7. Passenger

Primary Key: passenger_id

Foreign Key: booking_id referencing booking

No	Field name	Datatype(Size)	Key Constraints	Description of the field
1	Passenger_id	Bigint(20)	PK	Denotes the id of the passenger travelling in a package booked
2	Full_name	Varchar(100)	Not Null	Full name of the passengers travelling
3	Age	Int(10)	Not Null	Age of the passengers travelling
4	Booking_id	Bigint(10)	Foreign Key referencing booking	Foreign key for denoting the booking of the people

8. Message

Primary key: id

Foreign key: user_id referencing tbl_users

No	Field name	Datatype(Size)	Key Constraints	Description of the field
1	Id	Bigint(20)	PK	Denotes the id of the message being send
2	Content	Longtext	Not null	Contains the original message
3	Send_at	Datetime	Not null	Denotes the timestamp of when the message is being send by the user

4	Group_id	Int(11)	Not null	Denotes the group id in which the message is send
5	User_id	Int(11)	Foreign key referencing tbl_user	Denotes the user id of whom the message is send

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

System testing is a process of evaluating the functionality and performance of a software system or application as a whole, to ensure that it meets the specified requirements and is fit for purpose. The goal of system testing is to identify and correct any defects or issues that may affect the system's functionality, usability, reliability, security, and other quality attributes.

System testing is typically conducted after the completion of integration testing, which tests the interactions between different components or modules of the system. System testing involves testing the entire system in a realistic environment, using real-world data and scenarios, to simulate the end user's experience.

System testing may include various types of testing, such as functional testing, performance testing, security testing, usability testing, and compatibility testing. It may be performed manually or using automated testing tools and techniques.

5.2 TEST PLAN

A test plan is a document that outlines the objectives, scope, approach, and resources required for testing a software system or application. The purpose of a test plan is to guide the testing process and ensure that all necessary tests are conducted and documented.

Here are some of the key components of a test plan:

- Test objectives: This section outlines the overall goals and objectives of the testing effort, and how the testing will help to achieve the project's overall goals.
- Test scope: This section defines the scope of the testing effort, including the areas of the system to be tested, the types of testing to be conducted, and any limitations or constraints on the testing effort.
- Test approach: This section outlines the overall approach to be taken for testing the system, including the testing methodology, tools and techniques to be used, and any specific testing strategies or tactics.
- Test schedule: This section provides a detailed timeline of the testing effort, including the start and end dates for each phase of testing, as well as any milestones or deadlines that must be met.
- Test resources: This section lists the resources required for the testing effort, including personnel, hardware, software, and any other equipment or facilities needed for testing.
- Test deliverables: This section lists the deliverables that will be produced during the testing effort, such as test plans, test cases, test reports, and any other documentation.

- Test risks: This section identifies the potential risks and issues that may arise during the testing effort, and outlines strategies for mitigating or addressing them.

Overall, a test plan is an essential tool for ensuring that a software system is thoroughly tested and meets the desired quality standards. It provides a roadmap for the testing effort, ensuring that all necessary tests are conducted and that the results are properly documented and reported.

5.2.1 Unit Testing

Unit testing is a type of software testing that involves testing individual units or components of a software application in isolation from the rest of the system. The purpose of unit testing is to ensure that each unit of code, such as a function or method, works correctly and meets its intended functionality.

Unit tests are typically automated, meaning that they can be run quickly and easily whenever changes are made to the code. They are also repeatable, meaning that the same tests can be run over and over again to ensure that the code continues to function as expected.

Unit testing can help catch bugs early in the development cycle, before they have a chance to cause larger problems in the system. It can also help developers to identify and fix issues more quickly, leading to faster development times and higher quality software.

In order to perform unit testing, developers use specialized testing frameworks and tools that help automate the process and provide feedback on the success or failure of individual tests. Common unit testing frameworks include JUnit for Java, NUnit for .NET, and pytest for Python, among others.

5.2.2 Integration Testing

Integration testing is a type of software testing that aims to test how different components or modules of a software application work together. The purpose of integration testing is to verify that these components can communicate and exchange data with each other as expected, and that they function correctly as a unified system.

Integration testing typically occurs after unit testing has been completed, and it may involve testing different modules or components of the application together in pairs or in larger groups. This can be done in a variety of ways, including using stubs or mock objects to simulate the behaviour of missing components, or by testing against a live system or database.

There are several different types of integration testing, including:

- Big Bang Integration: In this approach, all components are integrated at once and tested as a whole.

• Top-Down Integration: In this approach, testing starts from the top layer of the application and works its way down, integrating each layer as it goes.

• Bottom-Up Integration: In this approach, testing starts from the bottom layer of the application and works its way up, integrating each layer as it goes.

• Sandwich Integration: In this approach, testing starts from the top layer of the application, works its way down to the bottom layer, and then back up to the top layer again.

Integration testing is important because it helps to ensure that the different components of an application work together as intended, and that any problems or issues are identified and fixed before the application is released to users.

5.2.3 Validation Testing or System Testing

Validation testing, also known as system testing, is a type of software testing that aims to ensure that a software system meets its intended purpose and satisfies the requirements of the stakeholders. It is typically performed after integration testing has been completed and involves testing the entire system as a whole, rather than individual components.

The purpose of validation testing is to verify that the system functions correctly in the context of the larger environment in which it will be used. This includes testing the system's functionality, usability, performance, reliability, and security, among other factors.

Validation testing may involve a combination of manual and automated testing techniques, and may be conducted using various testing methods, such as:

• Functional testing: This involves testing the system's functionality to ensure that it meets the requirements of the stakeholders.

• Usability testing: This involves testing the system's ease of use and user interface to ensure that it is intuitive and user-friendly.

• Performance testing: This involves testing the system's speed, scalability, and resource usage to ensure that it can handle the expected workload.

• Security testing: This involves testing the system's security features and vulnerabilities to ensure that it is protected against unauthorized access or malicious attacks.

• Compatibility testing: This involves testing the system's compatibility with different platforms, devices, and software configurations.

Validation testing is important because it helps to ensure that the software system is ready for release and meets the expectations of the stakeholders. It can help to identify and address any issues or defects before the system is deployed, which can save time and resources in the long run.

5.2.4 Output Testing or User Acceptance Testing

Output testing, also known as user acceptance testing (UAT), is a type of software testing that involves testing the software system from the perspective of the end-user. The purpose of user acceptance testing is to verify that the system meets the requirements and expectations of the stakeholders, and that it functions correctly in the context of the user's workflow and environment.

User acceptance testing is typically performed by a group of users or stakeholders who are representative of the target audience for the software system. These users are given specific scenarios or tasks to perform, and they provide feedback on their experience using the system. The focus of user acceptance testing is on the output or results of the system, rather than the individual components or technical details.

The main objectives of user acceptance testing are to:

1. Verify that the system meets the requirements and expectations of the stakeholders.

2. Validate that the system is easy to use and understand.

3. Confirm that the system is reliable, accurate, and produces the expected output.

4. Ensure that the system integrates seamlessly with other systems or tools used by the users.

5. Identify any remaining defects or issues that need to be addressed before the system is released.

User acceptance testing is important because it provides a final check to ensure that the software system is ready for deployment and meets the needs of the end-users. It helps to validate that the system will be accepted and adopted by the users, which can ultimately determine the success of the project.

5.2.5 Automation Testing

Automation testing is a type of software testing that involves the use of tools and software to automate the execution of test cases and the comparison of actual results with expected results. The purpose of automation testing is to improve the efficiency and effectiveness of the testing process by reducing the time and effort required to execute tests and analyze results.

Automation testing can be applied to different types of testing, including unit testing, integration testing, and system testing. It involves the use of specialized tools and frameworks that can automate the testing process, such as:

- Test automation frameworks: These provide a structured approach to automate the testing process, including test case management, test data management, and reporting.
- Test scripting tools: These enable the creation and execution of automated test scripts, which can simulate user interactions with the software system.
- Test management tools: These enable the organization and scheduling of automated tests, and provide tools for tracking test results and defects.
- Automation testing offers several advantages over manual testing, including:

1) Increased efficiency: Automation testing can execute test cases faster and more accurately than manual testing, which can save time and reduce the cost of testing.

2) Improved test coverage: Automation testing can cover a larger number of test cases and scenarios than manual testing, which can help to identify defects and issues that may be missed by manual testing.

3) Increased accuracy: Automation testing can eliminate human errors and biases that may occur in manual testing, which can improve the accuracy and reliability of test results.

4) Reusability: Automated test scripts can be reused across different testing cycles and environments, which can save time and effort in the long run.

However, automation testing also has some limitations, such as the need for specialized skills and resources to develop and maintain automated tests, as well as the inability to test certain aspects of the system that require human intuition or judgment.

5.2.6 Selenium Testing

Selenium testing is a popular open-source automation testing tool used for web application testing. It allows testers to automate the testing of web applications across different browsers and platforms, using various programming languages such as Java, Python, C#, and more.

Selenium testing supports different types of testing, including functional testing, regression testing, and compatibility testing. It offers several features that make it a popular choice for web application testing, such as:

- Cross-browser testing: Selenium testing allows for automated testing of web applications across different browsers, including Chrome, Firefox, Safari, and Internet Explorer.

- Record and playback: Selenium IDE, a Selenium testing tool, allows testers to record and playback interactions with the web application, which can be useful for creating test scripts.

• Multi-language support: Selenium testing supports various programming languages, which allows testers to write test scripts in a language of their choice.

• Integration with other tools: Selenium testing can be integrated with other testing tools, such as TestNG and JUnit, for better test management and reporting.

• Parallel testing: Selenium Grid, another Selenium testing tool, allows testers to execute tests across different browsers and platforms in parallel, which can save time and increase testing efficiency.

Selenium testing is widely used for web application testing, particularly for testing complex web applications with dynamic content and user interfaces. However, it also has some limitations, such as the need for regular updates to keep up with changes in browsers and web technologies, as well as the complexity of maintaining and scaling test scripts in large and complex projects.

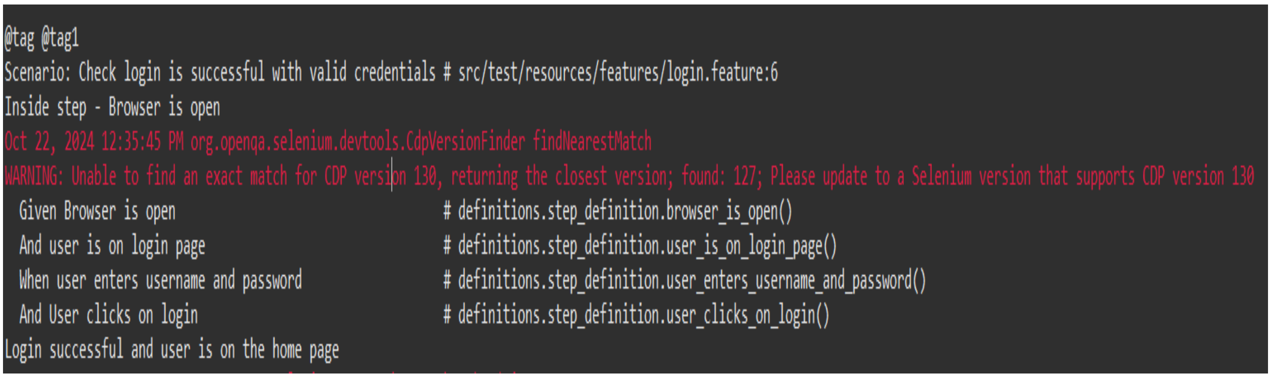
Test Case 1

Code

```
package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import java.time.Duration; // Import Duration
import io.cucumber.java.en.*;

public class step_definition {
    WebDriver driver = null;
    WebDriverWait wait;

    @Given("Browser is open")
    public void browser_is_open() {
        System.out.println("Inside step - Browser is open");
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\kiran\\eclipse-
workspace\\explore\\src\\test\\resources\\driver\\chromedriver.exe");     //
Update with your path
```

```
        ChromeOptions options = new ChromeOptions();
        options.setBinary("C:\\Program         Files\\BraveSoftware\\Brave-
Browser\\Application\\brave.exe");

        // Launch Brave Browser
        driver = new ChromeDriver(options);
        driver.manage().window().maximize();

        // Initialize WebDriverWait with a timeout of 20 seconds
        wait  =  new  WebDriverWait(driver,  Duration.ofSeconds(20));  //  Use
Duration for timeout
    }

    @And("user is on login page")
    public void user_is_on_login_page() {
        driver.navigate().to("http://127.0.0.1:8000/login");
        // Removed Thread.sleep, relying on waits in the next steps
    }

    @When("user enters username and password")
    public void user_enters_username_and_password() {
        // Wait for the username field to be visible and enter username

wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("username"))
);
        driver.findElement(By.id("username")).sendKeys("kirankurian");

        // Wait for the password field to be visible and enter password

wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("password"))
);
        driver.findElement(By.id("password")).sendKeys("1234");
    }

    @And("User clicks on login")
    public void user_clicks_on_login() {
        // Wait for the login button to be clickable and click it

wait.until(ExpectedConditions.elementToBeClickable(By.id("login")));
        driver.findElement(By.id("login")).click();
    }

    @Then("user is navigated to the home page")
    public void user_is_navigated_to_the_home_page() {
        // Wait for the logout button to be visible to confirm navigation to
the home page
        Boolean isLogoutDisplayed =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("logout"))).
isDisplayed();

        if (isLogoutDisplayed) {
            System.out.println("Login successful and user is on the home
page");
        } else {
            System.out.println("Login failed or not navigated to the home
page");
        }

        driver.close();
        driver.quit();
    }
}
```

Screenshot



```
@tag @tag1
Scenario: Check login is successful with valid credentials # src/test/resources/features/login.feature:6
Inside step - Browser is open
Oct 22, 2024 12:35:45 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 127; Please update to a Selenium version that supports CDP version 130
  Given Browser is open                          # definitions.step_definition.browser_is_open()
  And user is on login page                      # definitions.step_definition.user_is_on_login_page()
  When user enters username and password         # definitions.step_definition.user_enters_username_and_password()
  And User clicks on login                       # definitions.step_definition.user_clicks_on_login()
Login successful and user is on the home page
```

Test Report

Test Case 1					
Project Name: Explore Hub					
Login Test Case					
Test Case ID: Test_1			**Test Designed By: Kiran Kurian Philip**		
Test Priority (Low/Medium/High):			**Test Designed Date:22/10/24**		
Module Name: Login			**Test Executed By: Nimmy Francis**		
Test Title: Login Test case			**Test Execution Date:22/10/24**		
Description:			Test executed successfully		
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigate to login page	N/A	Login page loads successfully	Login page loaded successfully	Pass
2	Fill in the username and password	“kirankurian” for username and “1234” for password	Username and password fields are successfully filled in	Username and password fields are successfully filled in	Pass
3	Submit the login form	N/A	User is successfully logged in	User is successfully logged in	Pass
4	Close the browser window	N/A	Browser window is successfully closed	Browser window is successfully closed	Pass
Post-Condition: User successfully logged in					

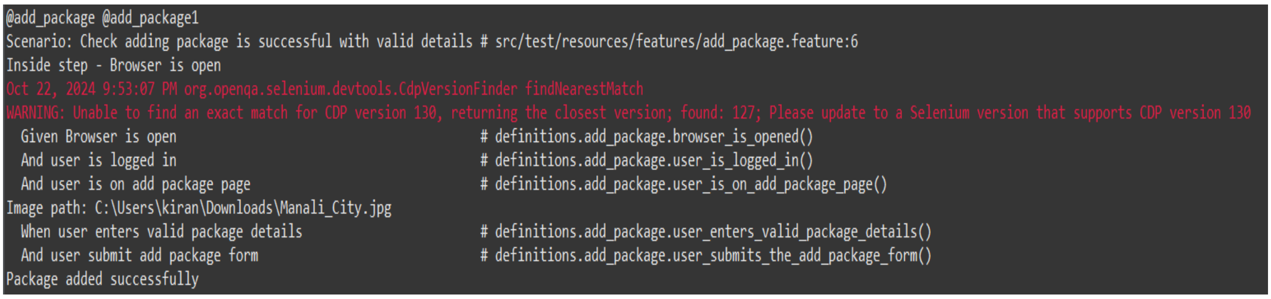
Test Case 2: Add package

Code

```
package definitions;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import io.cucumber.java.en.*;
```

```
public class add_package {
    WebDriver driver = null;

    @Given("Browser is open")
    public void browser_is_opened() {
        System.out.println("Inside step - Browser is open");
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\kiran\\eclipse-
workspace\\explore\\src\\test\\resources\\driver\\chromedriver.exe");
        ChromeOptions options = new ChromeOptions();
        options.setBinary("C:\\Program         Files\\BraveSoftware\\Brave-
Browser\\Application\\brave.exe");
        driver = new ChromeDriver(options);
        driver.manage().window().maximize();
    }

    @And("user is logged in")
    public void user_is_logged_in() {
        // Navigate to login page
        driver.navigate().to("http://127.0.0.1:8000/login");
        driver.findElement(By.id("username")).sendKeys("mytravels");
        driver.findElement(By.id("password")).sendKeys("mytravel");
        driver.findElement(By.id("login")).click();
    }

    @And("user is on add package page")
    public void user_is_on_add_package_page() throws Exception {
        driver.navigate().to("http://127.0.0.1:8000/addpackage/");
        Thread.sleep(2000);
    }

    @When("user enters valid package details")
    public void user_enters_valid_package_details() {
        driver.findElement(By.id("title")).sendKeys("Mountain Adventure");
        driver.findElement(By.id("description")).sendKeys("Sample
description for testing the add package option");
        driver.findElement(By.id("origin")).sendKeys("Kerala");
        driver.findElement(By.id("destination")).sendKeys("Himalayas");
        driver.findElement(By.id("price")).sendKeys("1500");
        driver.findElement(By.id("discount_percentage")).sendKeys("10");
        driver.findElement(By.id("duration")).sendKeys("3D/2N");

driver.findElement(By.id("itinerary")).sendKeys(";laljk;ldjkjhdkajhldkajhlk
jhlakjdhlakjahlk");
        String imagePath = "C:\\Users\\kiran\\Downloads\\Manali_City.jpg";
      System.out.println("Image path: " + imagePath);
      driver.findElement(By.id("images")).sendKeys(imagePath);
    }

    @And("user submit add package form")
    public void user_submits_the_add_package_form() {
        driver.findElement(By.id("add_package")).click();
    }

    @Then("package is added successfully")
    public void package_is_added_successfully() {
        System.out.println("Package added successfully");
        driver.close();
        driver.quit();
    }
}
```

Screenshot



```
@add_package @add_package1
Scenario: Check adding package is successful with valid details # src/test/resources/features/add_package.feature:6
Inside step - Browser is open
Oct 22, 2024 9:53:07 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 127; Please update to a Selenium version that supports CDP version 130
  Given Browser is open                                         # definitions.add_package.browser_is_opened()
  And user is logged in                                         # definitions.add_package.user_is_logged_in()
  And user is on add package page                               # definitions.add_package.user_is_on_add_package_page()
Image path: C:\Users\kiran\Downloads\Manali_City.jpg
  When user enters valid package details                        # definitions.add_package.user_enters_valid_package_details()
  And user submit add package form                              # definitions.add_package.user_submits_the_add_package_form()
Package added successfully
```

Test report

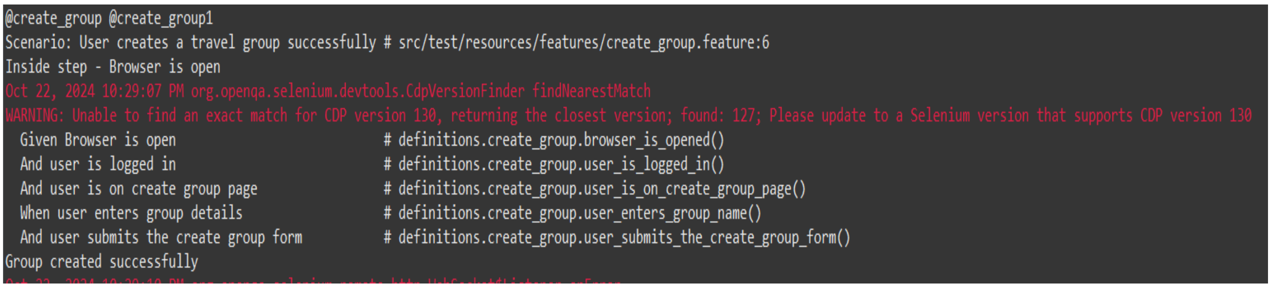
<table>
<tr><th colspan="6">Test Case 2</th></tr>
<tr><td colspan="6"><b>Project Name: Explore Hub</b></td></tr>
<tr><td colspan="6"><b>Add Package Test Case</b></td></tr>
<tr><td colspan="3"><b>Test Case ID: Test_2</b></td><td colspan="3"><b>Test Designed By: Kiran Kurian Philip</b></td></tr>
<tr><td colspan="3"><b>Test Priority (Low/Medium/High):</b></td><td colspan="3"><b>Test Designed Date:22/10/24</b></td></tr>
<tr><td colspan="3"><b>Module Name</b>: Add Package</td><td colspan="3"><b>Test Executed By: Nimmy Francis</b></td></tr>
<tr><td colspan="3"><b>Test Title: Add Package Test case</b></td><td colspan="3"><b>Test Execution Date:22/10/24</b></td></tr>
<tr><td colspan="3"><b>Description:</b></td><td colspan="3">Test to add a travel package by travel agency</td></tr>
<tr><td colspan="6"><b>Pre-Condition:</b> User has valid username and password</td></tr>
<tr><td><b>Step</b></td><td><b>Test Step</b></td><td><b>Test Data</b></td><td><b>Expected Result</b></td><td><b>Actual Result</b></td><td><b>Status (Pass/ Fail)</b></td></tr>
<tr><td>1</td><td>Navigate to login page</td><td>N/A</td><td>Login page loads successfully</td><td>Login page loaded successfully</td><td>Pass</td></tr>
<tr><td>2</td><td>User is logged in</td><td>“mytravels” for username and “mytravel” for password</td><td>Username and password fields are successfully filled in and logged in</td><td>Username and password fields are successfully filled in and logged in</td><td>Pass</td></tr>
<tr><td>3</td><td>User is on add package page</td><td>N/A</td><td>User landed on the add package form page</td><td>User landed on the add package form page</td><td>Pass</td></tr>
<tr><td>4</td><td>User enters valid package details</td><td>User enters the data of package details</td><td>User enters valid package details</td><td>User enters valid package details</td><td>Pass</td></tr>
<tr><td>5</td><td>User submit the form</td><td>N/A</td><td>User submit the valid form</td><td>User submit the valid form</td><td>Pass</td></tr>
<tr><td>6</td><td>Close the browser window</td><td>N/A</td><td>Browser window is successfully closed</td><td>Browser window is successfully closed</td><td>Pass</td></tr>
<tr><td colspan="6"><b>Post-Condition: The package is inserted successfully</b></td></tr>
</table>

Test Case 3: Create group

Code

```
package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.support.ui.WebDriverWait;
import io.cucumber.java.en.*;

public class create_group {
    WebDriver driver = null;
    WebDriverWait wait;

    @Given("Browser is open")
    public void browser_is_opened() {
        System.out.println("Inside step - Browser is open");
        System.setProperty("webdriver.chrome.driver",
"C:\\Users\\kiran\\eclipse-
workspace\\explore\\src\\test\\resources\\driver\\chromedriver.exe");
        ChromeOptions options = new ChromeOptions();
        options.setBinary("C:\\Program         Files\\BraveSoftware\\Brave-
Browser\\Application\\brave.exe");
        driver = new ChromeDriver(options);
        driver.manage().window().maximize();
    }

    @And("user is logged in")
    public void user_is_logged_in() {
        driver.navigate().to("http://127.0.0.1:8000/login");
        driver.findElement(By.id("username")).sendKeys("kirankurian");
        driver.findElement(By.id("password")).sendKeys("1234");
        driver.findElement(By.id("login")).click();
    }

    @And("user is on create group page")
    public void user_is_on_create_group_page() {
        driver.navigate().to("http://127.0.0.1:8000/create_group");
    }
    @When("user enters group details")
    public void user_enters_group_name() {
        driver.findElement(By.id("group_name")).sendKeys("Adventure
seekers");
        driver.findElement(By.id("destination")).sendKeys("Himalayas");
        driver.findElement(By.id("date")).sendKeys("10-11-2024");
        driver.findElement(By.id("max_members")).sendKeys("4");
        driver.findElement(By.id("gender")).sendKeys("Male");

driver.findElement(By.id("description")).sendKeys("lja;lkdjlakjlhadjhlakjhp
uoheiubkdjb");
    }

    @When("user submits the create group form")
    public void user_submits_the_create_group_form() {
        driver.findElement(By.id("create_group")).click();
    }

    @Then("group should be created successfully")
    public void group_should_be_created_successfully() throws Exception {
        System.out.println("Group created successfully");
        Thread.sleep(2000);
        driver.close();
        driver.quit();
    }
}
```

Screenshot



```
@create_group @create_group1
Scenario: User creates a travel group successfully # src/test/resources/features/create_group.feature:6
Inside step - Browser is open
Oct 22, 2024 10:29:07 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 127; Please update to a Selenium version that supports CDP version 130
  Given Browser is open                              # definitions.create_group.browser_is_opened()
  And user is logged in                              # definitions.create_group.user_is_logged_in()
  And user is on create group page                   # definitions.create_group.user_is_on_create_group_page()
  When user enters group details                     # definitions.create_group.user_enters_group_name()
  And user submits the create group form             # definitions.create_group.user_submits_the_create_group_form()
Group created successfully
```

Test report

Test Case 3	
Project Name: Explore Hub	
Create group Test Case	
Test Case ID: Test_3	**Test Designed By: Kiran Kurian Philip**
Test Priority (Low/Medium/High):	**Test Designed Date:22/10/24**
Module Name: Create Group	**Test Executed By: Nimmy Francis**
Test Title: Create group Test case	**Test Execution Date:22/10/24**
Description:	Travel group is created by the user
Pre-Condition: User has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loaded successfully	Pass
2	User is logged in	“kirankurian” for username and “1234” for password	Username and password fields are successfully filled in and logged in	Username and password fields are successfully filled in and logged in	Pass
3	User is on create group page	N/A	User landed on the create group form page	User landed on the create group form page	Pass
4	User enters valid group details	User enters the data of group details	User enters valid group details	User enters valid group details	Pass
5	User submit the form	N/A	User submit the valid form	User submit the valid form	Pass
6	Close the browser window	N/A	Browser window is successfully closed	Browser window is successfully closed	Pass

Post-Condition: Travel group created successfully

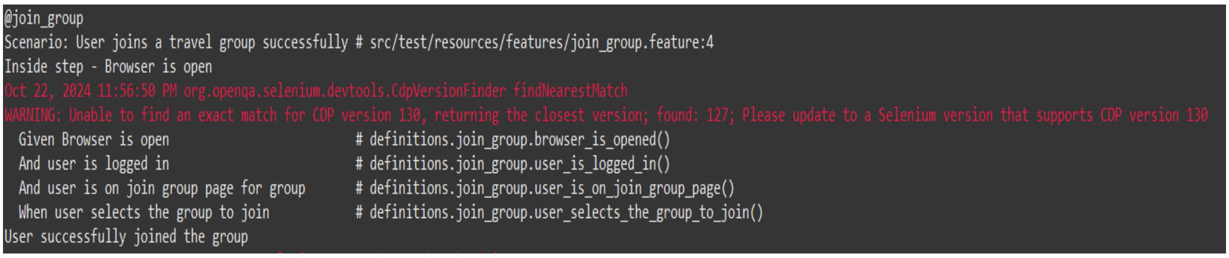
Test Case 4: Join group

Code

```
package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.openqa.selenium.support.ui.ExpectedConditions;
import io.cucumber.java.en.*;
import org.openqa.selenium.WebElement;
import static org.junit.Assert.assertTrue;
import java.time.Duration;
import java.util.List;
import org.openqa.selenium.Alert;
import org.openqa.selenium.TimeoutException;
import org.openqa.selenium.UnhandledAlertException;

public class join_group {
    WebDriver driver = null;
    WebDriverWait wait;
    String groupId;

    @Given("Browser is open")
    public void browser_is_opened() {
        System.out.println("Inside step - Browser is open");
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\kiran\\eclipse-
workspace\\explore\\src\\test\\resources\\driver\\chromedriver.exe");
        ChromeOptions options = new ChromeOptions();
        options.setBinary("C:\\Program Files\\BraveSoftware\\Brave-
Browser\\Application\\brave.exe");
        driver = new ChromeDriver(options);
        driver.manage().window().maximize();
    }

    @And("user is logged in")
    public void user_is_logged_in() {
        driver.navigate().to("http://127.0.0.1:8000/login");
        driver.findElement(By.id("username")).sendKeys("kirankurian");
        driver.findElement(By.id("password")).sendKeys("1234");
        driver.findElement(By.id("login")).click();
    }

    @And("user is on join group page for group")
    public void user_is_on_join_group_page() {
        driver.navigate().to("http://127.0.0.1:8000/available_group");
    }

    @When("user selects the group to join")
    public void user_selects_the_group_to_join() {
        List<WebElement> joinButtons = driver.findElements(By.className("join-btn"));
        if (!joinButtons.isEmpty()) {
            WebElement firstJoinButton = joinButtons.get(0);
            String groupName = driver.findElement(By.xpath("//*[contains(@class,
'group-box')]//*[contains(@class, 'join-btn') and @data-group-id='" +
firstJoinButton.getAttribute("data-group-id") + "']/preceding-
sibling::*[contains(@class, 'group-name')]")).getText();
```

```
            firstJoinButton.click();
            try {
                WebDriverWait wait = new WebDriverWait(driver,
Duration.ofSeconds(10));
                Alert alert = wait.until(ExpectedConditions.alertIsPresent());
                String alertText = alert.getText();
                alert.accept();
                assertTrue(alertText.contains("You have successfully joined the group
\"" + groupName + "\""));
            } catch (TimeoutException e) {
                System.out.println("No alert appeared after clicking join button");
            } catch (UnhandledAlertException e) {
                System.out.println("Alert not handled: " + e.getMessage());
            }
        } else {
            System.out.println("No join buttons found on the page.");
        }
    }

    @Then("user should see a success message")
    public void user_should_see_a_success_message() throws Exception {

        System.out.println("User successfully joined the group ");
        Thread.sleep(2000);
        driver.close();
        driver.quit();
    }
}
```

Screenshot



```
@join_group
Scenario: User joins a travel group successfully # src/test/resources/features/join_group.feature:4
Inside step - Browser is open
Oct 22, 2024 11:56:50 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 130, returning the closest version; found: 127; Please update to a Selenium version that supports CDP version 130
  Given Browser is open                          # definitions.join_group.browser_is_opened()
  And user is logged in                          # definitions.join_group.user_is_logged_in()
  And user is on join group page for group       # definitions.join_group.user_is_on_join_group_page()
  When user selects the group to join            # definitions.join_group.user_selects_the_group_to_join()
User successfully joined the group
```

Test report

Test Case 4	
Project Name: Explore Hub	
Join group Test Case	
Test Case ID: Test_4	**Test Designed By: Kiran Kurian Philip**
Test Priority (Low/Medium/High):	**Test Designed Date:22/10/24**
Module Name: Create Group	**Test Executed By: Nimmy Francis**
Test Title: Create group Test case	**Test Execution Date:22/10/24**
Description:	Test to join in a travel group by user
Pre-Condition: User has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)
1	Navigate to login page	N/A	Login page loads successfully	Login page loaded successfully	Pass
2	User is logged in	“kirankurian” for username and “1234” for password	Username and password fields are successfully filled in and logged in	Username and password fields are successfully filled in and logged in	Pass
3	User is on available group page	N/A	User landed on the available group page	User landed on the available group page	Pass
4	User clicks the join button	N/A	User clicks the join button	User clicks the join button	Pass
5	Close the browser window	N/A	Browser window is successfully closed	Browser window is successfully closed	Pass
Post-Condition: User successfully joined the travel group					

CHAPTER 6

IMPLEMENTATION

6.1INTRODUCTION

Implementation is the process of translating software designs and specifications into a functional software system. It involves the actual coding, testing, and integration of software components to create a working software product.

The implementation phase is a critical step in the software development life cycle (SDLC) because it involves turning the abstract ideas and plans into tangible software products that can be used by end-users. The implementation process involves several steps, including:

- Coding: Writing code in the programming language specified in the design phase.
- Testing: Ensuring that the code functions correctly and meets the design specifications. Testing can include unit testing, integration testing, and system testing.
- Debugging: Identifying and fixing any errors or defects in the code.
- Integration: Combining the individual software components into a cohesive system that functions as intended.
- Deployment: Installing the software on the target hardware and making it available to end-users.

During the implementation phase, it is important to follow coding and testing standards to ensure the software is reliable, maintainable, and scalable. It is also important to document the code and maintain version control to facilitate future maintenance and updates. Collaboration between developers and other stakeholders, such as testers, project managers, and end-users, can help to ensure that the software meets the needs and expectations of all stakeholders.

6.2 IMPLEMENTATION PROCEDURES

The implementation phase of the software development life cycle (SDLC) involves a series of procedures that must be followed to ensure that the software product is developed according to the design specifications and meets the quality and performance standards. Below are some of the typical procedures involved in the implementation phase:

- Coding: This involves writing the actual code for the software using the programming language specified in the design phase. Developers must follow coding standards and guidelines to ensure that the code is maintainable and scalable.
- Testing: Testing is a critical part of the implementation phase, and it involves testing the software at different levels, including unit testing, integration testing, and system testing. Testers must develop test cases and test scripts based on the design specifications to ensure that the software functions correctly.
- Debugging: During testing, issues and bugs may be identified in the software code. Developers

must identify and fix these errors or defects to ensure that the software functions as intended.

• Integration: After individual software components have been tested and debugged, they must be integrated into a cohesive system. The integration process must be carefully managed to ensure that the software functions as intended and that there are no conflicts or errors.

• Deployment: Once the software has been integrated, it must be deployed to the target hardware and made available to end-users. Deployment can be a complex process that involves several steps, including installation, configuration, and testing.

• Documentation: It is essential to document the software code and the implementation procedures to ensure that the software can be easily maintained and updated in the future. Documentation can include user manuals, technical documentation, and code comments.

• Version control: Version control is critical during the implementation phase to ensure that changes to the software code are tracked and managed properly. Version control tools such as Git or SVN can be used to manage code changes and track different versions of the software product.

By following these procedures, software developers can ensure that the software product is developed according to the design specifications and meets the quality and performance standards required by the end-users.

6.2.1 User Training

User training is a critical component of any software development project. It involves providing end-users with the necessary knowledge and skills to effectively use the software product. The goal of user training is to ensure that end-users can use the software product efficiently and effectively to meet their business needs.

Below are some common steps involved in user training:

• Identify training needs: Before training can begin, it is essential to identify the training needs of the end-users. This can be done by conducting a needs analysis, which involves assessing the current skills and knowledge of the end-users and identifying any gaps that need to be addressed.

• Develop training materials: Once the training needs have been identified, training materials can be developed. This can include user manuals, online tutorials, and training videos. The training materials should be designed to be easy to understand and follow, and should cover all the necessary features and functionalities of the software product.

• Conduct training sessions: Training sessions can be conducted in-person or online, depending on the needs of the end-users. The training sessions should be interactive and engaging, and should provide opportunities for end-users to ask questions and practice using the software product.

- Assess learning: After the training sessions, it is important to assess whether the end-users have learned the necessary skills and knowledge to effectively use the software product. This can be done through quizzes, assessments, or practical exercises.
- Provide ongoing support: Even after the initial training, it is important to provide ongoing support to end-users. This can include a helpdesk or support team that can provide assistance when needed, as well as updates to the training materials to reflect any changes or updates to the software product.

Effective user training can help to ensure that end-users are able to use the software product efficiently and effectively, which can lead to increased productivity and improved business outcomes.

6.2.2 Training on the Application Software

Training on application software is an essential component of software implementation. It is the process of providing end-users with the knowledge and skills necessary to operate and use the application software effectively. The goal of training is to ensure that end-users are able to utilize all the features and functionalities of the software to perform their tasks and achieve their goals. Below are some best practices for providing effective training on application software:

- Identify the target audience: Before developing training materials, it is essential to identify the target audience and their specific needs. Different groups of end-users may require different levels of training or different types of training materials.
- Develop training materials: Training materials can include user manuals, online tutorials, videos, and live training sessions. The training materials should be designed to be easy to understand and follow, and should cover all the necessary features and functionalities of the software product.
- Use interactive training methods: Interactive training methods such as demonstrations, simulations, and hands-on exercises can help end-users to better understand and retain the training materials.
- Provide ongoing support: After the initial training, it is important to provide ongoing support to end-users. This can include a helpdesk or support team that can provide assistance when needed, as well as updates to the training materials to reflect any changes or updates to the software product.
- Evaluate training effectiveness: It is important to evaluate the effectiveness of the training by assessing the end-users' knowledge and skills before and after the training. This can help identify

areas that may require additional training or support.

• Provide refresher training: As the software product evolves, it is important to provide refresher training to end-users to ensure that they are up-to-date with the latest features and functionalities. Effective training on application software can help to ensure that end-users are able to use the software product effectively, which can lead to increased productivity and improved business outcomes.

6.2.3 System Maintenance

System maintenance is the process of keeping a software system up-to-date and functioning properly after it has been deployed. It involves a variety of activities that are designed to ensure that the system remains stable, secure, and efficient over time. Some of the key activities involved in system maintenance include:

• Updates and patches: One of the most important aspects of system maintenance is keeping the system up-to-date with the latest software updates and security patches. These updates and patches are released periodically by software vendors to fix bugs, add new features, and address security vulnerabilities. Applying these updates in a timely manner is essential to ensure that the system remains secure and reliable.

• Performance monitoring: Regular performance monitoring is another key aspect of system maintenance. This involves tracking system performance metrics such as CPU usage, memory utilization, and network traffic to identify potential issues and optimize system performance.

• Backup and recovery: Backing up system data and ensuring that it can be recovered in the event of a system failure or data loss is an important part of system maintenance. This involves creating regular backups of system data and testing the recovery process to ensure that it works as expected.

• Security management: Maintaining system security is critical to protecting sensitive data and preventing unauthorized access to the system. This involves implementing security controls such as firewalls, antivirus software, and access controls, and regularly monitoring the system for potential security breaches.

• Hardware maintenance: Ensuring that hardware components such as servers, storage devices, and network devices are functioning properly is another key aspect of system maintenance. This involves performing regular maintenance tasks such as cleaning, repairs, and upgrades to ensure that the hardware components remain in good working condition.

Effective system maintenance is critical to ensuring that a software system remains reliable, secure, and efficient over time. By implementing a proactive system maintenance strategy, organizations can minimize downtime, prevent system failures, and optimize system performance.

6.2.4 Hosting

Hosting refers to the process of storing and serving website files on a remote server, which can be accessed by visitors over the internet. When you create a website, you need to have it hosted on a server so that it can be available for others to view. There are various types of hosting options available such as shared hosting, dedicated hosting, VPS hosting, cloud hosting, etc. The choice of hosting depends on the size of the website, its traffic volume, and the level of control and flexibility required by the owner.

Render

Render is a cloud hosting service that provides a streamlined platform for deploying web applications, APIs, static sites, and databases. It simplifies the process of hosting applications by automating tasks such as server setup, scaling, and load balancing, which allows developers to focus on building their applications rather than managing infrastructure. Render supports various languages and frameworks, including Django, and offers seamless integration with Git for continuous deployment.

Procedure for hosting a website on render:

Step 1: Login to your render account

Step 2: Create a new project

Step 3: Create a requirements.txt file which will include all your dependencies to be installed.

Step 4: Upload the content which is to be hosted into GitHub

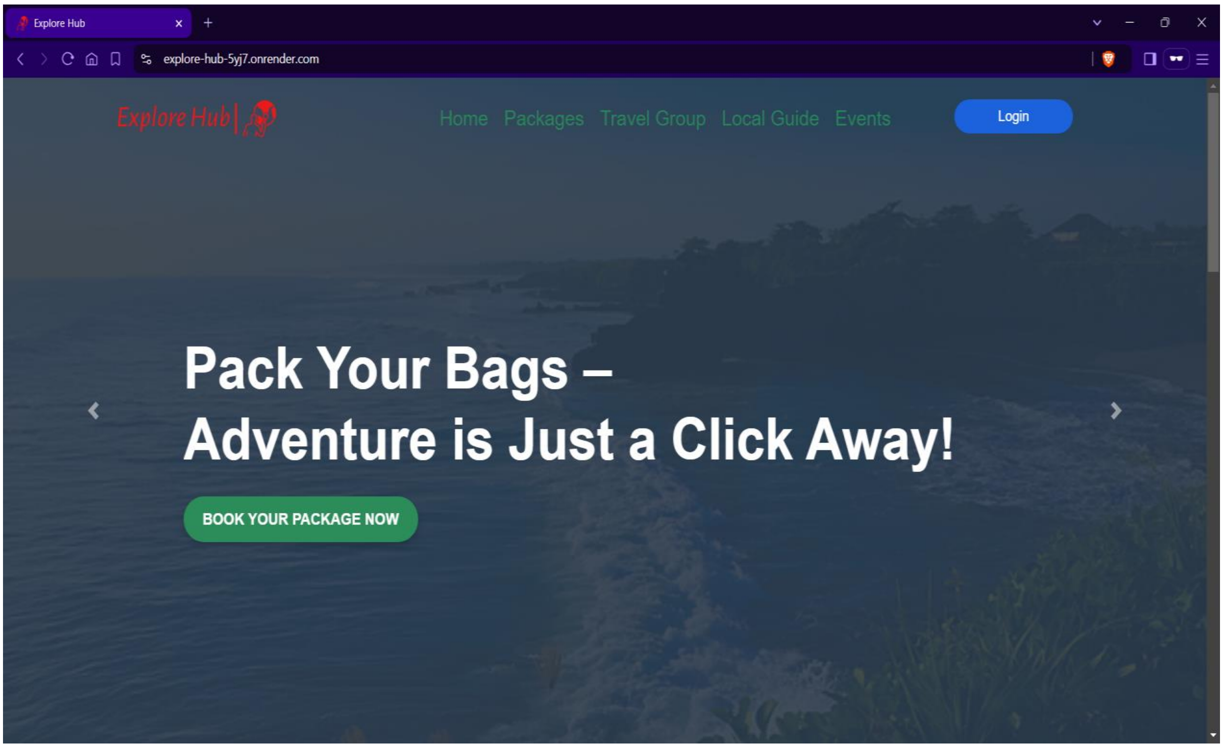
Step 5: Change the setting of your project to include the host also and then deploy.

Hosted Website: Render

Hosted Link: https://explore-hub-5yj7.onrender.com/

Explore Hub

Screenshot



Pack Your Bags –
Adventure is Just a Click Away!

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The Explore Hub project successfully delivers a dynamic platform for users to connect, form travel groups, and plan trips together based on shared interests. By allowing users to create or join groups and communicate with fellow travellers, the platform fosters a collaborative and enjoyable travel experience. The integration of travel agency packages ensures that users can easily find organized trips while maintaining the flexibility to customize group travel. Through careful design and robust functionality, this project demonstrates a practical and user-friendly solution for modern travel planning.

This mini-project has laid the foundation for a scalable travel platform that can be further enhanced with additional features such as real-time booking, advanced user communication tools, and personalized recommendations. The project has also provided valuable insights into system design, database management, and user experience considerations, positioning it as a strong base for future development in the travel domain.

7.2 FUTURE SCOPE

The Explore Hub project has the potential for significant expansion and enhancement. Some possible future developments include:

- Personalized Recommendations: Using machine learning algorithms to recommend travel groups, destinations, or packages to users based on their preferences, travel history, and behaviour.
- Travel Agency Dashboard Enhancements: Offering more detailed analytics and tools for travel agencies to manage bookings, monitor user activity, and promote their travel packages more effectively.
- User Reviews and Ratings: Enabling users to rate their experiences with different travel groups and packages, helping others make informed decisions based on past travellers' feedback.

These future enhancements will broaden the platform's appeal, offering a richer and more personalized experience for both users and travel agencies, ensuring its relevance in the competitive travel industry.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Django for Beginners: Build Websites with Python and Django by William S. Vincent
- Python Crash Course by Eric Matthes

WEBSITES:

- https://docs.djangoproject.com
- https://docs.python.org/3/
- https://razorpay.com/docs/
- https://getbootstrap.com

CHAPTER 9

APPENDIX

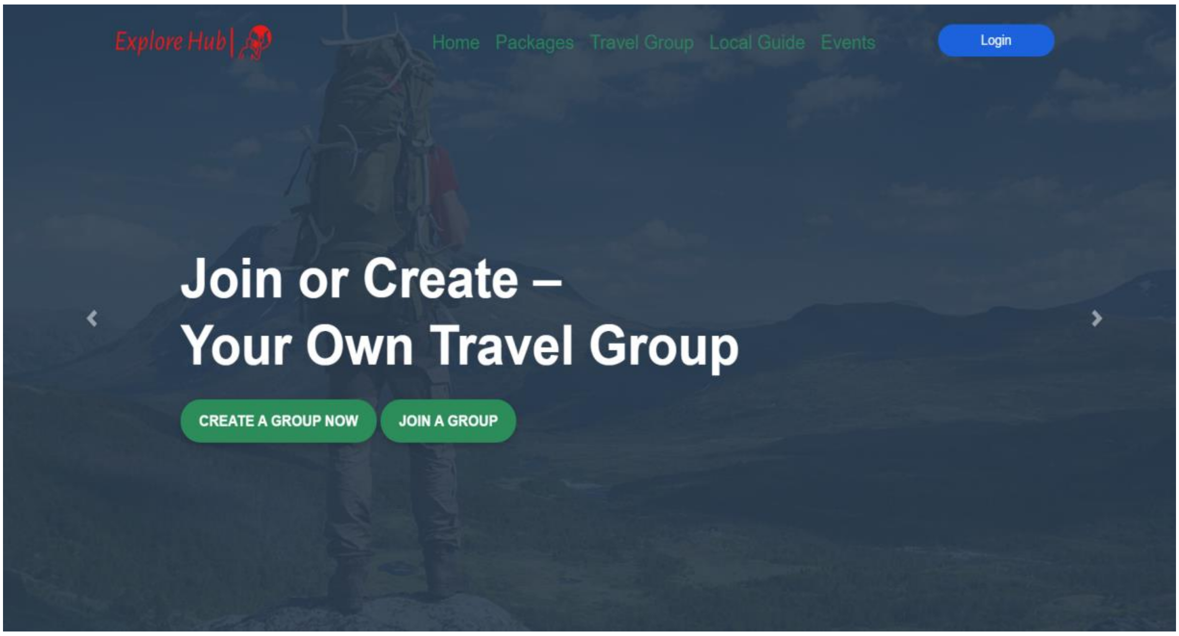
9.1 Sample Code

Package Listing

```
{% extends 'layout_main.html' %}
{% load static %}
{% block content %}
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{% static 'assets/css/package_style.css' %}">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick-theme.css">
    <script src="{% static "assets/vendors/jquery/jquery-3.4.1.js" %}"></script>
    <script src="https://cdn.jsdelivr.net/npm/slick-carousel@1.8.1/slick/slick.min.js"></script>
</head>

<div class="body">
    <div class="search-container">
        <form class="search-form">
            <input type="text" id="search-bar" placeholder="Search" aria-label="Search Packages">
            <button type="submit" class="search-button">Search</button>
        </form>
    </div>

    <div class="package-list" id="package-list">
        {% for package in packages %}
            <div class="package-box">
                <div class="slider-container">
                    {% for image in package.package_images.all %}
                        <div class="slide">
                            <img src="{{ image.image.url }}" alt="{{ package.title }}">
                        </div>
                    {% empty %}
                        <div class="slide">
                            <img src="{% static 'assets/images/default_image.jpg' %}" alt="Default Image">
                        </div>
                    {% endfor %}
                </div>
                <h4>{{ package.title }}</h4>
                <p>{{ package.description }}</p>
                <div class="price-info">
                    {% if package.discount_percentage %}
                        <p class="original-price">Price: <span class="strikethrough">{{ package.price
}}</span></p>
                        <p class="discounted-price">Discounted Price: {{
package.discounted_price|floatformat:2 }}</p>
                        <span class="discount-badge">You Save: {{ package.you_save|floatformat:2
}}</span><br>
                    {% else %}
                        <p class="price">Price: {{ package.price }}</p>
                    {% endif %}
                </div>
                <p>No. of Days: {{ package.duration }}</p>
                <p>Origin: {{ package.origin }}</p>
                <p>Destination: {{ package.destination }}</p>
                <p class="charges">Cancellation: {{ package.cancellation|yesno:"Yes,No" }}</p>
                <a href="{% url 'package_detail' package.package_id %}" class='btn_book'>View
Details</a>
            </div>
        {% empty %}
            <p>No packages available at the moment.</p>
        {% endfor %}
    </div>
</div>

<script type="text/javascript">
    $(document).ready(function(){
        $('.slider-container').slick({
            dots: true,
            infinite: true,
            speed: 500,
            slidesToShow: 1,
            adaptiveHeight: true,
            autoplay: true,
            autoplaySpeed: 2000,
            arrows: true,
        });

        $('#search-bar').on('keyup', function() {
            const query = $(this).val();

            $.ajax({
                url: "{% url 'package_search' %}",
                data: { 'query': query },
                success: function(data) {
                    $('#package-list').html(data);
                }
            });
        });
    });
</script>

<style>

</style>

{% endblock %}
```

9.2 Screen Shots

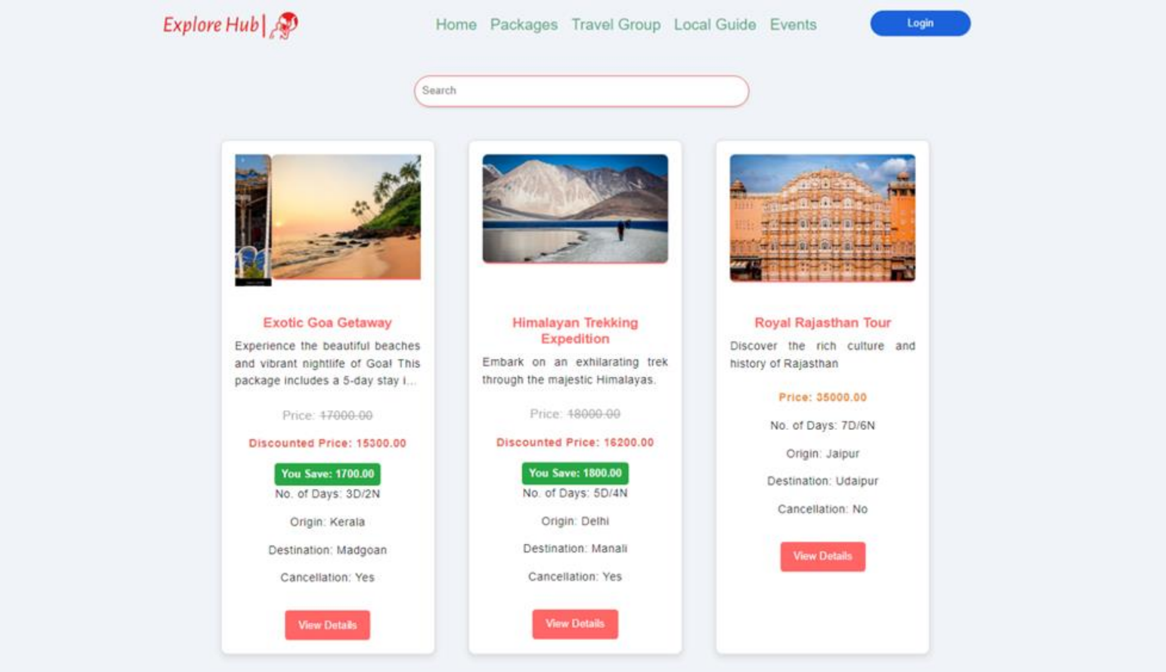
Home Page



Join or Create –
Your Own Travel Group

Fig 9.2. 1: Home page

Package Listing



Exotic Goa Getaway

Experience the beautiful beaches and vibrant nightlife of Goa! This package includes a 5-day stay i...

Price: ~~17000.00~~

Discounted Price: 15300.00

You Save: 1700.00

No. of Days: 3D/2N

Origin: Kerala

Destination: Madgoan

Cancellation: Yes

Himalayan Trekking Expedition

Embark on an exhilarating trek through the majestic Himalayas.

Price: ~~18000.00~~

Discounted Price: 16200.00

You Save: 1800.00

No. of Days: 5D/4N

Origin: Delhi

Destination: Manali

Cancellation: Yes

Royal Rajasthan Tour

Discover the rich culture and history of Rajasthan

Price: 35000.00

No. of Days: 7D/6N

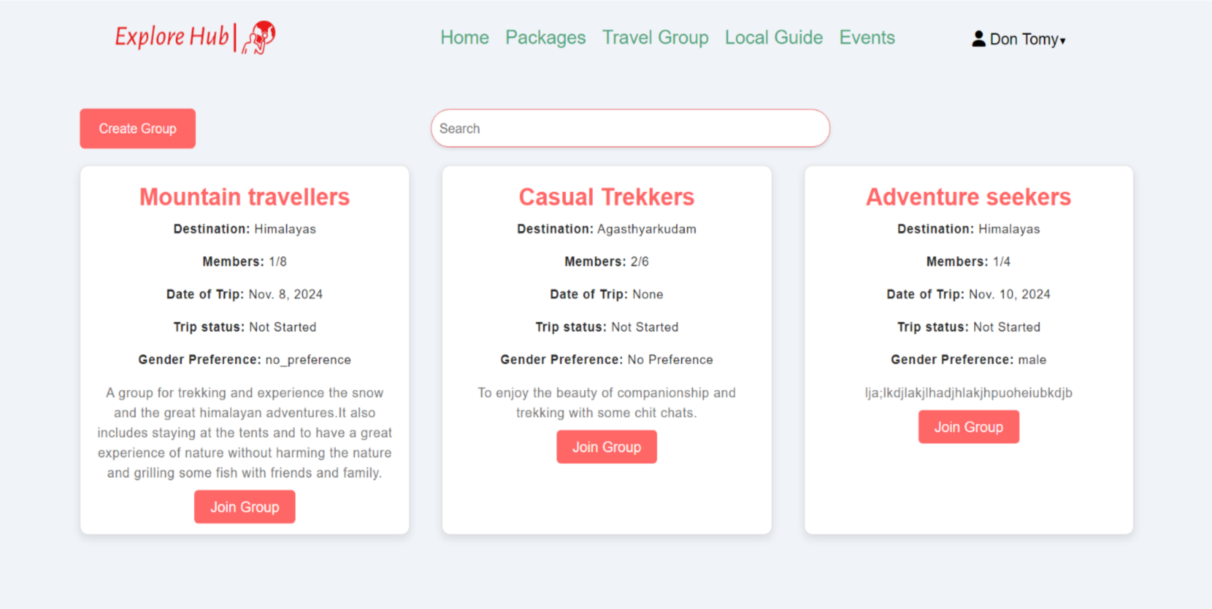
Origin: Jaipur

Destination: Udaipur

Cancellation: No

Fig 9.2. 2: Package Listing

Group Listing



Mountain travellers

Destination: Himalayas

Members: 1/8

Date of Trip: Nov. 8, 2024

Trip status: Not Started

Gender Preference: no_preference

A group for trekking and experience the snow and the great himalayan adventures.It also includes staying at the tents and to have a great experience of nature without harming the nature and grilling some fish with friends and family.

Join Group

Casual Trekkers

Destination: Agasthyarkudam

Members: 2/6

Date of Trip: None

Trip status: Not Started

Gender Preference: No Preference

To enjoy the beauty of companionship and trekking with some chit chats.

Join Group

Adventure seekers

Destination: Himalayas

Members: 1/4

Date of Trip: Nov. 10, 2024

Trip status: Not Started

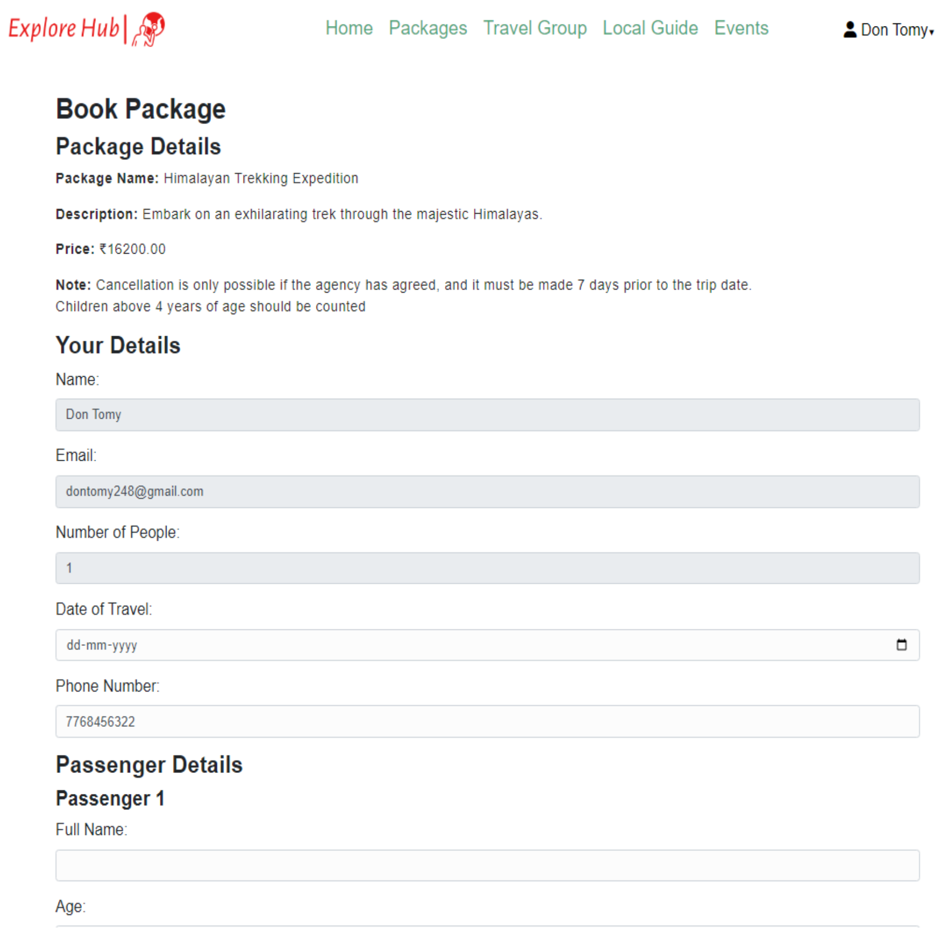
Gender Preference: male

lja;lkdjlakjlhadjhlakjhpuoheiubkdjb

Join Group

Fig 9.2. 3: Group Listing

Package booking confirmation



Book Package

Package Details

Package Name: Himalayan Trekking Expedition

Description: Embark on an exhilarating trek through the majestic Himalayas.

Price: ₹16200.00

Note: Cancellation is only possible if the agency has agreed, and it must be made 7 days prior to the trip date.
Children above 4 years of age should be counted

Your Details

Name:
Don Tomy

Email:
dontomy248@gmail.com

Number of People:
1

Date of Travel:
dd-mm-yyyy

Phone Number:
7768456322

Passenger Details

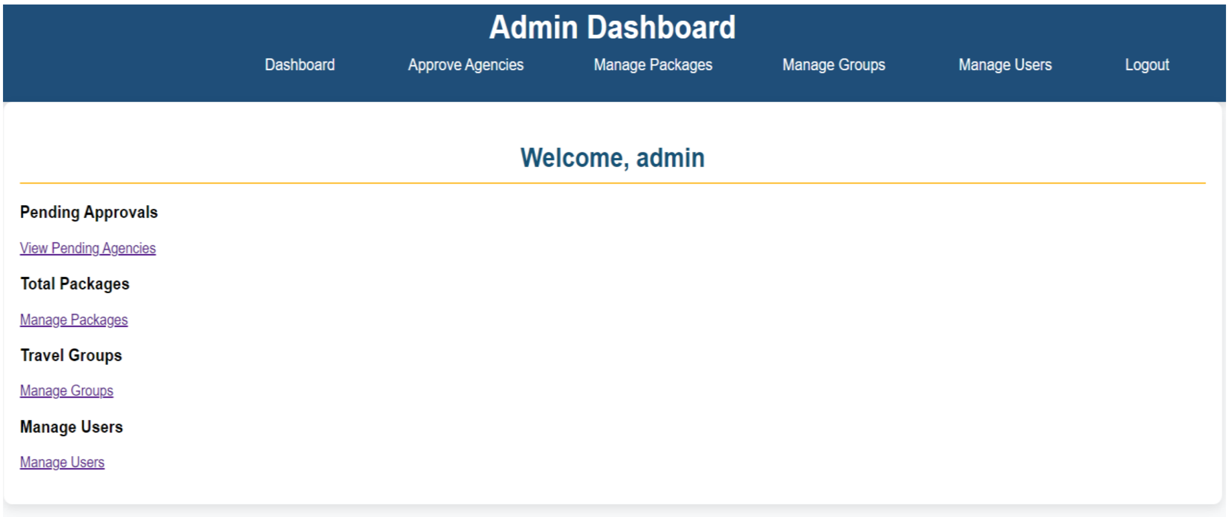
Passenger 1

Full Name:

Age:

Fig 9.2. 4: Booking Confirmation

Admin home page



Admin Dashboard

Welcome, admin

Pending Approvals

View Pending Agencies

Total Packages

Manage Packages

Travel Groups

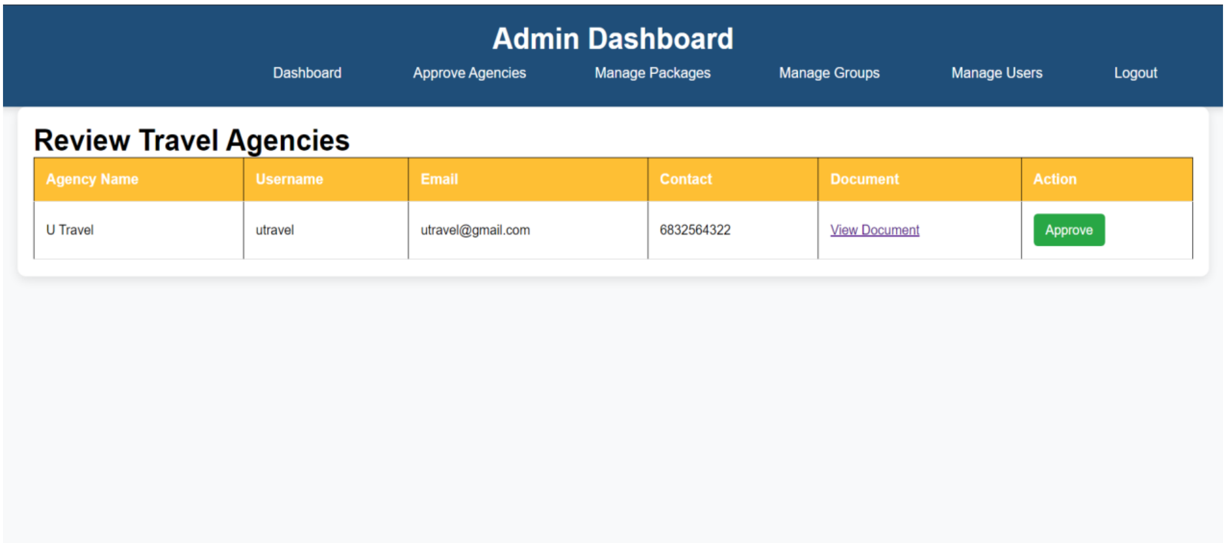
Manage Groups

Manage Users

Manage Users

Fig 9.2. 5: Admin Home Page

Admin approve agency



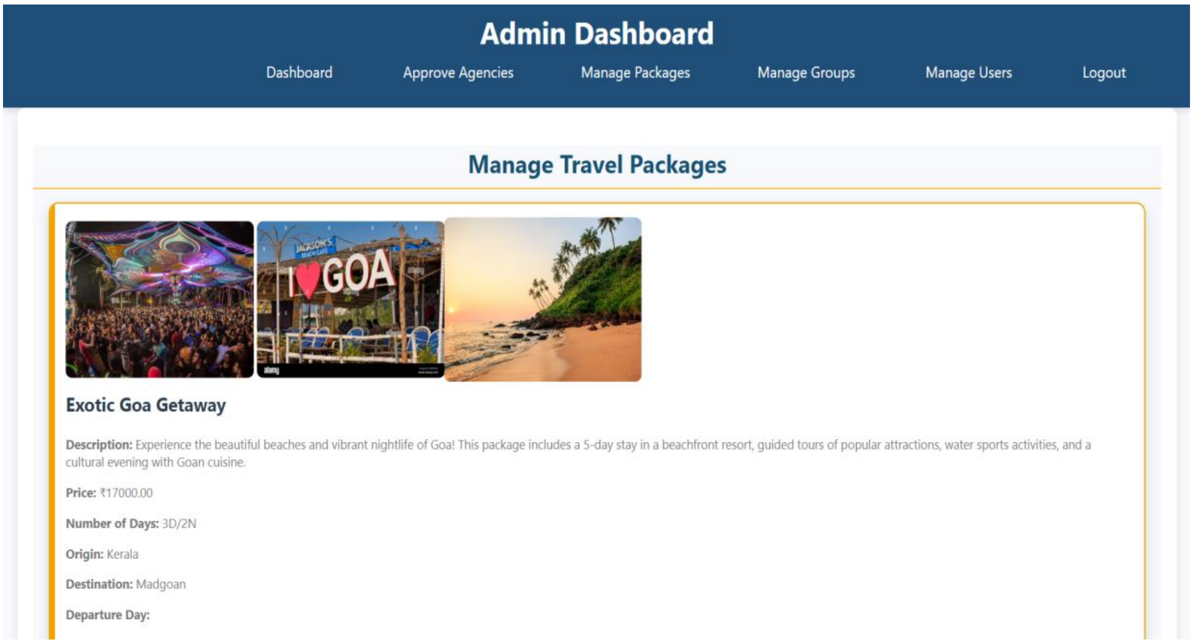
Admin Dashboard

Review Travel Agencies

Agency Name	Username	Email	Contact	Document	Action
U Travel	utravel	utravel@gmail.com	6832564322	View Document	Approve

Fig 9.2. 6: Admin approve agency

Admin manage travel packages



Manage Travel Packages

Exotic Goa Getaway

Description: Experience the beautiful beaches and vibrant nightlife of Goa! This package includes a 5-day stay in a beachfront resort, guided tours of popular attractions, water sports activities, and a cultural evening with Goan cuisine.

Price: ₹17000.00

Number of Days: 3D/2N

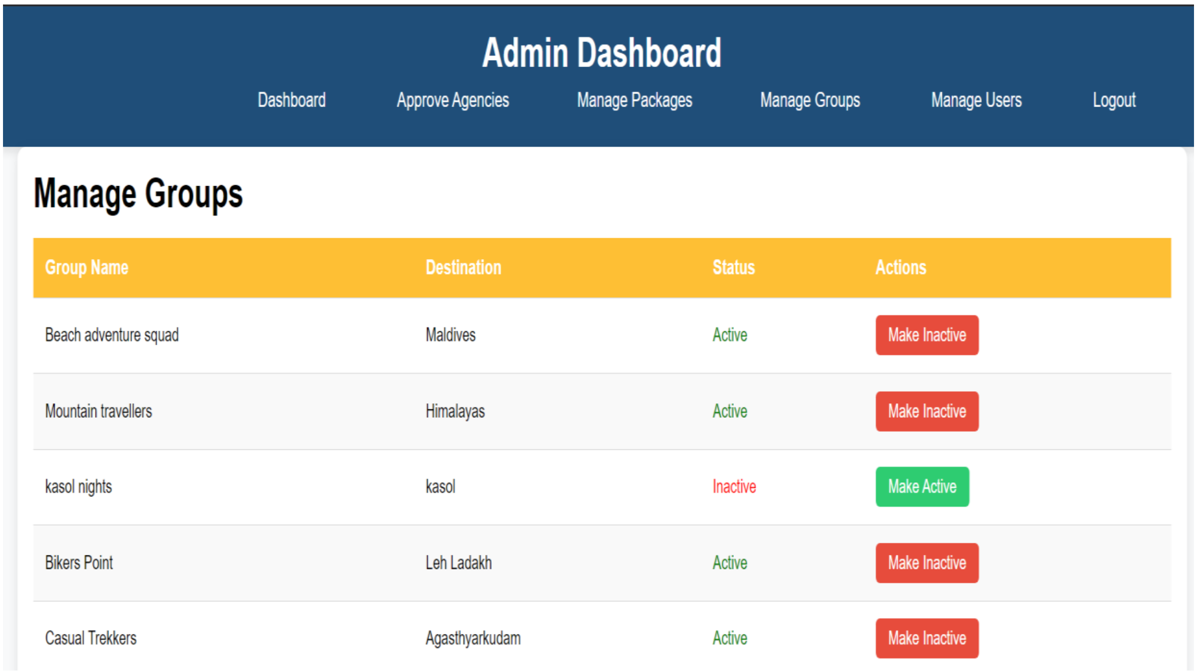
Origin: Kerala

Destination: Madgoan

Departure Day:

Fig 9.2. 7: Admin manage travel packages

Admin manage groups

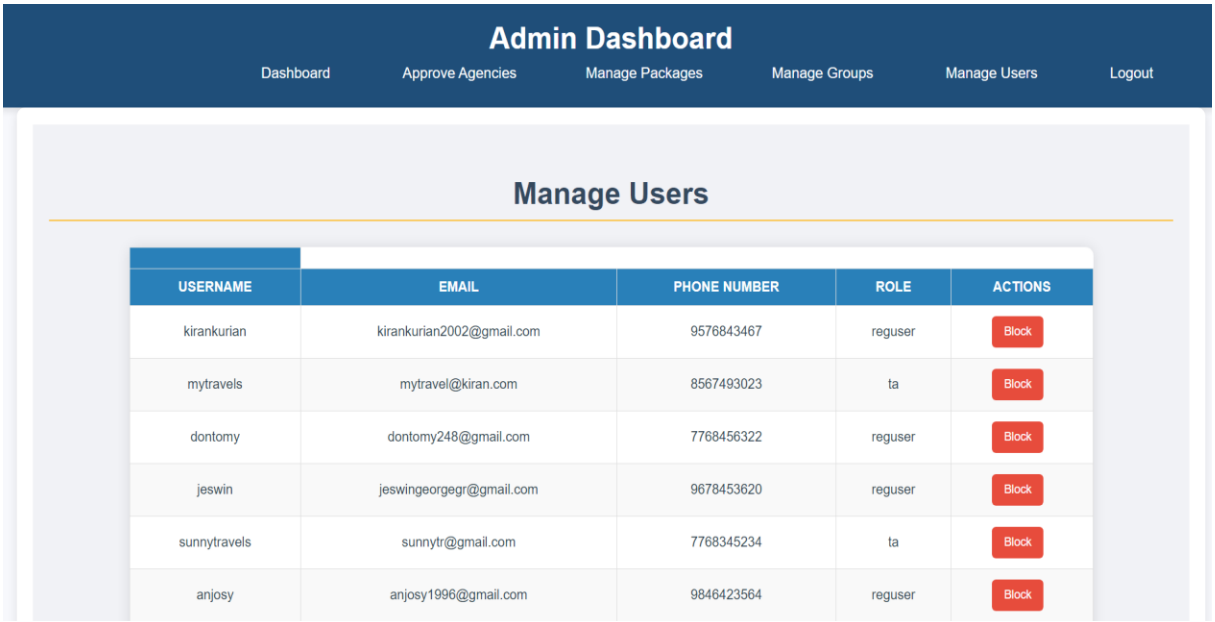


Manage Groups

Group Name	Destination	Status	Actions
Beach adventure squad	Maldives	Active	Make Inactive
Mountain travellers	Himalayas	Active	Make Inactive
kasol nights	kasol	Inactive	Make Active
Bikers Point	Leh Ladakh	Active	Make Inactive
Casual Trekkers	Agasthyarkudam	Active	Make Inactive

Fig 9.2. 8: Admin manage groups

Admin manage users

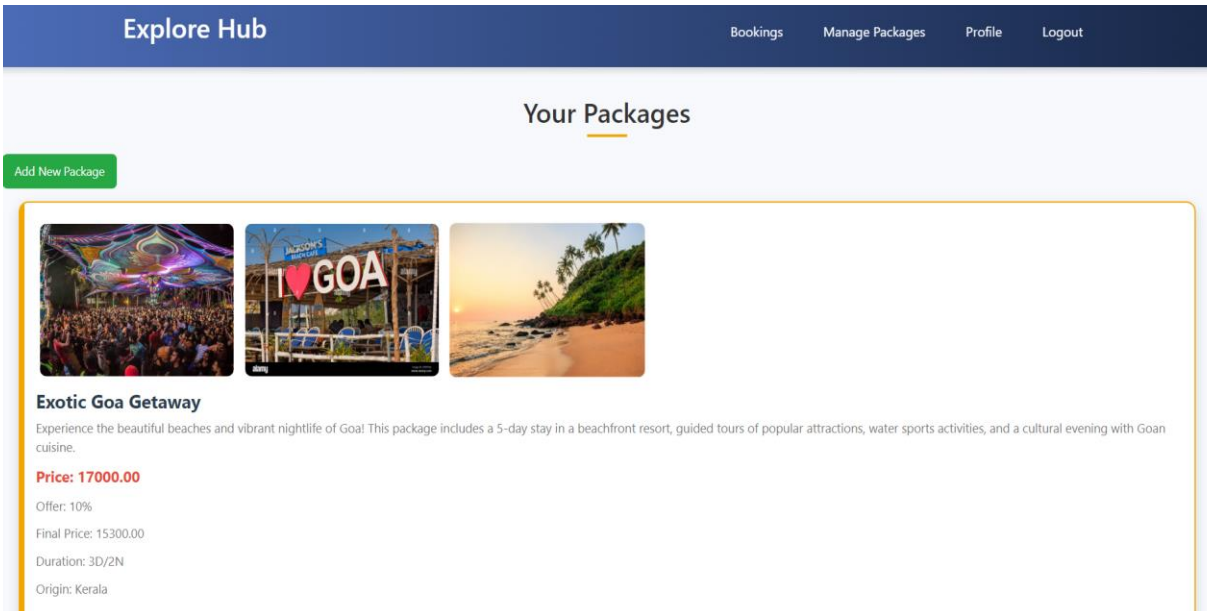


Manage Users

USERNAME	EMAIL	PHONE NUMBER	ROLE	ACTIONS
kirankurian	kirankurian2002@gmail.com	9576843467	reguser	Block
mytravels	mytravel@kiran.com	8567493023	ta	Block
dontomy	dontomy248@gmail.com	7768456322	reguser	Block
jeswin	jeswingeorgegr@gmail.com	9678453620	reguser	Block
sunnytravels	sunnytr@gmail.com	7768345234	ta	Block
anjosy	anjosy1996@gmail.com	9846423564	reguser	Block

Fig 9.2. 9: Admin manage users

Travel agency home page



Your Packages

Exotic Goa Getaway

Experience the beautiful beaches and vibrant nightlife of Goa! This package includes a 5-day stay in a beachfront resort, guided tours of popular attractions, water sports activities, and a cultural evening with Goan cuisine.

Price: 17000.00

Offer: 10%

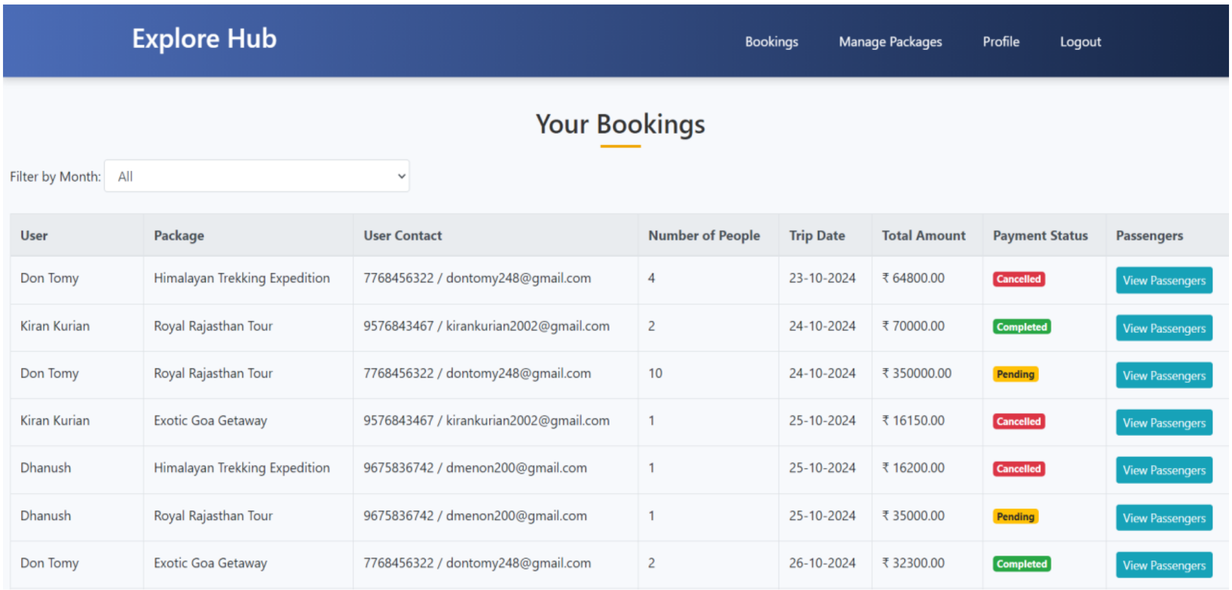
Final Price: 15300.00

Duration: 3D/2N

Origin: Kerala

Fig 9.2. 10: Travel agency home page

Travel agency booking’s view



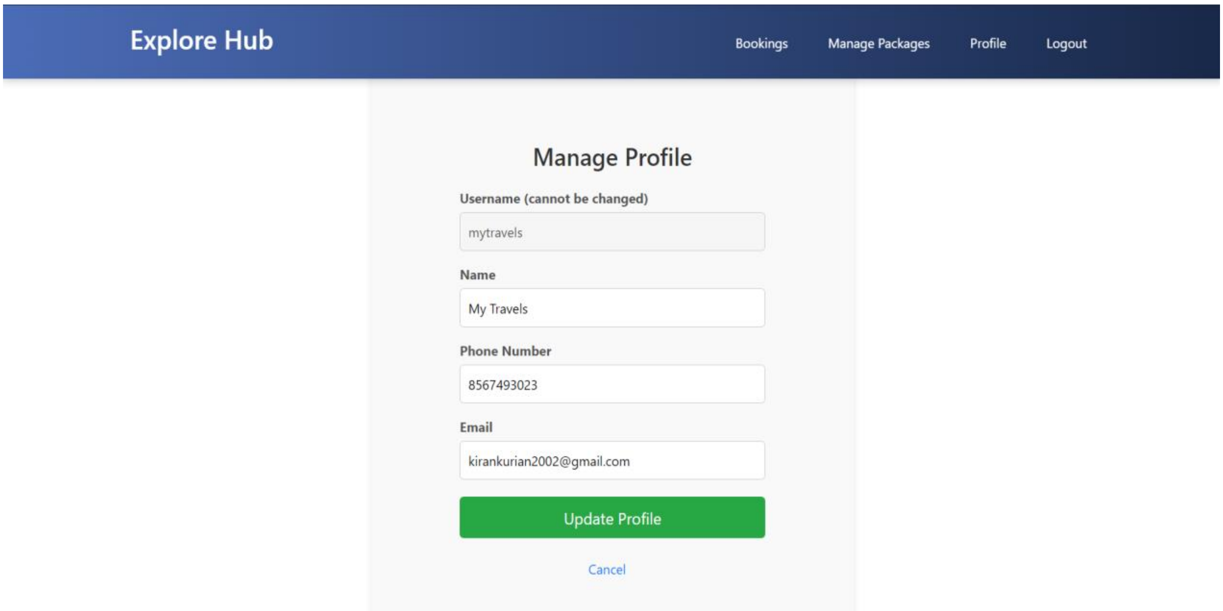
Your Bookings

Filter by Month: All

User	Package	User Contact	Number of People	Trip Date	Total Amount	Payment Status	Passengers
Don Tomy	Himalayan Trekking Expedition	7768456322 / dontomy248@gmail.com	4	23-10-2024	₹ 64800.00	Cancelled	View Passengers
Kiran Kurian	Royal Rajasthan Tour	9576843467 / kirankurian2002@gmail.com	2	24-10-2024	₹ 70000.00	Completed	View Passengers
Don Tomy	Royal Rajasthan Tour	7768456322 / dontomy248@gmail.com	10	24-10-2024	₹ 350000.00	Pending	View Passengers
Kiran Kurian	Exotic Goa Getaway	9576843467 / kirankurian2002@gmail.com	1	25-10-2024	₹ 16150.00	Cancelled	View Passengers
Dhanush	Himalayan Trekking Expedition	9675836742 / dmenon200@gmail.com	1	25-10-2024	₹ 16200.00	Cancelled	View Passengers
Dhanush	Royal Rajasthan Tour	9675836742 / dmenon200@gmail.com	1	25-10-2024	₹ 35000.00	Pending	View Passengers
Don Tomy	Exotic Goa Getaway	7768456322 / dontomy248@gmail.com	2	26-10-2024	₹ 32300.00	Completed	View Passengers

Fig 9.2. 11: Travel agency booking view

Travel agency update profile



Manage Profile

Username (cannot be changed): mytravels

Name: My Travels

Phone Number: 8567493023

Email: kirankurian2002@gmail.com

Fig 9.2. 12: Travel agency update

9.3 Git Log

```
commit ffc6a745b78ca147a8a2a7c2d2acd4c77cb9e52b
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Oct 14 22:37:04 2024 +0530

    minor updates

commit 49f375f356b28585e95dcfd1fb58c1158ac77101
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sat Oct 12 15:21:14 2024 +0530

    Travel agency booking view

commit 9a0fed14c63e8f74d908a83772a4ed5295809c29
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sat Oct 12 13:58:49 2024 +0530

    Cancel booking

commit 65deb7a6242c7d31c103ee49de0e2a7747fd3719
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Fri Oct 11 23:09:20 2024 +0530

    discount for travel packages

commit 00bc312f69ca41a4fbdba91b281c9d06902d3453
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Wed Oct 9 23:57:33 2024 +0530

    booking with payment

commit a5948bfb486f589835844095e069dbb9ef4e9d0e
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Oct 7 18:26:48 2024 +0530
```

```
    #

commit ef22407423f8b509cb567de3c73583c75d533a9d
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Oct 7 18:18:12 2024 +0530

    some minor errors

commit 91190513571c6aead94737e8bbd87de6aa66a89e
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Oct 7 16:06:53 2024 +0530

    group creation and joining

commit 2ebaf453e70db46a60ad330f8598df3709779f5f
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Thu Oct 3 22:09:17 2024 +0530

    Group creation

commit 36b8fd42c60750c999e03dc558de38f7f162f70f
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Thu Oct 3 18:16:38 2024 +0530

    updates to admin module in managing

commit e371ce772e0b734ed1befe61dc044b68dfd204e5
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 30 12:12:15 2024 +0530

    minor updates

commit 3395aa8bc9ba115f514f48bbec4e44ac336bf051
```

```
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 30 11:45:02 2024 +0530

    add session

commit 0075fc7dc1ca00198b13ccb97c216b1c926a269f
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Wed Sep 25 11:35:26 2024 +0530

    minor validations

commit f9078a00bff8d36b31f1f211abb4af3069d40161
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Tue Sep 24 22:49:01 2024 +0530

    minor bugs

commit b5e05fe4355117634b92e2f547b10a21b22fada9
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Tue Sep 24 18:15:45 2024 +0530

    Google Login

commit 67ef363c433542a26a71c57bf293c426222050a7
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Tue Sep 24 08:17:46 2024 +0530

    #

commit 990dc7c1a4542219a091f5935841eca55097a261
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 23 21:56:02 2024 +0530

    #
```

```
commit e0b1260f3bba1ceabd9f0c7b340e472256ed8d6b
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 23 21:50:14 2024 +0530

    #

commit 8a184298687d92208b943b977674b382b0cbf805
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 23 20:36:45 2024 +0530

    #

commit fb38dfbaf55649d1a2a572788a60826911762601
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 23 19:28:38 2024 +0530

    added documents

commit a4f59068b3a750d8bf77899c41543b093726a397
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 23 15:13:50 2024 +0530

    Travel agency profile update

commit 7540b9f2c940e0cd4c0abacbbfe253e8e28a509e
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 23 14:01:04 2024 +0530

    design error correction

commit 3fbd39546614fd89333281f960d7ccfef87ba53c
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 23:17:57 2024 +0530
```

```
    added phone number validation

commit 3872d65092cd16cac0340a0995d381f2131a4945
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 23:05:52 2024 +0530

    manage user by admin

commit 908c4c34adbee4f275589377244c2461c82edba0
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 16:05:04 2024 +0530

    pacakage detail viewing

commit 00540fe704df51b769100b935d8955eb25efef08
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 13:09:39 2024 +0530

    Package Updation

commit a398bc24c9d17b4c96a8b32c58b6684755fda94c
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 12:36:13 2024 +0530

    Package Updation

commit e777074d334c8153217d4015906a0e94adf58bfc
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 10:38:54 2024 +0530

    Package Insertion Updated

commit 58e476c9e2169b3fe85d4da9406a6fbd5dec4772
```

```
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Sun Sep 22 08:07:53 2024 +0530

    Forgot Password Complete

commit 31fec9773ad79d7b2838ccdff7e80d4282e504ed
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Thu Sep 19 19:36:08 2024 +0530

    update

commit b8d3fc7a5d053ea9972c06285c08b5d43c9bc315
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Thu Sep 19 17:55:18 2024 +0530

    forgot password

commit 8fc1c0c43d9848db8b70db0f97c19d1ca43e33b7
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Fri Sep 6 14:53:15 2024 +0530

    .

commit 3c0b90a6bf17a2a147fe27f2dc41bf8ddb0beba3
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Fri Sep 6 12:42:05 2024 +0530

    some minor design changes

commit 18b761af5f5ecd14529bedc2161e0a4170c27451
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Wed Sep 4 21:00:22 2024 +0530

    Bug in admin management solved
```

```
commit 119ef3dcbfd8ac24cd59fda70f4088ac34b59584
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Wed Sep 4 20:56:11 2024 +0530

    /

commit 97bcc9a7003e2bdfd54862d097ba09c505cbfaf0
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Wed Sep 4 20:51:45 2024 +0530

    admin manage packages

commit 3a8c78a87f8caafbda00764d61c7f50ebe7e7228
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Tue Sep 3 07:38:23 2024 +0530

    Add admin to manage packages

commit ce655ae386bbd4737e13c4d2d5272388d9642fae
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 2 14:52:55 2024 +0530

    Deleting Unnecessary

commit 8660a61807e15a3f3e13471e51b55e60247f8df8
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 2 14:51:40 2024 +0530

    Arranging

commit 0e01631b38e647a44f564ccab87cce2eaea793fd
Author: kiran-kurian <kirankurian02@gmail.com>
Date:   Mon Sep 2 14:47:27 2024 +0530
```

```
    First Commit

commit f4f5a84fb55175f035c96ce75a695d780b8bcde4
Author: Kiran Kurian Philip <kirankurian02@gmail.com>
Date:   Tue Aug 27 21:19:41 2024 +0530

    Add files via upload

commit 1022bbe87517b03fec8f6436339f75bd632fa574
Author: Kiran Kurian Philip <kirankurian02@gmail.com>
Date:   Tue Aug 27 09:43:33 2024 +0530

    Add files via upload

commit c92713c66d02598370e794d83c1597be7570d717
Author: Kiran Kurian Philip <kirankurian02@gmail.com>
Date:   Fri Aug 23 14:27:39 2024 +0530

    Add files via upload

commit 9bd9d2be29d2180d69a13734a6cec7d522104ad7
Author: Kiran Kurian Philip <kirankurian02@gmail.com>
Date:   Fri Aug 23 14:25:53 2024 +0530

    Delete UML DIAGRAMS.pdf

commit 755772461892e0850398fb7c740d444dc326dfd9
Author: Kiran Kurian Philip <kirankurian02@gmail.com>
Date:   Fri Aug 23 14:25:14 2024 +0530

    Add files via upload
```