1. Aim of the Project

The aim of this project is to create a Tourist Information System that provides detailed information about popular tourist destinations in India. This system allows users to input the name of a destination and receive comprehensive details about attractions, activities, and transportation options available at that location.

2. Business Problem

Tourists often require reliable and detailed information about their destinations to plan their trips effectively. The lack of a centralized and user-friendly platform to access this information can lead to inconvenience and a less enjoyable travel experience. The problem is to develop a system that consolidates tourist information and makes it easily accessible to users.

3. Project Description

The project involves developing a Python-based command-line application that stores and retrieves tourist information for various destinations in India. The system uses a predefined dataset of destinations, including details on attractions, activities, and transportation. Users can interact with the system by entering a destination name to get relevant information, enhancing their travel planning process.

4. Functionality

Add Destination:\*\* Allows the addition of a new destination with its associated information.

Retrieve Information:\*\* Fetches and displays information about a specific destination entered by the user.

List Destinations:\*\* Lists all available destinations to inform users about their options.

User Interaction:\*\* Provides an interactive command-line interface for user input and output.

5. Input Versatility with Error Handling and Exception Handling

Input Handling:\*\* The system prompts the user to enter the name of a destination.

Error Handling:\*\* If the user inputs a destination that is not in the system, the system responds with "Destination not found."

Exception Handling:\*\* The system gracefully handles keyboard interrupts (Ctrl+C) and end-of-file errors (Ctrl+D) to exit the program without crashing.

6. Code Implementation (Description)

The code is structured into two main classes: `Destination` and `TouristInformationSystem`, along with a `main` function to run the application.

Destination` Class

Represents a tourist destination and stores its name, attractions, activities, and transportation options.

`TouristInformationSystem` Class

Manages a collection of `Destination` objects, providing methods to add destinations and retrieve information.

#### `main` Function

- Initializes the `TouristInformationSystem`.

- Populates it with predefined destination data.

- Provides an interactive loop for user input to fetch destination information.

- Handles user exit and unexpected errors gracefully.

Here is the complete code:

```python

class Destination:

def \_\_init\_\_(self, name, attractions=None, activities=None, transportation=None):

self.name = name

self.attractions = attractions or []

self.activities = activities or []

self.transportation = transportation or []

def \_\_str\_\_(self):

return self.name

class TouristInformationSystem:

def \_\_init\_\_(self):

self.destinations = {}

def add\_destination(self, destination):

self.destinations[destination.name] = destination

def get\_destination\_info(self, destination\_name):

destination = self.destinations.get(destination\_name)

if not destination:

return "Destination not found."

info = f"Tourist Information for {destination\_name}:\n"

info += f"Attractions: {', '.join(destination.attractions)}\n"

info += f"Activities: {', '.join(destination.activities)}\n"

info += f"Transportation: {', '.join(destination.transportation)}\n"

return info

def main():

tourist\_info = TouristInformationSystem()

destinations\_data = {

"Agra": {"attractions": ["Taj Mahal", "Agra Fort", "Fatehpur Sikri"],

"activities": ["Sightseeing", "Heritage Walks", "Shopping"],

"transportation": ["Bus", "Train", "Taxi"]},

# Additional destination data...

}

for destination\_name, destination\_data in destinations\_data.items():

destination = Destination(name=destination\_name,

attractions=destination\_data.get("attractions", []),

activities=destination\_data.get("activities", []),

transportation=destination\_data.get("transportation", []))

tourist\_info.add\_destination(destination)

print("Welcome to the Tourist Information System for Popular Destinations in India.")

print("Enter destination name to get tourist information.")

print("Available destinations: " + ", ".join(destinations\_data.keys()))

try:

while True:

destination\_name = input("Enter destination name (or type 'exit' to quit): ")

if destination\_name.lower() == 'exit':

print("Exiting the system.")

break

print(tourist\_info.get\_destination\_info(destination\_name))

except (KeyboardInterrupt, EOFError):

print("\nExiting the system.")

except Exception as e:

print(f"An unexpected error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

```

7. Results and Outcomes

- \*\*User Experience:\*\* Users can easily get detailed information about various tourist destinations in India, including attractions, activities, and transportation.

- \*\*Efficiency:\*\* The system quickly retrieves and displays information based on user input.

- \*\*Error Handling:\*\* The system effectively handles errors and exceptions, ensuring a smooth user experience.

8. Conclusion

The Tourist Information System provides a valuable tool for tourists planning their trips to various destinations in India. By consolidating and presenting essential information in a user-friendly manner, the system enhances the travel planning process, helping users make informed decisions. The project successfully addresses the need for accessible and reliable tourist information.