Table of Content

TABLE OF CONTENT i

[LIST OF FIGURES ii](#_Toc198537704)

[LIST OF TABLES iii](#_Toc198537705)

[LIST OF ABBREVIATIONS iv](#_Toc198537706)

[CHAPTER 1 1](#_Toc198537707)

[INTRODUCTION 1](#_Toc198537708)

[1.1 Background 1](#_Toc198537709)

[1.2 Problem Statement 1](#_Toc198537710)

[1.3 Objectives 2](#_Toc198537711)

[CHAPTER 2 3](#_Toc198537712)

[METHODOLOGY 3](#_Toc198537713)

[2.1 System Requirements 3](#_Toc198537714)

[2.1.1 Functional Requirements 3](#_Toc198537715)

[2.1.2 Non-Functional Requirements 4](#_Toc198537716)

[2.1.3 Requirement Prioritization Table 5](#_Toc198537717)

[2.2 Design 6](#_Toc198537718)

[CHAPTER 3PROJECT SCHEDULE 10](#_Toc198537719)

[3.1 Gantt Chart 10](#_Toc198537720)

[CHAPTER 4 11](#_Toc198537722)

[POSSIBLE OUTCOMES AND FUTURE WORK 11](#_Toc198537723)

[4.1. Possible Outcome 11](#_Toc198537724)

[4.2 Future Work 11](#_Toc198537725)

[REFERENCES 13](#_Toc198537726)

# LIST OF FIGURES

|  |  |
| --- | --- |
| **Figures** | **Page No.** |
| Figure 2.1: Use Case Diagram of Hamro yatra | 6 |
| Figure 2.2: Schema Diagram of Hamro yatra | 7 |
| Figure 2.3: Flow Chart of Hamro yatra | 8 |
| Figure 2.4: Sequential Diagram of Hamro yatra | 9 |
| Figure 3.1: Gantt Chart | 10 |

# LIST OF TABLES

|  |  |
| --- | --- |
| **Tables** | **Page No.** |
| Table 2.1: Requirement Prioritization Table | 5 |
|  |  |

# LIST OF ABBREVIATIONS

BCA Bachelor’s in Computer Application

CSS Cascading Style Sheet

DB Data base

HTML Hyper Text Markup Language

# CHAPTER 1

## INTRODUCTION

## 1.1 Background

The rapid growth of the tourism industry has created a strong demand for digital platforms that simplify travel planning and booking. In today’s fast-paced environment, most travelers prefer to manage their tours online instead of relying on traditional travel agencies. They expect quick access to information, real-time availability, and the convenience of booking packages from anywhere.

Currently, many travelers face challenges when trying to compare different travel options, check availability, and make secure bookings in one place. Similarly, travel agencies often lack efficient systems to manage bookings, update tour information, and maintain records. These gaps create inefficiencies for both tourists and service providers.

To address these issues, this project proposes the development of a **Hamro yatra**—a web-based application that allows users to browse, search, and book travel packages easily. The system will also provide administrators with tools to manage package listings, handle bookings, and ensure smooth operation of the platform. Once implemented, this system will offer a user-friendly and centralized solution for both tourists and tour operators, improving the overall experience and operational efficiency.

## 1.2 Problem Statement

In the current scenario, many tourists still rely on offline methods or scattered online resources to plan and book their travel packages. This can lead to confusion, limited access to available options, booking delays, and a lack of transparency in pricing and availability. Traditional booking systems are also prone to manual errors and do not offer personalized recommendations based on user preferences or history.

Furthermore, for travel service providers and administrators, managing bookings, packages, and customer records manually is time-consuming and inefficient. It becomes challenging to keep the data up-to-date and accessible to users in real time.

Hence, there is a strong need for a centralized, user-friendly web application that simplifies the travel package booking process for both tourists and administrators, while ensuring accuracy, speed, and security

## 1.3 Objectives

The primary objective of the Tourist Ticket Booking Management System is to develop a web-based application that simplifies the process of browsing, selecting, and booking travel packages. This system aims to meet the needs of both tourists and administrators through the following specific objectives:

1. **Provide a convenient platform** for tourists to search and book travel packages based on destination, budget, and duration.
2. **Enable real-time booking confirmation** and show up-to-date availability status.
3. **Offer secure login and authentication** for both users and administrators.
4. **Allow administrators to manage** travel packages, bookings, and customer records through a centralized dashboard.
5. **Reduce manual errors, paperwork,** and booking delays.

# CHAPTER 2

## METHODOLOGY

Different steps were followed in order to develop the entire system. Here’s the breakdown of each of the phases.

## 2.1 System Requirements

The successful development of the *Hamro Yatra – Tourist Management System* depends on understanding and implementing clear system requirements. These requirements are divided into three main categories: Functional Requirements, Non-Functional Requirements, and Requirement Prioritization.

### 2.1.1 Functional Requirements

Functional requirements define the specific behavior and features of the system. They outline what the system should do to meet the needs of both tourists (users) and administrators.

Tourist-side Functionalities:

* Tourists can register and log in securely.
* They can browse available tour packages easily.
* They can search for packages using filters such as destination, date, duration, and budget.
* Tourists can view detailed package information, including itinerary, pricing, and photos.
* Users can book tour packages and receive booking confirmation.
* They can view their booking history and status.
* If allowed, tourists can cancel or reschedule their bookings based on policy.

Administrator-side Functionalities:

* Admins can log in securely to the system.
* Admins have access to add, edit, or delete tour packages.
* They can view all bookings and update booking statuses.
* Admins can manage user accounts, including activating or deactivating tourist accounts.
* Optionally, admins can access reports and usage statistics to monitor system activity.

### 2.1.2 Non-Functional Requirements

Non-functional requirements define how the system performs rather than what it does. They ensure the system is efficient, secure, and scalable.

Usability:

* The interface should be simple, clean, and user-friendly for both tourists and admins.
* Users should navigate the system intuitively without requiring training.

Performance:

* Pages should load quickly, and the system should process bookings within a few seconds.
* It must support multiple users accessing the system simultaneously without lag.

Scalability:

* The system should be scalable to support a growing number of users, packages, and bookings as the platform expands.

Security:

* User data must be protected through secure logins and encrypted passwords.
* Admin functionalities should be accessible only to authorized users.

Availability:

* The platform should remain accessible 24/7 to allow users to browse and book packages at any time.

Maintainability:

* The system should be easy to maintain, fix bugs, and improve over time with minimal disruption.

Compatibility:

* The platform should function smoothly on various devices (desktop, mobile, tablet) and across different browsers (Chrome, Firefox, Safari, etc.).

### 2.1.3 Requirement Prioritization Table

| **Requirement** | **Type** | **Priority** |
| --- | --- | --- |
| User registration and login | Functional | High |
| Admin login and dashboard access | Functional | High |
| Search travel packages by destination | Functional | High |
| Filter and sort travel packages | Functional | Medium |
| View full package details | Functional | High |
| Book a travel package | Functional | High |
| Recommendation system | Functional | Medium |
| View and manage booking history | Functional | Medium |
| Add/edit/delete packages (admin) | Functional | High |
| View and manage bookings (admin) | Functional | High |
| Secure access control | Non-Functional | High |
| User-friendly interface | Non-Functional | High |
| Fast loading and response time | Non-Functional | Medium |
| Cross-device and browser compatibility | Non-Functional | Medium |
| 24/7 availability | Non-Functional | Medium |
| Easy to update and maintain | Non-Functional | Medium |

Table 2.1: Requirement Prioritization Table

## 2.2 Design

The design phase plays a vital role in the development of the Hamro Yatra – Tourist Management System. This phase focuses on building a secure, user-friendly, and scalable platform that allows tourists to easily book tours and enables administrators to manage packages and users effectively.

The system follows a modular design approach, where each part (such as login, booking, and admin dashboard) is developed independently but works together as a complete system. The design includes system architecture, database structure, and interface layout to ensure efficiency and maintainability.

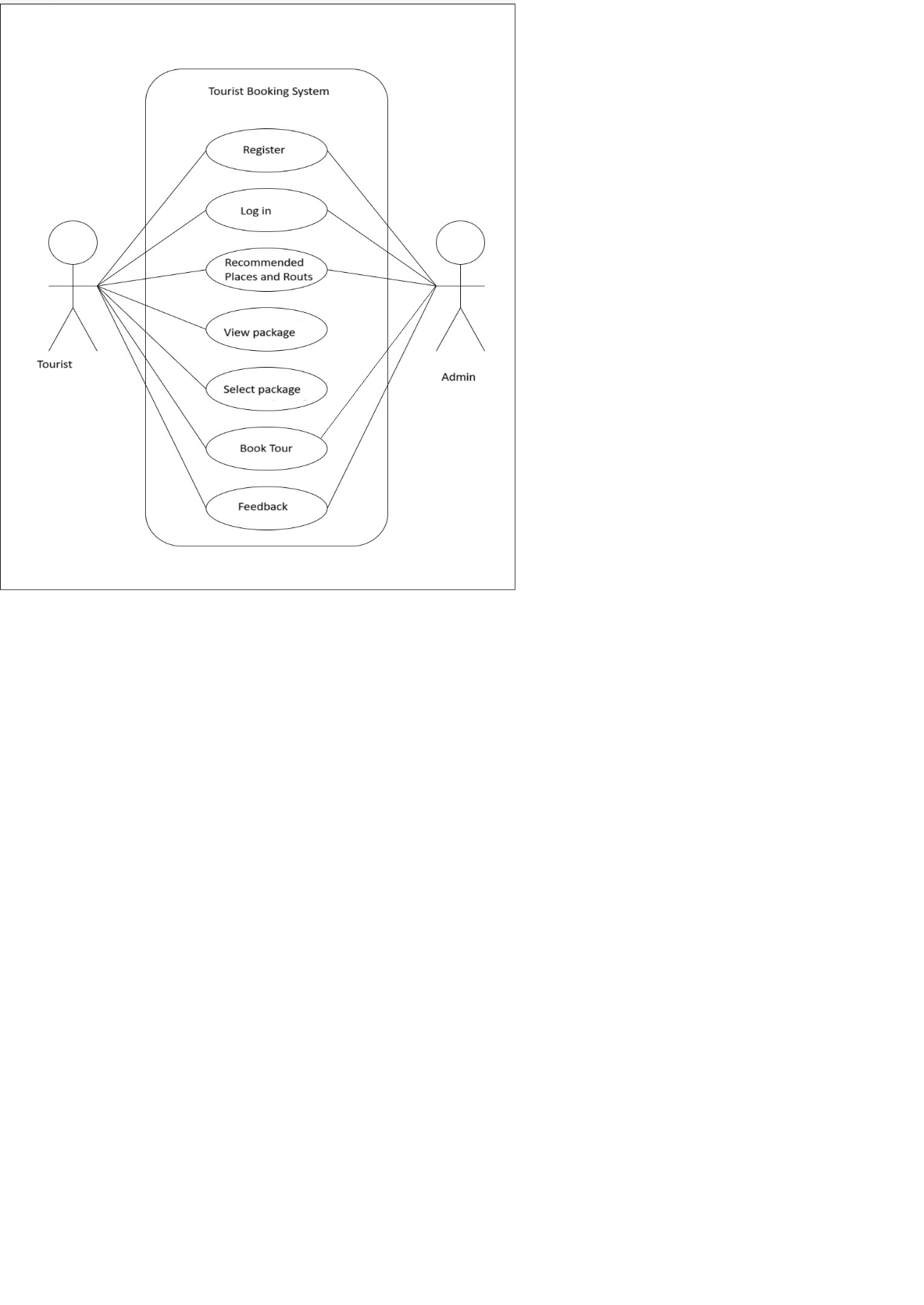


Figure 2.1: Use Case Diagram of Hamro yatra

A use case diagram for a tourist booking system visually represents the interactions between users (like tourists and administrators) and the system. It shows what users can do, such as book a tour or managing bookings, and highlights the main functions of the system. This diagram helps in understanding the system's features and how different users interact with it.

A diagram of a computer

AI-generated content may be incorrect.

Figure 2.2: Schema Diagram of Hamro yatra

The Schema diagram for Hamro Yatra represents the core structure of how the system operates. The system involves several key entities: User, Admin, Booking, and Map Routes. Users register with a unique ID, username, email, and password, and they can select or search for tourist locations. These locations, defined by a name and ID, are displayed in a store, which holds multiple locations and is managed by an admin. Admins have their own credentials and are responsible for overseeing the stores. Users can make bookings based on selected locations, where each booking stores details like cost and distance. Additionally, users can view and search map routes using route names and distances to help plan their trips effectively. This model ensures a well-organized and interactive travel management system for tourists.

A diagram of a flowchart

AI-generated content may be incorrect.

Figure 2.3: Flow Chart of Hamro yatra

**Hamro Yatra** offers a seamless and user-friendly platform for both tourists and administrators. Tourists can register or log in, explore and filter tour packages, confirm bookings to receive a unique Booking ID, and view their booking history. Admins log in securely to manage tour packages and bookings, including adding, editing, deleting, canceling, or rescheduling. Both users conclude their sessions by logging out. The system ensures role-based access, simplicity, and efficient tour management for a smooth user journey.

A diagram of a diagram of a building

AI-generated content may be incorrect.

Figure 2.4: Sequential Diagram of Hamro yatra

**Hamro Yatra** illustrates a streamlined and interactive booking process. Tourists begin by logging into the system, with validations in place to handle incorrect credentials. Once logged in, users browse and select from available tour packages, which are dynamically checked against the database for real-time availability. Upon confirmation, the system updates the booking records and ensures data consistency. In cases where packages are unavailable, alternatives can be suggested. The backend supports admin-level management and package updates. Overall, the process emphasizes smooth user experience, automated booking updates, and robust communication across components like the login, dashboard, database, and admin systems.

# CHAPTER 3PROJECT SCHEDULE

## 3.1 Gantt Chart

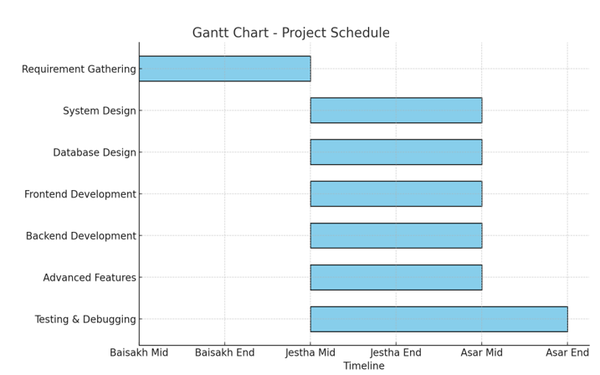


Table 3.1: Gantt Chart

The Gantt chart visually represents your project's timeline across the months of Baisakh, Jestha, and Asar. It outlines the major tasks and shows when each will be carried out. The project starts in Baisakh with requirement gathering and system design, which helps lay the foundation. In Jestha, the development phase takes place — this includes database design, frontend, backend, and implementing advanced features. Finally, in Asar, the focus shifts to testing, debugging, and finalizing the report.

# This structured timeline ensures smooth project flow, avoids task overlap, and helps manage the project efficiently within the planned duration.

# CHAPTER 4

## POSSIBLE OUTCOMES AND FUTURE WORK

## 4.1. Possible Outcome

The Tourist Ticket Booking Management System is expected to achieve several key outcomes once it is implemented and deployed:

Improved User Experience Tourists will benefit from a streamlined, easy-to-navigate platform that allows them to search for, compare, and book travel packages efficiently. The system’s user-friendly interface, real-time availability, and secure booking process will make travel planning more accessible and less time-consuming.Enhanced Administrative Control: Administrators will be able to manage travel packages, user bookings, and customer records from a centralized dashboard. The automated management of bookings, coupled with a secure login system, will reduce human errors and save administrative time.Increased Efficiency: The system will eliminate manual handling of bookings, package updates, and user queries, thereby reducing the administrative workload and minimizing potential booking conflicts. This increased efficiency will be especially beneficial for travel agencies looking to scale their operations.Real-Time Booking and Availability: The system will provide tourists with up-to-date information on package availability, ensuring that they can book tours in real-time without delays. This real-time functionality will enhance trust and transparency between the service provider and customers.Scalable and Maintainable System: The system will be designed with future growth in mind, ensuring that it can handle an increasing number of users, bookings, and new features over time. The modular architecture will facilitate easy updates and maintenance.Data Insights for Administrators: Administrators will be able to generate reports and gain insights into booking trends, popular packages, and user demographics, allowing for better decision-making and marketing strategies.Overall, the expected outcome of this project is to create a highly efficient, user-friendly, and scalable platform that transforms the way tourists book travel packages, while also providing travel agencies with a more streamlined and organized way to manage their operations.

4.2 Future Work

While the primary goal of this project is to build a functional tourist ticket booking system, there are several potential future enhancements that can be incorporated, such as:Payment Gateway Integration:Integrating a secure payment gateway (e.g., PayPal, Stripe) to allow tourists to pay directly for bookings, making the system more comprehensive. Mobile Application:Developing a mobile version of the system to allow tourists to book and manage packages from their smartphones.Personalized Recommendations:Implementing a recommendation engine that suggests travel packages based on user behavior, preferences, or past bookings. Multilingual Support:Expanding the system to support multiple languages, enabling tourists from different regions to use the platform more effectively.Customer Support Chatbot:Adding a chatbot for real-time customer support to answer common questions, assist with booking issues, or provide tour details.These future enhancements will improve the system’s functionality and widen its user base.

# REFERENCES

[1] Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.

[2] Pressman, R. S. (2020). *Software Engineering: A Practitioner’s Approach*. McGraw-Hill.

[3] Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems*. Pearson.

[4] World Travel & Tourism Council. (2023). *Economic Impact Report: Nepal*. Retrieved from [WTTC](https://wttc.org/)

[5] Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann.