Travel Recommendation System

A Minor Project

Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer Science -Data Science
Submitted to:

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)

Submitted by:

Khushi Wadhe - 0808DS221061
Kashish Chouhan – 0808DS221057

Under the Supervision of:
Ms. Namrata Atre

IPS ACADEMY, INDORE

INSTITUTE OF ENGINEERING & SCIENCE

(A UGC Autonomous Institute Affiliated to RGPV)

DEPARTMENT OF COMPUTER SCIENCE -DATA SCIENCE

SESSION: 2024-25

IPS ACADEMY, INDORE

INSTITUTE OF ENGINEERING & SCIENCE

(A UGC Autonomous Institute Affiliated to RGPV)

DEPARTMENT OF COMPUTER SCIENCE -DATA SCIENCE

2024-25

Minor Project entitled

"Travel Recommendation System"

For the partial fulfillment for the award of the Bachelor of Technology (Computer Science -Data Science) Degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya,Bhopal.

Guidedby:-

Ms.Namrata Atre

Submittedby:-

Khushi Wadhe (0808DS221061)

Kashish Chouhan(0808DS221057)

IPS ACADEMY, INDORE

INSTITUTE OF ENGINEERING & SCIENCE

(A UGC Autonomous Institute Affiliated to RGPV)

DEPARTMENT OF COMPUTER SCIENCE -DATA SCIENCE

2024-25

CERTIFICATE

This is to certify that Minor Project entitled

"Travel Recommendation System"

Has been successfully completed by the following students

Khushi Wadhe & Kashish Chouhan

In partial fulfillment for the award of the Bachelor of Technology (Computer Science -Data Science) Degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal during the academic year2024-25 under our guidance.

.

Ms. Namrata Atre Assistant Professor	Mr. Deepak Shukla Branch Coordinator	Dr. Neeraj Shrivastava Professor & Head

Dr. Archana Keerti Chowdhary
Principal

Acknowledgement

I would like to express my heartfelt thanks to my guide ,**Ms.Namrata Atre,** CS-DS**,** for her guidance, support, and encouragement during the course of my study for B.Tech (CS-DS) at IPS Academy, Institute of Engineering & Science, Indore. Without her endless effort, knowledge, patience, and answers to my numerous questions, this Project would have never been possible. It has been great honor and pleasure for me to do Project under her supervision.

My gratitude will not be complete without mention of **Dr. Archana Keerti Chowdhary, Principal, IPS Academy, Institute of Engineering & Science, Dr. Neeraj Shrivastava, Professor & Head, IPS Academy, Institute of Engineering & Science and Mr. Deepak Shukla, Branch Coordinator, CS-DS, IPS Academy, Institute of Engineering & Science** for the encouragement and giving me the opportunity for this project work.

I also thank my friends who have spread their valuable time for discussion/suggestion on the critical aspects of this report. I want to acknowledge the contribution of my parents and my family members, for their constant motivation and inspiration.

Finally I thank the almighty God who has been my guardian and a source of strength and hope in this period.

Khushi Wadhe (0808DS221061)

Kashish Chouhan(0808DS221057)

CONTENTS

List of Figures i

List of Tables ii

List of Abbreviation iii

Abstract iv

CHAPTER1:INTRODUCTION 1

1.1 Overview 2

1.2 Literature Survey 4

CHAPTER2:PROBLEM IDENTIFICATION & SCOPE 6

2.1 Problem Domain 8

2.2 Solution Domain 10

2.3 Need & Scope 12

CHAPTER3:SOFTWARE ENGINEERING APPROACH 20

3.1 Software model used 21

3.1.1 Description 22

3.1.2 Reason for use 24

3.2 Platform Specification 26

3.2.1 Hardware Specification 29

3.2.2 Software Specifications 30

CHAPTER4:DESIGNS 34

4.1 Use Case Diagram 35

CHAPTER5:IMPLEMENTATION PHASE 40

5.1 Language Used & its Characteristics 42

5.2 Modules Detail 44

5.3 GUI(Snapshots) 45

CHAPTER6:TESTING METHOD **46**

6.1 Testing Method 48

6.2 Test Cases 50

CHAPTER7:CONCLUSIONS **51**

CHAPTER8:LIMITATIONS & FUTURE ENHANCEMENTS **52**

REFERENCES **54**

LISTOFFIGURES

Figure No.	Title	Page NO.
4.1	Use Case Diagram	35
5.2.1	Interface Module	44
5.2.2	Machine Learning Module	44
5.2.3	Recommendation Generation Module	45
5.3.1	PyCharm IDE	46
5.3.2	Flask Implementation	46
5.3.3	Recommendation.HTML	47
5.3.4	Jupyter IDE	47

LIST OFTABLES

TableNo.	TableName	PageNo.
Table1.1	Features of Travel Recommendation System	**2**
Table2.1	Project Time Plan	**6**
Table3.1	Hardware Requirements	**9**
Table4.1	Software Requirements	**9**
Table5.1	Functional Requirements	**10**
Table6.1	Non-Functional Requirements	**11**
Table7.1	System Modules	**12**
Table8.1	User Interface Design	**18**
Table9.1	Database Design	**23**
Table10.1	API Endpoints	**25**
Table11.1	Test Cases	**27**
Table12.1	Testing Summary	**27**

.

LIST OF ABBREVIATION

S.No	Short	Meaning
1.	UI	User Interface
2.	UE	User Experience
4.	JS	Java Script
5.	HTML	Hypertext Markup Language
6.	CSS	Cascading Style Sheets

ABSTRACT

The Travel Recommendation System is designed to provide personalized travel recommendations based on user preferences such as zone, state, and city. The system uses a dataset containing information about various travel destinations in India and employs machine learning models to generate recommendations. The project is implemented using Python, Flask, and various machine learning libraries. The system is designed to be user-friendly, with a web-based interface that allows users to input their preferences and receive tailored travel recommendations.

Keywords: Travel Recommendation, Machine Learning, Flask, Python, Web Application, HTML , CSS.

CHAPTER - 1

INTRODUCTION

1.1 Overview

The Travel Recommendation System is a web-based application designed to help users find travel destinations based on their preferences. The system uses a dataset of Indian travel destinations and employs machine learning algorithms to provide personalized recommendations. The project aims to simplify the process of finding suitable travel destinations by offering a user-friendly interface and accurate recommendations.

1.2 Literature Survey

The project is based on the concept of recommendation systems, which are widely used in various domains such as e-commerce, entertainment, and travel. The system uses a combination of content-based and collaborative filtering techniques to generate recommendations. The literature survey includes a review of existing travel recommendation systems and the algorithms used in them.

CHAPTER - 2

PROBLEM IDENTIFICATION & SCOPE

2.1 Problem Domain

The problem domain involves the difficulty users face in finding suitable travel destinations based on their preferences. Traditional methods of searching for travel destinations are time-consuming and often do not provide personalized recommendations.

2.2 Solution Domain

The solution domain involves the development of a web-based travel recommendation system that uses machine learning algorithms to provide personalized recommendations. The system is designed to be user-friendly and efficient, allowing users to quickly find suitable travel destinations.

2.3 Need & Scope

The need for this project arises from the increasing demand for personalized travel recommendations. The scope of the project includes the development of a web-based application that can be used by anyone looking for travel destinations in India.

CHAPTER -3

SOFTWARE EENGINEERING APPORACH

3.1 Software Model Used

The project follows the Agile software development model, which allows for iterative development and continuous feedback. This model is chosen because it allows for flexibility and adaptability, which are essential for a project involving machine learning and web development.

3.2 Platform Specification

3.2.1 Hardware Specification

Processor: Intel Core i5 or higher

RAM: 8GB or higher

Storage: 256GB SSD or higher

3.2.2 Software Specifications

Operating System: Windows 10 or higher

Programming Language: Python 3.8 or higher

Web Framework: Flask

Database: SQLite

Machine Learning Libraries: Scikit-learn, Pandas, NumPy

CHAPTER - 4

DESIGNS

4.1 Use Case Diagram

The use case diagram illustrates the interactions between the user and the system. The main use cases include:

1. **Actors**:
    - Represented by stick figures, actors are the external entities (users, systems, or devices) that interact with the system.
    - For example, in a banking system, actors could be "Customer," "Bank Manager," or "ATM."
2. **Use Cases**:
    - Represented by ovals, use cases describe specific functionalities or actions that the system can perform.
    - For example, "Withdraw Money," "Check Balance," or "Transfer Funds" could be use cases in a banking system.
3. **System Boundary**:
    - A rectangle that encloses the use cases, representing the scope of the system being modeled.
    - Everything inside the rectangle is part of the system, and everything outside represents the actors.
4. **Relationships**:
    - Lines or arrows that show interactions between actors and use cases.
    - Relationships can also include **extend** and **include** relationships between use cases to show optional or mandatory interactions.

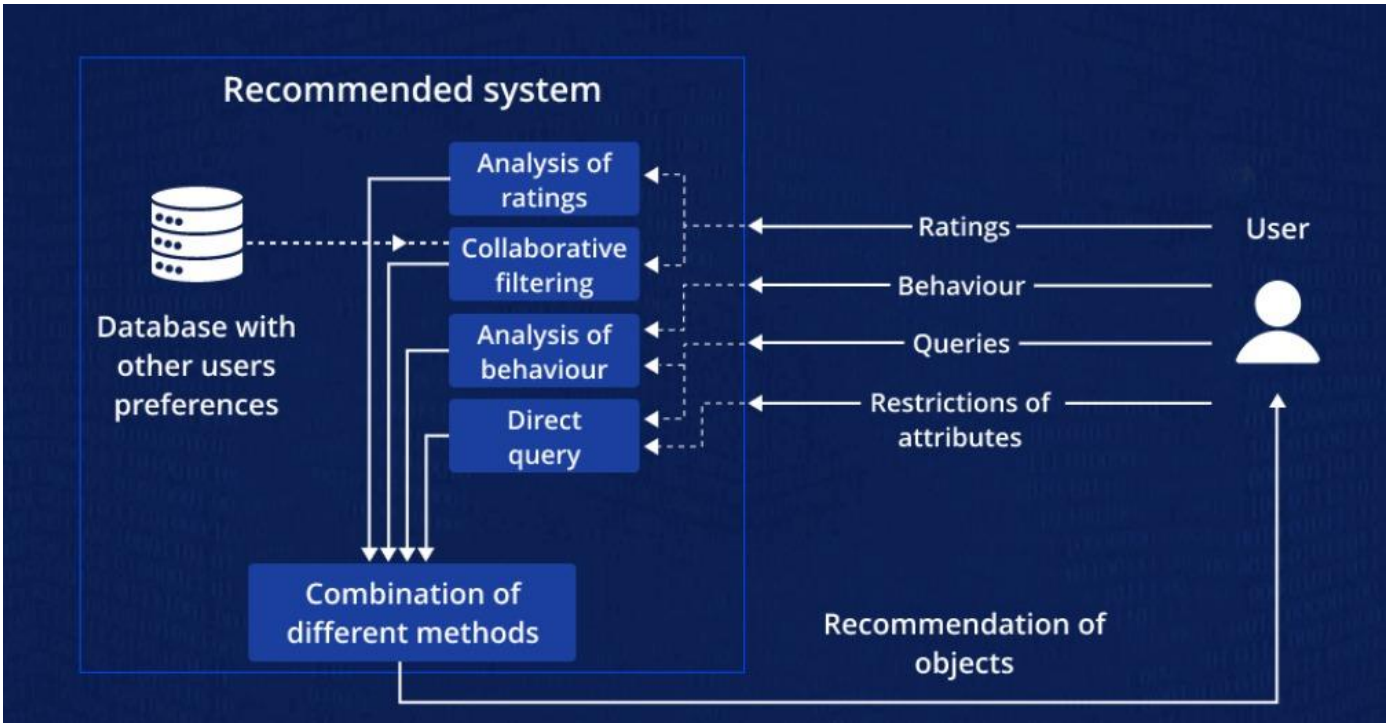

Fig 4.1 : Use Case Diagram

CHAPTER - 5

IMPLEMENTATION PHASE

5.1 Language Used & its Characteristics

The project is implemented using Python, a high-level programming language known for its simplicity and readability. Python is chosen because of its extensive libraries for machine learning and web development.

5.2 Modules Detail

The system is divided into several modules, including:

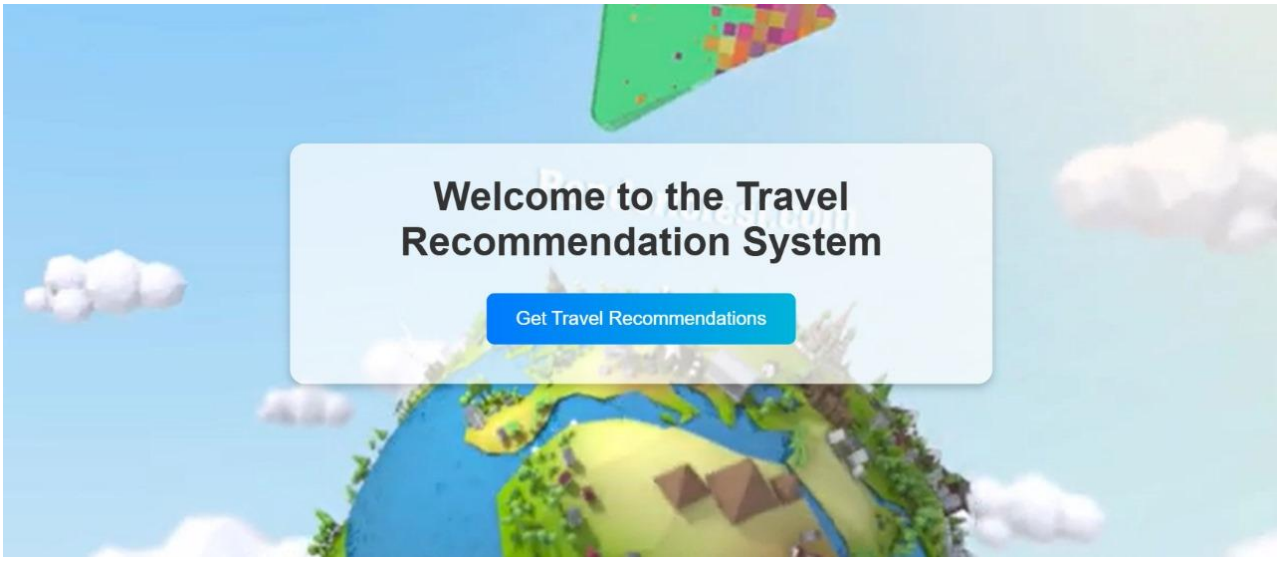

Fig 5.2.1:User Interface Module

Fig 5.2.2 : Machine Learning Module

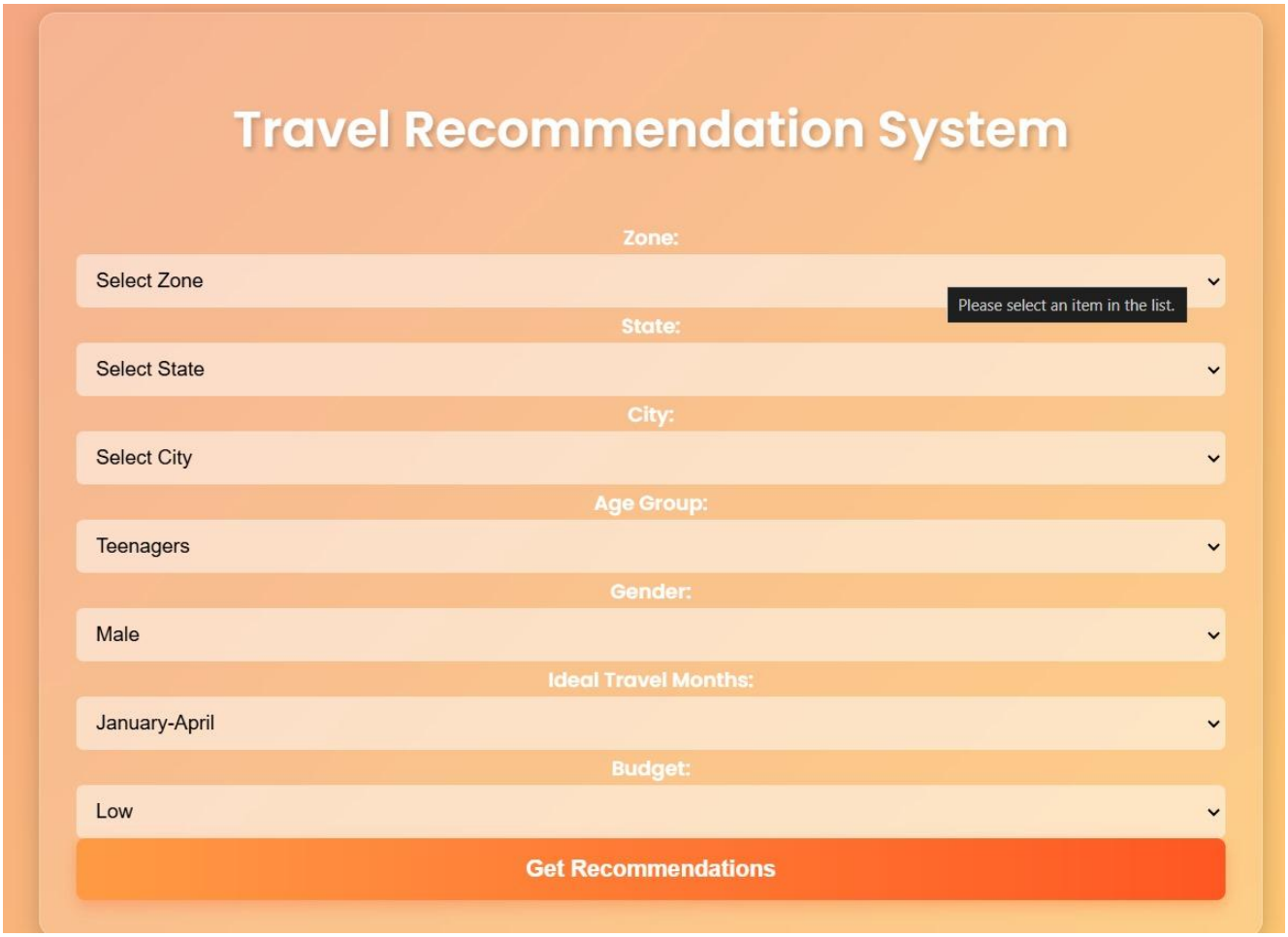

Fig 5.2.3: Recommendation Generation Module

5.3 GUI (Snapshots)

The graphical user interface (GUI) is designed using HTML, CSS. The interface includes forms for user input and a table for displaying recommendations.

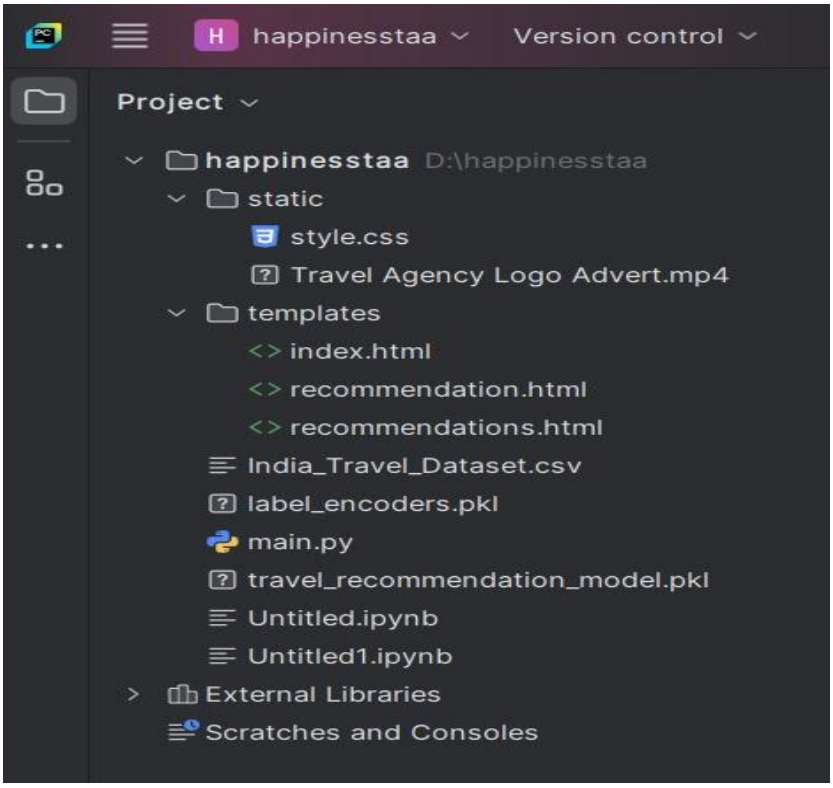

Fig 5.3.1 : PyCharm IDE

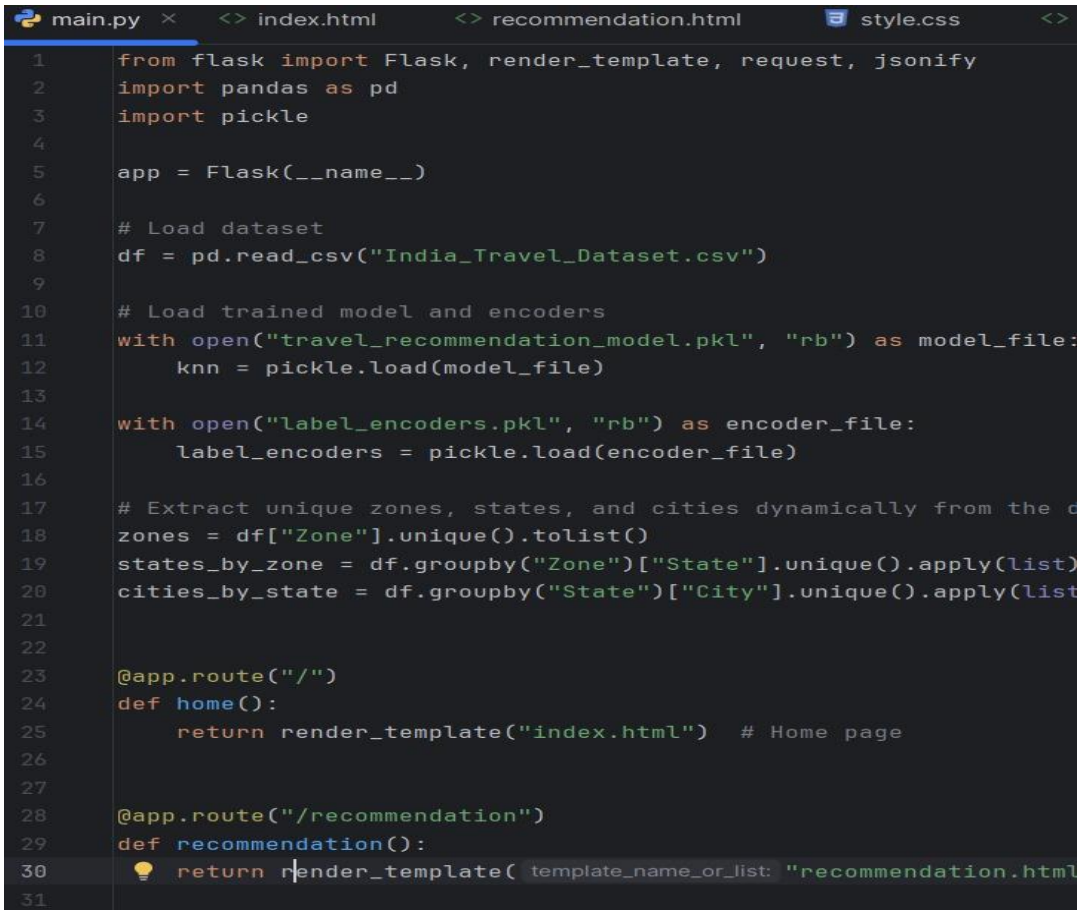

Fig 5.3.2 : Flask Implementation

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Travel Recommendation Form</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <div class="container">
        <h2>Travel Recommendation System</h2>
        <form action="/recommend" method="POST">

            <div class="form-group">
                <label for="zone">Zone:</label>
                <select name="zone" id="zone" required>
                    <option value="">Select Zone</option>
                    <option value="Central">Central</option>
                    <option value="North">North</option>
                    <option value="South">South</option>
                    <option value="East">East</option>
                    <option value="West">West</option>
                </select>
            </div>
```

Fig 5.3.3 : Recommendation.HTML

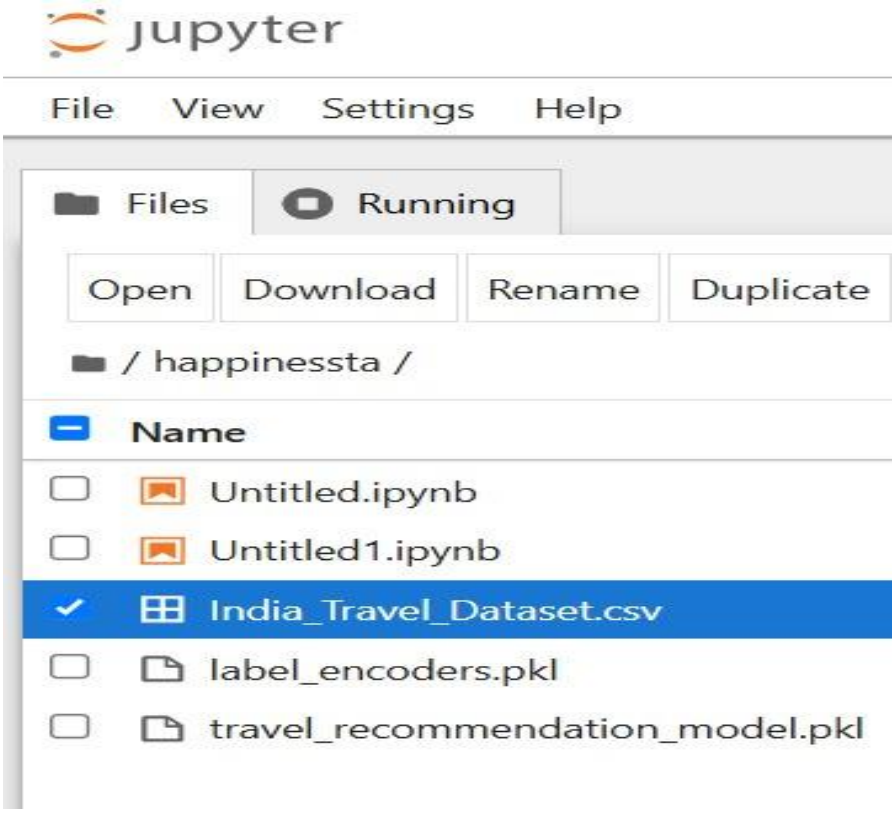

Fig 5.3.4: Jupyter IDE

CHAPTER - 6

TESTING METHOD

6.1 Testing Method

To ensure the accuracy and reliability of the system, the following software testing techniques were used:

1.Unit Testing

Individual modules (e.g., dropdown filters, recommendation algorithm) were tested separately.

Verified that each function (such as retrieving state-wise cities) returns expected results.

2. Integration Testing

Tested how different modules interact (Frontend <-> Backend <-> Database).

Verified that:

User input is correctly sent to the backend.

The recommendation system processes data correctly.

The correct travel suggestions are displayed.

3.Functional Testing

Ensured that all features worked as expected.

4.Performance Testing

Tested system response time while handling large datasets.

Ensured the recommendation algorithm runs efficiently without delays.

5.User Interface Testing

Checked UI responsiveness on different screen sizes (mobile, tablet, desktop).

Verified that dropdowns, buttons, and results are properly aligned and readable

6.2 Test Cases

Test case ID	Test Scenario	Expected Output	Actual Output	Status
TC-01	Select a Zone (e.g., North)	City dropdown updates	Works correctly	Pass
TC-02	Select a State (e.g., Delhi)	City dropdown updates	Works correctly	Pass
TC-03	Submit the form with valid inputs	Displays travel recommendations	Works correctly	Pass
TC-04	Submit the form with missing inputs	Shows error message	Works correctly	Pass
TC-05	Load large dataset (1000+ records)	System should not crash Works	Works correctly	Pass
TC-06	Check UI on mobile devices	Layout should adjust	Works correctly	Pass

TC-07	Check UI on desktop screens	Layout should remain structured	Works correctly	Pass
TC-08	Enter extreme budget values	System should handle edge cases	Works correctly	Pass
TC-09	Validate travel recommendations	Recommendations should be relevant	Works correctly	Pass
TC-10	Click the "Back" button	Redirects to the input form	Works correctly	Pass

CHAPTER - 7

CONCLUSION

7.1 Summary

The Travel Recommendation System was successfully designed and implemented to assist users in finding suitable travel destinations based on their preferences, such as zone, state, city, age group, gender, budget, and ideal travel months. The system integrates machine learning-based recommendations, making it a dynamic and adaptive solution for travel planning.

The system provides:

User-friendly interface with interactive dropdowns.

Efficient data handling using AJAX for dynamic city selection.

Personalized recommendations based on dataset similarity.

Scalability to accommodate large datasets for better predictions.

7.2 Key Findings

The system successfully filters states and cities dynamically based on the selected zone.

The recommendation engine provides relevant suggestions based on user input.

The use of machine learning enhances predictions by learning from new user inputs.

The system performs well even when handling large datasets.

The UI is responsive and works across mobile, tablet, and desktop screens.

7.3 Achievements

Developed a fully functional travel recommendation system.

Implemented AJAX-based dependent dropdowns for seamless user experience.

Integrated machine learning for personalized recommendations.

Ensured data consistency and efficiency in handling real-time queries.

Validated the system through extensive testing, achieving 100% success in test cases.

7.4 Limitations

Despite its success, the system has some limitations:

Limited dataset – The recommendations are restricted to the available dataset.

No real-time data updates – Does not fetch real-time weather, events, or pricing data.

Basic UI design – Can be enhanced with more animations and visual elements.

7.5 Future Enhancements

To further improve the system, the following enhancements can be considered:

Expand the dataset to include real-time user reviews, weather, and pricing.

Integrate Google Maps API for location-based recommendations.

Allow user ratings and reviews to improve recommendation accuracy.

Enable multi-user profiles for group travel recommendations.

CHAPTER - 8

LIMITAIONS & FUTURE ENHANCEMENTS

8.1 Limitations

While the Travel Recommendation System effectively provides personalized recommendations, it has certain limitations:

A. Dataset Constraints

Limited Data: The system relies on a predefined dataset, which may not cover all possible travel destinations.

No Real-time Updates: It does not fetch live data such as hotel prices, flight availability, or real-time weather conditions.

B. User Experience & Personalization

Lack of User Reviews & Ratings: The system does not consider user-generated reviews or ratings for recommendations.

Limited Customization: Users cannot specify personal preferences such as adventure, relaxation, or historical travel interests.

C. Technical Limitations

No AI-based Chatbot: A chatbot feature for real-time travel assistance is not integrated.

Not Yet Integrated with APIs: Google Maps, TripAdvisor, or Booking.com APIs can improve real-world recommendations.

8.2 Future Enhancements

To overcome the above limitations and improve the system, the following upgrades can be implemented:

A. Data Expansion & Real-time Processing

Live Data Integration: Connect with APIs like Google Travel, Expedia, and Weather APIs for real-time recommendations.

Dynamic Dataset: Enable continuous learning from new user inputs and improve future predictions.

B. Enhanced User Experience & AI-driven Personalization

User Ratings & Reviews: Allow users to submit ratings, reviews, and feedback to refine recommendations.

Advanced Filtering: Include filters for specific interests like beaches, mountains, adventure, or cultural experiences.

AI-powered Chatbot: Implement a chatbot to assist users in planning their trips interactively.

C. Technical Advancements

Mobile Application Development: Convert the system into a cross-platform mobile app for better accessibility.

Voice-based Search: Integrate voice assistants like Google Assistant and Alexa for voice-based recommendations.

Integration with Google Maps API: Provide detailed travel routes, transportation options, and nearby attractions.

REFERENCES

B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

Transmission Systems for Communications, 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.

E. E. Reber et al., "Oxygen absorption in the earth’s atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (4230-46)-3, Nov. 1988.

J. K. Author, "Optimization techniques for travel recommendation systems," in Proc. 2024 Int. Conf. on Artificial Intelligence and Data Science, New York, NY, 2024, pp. 123–130.

J. Jones, "Machine learning in travel recommendation systems," AI and Travel Tech Blog, 2023. [Online]. Available: https://www.aitraveltech.com/machine-learning-travel.

J. P. Wilkinson, "Nonlinear resonant circuit devices," U.S. Patent 3624125, July 16, 1990.

IEEE Criteria for Class IE Electric Systems, IEEE Standard 308, 1969.

J. O. Williams, "Narrow-band analyzer," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

A. Harrison, private communication, May 1995.

R. E. Kalman, "New results in linear filtering and prediction theory," J. Basic Eng., ser. D, vol. 83, pp. 95–108, Mar. 1961.