GroQ SQL Travel Planner: Book Flights, Hotels, Tours, Weather-Based Recommendations and Chatbot Support with Firebase Login and SuperBase Captcha

Gayathri Ramasamy, Gurupriya M., Gamidi Rohan, Vinitha Chowdary A.,
Tejaswi Muppala and M. Hemasri
Department of Computer Science and Engineering, Amrita School of Computing, Bengaluru, Karnataka, India

Keywords: Chatbot Support, Firebase Authentication, Flight Booking, Google Sign-In, GroQ API, Guided Tours, Hotel Reservations, MySQL Database, Node.js Backend, OpenWeatherMap API, React.js Frontend, RESTful API, Superbase CAPTCHA, Travel Planner, Weather Recommendations.

Abstract: GroQ SQL travel planner is an online portal encompassing all the important features to make and manage a scheduled trip such as flight booking, hotel reservation, guided tours, with weather-based recommendations and a chatbot support. For a seamless user experience, this application employs a React.js frontend, a MySQL database, and Node.js and Express for the backend. For authentication, Firebase provides Authentication and Google Sign-in and to avoid bot access Superbase provides CAPTCHA. The system has the ability to make use of RESTful APIs to dynamically connect the frontend with that backend and database and filter the options in real-time improving the way in which the frontend is linked to the backend and database. Weather data from Open Weather Map API and a GroQ-based chatbot further enrich users' experience to provide an AI driven approach. This application can further filter users' needs matching, searching for certain places, services, and personalizing the trip, and thus, making this a robust application which provides travel solutions.

1 INTRODUCTION

Tour planning can be seen as dealing with a disparate set of services for scheduling flights, accommodation, and excursions as well as such issues as weather conditions, security and individual customers. This makes the process rather a lengthy and tedious exercise and at the same time fraught with several errors most especially if the end user does not have prior experience with the tools and/or the resources at his disposal. Having no single platform just amplifies the issue, and as a result, there is inconvenience and compromised satisfaction with travel.

This poses on society disadvantage in that it denies society the best travel itineraries, leads to overcharge and has negative effects on the environment because of wrong travel paths and overbooking of the accommodation facilities. It also presents difficulties for the travel operators who are already under pressure to satisfy user requirements appropriately. The demand of more simplified and easier to approach solution is highly essential to improve the UX and business performances.

These challenges are responded to by the 'GroQ SQL Travel Planner' that provides a single package for flight procurement, as well as hotel bookings, guiding services, and even climate-dependent suggestions. Through Firebase Authentication, Google Sign-In, and superbase CAPTCHA feature it allows secure and efficient user login. Weather Application by OpenWeatherMap API integrated with AI along with GroQ empowered weather Chatbot gives real time weather details \& personalized travel recommendations. Besides, users can sort places according to personal choice, for example, the cost or the weather or other interests they have thus providing results which will meet the customers' needs most.

This work is unique, as it proposes a true integration of AI, safe authentication, and on-demand dynamic data offering to create a customized travel experience. The platform is accountable to UN Sustainable Development Goals (UN SDGs) 9 and 12

Ramasamy, G., M., G., Rohan, G., A., V. C., Muppala, T. and Hemasri, M.
GroQ SQL Travel Planner: Book Flights, Hotels, Tours, Weather-Based Recommendations and Chatbot Support with Firebase Login and SuperBase Captcha.
DOI: 10.5220/0013943300004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 5*, pages 758-766
ISBN: 978-989-758-777-1

on Industry; Innovation \& Infrastructure and Responsible Consumption \& Production respectively because it helps in reducing unending wastage of resources while planning for travel.

The rest of this paper is organized as follows: Section II contains the literature review followed by Section III will give an overview of how the system is implemented and the methodology and what major features are. Section IV will provide details on results. Section V will give information regarding future work and conclusion and possible expansions of this system.

2 RELATED WORKS

Ramasamy et al. developed a brain tumor segmentation model based on Link-Net, using ResNet152 as the backbone architecture and four MRI modalities as the input. The proposed model obtains a very promising performance in the BraTS 2020 dataset with a Dice coefficient of 0.7773 and a Jaccard index of 0.7169. Future work will focus on improving efficiency and performance at the pixel level.

Tuba et al. present the porting of FreeRTOS to a RISC-V architecture on the SPIKE simulator, with improvements in documentation and maintenance. They have made an assessment of its real-time features-inter-process communication and mutex-which guarantees that task creation, deletion, and context switching are very efficient for time-critical embedded systems.

Traykov et al., propose a testing framework that could be used for LLPs to test their security concerning known vulnerabilities, such as prompt injection and denial of service. It was tested on three models-Llama3-70b, Mixtral-8x7b, and Gemma-7b-out of which one had flaws, proving the efficiency of the framework. Future work includes increasing the coverage done by tests and integrating it with blockchain for transparent reporting.

Ramasamy et al., present an XGBoost framework with hyperparameter tuning for the prediction of Type-2 Diabetes Mellitus in the context of the PIMA Indian Diabetes dataset. For this purpose, the model is aggressively trained by using pre-processing, feature extraction, and then 10-fold cross-validation.

L Bianchi et al. fine-tuned it with Grid Search; it attained an accuracy of 94.5\%, outperforming SVM, K-NN, and QDA.Reddy et al., present an AI-enabled stress analysis system using GPT-3.5, designed to be highly scalable by using NoSQL databases. A Flask web application collects user input, analyzes the text using AI, and provides specific suggestions regarding stress management; MongoDB processes real-time data in a secure manner.

Neszlényi et al., present AssistantGPT, a platform that utilizes OpenAI's GPT API for both voice and text-based tasks. Using a React interface and FastAPI backend, it simplifies tasks like file management and script execution. Evaluation results demonstrate a 62.5\% reduction in task times and increased user satisfaction, highlighting its potential to boost productivity, despite challenges such as internet reliance and customization complexity.

Dong et al., examine cloud-native databases, emphasizing their benefits such as elasticity and cost-efficiency. It covers OLTP/OLAP architectures, innovations such as compute-storage disaggregation, and scalability methods. Case studies from Snowflake, Redshift, and Aurora showcase improved performance, with future research aimed at serverless architectures, multi-cloud services, and security.

Zhou et al., examine the role of AI in improving database functions and how databases support AI deployment. It highlights innovations like deep learning for cardinality estimation, reinforcement learning for query optimization, and AI-powered data cleaning. The study also explores challenges such as hybrid data models and AI-DB co-optimization, with future research focused on integrating AI with databases to solve complex data challenges.

Pramono et al. present a secure authentication framework for an employee presence system, utilizing Firebase Authentication, RESTful APIs, and JWTs. It ensures secure data exchange and optimal performance, with future work focused on integrating multi-factor authentication with biometrics for enhanced security.

Rajappa et al., discuss the implementation of the PingER project on Android devices, utilizing Firebase to monitor global internet performance. The Android app gathers real-time data on metrics like latency, jitter, and packet loss, expanding the coverage of the PingER network. The results highlight successful performance monitoring, with future plans to integrate advanced analytics and visualization tools.

Nitu et al., improve a personalized travel recommendation system (PTRS) by incorporating recency effects using Twitter data. The system employs machine learning for tweet classification, sentiment analysis, and recency weighting, achieving 75.23\% accuracy and outperforming previous models. Future efforts will aim to refine the models, expand the dataset, and integrate additional social media platforms.

Singh et al., present a MERN stack-based web streaming application leveraging WebRTC and RTMP for real-time, cross-platform video streaming. The system provides low-latency, scalability, and security, featuring adaptive bitrate and error recovery capabilities. Plans for further development include adding VR support, integrating social media, and enhancing user analytics.

Kulkarni et al., present the development of an online food ordering system with a recommendation module, utilizing ASP.NET, SQL Server, and the K-Nearest Neighbors algorithm. The system improves user experience and restaurant operations by suggesting items based on previous orders. The results indicate enhanced customer satisfaction and efficiency. Future developments will focus on refining recommendations and enabling real-time adaptation.

Manish et al., utilizess LLMs and NLP to optimize resume parsing, processing resumes in 2.0 seconds with high precision. Although format variability poses a challenge, it enhances ATS by providing structured and actionable insights.

Yeong et al., presents a telematics system that utilizes IoT, cloud computing, and analytics for real-time alerts and data-driven decisions, facilitating machinery monitoring and optimization with the potential for predictive analytics.

3 METHODOLOGY

The GroQ SQL Travel Planner is a web-based tool implemented with a highly customizable and flexible structure to support travel planning with a dynamic and modular architecture. The architecture adopted in the system development entails frontend, backend and the database levels. Furthermore, other external APIs and AI capabilities are incorporated to improve the user interface of the application.

3.1 Frontend

- Built using React.js to provide an interactive and responsive user interface.
- Includes pages for login, sign-up, and forgot password, home, Navigation Bar, Hotel cards along with main features like flight booking, hotel reservation, guided tours, and weather based travel recommendation chatbot.
- User authentication is handled through Firebase, enabling secure login via email/password or Google Sign-In.
- CAPTCHA verification is implemented using hCaptcha to prevent bot access.

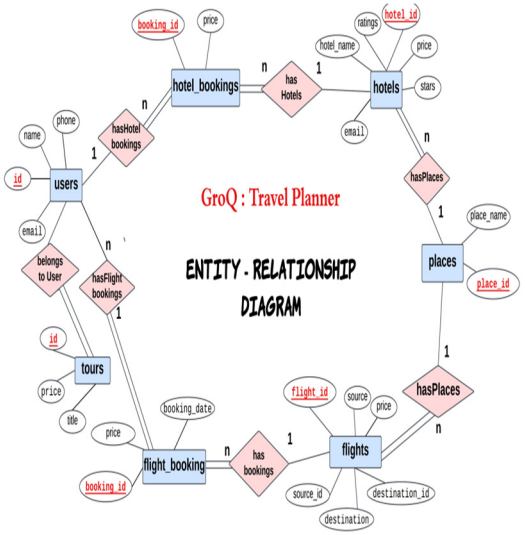

Figure 1: Entity-relationship diagram of GroQ-SQL travel booking.

3.2 Backend

- Developed using Node.js and Express.js, the backend serves as the intermediary between the frontend and the database. Implements RESTful APIs for handling user actions such as booking flights and hotels, retrieving weather recommendations, and interacting with guided tours. Includes custom endpoints to dynamically fetch and filter user preferences in real-time.

3.3 Database

- Uses MySQL to store and manage data such as user information, hotels, flights, places, and booking records. Structured with normalized tables for optimized data retrieval and storage. User-related data (e.g., name, email, and phone) is securely stored and accessed through structured queries.

The Entity-Relationship (ER) diagram for the GroQ: Shown in the context of the Travel Planner Figure.1., proposed is a logical and well thought through concept of the database schema which will take the role in the system and allow the application to readily support the process of travel management. Fundamentally, there is the database relates many to many relationship Users who make many Hotel bookings Flights bookings and can be part of many tours. They are linked with entities as Hotels which have fields of hotel name, price and ratings, Flights

containing fields such as source, destination and price etc. It also include Places, in which hotels and flights are directly connected to destinations, and Tours which enables users to travel through various guided tours. With relationships clearly defined through cardinalities like 1: n and n: n, the schema has no problem accommodating these transformational scenarios and provides a solid foundation for functionalities such as dynamic filters, real-time updates and recommendation. Front-end is based on React.js Framework, Back-end is Node.js, Database is MySQL, and the system incorporates RESTful API features such as an Open Weather Map for integrating weather-based recommendations and a GroQ-based AI Chat bot. For security guarantees, there is Firebase Authentication to eliminate bot access while Superbase CAPTCHA also helps to safe the system from bot activity. Combined, this design enables the user to have an easy way to plan and organize their trip from one interface to another seamlessly.

3.4 External APIs and AI

OpenWeatherMap API: Provides weather data to recommend travel destinations and activities based on real- time conditions.
GroQ-based Chatbot: AI-driven chatbot assists users with queries related to travel planning and personalized recommendations.
Firebase Authentication: Ensures secure user login with support for email/password authentication and Google Sign-In.
Superbase Authentication: Superbase for Captcha authentication and API using hCaptcha integration.

4 FUNCTIONAL MODULES

4.1 User Authentication and Management

The module for User Authentication and Management enforces proper access control when users login into the application. Both the email & password registration and Google sign-in are supported for the application using Firebase Authentication. The user enters their name, email, phone number, validates the email format and password and the application safely signs them up on Firebase. Also, the collected user data is transferred to the backend and saved into MySQL including personal user data. The login process includes Firebase authentication, and, to stop bots, we use HCaptcha. Google Sign-In is also fortified with CAPTCHA validation to make user verification extremely effective. Users are offered two options to regain access to their account if they forgot the password: through email or phone number. If the reset type is email, a password reset e-mail is forwarded to the user, and they are given instructions on how to safely alter a password. For phone-based resets, Firebase's RecaptchaVerifier is used to perform the verification by sending the code to the user's phone then the password updates upon confirming the code. Checks such as data format check, mismatch on passwords, and reCAPTCHA add the security aspect and improve the authenticity of the system to deliver a safe authentication method. Figure. 2. Shows the login page, followed by Figure. 3. Which shows the forgot password page while Figure. 4. Shows the sign-up page.

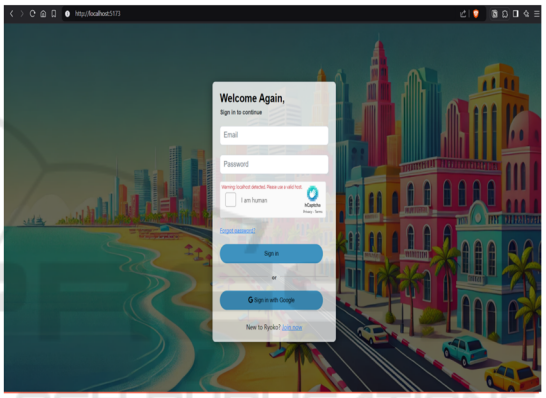

Figure 2: Login page.

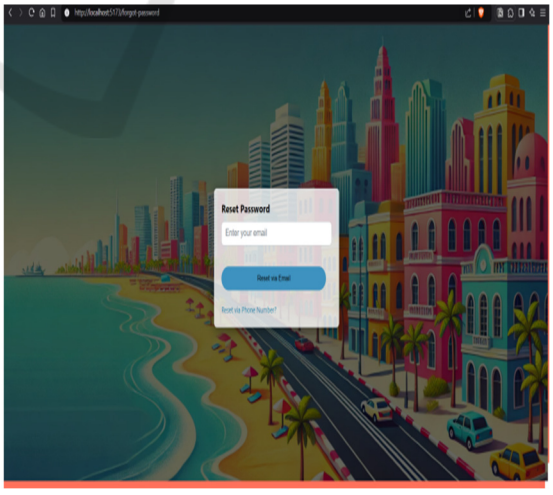

Figure 3: Forgot password page.

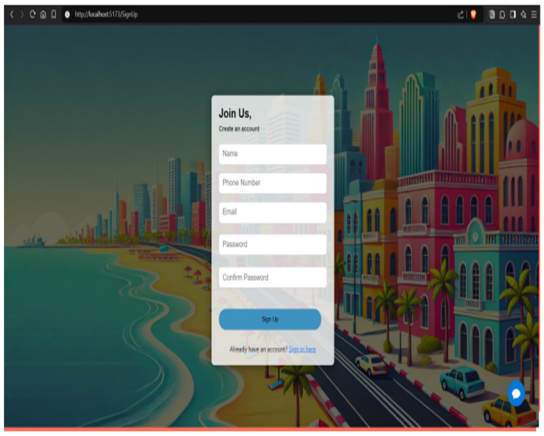

Figure 4: Signup page.

4.2 Home and Navbar

The Home Page is the primary system interface that allows users to make a direct connection to primary travel planning facilities such as flight schedules and booking, hotel reservations, etc., weather and other updates. As for it, the Navbar contains clear navigation icons for such main areas as user identification, history of orders, and travel preferences. Figure.5. below shows the navigation bar.

4.3 Chatbot

Chatbot GroQ Chatbot: Offers AI-driven assistance to address user queries and provide personalized travel recommendations. The ChatBotPopUp component is integrated here as a button pop - up which hovers over the screen on the bottom - right corner. It moves up and down based on the visibility of the" backToTop" button which has similar looks and behavior. This window/ pop - up that hovers, has two tabs where one is a chatbot that recommends the user based on his location and holiday preference. It can take everything into consideration from budget, location, likes and dislikes. The GroQ based Chatbot works with the "llama3-8b-8192" model with an API key that's assigned to new users. Once a client object is created in the code, it is programmed to take two inputs: One is where we initialize a message to tell the model how to perform. This gives the model context of how it is supposed to behave, and what it's supposed to know before beginning a chat. This is where we describe the bot that it is a travel recommendation bot focused on India, and that its goal is to provide specific travel recommendations based on user input. The bot begins by greeting the user and asking for their preferences, then provides a destination suggestion. Since the bot has issues with maintaining token length, and it looks odd in the chat window, we ask it to maintain short and crisp answers, and questions. And to address another issue where it asks questions again and again, we ask the model to not ask more than 3 questions whatever the case is.

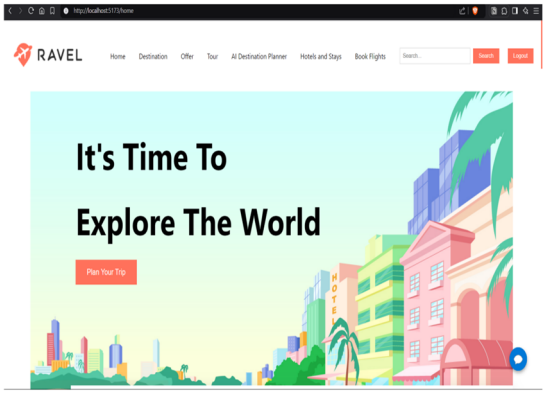

Figure 5: Home and navbar page.

After the initial message, we route the user messages to the chat area, to which the model starts to respond. These responses are then put into the chat window which in turn makes it look interactive.

Weather Integration: Integrates OpenWeatherMap API to offer real-time weather-based suggestions. The weather integration is implemented as a new tab since it gets complicated to not get deeply involved with a model yourselves and only depend on the API. This is because, to have a model that does both recommendation based on user preferences and recommend based on weather conditions on the next few days. The major problem is the one we arrive to over here, world standard, and widely accepted models like LLaMa or GPT are not given internet access to access weather APIs or monitor weather conditions, or get regular updates on news or whatever. So what we do here is, we update the database once in every 24hrs of the latest data of major cities in India, put it onto superbase's PostgreSQL database, it's about 1160 reading (1 reading every 3 hours for over 5 days) monitoring conditions like date - time, temperature, humidity, weather description, and city name. Now this PostgreSQL database can be read by a RAG model to make decisions with the help of a base backbone model like GPT or oLLaMa or Anthropic, with combination of user preferences and season to make a very sophisticated and specific combination and concoction which works beautifully. Figure 6 shows the Database UML Diagram.

4.4 Database

• MySQL Database:
- Stores user details, flight and hotel data, booking records, and guided tours.
- Ensures efficient data retrieval and secure management of sensitive user information.
- Table Relationships: Designed to maintain data integrity and optimize search and filter operations.

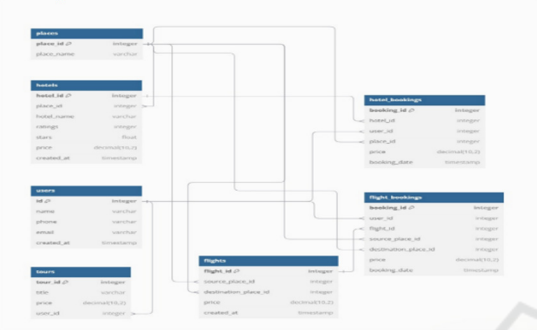

Figure 6: Database UML diagram.

4.5 Airplane Booking

The flight booking system consists of three input parameters that include source, destination, and travel date to generate flight fares in real-time. Users can choose flight options and a record is made in a MySQL database which contains details of the booking and the user is notified through a confirmation pop up. It also guarantees easy linking with a backend API to get place options and their prices and update the price field in real time. The locations are in dropdowns and dates in a calendar to make the booking as easy as possible for the clients. Figure 7. shows the flight bookings component.

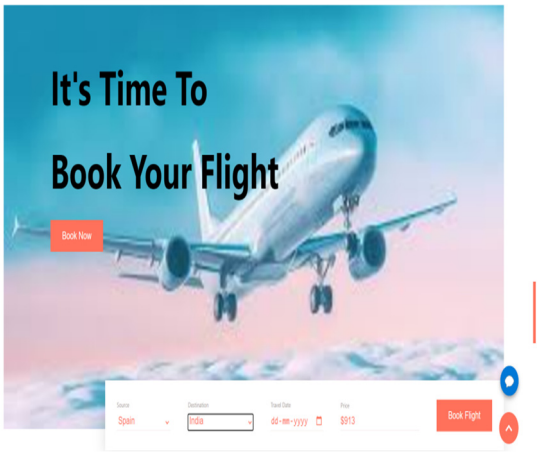

Figure 7: Flights booking.

4.6 Tour Booking

The booking of the tours involves choosing from available tour destinations provided, the cost for the tour, image, and the reviews of the tour. With just a click, users can secure the tours they want and will only need a click to confirm the choice with the pop up box; the tours are saved in a database associated with the user account. A backend API is integrated to save the bookings while the system was designed with an attractive graphical design and dynamic features to enhance the users' interest. Figure. 10. Shows the tour booking page.

4.7 Hotel Booking

The hotels page comprises a list of Hotels filtered by locations and prices to help the users in selecting the required hotels. Booking of the hotels can be done very easily with all the bookings recorded in the database and optional recommendations from the site being included for greater usability. The system is also dynamic especially with regards to input such as destination and price range to ensure that a user gets a friendly interface for efficient hotel bookings. Figure. 11. Shows the Hotel bookings.

Figure 8: Hotel booking.

(a) Tours page (b) Updated database after tour booking

Figure 9: Tour booking.

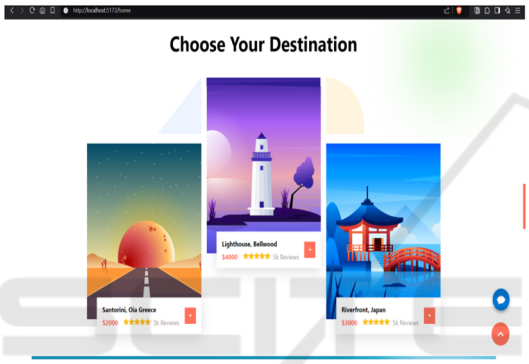

Figure 10: Tour booking.

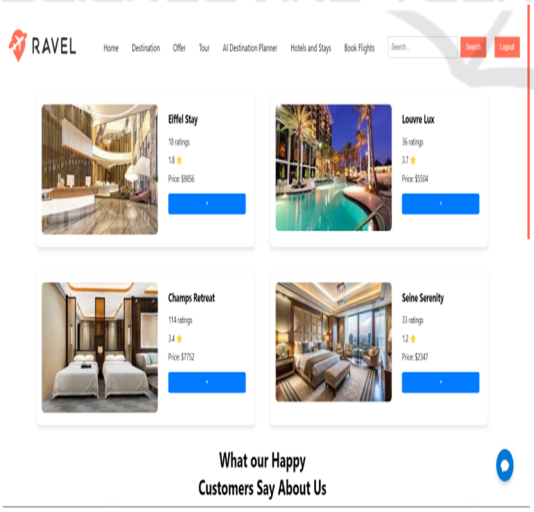

Figure 11: Hotel booking.

4.8 Backend

Express.js was used to create the backend which interacts with a MySQL database to offer a variety of travel-related services among them are the services of user management, flight and accommodations, and tour booking services. The endpoints include signing up the user, getting flight prices, getting hotels by location and price, and processing bookings; all mysql database information is encrypted. Improper backend also has capability to manage the last active user which is quite useful for bespoke operations. The high level of constants' check, dynamic SQL statement execution, and data validation are considered as the crisp factors for the application and its steady growth.

5 RESULTS

5.1 Hotel Booking

In the hotel booking part we have two ways, one is through the search bar and one is through the home page. when i click on the search bar in the navbar, a list of places in the dropdown list will appear, when you select a place it will redirect you to a new page where all the hotels corresponding to that place will be fetched from the database and displayed. when you click on the "+" button, an alert will appear if you want to confirm the booking, if you click on okay then the, the hotel bookings table in the database will be undated. another way is to set the price range and based on the price range and the selected place, the filtered hotels based on the price range will be fetched and displayed. The images related to Hotel Bookings are displayed in Figure. 8.

5.2 Tour Bookings

In tour bookings the user can book multiple tours, and once the + button is clicked an alert will be displayed to confirm booking and the tour will be booked and updated in the database. Figure. 9. shows all the related images for Tour Bookings.

5.3 Airplane Booking

In the Airplane Bookings, the user has to choose the source, destination and dates for travelling and the price changes according to the source and destination selected. If any one of the fields is not selected and the 'Book Flight' button is clicked then a message box will be shown. If all the details are filled it will result in successful booking in the database. The alert is shown in Figure. 12. while the database after a booking is shown in Figure. 13.

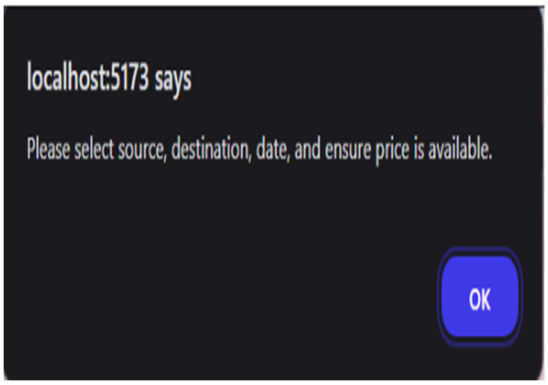

Figure 12: Alert for empty field before booking.

booking_id	user_id	flight_id	source_place_id	destination_place_id	price	booking_date
1	1	45	2	8	750.00	2024-11-28
2	3	120	5	10	920.00	2024-11-27
3	6	233	3	9	1100.00	2024-11-27
4	8	88	4	11	850.00	2024-11-28
5	10	301	7	15	980.00	2024-11-25
6	2	502	1	6	600.00	2024-11-25
7	5	147	12	13	1150.00	2024-11-28
8	7	399	9	14	700.00	2024-11-28
9	9	220	11	2	1030.00	2024-11-27
10	11	510	10	3	1180.00	2024-11-26

Figure 13: Updated database after airplane booking.

5.4 ChatBot Integration

Implementing a chatbot within the platform provided a highly individualized travel planning due to the GroQ-based model run by LLaMa3-8b. It was very smart in anticipating the kind of inputs that you would want to input like location, budget and preferred locale to offer relevant travel advice. The AI Integrated chatbot successfully provided contextual recommendations with the integration of weather details opted from the Open Weather Map API as shown in Figure. 14. Metered logical data being used was updated every one day to ensure that it was sufficient and timely. For improved efficiency, the use of follow-ups was moderate at most three in order to avoid long strings of conversations that may bore the user.

The user interface of the chatbot was designed to appear as a pop up so as not to intrude with the rest of the other modules of the platform. Its AI-based logic focused on providing fluent and low delay interactions that will enhance users' perceived satisfaction. The combination of the dynamic visualizations of the weather updates as well as the smart suggestion systems made the chatbot a very useful tool that folds into the design of the interactive travel planning well and effectively.

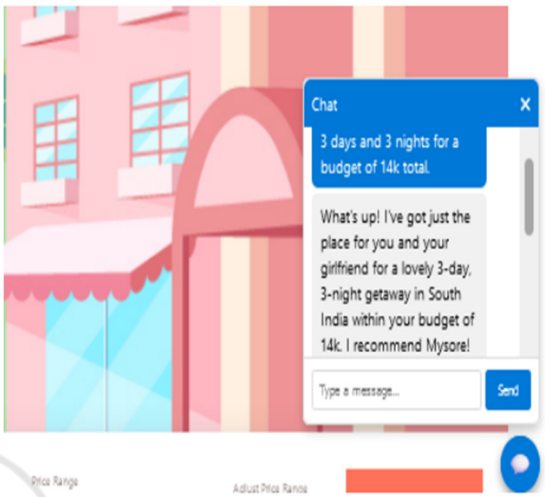

Figure 14: GroQ ChatBot integration.

6 CONCLUSIONS

Enter the GroQ SQL Travel Planner A Complete Intelligent Solution to Modern Travel Planning Problems It streamlines user convenience by combining flight booking, hotel reservation, and guided tours into one service. Utilizing current technologies such as React. js, Node. js, and MySQL, the system enables efficient and seamless interactions. Adding Firebase Authentication and Superbase CAPTCHA, protects your application from unwanted interference. The user experience is more personalized with real-time weather-based recommendations and a GroQ-based chatbot that makes travel more flexible and informative. The modularity of the architecture allows for scalability and future improvements/expansions. A well-structured ER model enables dynamic filtering and fast decisions, ensuring efficient data handling. This example demonstrates the power of AI fetch-focused help through its integration of the LLaMa and OpenWeatherMap APIs. In conclusion, the project takes a well-rounded approach by providing optimized travel suggestions that account for both user needs and environmental considerations. A solid potential for application in the real world and allow its evolvement in this travel tech space.

REFERENCES

A. Rajappa, A. Upadhyay, A. S. Sabitha, A. Bansal, B. White, and L. Cottrell, "Implementation of pinger on android mobile devices using firebase," in 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence), 2020, pp. 698–703.

B. A. Reddy, G. Someswara Reddy, K. Lokesh, S. B, and R. M, "Aidriven stress analysis and management: A novel approach leveraging openai and nosql databases to empower students in stress management and promote overall student well-being and academic progression," in 2024 International Conference on Computational Intelligence and Computing Applications (ICCICA), vol. 1, 2024, pp. 210–215.

D. J. Yeong, K. Panduru, and J. Walsh, "Smart agriculture: Software platform for telematics monitoring in farm machinery," in 2024 35th Irish Signals and Systems Conference (ISSC), 2024, pp. 1–6.

H. Dong, C. Zhang, G. Li, and H. Zhang, "Cloud-native databases: A survey," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 12, pp. 7772–7791, 2024.

K. Traykov, "A framework for security testing of large language models," in 2024 IEEE 12th International Conference on Intelligent Systems (IS), 2024, pp. 1–7.

K. Balazs Neszl ´ enyi, A. Milos, and A. Kiss, "Assistantgpt: Enhancing ´ user interaction with llm integration," in 2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY), 2024, pp. 000 619–000 624.

L. H. Pramono and Y. K. Yana Javista, "Firebase authentication cloud service for restful api security on employee presence system," in 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2021, pp. 1–6.

Ojas, Ojas, Aarav Vashishtha, Sumit Kumar Jha, Vivek Kumar Shah, Rohit Kumar, and Gayathri Ramasamy. "James: Enhancing Judicial Efficiency with Smart Administration." Available at SSRN 5091509 (2024).

P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," Big Data Mining and Analytics, vol. 4, no. 3, pp. 139–154, 2021.

SR, M., & Ramasamy, G. (2024). Heartbeat Scream Detection System.

V. Sevagen, H. Pabbati, P. Chanda, and A. Kumar, "Intelligent chatbot for student monitoring and mentoring," in ICT Systems and Sustainability, M. Tuba, S. Akashe, and A. Joshi, Eds. Singapore: Springer Nature Singapore, 2023, pp. 393–399.

V. K. Singh, P. Awasthi, L. Tripathi, S. Thakur, and P. Singh, "Mern (mongodb, express-js, react-js, node-js) stack web-based streaming and broadcasting application," in 2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT), vol. 1, 2024, pp. 1–6.

V. Manish, Y. Manchala, Y. Vijayalata, S. B. Chopra, and K. Y. Reddy, "Optimizing resume parsing processes by leveraging large language models," in 2024 IEEE Region 10 Symposium (TENSYMP), 2024, pp. 1–5.

Vashishtha, A., Kumar Jha, S., Kumar Shah, V., Prasad Chauhan, J., Bhagat, K., & Ramasamy, G. (2024). Empowering Student Success: A Comprehensive Task Management and Notification System. Available at SSRN 5091489.

X. Zhou, C. Chai, G. Li, and J. Sun, "Database meets artificial intelligence: A survey," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 3, pp. 1096–1116, 2022.