Lazy Wanderer – A Smart Travel Planner

Challa Ashok Kumar Reddy, Chepala Shaik Shoaib Basha, Madhuchetan, P S Lalith Sahan
Department of Computer Science REVA University Karnataka, India

***Abstract*— Planning a trip is normally associated with dealing with many aspects of the process at once, such as choosing destinations, estimating budgets, scheduling itineraries, among others. Traditionally, people have to go across several interfaces and manually manage everything which leads to inefficiencies and inability to be personalized. This paper aims at developing a solution which would simplify this process by means of designing and presenting "Lazy Wanderer," the Smart Travel Planner. The proposed system is implemented using the architecture that consists of the client (the front-end) and the server (the back-end). The client-side is developed using Angular. The back-end utilizes Spring Boot with MySQL. Using the developed system, one can plan his/her trips in a way that takes into account individual preferences concerning budget, length, and destinations. Furthermore, it suggests routes, manages expenses during the trip, and considers real-time information like weather predictions to help with better decision-making. The suggested system implements the client-server paradigm, where requests from users are processed by the back-end, combined with pre-stored and actual data. It then returns results in the form of itineraries. According to preliminary evaluation, this approach helps in improving planning effectiveness and user experience. The developed Lazy Wanderer Smart Travel Planner shows the potential of applying contemporary web technologies and intelligent data processing algorithms to develop practical solutions for smart tourism applications. Possible extensions may include adding machine learning techniques, enabling mobile platforms' support, and integrating booking functionality.**

I. INTRODUCTION

Previously, people could not simply travel. First of all, they had to spend a considerable amount of time researching, as everything needed was scattered across various websites. Therefore, travelers would have to choose hotels, tourist spots, estimate costs, and calculate the travel route. In addition to consuming the significant amounts of time, such a fragmented approach is resulted in a poorly developed itineraries along with the insufficient budget management and also a poor personalization.

Due to web-based solutions and intelligent systems, traveling has become easier and more efficient than ever before. However, despite the large number of applications for travel planning, the vast majority of currently available applications and platforms still suffer from a number of drawbacks, among which include inflexibility, insufficient personalization and adaptation abilities, as well as inability to integrate and work with heterogeneous sources.

The problem becomes even more complicated upon acknowledging that travel planning itself is a multi-constraint decision-making problem. Specifically, a traveler has to optimize several factors at once, including costs, timing, personal preferences, and logistic issues. Existing research on travel recommendation systems reveals the importance of utilizing heterogeneous sources of data along with the advanced analysis techniques.

Typically, the traditional approach to travel recommendations are associated with rule-based reasoning and fixed databases, which results in inefficiency and inability to meet customers' needs. Consequently, the objective of existing intelligent systems is to facilitate travel planning by considering individual preferences and building optimal itineraries. However, at present, a number of AI-based techniques remain conceptual, with no practical application within full-stack travel recommendation systems. The "Lazy Wanderer" system will bridge the gap between theoretical knowledge and practice by integrating sophisticated techniques of intelligent travel planning into a robust full-stack web application architecture.

In particular, the angular-based front end of the app has an intuitive interface, allowing users to input the necessary information regarding the destination, budget, planned time of travel, personal preferences, and so on. Then, the back end receives such input and performs all calculations and logical actions. Meanwhile, the back end communicates with the MySQL database through the repository layer in order to provide rapid retrieval and storage of relevant data. Such architecture simplifies further expansion and scalability of the application significantly.

Perhaps, the strongest feature of the Lazy Wanderer system is the usage of data. The system is designed for managing various types of information like user details, trips, itineraries, hotels, attractions, routes, preferences, and service costs, as shown in the class and ER diagrams. By arranging this data in a structured way makes easy to store, access, and manage all features efficiently. The structured method also supports the development of advanced features like personalized recommendations and accurate budget tracking.

In the current world, where personalization playing a key role, this system focuses on customization of travel plans to individual users. It also saves details about the user preferences, include their likes, dislikes, user interests, and budget restrictions. Using this information, the system can able to generate customized timetable that better match the each user's needs, making the overall travel planning experience more relevant and user-friendly.

II. LITERATURE SURVEY

Smart travel planning has become very popular these days because people prefer trips that are more personalized and efficient. Many researchers have been suggested different approaches using AI, recommendation systems, and optimization techniques to make itinerary planning automatic and improve user experience. Even though there has been good progress, there are still some limitations in current systems, especially in real-life usage, combining different features together, and adjusting to changing situations.

A. Travel Planning Systems Based on AI

In recent years, researchers have focused more on using artificial intelligence to simplify travel planning. AI-based systems are capable to study user preferences, past data, and current conditions to generate travel plans that match individual needs. This reduces manual work and makes decision-making easier for users. For example, recommendation systems consider user interests, budget, and other factors to suggest suitable travel options. Research also shows that machine learning models can be used to automatically generate itineraries. These models are helpful in selecting destinations, planning routes, and estimating costs. One advantage is that these systems can adapt in real time by considering factors like weather changes and availability, which makes the planning process more flexible and useful.

But most of these systems are mostly theoretical or algorithm-based, and they don't work with the full-stack applications. The proposed Lazy Wanderer system, is on the other hand, filling this gap by combining smart planning ideas with a real-world web-based implementation that uses Angular, Spring Boot, and MySQL.

B. Systems for Personalized Recommendations

Recommendation systems are very important for planning the trips because they help with the problem of too much information. Collaborative filtering and content-based filtering are two ways that traditional systems give suggestions. More advanced methods uses hybrid models that mix different techniques to make things more accurate and tailored to the each user. Studies show that today's recommendation systems take into account a number of factors, including user preferences, travel history, budget, and where the user is. These systems make users happier by giving them personalized travel experiences.

Even with these much improvements, many current systems only care about how accurate their recommendations are and they don't include other important features like managing expenses, structuring itineraries, and getting real-time updates. The Lazy Wanderer system builds on these methods by combining them all into a full trip management framework that connects user preferences with trip creation, itinerary planning, and budget tracking.

C. Suggestions for and improvements to your itinerary

Creating an itinerary is a difficult task which requires choosing and arranging Points of Interest (POIs) while taking into account many factors, such as time, cost, and user preferences. Some research studies utilize the optimization techniques to model this problem, including the Traveling Salesman Problem (TSP) and the Orienteering Problem (OP).

New developments include interactive systems that allows users to change their travel plans on the fly based on what they want. Also, clustering algorithms and deep learning methods are used to group destinations and make travel plans that work well.

But current systems often:
• Put a lot of weight on optimization algorithms
• Don't worry about practical things like keeping continuous
track of costs
• Not being able to work with real-time data

The Lazy Wanderer system fills all these gaps by combining itinerary planning along with real-time data integration and financial tracking. This makes it more useful and user-friendly.

D. Finding the best route and navigation

Another important part of travel planning is finding the best routes. Researchers working on this by using methods like shortest-path algorithms, graph models, and optimization techniques. The main aim is to reduce travel time while still giving users a good and comfortable experience. These methods helps in deciding the most efficient way to move from one place to another during a trip.

Recent studies also highlight the importance of using real-time data and different types of transport, like buses, trains, or cars, in route planning to travel. However, many of these systems only work well in theory or simulation and are not very useful in the real-life situations accurately. In contrast, the Lazy Wanderer system uses the Google Maps API to find the best ever routes in real time, which makes it more practical and accurate for actual travel.

E. Systems that work in real time and are aware of their surroundings

Modern travel planning systems are using more and more real-time data, like weather reports, traffic updates, and service availability. AI-based systems use all of these factors to make suggestions that are tailored to the situation. Studies show that being able to adapt in real time greatly improves the user experience and the feasibility of the itinerary. But a lot of systems don't do a good job of combining real-time data with recommendation and optimization systems.
The suggested system solve this problem by combining:
• Services for optimizing routes
• APIs for the weather
• APIs for travel data

F. New trends, patterns and advanced methods

• Adaptive planning use reinforcement learning that helps to the system to improve plans over time based on user actions.
• Deep learning is used to understand user preferences better and give more accurate suggestions for them.
• The AI gives clear reasons for its suggestions, so that users can trust the system more.

These methods are there to make the things more personalized, flexible, and trustworthy for its users. But most of these systems are still being tested and haven't been used in the real world yet.

G. A comparison of the proposed system with other systems

1. Theoretical v/s Practical Implementation
Most research papers are about for algorithmic models, but our Lazy Wanderer is a full working system that includes a front end, back end, and database integration.

2. Few features vs A platform that works with the other platforms
Many existing systems are there which only deal with certain parts, like route optimization with recommendations.
Lazy Wanderer includes:
• Planning the trip
• Optimizing routes
• Making an itinerary
• Updates in real time
• Keeping track of costs

3. Systems that are static vs The systems that are dynamic
Many of the research models use static datasets, but our Lazy Wanderer uses real-time data for planning that changes over time.

4. No planning for money
Most systems don't keep track of its user expenses and budgets, but the Lazy Wanderer has a separate module just only for managing the money.

5. Ability to grow and be useful
Many of research models do not have a user-friendly interface, but Lazy Wanderer does, because it has an interactive Angular-based UI and this UI makes the interface open smoothly and gives a clear overview of the options which are available in the application and easy to go through the application even for new the users.

Table 1 Summary of Related Work, Identified Research gaps, and Proposed Contributions

REFERENCES	ADDRESSED GAP	PROPOSED CONTRIBUTION
Gavalas et al.	It mainly focused on optimization models	It provides a completely implemented Real world system
Halder et al.	It has Limited real time adaptability	It is Integrated using Real Time APIs
Shrestha et al.	It works on only recommendation-based system	It combines both recommendation withiternary planning
Zheng et al.	It uses historical or previous Data	It uses real time and user preference data
Kurashima et al.	It has limited system integration	It has routing,itenary and expenses
Smirnov et al.	It has limited Scalability	It Uses Scalable client server architecture

H. Gaps in Research Found

The survey reveals the following deficiencies:

- AI models and full-stack systems don't work well together
- Not much help with adapting in real time
- No features for tracking money
- Not good at planning trips with many purposes
- Not having implementations that are easy to use and can grow with the business

I. What the proposed system will do

The Lazy Wanderer Smart Travel Planner fills in these gaps by:

- Giving you a complete travel planning platform
- Putting together recommendation, optimization, and processing in real time
- Helping with managing budgets and expenses
- Putting in place a client-server architecture that can grow
- Giving a useful and easy-to-use answer

III. PROPOSED FRAMEWORK

The suggested Lazy Wanderer Smart Travel Planner Framework aims to implement a smart approach to planning one's travels by taking into account preferences of each person, processes performed by the system, and information sources available at a given moment of time. In particular, this framework is based on a multi-level architecture that allows for smooth and efficient data processing and system functionality. Firstly, the user interaction level is used by the travelers, who enter their inputs related to the place, budget, dates of travel, and other individual preferences. They enter their information through the Angular-based frontend of the system, which provides users with a convenient way of communicating with the system and ensuring that data entered is sent to the back end of the application.

Finally, the third layer, which is associated with the data processing and validation, is implemented within the Spring Boot framework. In this regard, the information entered is validated, processed according to business rules, and further analyzed. As such, the framework will include a recommendation and planning layer, which will be responsible for generating customized travel plans. In order to generate personalized recommendations, the layer will use preferences, historical data, and context to identify recommended destinations, events, and activities for travelers. The framework uses a hybrid method of combining rules-based recommendations with data-based suggestions.

Moreover, the framework uses a routing optimization layer, which takes advantage of Google Maps APIs for determining efficient travel routes. In this way, it will be possible to identify the shortest or the most convenient ways to get from one destination to another, taking into account the distances and travel times. An important part of the framework design is its ability to take into account additional factors that may affect the travel plans, thus making it highly adaptive and reliable. As such, the framework includes a real-time data integration layer, which allows incorporating external data sources in order to make necessary adjustments to travel plans based on weather, travel time and other factors. One of the unique features of this framework is the integration of the expense management layer, which makes it possible for users to monitor expenses associated with traveling and stay within the predetermined budget during their travel.

The data storage layer comprises the core of the architecture in which all user data, information on trips, itineraries, and expenses is kept within a MySQL database. Entities and their relations are organized into an effective database schema to ensure efficient operation and consistency. Lastly, the presentation layer allows displaying the generated travel itineraries, routes, and associated expenses to the end-user in an interactive way by means of frontend technologies.

Proposed Framework Workflow
The whole flow of operations in the proposed framework can be presented as follows:
1. User enters travel-related data into the frontend application
2. The data is transferred to the backend side for validation
3. A recommendation engine generates a list of personal itineraries
4. Paths within the itineraries are optimized for efficiency
5. Real-time data is considered to make the system adaptive
6. Financial aspects of travel are managed via expense tracking
7. Final itinerary is generated and shown to the user

Benefits of the Proposed Framework
• Structured and modular approach to travel planning
• Allows for personalization according to user preferences
• Dynamic updating based on real-time data
• All functionality combined in one platform
• Visualization for better user experience

Finally, the framework is built in such a manner that it is easily extensible in nature. This means that it can be used for integrating various other features such as advanced machine learning models.

A. Functional Block of the Framework

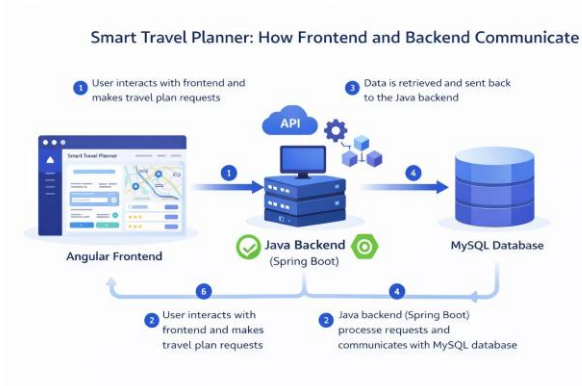


Figure 1 Conceptual sequence diagram of the proposed Smart Travel Planner

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

The planned Lazy Wanderer Smart Travel Planner will be built using a structured client-server architecture and a full-stack implementation approach. The system combines the frontend, backend, database, and external services to make planning a trip easy and smooth. The design of project says that this unified architecture make sure that this system can grow, is modular, and that the parts can talk to each other well.

A. The Overall Structure

The Angular-based front end acts as the client side , and the Spring Boot application act as the backend server side in a client-server model. RESTful APIs allow these parts to talk with each other, which makes them easier to share data and interact in the real time. The Angular framework is used on the client side to build a user interface that do changes and responds to what the user does. There are UI components, business logic components, and service layers in the frontend. Users can enter information about their trip, such as the destination, budget, travelling dates, and preferences, through the user interface. Angular components take these inputs and send HTTP requests to backend services to process them.

Spring Boot is used for building a backend on the server side. It is split into three layers: controller, service, and repository layers. The controller layer take care of requests that comes in, the service layer handles business logic, and the repository layer handles all the interactions with the database. This layered structure makes it easier to keep things separate and making them easier to fix.

B. Connecting the Frontend and Backend
REST APIs make it possible for the frontend and backend to work in together. When a user enters his travel information through the Angular interface, it is sent to the backend using HTTP methods like GET, POST, PUT, and DELETE. The backend handles the request and sends back the generated answer in a JSON format. Angular services take care of API communication, will be making sure that data is sent quickly and easily. Real-time updates and better user interaction are possible thanks to features like two-way data binding and a component based architecture. This combination makes sure that the system stays responsive and easy to use.

C. Database Design and Implementation
MySQL RDBMS serves as the database where Stiva MySQL, the solution store and manages all data of applications. Database schema. The structured designs of the database schema contain elements like user, trip, itinerary, hotel, attraction, route, preferences and expense. All entities(or)objects are linked to their mapping and each of them, primarily using one-to-many mappings. An example of this we can take a user having many trips, and each trip could have many timetables and expense records. In this design, we are forming our data in such a way that it takes less space and can quickly be fetched. The repository layer which is in the backend, tries to interact with database for performing all the CRUD operations to ensure consistency and reliability of data.

D. Module-Based Implementation
The trilateral system is put into operation in terms of interrelated modules each representing specific functions:

User Management Module : Supports user registrations, log-ins and profile management.

Trip Planning Module: A tool that help users to create & manage travelling plans by specifying its destination, budget, etc.

Itinerary Generating Module: Creates an organized travel, schedules by arranging beautiful attractions and activities.

Recommendation Module: Offers some personalized suggestions based on user entered preferences and stored data.

Route Optimization Module: Uses Google Maps API to find efficient travel routes and navigation.

Expense Management Module: Allows all the users to track and categorize their travel expenses.

Real-Time Data Module: Combines weather and travel APIs to provide current real time information.

This modular approach makes the system even more flexible and allows for easy addition of new features.

E. API Integration & Real-Time Data Processing
External APIs also have been integrated into this system to improve functionality. The Google Maps API is utilized majorly for route optimization and navigation, whereas weather APIs give real time weather data. Finally, APIs which are related to travel will be utilized to provide information on hotels and places of chosen interest. These integrations allow for dynamic updates based on changing conditions.

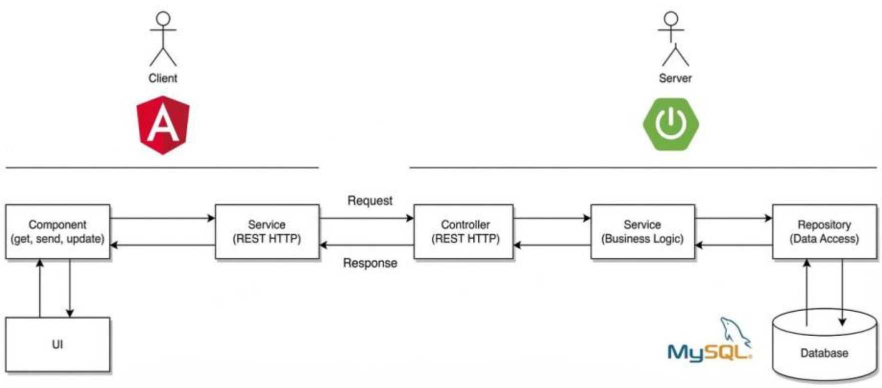


Figure 2 System architecture of the proposed Lazy Wanderer

F. Data Flow and Process Flow
The below following information represents the flow of data:

The user inputs their information about the upcoming trip through the Angular front-end

The request will be made to the back-end by using REST APIs

The request is handled by the back-end and here business rules are been applied

There is interaction between the system and its MySQL database

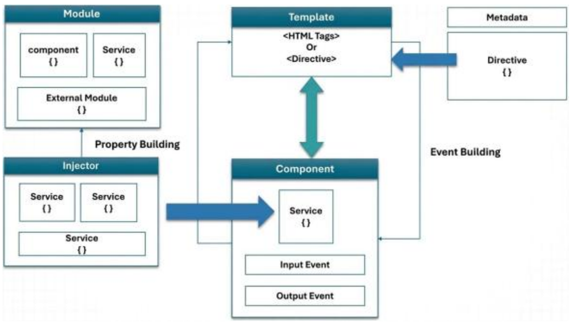
There are also some interactions happen via external

APIs for getting the real time information
A response in the form of JSON is generated by the back-end
The result is visible to the users through the front-end
The above workflow make sure of efficiency in communication and smooth performance of system operations.

G. Strengths of the Integrated Model

There are a number of strengths that arise from the integrated model, such as increased scalability, modularity, and efficiency in data processing. Separating front-end, back-end, and database components makes them independently developable and maintainable. The use of current frameworks leads to enhanced performance and responsiveness.

For user safety and protection of personal data and operations within the application, basic security mechanisms have been introduced into the design of the application. First of all, authentication is provided so that the identity of a certain user could be verified in order to use any functions such as creating a trip plan or expense tracking. In addition, authorization is needed to prevent unauthorized access to sensitive information by users, so that only authorized users would be able to edit their own data.



Flow: Controller→ Service→ Repository→ Database

Figure 3 Data flow diagram illustrating information flow across system components

V. EXPERIMENTAL SETUP AND RESULTS

A. Experiment Design

The design of the Lazy Wanderer Smart Travel Planner was developed in order to assess the performance of the system from the perspectives of usability, efficiency, and accuracy of the developed travel plans. The system has been developed based on the implementation of the web-based application with the use of such technologies as Angular, Spring Boot, and MySQL database. At the same time, several external APIs have been added to the system, among which one can point out Google Maps and weather services. The system was then tested under controlled conditions by utilizing the same configuration as a typical computer, which includes a contemporary web browser with reliable internet connectivity and a development server within the same machine. The backend services were hosted on the local server, while the frontend was accessed using a browser. The database was designed to save the users' information, travel plans, and costs.

In testing the system, several test scenarios were being designed by depending on the travel situations. Scenarios included differences in the preferences of users, for example, whether the traveler has budget constraints or is traveling for how long to what kind of destinations such as rural areas, tourist spots, or a mix of both. Testing of the system were being done using separate functionalities of their application. The functions consisting of itinerary creation, optimization of routes, processing of real time data, and calculation of the expenditures during travelling. Testing involved both individual and integrated tests of the functionalities.

B. Evaluation Criteria

System performance evaluations is based on the following criteria:
Personalization Accuracy: This calculate how close the suggested travel itineraries are aligned with users personal preferences.
Responding Time: It evaluates the timing that the system require for the respond to its user inputs and generate the results.
Route Optimization: This evaluates the extent up to which the route optimizing process reduces travel distances and times.
Real-Time Adaptation: It evaluates whether the system can possible to dynamically adapt its travel itineraries based on real-time data like weather conditions.
User Satisfaction: It evaluates the user experience with respect to usability and the quality of the system outputs is verified.
These criteria help in evaluating the performance of both system and user experience.

C. Experimental Results

The outcomes resulting or coming from the experiment tells that the performance of the Lazy

Wanderer system is excellent in all tested cases. The system showed greater efficiency in creation of travel plans according to personal preferences and information related to the context. Users received customized travel plans in accordance to their preferences and budget limitations. In terms of response time, it was all efficient since most of the queries had fulfilled within a few seconds, which guarantees a good user experience. The application of RESTful APIs in addition of to efficient backend processing enabled reducing latency times and improving data acquisition.

Regarding route optimization, the results showed high efficiency in planning effective routes for traveling. Through the application of the Google Maps API, the system suggested optimal routes that ensured reduced time and distance during traveling, which was particularly important in traveling to several destinations at once. The inclusion of real-time data added to the flexibility of the system. It was successful in incorporating the weather and travel-related data into the planning procedure. This allowed the user to make an informed choice about his journey. This flexibility of the system is what makes it stand out from other static travel planners.The expenditure management aspect was also successful as it assisted users in monitoring their expenses and classifying them.

D. Comparison

Comparison between travel planning using traditional methods and current methods can reveal some important benefits provided by the new proposed approach. Traditional methods involve much work, while Lazy Wanderer takes care of all things automatically in very little time.

Table 2 Comparison between different travel planning methods

Features	Traditional Methods	Current Methods	Proposed Method
Personalization	Low	Moderate	High
Real-Time Updates	Not Available	Limited	Available
Route Optimization	Manual	Partial	Advanced
Expense Management	Not Available	Rare	Available
Integration	Low	Moderate	High

In this way, the benefits of the new approach become evident through comparison.

E. Discussion

The outcomes which are from the experiment indicate that the Lazy Wanderer is a system effectively solves travel planning problems. The use of several modules within one application ensures better usability and less reliance on numerous software programs possibly. The combination of personalized information, real-time flexibility, and cost control makes the system a useful tool for today's travelers. Nevertheless, some drawbacks have been identified. First, the functionality of the system is influenced by the accessibility of third-party APIs. Second, the present design of the system relies on rule-based recommendations. This approach could be improved using more complex machine learning techniques. Based on the results of the experiment, the Lazy Wanderer can efficiently solve the problem of trip planning. It is due to the fact that multiple modules are used in one program which guarantees better usability. Thus, users will no longer need to download many different programs to solve their problem.

VI. DISCUSSION AND CONCLUSION

The development of Lazy Wanderer Smart Travel Planner reveals the successful implementation of using full-stack web development technologies in conjunction with aiming in structuring and intelligence in travel planning algorithms. This project has successfully integrated a variety of functions, such as a trip planning, route optimization, incorporation of real-time data, and cost management in one platform, thereby overcoming many shortcomings inherent in classical travel planning approaches. Firstly, an important observation made during the design is the noticeable boost of the efficiency of the travel-planning process. Traditional travel planning involves the use of multiple apps by travelers; however, using the proposed system, users can easily and effectively manage their trips. It should be noted that the frontend was built using the framework of Angular; therefore, the interface is responsive and interactive, while the backend was developed using Spring Boot technologies.

The proposed system performs well in providing personalized recommendations, which is an important feature in modern travel apps. It creates travel plans based on user inputs such as budget, travel time, and personal preferences, making the experience more suitable for each user. At present, the system has mainly uses structured data and simple rules, but it has the potential to include machine learning in the future to improve accuracy. Another strong point is its ability to use real-time data through APIs, allowing it to adjust recommendations based on changes like traffic or weather so users get updated information. However, relying on external APIs can sometimes lead to delays or temporary issues. In addition, the system includes an expense management feature, which makes it more useful, as many existing systems only focus on recommendations and do not help users manage their travel budget.

With Lazy Wanderer, it is possible for the users to track their spending throughout the entire journey. As a result, a user benefits from some additional advantages, such as financial control and making the right decision about where to go and what to do while traveling. There are certain weaknesses related to the examined system. Particularly, the recommendation module that cannot be viewed as being sophisticated due to the lack of an advanced learning model. Also, the problem with scalability might appear when dealing with huge amounts of data and streams of users. In general, the system manages to accomplish all goals and provide a flexible and functional solution to the problem are under examination. Along with simplification of the travel-planning process, the offered system improves the quality of travel experiences due to the personalized recommendations and quick reaction of the system.

Given certain improvements like adds the intelligent way learning model, integrating the functionality into the mobile application, and addition the possibility of booking tickets online, the system demonstrates great prospects for a implementation and further development. To sum up, the current research demonstrates that the combination of modern web technologies and intelligent data processing can be an effectively used to address the emerging needs of the traveling audience. The offered solution is a valuable contribution to the area of smart tourism and opens up new perspectives regarding the development of travel systems.

VII. ACKNOWLEDGEMENT

The authors would like to extend their heartfelt thanks to their project guide,prof Darshan M Patel, for his constant guidance, encouragement, and valuable suggestions that greatly contributed to the successful completion of this project.

REFERENCES

[1] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “A survey on tourist trip design problems and algorithms,” Archives of Computational Methods in Engineering, vol. 21, no. 3, pp. 263–286, 2014.
[2] S. Halder, K. H. Lim, J. Chan, and X. Zhang, “A survey on personalized itinerary recommendation: From optimisation to deep learning,” Applied Soft Computing, vol. 152, 2024.
[3] D. Shrestha, T. Wenan, N. Rajkarnikar, and S.-R. Jeong, “Personalized tourist recommender system: A data-driven and machine-learning approach,” Computation, vol. 12, no. 3, 2024.
[4] S. P. R. Asaithambi, R. Venkatraman, and S. Venkatraman, “A thematic travel recommendation system using an augmented big data analytical model,” Technologies, vol. 11, no. 1, 2023.
[5] C. Huda, R. Suryani, and A. Azhari, “Smart tourism recommender system modeling based on hybrid technique and content-boosted collaborative filtering,” 2025.
[6] X. Xiao, P. Wang, Y. Ge, J. Luo, H. Chen, and Y. He, “GeoKG-HSA: A framework for habitat suitability assessment with geospatial knowledge graphs,” International Journal of Applied Earth Observation and Geoinformation, 2025.
[7] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining interesting locations and travel sequences from GPS trajectories,” in Proc. 18th Int. Conf. World Wide Web (WWW), 2009, pp. 791–800.
[8] M. Baraglia, F. Chessa, R. Perego, and D. Silvestri, “A collaborative filtering approach for recommending travel itineraries,” in Proc. Int. Conf. Web Intelligence, 2012, pp. 93–100.
[9] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” IEEE Trans. Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.
[10] P. Vansteenwegen and W. Souffriau, “The orienteering problem: A survey,” European Journal of Operational Research, vol. 209, no. 1, pp. 1–10, 2011.

[11] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in Proc. ACM Conf. Electronic Commerce, 1999, pp. 158–166.
[12] K. Kapoor, K. Tamilmani, N. P. Rana, and Y. K. Dwivedi, "Advances in social media research: Past, present and future," Information Systems Frontiers, vol. 20, no. 3, pp. 531–558, 2018.
[13] H. Wang, Y. Li, and Y. Sun, "A deep learning approach for personalized travel recommendation," IEEE Access, vol. 8, pp. 123–134, 2020.
[14] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotagged photos," Knowledge and Information Systems, vol. 37, no. 1, pp. 37–60, 2013.
[15] A. Smirnov, A. Kashevnik, and N. Teslya, "Context-aware recommender system for tourism," Procedia Computer Science, vol. 31, pp. 611–620, 2014.