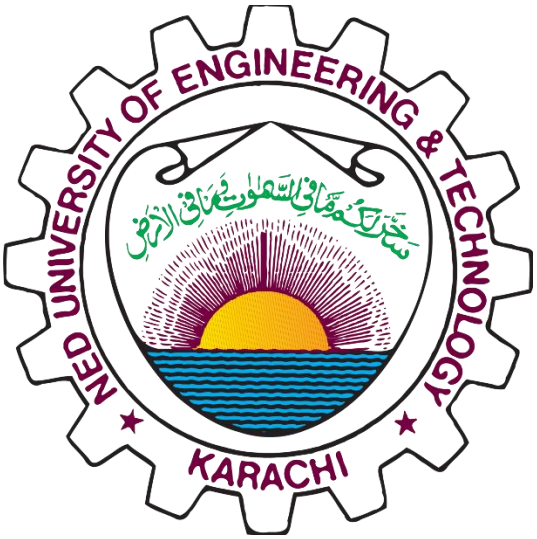
TOURISM MANAGEMENT SYSTEM

Project Group ID: 5

ZAID AHAMED	**CS- 22154**
MUHAMMED	**CS- 22155**
FATHIMA AMNA	**CS- 22156**



CS-222

DATABASE MANAGEMENT SYSTEM

CEP REPORT

Submitted to: Sir Ali Akhtar

Submission Date: 10.07.2024

Department of Computer and Information Systems Engineering

ABSTRACT

The Tourism Management System is a web-based application developed to manage tourism activities effectively. This project was undertaken for a fictitious travel agency named "Blue Star" with the objective of addressing common challenges in the tourism industry, such as inefficient booking processes, complex tour package management, and user information handling.

The system provides a platform for users to register, browse available tour packages, book tours, and overview their bookings. Administrators can create and manage tour packages, oversee user information and booking activities through a dedicated admin dashboard. The application leverages fundamental CRUD (Create, Read, Update and Delete) operations using a MySQL database to ensure data consistency and reliability.

The project involved several key phases, including requirement formulation and analysis, conceptual design using an Entity-Relationship (E-R) model, logical design through normalization, and implementation design with SQL statements. The front-end application was developed using [HTML,CSS, JAVASCRIPT], and features a user-friendly interface for seamless interaction.

The implementation of the system demonstrates significant improvements in booking efficiency, tour package management, and user data handling. Additionally, the project highlights the importance of database security and suggests future enhancements to further optimize the system.

Overall, the Tourism Management System project showcases the practical application of database management principles in solving real-world problems in the tourism industry, providing a robust and efficient solution for managing tourism activities.

TABLE OF CONTENTS

INTRODUCTION

Tourism Management System is a web-based application designed to manage and streamline tourism activities efficiently. This system offers a comprehensive platform for users and administrators. The primary goal is to automate and simplify the management of tour bookings, user registrations, and tour package creation, thereby enhancing the overall user experience.

The system provides distinct functionalities for different user roles:

- **Users:** Users can register on the platform, browse available tour packages, book tours, and overview their bookings. The intuitive interface ensures that users can effortlessly navigate through the various options, making the process of booking and managing tours straightforward and user-friendly.
- **Administrators:** Administrators have the ability to create and manage tour packages, oversee booking activities and user information. This role is crucial for maintaining the system's integrity, ensuring that all information is current, accurate, and reliable.

Source of the Project

- This project is based on the hypothetical requirements of a fictitious travel agency called "Blue Star" The objective was to design and develop a system that addresses common challenges faced by such organizations, specifically those related to managing tour bookings and user information efficiently.

Problem Statement

The tourism industry often struggles with managing bookings, updating tour packages, and handling user information manually, leading to errors, inefficiencies, and customer dissatisfaction. The main issues identified at Blue Star include:

- **Inefficient Booking Process:** Manual handling of bookings often results in double booking, errors, and delays.
- **Complex Tour Package Management:** Updating and managing tour packages manually is time-consuming and prone to errors.
- **User Information Management:** Storing and updating user information manually can lead to data inconsistencies and security concerns.
- **Limited Accessibility:** Without an online platform, users find it challenging to access information and make bookings at their convenience.

Objectives

The Tourism Management System aims to resolve these issues by:

- **Automating Booking Processes:** Providing a reliable and error-free booking system.
- **Streamlining Package Management:** Allowing administrators to create and update tour packages effortlessly.
- **Enhancing Data Management:** Ensuring consistent management of user information.
- **Improving Accessibility:** Offering a web-based platform accessible to users anytime, anywhere.

System Features

The key features of the Tourism Management System include:

- **User Registration and Management:** Secure registration process.
- **Tour Booking:** Easy and intuitive booking process with instant confirmation.
- **Admin Dashboard:** Comprehensive dashboard for administrators to manage tours and oversee user information.
- **CRUD Operations:** Demonstrating fundamental Create, Read, Update, Delete operations using a MySQL database

This project aims to showcase the practical application of database management concepts within a real-world context, providing a robust solution to common issues in the tourism industry.

REQUIREMENT FORMULATION AND ANALYSIS

Requirement Formulation and Analysis is a crucial phase in developing the Tourism Management System. It involves gathering, analyzing, and documenting information to identify user views, data requirements, and system functionalities necessary to support the organization's operations effectively.

Inputs to the Process

The inputs to this process include:

- **User Information Requirements:** Identification of data items and their relationships used by different user roles (e.g., administrators, customers).

User Views

Each major user view defines the requirements from the perspective of specific job roles or enterprise application areas. These views help in understanding what functionalities the system should support.

Information Gathered for Each User View

For each identified user view, the following information is gathered:

Administrator's View

- **Description of Data Used or Generated:**
  - User Profiles: User information, login credentials.
  - Tour Packages: Packages, locations, pricing, availability.
- **Details of Data Usage or Generation:**
  - Manage tour packages: create, update and delete packages.

Customer's View

- **Description of Data Used or Generated:**
  - Personal Information: Name and Information.
  - Booking Details: Tour package selected, booking status.
- **Details of Data Usage or Generation:**

- Browse available tour packages.
- Book tours, view booking history.

Fact-Finding Techniques

To gather these requirements, various fact-finding techniques were employed including:

- **Interviews:** Questions asked focused on job roles, data usage, and system expectations.
- **Observations:** Studying current processes and interactions with existing systems.
- **Questionnaires:** Collecting structured feedback from stakeholders to validate requirements.

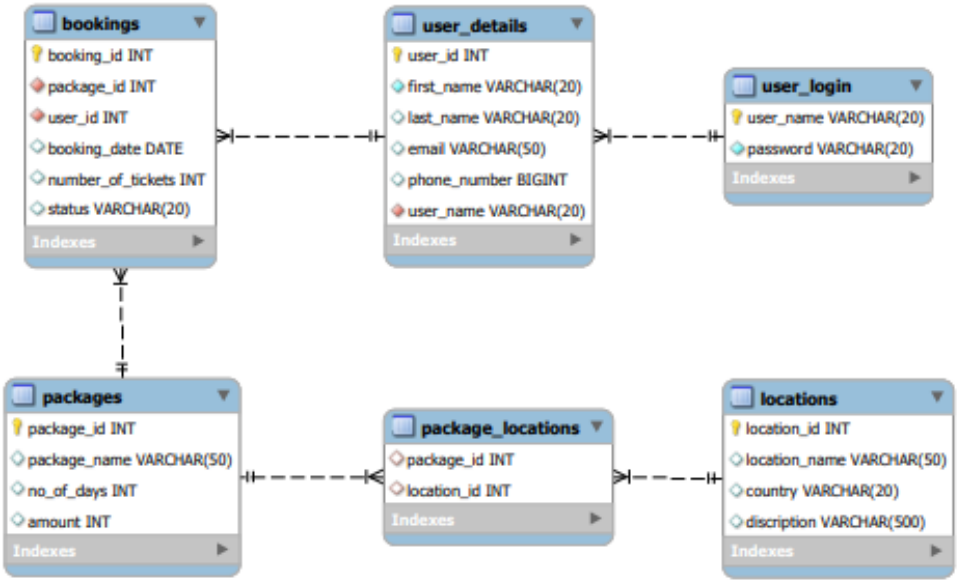
System Requirements

General system requirements were identified through interviews and observations, focusing on:

- **Security Requirements:** Desired levels of data protection and access control.
- **Backup and Recovery:** Strategies to ensure data integrity and availability in case of system failures.

Requirement Formulation and Analysis ensures that all user views, data elements, and operational needs are identified and documented comprehensively. This structured approach serves as the foundation for designing a robust Tourism Management System that meets stakeholder expectations and enhances operational efficiency.

CONCEPTUAL DESIGN



The Entity-Relationship (ER) diagram for the Tourism Management System project consists of six main entities: 'bookings', 'user_details', 'user_login', 'packages', 'package_locations', and 'locations'. Here's a detailed description of each entity and their attributes:

1. **Bookings**

- **Attributes:**
  - booking_id (INT): Primary Key
  - package_id (INT): Foreign Key referencing package_id in the packages table
  - user_id (INT): Foreign Key referencing user_id in the user_details table
  - booking_date (DATE): Date of the booking
  - number_of_tickets (INT): Number of tickets booked
  - status (VARCHAR(20)): Status of the booking (e.g., confirmed, canceled)

2. **User Details**

- **Attributes:**
  - user_id (INT): Primary Key
  - first_name (VARCHAR(20)): First name of the user
  - last_name (VARCHAR(20)): Last name of the user
  - email (VARCHAR(50)): Email address of the user

- phone_number (BIGINT): Phone number of the user
- user_name (VARCHAR(20)): User name of the user (Foreign Key referencing user_name in the user_login table)

3. **User Login**

- **Attributes:**
  - user_name (VARCHAR(20)): Primary Key
  - password (VARCHAR(20)): Password for the user account

4. **Packages**

- **Attributes:**
  - package_id (INT): Primary Key
  - package_name (VARCHAR(50)): Name of the package
  - no_of_days (INT): Number of days of the package
  - amount (INT): Cost of the package

5. **Package Locations**

- **Attributes:**
  - package_id (INT): Foreign Key referencing package_id in the packages table
  - location_id (INT): Foreign Key referencing location_id in the locations table

6. **Locations**

- **Attributes:**
  - location_id (INT): Primary Key
  - location_name (VARCHAR(50)): Name of the location
  - country (VARCHAR(20))
  - discription (VARCHAR(500)): Description for the location

Relationships

- The relationship between user_login and user_details is One-to-One.
- The relationship between user_details and bookings is One-to-Many.
- The relationship between packages and bookings is One-to-Many.
- The relationship between packages and package_locations is Many-to-Many.
- The relationship between locations and package_locations is Many-to-Many.

NORMALISATION

1. PROJECT SCHEMA(INITIAL)

• Users (user_id, first_name, last_name, email, phone_number, user_name, password)

• Locations (location_id, location_name, country,description)

• Package_details (package_id, location_id, package_name, no_of_days, amount)

• Bookings (booking_id, package_id, user_id, booking_date,no_of_tickets, booking_status)

2. FIRST NORMALISATION FORM(1NF)

• All tables are containing single values at the intersection of each row and column. So, all the tables are in First Normal Form.

3. SECOND NORMALISATION FORM(2NF)

• The Users table has the following full and partial functional dependencies on PK:

➢ user_id, user_name → first_name, last_name,

email, phone_number (Full Dependency)

➢ user_id → first_name, last_name, email,

phone_number, user_name (Partial Dependency)

➢ user_name → password (Partial Dependency)

• The new relation have the following form:

➢ user_details (user_id, first_name, last_name, email,

phone_number, user_name)

➢ user_login (user_name, password)

user_id	first_name	last_name	email	phone_number	user_name
1	John	Doe	john.doe@example.com	1234567890	John1234
2	Alice	Smith	alice.smith@example.com	1987654321	Alice1235
3	Michael	Johnson	michael.j@example.com	1122334455	Michael1236
4	Emily	Brown	emily.b@example.com	9988776655	Emily1237
5	David	Wilson	david.w@example.com	4455667788	David1238

user_name	password
John1234	Doe1234
Alice1235	Smith1235
Michael1236	Johnson1236
Emily1237	Brown1237
David1238	Wilson1238

• All other tables are already in 2NF Form.

4. THIRD NORMALISATION FORM(3NF)

• The Package_details table has the following transitive dependency:

➢ package_id → package_name, no_of_days, amount

(Transitive dependency on PK)

• The new relations have the following form:

➢ packages (package_id, package_name, no_of_days,

amount)

➢ Package_locations (package_id, location_id)

package_id	package_name	no_of_days	amount
1	Explore North Pakatan	8	$100
2	Explore South Pakistan	2	$50
3	Explore Sri Lanka	10	$850
4	Explore Turkey	6	$500

package_id	location_id
1	101
1	102
1	103
3	104
3	105
3	106
2	107
2	108
2	109
4	110
4	111
4	112

• All other tables are already in 3NF Form.

IMPLEMENTATION DESIGN

1.user_details

```
CREATE TABLE user_details (
  user_id int NOT NULL AUTO_INCREMENT,
  first_name varchar(20) NOT NULL,
  last_name varchar(20) DEFAULT NULL,
  email varchar(50) DEFAULT NULL,
  phone_number bigint DEFAULT NULL,
  user_name varchar(20) NOT NULL,
  PRIMARY KEY (user_id),
  UNIQUE KEY user_id_UNIQUE (user_id),
  KEY user_name (user_name),
  CONSTRAINT user_details_ibfk_1 FOREIGN KEY (user_name) REFERENCES user_login (user_name)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

2.user_login

```
CREATE TABLE user_login (
  user_name varchar(20) NOT NULL,
  password varchar(20) NOT NULL,
  PRIMARY KEY (user_name)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

3. locations

```
CREATE TABLE locations (
  location_id int NOT NULL AUTO_INCREMENT,
  location_name varchar(50) DEFAULT NULL,
  country varchar(20) DEFAULT NULL,
  discription varchar(500) DEFAULT NULL,
  PRIMARY KEY (location_id),
  UNIQUE KEY location_id_UNIQUE (location_id)
) ENGINE=InnoDB AUTO_INCREMENT=117 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

4. bookings

```
CREATE TABLE bookings (
  booking_id int NOT NULL AUTO_INCREMENT,
  package_id int NOT NULL,
  user_id int NOT NULL,
  booking_date date DEFAULT NULL,
  number_of_tickets int DEFAULT NULL,
  status varchar(20) DEFAULT NULL,
  PRIMARY KEY (booking_id),
  UNIQUE KEY booking_id_UNIQUE (booking_id),
  KEY package_id (package_id),
  KEY user_id (user_id),
  CONSTRAINT bookings_ibfk_2 FOREIGN KEY (package_id) REFERENCES packages (package_id),
  CONSTRAINT bookings_ibfk_3 FOREIGN KEY (user_id) REFERENCES user_details (user_id)
) ENGINE=InnoDB AUTO_INCREMENT=23 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

5. package_locations

```
CREATE TABLE package_locations (
  package_id int DEFAULT NULL,
  location_id int DEFAULT NULL,
```

```
  KEY package_id (package_id),

  KEY location_id (location_id),

  CONSTRAINT package_locations_ibfk_1 FOREIGN KEY (package_id) REFERENCES
packages (package_id),

  CONSTRAINT package_locations_ibfk_2 FOREIGN KEY (location_id) REFERENCES
locations (location_id)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

6. packages

```
CREATE TABLE packages (

  package_id int NOT NULL AUTO_INCREMENT,

  package_name varchar(50) DEFAULT NULL,

  no_of_days int DEFAULT NULL,

  amount int DEFAULT NULL,

  PRIMARY KEY (package_id),

  UNIQUE KEY package_id_UNIQUE (package_id)

) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci
```

Frontend Implementation

Implementing the front-end application for the BLUESTAR involves using modern web development tools such as HTML5, CSS3 and JAVASCRIPT for building dynamic user interfaces. Here's a detailed description of the implementation and features:

Tools Used
HTML5, CSS3, and JAVASCRIPT are utilized and integrated with the Flask, and MySql backend. It ensures a responsive and intuitive user experience, crucial for accessing and managing information and data.

Forms and Reports

1. Users

Forms:

- **Registration Form:** Allows new users to sign up by providing personal information such as name, contact details, and preferences.
- **Login Form:** Enables users to log in using their credentials.
- **Profile Update Form:** Lets users update their personal information, preferences, and travel history.
- **Booking Form:** Allows users to book tours or packages by selecting options, dates, and providing payment information.

Reports:

- **Booking History Report:** Displays the user's past bookings, including details such as dates, destinations, and costs.

2. Admins

Forms:

- **User Management :** Enables admins to view user accounts.
- **Package Management Form:** Allows admins to create, update, or delete tour packages.

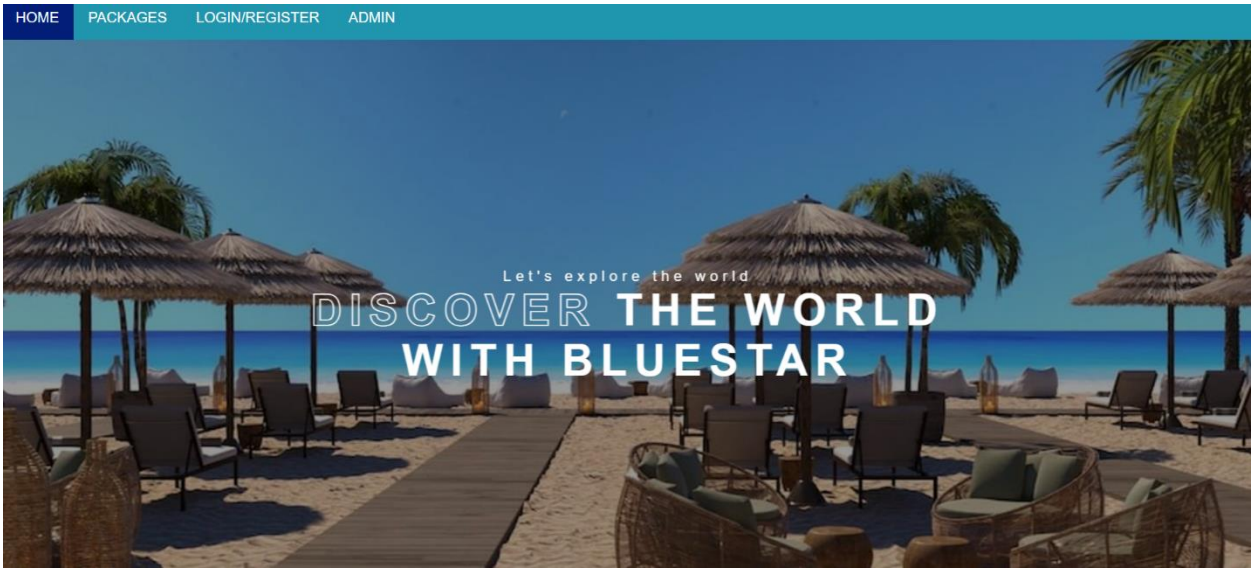
3. Packages

Forms:

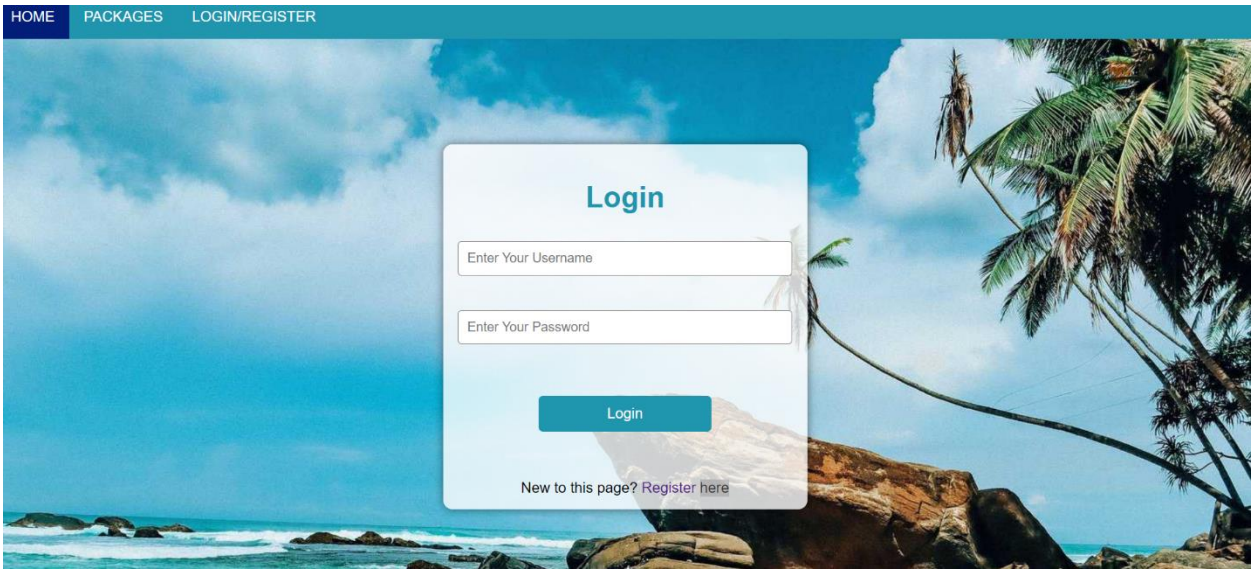
- **Package Creation Form:** Allows admins to create new tour packages by entering details such as destination, itinerary, price, and availability.
- **Package Update Form:** Enables admins to update existing packages with new information or changes.
- **Package Deletion Form:** Allows admins to remove outdated or unavailable packages from the system.
- **Package Booking Form:** Used by users to select and book available packages.

Project Snapshots

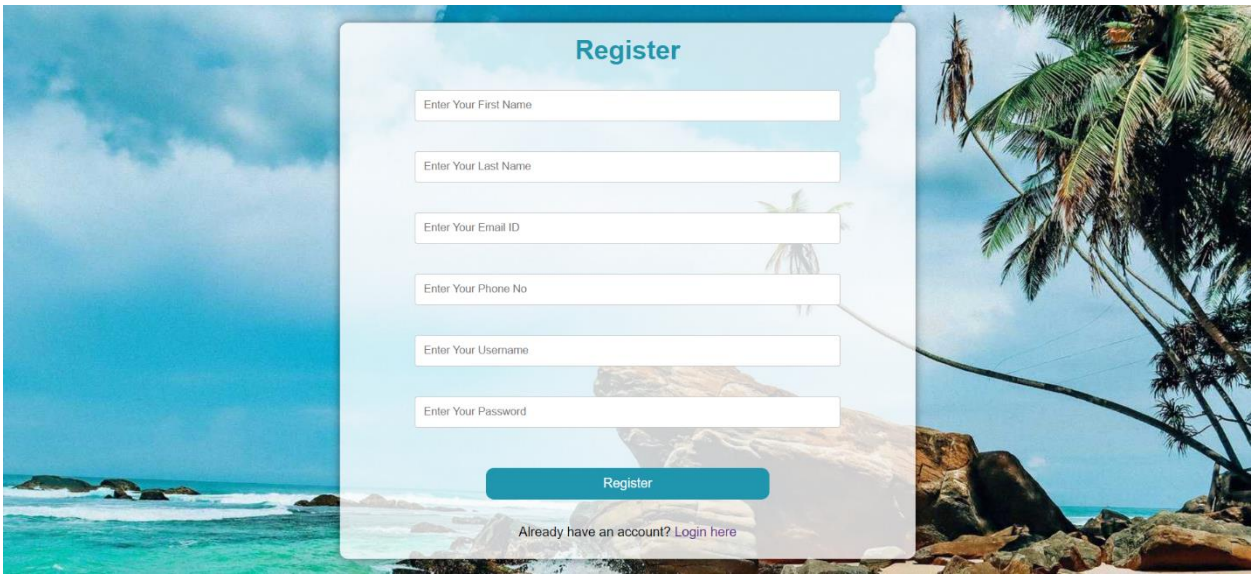
Startup Page



Login Page



Register Page

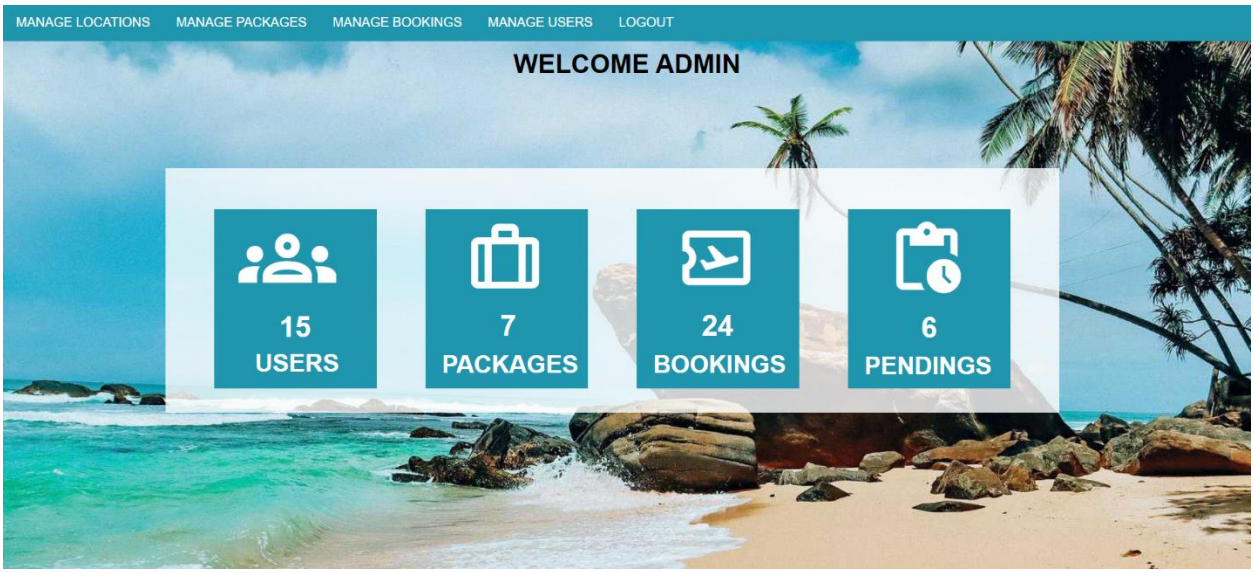


LoggedIn Page

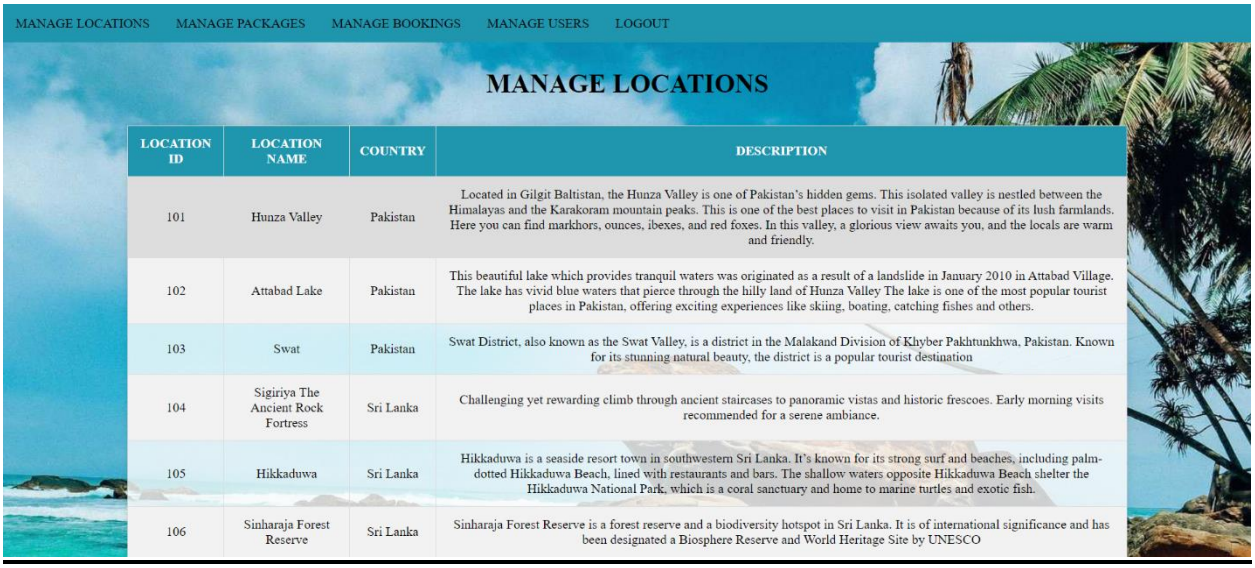
Booking Histroy Page

Packages Page

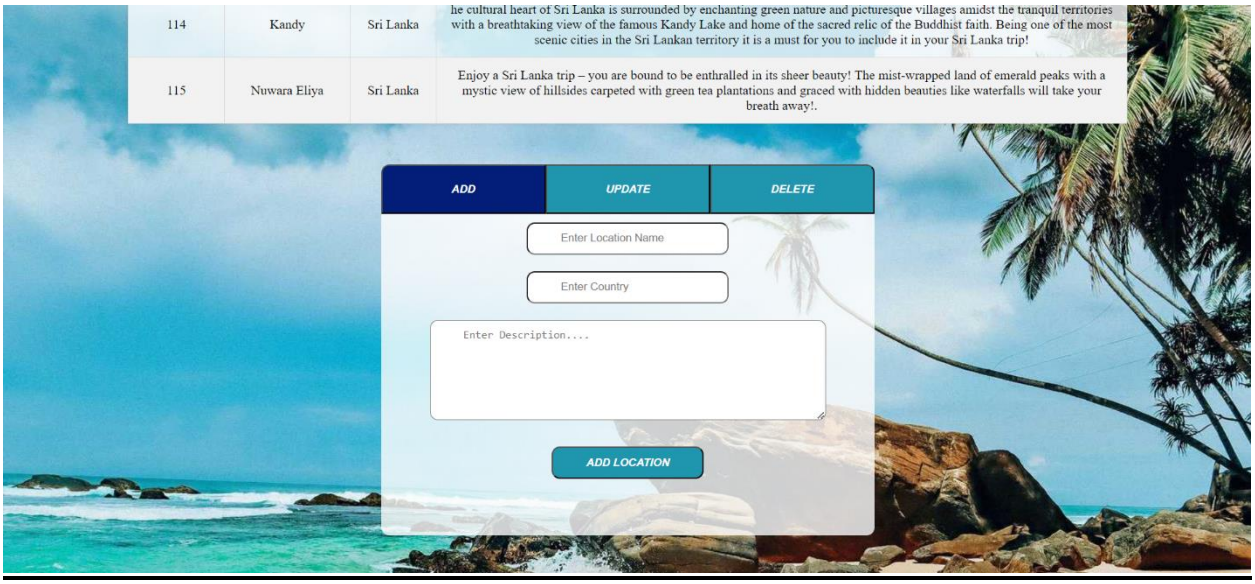
Admin Page

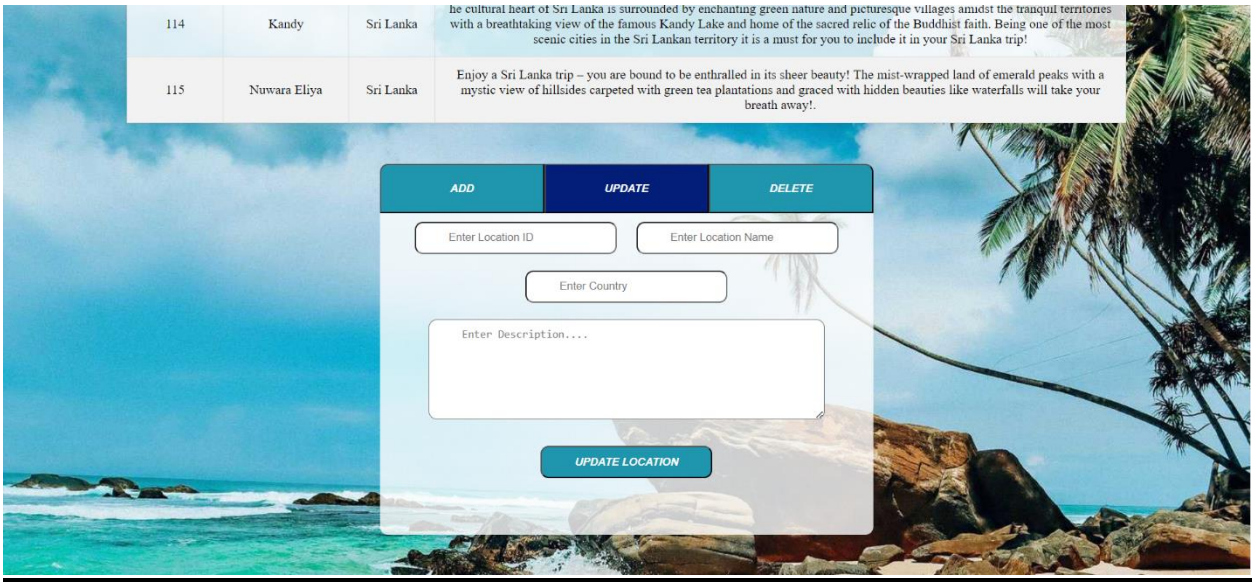


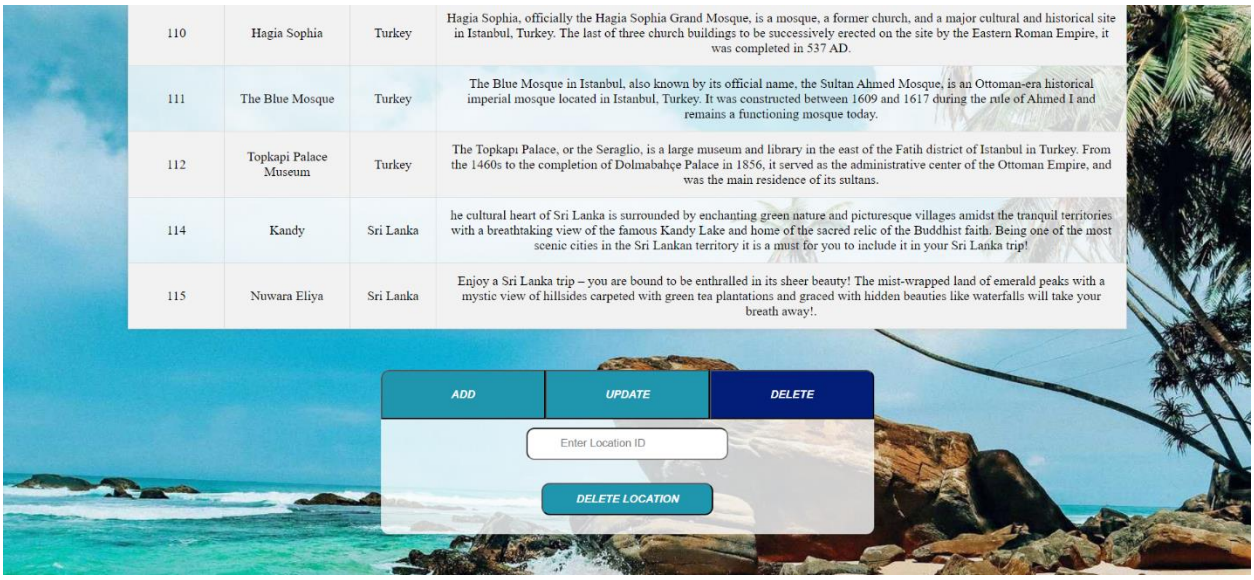
Manage locations Page



LOCATION ID	LOCATION NAME	COUNTRY	DESCRIPTION
101	Hunza Valley	Pakistan	Located in Gilgit Baltistan, the Hunza Valley is one of Pakistan's hidden gems. This isolated valley is nestled between the Himalayas and the Karakoram mountain peaks. This is one of the best places to visit in Pakistan because of its lush farmlands. Here you can find markhors, ounces, ibexes, and red foxes. In this valley, a glorious view awaits you, and the locals are warm and friendly.
102	Attabad Lake	Pakistan	This beautiful lake which provides tranquil waters was originated as a result of a landslide in January 2010 in Attabad Village. The lake has vivid blue waters that pierce through the hilly land of Hunza Valley The lake is one of the most popular tourist places in Pakistan, offering exciting experiences like skiing, boating, catching fishes and others.
103	Swat	Pakistan	Swat District, also known as the Swat Valley, is a district in the Malakand Division of Khyber Pakhtunkhwa, Pakistan. Known for its stunning natural beauty, the district is a popular tourist destination
104	Sigiriya The Ancient Rock Fortress	Sri Lanka	Challenging yet rewarding climb through ancient staircases to panoramic vistas and historic frescoes. Early morning visits recommended for a serene ambiance.
105	Hikkaduwa	Sri Lanka	Hikkaduwa is a seaside resort town in southwestern Sri Lanka. It's known for its strong surf and beaches, including palm-dotted Hikkaduwa Beach, lined with restaurants and bars. The shallow waters opposite Hikkaduwa Beach shelter the Hikkaduwa National Park, which is a coral sanctuary and home to marine turtles and exotic fish.
106	Sinharaja Forest Reserve	Sri Lanka	Sinharaja Forest Reserve is a forest reserve and a biodiversity hotspot in Sri Lanka. It is of international significance and has been designated a Biosphere Reserve and World Heritage Site by UNESCO

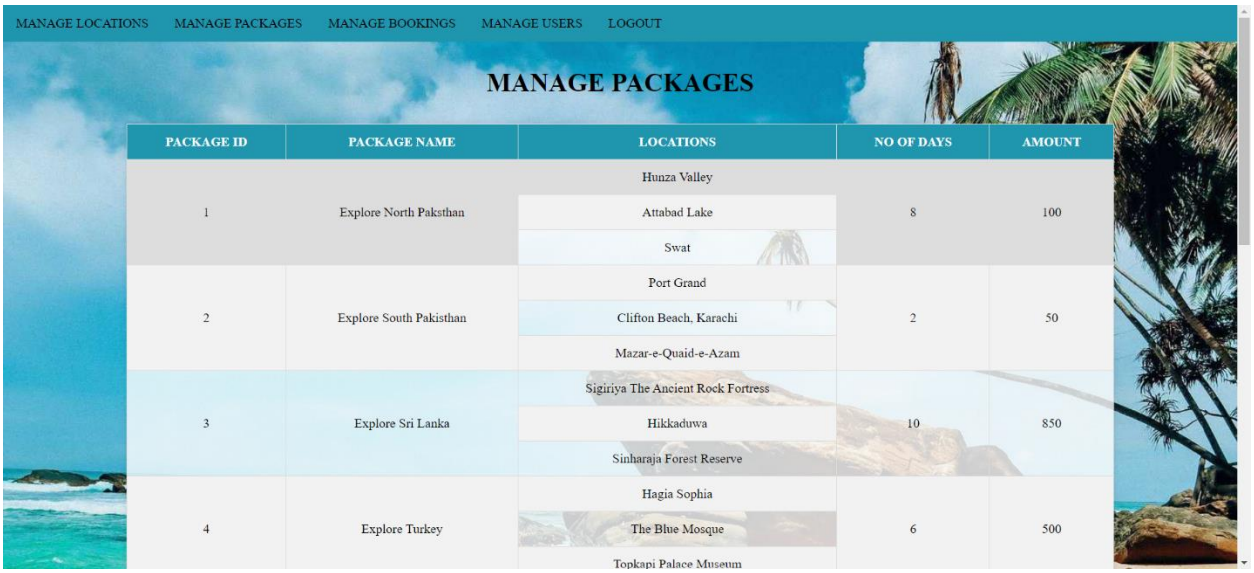

114
Kandy
Sri Lanka
he cultural heart of Sri Lanka is surrounded by enchanting green nature and picturesque villages amidst the tranquil territories with a breathtaking view of the famous Kandy Lake and home of the sacred relic of the Buddhist faith. Being one of the most scenic cities in the Sri Lankan territory it is a must for you to include it in your Sri Lanka trip!
115
Nuwara Eliya
Sri Lanka
Enjoy a Sri Lanka trip – you are bound to be enthralled in its sheer beauty! The mist-wrapped land of emerald peaks with a mystic view of hillsides carpeted with green tea plantations and graced with hidden beauties like waterfalls will take your breath away!.
ADD
UPDATE
DELETE
Enter Location Name
Enter Country
Enter Description....
ADD LOCATION


114
Kandy
Sri Lanka
he cultural heart of Sri Lanka is surrounded by enchanting green nature and picturesque villages amidst the tranquil territories with a breathtaking view of the famous Kandy Lake and home of the sacred relic of the Buddhist faith. Being one of the most scenic cities in the Sri Lankan territory it is a must for you to include it in your Sri Lanka trip!
115
Nuwara Eliya
Sri Lanka
Enjoy a Sri Lanka trip – you are bound to be enthralled in its sheer beauty! The mist-wrapped land of emerald peaks with a mystic view of hillsides carpeted with green tea plantations and graced with hidden beauties like waterfalls will take your breath away!.
ADD
UPDATE
DELETE
Enter Location ID
Enter Location Name
Enter Country
Enter Description....
UPDATE LOCATION

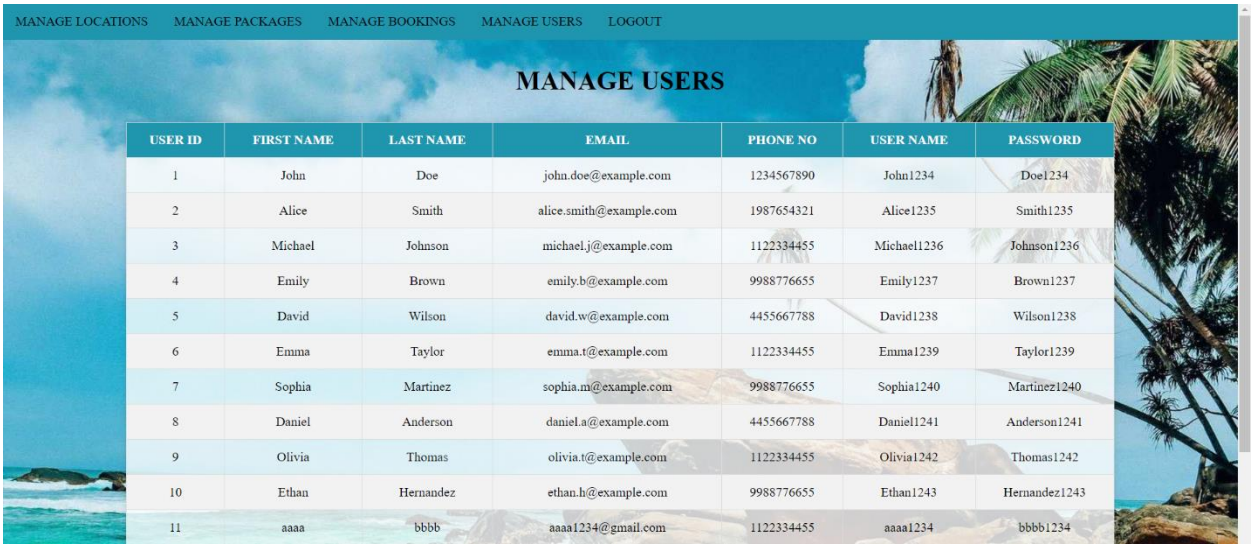


Manage Bookings Page

Manage Packages Page



Manage Users Page



Database Security and Future Improvements

1. Authentication and Authorization

User Authentication:

- Implement strong user authentication using libraries like Flask-Login to manage user sessions.
- Enforce strong password policies (e.g., minimum length, complexity).
- Consider using multi-factor authentication (MFA) for an added layer of security.

Role-Based Access Control:

- Define roles and permissions to ensure users only have access to the necessary parts of the application.
- Use role-based access control (RBAC) to manage user permissions effectively.

Input Validation:

- Validate and sanitize all user inputs to prevent injection attacks and ensure data integrity.

2. Regular Backups

- Implement regular, automated backups of your database to prevent data loss.
- Store backups securely and ensure they are encrypted.

Contribution of each student

Admin Module

- **Developed by:** Zaid
- **Technologies Used:** Flask, MySQL
- **Description:** The admin module allows administrators to manage users, tour packages, and bookings. It includes functionalities for creating, updating, and deleting packages, generating reports, and monitoring system activities. The module ensures data integrity and provides secure access to sensitive information.

User Module

- **Developed by:** Muhammed
- **Technologies Used:** Flask, MySQL
- **Description:** The user module provides a seamless experience for tourists to register, log in, browse available tour packages, and make bookings. It ensures secure handling of user data and transactions.

Frontend

- **Developed by:** Amna
- **Technologies Used:** HTML5, CSS3, JavaScript
- **Description:** The frontend provides an intuitive and responsive user interface for the tourism management system. It offers a visually appealing design, easy navigation, and a user-friendly experience across different devices. The frontend integrates seamlessly with the backend to display real-time data and ensure smooth user interactions.