TOUR ROUTE OPTIMISER USING TSP A HEURISTIC ALGORITHMIC APPROACH

CSD416 Project Phase II

CS20D04 CHN20CS006 Abin Skaria
CS20D32 CHN20CS056 Jais Tomy
CS20D47 CHN20CS093 Sahal Basheer
CS20D50 CHN20CS100 Shiwine K Sebastian
CS20D53 CHN20CS104 Souhrid Suresh
B. Tech. Computer Science & Engineering

Department of Computer Engineering
College of Engineering Chengannur
Alappuzha 689121
Phone: +91.479.2455125
http://www.ceconline.edu
hod.cse@ceconline.edu

2024

College of Engineering Chengannur
Department of Computer Engineering

C E R T I F I C A T E

This is to certify that, this report titled ***TOUR ROUTE OPTIMISER USING TSP A HEURISTIC ALGORITHMIC APPROACH*** is a bonafide record of the work done by **CS20D04 CHN20CS006 Abin Skaria, CS20D32 CHN20CS056 Jais Tomy, CS20D47 CHN20CS093 Sahal Basheer, CS20D50 CHN20CS100 Shiwine K Sebastian and CS20D53 CHN20CS104 Souhrid Suresh**, Eighth Semester B. Tech. Computer Science & Engineering students, for the course work in **CSD416 Project Phase II**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, B. Tech. Computer Science & Engineering of **APJ Abdul Kalam Technological University**.

Guide

Syeatha Merlin Thampi
Asst. Professor
in Computer Engineering

Coordinator

Leya G
Asst. Professor
in Computer Engineering

Head of the Department

May 13, 2024

Dr. Manju S Nair
Associate Professor
in Computer Engineering

Permission To Use

In presenting this project dissertation at College of Engineering Chengannur(CEC) in partial fulfillment of the requirements for an undergraduate degree from APJ Abdul Kalam Technological University, we agree that the libraries of CEC may make it freely available for inspection through any form of media. We further agree that permission for copying of this dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the Head of the Department of Computer Engineering. It is understood that any copying or publication or use of this dissertation or parts thereof for financial gain shall not be allowed without our written permission. It is also understood that due recognition shall be given to us and to CEC in any scholarly use which may be made of any material in this project dissertation.

Abin Skaria, Jais Tomy, Sahal Basheer, Shiwine K Sebastian, Souhrid Suresh

---

Statement of Authenticity

We hereby declare that this submission is our own work and to the best of our knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at College of Engineering Chengannur(CEC) or any other educational institution, except where due acknowledgement is made in the report. Any contribution made to our work by others, with whom we have worked at CEC or elsewhere, is explicitly acknowledged in the report. We also declare that the intellectual content of this report is the product of our own work done as per the **Problem Statement** and **Proposed Solution** sections of the project dissertation report. We have explicitly stated the major references of our work. We have also listed all the documents referred, to the best of our knowledge.

Abin Skaria, Jais Tomy, Sahal Basheer, Shiwine K Sebastian, Souhrid Suresh

Acknowledgements

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to **Dr. Manju S Nair**, Head of Department, Computer Science and Engineering, College of Engineering Chengannur for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to **Smt. Leya G**, Assistant Professor, Computer Science and Engineering department, College of Engineering Chengannur for the support and cooperation.

We would like to place on record our sincere gratitude to our project guide **Smt. Syeatha Merlin Thampi**, Assistant Professor, Computer Science and Engineering department, College of Engineering Chengannur for the guidance and mentorship throughout this work.

We would like to give proper credit to the authors of the main reference material, we have used for this project.

Finally we thank our family, and friends who contributed to the successful fulfillment of this work.

Abin Skaria, Jais Tomy, Sahal Basheer, Shiwine K Sebastian, Souhrid Suresh

Abstract

Tourism is a thriving industry that contributes significantly to the global economy. As tourists seek best experiences, the need for efficient itinerary planning becomes paramount. This project explores the application of the Traveling Salesman Problem (TSP) in the context of tourism to optimize tourist itineraries. The primary objective of this project is to develop a software that assists tourists in creating optimal travel routes by solving the TSP. In the context of tourism, this problem translates to finding the most efficient route that allows tourists to explore a destination's key attractions while minimizing travel time. The approach involves collecting and curating data on tourist attractions and distances. The TSP algorithm is then applied to this dataset to generate optimized itineraries. The software will provide users with the flexibility to customize their preferences, such as prioritizing certain attractions and accommodating time constraints.

Contents

List of Figures

Chapter 1

Introduction

Tourism Planner is an innovative and user-friendly platform designed to simplify and enhance the travel planning experience. Seamlessly integrating cutting-edge technology, it offers a dual-sided interface catering to both administrators and users. Administrators play a pivotal role by providing and updating essential travel data, while users leverage the system to access, customize, and visualize information through an interactive map. Facilitated by a central server, Tourism Planner fosters collaboration, ensuring real-time data synchronization and scalability. This dynamic platform empowers users to plan their journeys efficiently and contribute to the collective enrichment of the travel community.

1.1 Proposed Project

In the dynamic world of global tourism, our project redefines travel planning by addressing the challenges through the renowned Traveling Salesman Problem (TSP). This combinatorial optimization tool finds the shortest route and ensuring efficiency. Our sophisticated software seamlessly integrates comprehensive data on attractions, distances, utilizing the TSP algorithm to craft meticulously optimized itineraries. Designed for flexibility, users can customize their adventures, prioritizing attractions and managing time constraints.

1.1.1 Problem Statement

The project aims to develop a software using the Traveling Salesman Problem (TSP) to assist tourists in creating optimal travel routes. This software aims to improve the efficiency of exploring tourist attractions.

1.1.2 Proposed Solution

- **TSP Application:** Apply the TSP algorithm to optimize tourist routes. TSP aims to find the shortest route visiting locations exactly once and returning to the starting point.
- **Project Focus:** Develop a software as the primary project focus. Assist tourists in creating optimal travel routes by solving the TSP.
- **Combinatorial Optimization:** TSP is a well-known combinatorial optimization problem. It is adapted for tourism to minimize travel time.

- **Data Collection and Curation:** Collect and curate data on tourist attractions and distances.Input this dataset into the TSP algorithm for itinerary optimization.
- **User Flexibility:** Provide users with flexibility in customizing preferences.Options include prioritizing attractions and accommodating time constraints.
- **Software Benefits:** Enhance user experience by offering streamlined travel planning.Cater to the diverse needs of tourists seeking personalized and efficient travel routes.

Chapter 2

Report of Preparatory Work

2.1 Literature Survey Report

2.1.1 IntechOpen, Travelling Salesman Problem [1]

The idea behind TSP to consider a problem from the everyday life from a mathematical point of view. A traveling salesman has to visit exactly once each one of a list of m cities and then return to the home city. He knows the cost of traveling from any city i to any other city j. In this the problem of finding algorithmic technique leading to good/optimal solutions for TSP (or for some other strictly related problems) is considered. TSP is a very attractive problem for the research community because it arises as a natural subproblem in many applications concerning the every day life. Indeed, each application, in which an optimal ordering of a number of items has to be chosen in a way that the total cost of a solution is determined by adding up the costs arising from two successively items, can be modelled as a TSP instance. Thus, studying TSP can never be considered as an abstract research with no real importance.

2.1.2 On solving TSP by genetic algorithms [2]

Presenting a genetic algorithm for solving the traveling salesman problem using genetic algorithms to optimality for traveling salesman problems with up to cities. Proposed a genetic algorithm for the traveling salesman problem that generates very good but not optimal solutions for traveling salesman problems with and cities. Improved this approach by enhancing all basic components of that genetic algorithm. For our experimental investigations, used the traveling salesman problems TSP with cities, which were solved to optimality. Could solve medium-sized traveling salesman problems with up to cities in minutes average runtime on a SUN workstation. Furthermore, could solve traveling salesman problems with up to cities optimally in an acceptable time limit (e.g., the average runtime on a SUN workstation for the TSP is about minutes). The greatest examined problem with cities could be approximately solved by constructing a tour with a length percentage over the optimum.

2.1.3 An expert system for tourists using Google Maps API [3]

In the field of eTourism, it is important to present prominent objects of tourists destinations. Nowadays demand for eTourism applications is rising and the customers need rapid software development. Google Maps API is a technology provided by Google based on AJAX, which powers many map-based services. This paper presents an expert system for tourists. The realized software uses free, public API service from Google Maps. The system utilizes a knowledge base formed by tracking user actions. The expert module suggests information of special interest to the user.

2.1.4 Algorithm Studied

We studied Lin Kernighan Algorithm and Nearest Neighbor algorithm for implementing the routing functionality and how it can be implemented. Lin-Kernighan: The Lin Kernighan algorithm works by optimizing the initial path generated by swapping the edges for the given number of iterations. Nearest-Neighbor: The Nearest Neighbor algorithm offers a an approximate path by choosing the nearest path from the initial point in a timely manner.

2.2 System Study Report

In our comprehensive system study, we conducted a thorough examination of prominent platforms like google maps, Wanderlog, Tripadvisor, and also we explored open source mapping system like Open Street Map. We have analyzed how the algorithm works in google maps. Drawing inspiration from Wanderlog and TripAdvisor we created our UI in a user friendly way. Our main focus was algorithm implementation and we researched about OpenStreetMap, which provided free map data.

Chapter 3

Project Design

This chapter outlines the architectural blueprint for the Tourism Planner, emphasizing efficiency and user-friendliness. It covers web application architecture, hardware and software requirements, offering insights for developers and stakeholders. The clear presentation of design principles, with diagrams and detailed descriptions, lays the groundwork for effective implementation in subsequent development stages.

3.1 Web Application Architecuture: General

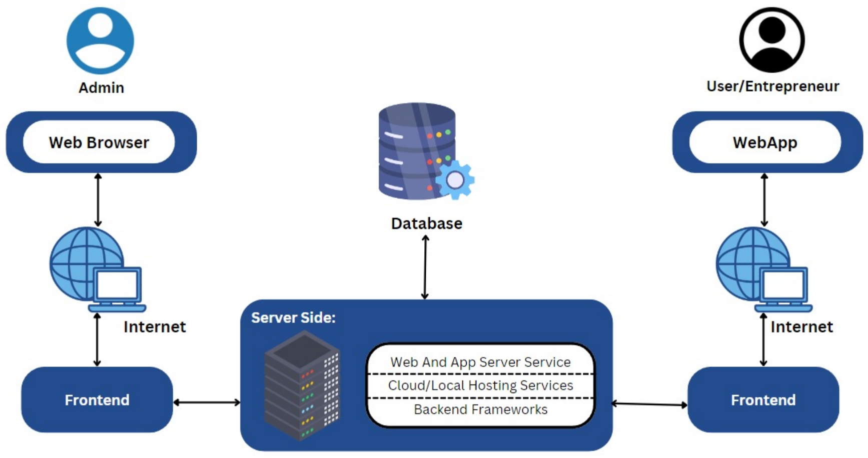


Figure 3.1: Web Application Architecture: General

The above diagram shows a web-based tourist information and planning system. Here's how it works:

Client Side:

- User/Entrepreneur: This section caters to two user types: tourists and entrepreneurs. Tourists can use the system to plan their trips, while entrepreneurs can manage their tourism-related businesses.

- Web Browser/Mobile App: Admin can access the system through a web browser on a computer and user access a mobile app on a smartphone or tablet and also through web browser.

Server Side:

- Database: This stores all the information needed for planning trips, such as tourist attractions, accommodation and reviews. It's likely a cloud-based database for scalability and accessibility.
- Web and App Server Service: This processes user requests, retrieves data from the database, and generates responses. It could use technologies like NodeJS or Python Flask.

Frontend/Backend:

- Frontend: This refers to the user interface that tourists and entrepreneurs interact with, likely built with frameworks like HTML, CSS and JS for the web and Flutter for mobile apps.
- Backend: This handles the behind-the-scenes logic of the system, such as data processing and communication with the database.

Additional components:

- Internet: This connects the user's device to the server.
- Cloud/Local Hosting Services: This refers to where the server and database are hosted. It could be a cloud platform like Amazon Web Services or Google Cloud Platform for scalability and reliability, or a local server for more control and customization.

Overall flow:

- User Interaction: Tourists browse attractions, accommodation options and facilities. Entrepreneurs manage their listings and update information.
- Data Retrieval: The system retrieves relevant data from the database based on user searches or filters.
- Information Presentation: The system presents the retrieved data to the user in a user-friendly format, such as lists, maps, or detailed pages.
- Booking/Management: Tourists can book tours, accommodation, or tickets through the system. Entrepreneurs can manage their bookings and update their listings.

Possible benefits:

- Convenience: Users can plan their entire trip in one place, from finding attractions to booking accommodation.
- Information access: The system provides a central hub for all tourism-related information, making it easy for users to find what they need.
- Business promotion: Entrepreneurs can reach a wider audience of potential customers through the system.

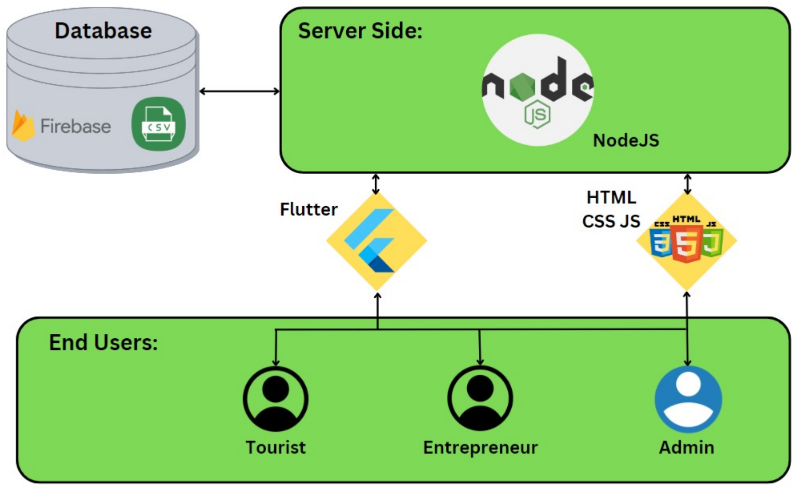


Figure 3.2: Web Application Architecture: Precise

3.2 Web Application Architecture: Precise to project

The system is divided into two main sections: the server-side and the end-user applications.

Server-side:

- Database: This stores all the relevant data about the user and the businesses using the firebase and CSV files for storing data about destinations.
- Server-side logic: This is where the system processes user queries and generates potential routes. It likely uses algorithms that consider factors like number of days and categories. The server-side logic implemented using NodeJS.

End-user applications:

- Flutter: This is a mobile app development framework, suggesting that the system may have a mobile app for tourists to use. The app would allow users to input their preferences and view suggested routes.
- Html, Css and Js: HTML defines the structure of your content, CSS determines the style and layout, and JavaScript makes the content interactive

Overall flow:

- User input: The user enters their preferences and desired travel dates into the mobile app.
- Route generation: The server-side logic processes the user input and queries the database to find potential routes that match the user's criteria.
- Route presentation: The system presents the suggested routes to the user on the mobile app. The routes may include information about attractions, and estimated travel times.

- User selection: The user selects a route or refines their preferences and generates new suggestions.

Additional components:

- Node: This could refer to a general processing unit or a specific technology like NodeJS.

3.3 Data-Flow Diagram

One of the key components of our project design is the Data Flow Diagram (DFD). The DFD provides a visual representation of the flow of data within our system. It helps us understand how the system interacts with external entities and how data moves through it.

3.3.1 Level-0

The Level 0 DFD, also known as a context diagram, gives a broad overview of the system. It illustrates how the system interacts with external entities.

- **System:** The Level 0 Data Flow Diagram (DFD) of the Tourism Planner system includes three key components: admin side, user side, and the central server. The server acts as the central hub, facilitating data flow between the admin and user sides and playing a crucial role in managing and storing information.

- **Admin-User Interaction:** Admins provide necessary data and updates to the server, ensuring the system's information remains accurate and up-to-date. Users access the system through the user interface, retrieving data from the server for travel planning purposes.

- **Server Functionality:** The server hosts the interactive map feature and manages data related to user contributions, such as adding enterprises or businesses. It enables collaboration, synchronization, and scalability, ensuring a seamless flow of information between admin and user sides while maintaining optimal system performance.

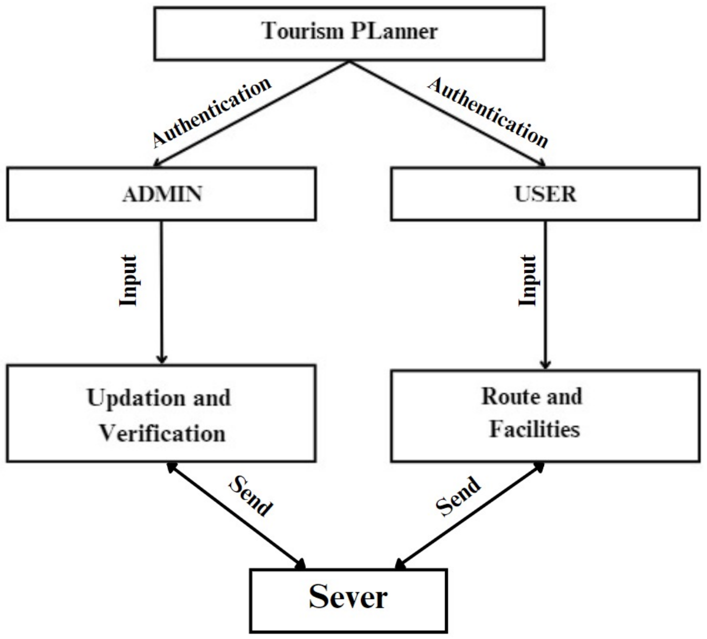


Figure 3.3: Data-Flow Diagram Level-0

This Level 0 DFD is a crucial part of our project design as it provides a high-level understanding of the system's processes and the flow of data. It aids in the clear communication of how the system is designed to work, making it an invaluable tool in our project design documentation.

3.3.2 Level-1

A Level 1 Data Flow Diagram (DFD) offers a detailed perspective of a system, breaking down major processes from the Level 0 DFD into sub-processes. Each subprocess is represented as a separate process, displaying associated data flows and data stores. This level provides a nuanced view, illustrating how data flows within the system and interacts with various entities. Essentially, it serves as an "exploded view" of the context diagram, offering insight into the main functions of the system. The choice of DFD level depends on the system's complexity and the desired level of detail, with higher levels providing a broad overview and lower levels delving into specific processes, data flows, and data stores. A combination of different DFD levels ensures a comprehensive understanding of the system

- **User Side Interaction:** Users can provide destinations they would like to visit through the system. The destination data is sent to the server for processing.
- **Optimal Path Calculation:** The server processes the user-provided destination data to generate an optimal path. The optimal path is then provided to the user.

- **Entrepreneur Interaction:** If the user is an entrepreneur, there is an option for them to add their business to the Tourism Planner.
- **Entrepreneur Data Submission:** Entrepreneurs can provide the necessary data required for adding their business to the Tourism Planner.
- **Admin's Role:** The role of the Admin is to verify the data submitted by users and entrepreneurs.
- **Data Verification and Update:** Admin verifies the data that needs verification. After verification, the Admin updates the verified data to the server.
- **Server Functionality:** The server acts as a central hub for processing user-provided destinations and managing added businesses.

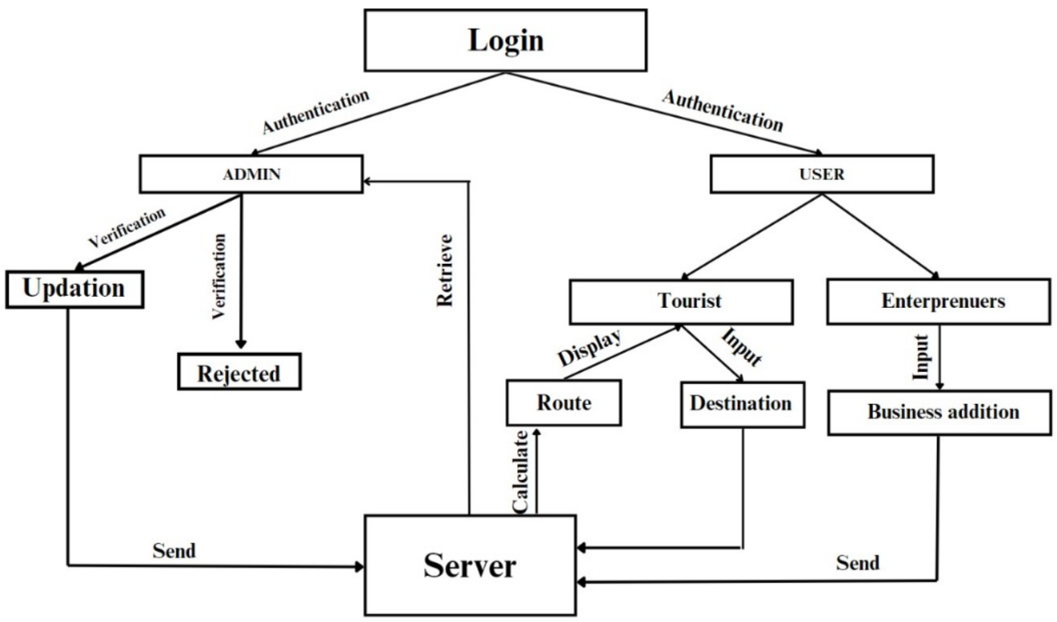


Figure 3.4: Data-Flow Diagram Level-1

3.3.3 Level-2

- **Modules in Tourism Planner:** The Tourism Planner comprises three main modules: user, admin, and server.
- **Authentication Process:** Authentication is used to verify whether a person is a user or an admin.
- **User Module:** Users can utilize the Tourism Planner for tour planning. To plan a tour, users provide their starting destination and places they would like to visit.

- **Data Flow to Server:** User-provided tour data is sent to the server for processing.
- **Optimal Path Calculation:** The server processes the user's data and returns an optimal path to the user.
- **Entrepreneur Interaction:** Users who want to add a business to the Tourism Planner need to submit necessary data.
- **Business Data Submission:** Entrepreneurs provide required data along with specifying the category for their business.
- **Data Flow to Admin for Verification:** The submitted data is sent to the admin side through the server for verification.
- **Admin's Responsibilities:** Admin verifies the authenticity of the submitted businesses. Admin is also responsible for adding new tourist-friendly places and making changes to the map.
- **Centralized Server Role:** The server acts as a centralized component for processing user requests, verifying business data, and managing additions to the map.

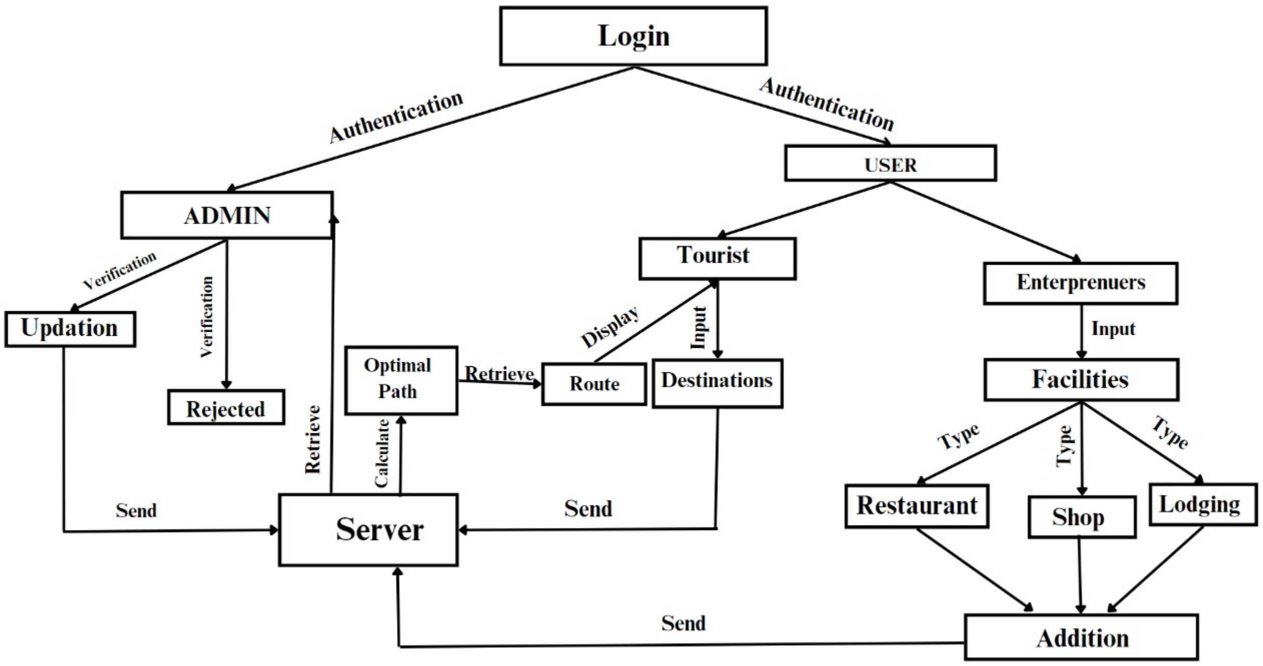


Figure 3.5: Data-Flow Diagram Level-2

3.4 Resource Requirements

3.4.1 Hardware & Software Requirements

Hardware:

- Smartphone(with android 8 or more)
- Laptop(processor i5 or greater)
- RAM 8gb or more
- Hard-disk 160gb or above

Software:

- Development tools such as Visual Studio Code (VSCode) for coding and project management.
- Fronted framework - Flutter, Html Css and Js
- Backend development framework used for web and application development - NodeJS
- Database Management System used for data storing and retrieval -Firebase and CSV
- Web hosting platform provided through the GitHub Student Pack or from College server.

3.4.2 Data Requirements

The system requires entrepreneur to provide their Business name, type of business and location.

3.5 Database Design

The foundation of our forum system lies in a well-thought-out database design, visualized through an Entity-Relationship (ER) Diagram. This diagram serves as a crucial guide, offering a comprehensive overview of the data entities and relationships within our system. It acts as a blueprint, illuminating the structure of our database and facilitating a clear understanding of data flow and dependencies.

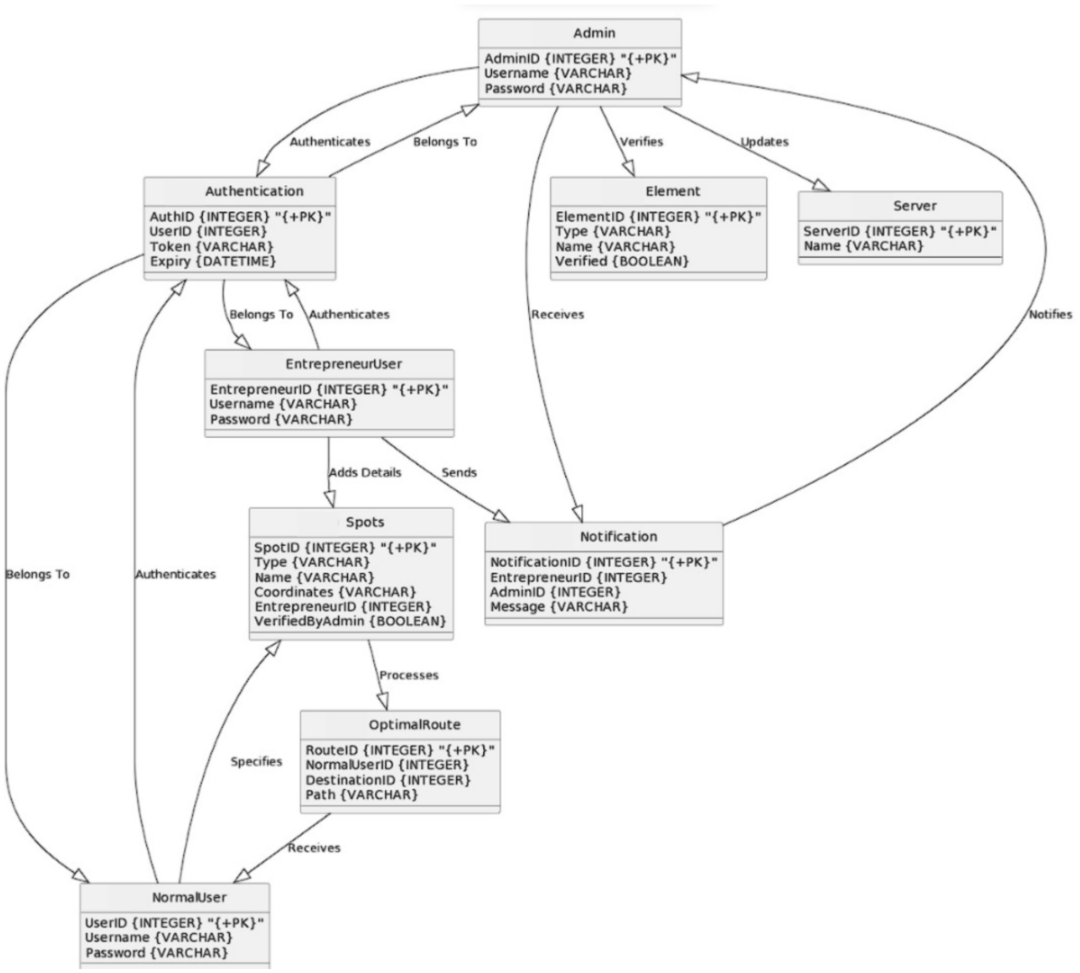


Figure 3.6: ER Diagram

Entities: These represent the fundamental building blocks of our system, each playing a distinctive role:

- **Admin:** This entity represents an administrator who has the authority to manage and oversee the system. Admins typically have privileges to perform tasks such as user management, data verification, and system configuration.
- **Authentication:** The authentication entity stores information related to user authentication, such as login tokens, expiry dates, and associated user IDs. This data is crucial for verifying the identity of users accessing the system.
- **EntrepreneurUser:** This entity represents users who are entrepreneurs or business owners using the system. They may have specific roles and permissions tailored to their needs, such as adding business locations or managing their profiles.
- **Spots:** The spots entity represents locations or points of interest within the system. It includes details such as the spot's type, name, coordinates, the user who added it, and

whether it has been verified by an admin for accuracy.

- **Element:** The element entity represents individual data elements within the system. These could be various types of information stored in the database, each identified by a unique ID and categorized by name and type.
- **Server:** This entity represents a server within the system architecture. Servers play a crucial role in hosting and managing the system's data and services, ensuring smooth operation and accessibility for users.
- **Notification:** The notification entity stores information about notifications sent within the system. It includes details such as the message content, the IDs of the admin and user involved, and other relevant data for tracking and managing notifications.
- **OptimalRoute:** This entity represents the optimal route feature within the system, which calculates and stores the best route for users to reach their destination. It includes details such as the user ID, destination ID, and specific path information for navigation.
- **NormalUser:** The normal user entity represents regular users of the system who may not have administrative or business-related roles. These users typically interact with the system for tasks such as browsing spots, receiving notifications, and accessing services.

  Each element in the ER Diagram plays a specific role in defining the structure and functionality of the system, facilitating interactions between users, data, and system components to support its overall operation and user experience.

Relationships: The connections define the interplay between entities, elucidating the dynamics of our system:

3.6 Algorithms Implementation

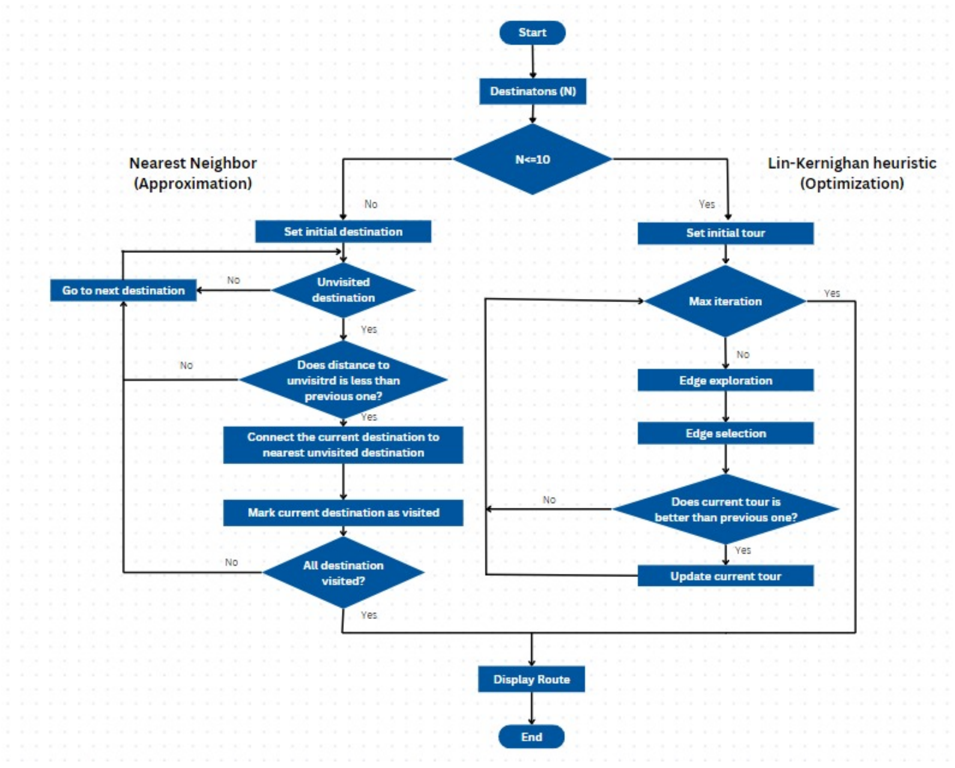

Start
Destinatons (N)
N<=10
Nearest Neighbor (Approximation)
Lin-Kernighan heuristic (Optimization)
No
Yes
Set initial destination
Set initial tour
Go to next destination
Unvisited destination
Does distance to unvisitrd is less than previous one?
Connect the current destination to nearest unvisited destination
Mark current destination as visited
All destination visited?
Max iteration
Edge exploration
Edge selection
Does current tour is better than previous one?
Update current tour
Display Route
End

Figure 3.7: Algorithm Work Flow

3.6.1 Nearest Neighbor

```
Main Algorithm (nearestneighbor):
Start with an initial city (can be chosen randomly).
Create an empty path.
Mark the initial city as visited.
While there are unvisited cities:
Find the nearest unvisited city from the current city.
Add this city to the path.
Mark this city as visited.
Update the current city to the newly visited city.
Add the starting city to the end of the path to complete the cycle.
Return the generated path.

Main Function (main):
Initialize coordinates and paths.
Print the given coordinates.
Apply Nearest Neighbor algorithm to find the path.
```

```
Calculate the total distance of the generated path.
Print the generated path and its total distance.
Plot the generated path.
```

3.6.2 Lin-kernighan

```
Main Algorithm (linkernighan):
Start with an initial path, which can be generated randomly.
Begin the loop for optimization:
Initialize a flag improved as True.
While improved is True:
Set improved to False.
Iterate through the path:
Consider all possible swaps of subsequences in the path
check if the resulting path is better than the current path.
If an improvement is found, update the path and set improved to True.
Repeat until no more improvements can be made.
Return the optimized shortest path.

Main Function (main):
Initialize coordinates and paths.
Print the given coordinates.
Generate an initial random path.
Calculate the initial total distance of the path.
Apply Lin-Kernighan algorithm to find the shortest path.
Calculate the shortest total distance of the optimized path.
Print the shortest path and its total distance.
Plot the optimized shortest path.
```

3.7 Work Schedule

- **October to December (2023):** Get Familiarized with Google Maps API features. Analyzing various approximation and optimization algorithms related to tsp. Review and understand the best algorithm for implementation. Begin designing the project, including conceptualization and initial UI/UX planning.
- **January to April (2024):** Get Familiarized with ReactJs, Flutter and NodeJs. In-depth learning of Python. Commence the project development phase. Work on user interface development and Implementation of data structure and algorithm visualization. Address any technical challenges that arise during development. Then we enter the debugging and testing phase.

Chapter 4

Implementaion

The Tour Route Optimizer using TSP has made 95% of its development objectives. Achieving its goal of creating a user friendly platform for efficient route planning using TSP heuristic algorithms, tour planning and business management.
Our app focuses on routing algorithms Lin Kernighan(optimization) and Nearest Neighbor(approximation) for implementing the routing functionality, providing the suitable path.

4.1 Development Progress

4.1.1 Installation of Node Modules

In order to implement Back-end server functionality node modules are used.

4.1.2 Front-end Development

Front-end development of the tourism route planner has been completed using HTML, CSS and JavaScript. Home page, route planning and business pages have been crafted providing intuitive user interface for route planning and business management. We used the Leaflet library an open source library with mapping capabilities of openstreetmap for the maps used throughout the web-pages.

4.1.3 Back-End Development

Firebase serves as the back-end infrastructure to verify and manage user accounts. NodeJS is used to implement server functionality. Back-end development has been implemented successfully.

4.1.4 APIs

We used Mapbox API for calculating distances and for path selection based on real-world road data.

4.1.5 Testing

Tested the route planner by providing various inputs to find any faulty cases and bugs also ensured, the services are working properly.

4.1.6 Hosting on Cloud Server

We used Glitch for hosting the tourism route planner to make it available globally.

4.1.7 App Development

We have embedded the flexible website with Flutter for the development of the application.

4.2 User Interface Design

User interface is designed in such a way that the system tends to be more user friendly.

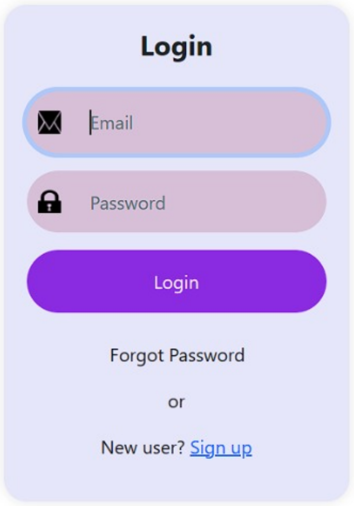
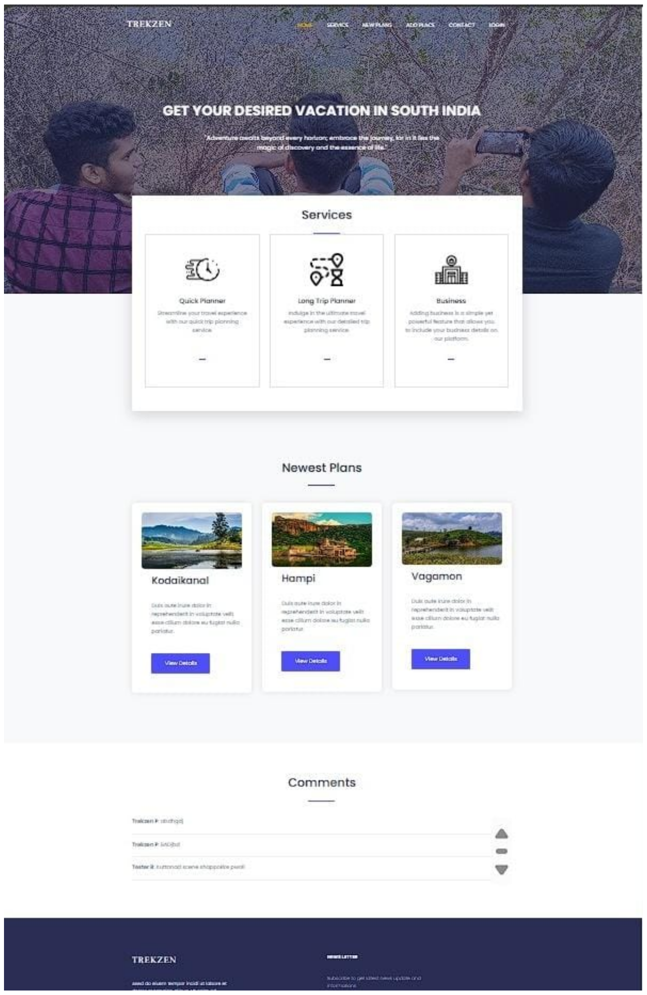


Figure 4.1: Login Page

Figure 4.1 shows the login page for user. User can login by providing their email and password.



GET YOUR DESIRED VACATION IN SOUTH INDIA

"Adventure awaits beyond every horizon; embrace the journey, for in it lies the magic of discovery and the essence of life."

Streamline your travel experience with our quick trip planning service.

Indulge in the ultimate travel experience with our detailed trip planning service.

Adding business is a simple yet powerful feature that allows you to include your business details on our platform.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Trekzen P: abdhgdj

Trekzen P: SADjbd

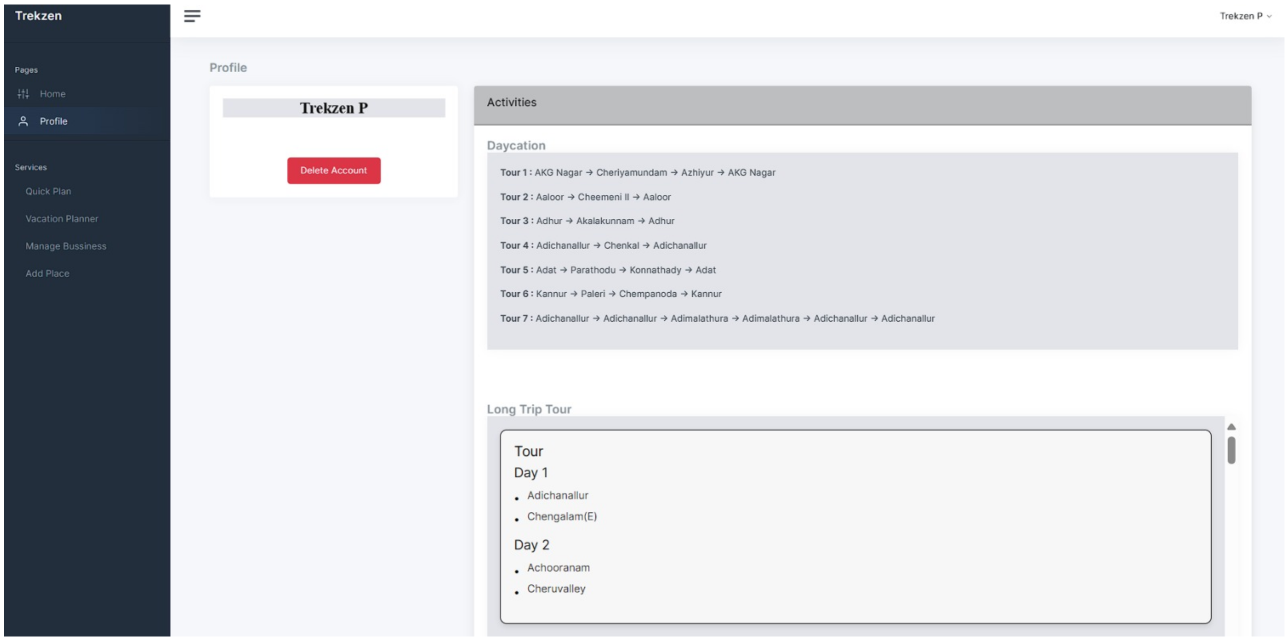
Tester ll: [illegible]

sed do eiusm tempor incidi ut labore et

Subscribe to get latest news update and informations

Figure 4.2: Home Page

Fig 4.2 shows the home page of Trekzen where user could view the services provided by the application. User can also view the comments of the users.



Daycation

Tour 1 : AKG Nagar → Cheriyamundam → Azhiyur → AKG Nagar

Tour 2 : Aaloor → Cheemeni II → Aaloor

Tour 3 : Adhur → Akalakunnam → Adhur

Tour 4 : Adichanallur → Chenkal → Adichanallur

Tour 5 : Adat → Parathodu → Konnathady → Adat

Tour 6 : Kannur → Paleri → Chempanoda → Kannur

Tour 7 : Adichanallur → Adichanallur → Adimalathura → Adimalathura → Adichanallur → Adichanallur

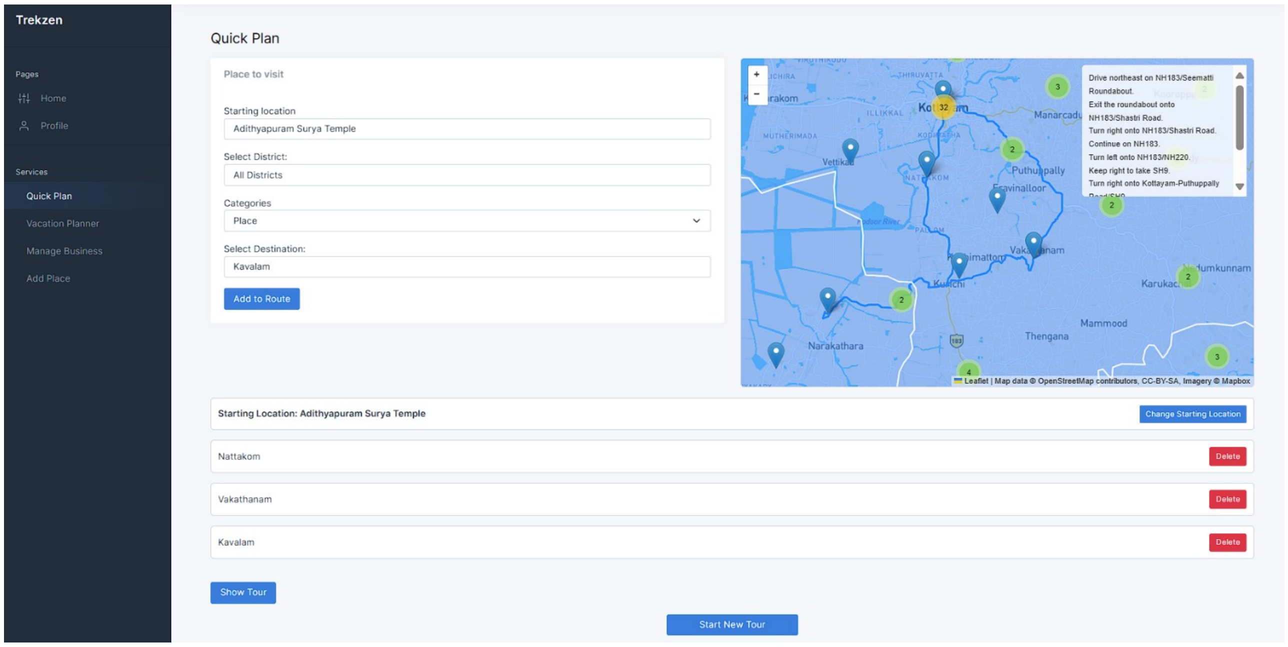
Long Trip Tour

Tour
Day 1
- Adichanallur
- Chengalam(E)

Day 2
- Achooranam
- Cheruvalley

Figure 4.3: Profile View

Figure 4.3 shows the profile view of user, which dispays the past tours of user both quick plan and vacationPlan.



Quick Plan

Place to visit

Starting location: Adithyapuram Surya Temple

Select District: All Districts

Categories: Place

Select Destination: Kavalam

Drive northeast on NH183/Seematti Roundabout.
Exit the roundabout onto NH183/Shastri Road.
Turn right onto NH183/Shastri Road.
Continue on NH183.
Turn left onto NH183/NH220.
Keep right to take SH9.
Turn right onto Kottayam-Puthuppally Road/SH9

Starting Location: Adithyapuram Surya Temple

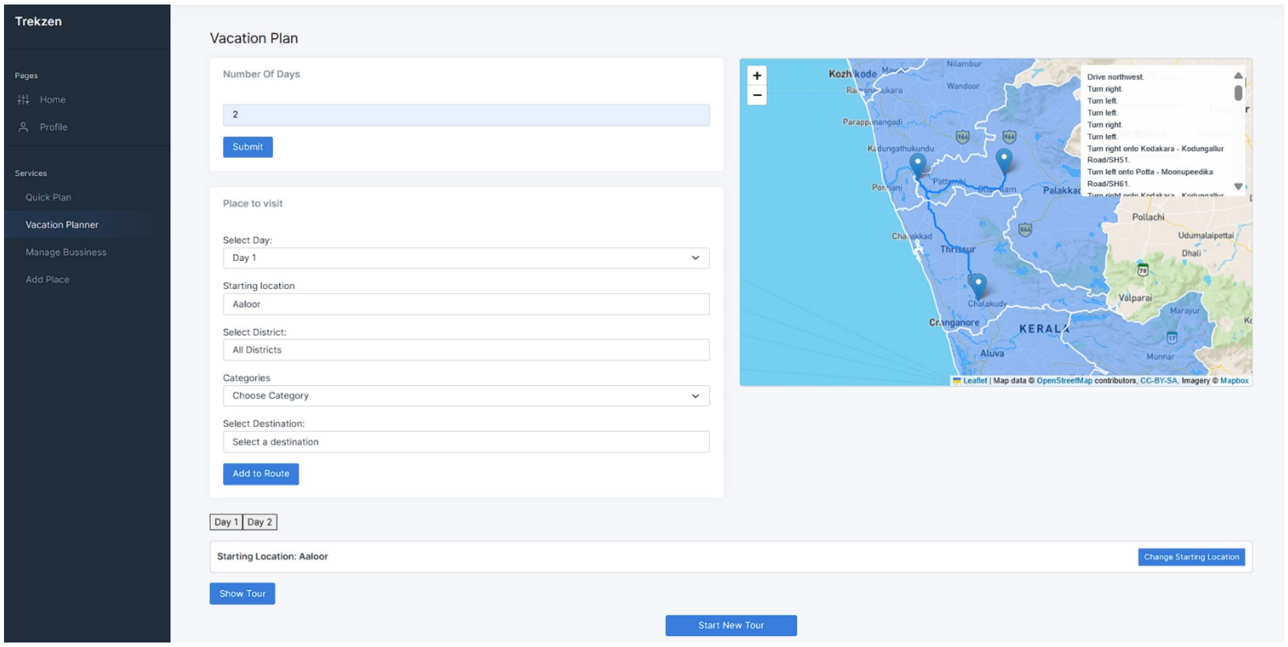
Nattakom

Vakathanam

Kavalam

Figure 4.4: Quick Plan Page

Figure 4.4 shows Quick plan page, where user can choose their destination and plan their tour.



Vacation Plan

Number Of Days: 2

Place to visit

Select Day: Day 1

Starting location: Aaloor

Select District: All Districts

Categories: Choose Category

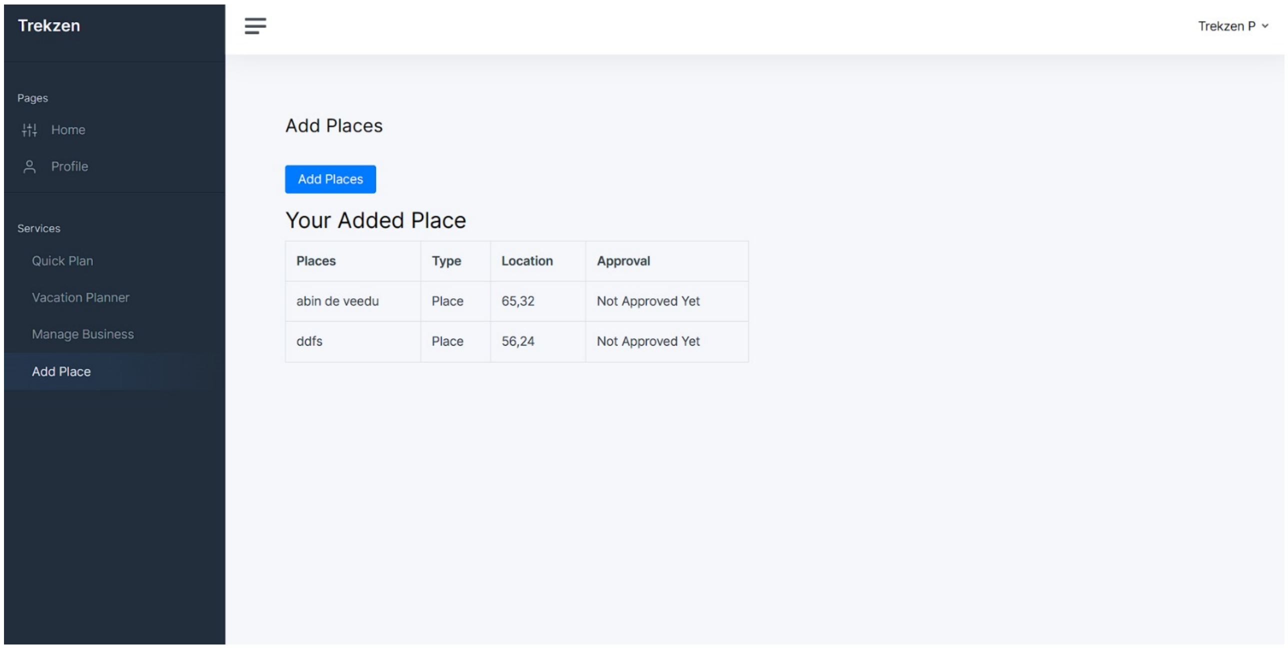
Select Destination: Select a destination

Drive northwest.
Turn right.
Turn left.
Turn left.
Turn right.
Turn left.
Turn right onto Kodakara - Kodungallur Road/SH51.
Turn left onto Potta - Moonupeedika Road/SH61.

Starting Location: Aaloor

Figure 4.5: Vacation Page

Figure 4.5 shows Vacation plan page, where user can plan their vacation of multiple days to destinations.

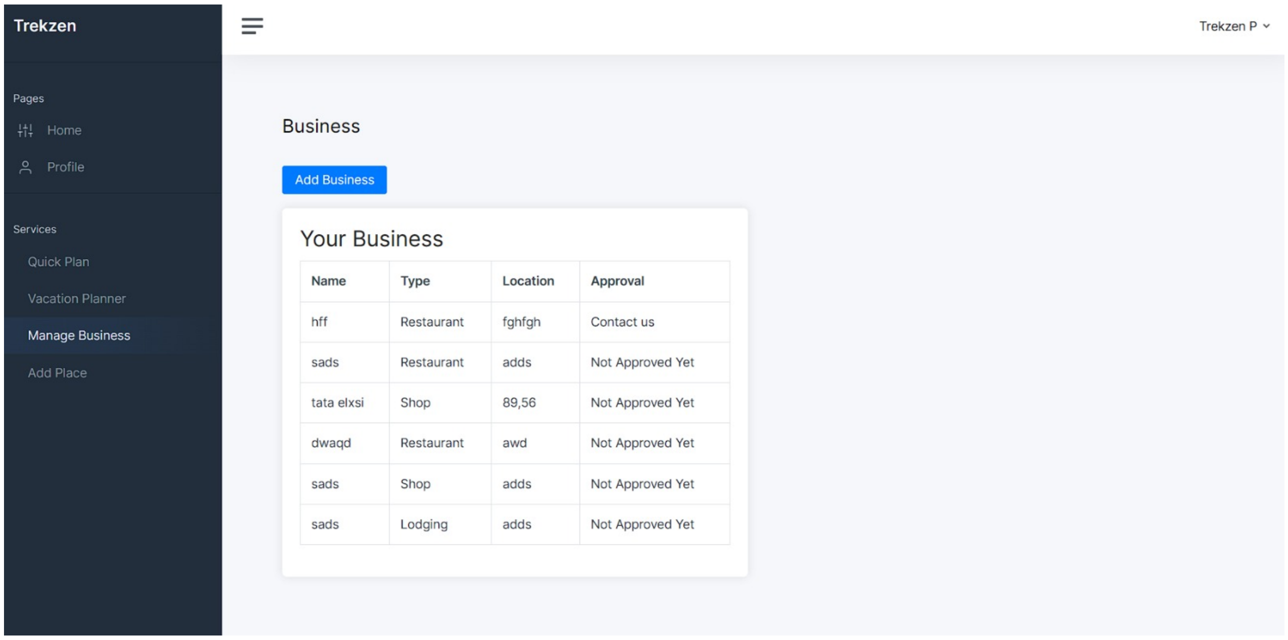


Your Added Place

Places	Type	Location	Approval
abin de veedu	Place	65,32	Not Approved Yet
ddfs	Place	56,24	Not Approved Yet

Figure 4.6: Add Place Page

Figure 4.6 shows the places suggested by user to admin and their current status after adding.



Your Business

Name	Type	Location	Approval
hff	Restaurant	fghfgh	Contact us
sads	Restaurant	adds	Not Approved Yet
tata elxsi	Shop	89,56	Not Approved Yet
dwaqd	Restaurant	awd	Not Approved Yet
sads	Shop	adds	Not Approved Yet
sads	Lodging	adds	Not Approved Yet

Figure 4.7: Manage Business Page

Figure 4.7 shows the business management page where user can add their businesses and there approval.

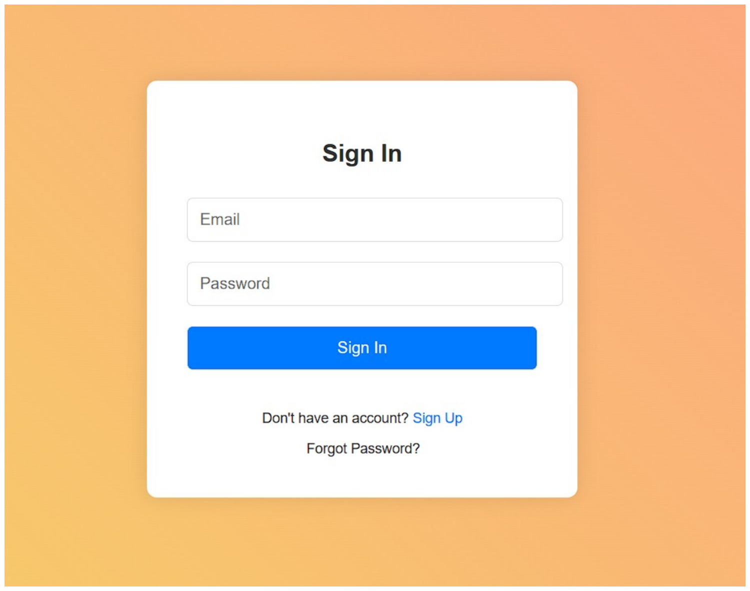
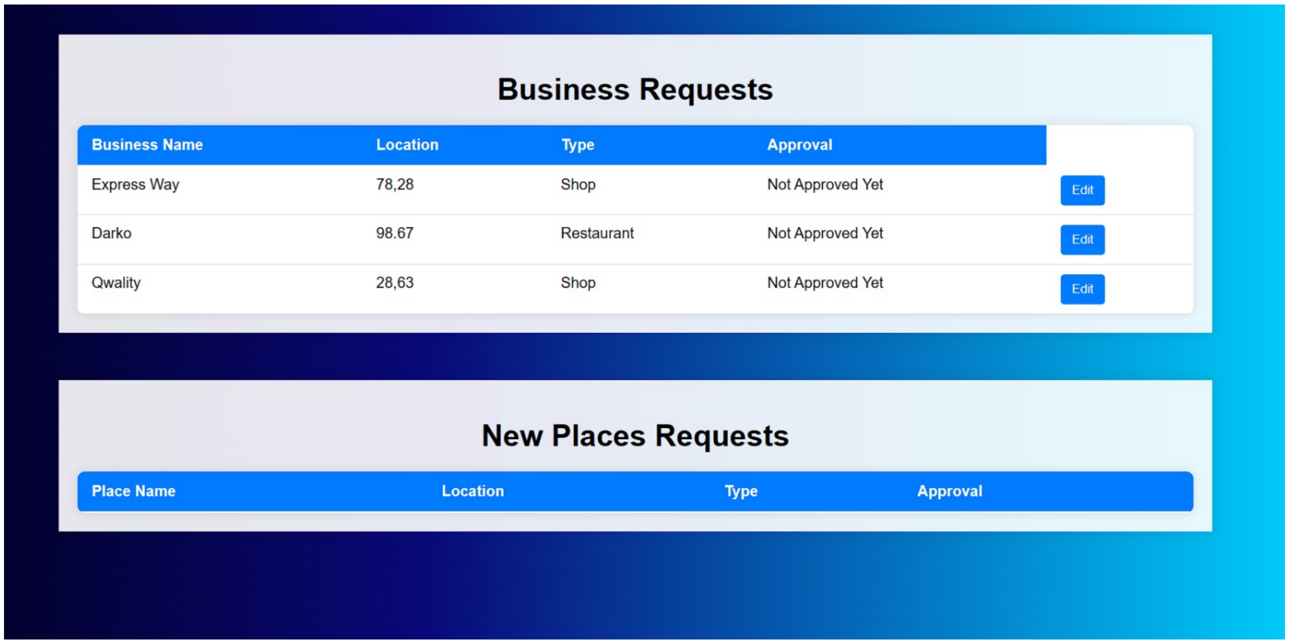


Figure 4.8: Admin Login Page

Figure 4.8 shows the login page for admins website. Admin can sign in by providing there Email and password.



Business Requests

Business Name	Location	Type	Approval	
Express Way	78,28	Shop	Not Approved Yet	Edit
Darko	98.67	Restaurant	Not Approved Yet	Edit
Qwality	28,63	Shop	Not Approved Yet	Edit

New Places Requests

Place Name	Location	Type	Approval

Figure 4.9: Admin Main Page

Figure 4.9 shows the Admin page where various request from user appear. It includes both the business and add place request.

Chapter 5

Results & Conclusions

The development and implementation of Tour route optimizer using TSP heuristic algorithms have been successfully completed offering users a convenient interface to plan their vacations, business management and weekend getaways. Through the utilization of app and website, it ensures a seamless user experience with robust functionality.

It provides users with a user-friendly interface, allowing them to easily plan their travels, adding and managing businesses. Users can add and search various categories of places and facilities.

The crux of tour route optimizer is the approximation(Nearest Neighbor) and optimization (Lin-Kernighan)algorithms which are used to appropriately find and provide the shortest possible routes from the places provided by the user. The algorithms are switched based on the number of inputs provided by the user in-order to reduce the time lag and providing the result in a timely manner.

Overall the successful completion of the project signifies a significant milestone in the implementation of heuristic algorithms in the tourism and travel route planning.

References

[1] Federico Greco, "IntechOpen, Travelling Salesman Problem" 2008 DOI:10.5772/66, ISBN:978-953-7619-10-7

[2] Heinrich Braun "On solving travelling salesman problems by genetic algorithms" 01 January 2005 Institut fiir Logik, Komplexitht und Deduktionssysteme, Universi t Karlsruhe Posffach 6980, D 7500 Karlsruhe, Deutschland, e-maih braun@ira.uka.de

[3] Aleksandar Pejic; Szilveszter Pletl; Bojan Pejic "An expert system for tourists using Google Maps API" 2009 7th International Symposium on Intelligent Systems and Informatics DOI: 10.1109/SISY.2009.5291141

[4] Wanderlog: Travel Planning Website, [online] ,Available: https://wanderlog.com/home (Accessed: December 1, 2023)

[5] TripAdvisor: Travel Planning Website, [online] ,Available: https://www.tripadvisor.in/ (Accessed: December 1, 2023)