

**A**  
**PROJECT REPORT ON**  
**ONLINE TOURISM MANAGEMENT SYSTEM**

**SUBMITTED BY**  
**Ms. Jagruti Rustum Khedkar**

**SUBMITTED TO**  
**SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**IN FULFILLMENT OF DEGREE**  
**MASTER OF COMPUTER APPLICATION (SEM-I)**

**UNDER THE GUIDANCE OF**  
**Dr. Shveti Chandan**  
**Through,**



**Sadhu Vaswani Institute of Management Studies for Girls, Koregaon  
Park, Pune-411001**

**2024-25**

# Certificate



This is to certify that the Mini project report entitled, “**ONLINE Tourism Management SYSTEM**” being submitted herewith in partial fulfilment of the requirement of the award of the degree of **MASTER**

**OF COMPUTER APPLICATION (SEM-I)** to Savitribai Phule Pune University, Pune is the result of the original project work completed by JAGRUTI RUSTUM KHEDKAR under my supervision and guidance and to the best of my knowledge and belief, the work embodied in this Project has not formed earlier the basis for the award of any Degree of similar title or any other University or examining body.

**Date:**

**Place: Pune**

**Dr. Shveti Chandan**  
**(Project Guide)**

**Dr. Neeta Raskar**  
**(HOD)**

**Dr. B. H. Nanwani**  
**(Director)**

## **DECLARATION BY STUDENT**

To,

The Director,

SVIMS, Koregaon Park, Pune

I, undersigned hereby declare that this project titled, “**ONLINE TOURISM MANAGEMENT SYSTEM**” written and submitted by me to SPPU, Pune, in partial fulfilment of the requirement of the award of the degree of **MASTER OF COMPUTER APPLICATION (MCA-I)** under the guidance of Dr. Shveti Chandan, is my original work.

I further declare that to the best of my knowledge and belief, this project has not been submitted to this or any other University or Institution for the award of any Degree.

**Place: Pune**

**Date:**

**(JAGRUTI RUSTUM KHEDKAR)**

## **ACKNOWLEDGEMENT**

I extend my sincere gratitude to Dr. B. H. Nanwani, Dr. Neeta Raskar and Dr. Shveti Chandan for allowing me to carry out the study and for their constant encouragement, valuable suggestions, and guidance during the research work.

I extend my special thanks to Dr. Shevti Chandan for their kind co-operation and inspiration.

I extend my special gratitude to my dearest family members and friends who encouraged and motivated me to complete the project report.

**Place: Pune**

**Date:**

**(Jagruti Rustum Khedkar)**

# INDEX

CHAPTER	DETAILS	PAGE NO
1	<b>CHAPTER 1: INTRODUCTION</b> 1.1 Client/Organization Profile 1.2 Need for System 1.3 Scope & Feasibility of Work 1.4 Operating Environment – H/w & S/w 1.5 Architecture of system 1.6 Detail Description of Technology Used	6 to 17
2	<b>CHAPTER 2: PROPOSED SYSTEM</b> 2.1 Proposed System 2.2 Objectives of System 2.3 User Requirements	18 to 20
3	<b>CHAPTER 3: ANALYSIS &amp; DESIGN</b> 3.1 DFD 3.2 Table specifications (Database) 3.3 ERD 3.4 Object Diagram 3.5 Class Diagram 3.6 Use Case Diagrams 3.7 Web Site Map Diagram (if Website)	21 to 27
4	<b>CHAPTER 4: USER MANUAL</b> 4.1 User Interface Design (Screens etc.) 4.2 Limitations 4.3 Future enhancement <b>BIBLIOGRAPHY</b> <b>ANNEXURE:</b> Sample program code	28 to 51

# **CHAPTER 1: INTRODUCTION**

## **1. Client Organization Profile**

The Tourism Management System (TMS) is designed to address the specific needs of tourism agencies and companies operating in the travel and hospitality industry. Below are the key aspects of the client organization profile:

### **Client Objectives**

The primary objectives of the tourism agency are:

- To simplify the management of tour packages, bookings, and customer inquiries.
- To enhance the customer experience by providing a user-friendly online platform.
- To improve operational efficiency by reducing manual work and errors.
- To boost revenue through improved service delivery and increased customer reach.

### **Target Users**

- Primary Users: Tourists and travelers who seek a seamless platform to book tour packages, check availability, and manage reservations.
- Secondary Users: Administrators of tourism agencies responsible for managing bookings, packages, and customer support.

### **Key Drivers for Digitization**

- Customer Expectations: Modern travelers demand an online presence and self-service platforms for tour bookings.
- Market Competition: The need to stay competitive in the tourism industry by adopting digital tools.
- Operational Efficiency: Automation reduces the time and cost of managing tour-related tasks.

## **Challenges in the Current System**

- **Manual Processes:** Inefficient handling of bookings, inquiries, and payment records.
- **Limited Customer Reach:** Lack of an online platform limits the visibility of tour packages.
- **Error-Prone Operations:** High chances of errors in manual record-keeping and scheduling.
- **Delayed Customer Service:** Slow response times due to reliance on traditional methods.

## **Role of the System in Transformation**

The Tourism Management System plays a transformative role by:

- **Streamlining Operations:** Automating bookings, ticket issuance, and package management.
- **Improving Customer Experience:** Offering an intuitive platform for users to browse, book, and manage tours.
- **Enhancing Data Accuracy:** Centralized storage and real-time access to information reduce errors.
- **Expanding Business Reach:** An online presence ensures accessibility to a wider audience.
- **Increasing Revenue:** Efficient processes and improved customer satisfaction result in higher profitability.

## **2. Need for the System**

The tourism industry is highly competitive and customer-driven, requiring businesses to adapt to the growing demand for online services and efficient management. The Tourism Management System (TMS) is developed to overcome the limitations of manual processes and meet modern customer expectations.

### **Key Challenges Resolved by the System**

#### **1. Time-Consuming Manual Processes**

- Challenge: Managing bookings, inquiries, and package details manually is slow and error-prone.
- Solution: Automation of bookings, ticket issuance, and customer queries ensures faster processing and reduces administrative workload.

#### **2. Limited Customer Access**

- Challenge: Without an online platform, businesses fail to reach potential customers globally.
- Solution: The system offers a web-based platform accessible from any device, significantly enhancing customer reach.

#### **1) Data Management and Record-Keeping Issues**

- Challenge: Storing and managing customer data, booking details, and inquiries in physical or disconnected systems leads to disorganization.
- Solution: A centralized database ensures secure, real-time access to data with efficient storage and retrieval capabilities.

#### **2) Delayed Customer Support**

- Challenge: Slow responses to customer inquiries and complaints due to manual handling impact customer satisfaction.
- Solution: Automated features like inquiry tracking and real-time availability checks improve response times and service quality.

#### **3) Inconsistent Revenue Growth**

- Challenge: Inefficient processes lead to missed opportunities for bookings and revenue generation.
- Solution: The system enables better resource management, accurate reporting, and a seamless booking experience, driving higher revenue.
- and additional services as the business expands.



### 3. Scope and Feasibility of Work

#### Scope of Work

The Tourism Management System (TMS) covers a wide range of functionalities designed to meet the needs of both customers and administrators. Key components of the scope include:

1. **User Registration and Login:** Allowing customers to create accounts and securely log in to access their profiles and booking history.
2. **Tour Package Management:** Enabling administrators to create, update, and delete tour packages with details like price, itinerary, and availability.
3. **Booking System:** Facilitating customers to book tours online, check availability, and make payments securely.
4. **Customer Inquiry Management:** Providing a platform for customers to submit inquiries and receive timely responses.
5. **Payment Integration:** Supporting secure online payment options for booking confirmations.
6. **Admin Dashboard:** Equipping administrators with tools to manage bookings, inquiries, and customer feedback.
7. **Tour History Tracking:** Allowing customers to view and manage their past and upcoming bookings.
8. **Responsive Design:** Ensuring compatibility across devices like desktops, tablets, and smartphones.
9. **Reporting and Analytics:** Generating reports for administrators to analyze booking trends and revenue.
10. **Scalability:** Supporting the addition of more features, users, and tour packages as the business grows.

#### Feasibility Work

The feasibility of the Tourism Management System is assessed in terms of technical, economic, and operational aspects:

##### 1. Technical Feasibility:

- Utilizes widely adopted technologies like PHP, MySQL, and Bootstrap, which are robust and scalable.
- Compatible with modern browsers and devices, ensuring accessibility.

- Easily deployable on affordable web hosting platforms.

## **2. Economic Feasibility:**

- Low maintenance cost ensures long-term affordability.
- The system's efficiency leads to reduced operational expenses.

## **3. Operational Feasibility:**

- Simple user interface ensures ease of use for customers and administrators.
- Automating repetitive tasks increases staff productivity.
- Enhances customer satisfaction, leading to higher retention rates.

## **4. Time Feasibility:**

- Can be developed and deployed within a reasonable timeline using agile methodologies.

## **5. Market Feasibility:**

- Meets the growing demand for digital solutions in the tourism industry.
- Provides a competitive edge in attracting tech-savvy customers.

## **6. Scalability:**

- The architecture supports future enhancements without requiring a complete system overhaul.

## **7. Legal Feasibility:**

- Complies with data protection and payment gateway regulations, ensuring secure transactions.

## **8. Resource Feasibility:**

- Requires minimal hardware and infrastructure to operate efficiently.
- Digitizing operations reduces paper usage, contributing to sustainable practices.

## **9. Social Feasibility:**

- Improves customer experience, fostering trust and brand loyalty among us

## 4. Operating Environment

The Tourism Management System (TMS) requires a suitable hardware and software setup to function efficiently. Below are the specifications:

### Hardware Requirements

**1. Server Hardware:**

- Processor: Intel Core i5 or equivalent.
- RAM: Minimum 8 GB.
- Storage: 250 GB SSD for fast data retrieval.
- Network: High-speed internet connection for seamless data transfer.

**2. Client Hardware:**

- Device: Desktop, laptop, tablet, or smartphone.
- Processor: Dual-core or higher for smooth browsing.
- RAM: Minimum 4 GB.
- Display: 1366x768 resolution or higher for optimal viewing.
- Network: Reliable internet connection (minimum 2 Mbps).

**3. Backup Hardware:** External storage (e.g., 1 TB HDD) for periodic data backups.

**4. Power Supply:** Uninterrupted Power Supply (UPS) for consistent server operation.

**5. Peripherals:** Standard keyboard, mouse, and display for administrative tasks.

### Software Requirements

**1. Server Software:**

- Operating System: Linux (preferred) or Windows Server.
- Web Server: Apache 2.4 or higher.
- Database Management System: MySQL 5.7 or higher.

**2. Client Software:**

- Operating System: Windows, macOS, Android, or iOS.
- Web Browser: Chrome, Firefox, Safari, or Edge (latest versions).
- 3. Development Tools:**
  - Programming Language: PHP 7.4 or higher.
  - Front-End Frameworks: Bootstrap 4, HTML5, and CSS3.
  - Scripting: JavaScript (with optional libraries like jQuery).
- 4. Payment Gateway Integration:** Software to support secure online payment (e.g., Razorpay, PayPal API).
- 5. Security Tools:**
  - SSL Certificate for secure HTTPS connections.
  - Firewall and antivirus software for protection against cyber threats.

## 5. Architecture of the System

The Tourism Management System (TMS) is built on a Three-Tier Architecture to ensure scalability, maintainability, and efficiency. The architecture separates the system into three layers:

### Three-Tier Architecture

#### 1. Presentation Layer:

- The user interface built using HTML, CSS, Bootstrap, and JavaScript.
- Accessible via web browsers on desktops, tablets, and smartphones.
- Provides an intuitive and responsive design for both customers and administrators.

#### 2. Business Logic Layer:

- The application logic implemented using PHP.
- Processes user requests, manages workflows, and enforces business rules.
- Handles functionalities like booking validation, user authentication, and inquiry processing.

#### 3. Data Layer:

- The backend database managed by MySQL.
- Stores essential information such as user profiles, tour packages, bookings, and payment records.
- Ensures data consistency and security through structured queries and database constraints.

### Advantages of Three-Tier Architecture

#### 1. Scalability:

- Each layer can be scaled independently to handle increased load, ensuring system growth without complete redesign.

#### 2. Maintainability:

- Separation of concerns allows easier updates and debugging within individual

layers without affecting others.

3. **Reusability:**

- Components from each layer can be reused in other projects or applications.

4. **Flexibility:**

- Supports changes to user interfaces or business logic without altering the database structure.

5. **Enhanced Security:**

- The architecture allows data validation and access control at different layers, reducing the risk of unauthorized access.

6. **Improved Performance:**

- Efficient communication between layers ensures faster response times and better load distribution.

7. **Compatibility:**

- Works seamlessly with modern web technologies, supporting integration with APIs, third-party tools, and cloud services.

8. **Simplified Testing:**

- Each layer can be tested independently, simplifying quality assurance and reducing testing time.

9. **Data Integrity:**

- Centralized database ensures consistent and reliable data storage and retrieval.

10. **Future-Ready:**

- The architecture can accommodate emerging technologies and evolving business needs with minimal disruption.

## 6. Detailed Description of Technology Used

The Tourism Management System (TMS) utilizes a combination of modern technologies to ensure robust functionality, scalability, and a seamless user experience. Below is an overview of the technologies used:

### 1. HTML (HyperText Markup Language)

- **Key Features:**
- Standard markup language for creating web pages.
- Supports multimedia integration such as audio and video.
- Compatible with modern web browsers.
- **Advantages:**
- Easy to learn and implement.
- Provides cross-platform support.
- Forms the foundation for all web-based applications.

### 2. JavaScript

- **Key Features:**
- Supports real-time updates without reloading the page.
- Offers extensive libraries and frameworks for development.
- Compatible with all major web browsers.

#### **Advantages:**

- Reduces server load by executing scripts on the client side.
- Easily integrates with other technologies like AJAX and jQuery.

### 3. PHP (Hypertext Preprocessor)

- **Key Features:**
- A server-side scripting language used for backend development.
- Processes user requests, handles forms, and manages dynamic content.
- Integrates seamlessly with databases like MySQL.
- Provides a secure environment for application logic.
- **Advantages:**
- Open-source and cost-effective.
- Easy to learn and widely supported.
- Highly scalable and customizable for dynamic web applications.

### 4. XAMPP

#### Key Features:

- Simplifies the setup of a local development environment.
- Provides a graphical interface for managing servers and databases.
- **Advantages:**
- Easy to install and configure.
- Ideal for testing and debugging PHP applications.
- Enables offline development before deployment to a live server.



## 5. MySQL

### Key Features:

- A powerful, open-source relational database management system.
- Supports Structured Query Language (SQL) for database operations.
- Provides scalability, security, and reliable data storage.
- **Advantages:**
- Fast and efficient for handling large volumes of data.
- Offers robust data integrity and security features.
- Compatible with various programming languages and platforms.

## 6. AJAX (Asynchronous JavaScript and XML)

### Key Features:

- Enables asynchronous communication between client and server.
- Updates parts of a web page without reloading the entire page.
- Works seamlessly with JavaScript, XML, JSON, and other technologies.
- **Advantages:**
- Enhances user experience by reducing page load times.
- Improves application performance with real-time updates.
- Enables the development of dynamic and interactive web applications.

## 7. jQuery

### Key Features:

- Provides cross-browser compatibility for consistent functionality.
- Includes a wide range of plugins for extended functionality.
- **Advantages:**
- Reduces development time with simple and concise syntax.

## **CHAPTER 2: PROPOSED SYSTEM**

### **1. Proposed System**

The Tourism Management System (TMS) is a web-based platform designed to address the challenges faced by tourism agencies. It replaces manual processes with a digitized system that streamlines operations, enhances customer experience, and improves overall efficiency.

**Key features of the proposed system include:**

- A secure admin panel for managing tour packages, bookings, inquiries, and payments.
- Integration with payment gateways for seamless online transactions.
- Real-time updates for tour availability, booking status, and inquiries.

The system aims to bridge the gap between tourism businesses and customers by leveraging technology to deliver a comprehensive and scalable solution.

### **2. Objectives of the System**

The Tourism Management System is designed with the following objectives:

1. To automate the booking process, reducing manual effort and errors.
2. To enhance operational efficiency by centralizing data storage and management.
3. To enable secure and seamless online payments.
4. To generate analytical reports for better decision-making and business growth.
5. To improve customer satisfaction and retention through better service delivery.
6. To increase the reach of tourism businesses by offering an accessible online platform.

### **3. User Requirements**

The system is divided into two main modules based on the roles of users:

#### **A. User Modules and Requirements**

These modules cater to end-users (customers) accessing the platform.

- 1. User Registration and Login:**
  - Customers must register to create an account with basic details like name, email, and password.
  - Secure login functionality with password encryption.
- 2. Tour Package Browsing:**
  - Users can view available tour packages with details like itinerary, cost, and availability.
  - Search and filter options to find specific packages.
- 3. Booking Management:**
  - Users can book tours online and view booking history.
  - Real-time updates on booking status.
- 4. Inquiry Submission:**
  - Users can submit inquiries about packages, bookings, or other concerns.
  - Option to receive responses from administrators via email or the portal.
- 5. Payment Processing:**
  - Secure payment options for booking confirmation.
  - Integration with popular payment gateways (e.g., PayPal, Razorpay).
- 6. Profile Management:**
  - Users can update their personal details and view their booking history.

#### **B. Admin Modules and Requirements**

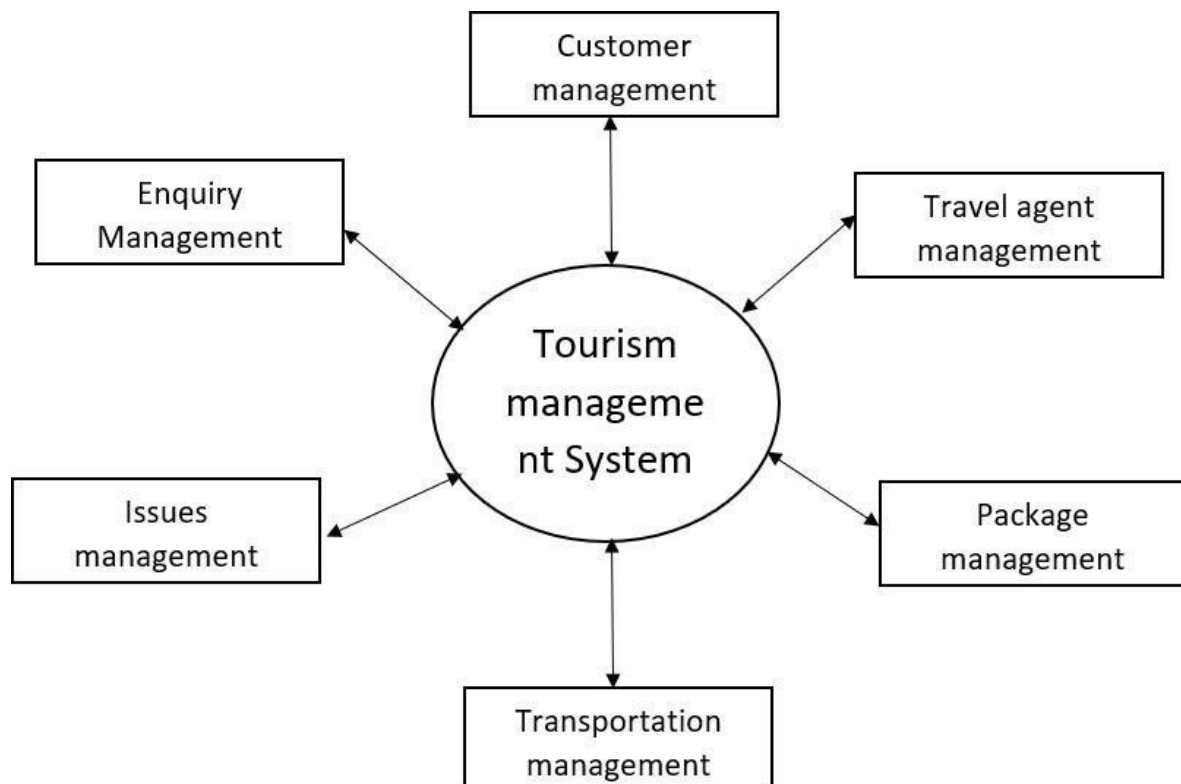
These modules are designed for administrators managing the platform.

1. **Admin Login:**
  - Secure access for administrators with multi-factor authentication (optional).
2. **Tour Package Management:**
  - Add, update, or delete tour packages.
  - Set pricing, availability, and package descriptions.
3. **Booking Management:**
  - View and manage customer bookings.
  - Update booking statuses (e.g., confirmed, canceled).
4. **Inquiry Management:**
  - Receive and respond to customer inquiries in real-time.
  - Track resolved and pending inquiries.
5. **Payment Records:**
  - Monitor payment transactions and generate payment receipts.
  - Ensure secure handling of sensitive customer payment data.
6. **Reports and Analytics:**
  - Generate reports on bookings, revenue, and customer feedback.
  - Analyze trends to make informed business decisions.
7. **User Management:**
  - View and manage registered users.
  - Reset passwords or deactivate accounts if required.
8. **System Settings:**
  - Configure system parameters, including email notifications and payment
  - Monitor server and database health for uninterrupted service.

## CHAPTER 3: ANALYSIS & DESIGN

### 3.1 Data Flow Diagrams (DFD)

DFD is the abbreviation for **Data Flow Diagram**. The flow of data in a system or process is represented by a Data Flow Diagram (DFD). It also gives insight into the inputs and outputs of each entity and the process itself. Data Flow Diagram (DFD) does not have a control flow and no loops or decision rules are present. Specific operations, depending on the type of data, can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. It is useful for analyzing existing as well as proposed systems.



## DATA FLOW DIAGRAM

### 3.2 Table Specifications (Database)

#### 1. User Table

Field Name	Data Type	Description	Constraints
UserID	INT	Unique identifier for each user	Primary key, Auto-Increment
Name	VARCHAR(50)	Full name of the user	Not Null
Email	VARCHAR(100)	User's email address	Unique
Password	VARCHAR(255)	User's encrypted password	Not Null
Role	ENUM('User','Admin')	Role assigned to the user	Default 'user'

#### 2. Package table

Field Name	Data Type	Description	Constraints
PackageID	INT	Unique identifier for each package	Primary key, Auto-Increment
Destination	VARCHAR(100)	Travel Designation	Not Null
Price	VARCHAR(102)	Cost of the package	Not Null
Duration	DECIMAL(10,2)	Duration of the package in days	Not Null
CreatedDate	DATETIME	Role assigned to the user	Default Current Timestrap

#### 3. Booking Table

Field Name	Data Type	Description	Constraints
BookingID	INT	Unique identifier for each booking	Primary key, Auto-Increment
UserID	INT	ID of the user making the booking	Foreign Key(User)
PackageID	INT	ID of user making the booking	Foreign Key (Package)
BookingDate	DATETIME	Date and Time of booking	Default current Timestamp
Status	ENUM('Confirmed','pending','Canceled')	Status of booking	Default 'Pending'

#### 4. Inquiry Table

Field Name	Data Type	Description	Constraints
InquiryID	INT	Unique identifier for each inquiry	Primary key, Auto-Increment
UserID	INT	ID of the user submitting the inquiry	Foreign Key(User)
Message	TEXT	Inquiry message submitted by the user	Not null
Response	TEXT	Admin response to the inquiry	
Date	DATETIME	Date of inquiry submission	Default current Timestamp

#### 5. Payment Table

Field Name	Data Type	Description	Constraints
PaymentID	INT	Unique identifier for each payment	Primary key, Auto-Increment
BookingID	INT	ID of the booking related to payment	Foreign Key(Booking)
UserID	INT	ID of the user making the payment	Foreign Key(User)
Amount	DECIMAL(10,2)	Amount paid	Not Null
Status	ENUM('Paid','Pending','Failed')	Payment status	Default 'Pending'
PaymentDate	DATETIME	Date and time of payment	Default current Timestamp

### 3.3 ER Diagram

The Entity-Relationship (ER) Diagram for the Online Tourism Management System (OTMS) represents the entities and their relationships within the system. Here's a detailed description of the diagram:

**1. User and UserType:**

- The User entity represents users of the system, including their userID, username, password, email, and userType (which is a foreign key referencing the UserType entity).
- The UserType entity defines the type of user, such as customer or admin.

**2. Booking and Tour:**

- The Booking entity represents bookings made by users, including bookingID, tourID (foreign key referencing Tour entity), userID (foreign key referencing User entity), bookingDate, and status.
- The Tour entity represents tours available in the system, including tourID, tourName, description, location, and price.

**3. Inquiry:**

- The Inquiry entity represents inquiries made by users, including inquiryID, userID (foreign key referencing User entity), inquiryMessage, inquiryDate, responseMessage, and responseDate.

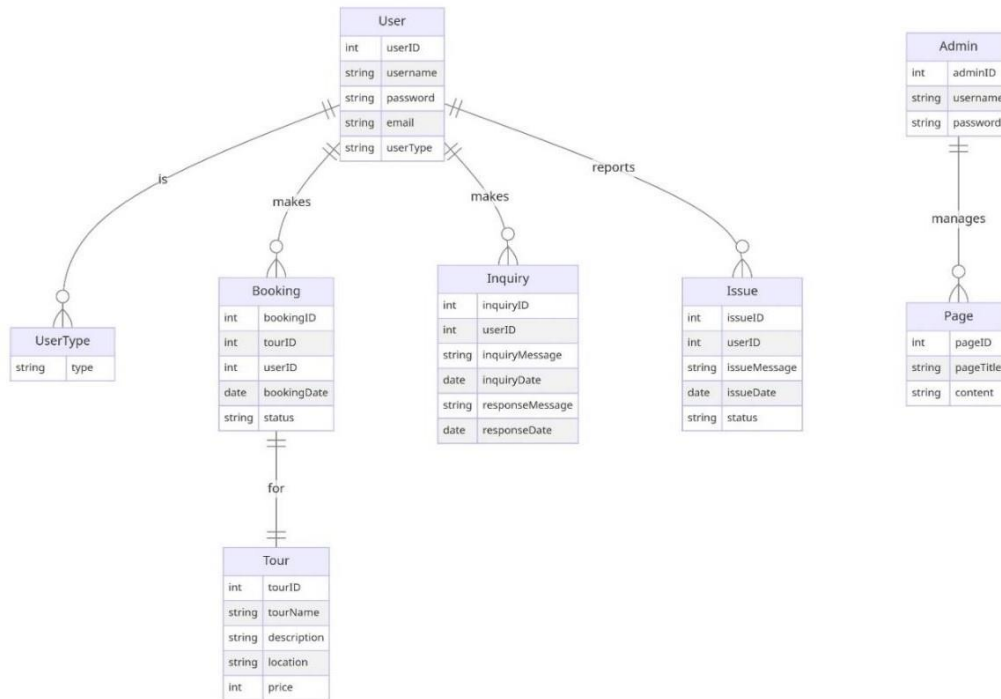
**4. Issue:**

- The Issue entity represents issues reported by users, including issueID, userID (foreign key referencing User entity), issueMessage, issueDate, and status.

**5. Page and Admin:**

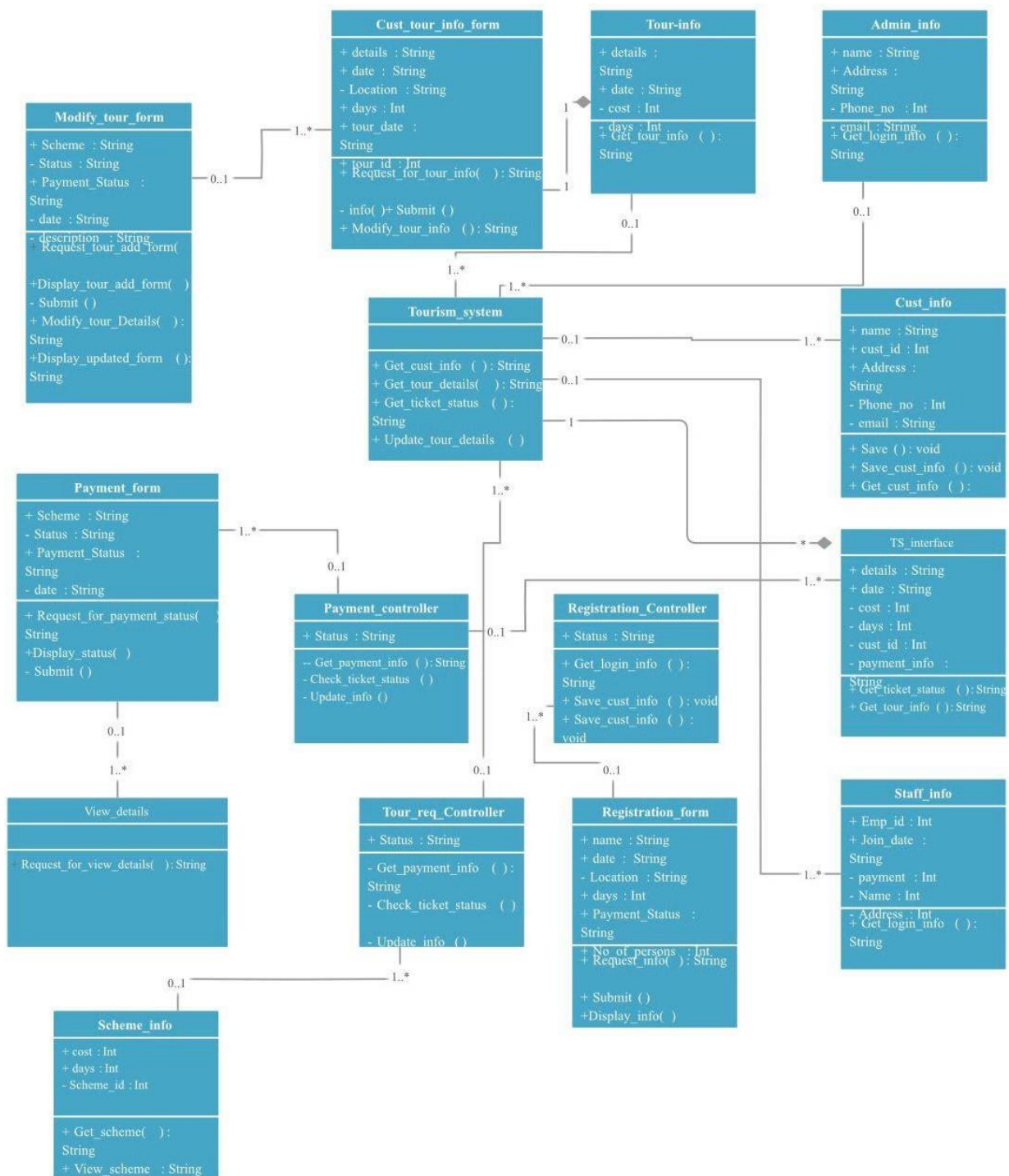
- The Page entity represents website pages, including pageID, pageTitle, and content.
- The Admin entity represents administrators of the system, including adminID, username, and password. Admins can manage pages.





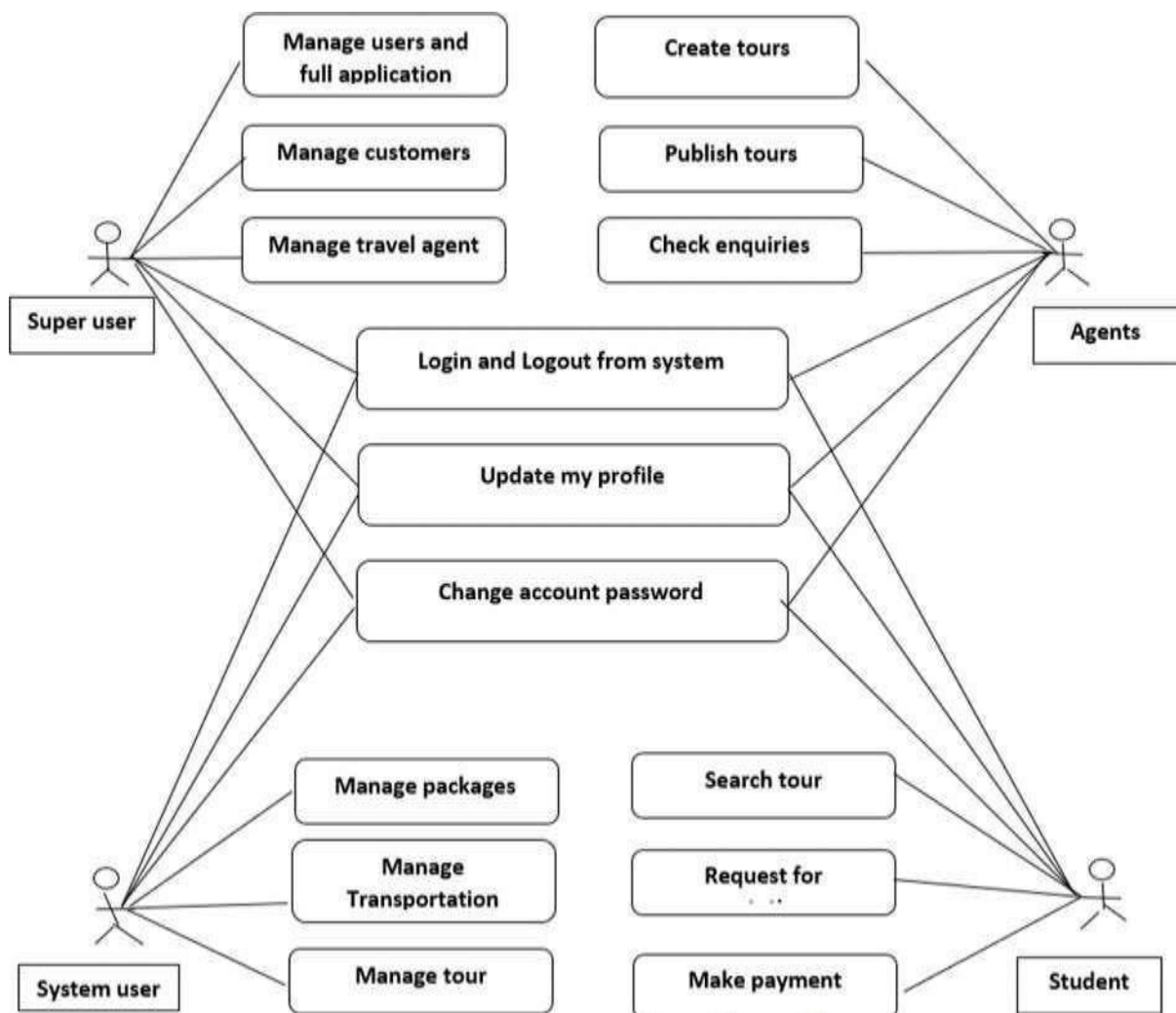
### 3.4 Class Diagram

In UML, class diagrams are one of six types of structural diagram. Class diagrams are fundamental to the object modeling process and model the static structure of a system. Depending on the complexity of a system, you can use a single class diagram to model an entire system, or you can use several class diagrams to model the components of a system.



### 3.5 Use Case Diagram

A Use Case Diagram in Unified Modeling Language (UML) is a visual representation that illustrates the interactions between users (actors) and a system. It captures the functional requirements of a system, showing how different users engage with various use cases, or specific functionalities, within the system. Use case diagrams provide a high-level overview of a system's behavior, making them useful for stakeholders, developers, and analysts to understand how a system is intended to operate from the user's perspective, and how different processes relate to one another. They are crucial for defining system scope and requirements.

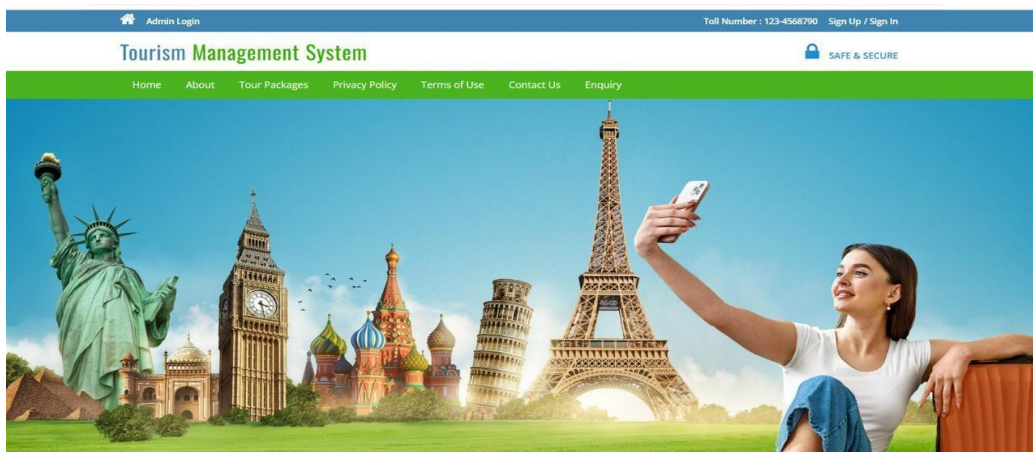


# CHAPTER 4: USER MANUAL

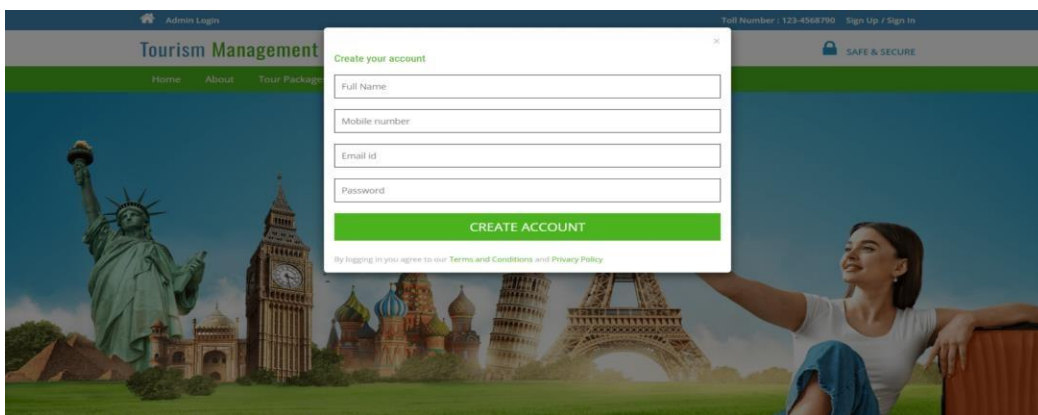
## 4.1 User Interface Design

### 1. Home Page

- **Header:** Navigation menu, contact info, sign-up/sign-in.
- **Banner:** Iconic landmarks and travel theme.
- **Tour Packages:** List of packages with details (type, location, features, price) and a "View More" option.
- **Stats:** 80,000 inquiries, 1,900 users, 7M+ bookings.
- **Footer:** Copyright and social links.



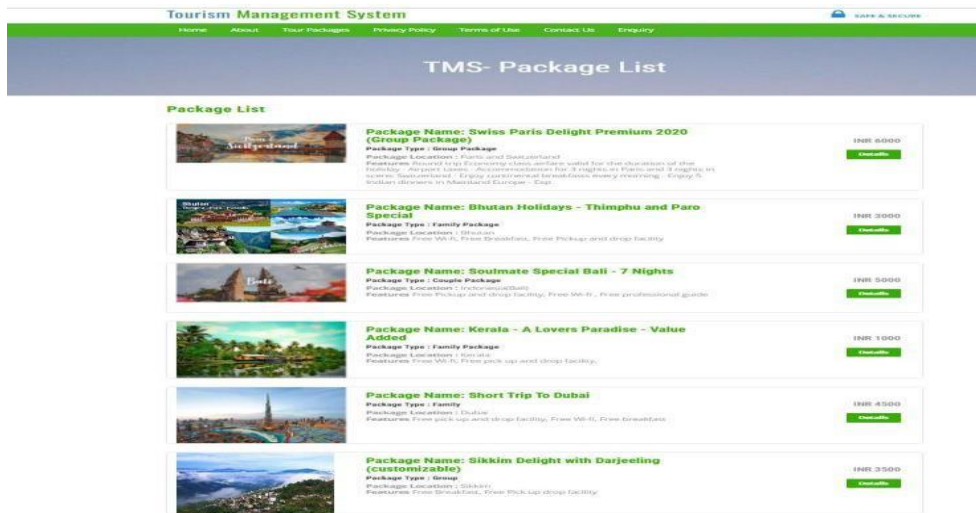
### 2. User Signup/ Registration



The image shows a **signup form** for the Tourism Management System. It includes fields for:

- Full Name
- Mobile Number
- Email ID
- Password

### 3. Package List



Purpose: Displays a list of tourism packages available for users to explore and book.

Features:

Detailed descriptions of packages, including location, type, features, and pricing.


### 4. Enquiry Form

Purpose: Allows users to submit inquiries regarding tourism packages or services.

Features:

Input fields for Full Name, Email, Mobile Number, Subject, and a Description box.

## Admin Page



The image shows a 'Sign In' page for an admin interface. It has a blue background. At the top, the text 'Sign In' is centered. Below it, there are two input fields: 'Username:' with the value 'admin' and 'Password:' with masked characters '.....'. A 'Forgot Password' link is positioned above the password field. A dark blue 'Sign In' button is located to the right of the password field. At the bottom, there is a 'Back to home' link.

Purpose: Provides a secure login interface for administrators to manage the tourism system.

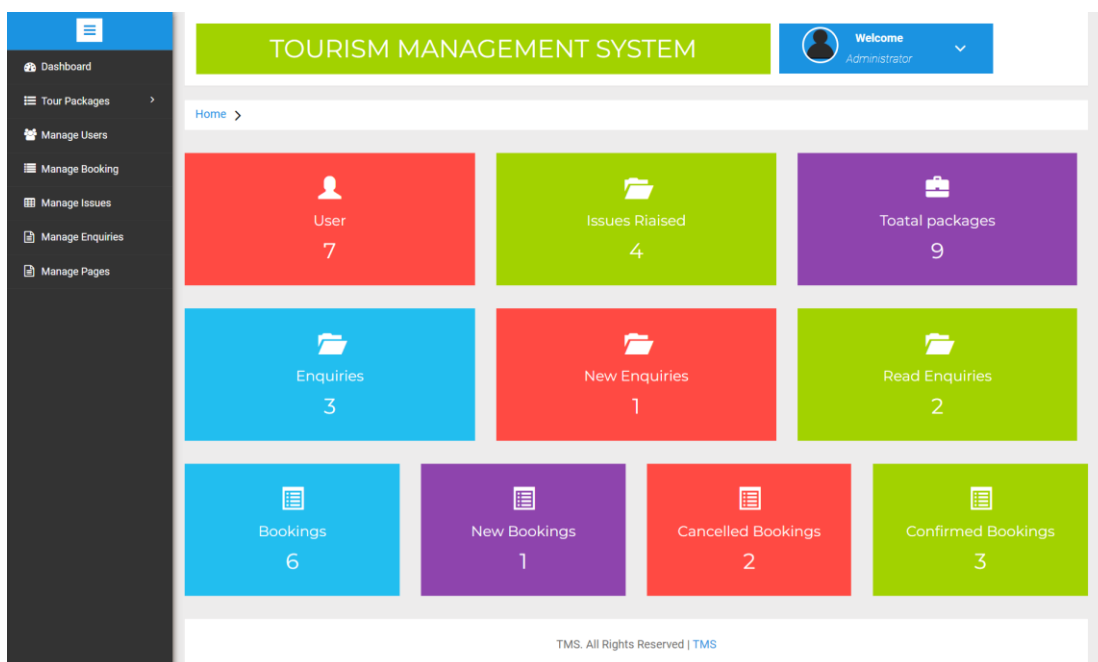
Features:

Fields for entering Username and Password.

Includes a “Forgot Password” link for account recovery.

Minimalistic design with a “Sign In” button for authentication.

## 6. Admin Dashboard



### Navigation Panel:

Options for managing tour packages, users, bookings, issues, enquiries, and pages

## 7. Create Package

The screenshot shows a web application interface for creating a package. The header is green with the text 'TOURISM MANAGEMENT SYSTEM'. Below the header, there is a blue bar with a user profile icon and the name 'Admin'. The main content area is titled 'Create Package' and contains several form fields: 'Package Name', 'Package Type', 'Package Location', 'Package Price in USD', 'Package Features', and 'Package Details'. Each field has a dropdown menu. At the bottom, there is a 'Package Image' section with a 'Choose File' button and a 'Save' button. A footer bar at the bottom of the page reads 'TMS All Rights Reserved | 2025'.

A **Create Package** page typically allows users to design and customize a new service or product package, whether for travel, software, subscriptions, or other services.

## 4.2 Limitation

The current version of the Tourism Management System (TMS) has a few limitations that can be addressed in future updates:

- **Limited Multilingual Support:** The system currently supports only one language, restricting its usability for non-English-speaking users.
- **Basic Analytics:** Advanced analytical features like predictive insights and customer behavior tracking are not yet implemented.
- **No Mobile App:** The system is web-based and lacks a dedicated mobile application for enhanced accessibility.
- **Offline Functionality:** Requires an active internet connection to operate, as offline mode is not supported.
- **Limited Customization:** Users and administrators have limited options to customize the platform's appearance and functionality.

### 4.3 Future Enhancements

To improve the system's functionality and user experience, the following enhancements are planned:

- **Multilingual Support:** Adding support for multiple languages to cater to a global audience.
- **Mobile Application:** Developing a native mobile app for Android and iOS platforms.
- **AI-Based Recommendation:** Integrating AI to suggest personalized tour packages based on user preferences.
- **Enhanced Reporting Tools:** Implementing advanced analytics dashboards with graphical reports and trend analysis.
- **Offline Mode:** Adding offline functionality for limited features, such as viewing already downloaded tour details.
- **Chatbot Integration:** Introducing AI-driven chatbots for instant query resolution and customer support.
- **Dynamic Pricing:** Implementing algorithms to adjust package prices based on demand, seasonality, and availability.
- **Customizable Themes:** Allowing administrators to personalize the platform's design and layout.
- **Advanced Security Features:** Incorporating features like two-factor authentication (2FA) and end-to-end encryption.
- **Third-Party Integrations:** Expanding integration options for social media platforms, marketing tools, and travel APIs.

This section highlights the current system's areas for improvement and outlines plans to ensure the Tourism Management System remains competitive and future-ready.



## BIBLIOGRAPHY

### 1. Books and Publications

- “Web Development with PHP and MySQL” by Robin Nixon, O’Reilly Media.
- “HTML and CSS: Design and Build Websites” by Jon Duckett, Wiley, 2011.
- “Learning JavaScript: JavaScript Essentials for Modern Application

### 2. Online Resources

- Official PHP Documentation: <https://www.php.net/docs.php>
- MySQL Documentation: <https://dev.mysql.com/doc/>
- W3Schools Tutorials:
- HTML: <https://www.w3schools.com/html/>
- CSS: <https://www.w3schools.com/css/>
- JavaScript: <https://www.w3schools.com/js/>

### 3. Development Tools and Frameworks

- Bootstrap Framework: <https://getbootstrap.com>
- jQuery Library: <https://jquery.com>
- XAMPP Software: <https://www.apachefriends.org>

### 4. Research Papers and Articles

- “The Role of ICT in the Tourism Industry” by Buhalis and Law, Elsevier, 2008.
- “Digital Transformation in Tourism: Challenges and Opportunities” by 2020.

### 5. APIs and Payment Gateways

- Razorpay API Documentation: <https://razorpay.com/docs/>
- PayPal Developer Resources: <https://developer.paypal.com>

### 6. Other References

- Personal notes and guidance from project mentors and college .

## ANNEXURE

### Sample Program Code

```
<?php
session_start();
include('includes/config.php');
if(strlen($_SESSION['alogin'])==0)
    {
header('location:index.php');
    }
else{
?>

<!DOCTYPE HTML>

<html>

<head>

<title>TMS | Admin Dashboard</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="application/x-javascript"> addEventListener("load", function() {
setTimeout(hideURLbar, 0); }, false); function hideURLbar(){
window.scrollTo(0,1); } </script>

<!-- Bootstrap Core CSS -->

<link href="css/bootstrap.min.css" rel='stylesheet' type='text/css' />

<!-- Custom CSS -->

<link href="css/style.css" rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="css/morris.css" type="text/css"/>

<!-- Graph CSS -->
```

```

<link href="css/font-awesome.css" rel="stylesheet">

<!-- jQuery -->
<script src="js/jquery-2.1.4.min.js"></script>
<!-- //jQuery -->

<link
href="//fonts.googleapis.com/css?family=Roboto:700,500,300,100italic,100,400'
rel='stylesheet' type='text/css'/>

<link href="//fonts.googleapis.com/css?family=Montserrat:400,700'
rel='stylesheet' type='text/css'>

<!-- lined-icons -->

<link rel="stylesheet" href="css/icon-font.min.css" type='text/css' />

<!-- //lined-icons -->

</head>

<body>

    <div class="page-container">

        <!--/content-inner-->

        <div class="left-content">

            <div class="mother-grid-inner">

                <!--header start here-->

                <?php include('includes/header.php');?>

                <!--header end here-->

                <ol class="breadcrumb">

                    <li class="breadcrumb-item"><a href="index.html">Home</a> <i
class="fa fa-angle-right"></i></li>

                </ol>

                <!--four-grids here-->

                <div class="four-grids">

                    <a href="manage-users.php" target="_blank">

```

```

<div class="col-md-4 four-grid">
    <div class="four-agileits">
        <div class="icon">
            <i class="glyphicon
glyphicon-user" aria-hidden="true"></i>
        </div>
        <div class="four-text">
            <h3>User</h3>

            <?php $sql = "SELECT id
from tblusers";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cnt=$query->rowCount();

?>            <h4> <?php echo
htmlentities($cnt);?> </h4>

        </div>
    </div>
</div>
</a>

<a href="manageissues.php" target="_blank">

```

```

        <div class="col-md-4 four-grid">
        <div class="four-w3ls">
            <div class="icon">
                <i class="glyphicon
glyphicon-folder-open" aria-hidden="true"></i>
            </div>
            <div class="four-text">
                <h3>Issues Riaised</h3>

```

```

        <?php $sql5 = "SELECT id from tblissues";
$query5= $dbh -> prepare($sql5);
$query5->execute();
$results5=$query5->fetchAll(PDO::FETCH_OBJ);
$cnt5=$query5->rowCount();
        ?>

```

```

        <h4><?php echo
htmlentities($cnt5);?></h4>

```

```

        </div>

```

```

    </div>

```

```

</div></a>

```

```

<a href="manage-packages.php" target="_blank">

```

```

        <div class="col-md-4 four-grid">

```

```

            <div class="four-wthree">

```

```

<div class="icon">
    <i class="glyphicon
glyphicon-briefcase" aria-hidden="true"></i>
</div>
<div class="four-text">
    <h3>Toatal packages</h3>

```

```

<?php $sql3 = "SELECT PackageId from
tbltourpackages";
$query3= $dbh -> prepare($sql3);
$query3->execute();
$results3=$query3->fetchAll(PDO::FETCH_OBJ);
$cnt3=$query3->rowCount();
?>

```

```

<h4><?php echo
htmlentities($cnt3);?></h4>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

<div class="clearfix"></div>

```

```

</div>

```

```

</a>

```

```

<a href="manage-enquires.php" target="_blank">

```

```

<div class="four-grids">

```

```

<div class="col-md-4 four-grid">
    <div class="four-agileinfo">
        <div class="icon">
            <i class="glyphicon
glyphicon-folder-open" aria-hidden="true"></i>
        </div>
        <div class="four-text">
            <h3>Enquiries</h3>

```

```

<?php $sql2 = "SELECT id from tblequiry";
$query2= $dbh -> prepare($sql2);
$query2->execute();
$results2=$query2->fetchAll(PDO::FETCH_OBJ);
$cnt2=$query2->rowCount();
?>

```

```

<h4><?php echo
htmlentities($cnt2);?></h4>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</a>

```

```

<a href="manage-enquires.php" target="_blank">

```

```

<div class="col-md-4 four-grid">

```

```

<div class="four-agileits">
    <div class="icon">
        <i class="glyphicon
glyphicon-folder-open" aria-hidden="true"></i>
    </div>
    <div class="four-text">
        <h3>New Enquiries</h3>

        <?php $sql ="SELECT id
from tblenquiry where (Status is null || Status=)";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$newenq=$query->rowCount();

        ?>        <h4> <?php echo
htmlentities($newenq);?> </h4>
    </div>
</div>
</div>
</div></a>

<a href="manage-enquires.php" target="_blank">
    <div class="col-md-4 four-grid">
        <div class="four-w3ls">
            <div class="icon">
                <i class="glyphicon
glyphicon-folder-open" aria-hidden="true"></i>
            </div>
            <div class="four-text">

```



<h3>Read Enquiries</h3>

```
<?php $sql5 ="SELECT id from tblequiry where (Status='1')";  
$query5= $dbh -> prepare($sql5);  
$query5->execute();  
$results5=$query5->fetchAll(PDO::FETCH_OBJ);  
$redenq=$query5->rowCount();  
?>
```

```
<h4><?php echo  
htmlentities($redenq);?></h4>
```

</div>

</div>

</div>

</a>

<div class="clearfix"></div>

</div>

<div class="four-grids">

<a href="manage-bookings.php" target="\_blank">

<div class="col-md-3 four-  
grid">

<div class="four-agileinfo">

<div class="icon">

<i class="glyphicon  
glyphicon-list-alt" aria-hidden="true"></i>

</div>

```

<div class="four-text">
    <h3>Bookings</h3>
    <?php $sql1 =
"SELECT BookingId from tblbooking";
$query1 = $dbh -> prepare($sql1);
$query1->execute();
$results1=$query1->fetchAll(PDO::FETCH_OBJ);
$cnt1=$query1->rowCount();
?>
    <h4><?php echo
htmlentities($cnt1);?></h4>
</div>
</div>
</div>
</div></a>
<a href="manage-bookings.php" target="_blank">
<div class="col-md-3 four-grid">
    <div class="four-wthree"
style="color:#ffc107 !important">
        <div class="icon">
            <i class="glyphicon
glyphicon-list-alt" aria-hidden="true"></i>
        </div>
        <div class="four-text">
            <h3>New Bookings</h3>

```

```

                                <?php $sql ="SELECT
BookingId from tblbooking where (status is null || status=");
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$newbookings=$query->rowCount();
                                ?>                                <h4> <?php echo
htmlentities($newbookings);?> </h4>
                                </div>
                                </div>
                                </div>
</a>

```

```

<a href="manage-bookings.php" target="_blank">

<div class="col-md-3 four-grid">
                                <div class="four-agileits">
                                    <div class="icon">
                                        <i class="glyphicon
glyphicon-list-alt" aria-hidden="true"></i>
                                    </div>
                                    <div class="four-text">
                                        <h3>Cancelled
Bookings</h3>

```

```

                                <?php $sql ="SELECT
BookingId from tblbooking where (status='2')";
$query = $dbh -> prepare($sql);

```

```

$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cancelbooking=$query->rowCount();

        ?>                <h4> <?php echo
htmlentities($cancelbooking);?> </h4>

                </div>

        </div>

</div>

</a>

        <a href="manage-bookings.php" target="_blank">

                <div class="col-md-3 four-grid">

                        <div class="four-w3ls">

                                <div class="icon">

                                        <i class="glyphicon
glyphicon-list-alt" aria-hidden="true"></i>

                                </div>

                                <div class="four-text">

                                        <h3>Confirmed
Bookings</h3>

                                <?php $sql ="SELECT
BookingId from tblbooking where (status='1')";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);
$cancelbooking=$query->rowCount();

        ?>                <h4> <?php echo
htmlentities($cancelbooking);?> </h4>

```

```

        </div>
    </div>
</div>
</a>
    <div class="clearfix"></div>
</div>

<!--//four-grids here-->


<div class="inner-block">


</div>
<!--inner block end here-->
<!--copy rights start here-->
<?php include('includes/footer.php');?>
</div>
</div>


        <!--/sidebar-menu-->
        <?php include('includes/sidebarmenu.php');?>
            <div class="clearfix"></div>


        </div>
        <script>
            var toggle = true;


            $(".sidebar-icon").click(function() {

```

```

        if (toggle)
        {
            $(".page-
container").addClass("sidebar-collapsed").removeClass("sidebar-collapsed-
back");

            $("#menu
span").css({"position":"absolute"});

        }
        else
        {
            $(".page-
container").removeClass("sidebar-collapsed").addClass("sidebar-collapsed-
back");

            setTimeout(function() {
                $("#menu
span").css({"position":"relative"});

            }, 400);

        }

        toggle =
!toggle;

    });

</script>

<!--js -->
<script src="js/jquery.nicescroll.js"></script>
<script src="js/scripts.js"></script>
<!-- Bootstrap Core JavaScript -->
    <script src="js/bootstrap.min.js"></script>
    <!-- /Bootstrap Core JavaScript -->

```

```

<!-- morris JavaScript -->
<script src="js/raphael-min.js"></script>
<script src="js/morris.js"></script>
<script>
    $(document).ready(function() {
        //BOX BUTTON SHOW AND CLOSE
        jQuery('.small-graph-box').hover(function() {
            jQuery(this).find('.box-button').fadeIn('fast');
        }, function() {
            jQuery(this).find('.box-button').fadeOut('fast');
        });
        jQuery('.small-graph-box .box-close').click(function() {
            jQuery(this).closest('.small-graph-box').fadeOut(200);
            return false;
        });

        //CHARTS
        function gd(year, day, month) {
            return new Date(year, month - 1, day).getTime();
        }

        graphArea2 = Morris.Area({
            element: 'hero-area',
            padding: 10,
            behaveLikeLine: true,
            gridEnabled: false,

```

```

    gridLineColor: '#dddddd',
    axes: true,
    resize: true,
    smooth:true,
    pointSize: 0,
    lineWidth: 0,
    fillOpacity:0.85,
    data: [
        {period: '2014 Q1', iphone: 2668, ipad: null, itouch:
2649},
        {period: '2014 Q2', iphone: 15780, ipad: 13799, itouch:
12051},
        {period: '2014 Q3', iphone: 12920, ipad: 10975, itouch:
9910},
        {period: '2014 Q4', iphone: 8770, ipad: 6600, itouch:
6695},
        {period: '2015 Q1', iphone: 10820, ipad: 10924, itouch:
12300},
        {period: '2015 Q2', iphone: 9680, ipad: 9010, itouch:
7891},
        {period: '2015 Q3', iphone: 4830, ipad: 3805, itouch:
1598},
        {period: '2015 Q4', iphone: 15083, ipad: 8977, itouch:
5185},
        {period: '2016 Q1', iphone: 10697, ipad: 4470, itouch:
2038},
        {period: '2016 Q2', iphone: 8442, ipad: 5723, itouch:
1801}
    ],

```



```
        lineColors:['#ff4a43','#a2d200','#22beef'],
        xkey: 'period',
    redraw: true,
    ykeys: ['iphone', 'ipad', 'itouch'],
    labels: ['All Visitors', 'Returning Visitors', 'Unique Visitors'],
        pointSize: 2,
        hideHover: 'auto',
        resize: true
    });

});
</script>
</body>
</html>
<?php } ?>
```

