TOUR PLANNING: A TIME-CONSCIOUS END-TO-END TOURIST PLACES RECOMMENDATION SYSTEM

A capstone project report is to be submitted in partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science and Engineering

By

Farhana Jaman 2020-1-60-006	Noshin Tasnim Khan 2020-1-60-221
Ayesha Chowdhury 2020-1-60-003	Fima Faria Lisa 2020-1-60-026

Under the supervision of

Dr. Md. Mostofa Kamal Rasel

Associate Professor

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Declaration:

Project Title TOUR PLANNING: A Time-Conscious End-to-End Tourist Places Recommendation System

Authors Farhana Jaman, Noshin Tasnim Khan, Ayesha Chowdhury, and Fima Faria Lisa

Student ID 2020-1-60-006, 2020-1-60-221, 2020-1-60-003, 2020-1-60-026

Supervisor Dr. Md Mostofa Kamal Rasel

We, the undersigned, hereby declare that the work presented in this capstone project is the outcome of the investigation performed by us under the supervision of Dr. Md Mostofa Kamal Rasel, Associate Professor, Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh. This project has been presented to the department to fulfill the requirements for our Bachelor of Science in Computer Science and Engineering degree. We declare that, this project has not been submitted elsewhere for any other degree or purpose. In addition, we acknowledge that our research received no personal or financial support.

Countersigned

Farhana Jaman (ID: 2020-1-60-006)

Noshin Tasnim Khan (ID: 2020-1-60-221)

Ayesha Chowdhury (ID: 2020-1-60-003)

Fima Faria Lisa (ID: 2020-1-60-026)

Dr. Md Mostofa Kamal Rasel

Dept. of Computer Science and Engineering

East West University

Dhaka, Bangladesh

Letter of Acceptance

This is to certify that the capstone project entitled **"A Time-Conscious End-to-End Tourist Places Recommendation System,"** submitted by Farhana Jaman (ID: 2020-1-60-006), Noshin Tasnim Khan (ID: 2020-1-60-221), Ayesha Chowdhury (ID: 2020-1-60-003), and Fima Faria Lisa (ID: 2020-1-60-026), undergraduate students of the Department of Computer Science and Engineering, has been completed under the supervision of **Dr. Md Mostofa Kamal Rasel**. This project fulfills the requirements for partially completing the Bachelor of Science in Computer Science and Engineering degree. Upon recommendation by the examination committee, we approve this work and certify that no part of this project has been submitted elsewhere for any degree, diploma, or publication.

1. Dr. Md Mostofa Kamal Rasel

Associate Professor

Dept. of Computer Science and Engineering

East West University

Dhaka, Bangladesh

2. Dr. Maheen Islam

Associate Professor

Department of Computer Science and Engineering

East West University

Acknowledgment

In the name of Allah, the Most Merciful, and the Most Compassionate, Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this capstone project. First and foremost, we would like to express our deep and sincere gratitude to our research supervisor, Dr. Md Mostofa Kamal Rasel, for allowing us to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity, and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers at the Department of Computer Science and Engineering at East-West University who taught us during our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Abstract

Tourism today plays an important role in world economies and cultural exchange, with millions of tourists traveling to places for unique experiences that fit their needs and schedules. A dynamic and complete travel recommendation system seems very much essential for this reason. Existing systems usually don’t account for users’ available time or provide a full route for multi-destination trips, resulting in less efficient and enjoyable travel experiences. This study addresses this gap by developing a "Time-Conscious Tourist Places Recommendation System" that optimizes routes based on user preferences, available time, starting location, and desired end destination.

Using data from 13,813 tourist spots across Bangladesh, the system applies several optimization algorithms, including Dijkstra's for shortest path and a Knapsack optimization algorithm, which scored the highest in our tests. Based on its performance, we selected the Knapsack algorithm as the final choice for optimizing travel routes under time constraints.

Results show that the system significantly improves travel planning by suggesting routes that maximize user satisfaction within given time limits. This tool is valuable for travelers and tourism agencies seeking more efficient itinerary planning. By focusing on time-aware, end-to-end route planning, this project fills a major gap in current systems, providing a new and practical solution for organized travel.

Table of Contents

List of Figures

1. Introduction

The section outlines establishing the importance of tourism in the global economy and the need for an advanced travel recommendation system that tailors itineraries to travelers' preferences and available time. By addressing the limitations of traditional sources, it highlights the potential for a system that combines personalized recommendations with efficient route optimization for enhanced user experience through structured and time-efficient travel planning.

1.1 Background

Tourism today plays an important role in world economies and cultural exchange, with millions of tourists traveling to places for unique experiences that fit their needs and schedules. Traditionally, they depend on many sources like guidebooks, word-of-mouth recommendations or online search engines, and travel agencies, but those sources often fail to deliver proper suggestions according to travellers' preferences and are also very time-consuming.

To resolve these challenges, a more dynamic and complete travel recommendation system seems very much essential. Unlike existing solutions that focus primarily on individual destinations without considering the traveler's available time, a new approach is needed that optimizes the sequence of multiple places within a set timeframe. Such a system would empower travelers by providing a smooth travel experience that includes not only personalized recommendations of places to visit but efficient route development in a sequence that best utilizes their time and enjoyment.

A modern recommendation system should incorporate key data points such as the traveler’s starting location, preferred endpoint destination, and total time available for exploration. Such input would enable the system to provide a well-rounded itinerary that ensures the traveler visits the most suitable places in an optimal order. These will enable the automatic generation of an appropriate itinerary which ensures he visits the best places in the most efficient order. It involves factors like distances between sights, estimated time at each stop, mode of travel, and other real-time constraining factors. The combination of each of these factors leads to a custom-fitted trip for the benefit of the traveler.

1.2 Research Questions

This section outlines the primary research questions that will guide the investigation into creating a travel recommendation system capable of providing personalized, multi-destination itineraries using the available time of the user.

- What are the main factors that influence travelers' choices when selecting multiple destinations within a limited timeframe?

- How can a travel recommendation system be designed to balance personalization with efficiency in itinerary planning?
- What types of data (for example: user preferences, current location, time constraints) are essential for creating a well-rounded and practical itinerary?
- How do existing travel recommendation systems fall short in providing sequential, multi-destination routes, and what gaps need to be addressed?
- What algorithms or machine learning models are most effective for generating optimized travel routes based on user input?
- How does real-time data integration (for example: such as traffic, weather, and attraction opening hours) impact the usability and reliability of travel itineraries?
- What measures can be implemented to ensure the recommendations are not only time-efficient but also enjoyable and aligned with user interests?
- How can user feedback be continuously collected and used to improve the recommendation system's accuracy and personalization over time?
- What are the potential challenges in collecting and managing user data while maintaining user privacy and data security?
- How can a travel recommendation system handle unexpected changes (for example: sudden closures or delays) to adapt and reconfigure the suggested itinerary on the go?

1.3 Research Objective

Route Optimization: One of the goals of this Capstone Project is to develop an effective travel recommendation system that might potentially optimize routes for multi-destination tours. This system focuses on generating itineraries that maximize user enjoyment while efficiently using the available time.

User-Centric Experience: The other goals of the Capstone Project include providing user-centered and personalized recommendations on the basis of the traveler's starting and preferred endpoint, along with the traveler's personal preferences. The system aims to deliver an engaging experience by incorporating user feedback and dynamically updating itineraries as needed.

Comprehensive Data Integration: The project emphasizes the integration of key data points, such as distances between locations, estimated travel and visit times, and mode of transportation. This objective ensures the generated itineraries are comprehensive and reliable.

Sustainability and Efficiency: This research looks toward a greener system by improving traveling routes to reduce fuel consumption and travel time. It shall enable sustainability by supporting smarter and more environmentally responsible travel.

1.4 Focus and Contribution

This project develops a Flask-based web application for personal travel recommendations. Users can input their current location, the end location, and the maximum travel time through an HTML form. From this input, advanced algorithms will develop the routes for travel that are appropriate and destinations to eventually give a highly efficient and user-friendly experience.

Our goal is to develop a friendly travel recommendation system, which can suggest rich travel for various preferences by using route planning and distance calculation algorithms for the optimization of user travel experiences.

Key contributions of this project include:

- **User Input Processing:** Created an interactive form for inputting the user's current location, desired end location, and maximum travel time in an easy way for the facilitation of travel recommendations.
- **Route Planning Algorithms:** Creation and integration of various algorithms that are helpful in calculating the optimal route of travel recommendations based on user input by considering distance and travel time.
- **Modern Technologies Integration:** Used Flask for responsive web development to integrate mapping APIs and increase efficiency and accuracy in travel route recommendations provided.
- **User-Friendly Interface:** Created a clean and intuitive web interface. This would make everything a lot easier for the user and would enable him or her to instantly get personalized options for travel routes and destinations.
- **Contextual Recommendations:** Moved the recommendation engine to user-defined, context-aware inputs such as personal travel preferences or route popularity, granting meaningful travel recommendations.
- **Performance Optimization:** Assessed and optimized the performance of the system to ensure fast responsiveness and relevance of recommendations for overall improvement of user satisfaction and engagement.

The project will demonstrate how web technologies coupled with algorithmic approaches develop a dynamic travel recommendation system. Emphasizing user inputs of the current location, end location, and maximum travel time, it is definitely of utmost benefit to travelers to produce an optimized personalized itinerary and an enriched traveling experience.

2. Background and Related Work

This section reviews methodologies and existing literature on travel recommendation systems, pointing out the key limitations of the existing approaches, such as a lack of multi-destination sequencing, taking into consideration an endpoint. Summaries of related studies are provided, highlighting the algorithms used and limitations in real-world applicability. Furthermore, this section outlines dataset challenges, such as inconsistencies in language, duplication of place names, and missing ratings, as well as API limitations impacting data completeness and analyses.

2.1 Existing Methodologies

Traditional travel recommendation systems do not take into consideration the number of places travelers can actually visit because of their time constraints. Hence, a system is required that will help in suggesting optimum sequential routes based on a user's starting point, endpoint, and available time to travel. Many of the current methodologies fail to incorporate the end-point in their suggestions, which leads to less fulfilling travel experiences. Moreover, no current system provides detailed travel routes and sequences for visiting various places within a specified timeframe.

2.2 Literature Review:

Prof. Shrikant et al. **[5]** proposed a personalized recommendation system that recommends places based on user interest, current location, and range of distance. It uses content-based filters to find similar interests among user profiles. Also, calculate Euclidean distance and use the KNN algorithm to recommend the places. Here, the apriori algorithm is used for classification. This system also filters out visited places, providing information like reviews and seasonal recommendations.

Sutomo, R., & Kaisha Pratama, D. **[2]** present an innovative approach to enhance tourism experiences in the City of Semarang, Indonesia. They develop an advanced recommendation system that integrates content-based and collaborative filtering techniques to offer personalized tourist suggestions. The paper also demonstrates the system's efficacy in simplifying travel decisions and uncovering Semarang's hidden treasures. Overall, it fills a research gap and also contributes to the field of tourism recommendation systems.

Mishal et al.**[3]** introduced a personalized and budget-friendly tour spot recommendation system based on content-based filtering, which helps to enhance the tourist experience by recommending the best picnic spots according to all user’s preferences, including budgets, interests, destination country, type of places, safety, transportation, and climate. This system compares the features of different spots and takes into account the user's previous history to recommend similar tour spots by using content-based filtering methods to predict the user's preferences accurately and suggest the best picnic spots.

Yanmei Zhang et. al.[4] proposed a route planning method that gives travel routes with multiple places considering multiple factors (that is, the distance between sites, initial travel position, initial departure time, time duration of the tour, total cost, scores, and popularities of sites) comprehensively, and a comprehensive attractiveness index rated routes. They used simulated annealing, greedy algorithm, and genetic algorithm where the genetic algorithm performed best on run time.

Title	Activity	Algorithm	Limitations
Traveler's Recommendation System Using Data Mining[1]	Recommend considering user's interest, current location, and range of distance. Filter out visited places.	Content-Based Filtering, KNN Algorithm and Apriori algorithm	1. Uses Euclidean distance, which may not accurately reflect real-world distances. 2. Does not account for travel time constraints. 3. Recommends only one place from the suggested spots 4. Does not provide travel routes or sequences for visiting multiple places within a specific timeframe.
Measuring Tourist Experience in Semarang City through an Advanced Recommendation System[2]	Enhances tourism in Semarang, Indonesia, by simplifying travel decisions, uncovering hidden treasures, and highlighting top-rated destinations and local favorites based on user ratings.	content-based and collaborative filtering	1. Does not account for travel time constraints. 2. Recommend only one place to visit from their list of suggested spots. 3. Does not provide travel routes and sequences for visiting multiple places within a specific time.

Tour Spot Recommendation System via Content-Based Filtering[3]	Enhance the tourist experience by recommending the best picnic spots according to all users preferences, including budgets, interests, destination country, type of places, safety, transportation and climate.	Content-Based Filtering	1. Does not consider traveler’s travel time constraints. 2. Does not consider current location, travel routes, and sequences for visiting multiple places.
A Tourism Route-Planning Approach Based on Comprehensive Attractiveness[4]	Recommends travel route with multiple places considering multiple factors including the distance between sites, initial travel position, initial departure time, time duration of tour, total cost, scores, and popularities of sites.	Genetic Algorithm	1. Conducts on a small dataset. If the dataset becomes larger, then time complexity increases exponentially which affects the real-time recommendation.

Table 1: Related work

Note: “None of those papers considered the endpoint while recommending the tour plans”

3. EXPERIMENTAL DESIGN, MATERIALS AND METHODS

This section discusses dataset details and its preparation, methods and methodology, E-R model ,user interface design, experimental setup and business model of this system.

3.1 Dataset Details

Tourist attraction data is information and statistics about the popular destinations and landmarks that attract tourists. This data contains information about visitor numbers, demographics, visitor satisfaction, popular landmarks, museums, national parks, historical sites, amusement parks, and economic impact. This section describes the details of the dataset and data representation that hold the research information as well as the structures of the files.

Dataset Details

- There are 495 sub-districts or upazilas in Bangladesh. The information regarding the upazilas was collected from the official outlet of the Bangladesh government, which is responsible for providing any official administrative information. We collected data for tourism using Google Place API for each upazila based on three keywords: points of interest, historical places, and tourist places. We collected information on about ten places for each keyword in each upazila.
- The collected data was JSON-encoded. Each record contains features like title, which is basically the place's name, upazila name, longitude, latitude, rating, rating count, category, id, and keyword. However, the raw dataset was not in proper format with suitable information. After processing all the records by data integration, validation, feature extraction, outlier removal, and prepossessing, we finally have 13,813 records about tourist spots across Bangladesh. To increase the portability and usability on different systems, the entire dataset was also converted to the format of JSON and SQL format. The size of the SQL file is 1.98 MB and the JSON file is 5.00 MB. The feature details of the dataset are provided in Table 2.
- A python code has been written to refine the dataset. It checks if the collected raw data is in the expected JSON format. If there are missing values in the data, the code fills them with appropriate default values. We also detect and remove the outliers, which geolocation is incorrectly assigned outside the territory of Bangladesh. We have extracted useful features and removed non-essential features. After processing, the data is transformed into a suitable format to store in the database table and we have got our final SQL and JSON file. This dataset contains essential features for recommender systems having data in a mix of English and Bangla.
- We observe that a combined dataset focusing on tourist places in each upazila of Bangladesh has not been publicly available yet. Therefore, this dataset will be able to serve researchers, business analysts, and the government in the field of tourism in Bangladesh.

It can give people valuable recommendations about tourist places, plan for the tours, and navigate the interesting places within the given time limit. By developing a freely available and standardised dataset, it can support developments in tourism.

Attribute	**Description**
ID	A unique identifier for each tourist location starting from 1.
Upazila Name	An administrative division in Bangladesh, functioning as a subunit of a district. Tourist spots are located in this place.
Search Key	Keywords were used to search for each attraction for search optimization.
Title	The name of the tourist attraction or place.
Address	The specific location of the tourist attraction.
Latitude	Geographic latitude indicates the exact location.
Longitude	Geographic longitude indicates the exact location.
Category	Type of tourist spot (e.g., park, historical landmark, beach).
Rating	Average user rating of the attraction on a scale of 1 to 5. The default value of this feature is -9 which shows the missing value.
Rating Count	Number of user reviews or ratings recorded for each place.

Table 2: Details of Attributes in TourismDatasetBD dataset

This dataset is suitable for developing personalized tourism recommendation systems by using features of user preferences including category, rating, rating count, location, such as latitude and longitude, and other relevant features. Various models of machine learning can be applied to develop this system, such as the application of content-based filtering, using features such as category and title for similar attraction suggestions. Geo-location-based recommendations can utilize features such as latitude and longitude to recommend spots that are near a location. Clustering will group tourist spots using features such as category and rating for targeted recommendations. Hereby, calculating the distance between various locations will determine how close one place is to another, which can be used to help the recommendations get sorted by geographical proximity. Finally, integration of all of these methods can make the recommendation system provide personalized suggestions for the current location of a tourist or his specified preferences. It might give tourists recommendations for popular tourists depending on the location, that are nearby or within a defined radius of their location, which will increase the tourism experience much better.

Number of records	13,813
How data are acquired	By accessing the official website of the Bangladesh Government and Google Place API.
Data format	JSON, SQL
Search Keys	Point of interest, Historical places, Tourist places.
Categories	Tourist attractions, parks, gardens, historically important mosques, temples, pagodas, monuments, bridges, etc.
Number of Features	10 (ID, search key, place name, title, address, latitude, longitude, category rating, ratingCount).
Data source locations	Official website of Bangladesh Government and Google Place API.
Where applicable	Suitable for developing personalized tourism recommendation systems using machine learning models like content-based filtering, clustering, geolocation, and others.

Table 3: TourismDatasetBD dataset

3.2 Dataset Collection and Preparation

Multiple steps were conducted to collect and process the tourism dataset of Bangladesh for research purposes in the smart tourism industry to ensure consistency, accuracy, and usability. Figure 1 shows the methodology for preparing the dataset. The steps in preparing the dataset are given below:

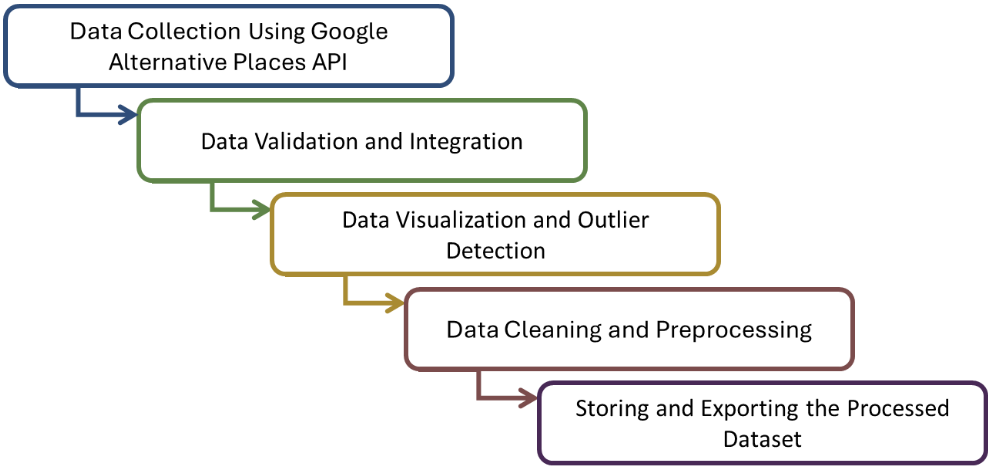

Data Collection Using Google Alternative Places API
Data Validation and Integration
Data Visualization and Outlier Detection
Data Cleaning and Preprocessing
Storing and Exporting the Processed Dataset

Figure 1 Methodology of TourismDatasetBD Dataset Preparation

3.2.1 Data Collection Using Google Alternative Places API

Google Alternative Places API website was used to extract data containing valuable information including title, address, longitude, latitude, category, rating and rating count. Three Keywords: Point of Interest, historical places, and tourist places, were used as searched keywords with upazila name (e.g. "point of interest at Atpara"). The list of all upazilas of Bangladesh has been collected from the Bangladesh National Information Website. The website had given a maximum of 10 places and a minimum of 0 places in JSON format for each query. The provided information had been saved in an individual file for each upazila.

3.2.2 Data Validation and Integration

A Python script was written which checks if the file contains correct JSON formatted data. If it was not in the correct format, then printed an error message and showed the files that had the issue. In this way, valid JSON format in files is ensured. All the files get merged into one JSON file and two additional fields named search_key and upazila_name were added to each record.

3.2.3 Data Visualization and Outlier Detection

A box plot was created to identify outliers by calculating the upper and lower limits using the data of latitude and longitude. This method helped in identifying and removing outliers based on limits. After outlier removal using the boxplot, latitude and longitude data points were plotted on a graph which shows a figure similar to the map of Bangladesh. In the graph, most of the data points were located within Bangladesh, but some points were found outside the expected boundaries, so some outliers were still present. From visualization, we expected the outliers can be placed in India as the data points were placed in the location of India according to the map. To further refine the data, records containing "India" as a substring in the address feature were investigated. India as a substring in the value of address feature of some records was found and those records got removed from the dataset to ensure the dataset only has Bangladesh-specific data.

3.2.4 Data Cleaning and Preprocessing

Outliers detected in the boxplot were removed based on the calculated upper and lower limits using the values of latitude and longitude. In addition, geographical visualisation showed some records are outside of Bangladesh's map, and they also got removed by address-based cleaning. Here, the feature named address, having "India" as a substring in their value was removed to have only Bangladesh-specific data. Besides, some records contain missing values in two features: rating and rating count. Default values were assigned wherever data was missing in order to standardize the dataset. The value 0 was given as the default for rating, and rating count, meaning this record does not have any rating or rating count value. In addition, a unique id was given to each record for the sake of identification and traceability. Also, features were rearranged to be consistent throughout all records, and non-essential features were removed. So, the final processed JSON formatted tourism dataset was ready for suitable use with vital information.

3.2.5 Storing and Exporting the Processed Data

The processed JSON data was uploaded to a MySQL database called tourism_db. A Python script connected to the database, created the tourismdatasetbd table if it did not already exist. The table included features such as id, upazila_name, search_key, title, address, latitude, longitude, category, rating, and ratingCount. After uploading, tourismdatasetbd.sql file got exported from the database as SQL tourism dataset.

3.2.6 Data Collection Materials

Data Collection Tool: Google Alternative Places API website.

Geographical Data Source: Bangladesh National Information Website for the complete list of upazilas.

Development Environment: Python code executed in PyCharm.

Database System: MySQL for data storage and management.

Data Storage Formats: JSON and SQL files.

3.3 Data Representation

This dataset is provided in both JSON and SQL format. The two forms of representation ensure that the dataset is flexible enough to allow for various uses and applications. The structure for both is given below.

3.3.1 JSON Data Representation:

The JSON file represents data as a structured array of objects, with each object containing features related to a tourist location. Figure 2 given below shows JSON data representation.

```
[
    {
        "id": "1",
        "upazila_name": "Atghoria",
        "search_key": "tourist places",
        "title": "শতবর্ষের বট গাছ",
        "address": "396G+X65",
        "latitude": "24.0624",
        "longitude": "89.3756",
        "category": "Tourist attraction",
        "rating": "3.8",
        "ratingCount": "12"
    },
    {
        "id": "15",
        "upazila_name": "Bheramara",
        "search_key": "tourist places",
        "title": "হিসনা খেঁয়াঘাট",
        "address": "2X9J+H84, Bheramara - Daulatpur Rd",
        "latitude": "24.0189",
        "longitude": "88.9808",
        "category": "National park",
        "rating": "0",
        "ratingCount": "0"
    }
]
```

Figure 2 JSON Data Representation

3.3.2 SQL Data Representation:

The information about the attribute and data type of the SQL file is provided below in Table 3 and Example 2 shows SQL data representation.

Database Name: tourism_db.

Table Name: tourismdatasetbd.

Attribute	Data Type
ID	INT
Upazila Name	VARCHAR
Search Key	VARCHAR
Title	VARCHAR
Address	VARCHAR
Latitude	DECIMAL
Longitude	DECIMAL
Category	VARCHAR
Rating	FLOAT
Rating Count	INT

Table 3: Attribute and Data type of tourist_places table

```
INSERT INTO `tourismdatasetbd` (`id`, `upazila_name`, `search_key`, `title`, `address`,
`latitude`, `longitude`, `category`, `rating`, `ratingCount`) VALUES
(1, 'Atghoria', 'tourist places', 'শতবর্ষের বট গাছ', '396G+X65', 24.0624, 89.3756, 'Tourist
attraction', 3.8, 12),
(2, 'Atghoria', 'tourist places', 'দাদার ঘাট', 'Bangladesh', 24.1132, 89.3839, 'Tourist attraction', 5,
2);
```

Figure 3 SQL Data Representation

The dataset is provided in both JSON and SQL files. The JSON data is used as a flexible data format for storing and managing semi-structured data in NoSQL databases like MongoDB and CouchDB. It is well-suitable for web APIs and applications requiring dynamic, hierarchical data structures and big data, while the SQL data format provides a robust solution for integrating and managing the dataset in relational database systems. The lightweight nature of JSON is perfect for exchanging data in web applications of tourism and geolocation-based recommendation systems and machine learning models. Geographical Distribution of tourist places in every upazila of Bangladesh is shown in figure 4.

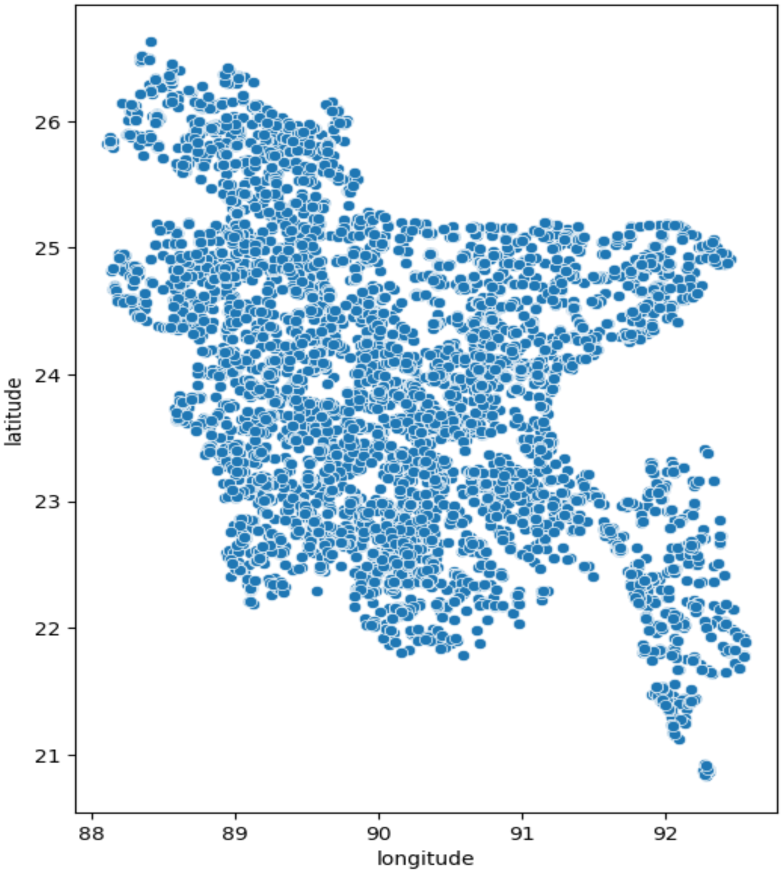


Figure 4 Geographical Distribution of tourist places in every upazila of Bangladesh.

3.4 Methods

In this study, we tested four different optimization algorithms and two distance calculating maths to find the best travel route between a starting point (Damudya Upazila Pond) and an endpoint (Dhanbari Nawab Palace). The goal was to choose routes that met the user's travel time limit, while balancing factors like ratings, travel time, and fuel cost.

3.4.1 Optimization Algorithms

Each algorithm had a unique approach to scoring and selecting routes. We are describing the algorithms below.

Knapsack with Dynamic Optimization:

- **Purpose**: This method aimed to find the highest-scoring route by choosing stops based on their ratings and minimizing fuel costs, all within a set travel time limit.
- **Process**: We used a method similar to the "knapsack problem." Imagine you have a bag with limited space (in this case, the travel time limit) and you want to fit in the most valuable items (or, here, the best-rated stops). The algorithm looked at different route segments and picked the combination that provided the best overall score without exceeding the time limit.
- **Outcome**: This method gave us high-quality routes that scored well but took a bit longer to compute compared to other methods.

Multiple Criteria Decision Making (MCDM):

- **Purpose**: This approach was designed to balance multiple factors — travel time, ratings, and fuel consumption — instead of focusing on just one.
- **Process**: MCDM assigned a score to each route by taking all the factors into account equally. It didn't prioritize one factor over another, meaning it looked for routes that had a good mix of high ratings, reasonable travel times, and low fuel costs.
- **Outcome**: This method was faster than the knapsack approach and produced routes that were well-balanced across different factors.

Knapsack with Traffic Optimization:

- **Purpose**: This version of the knapsack method also considered changes in traffic, which can impact travel time and fuel usage.
- **Process**: The algorithm adjusted travel speeds based on simulated traffic conditions. For example, in high-traffic situations, the algorithm slowed down the travel speed, increasing the expected travel time and fuel consumption. Then, it selected the best route by applying the knapsack method with these traffic-adjusted times and costs.

- **Outcome**: This method was effective for planning routes where traffic conditions could change. It was also faster than the regular knapsack method because it focused on real-world factors like traffic, making it practical for real-time route planning.

Simulated Annealing:

- **Purpose**: This method used a probabilistic approach to explore a variety of routes without checking every single option, making it faster but still effective.
- **Process**: Simulated Annealing started by exploring routes with some randomness, allowing it to consider routes that might not seem ideal at first. Over time, it reduced this randomness (a process known as "cooling") to focus on the better-performing routes. This helped avoid getting stuck on one route too early and allowed the algorithm to keep exploring until it found a good solution.
- **Outcome**: This algorithm quickly found a near-optimal route by exploring different combinations. It might not find the absolute best route, but it’s useful when speed is more important than perfect accuracy.

3.4.2 Distance Calculation Methods

In this tourism route optimization recommendation system, choosing the appropriate distance measurement method is crucial for achieving accurate route planning. Two commonly used methods are the Haversine distance and the Euclidean distance, each with its own advantages and limitations. Here, we compare these two methods to determine which is best suited for this system.

Haversine Distance

The Haversine formula calculates the shortest distance between two points on the Earth's surface, taking the Earth's curvature into account. It provides an "as-the-crow-flies" distance, using latitude and longitude coordinates, which makes it ideal for geographic applications where the curvature of the Earth affects accuracy.

Benefits:

- **High Accuracy**: Haversine is particularly accurate over longer distances, such as those spanning cities or regions. This makes it suitable for a system that needs realistic distance measurements for travel planning.
- **Relevance for Travel Planning**: As it closely reflects actual geographic separation, Haversine distance aligns well with the goals of estimating travel time and fuel costs.
- **Practicality for Various Routes**: The Haversine calculation is reliable even for routes that are not straight lines, offering a closer estimate of real-world travel distances.

Drawbacks:

- **Higher Computational Load**: The Haversine formula involves trigonometric calculations, which are computationally intensive. For very large datasets, this can lead to longer processing times compared to simpler methods.

Euclidean Distance

The Euclidean formula calculates the straight-line distance between two points in a flat, two-dimensional plane. It assumes a flat surface and is based on the Pythagorean theorem, making it a simpler and faster calculation than the Haversine formula.

Benefits:

- **Simplicity and Speed**: Euclidean distance is quicker to compute because it uses straightforward mathematical operations. This could be advantageous if the system needs to process a large number of points rapidly.
- **Suitable for Small, Flat Areas**: Euclidean distance is sufficiently accurate for smaller, localized areas where the curvature of the Earth has minimal impact.

Drawbacks:

- **Lower Accuracy Over Large Distances**: Euclidean distance tends to underestimate distances over large areas, as it doesn't account for the Earth's curvature. This can lead to errors in travel time and fuel estimates in a real-world travel scenario.
- **Limited Practical Use in Real-World Geography**: The assumption of a flat plane reduces its accuracy and practicality for travel planning over larger, varied geographic areas.

A comparison table between those distance calculation methods is given below in table 4.

Aspect	**Haversine Distance**	**Euclidean Distance**
Accuracy	Highly accurate for long distances.	Less accurate, especially over long distances.
Computational Load	Higher due to trigonometric calculations.	Lower due to simpler math operations.
Use Case Suitability	Best for real-world travel over large areas.	Works for small, flat areas; less suitable for large geographic areas.
Real-World Relevance	Accounts for Earth's curvature.	Ignores curvature, assumes flat plane.

Alignment with System Goals	High (due to realistic distance estimates).	Moderate to Low, depending on area size.

Table 4: Comparison table between haversine and euclidean distance calculation method.

In this system, where accurate route planning between distant points is essential, **Haversine distance is generally the better choice**. It aligns with the real-world geographic conditions, leading to more accurate calculations of travel time and fuel costs. Although it’s slightly more computationally intensive, this is outweighed by the benefits of accuracy, especially for an application involving travel routes over potentially large geographic areas.

3.5 Methodology

This study presents a method for creating a tourism recommendation system that finds the best travel route between two locations. The system uses database management, distance calculations, graph building, and optimization algorithms to find the most efficient path within a set time limit. The process is divided into two main stages: **Graph Construction** and **Recommendation**, each with smaller steps as shown in the block diagram.

3.5.1 Graph Construction

The first phase involves setting up the database, preparing data, and building a connected graph of possible routes between locations. This graph is the foundation for finding and optimizing travel paths. The working flow of graph construction is shown in figure 5.

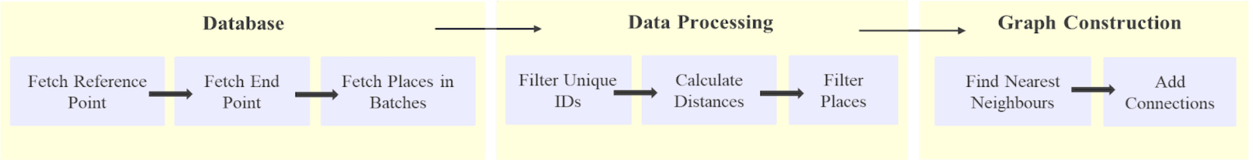


Figure 5 Graph Construction

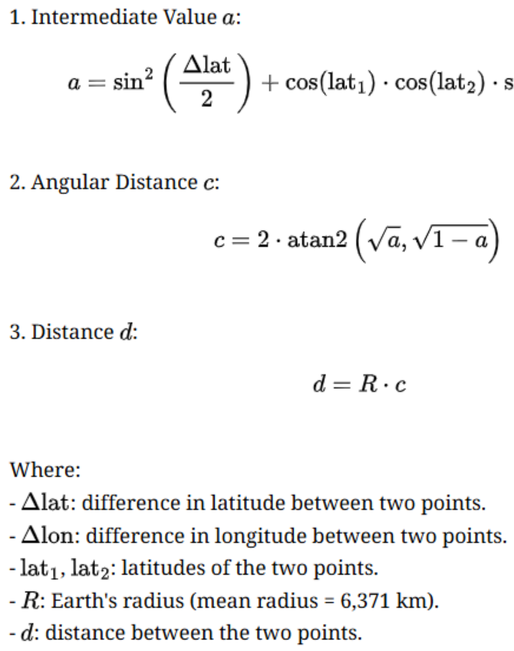
Database:

- **Fetch Reference Point**: The starting point (reference location) is retrieved from the database based on user input. This is done through a SQL query that finds the location matching the user’s input.
- **Fetch End Point**: The destination (end location) is also fetched from the database based on user input, using a similar SQL query.

- **Fetch Places in Batches**: To manage large amounts of data efficiently, locations (places of interest) are retrieved in batches from the database. This is done using the process_in_batches function, which collects data in small portions, helping to save memory and improve speed. Each location has details like latitude, longitude, rating, and name.

Data Processing:

- **Filter Unique IDs**: After retrieving the data, the system removes duplicate locations to ensure each entry is unique. The filter_unique_ids function keeps only one entry for each unique title.
- **Calculate Distances (Haversine)**: The Haversine formula calculates the distance between pairs of locations, taking the Earth's curvature into account. This formula gives accurate "as-the-crow-flies" distances, which are used to estimate travel time. The formula is given below:



1. Intermediate Value a:

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2\left(\frac{\Delta\text{lon}}{2}\right)$$

2. Angular Distance c:

$$c = 2 \cdot \text{atan2}\left(\sqrt{a}, \sqrt{1-a}\right)$$

3. Distance d:

$$d = R \cdot c$$

Where:
- Δlat: difference in latitude between two points.
- Δlon: difference in longitude between two points.
- $\text{lat}_1, \text{lat}_2$: latitudes of the two points.
- R: Earth's radius (mean radius = 6,371 km).
- d: distance between the two points.

- **Filter Places**: Once distances are calculated, locations are filtered based on how close they are to the direct route between the start and endpoint. The get_filtered_places function selects only locations within a certain distance, which helps focus on relevant locations and improve efficiency.

Graph Construction:

- **Find Nearest Neighbors**: Using the filtered list of locations, the system identifies nearby locations (nodes) within a set distance. The fixed_distance_dijkstra function finds and connects nodes that are close enough to each other, creating potential travel segments.

- **Add Connections**: The system then builds a graph by linking these neighboring nodes. The create_graph_from_connections function creates an undirected graph where each location is a node, and each connection (route) is a weighted edge based on distance. This connected graph is used for finding paths and optimizing routes in the next phase. A manually created graph for visualization is showed in figure 6.

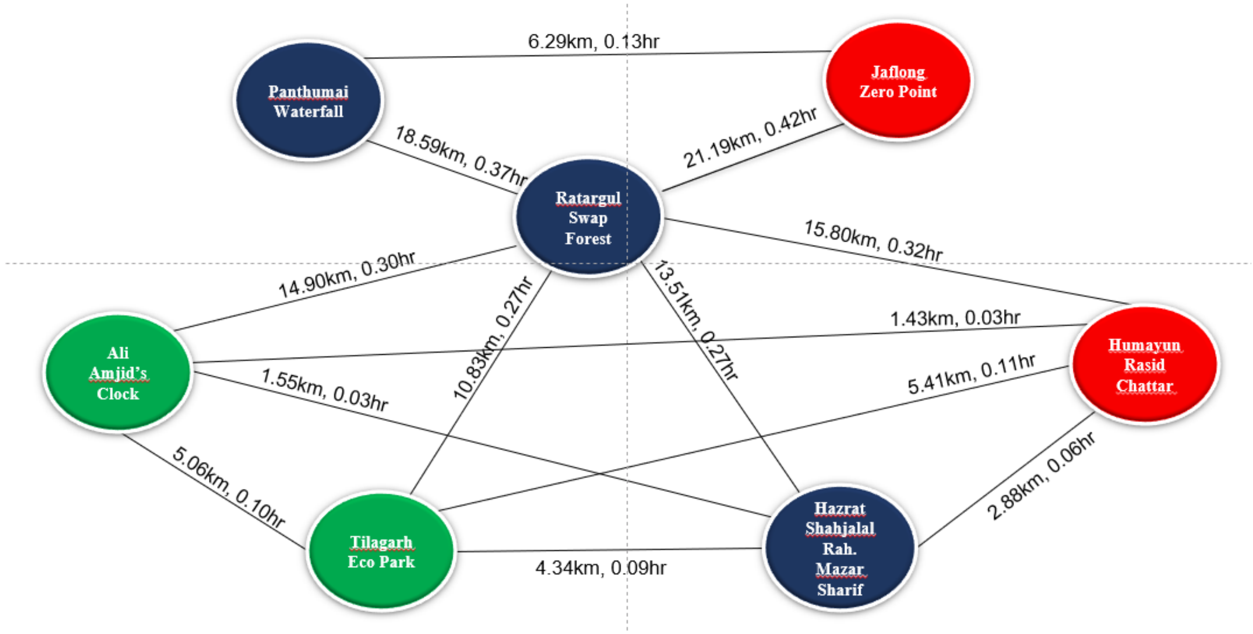


Figure 6 Manual Graph with distance and time

3.5.2 Recommendation

The second phase focuses on finding, scoring, and selecting the best route from the start to the end location. This involves finding possible paths, applying optimization algorithms, and preparing the final output. The flow of recommendation methodology is shown in figure 7.

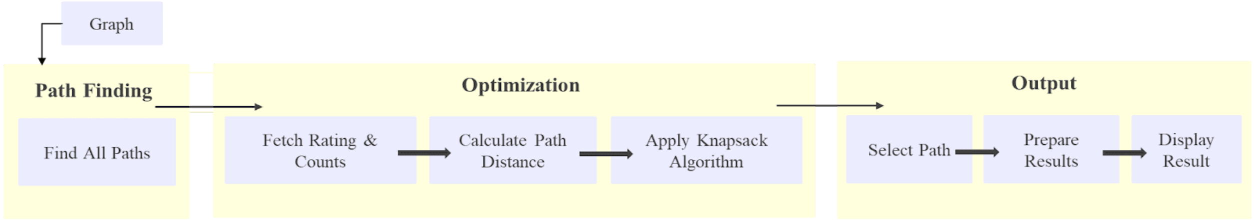


Figure 7 Recommendation Phrase.

Path Finding:

- **Find All Paths**: Using the graph, the system finds all possible paths from the starting point to the endpoint. The find_all_paths function does this, generating multiple route options that will be evaluated by the optimization algorithms. This approach ensures a wide range of possible paths are considered.

Optimization:

- **Fetch Ratings & Counts**: To evaluate the quality of each route, the system retrieves ratings and rating counts for each location in the paths. The fetch_ratings_and_counts function pulls this data from the database, allowing the system to consider each location's popularity.
- **Calculate Path Distance**: For each path, the system calculates the total travel distance by adding up the distances between each node. The calculate_path_distance function performs these calculations, helping to estimate travel time. Fourty minutes has been added to this estimated time as visiting time for each place as a default visiting time. For each segment of the path, the distance is calculated using the Haversine formula.
- **Apply Knapsack Algorithm**: A knapsack-style optimization algorithm is used to maximize the route score within the time. The calculate_knapsack_score function evaluates each path based on factors like ratings, and travel time. This algorithm balances these factors to find the highest-scoring route within the travel constraints. Specifically, we used **Knapsack with Dynamic Optimization**, which selects the segments that maximize the total score without exceeding the time limit.

  To find the highest-scoring route within the given time constraint, a dynamic programming (DP) table is used. This table keeps track of the maximum score that can be achieved for each time limit with each route segment.

  The DP formula is as follows:

  $$DP_{i,t} = \max\left(DP_{i-1,t}, DP_{i-1,t-SegmentTime_i} + SegmentProfit_i\right)$$

  where:

  - i represents the index of the route segment.
  - t represents the available time.
  - $DP_{i,t}$ stores the maximum profit obtainable using the first i segments within an available time t.
  - $SegmentTime_i$ is the time required to travel the current segment.
  - $SegmentProfit_i$ is the score for the segment, calculated based on factors such as ratings, rating count, and time duration.

  In this approach:

  The **max** function compares:

  - **Not including the current segment**: $DP_{i-1,t}$ which is the score without using segment i.
  - **Including the current segment**: $DP_{i-1,t-SegmentTime_i} + SegmentProfit_i$, which is the score if we include segment i, adding its profit while deducting its time from the total available time.

This method allows the system to balance the total score with the travel time constraint, ensuring the optimal route within the given time limit.

Output:

- **Select Path**: After scoring, the best route is chosen based on the optimization results. Routes are sorted by their total score, and the highest-scoring route that meets the time.
- **Prepare Results**: The selected route, along with details like total score, and travel time, is formatted for display. This step organizes the information so that it's easy for users to understand.
- **Display Result**: Finally, the system shows the recommended route to the user. The output includes route details, the total score, and estimated travel time, providing a clear and helpful recommendation.

This methodology combines database management, distance calculations, graph theory, and optimization techniques to create an efficient and effective route recommendation system. The block diagram illustrates each step, showing the process from data collection and processing to graph building, pathfinding, and final results. This approach ensures that only relevant locations are considered, optimal routes are identified, and the best recommendation is provided based on the user's needs and preferences.

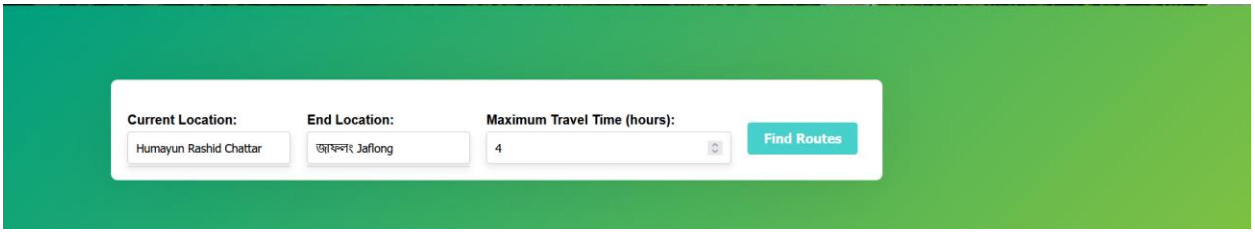
3.6 Design and Framework

This section discuss about user interface and interaction with system, admin panel and E-R diagram of this tourism recommendation system.

3.6.1 User Panel

This travel recommendation system simplifies route planning with popular tourist spots by providing users with top travel routes, available travel times, and detailed maps. Users can easily compare and select their preferred route and also share feedback to improve the system.

Travel Route Planning:



Current Location: Humayun Rashid Chattar

End Location: জাফলং Jaflong

Maximum Travel Time (hours): 4

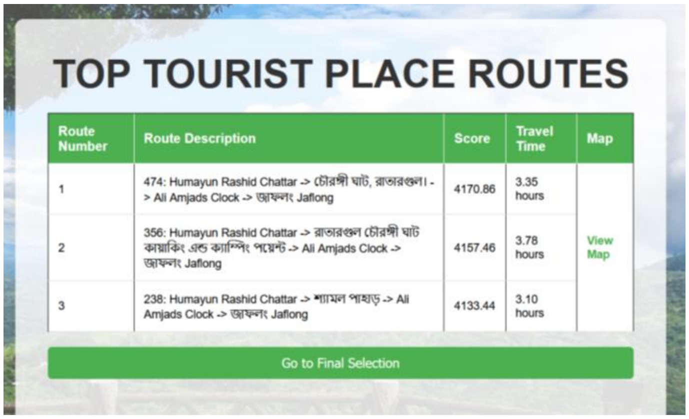
Figure 8 User Input Interface.

In figure 8, this form helps users to plan their travel routes by providing:

- **Location Input:** Fields to enter the starting and ending locations, with real-time suggestions for user convenience.
- **Travel Time Limit:** An input for specifying the maximum travel time in hours.
- **Route Submission:** A "Find Routes" button that submits the form to generate possible travel routes based on user input.

It simplifies the travel planning by capturing essential details in an organized, user-friendly format.

Recommended Travel Routes:



TOP TOURIST PLACE ROUTES

Route Number	Route Description	Score	Travel Time	Map
1	474: Humayun Rashid Chattar -> চৌরঙ্গী ঘাট, রাতারগুল। -> Ali Amjads Clock -> জাফলং Jaflong	4170.86	3.35 hours	View Map
2	356: Humayun Rashid Chattar -> রাতারগুল চৌরঙ্গী ঘাট কায়াকিং এন্ড ক্যাম্পিং পয়েন্ট -> Ali Amjads Clock -> জাফলং Jaflong	4157.46	3.78 hours	
3	238: Humayun Rashid Chattar -> শ্যামল পাহাড় -> Ali Amjads Clock -> জাফলং Jaflong	4133.44	3.10 hours	

Figure 9 Top Travel routes are displaying according to user preference.

In figure 9, this **"Top Travel Routes"** section shows the top 3 routes that present a list of recommended travel routes, ranked by preference or score. Each route entry in the table includes the following details:

- **Route Number:** Displays the rank or order of the route, starting from the top recommended route.
- **Route Description:** Lists the specific locations and stops included in each route, formatted sequentially to give an overview of the travel path. For example, routes might start at a bridge or ghat and end at a historical site.
- **Score:** The score provided means if the user followed highest scored route, they can visit most popular tourist places in their available time between current and end location they provided.
- **Travel Time:** This shows the estimated time to complete each route, providing users with an idea of the journey duration.
- **Map: The** "View Map" link allows users to view a detailed map of the route.

Below the table, a large green button labeled **"Go to Final Selection"** is available, suggesting users can make a final choice. This interface design is likely intended to help users easily compare and choose among top travel route options based on score, duration, and destinations.

Interactive Travel Route Map:

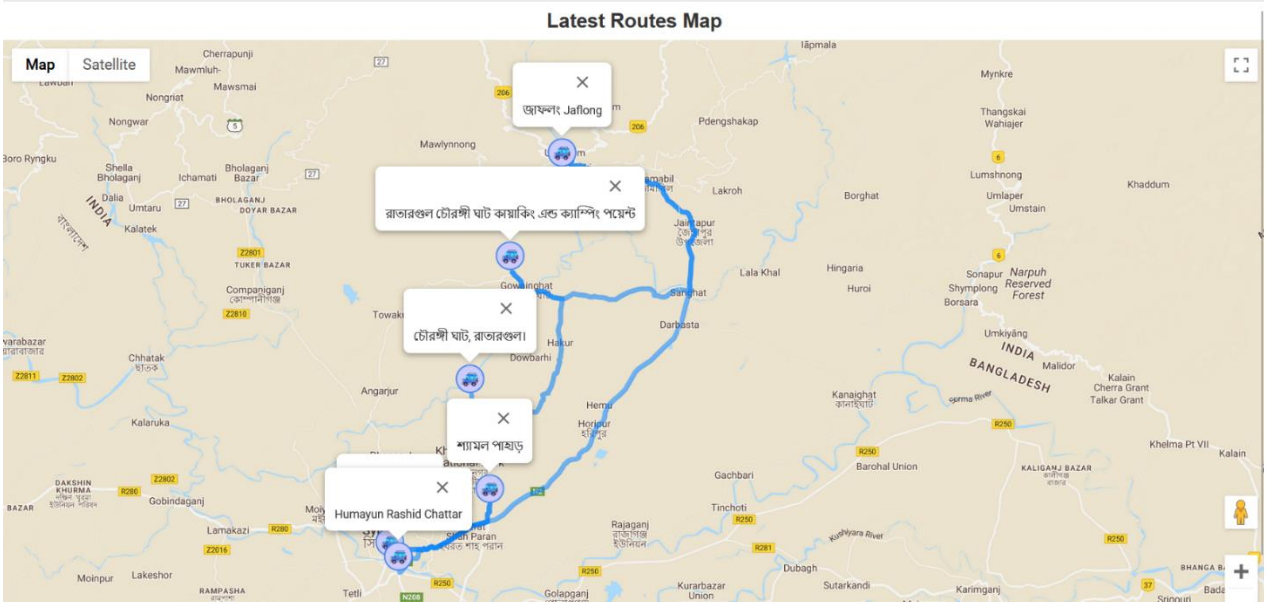
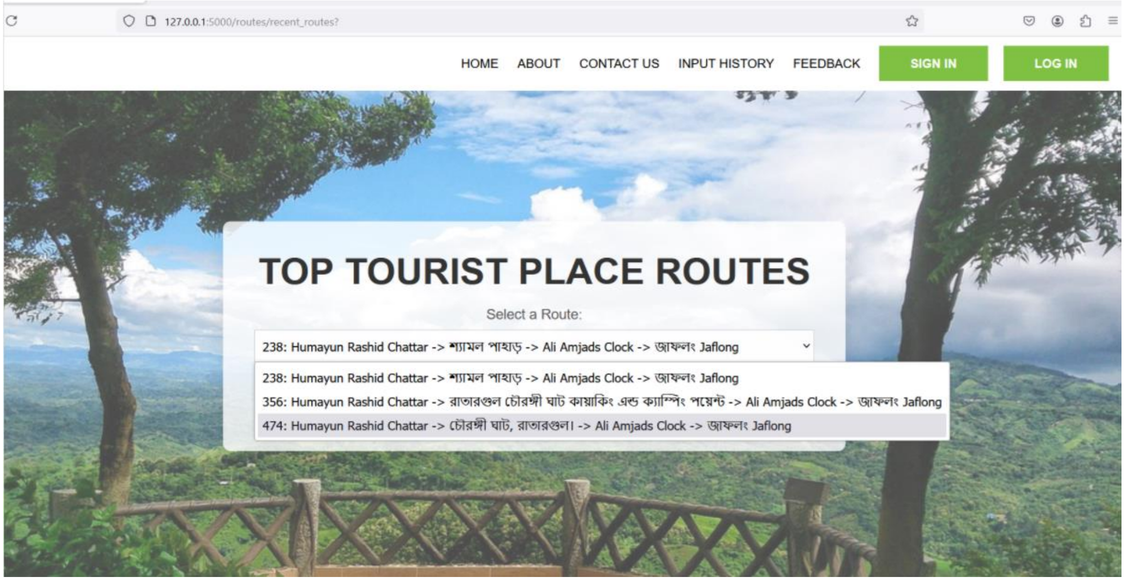


Figure 10 Travel routes in Map.

In figure 10, This map shows a recommended travel route with tourist spots between current and end location in user's available time they provide. Users can explore the route, check landmarks, and see estimated travel times to plan their journey better.

Final Route Selection:



TOP TOURIST PLACE ROUTES

Select a Route:

238: Humayun Rashid Chattar -> শ্যামল পাহাড় -> Ali Amjads Clock -> জাফলং Jaflong

238: Humayun Rashid Chattar -> শ্যামল পাহাড় -> Ali Amjads Clock -> জাফলং Jaflong
356: Humayun Rashid Chattar -> রাতারগুল চৌরঙ্গী ঘাট কায়াকিং এন্ড ক্যাম্পিং পয়েন্ট -> Ali Amjads Clock -> জাফলং Jaflong
474: Humayun Rashid Chattar -> চৌরঙ্গী ঘাট, রাতারগুল। -> Ali Amjads Clock -> জাফলং Jaflong

Figure 11User selecting the final route from top recommended routes.

Once the user clicks "**Go to Final Selection**" the system will display the top 3 recommended options which is shown in figure 11. The user can then review these suggestions and choose their preferred path based on the recommendations provided.

Feedback Panel:

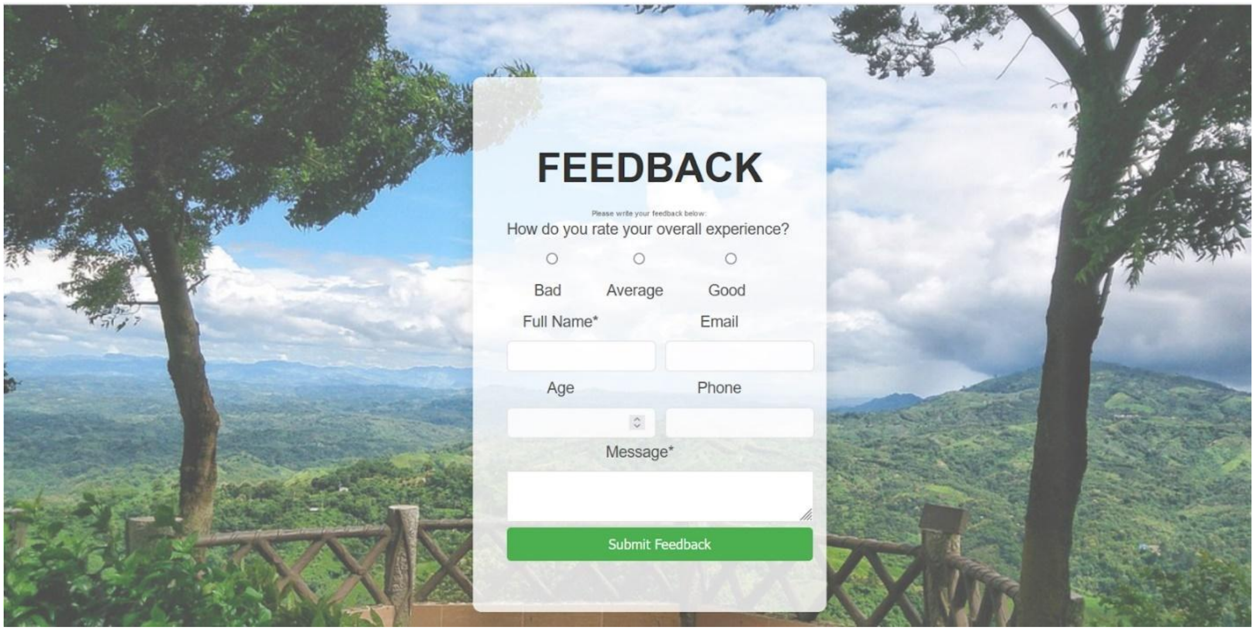


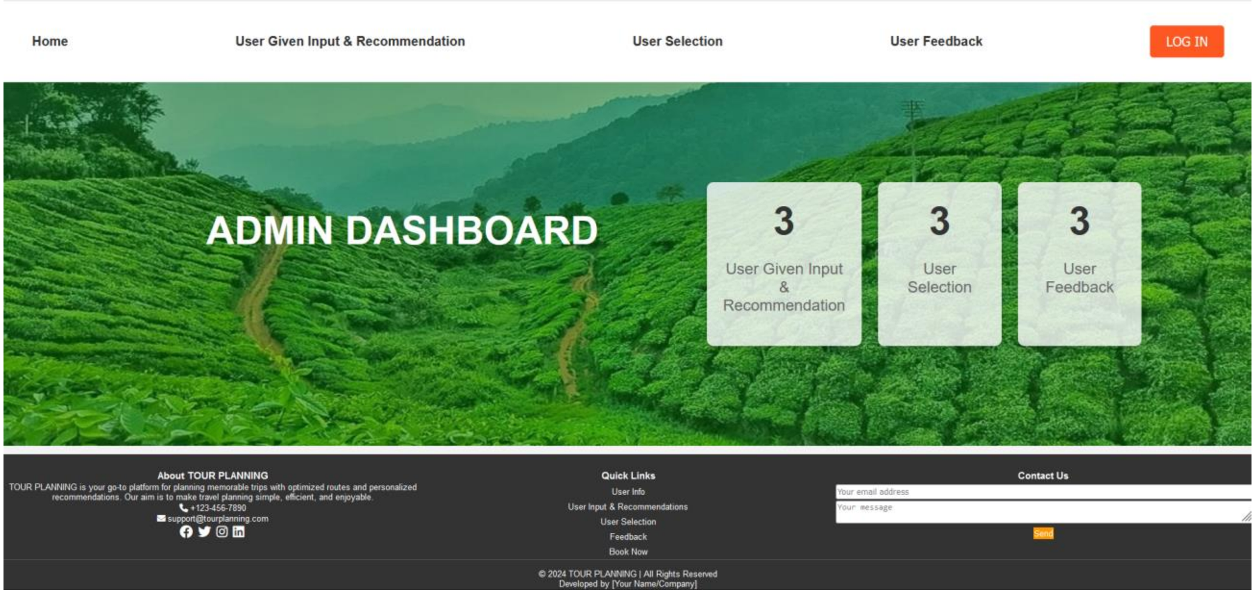
Figure 12 Feedback form for user.

Users have the option to provide feedback on their experience with the travel recommendation system by rating it as "Bad," "Average," or "Good." To submit feedback, users are required to provide their full name, email, age, and phone number. Additionally, users can include a feedback message to share their thoughts or suggestions. The feedback form is shown in figure 12. After submitting their feedback, users receive a "**Thank You**" message and a confirmation that their route has been successfully selected.

3.6.2 Admin Panel

The admin panel is a Flask-powered web application that is designed for managing both user data and feedback entries stored in a MySQL database. It offers an engaging interface for administrators to handle data management tasks and keep information up-to-date efficiently. This is a modular application with two distinct sections: feedback and user management. It is structured using Flask Blueprints-review_app for feedback-related functionality and user_app for user-related tasks.

Dashboard:



TOUR PLANNING is your go-to platform for planning memorable trips with optimized routes and personalized recommendations. Our aim is to make travel planning simple, efficient, and enjoyable.

☎ +123-456-7890
✉ support@tourplanning.com

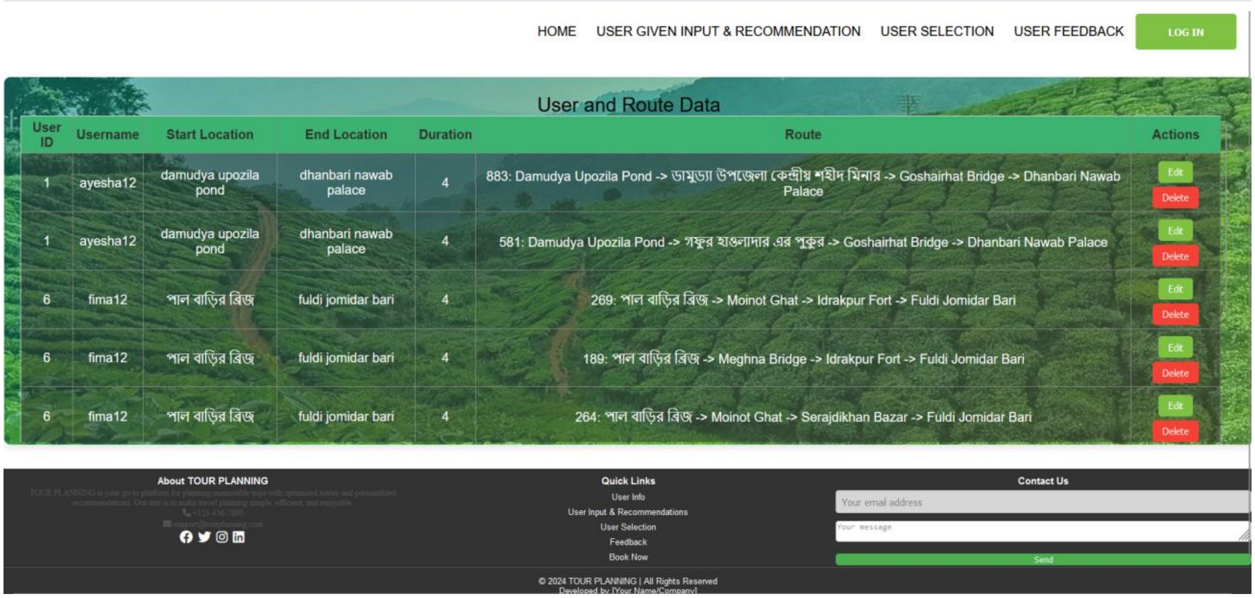
© 2024 TOUR PLANNING | All Rights Reserved
Developed by [Your Name/Company]

Figure 13 Admin Dashboard.

In figure 13, The dashboard route is the main entry point for the application. The dashboard.html template is rendered here. It provides:

- A summary or overview of the application's different sections, such as Feedback Management, User Management, and Route Selection.
- It provides easy access to all the key features, making it simple for the admin to navigate and perform tasks like managing feedback, users, and routes efficiently.

User and Route Data:



User and Route Data

User ID	Username	Start Location	End Location	Duration	Route	Actions
1	ayesha12	damudya upozila pond	dhanbari nawab palace	4	883: Damudya Upozila Pond -> ডামুড্যা উপজেলা কেন্দ্রীয় শহীদ মিনার -> Goshairhat Bridge -> Dhanbari Nawab Palace	Edit Delete
1	ayesha12	damudya upozila pond	dhanbari nawab palace	4	581: Damudya Upozila Pond -> গফুর হাওলাদার এর পুকুর -> Goshairhat Bridge -> Dhanbari Nawab Palace	Edit Delete
6	fima12	পাল বাড়ির ব্রিজ	fuldi jomidar bari	4	269: পাল বাড়ির ব্রিজ -> Moinot Ghat -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
6	fima12	পাল বাড়ির ব্রিজ	fuldi jomidar bari	4	189: পাল বাড়ির ব্রিজ -> Meghna Bridge -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
6	fima12	পাল বাড়ির ব্রিজ	fuldi jomidar bari	4	264: পাল বাড়ির ব্রিজ -> Moinot Ghat -> Serajdikhan Bazar -> Fuldi Jomidar Bari	Edit Delete

TOUR PLANNING is your go-to platform for planning memorable trips with optimized routes and personalized recommendations. Our aim is to make travel planning simple, efficient, and enjoyable.

☎ +123-456-7890
✉ support@tourplanning.com

© 2024 TOUR PLANNING | All Rights Reserved
Developed by [Your Name/Company]

Figure 14 User and Route Data information.

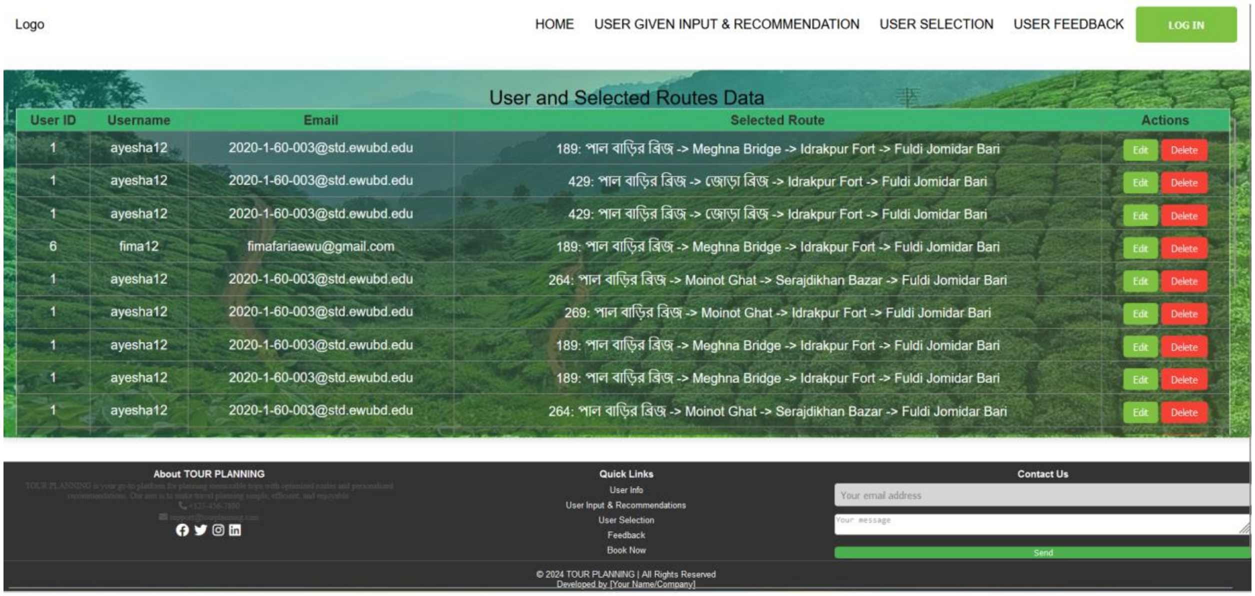
This figure 14 focuses on managing data related to both users and routes, offering the ability to:

Viewing User and Route Data: The system fetches and displays data by combining the “users” and “routes” tables based on a relationship between them. It presents user-related information such as their “username” and “user_id” and also it added route details like “start_location”, “end_location”, “duration”, and “route_id”. This combined data is shown in a table format for administrators to easily review.

Edit User Information: Admins can directly update user information from within the admin panel. By selecting a user, they can modify information such as the username. Once changes are submitted, the system securely updates the user data in the database, with proper error handling to ensure a smooth and reliable process.

- **Delete Routes:** This feature gives the administrator the rights to delete certain routes from the database. When a route is no longer needed,the administrator can choose the ID of the route he wants to delete , triggering a query to remove it. The system confirms that the record was successfully deleted; outdated or incorrect routes are removed efficiently.

User and Selected Routes Data:



User and Selected Routes Data

User ID	Username	Email	Selected Route	Actions
1	ayesha12	2020-1-60-003@std.ewubd.edu	189: পাল বাড়ির ব্রিজ -> Meghna Bridge -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	429: পাল বাড়ির ব্রিজ -> জোড়া ব্রিজ -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	429: পাল বাড়ির ব্রিজ -> জোড়া ব্রিজ -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
6	fima12	fimafariaewu@gmail.com	189: পাল বাড়ির ব্রিজ -> Meghna Bridge -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	264: পাল বাড়ির ব্রিজ -> Moinot Ghat -> Serajdikhan Bazar -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	269: পাল বাড়ির ব্রিজ -> Moinot Ghat -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	189: পাল বাড়ির ব্রিজ -> Meghna Bridge -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	189: পাল বাড়ির ব্রিজ -> Meghna Bridge -> Idrakpur Fort -> Fuldi Jomidar Bari	Edit Delete
1	ayesha12	2020-1-60-003@std.ewubd.edu	264: পাল বাড়ির ব্রিজ -> Moinot Ghat -> Serajdikhan Bazar -> Fuldi Jomidar Bari	Edit Delete

About TOUR PLANNING

TOUR PLANNING is your go-to platform for planning memorable trips with optimized routes and personalized recommendations. Our aim is to make travel planning simple, efficient, and enjoyable.

+123-456-7890

support@tourplanning.com

© 2024 TOUR PLANNING | All Rights Reserved
Developed by [Your Name/Company]

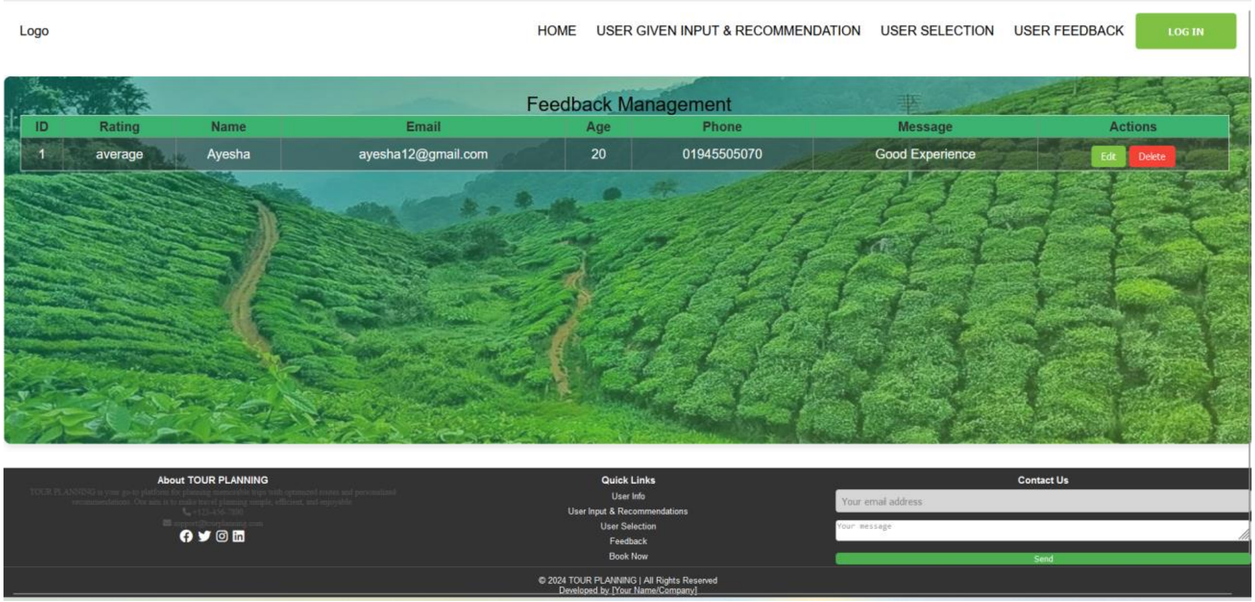
Figure 15 User and Selected Route Data Information.

In figure 15, this interafce manages the relationship between users and the routes they have selected. It includes functionality for:

- Displaying a list of routes selected by users, showing details of users such as their username and email.
- Allowing administrators to edit user details (username, email) when necessary.
- Deleting user-selected routes as required.

This module helps administrators track and manage the routes that users have selected, providing a streamlined way to update or remove routes as needed.

Feedback Management



Feedback Management

ID	Rating	Name	Email	Age	Phone	Message	Actions
1	average	Ayesha	ayesha12@gmail.com	20	01945505070	Good Experience	Edit Delete

About TOUR PLANNING

TOUR PLANNING is your go-to platform for planning memorable trips with optimized routes and personalized recommendations. Our aim is to make travel planning simple, efficient, and enjoyable.

☎ +123-456-7890

✉ support@tourplanning.com

© 2024 TOUR PLANNING | All Rights Reserved
Developed by [Your Name/Company]

Figure 16 Users Feedback Data.

In figure 16, this interface is responsible for managing user feedback. Administrators can:

- View all feedback submitted by users.
- Edit specific feedback entries, including fields like rating, name, email, age, phone number, and the feedback message.
- Delete feedback entries from the database.

This module allows administrators to maintain and update the feedback provided by users, ensuring that it stays relevant and accurate.

3.6.3 E-R Model

The figure 17 shows an entity-relationship diagram for the database with six tables: routes, users, user_selected_routes, dataset, feedback, and their primary and foreign key relationships. Those table are responsible to make this recommendation system with those interfaces.

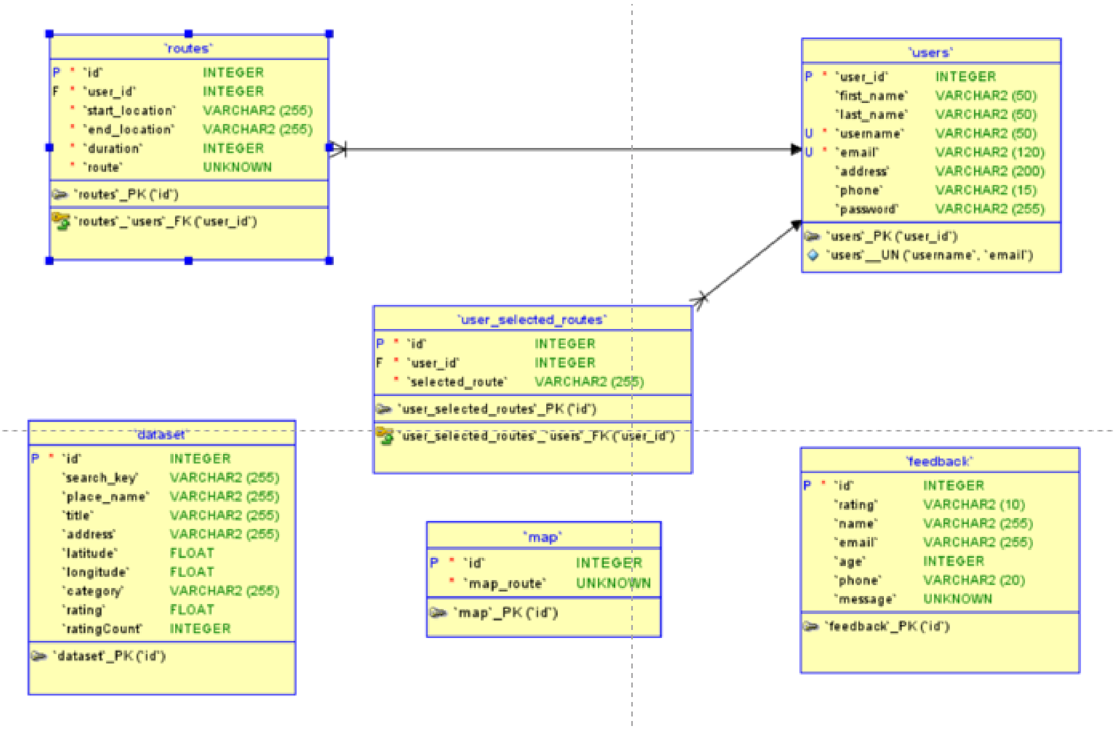


Figure 17 E-R diagram of Tourism Recommendation System.

Here's a breakdown of each table and its relationships:

routes Table: It stores the value user has given as input and the routes showed to used as top 3 recommended routes according to user start location, end location and travel duration.

Columns:

- id: Primary key (INTEGER)
- user_id: Foreign key referencing the user_id column in the users table
- start_location, end_location: Both are VARCHAR(255)
- duration: INTEGER
- route: text

There is a foreign key (user_id) linking each route to a specific user in the users table, creating a one-to-many relationship from users to routes.

users Table: This table stores user information.

Columns:

- user_id: Primary key (INTEGER)
- first_name, last_name: VARCHAR(50)

- username: VARCHAR(50), unique constraint
- email: VARCHAR(120), unique constraint
- address: VARCHAR(200)
- phone: VARCHAR(15)
- password: VARCHAR(255)

This table acts as the parent table in relationships with both routes and user_selected_routes tables, which reference user_id.

user_selected_routes Table: This table stores users selected routes.

Columns:

- id: Primary key (INTEGER)
- user_id: Foreign key referencing the user_id column in the users table
- selected_route: VARCHAR(255).

The table hase a foreign key (user_id) pointing to users.user_id, indicating which user selected specific routes. The foreign key relationship is marked with ON DELETE CASCADE, meaning if a user is deleted from users, related entries in user_selected_routes are automatically removed.

dataset Table: This table has all the information about the tourist spots. Those information are used to recommend user top routes according to user's preference.

Columns:

- id: Primary key (INTEGER)
- search_key, place_name, title, address: VARCHAR(255)
- latitude, longitude: FLOAT
- category: VARCHAR(255)
- rating: FLOAT
- ratingCount: INTEGER
- Relationships:

This table have no direct relationship with other tables.

feedback Table: The feedback, a user has given, will get stored in this table.

Columns:

- id: Primary key (INTEGER)
- rating: VARCHAR(10)
- name: VARCHAR(255)
- email: VARCHAR(255)
- age: INTEGER
- phone: VARCHAR(20)
- message: text

This table have no direct relationship with other tables.

Map Table: This table stores the information of recommended route, the system has provided. This information then uses to show route in map

Columns:

- id: Primary key (INTEGER)
- map_route: text

This table have no direct relationship with other tables.

Key Observations:

- **Foreign Key Definitions**: The routes and user_selected_routes tables each have foreign keys referencing users.user_id, establishing links back to specific users.
- **Unique Constraints**: The users table has unique constraints on username and email, ensuring no duplicates in these fields.

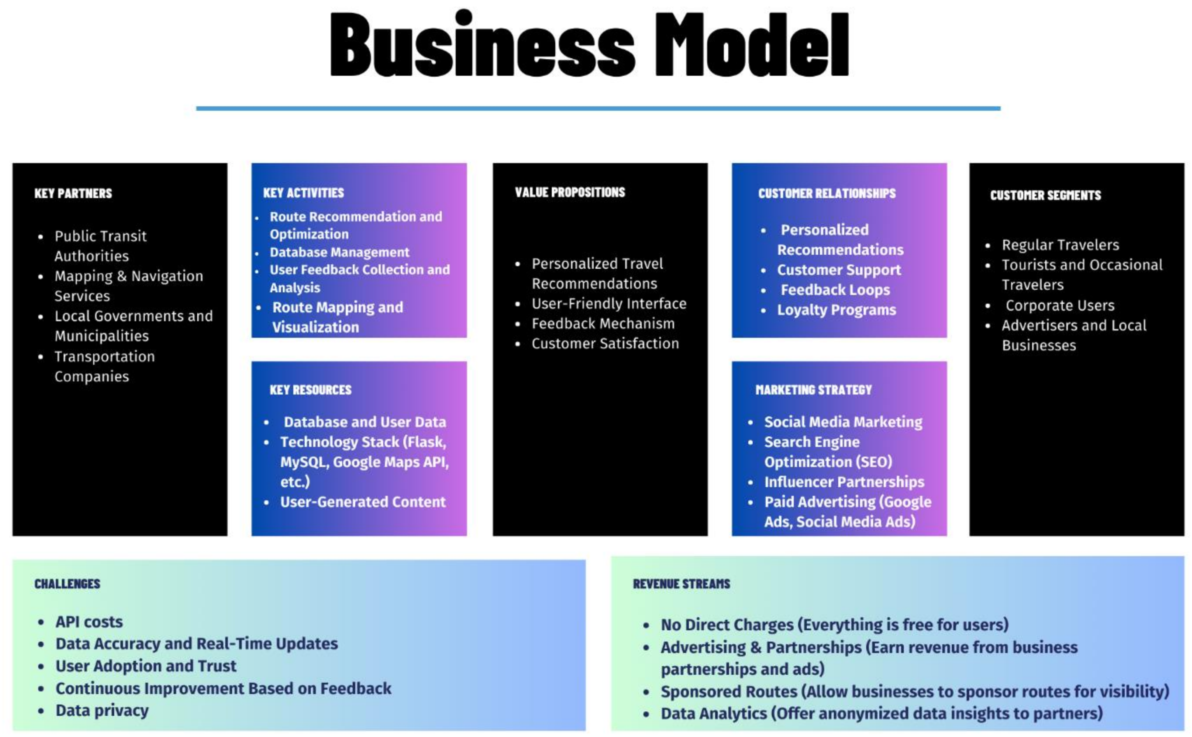
3.7 Experimental Setup

The experimental setup for this recommendation system integrates several components for data management, algorithmic processing, and user interface design:

- **Development Environment**: PyCharm was used as the primary IDE to develop the backend and recommendation algorithms, along with the frontend interface.
- **Database Management**: The database was implemented in MySQL using XAMPP to manage and store information about users, routes, and feedback. A processed JSON dataset containing detailed information on tourist spots was imported into the database and structured in SQL format for efficient querying.
- **Backend Framework**: Flask, a Python web framework, was utilized to handle server-side logic, manage user requests, and serve the recommendation results. Flask routes facilitated interactions between the front end and backend, providing a seamless experience for users.
- **Frontend Interface**: HTML, CSS, JavaScript, and Flask were used to create a user-friendly interface for inputting travel details, viewing suggested routes, and providing feedback. The design emphasizes simplicity and responsiveness, allowing users to easily select their starting and ending locations, set time constraints, and receive optimized route recommendations.
- **Google Map API**: The Google Map API key has been used to show the route in the map for better visualization for the traveller. To use this key, we have paid to pay google cloud. This is the cost of using this model.

3.8 Business Model

This section describes the Business Model for a Travel Recommendation System that helps people find the best travel routes without charging them. This comprehensive model guides the system's development, user engagement, and revenue generation to ensure a sustainable and valuable travel planning service.



Business Model

KEY PARTNERS

- Public Transit Authorities
- Mapping & Navigation Services
- Local Governments and Municipalities
- Transportation Companies

KEY ACTIVITIES

- Route Recommendation and Optimization
- Database Management
- User Feedback Collection and Analysis
- Route Mapping and Visualization

KEY RESOURCES

- Database and User Data
- Technology Stack (Flask, MySQL, Google Maps API, etc.)
- User-Generated Content

VALUE PROPOSITIONS

- Personalized Travel Recommendations
- User-Friendly Interface
- Feedback Mechanism
- Customer Satisfaction

CUSTOMER RELATIONSHIPS

- Personalized Recommendations
- Customer Support
- Feedback Loops
- Loyalty Programs

MARKETING STRATEGY

- Social Media Marketing
- Search Engine Optimization (SEO)
- Influencer Partnerships
- Paid Advertising (Google Ads, Social Media Ads)

CUSTOMER SEGMENTS

- Regular Travelers
- Tourists and Occasional Travelers
- Corporate Users
- Advertisers and Local Businesses

CHALLENGES

- API costs
- Data Accuracy and Real-Time Updates
- User Adoption and Trust
- Continuous Improvement Based on Feedback
- Data privacy

REVENUE STREAMS

- No Direct Charges (Everything is free for users)
- Advertising & Partnerships (Earn revenue from business partnerships and ads)
- Sponsored Routes (Allow businesses to sponsor routes for visibility)
- Data Analytics (Offer anonymized data insights to partners)

Figure 18 Business Model.

Essential data for this Travel Recommendation System is provided by key partners, such as public transit authorities and mapping services. The core activities focus on generating customized route suggestions and continuously enhancing the service through user feedback, ensuring an ever-improving, user-friendly experience.

To build strong connections, the system offers customer support and loyalty programs to keep users engaged and foster long-term relationships.

Though The system is completely free for users, It generates revenue through targeted advertising, strategic partnerships, and sponsored routes that allow businesses to promote their offerings. With a marketing strategy on social media, SEO, and influencer collaborations, the model attracts a wide audience. This approach ensures the system remains valuable, accessible, and financially sustainable, offering an essential travel solution at no cost to users.

4. Result and Discussion

The results for each algorithm are presented in tables showing the recommended routes, scores, travel times. The input of the result is also given below in table 5.

Input:

Travel Time: 4 hours

Current Location: Humayun Rashid

End Location: জাফলং Jaflong

Algorithm	Route	Score	Travel Time
Knapsack with dynamic optimization **Runtime:** 1.45 seconds	Humayun Rashid Chattar -> চৌধুরী ঘাট, রাতারগুল -> Ali Amjads Clock -> জাফলং Jaflong	4170.86	3.35 hours
	Humayun Rashid Chattar -> রাজরাজেশ্বর চৌধুরী ঘাট -> Ali Amjads Clock -> জাফলং Jaflong	4157.46	3.78 hours
	Humayun Rashid Chattar -> ম্যাসাল পাহাড় -> Ali Amjads Clock -> জাফলং Jaflong	4133.44	3.10 hours
Multiple criteria decision making **Runtime:** 21.84 seconds	Humayun Rashid Chattar -> চৌধুরী ঘাট, রাতারগুল -> Ali Amjads Clock -> জাফলং Jaflong	4167.68	3.35 hours
	Humayun Rashid Chattar -> Ali Amjads Clock -> রাহাত মঞ্জিল–> জাফলং Jaflong	4147.75	3.03 hours
	Humayun Rashid Chattar -> Ali Amjads Clock -> গাজিউর হাওর-> জাফলং Jaflong	4143.84	3.11 hours
Knapsack dynamic with Traffic optimization **Runtime:** 21.12 seconds	Humayun Rashid Chattar -> Ali Amjads Clock -> Jaflong Zero Point -> জাফলং Jaflong	6844.59	3.10 hours
	Humayun Rashid Chattar -> Ali Amjads Clock -> Shahi Eidgah দক্ষিন চত্বর –> জাফলং Jaflong	6363.74	3.10 hours
	Humayun Rashid Chattar -> Ali Amjads Clock -> Tilagarh Eco Park -> জাফলং Jaflong	5976.23	3.37 hours

Simulated Annealing **Runtime:** 25.49 seconds	Humayun Rashid Chattar -> চৌধুরী ঘাট, রাতারগুল ->Ali Amjads Clock ->জাফলং Jaflong	4170.86	3.35 hours
	Humayun Rashid Chattar -> Ali Amjads Clock -> রাহাত মঞ্জিল –> জাফলং Jaflong	4150.54	3.03 hours
	Humayun Rashid Chattar -> Ali Amjads Clock -> গাজিউর হাওর -> জাফলং Jaflong	4146.72	3.11 hours

Table 5: Results of those four optimization algorithms.

Knapsack with Dynamic Optimization performed well in terms of both score and runtime. It offered a high score (4170.86) with a low runtime (1.45 seconds), making it suitable for scenarios requiring quick and reliable decisions.

Knapsack with Traffic Optimization slightly increased the runtime (21.12 seconds) due to additional computations for traffic considerations. However, it still maintained the highest score (6844.59) with the primary route and proposed an alternative route with reduced travel time (3.37) but a slightly lower score (5976.23).

Simulated Annealing provided similar results to Knapsack but with a higher runtime (25.49 seconds). While it ensures global optimization, the added computational time may not justify its use for real-time decision-making.

Multiple Criteria Decision-Making balanced performance by offering high scores (4167.68 for the best route and 4147.75 for the alternative) and comparable runtimes (21.84 seconds). This method is particularly useful for scenarios requiring flexibility in route selection based on varying criteria.

Discussion:

Overall, the **Knapsack with Dynamic Optimization** stands out as the most efficient approach for low runtime routes with minimal computational overhead compared to other methods. This is why we use it as the core of our system. Traffic optimization and multiple criteria methods add value by providing alternative routes based on specific constraints such as travel time or multiple criteria, offering more adaptable solutions for diverse scenarios.

5. Conclusion & Future Work

This project focuses on creating a user-centered travel recommendation system that generates multi-destination, time-optimized itineraries customized to user preferences. It highlights how traveler choices are influenced by factors such as time constraints, attraction popularity, personal interests, and convenience, which are smoothly integrated into the system. By balancing personalization and efficiency, the system combines user input with optimized route planning to deliver practical yet tailored itineraries.

Essential data inputs, including user preferences, current location, and time constraints, enable the creation of relevant and comprehensive travel plans. Addressing a common gap in existing systems, this project prioritizes logical multi-destination sequencing and end-point inclusion within given timeframes. Advanced algorithms such as genetic algorithms and collaborative filtering are utilized to balance budget, time, and traveler preferences effectively.

While the system does not yet incorporate real-time data, enhancements like traffic updates, weather conditions, and attraction schedules could significantly improve accuracy and adaptability. Incorporating user interest data ensures recommendations are both enjoyable and time-efficient, while continuous feedback loops refine personalization and responsiveness over time.

Additionally, the system upholds privacy by avoiding personal data collection and ensuring robust security measures. Although it does not adapt to sudden closures or delays in real-time, it offers optimized itineraries that users can adjust as needed, providing a flexible and reliable travel planning experience.

This research establishes a foundation for a practical and efficient travel recommendation system that bridges current gaps while prioritizing user needs and adaptability.

5.1 Research Findings and Contributions

This project improves travel recommendation systems by offering a structured way of multi-destination planning, which is currently lacking in many solutions. Considering starting and ending points, route optimization, and machine learning, this system improves travel experiences to ensure user interests and data security.

5.2 Significance of Results

The system designs personalized travel plans based on both time and real-time data to help travelers make better decisions faster, with great satisfaction and reduced decision fatigue.

5.3 Potential Applications

This system can be used by tourism agencies, travel apps, and personal travel planners. It can also integrate with local businesses like hotels and restaurants to offer more travel options.

5.4 Limitations

Challenges include managing inconsistent data across languages and dealing with limited real-time data due to API restrictions, especially in areas with poor data sources.

5.6 Future Work

Future improvements could include:

- **Real-Time Data:** Adding traffic, weather, and attraction data for more dynamic itineraries.
- **Machine Learning:** Enhancing personalization and refining recommendations over time.
- **Sustainable Tourism:** Adding eco-friendly travel options and tracking carbon footprints.
- **Local Collaborations:** Partnering with local businesses for better deals and real-time updates.
- **Offline and Predictive Features:** Including features like offline and predictive, which allow access offline and predict travel trends through analytics.

The project will help individualized travel planning in terms of smoothness and adaptability, therefore opening a wide array of opportunities for enhancements in travel technology.

6. References

1. S. Kokate, A. Gaikwad, P. Patil, M. Gutte and K. Shinde(2018). "Traveler's Recommendation System Using Data Mining Techniques," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, pp. 1-5, 2018, doi: 10.1109/ICCUBEA.2018.8697862.
2. Sutomo, Rudi & Pratama, Daffa, “Measuring Tourist Experience in Semarang City through an Advanced Recommendation System,” Jurnal Komunikasi, Sains dan Teknologi, vol.2, p.192-200, 2023, https://doi.org/10.61098/jkst.v2i2.56.
3. M. Muneer, U. Rasheed, S. Khalid and M. Ahmad, "Tour Spot Recommendation System via Content-Based Filtering," *16th International Conference on Open Source Systems and Technologies (ICOSST)*, Lahore, Pakistan, 2022, pp. 1-6, 2022, doi:10.1109/ICOSST57195.2022.10016820.
4. Y. Zhang, L. Jiao, Z. Yu, Z. Lin and M. Gan, "A Tourism Route-Planning Approach Based on Comprehensive Attractiveness," in *IEEE Access*, vol. 8, pp. 39536-39547, 2020, doi: 10.1109/ACCESS.2020.2967060.
5. E. Caterino, “Tourist Attraction Data: Best Datasets & Databases 2024 | Datarade.” https://datarade.ai/data-categories/tourist-attraction-data
6. W. Shafqat and Y.-C. Byun, “A Context-Aware Location Recommendation System for Tourists Using Hierarchical LSTM Model,” *Sustainability*, vol. 12, no. 10, p. 4107, May 2020, doi: 10.3390/su12104107.
7. S. Bashir, S. Raza, and V. Misic, “BERT4Loc: BERT for Location—POI Recommender System,” *Future Internet*, vol. 15, no. 6, p. 213, Jun. 2023, doi: 10.3390/fi15060213.

8. N. Nurseitov, M. Paulson, R. Reynolds, C. Izurieta, Department of Computer Science, and Montana State University – Bozeman, “Comparison of JSON and XML Data Interchange Formats: A Case Study,” 2019. [Online]. Available: https://www.cs.montana.edu/izurieta/pubs/IzurietaCAINE2009.pdf

9. D. Peng, L. Cao, and W. Xu, “Using JSON for data exchanging in web service applications,” *Journal of Computational Information Systems*, vol. 7, no. 16, pp. 5883–5890, December 2011. [Online]. Available: https://www.researchgate.net/publication/265874991.

10. D. V. Shetty and S. J. Chidimar, “Comparative study of SQL and NoSQL databases to evaluate their suitability for big data application,” *International Journal of Computer Science and Information Technology Research*, vol. 4, no. 2, pp. 314–318, Apr.–Jun. 2016. [Online]. Available: www.researchpublish.com.

11. H. Ashtari, “NoSQL Basics: Features, Types, and Examples - Spiceworks Inc,” *Spiceworks Inc*, Oct. 18, 2022. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-nosql/#:~:text=Document%2Doriented%20NoSQL%20database%20solutions,Amazon%20SimpleDB%2C%20and%20Lotus%20Notes.

12. C. Huda, Y. Heryadi, N. Lukas, and W. Budiharto, “A Tourism Dataset from Historical Transaction for Recommender Systems.,” *Data in Brief*, vol. 52, p. 109990, Dec. 2023, doi: 10.1016/j.dib.2023.109990.

13. W. Tourism and O. Unwto, “UNWTO World Tourism Barometer and Statistical Annex, May 2020,” *UNWTO World Tour. Barom.*, vol. 18, no. 2, pp. 1–48, 2020, doi: 10.18111/wtobarometereng.2020.18.1.2.

14. U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, “Smart tourism: foundations and developments,” *Electron. Mark.*, vol. 25, no. 3, pp. 179–188, 2015, doi: 10.1007/s12525-015-0196-8.

15. S. García, S. Ramírez-Gallego, J. Luengo, et al., "Big data preprocessing: methods and prospects," *Big Data Anal.*, vol. 1, p. 9, 2016. doi: 10.1186/s41044-016-0014-0.

7. Appendix

7.1 Appendix A

Addressing of COs, Knowledge Profile (K), and Complex Engineering Problem (EP):

CO	CO Descriptions	K	EP

CO3	Analyze various aspects of the objectives for designing a solution for the capstone project.	**(i)Problem Analysis[K1, K2, K3, K4]** **K1:** Understanding the limitations of traditional tourism recommendation systems. **K2:** Identifying the problem of time constraints faced by travelers in traditional systems. **K3:** Recognizing the gap in existing systems, specifically in considering time-specific travel plans. **K4:** Understanding the challenges in developing a personalized and effective recommendation system for tourism.	**(i) Problem Analysis [EP1, EP2, EP3, EP6, EP7]** **EP1:** Analyzing system requirements based on the time constraints of the users. **EP2:** Identifying and documenting key factors that influence tourist decision-making and route planning. **EP3:** Evaluating the performance of existing recommendation systems and how they fail to meet traveler needs. **EP6:** Gathering data through appropriate means (e.g., Google Places API) to inform system recommendations. **EP7:** Assessing the feasibility of the solution by considering available resources, datasets, and technology.

CO4	Design and develop solutions for the capstone project that meet public health and safety, cultural, societal, and environmental considerations.	**(i) Design and Implementation [K5]** **Public Health** **K5:** Developing an effective, personalized tourist recommendation system	**(i) Design and Implementation [EP1, EP2, EP4, EP5, EP6, EP7]** **EP1:** Designing the system architecture based on user input **EP2:** Designing the user interface for easy interaction with the system **EP5:** Integrating the system with real-time data sources **EP6:** Implementing a database system for storing and managing tourist data **EP7:** Testing and optimizing the system for performance and user experience
CO5	Identify and apply modern engineering and IT tools for the design and development of the capstone project.	**Materials and Devices [K6]** **K6:** Utilization of technologies and tools for data gathering, processing, and recommendation system development	**Materials and Devices [EP1, EP2, EP4, EP5]** **EP1:** Using the Google Places API for data collection **EP2:** Developing a MySQL database for storing tourist attraction data **EP4:** Implementing Python for data processing and transformation

7.2 Appendix B

Addressing of COs, Knowledge Profile (K), and Complex Engineering Problem (EP):

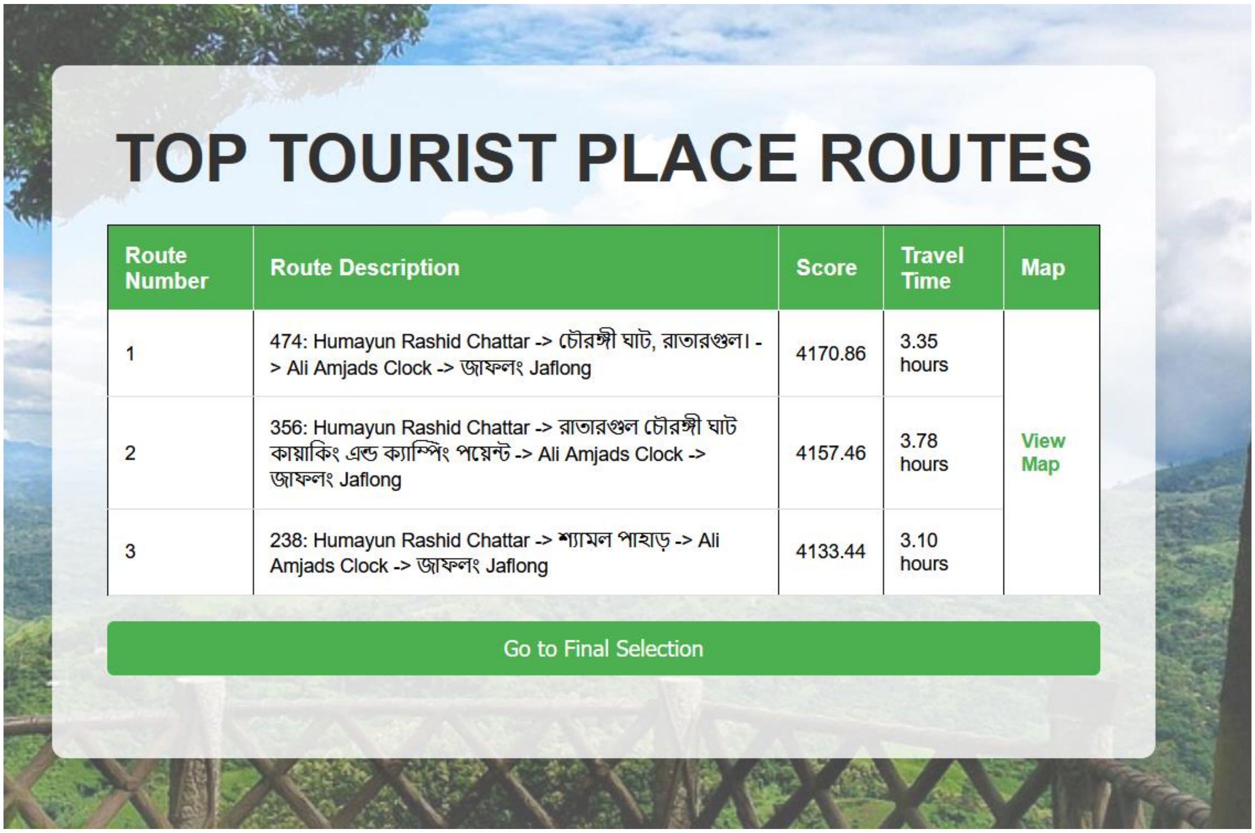
CO	CO Descriptions	K	EP
CO3	Analyze various aspects of the objectives for designing a solution for the capstone project.	**(i)Problem Analysis[K1, K2, K3, K4]** **Distance Calculation:** Uses Haversine formula to calculate distances between coordinates. **Filtering Places**: Removes duplicates and selects places based on proximity to the route and start/end points. **Graph Construction:** Creates a graph representing connections between places. **Pathfinding:** Finds possible routes between start and endpoints.	**(i) Problem Analysis [EP1, EP2, EP3, EP6, EP7]** **Distance Accuracy:** Ensures accurate distance calculations and route filtering. **Unique Identification:** Filters out duplicate places to ensure relevance. **Efficient Graph Building:** Constructs a graph to analyze connections and paths. **Path Evaluation:** Evaluate different routes for feasibility and relevance.
CO4	Design and develop solutions for the capstone project that meet public health and safety, cultural, societal, and environmental considerations.	**(i) Design and Implementation [K5]** **Public Health Considerations:** Optimized route planning using the Knapsack algorithm. Prioritized data privacy with secure protocols, supporting sustainability and user trust.	**(i) Design and Implementation [EP1, EP2, EP4, EP5, EP6, EP7]** **Safety and Accuracy:** Ensures safe and accurate route recommendations by filtering and evaluating paths. **Cultural and Societal Considerations:** Incorporates user-defined locations and preferences in the design.

CO5	Identify and apply modern engineering and IT tools for the design and development of the capstone project.	**Materials and Devices [K6]** **Modern Tools:** Uses Python and libraries (e.g., MySQL. connector, math, heapq) for data processing and analysis.	**Materials and Devices [EP1, EP2, EP4, EP5]** **Data Handling:** Manages large datasets with batch processing. **Graph Algorithms:** Applies graph-based algorithms for pathfinding and distance calculations.
CO6	Assess and address societal, health, safety, legal, and cultural aspects related to the implementation of the capstone project considering the relevant professional and engineering practices and solutions.	**Social and Environmental Impact of Engineering [K7]** **Societal Impact:** Provides tools for personalized route planning based on user preferences. **Safety Considerations:** Ensures safe travel recommendations by filtering places and paths.	**Social and Environmental Impact of Engineering [EP2, EP5, EP6]** **User-Centric Design:** Considers user input and preferences in route planning. **Safety and Relevance:** Ensures routes are safe and relevant to user-defined criteria.

8. User Feedback Questionnaire

In this section, we gathered feedback from over 80 users who have experienced our travel recommendation system website. Their reviews provide valuable insights into the strengths and areas for improvement to help us enhance the overall user experience and functionality of the platform.

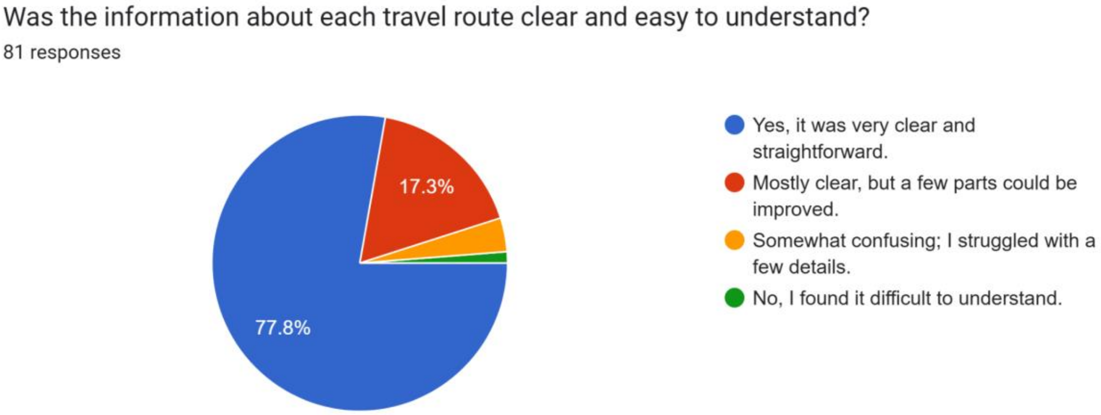
This travel recommendation system simplifies route planning with popular tourist spots across each Upazila in Bangladesh. Users can have the top travel routes with popular tourist spots between their current and end location, that can be visited within their available travel time. Detailed map that shows a travel route with tourist spots between current and end location in user's available time they provide. Top three routes will be provided and user can view those routes in map. Users can easily compare and select their preferred route and also they have the option to share feedback to help improving the system's accuracy and user experience for travelers across all areas of Bangladesh. The score provided means if the user followed highest scored route, they can visit most popular tourist places in their available time between current and end location they provided.



TOP TOURIST PLACE ROUTES

Route Number	Route Description	Score	Travel Time	Map
1	474: Humayun Rashid Chattar -> চৌরঙ্গী ঘাট, রাতারগুল। -> Ali Amjads Clock -> জাফলং Jaflong	4170.86	3.35 hours	View Map
2	356: Humayun Rashid Chattar -> রাতারগুল চৌরঙ্গী ঘাট কায়াকিং এন্ড ক্যাম্পিং পয়েন্ট -> Ali Amjads Clock -> জাফলং Jaflong	4157.46	3.78 hours	
3	238: Humayun Rashid Chattar -> শ্যামল পাহাড় -> Ali Amjads Clock -> জাফলং Jaflong	4133.44	3.10 hours	

Question 1:

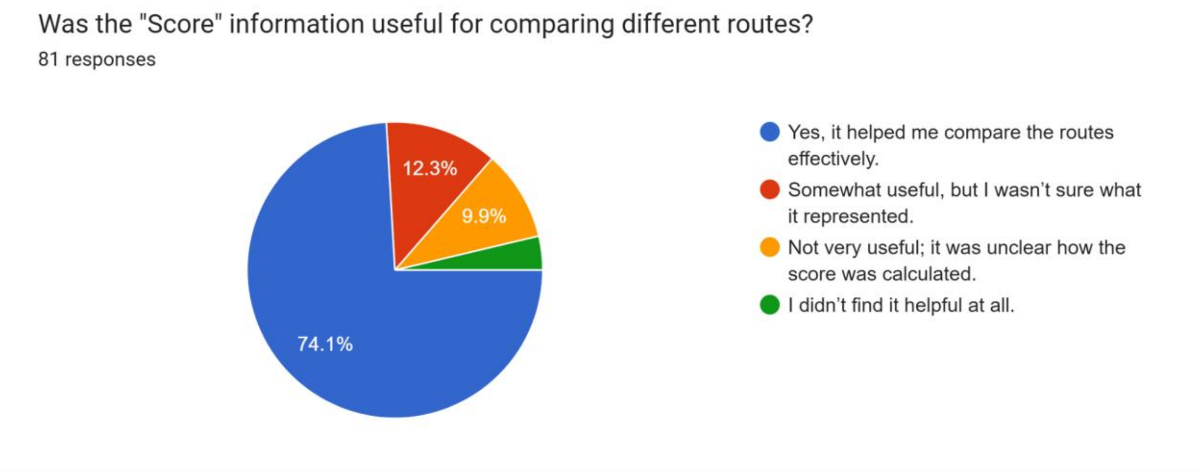


Was the information about each travel route clear and easy to understand?

81 responses

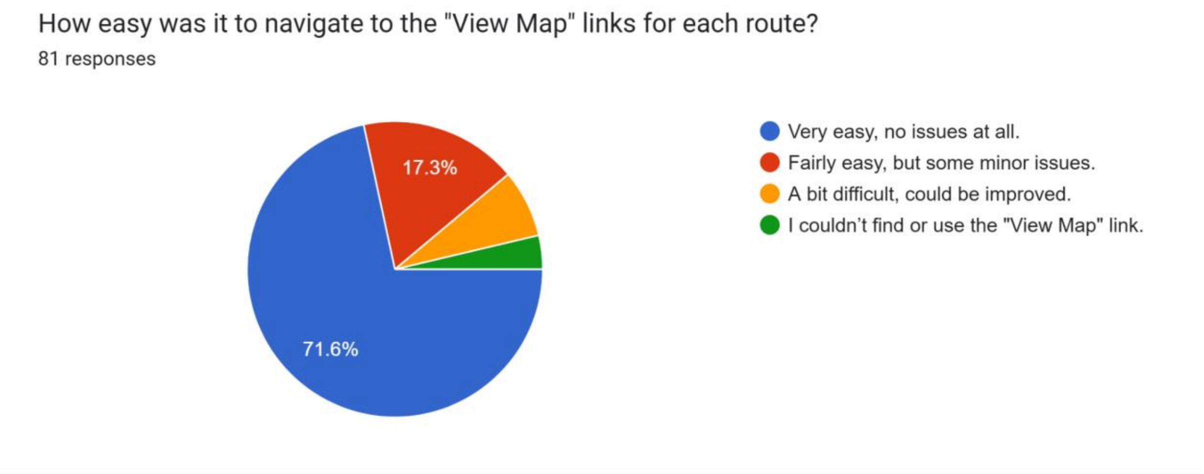
The survey results indicate that majority users (77.8%) found the travel route information very clear and straightforward, which is great. However, the remaining 22.2% faced some level of difficulty, with 17.3% suggesting minor improvements and a small percentage indicating confusion or finding the information hard to understand

Question 2:

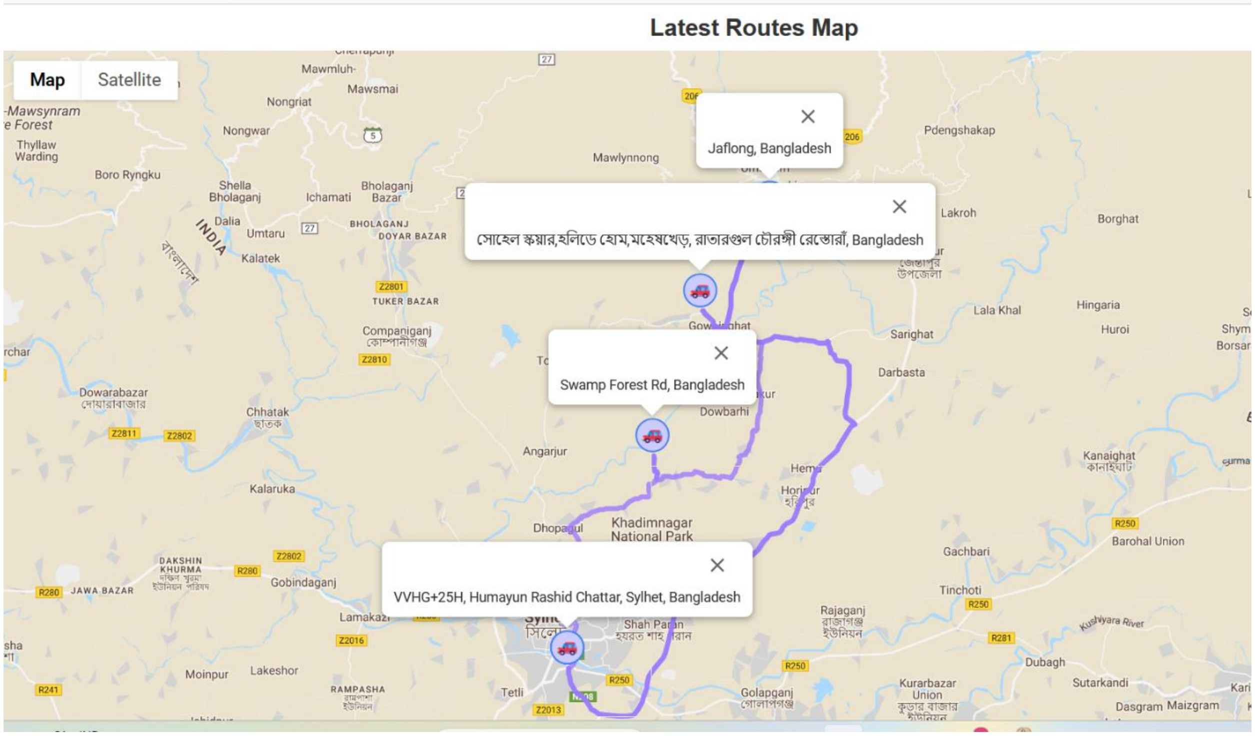


The survey, with 81 participants, evaluated the usefulness of "Score" information for comparing routes. The majority (74.1%) found it highly effective, while 12.3% found it somewhat useful but unclear. About 9.9% felt it lacked transparency in how the score was calculated, and 3.7% found it entirely unhelpful. Overall, most participants viewed the score as beneficial, but some highlighted issues with clarity and transparency.

Question 3:



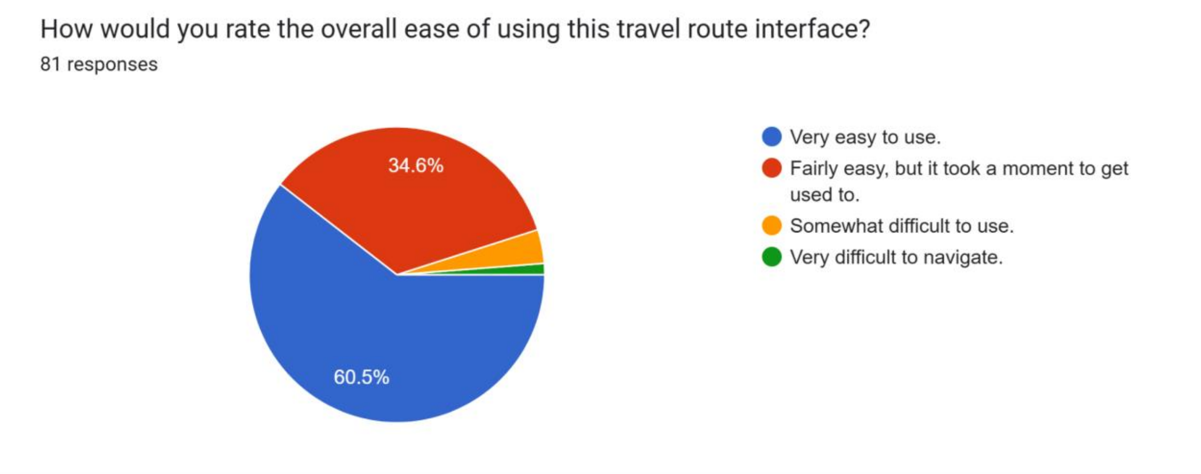
The survey evaluated how easy it was for 81 participants to navigate the "View Map" links for each route. Most participants (71.6%) found it very easy with no issues. Another 17.3% found it fairly easy but experienced minor issues. A smaller group (8.6%) reported some difficulty and suggested improvements, while 2.5% couldn’t find or use the link. Overall, the majority had a positive experience, though a few faced challenges.



Latest Routes Map

Map
Satellite
Jaflong, Bangladesh
সোহেল স্কয়ার,হলিডে হোম,মহেষখেড়, রাতারগুল চৌরঙ্গী রেস্তোরাঁ, Bangladesh
Swamp Forest Rd, Bangladesh
VVHG+25H, Humayun Rashid Chattar, Sylhet, Bangladesh
Khadimnagar National Park
Shah Paran
Chhatak
Companiganj
Dowarabazar
Kanaighat
Gachbari
Darbasta
Sarighat
Golapganj
INDIA

Question 4:

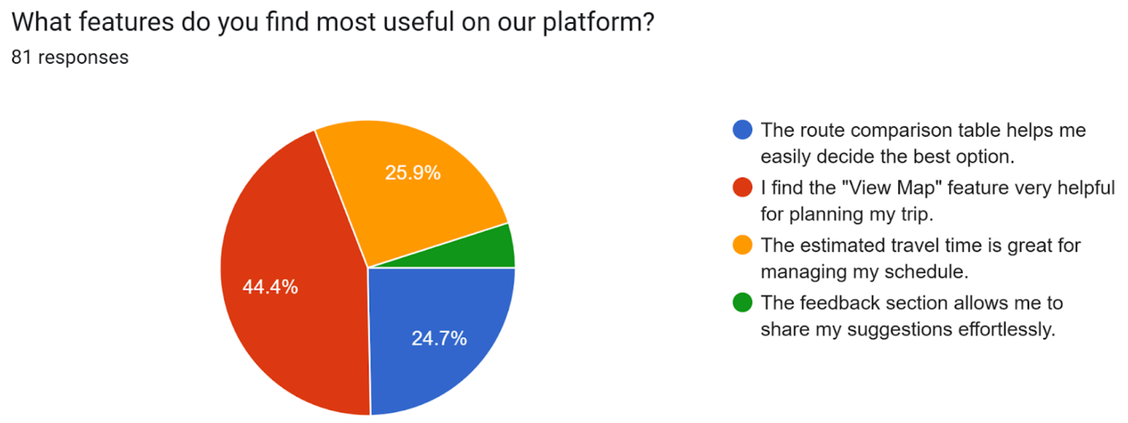


How would you rate the overall ease of using this travel route interface?

81 responses

The survey evaluated the overall ease of using a travel route interface among 81 participants. Most respondents (60.5%) found it very easy to use, while 34.6% found it fairly easy but needed a moment to get used to it. A small portion (3.7%) reported it was somewhat difficult to use, and only 1.2% found it very difficult to navigate. Overall, the majority had a positive experience with the interface.

Question 5:



What features do you find most useful on our platform?

81 responses

The survey explored the most useful features on a travel platform among 81 participants. The majority (44.4%) found the "View Map" feature highly helpful for trip planning. Around 25.9% highlighted the estimated travel time as valuable for managing schedules, while 24.7% appreciated the route comparison table for making informed decisions. A smaller group (5%) favored the feedback section for sharing suggestions. Overall, the results emphasize the importance of visual and time-saving tools in enhancing user experience.

User Feedback: Strengths and Areas for Improvement in the Travel App

Positive Feedback:

- Very good website.
- Everything is very easy and smooth and useful for finding the best travel destinations. User-friendly.
- It is easy to use.
- This system is very easy to use with a simple design. I wish to see more improvements.

- All the steps are good.
- Very positive.
- This app is really great for planning a trip.

Negative Feedback:

- Routes on the Map: The names of the tourist spots on the map were blocking the visibility of the routes, which made it difficult to see the paths clearly. A better layout could help ensure both the names and the routes are visible without obstruction.
- Final Route Selection: The use of a dropdown option for final route selection may cause confusion for users. It would be clearer if direct route options were provided, eliminating the need for the dropdown and making the selection process smoother.