CHAPTER 1

INTRODUCTION

## Background

The tourism sector is a crucial driver of economic growth, particularly in developing countries. It offers opportunities to create productive and inclusive jobs, stimulate innovative businesses, and finance the conservation of natural and cultural assets. The impact of tourism extends beyond direct economic benefits; it fosters cultural exchange and understanding, contributing to global peace and cooperation.

However, the tourism industry also faces significant challenges. Physical documents are susceptible to damage, loss, and deterioration over time. Natural disasters can destroy paper records, leading to permanent loss of valuable data. Manual processes for booking and managing tourist information are time-consuming and prone to human error. Long queues and delays can frustrate tourists and reduce the overall efficiency of service providers. Inconsistent data entry and difficulties in retrieving information further exacerbate these inefficiencies. These issues can diminish the quality of the tourist experience and negatively impact local communities and ecosystems. Therefore, sustainable tourism practices and efficient management systems are essential to mitigate these negative effects.

The development of a comprehensive Tourist Management System (TMS) is proposed to address these issues by streamlining operations, improving tourist satisfaction, and promoting sustainable practices. A TMS can integrate various functions such as ticket booking, information dissemination, and feedback collection into a unified platform. By integrating a Ticket Booking System (TBS) within the TMS framework, this initiative aims to simplify the booking process, enhance accessibility, and provide a more seamless and enjoyable experience for tourists. This approach not only addresses current inefficiencies but also lays the foundation for a more resilient and sustainable tourism sector.

Moreover, a well-designed TMS can support data-driven decision-making. By collecting and analysing data on tourist behaviour, preferences, and impacts, tourism managers can identify trends, predict future demand, and implement targeted interventions. This can help in optimizing resource use, improving marketing strategies, and enhancing overall visitor experiences.

While tourism offers significant economic and social benefits, it also presents challenges that require careful management. The development and implementation of a comprehensive Tourist Management System, including a Ticket Booking System, represent a proactive approach to achieving these goals.

## **Statement of the problem**

# In the current landscape of tourism, one of the significant challenges faced by tourists and service providers a like is the cumbersome process of ticket booking and managing.

# Traditional methods often involve long queues, manual paperwork, and limited accessibility, leading to frustration, inefficiencies, and missed opportunities.

# There is a critical need for a streamlined Ticket Booking System (TBS) within a comprehensive Tourist Management System (TMS) framework to simplify the booking process, improve accessibility, and enhance overall tourist experiences.

## **1.3 Project objectives**

# The objective of this project is to streamline and automate the ticket booking process for tourists to enhance efficiency and accuracy in managing ticket reservations.

# Automation of Ticket Booking:

# The primary goal is to eliminate manual ticket booking processes by implementing a system that automatically manages ticket bookings in real-time, including the creation, viewing, and deletion of tickets. This system aims to provide a seamless and user-friendly experience for both tourists and administrators, ensuring accurate and efficient management of ticket reservations.

# **Limitation**

# The limitations of the provided Tourist Management System (TMS) program are:

# Concurrent Access: No support for handling simultaneous file access.

# Interface: Command-line based, not user-friendly.

# Security: lacks comprehensive security.

# Error Handling: Basic error handling; program often terminates on errors.

Scalability: Performance may degrade with many records.

Limited flexibility for adding new fields or changing data formats

CHAPTER 2

LITERATURE REVIEW

Tourism Management Systems (TMS) play a pivotal role in optimizing operations within the tourism industry. Researchers and practitioners have explored various approaches to designing and implementing TMS using the C/C++ programming languages. Below, we highlight key studies and their contributions:

Existing Resources:

1. A Tourism Management System in C (GeeksforGeeks, 2022)

This tutorial outlines a basic tourist management system in C. Users can book, cancel, and view tour packages, manage accounts, and change passwords. The source code is freely downloadable.

1. Travel Agency Management System in C++ with MySQL (Code With C, 2023)

This article showcases a C++ travel agency management system that interacts with a MySQL database. Users can add, edit, and delete traveler and trip information. The article details the program's functionalities.

1. Travel Management System (JAHANGEERQADIREE, 2013)

This SourceCodester resource details a Travel Management System with functionalities that include an About Us section for travel agency information, ticket booking and cancellation, train information display (availability, arrival/departure times), and domestic and international flight searches (one-way/round-trip). It allows for departure and arrival date selection, captures passenger details such as gender and the number of adults or children, offers class selection (business/economy), and generates fares based on user input.

Extensive research indicates that developing TMS using the C language has been a persistent focus in both academia and industry. The findings highlight C's effectiveness in managing the intricate aspects of tourist operations. Additionally, there is a continuous drive to improve these systems' functionality and adaptability to cater to the ever-changing demands of the tourism sector in the digital era (World Bank).

CHAPTER 3

## **METHODOLOGY**

This Tourist Management System is designed to manage tourist information using a text-based interface. We begin with planning, then gather and analyze information. Next, we design and develop the system, followed by testing.

## **Figure 3.1: Methodology**

3.1 Planning Phase

The team first had discussions and brainstorming sessions to determine the project scope, and they agreed to develop a Tourist Management System (TMS). They then created a detailed plan that outlined the development approach, timeline, budget, and task assignments. After that the project goal and objective were also decided finally detailed Requirement Analysis was done.

**3.2** **Requirement Analysis Phase**

User requirements were also collected from stakeholders, including system users and administrators. Then entire system was thoroughly analyzed to identify all the requirements for the project and defined to ensure that the project would be developed in a way that fulfills their criteria and main purpose for creating the project.

TheseKey functionalities identified were:

* Adding, deleting, and viewing tourist records
* Secure file handling for data persistence
* Robust error handling to ensure reliability

**3.3** **System Design Phase**

The system architecture was designed to encompass the following modules:

* **Information Management**: Handling tourist data such as name, destination, date, and ticket number.
* **Record Operations**: Implementing functions to add, delete, and view records by ticket number.
* **File Handling**: Managing data storage and retrieval using CSV files for persistence.

The user interface was designed as a Character/Command Line User Interface (CUI), ensuring simplicity and efficiency in user interactions.

**3.4** **System Development Phase**

The Tourist Management System was developed using the C programming language. Key tasks included:

* Defining Data Structures: Structuring Tourist and Destination structures to store relevant information.
* Generating Ticket Numbers: Implementing a function to generate unique ticket numbers.
* File Operations: Writing functions to perform CRUD operations on tourist records stored in a CSV file.
* Error Handling: Implementing mechanisms to manage and report errors, ensuring system stability.

# **3.5 Testing Phase**

Comprehensive testing was conducted to validate the system's functionality:

* **Unit Testing**: Individual testing of each function to verify correctness.
* **Integration Testing**: Ensuring seamless interaction between system components.
* **User Acceptance Testing (UAT)**: Soliciting feedback from potential users to ensure the system meets their needs and expectations.

Following this structured methodology used for the development of Tourist Management System that efficiently manages tourist records, aligning closely with user requirements while ensuring reliability and efficiency.

**3.6. Flow chart**

The Tourist Management System (TMS) program efficiently manages tourist information and destinations through a user-friendly menu-driven interface. The flow chart outlines the sequence of operations, from initiation to termination. The program begins with the Start step. Once started, the user is presented with a menu of six options: adding a new tourist, deleting tourist data, adding a new destination, displaying information about a specific tourist, displaying all tourist data, and exiting the program. The user's selection determines the subsequent branch of operations.

Choice Processing handles the user's selection by branching into the corresponding task:

1. If Choice == 1 (Add Tourist), the program prompts the user to input the tourist's name and select a destination. The current date is automatically selected to ensure accurate record-keeping. A unique ticket number is generated and stored in the database. Then the program returns to the main menu.
2. If Choice == 2 (Delete Tourist Data): Using the provided ticket number, the program removes the tourist's record from the database. After the record is deleted, the program returns to the main menu.
3. If Choice == 3 (Display Tourist Data): Using the provided ticket number, the program displays the tourist's record from the database. Then the program returns to the main menu.
4. If Choice == 4 (Display All Data): The program retrieves all tourist records from the database. It displays this comprehensive list, allowing the user to view all stored tourist information. After displaying the data, the program returns to the main menu.
5. If Choice == 5 (Add New Destination): The program prompts the user to provide details about a new destination. This new destination data is saved to the database for future use. After successfully adding the destination, the program returns to the main menu.
6. If Choice == 6 (Exit): The program terminates, ending the execution.

The flow chart ensures each operation is processed efficiently, with the program returning to the main menu after each task. This structure simplifies user interactions and keeps the database up to date, facilitating effective tourist management.

Enter choice: **1** to add tourist, 2 to delete data, 3 to display specific tourist data ,**4** to display all data,**5** to add new destination, **6** to exit

If choice== 1

If choice =6

6

If choice==2

Add tourist.

Delete tourist data.

Invalid choice

Yes

Yes

Yes

No

Yes

If choice =5

If choice==3

Add new destination

If choice==4

Display Tourist data

Find the key

Not found

Display all data of tourist

No

No

No

Yes

No

No

found

**Figure 3.2: Flow chart of TMS**

3.7 Data Flow Diagram(DFD)

Level 0 DFD provides an overall view of the Tourist Management System. Where admin is the entity that interacts with the system and is responsible for managing and controlling it and sending commands to the system. The Tourist Management System accepts commands and implements them. The database is the central data storage unit, storing information and providing data to the Tourist Management System when needed.

### Request

Database

# Admin

Response

### Figure 3.3 Level 0 DFD of TMS

Level 1 DFD offers a detailed view of the system by breaking down major processes. In our system, the admin manages the system by issuing commands such as adding a tourist, viewing tourist data, deleting data, adding a destination, and exiting the program.

When the admin sends requests like "Add Tourist," "View Tourist," "Delete Tourist," "Add New Destination," and "Exit," the system processes these requests and generates appropriate responses. When adding tourist data, the system generates a ticket, selects the current date, saves the data to the database, and provides the ticket to both the tourist and the admin.

The database is the central storage unit, maintaining all information about tourists, destinations, ticket numbers, and other relevant details.

This setup ensures seamless data flow between the admin, system and the database, with requests sent from the admin to the processes and responses returned to the admin.

Tourist

Response

Request

# Admin

Response

Request

Response

Database

Request

Response

Request

Response

Request

Response

Figure 3.4: Level 1 DFD of TMS

CHAPTER 4

RESULT AND DISCUSSION

## The Tourist Management System (TMS) project has been successfully implemented with a comprehensive system for managing tourist bookings, destinations, and related data. The system offers a user-friendly interface for users to perform various tasks, including:

1. **Ticket Booking:** Successfully creates a new ticket with a unique ID, recording the tourist's name, chosen destination, generated ticket number, and automatic date of entry.
2. **Destination Management:** Allows the addition of new tourist destinations to the system, with admins inputting the name, price, and facilities of the new destination using the key Desti111.
3. **Ticket Viewing:** Displays the requested ticket information, including the ticket ID, tourist’s name, date of entry, and destination.
4. **Data Deletion:** Securely and permanently deletes a tourist’s data from the system.
5. **File Operations:** Creates a "tourist.csv" Excel file, supporting operations to view and copy data from the CSV file where tourist records are stored.

Overall, the Tourist Management System project demonstrates the effectiveness of using structured programming to develop a comprehensive management system. This project was designed with key functionalities including booking tickets, adding destinations, viewing tickets, deleting tourist data, and managing file operations. This project not only reinforces core programming concepts, such as file handling and data structures, but also emphasizes the importance of creating user-centric applications. Through repeated testing and improvements, each feature was adjusted to manage tourist records easily and efficiently. This project serves as a solid foundation for more complex systems and highlights the versatility of structured programming in developing functional and efficient software solutions.

CHAPTER 5

CONCLUSION AND FUTUREWORKS

### 5.1. Conclusion

The Tourist Management System (TMS) application offers an efficient solution for managing tourist records, designed with flexibility for future updates and modifications. Developed using the VS Code text editor, this application simplifies the process of handling tourist information.

The primary goal of creating the "Tourist Management System" was to streamline the management of tourist data, including ticket bookings, viewing, and deleting records. The application enables the storage and retrieval of essential tourist information, making it easier to manage bookings and enhance the overall tourist experience. The development process utilized tools such as VS Code, 64-bit laptops (Lenovo Intel Core i5, and Acer Aspire 7), and both SSD and HDD storage solutions.

### 5.2. Future works

The following features are expected to be developed soon which may also depend upon user feedback and responses

* Design the system to handle many tourists and data
* save the data of tourist after signing off.
* Implement a feedback mechanism for users to provide by agent stake holder s and report issues.
* Developing a website and a GUI-based application for tourists to manage their travel needs.

REFERENCE

World Bank. (2022). Tourism and Competitiveness. Retrieved from <https://www.worldbank.org/en/topic/competitiveness/publication/tourism-and-competitiveness>

Pressman, R. S. (2014). Software Engineering: A Practitioner’s Approach. McGraw Hill Education.

Image (2024). Waterfall Project Management Approach. Retrieved from <https://www.octalsoftware.com.sg/waterfall-project-management-approach>

SuiteFiles.(2023). Folder Structures Guide. Retrieved from https://www.suitefiles.com/folder-structures-guide

Information Technology & Tourism (ITTJ) Journal. Retrieved from [https://www.ingentaconnect.com](https://www.ingentaconnect.com/)

Project Management Institute (PMI). A Guide to the PMBOK Guide. Retrieved from <https://www.pmi.org/pmbok-guide-standards/foundational/pmbok>

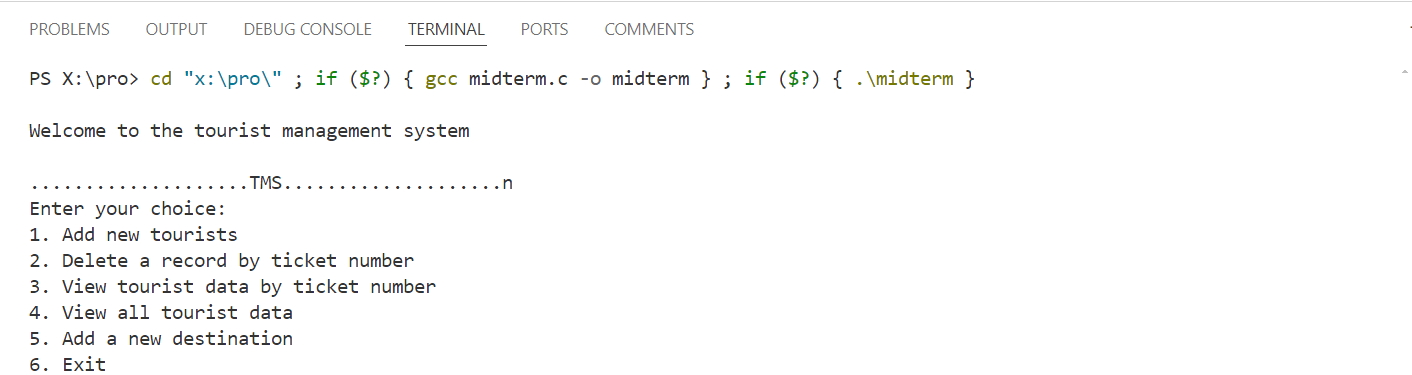
GeeksforGeeks. (2022). A tourism management system in C. Retrieved from <https://www.geeksforgeeks.org/tourism-management-system-in-c-c/>

Code With C. (2023). Travel agency management system in C++ with MySQL. Retrieved from<https://www.codewithc.com/travel-agency-management-system-in-c-withmysql/?amp=1>

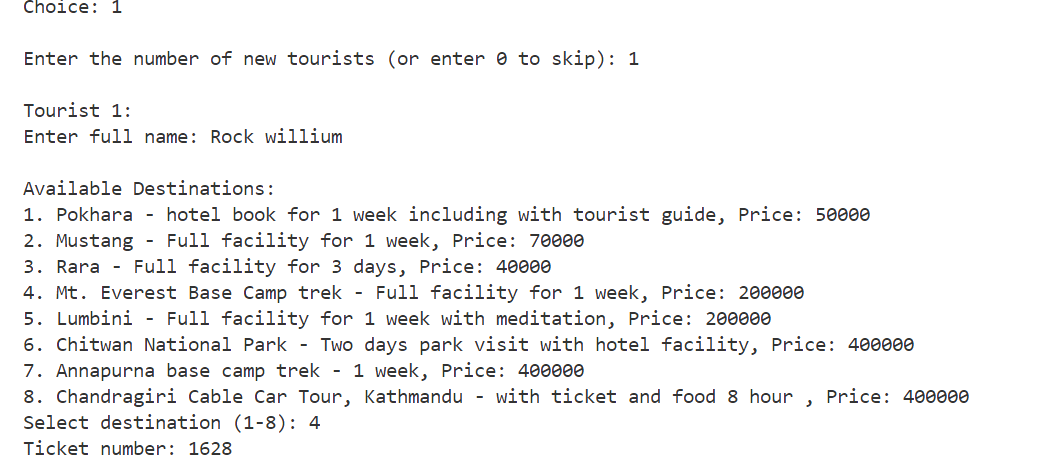
JAHANGEERQADIREE. (2013). Travel management system. Retrieved from <https://www.sourcecodester.com/cc/6165/travel-management-system.html>

APPENDICES

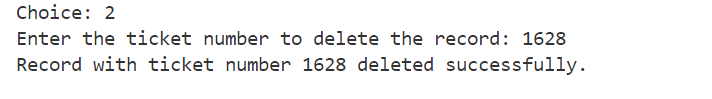
Appendix:1 Main Menu:



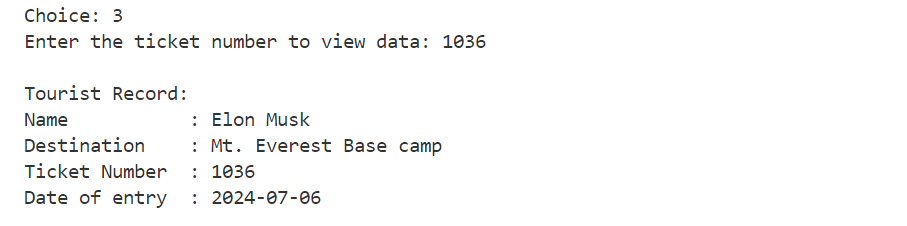
Appendix:2 Add new tourists:



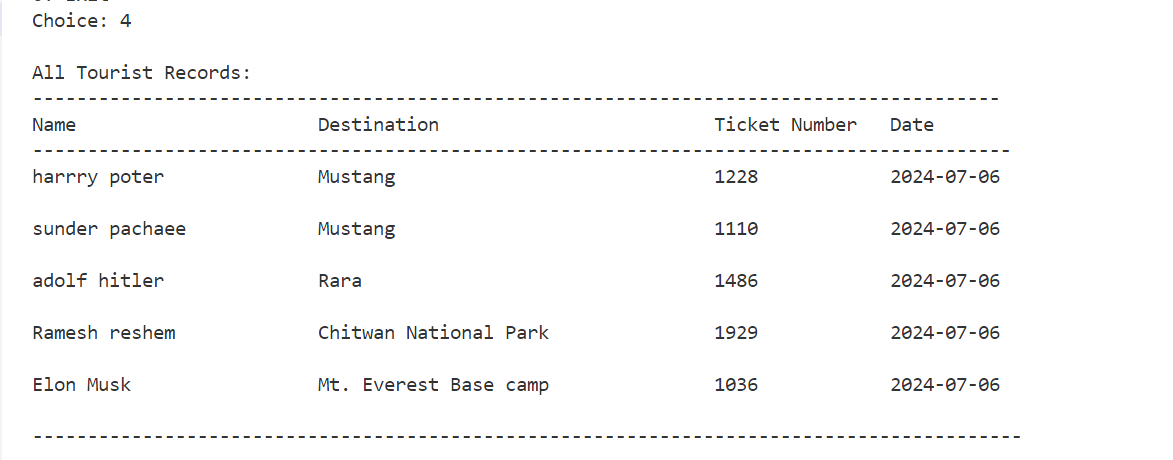
Appendix:3 Delete record of tourist:



Appendix:4 View tourist data by ticket number:

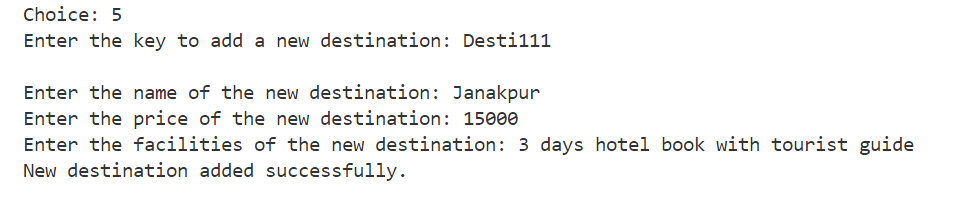


Appendix:5 View all tourist data:

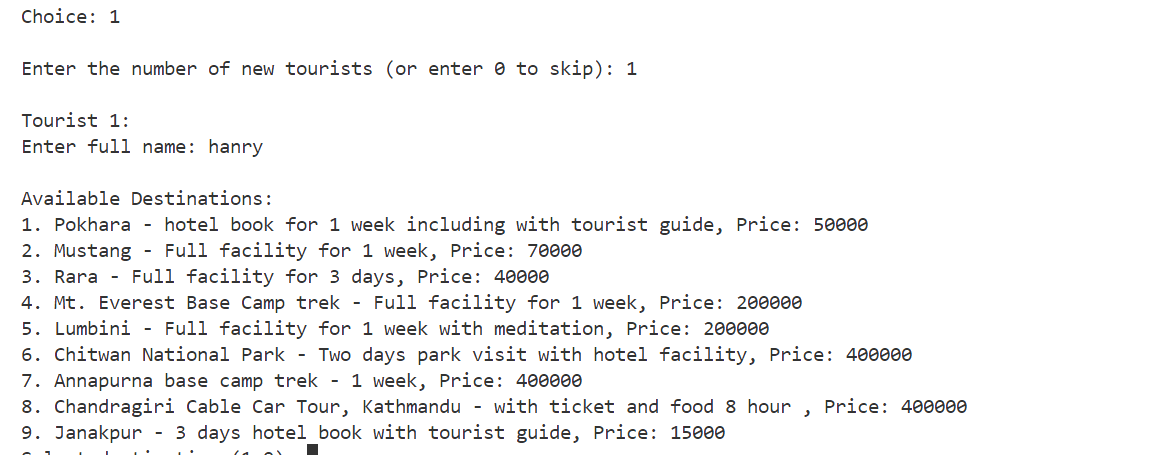


Appendix:6 Add New destination using Key:

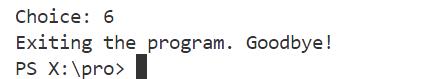
Key: Desti111



Appendix:7 After adding new destination



Appendix:8 Exit the program:



Appendix:9 File view tourist.csv

