TRAVEL TICKET BOOKING SYSTEM

Marimuthu K[1]
Department of Computer Science and Engineering
Presidence University
Bangalore, Karnataka, India
marimuthu.k@presidencyuniversity.in

Lavanya M H[2]
Department of Computer Science and Engineering
(Data Science)
Presidence University
Bangalore, Karnataka, India
lavanyamh23@gmail.com

Dabbara Asritha[2]
Department of Computer Science and Engineering
(Data Science)
Presidence University Bangalore, Karnataka, India
dabbaraasritha7@gmail.com

Sindhu M[2]
Department of Computer Science and Engineering
(Data Science)
Presidence University
Bangalore, Karnataka, India
sm1682021@gmail.com

Jayashree R[2]
Department of Computer Science and Engineering
(Data Science)
Presidence University Bangalore, Karnataka, India
jayajayashree78@gmail.com

Abstract*—The Travel Ticket Booking System is an intelligent assistant designed to automate and manage travel service provision through a conversational interface. The system built to work smoothly on Telegram platform seeks to facilitate the approach of booking travel tickets and managing interactions with users in a responsive and efficient way. It solves the intricacies of the existing ticketing platforms by offering a combined, available, convenient, and easy to use solution that can be employed anywhere with connectivity to the internet. This system is planned to cater for such important functionalities as planning trips, booking tickets and supplying users with timely alerts and notifications. Apart from that, it helps in system level diagnostics, travel status, and integration of auxiliary modules for good service delivery. They have tested the platform both in different test environments and operational scenarios to prove its ability to manage concurrent queries, sustain the system's uptime, and return results quickly. The architecture of the Travel Ticket Booking System focuses on modularity, scalability, and adaptability making it a future proof solution for smart mobility, convenient way for navigation and personalized travel management. It does not only meet the short term needs of travel ticket bookings but also offers a foundation for potential future technologies such as intelligent routing, on the spot analytics and predictive trip planning. The light-weight deployment framework of the system also enables it to be implemented in both individual and in enterprise over minimal consumption of IT resources.

***Index Terms*—Keywords: Architecture, Diagnostics, Integration, Modularity, Predictive, Routing, Scalability, Uptime, Deployment, Automation.**

I. Introduction

The world of travel and transportation have had explosive change with the consolidation of intelligent automation and digital interfaces. With the increased dependence of more people and enterprises on online platforms to handle their travel requirements, the need for rapidly responsive, adaptive, and user-friendly systems has become vastly greater. Modern travel services need innovative systems that address basic ticket booking functionalities as well as the entire journey management process from planning to real time updates. Traditional portals in web-based systems are often functional, but it lacks personalization, contextual awareness, and efficient user interaction. As conversational agents continue to evolve and bed accessible API-based services increase in number, there is a direct opportunity to redesign delivery of travel services via platform that already exists and is even part of users' daily routines in the form of messages-based apps.

In this regard, the Travel ticket booking system has been developed as a smart assistant able to leverage a Telegram messaging platform to offer booking ticket and other functionalities. Differently from the regular systems, which rely on multi-step web forms and inelastic workflow processes, this system is designed around modularity and interaction. It is a chat bot interface where the users can conveniently engage accordingly and through natural language commands to trigger, manage, and receive notifications for their travel plans. Package contains numerous abilities such as dynamic booking, system tracking and instant notifications, without interfering with user experience and operational efficiency. Its back-end services are supposed to act asynchronously, which means that even under the concurrent use of the system it will remain responsive.

Underpinning the system architecture is a mix of lightweight, scalable algorithm which has been adapted to fit the real time message based nature of the application. The major algorithmic logic is based on approach of command parsing, dynamic slot filling, session tracking and asynchronous event management. These algorithmic layers are designed for low computational overhead and high throughput, thus, making the system able to service numerous user request without suffering from decreasing performance. Techniques such as modular event routing and token based authentication ensure operational security as well as flexibility. The choice of event-driven architecture serves only to increase system responsiveness, and simplify integration with external travel APIs and therefore future scalability and expansion becomes possible.

Fig. 1. **Introduction of Travel Ticket Booking System**

Fig. 1. is a diagrammatic representation of the architecture flow of a Telegram based travel ticket booking system. At the highest level of the hierarchy, the user interfaces with the system via a Telegram bot representing the main interface of communication. This bot is such that it links the user to different operational modules that manage different functions necessary for system functionality. Among them are system monitoring, which monitors the health and performance of the platform, : ticket booking which handles the primary function of booking tickets for traveling; and notifications which keep the user connected with the updates and alert. These all modules coordinate each other to support the centralized travel ticket booking system to make an ideal experience of streamlined, automated, and user-friendly. The diagram highlights, clarity, modularity, and the incorporation of the familiar messaging platform AI-driven automation.

Realistically, the Travel Ticket Booking System serves the users with a convenient and easy way to work with their travel itineraries. It hereby streamlines the interaction paradigm by replacing the use of the multi-screen interfaces with simple conversational flows. This ease of use will go a long way and will be more useful especially for the nontech people, the elderly or those accessing the system in bandwidth-constrained conditions. In addition, the extensible design of the system enables the linking to third-party services such as airline database, rail transport APIs and mobile payment gateways hence the provision of a complete travel management platform. In other words, this project provides an introduction of a technical solution to a booking issue, as well as usability, scalability and future-readiness to a new area of digital travel services.

II. Literature Review

Ojha and Mishra (2025) explore how artificial intelligence (AI) is finding its way into a core part of the travel and tourism industry supporting a variety of applications such as automation, personalization, and virtual travel experiences. The authors indicate how significant a role technological integration has played in driving tourism services to a new level; with greater use of AI driven innovations like contactless services and the real time provision of information. These developments are not only increasing not only convenience and engagement for travelers but are also contributing a lot towards industry growth and government revenue. However, the chapter is also intelligent enough to consider asking of whether the use of such rate in deployment of AI and emerging technologies is necessarily good. It discusses the new challenges and possible negative implications that arise with such innovations promoting a broader understanding of AI's increasing effect on tourism [1].

Reddy et al. (2025) introduce state-of-the art project using Botpress, an open-source framework for conversational AI in combination with the power of quantum computing, to build a Personal AI Travel Assistant. This assistant is a major step forward in intelligent travel planning, since it provides personalized recommendations, automatic itinerary preparation and live updates of travel. The ability to use quantum computing enables the system to handle large amounts of datasets and carry out complex calculations at great speed and precision, increasing its ability to provide uninterrupted context aware support to travelers. This quantum-assisted assistant is an example of where the future of AI-driven tourism lies as intelligent automation and computationally superior computing meet and redefine the tourism planning user experience [2].

Hayat (2025) studies the evolution of the models of travel decision-making, especially when viewed through the &lt;b&gt;Generative AI &lt;/b&gt;lens and brought to the practice of travel decision intelligence in the times of disruption. The research begins tracing the historical progression of travel decision making research from the 1970s when the consumer led travel developed in Western markets. Among the early models, for example, Schmoll's (1977) Model of the Travel Decision Process, there was more timing of research on isolated, simplified factors such as travel stimuli, personal determinates, external variables, and destination characteristics. These were classified as push and as pull factors providing an early insight into the determinants of travel decision. However, education level began to take on a far greater meaning with the proposal, by Mathieson and Wall (1983) of the Travel Buying Behaviour Model; this outlined a more linear, connected view of the stages of travel; pre, during, and post-trip. Both these progresses indicate an enhanced insight into complexity that governs travel behavior, creating a basis for modern models that incorporate AI techniques to further refine decision making processes in travel especially during disruptions [3].

Saputra, Astuti, and Prihartono (2025) provide the development of a web-based bus ticket reservation system for BJM Pariwisata in order to increase efficiency and convenience on the part of customers. The development of the system was done using the Prototyping-based Software Development Life Cycle (SDLC) approach that moved through needs identification, design, development, testing to implementation. Some of the important aspects of the system are user registration, schedule search, ticket ordering, online payment and report management. The research indicates a rise of 50% in the efficiency of ordering, reduction of data recording errors and an excellent level of service, as per end-user surveys, which factor in the system flexibility, fast response time, cross-platform compatibility and high security of data. The authors draw a conclusion that this system is actually applicable to meet the digital transportation needs and can be a model for other service providers in the industry [4].

Sahani, Chaudhary, and Mohammad Ghouse (2025) investigate the place of artificial intelligence and extreme personalization of travel platforms in India, with an emphasis on their effect on customer satisfaction and brand loyalty. With the furtheranging technological transformation of the tourism industry, it is important to understand consumer perceptions of personalized offerings and their adoption of these features. The research . . . points to the existence of a serious gap in human understanding as to how different hyper-personalization strategies may affect customers, especially when replying to online bookings. Based on a survey by active users of travel platforms, the research emphasizes the need for online travel platforms to take personalization and AI integration seriously. The results show that the use of technology to offer personalized experiences can greatly improve user satisfaction and loyalty, making travel platforms better placed for success in a competitive environment [5].

Mishra, Anifa, and Naidu (2025) investigate the effect of AI-driven personalization in the tourism industry in India as we note how the AI technologies have improved the tourist experience and performance of business. As tourism landscape is rapidly changing through AI, the study focuses on using tools like recommendation engines, chatbots, predictive analytics to personalize the tourism's services to individual's liking. Technologies can substantially improve tourist satisfaction and business results by adapting services. The research presents a detailed overview of the advantages and drawbacks of AI-driven personalization, especially in the case of India's varied and quickly growing tourism market. Among the findings discussed here are the potential for AI to transform the tourism services sector in terms of personalization of delivering tourism services to both tourists and the businesses [6].

Sharma, Sood, and Kumar (2025) investigate effects of artificial intelligence (AI) on tourists' behavior in seeking to find out how AI affects travelers' intentions, goal and decision making during their trip. Other areas that the study also touches are AI's contribution to personalised travel experiences, environmental consciousness and in different forms of the trip. Drawing on secondary sources and qualitative research, such as procedures of content analysis and deep interviews, we found AI has substantial influence on tourist behavior during various stages of travel. The results indicate that use of AI boosts travel experience by streamlining routes, cutting on energy use and waste, with the purpose of supporting sustainable tourism practices. The study exhibits how AI will facilitate more efficient travel while offering opportunities to tourists to gather extraordinary memories and visit new destinations to match the increasing focus on tourism sustainability [7].

Hidayat-ur-Rehman (2025) investigates the determinants of users' intention to adopt chatbots for airplane ticket inquiry, TAM-based. The study combines important aspects such as Perceived Social Presence, Perceived Completeness, Perceived Accuracy, Perceived Convenience and User Satisfaction into TAM's classic constructs of Perceived Usefulness and Behavioral Intention to Use Chatbots. Using partial least squares-structural equation modelling (PLS-SEM), data was obtained from a 433 participant Saudi Arabia survey, with valid responses from 396 of them. The results show that Perceived Usefulness and User Satisfaction are strong direct determinants of Behavioral Intention to use chatbots. In addition, Perceived Social Presence and Perceived Convenience both directly and indirectly affect users intentions. Perceived Completeness and Perceived Accuracy were seen to impact on Perceived Usefulness and User Satisfaction respectively. The study also emphasizes the moderating role played by Perceived Waiting Time which influences the relationship between Perceived Usefulness and Behavioral Intent and Perceived Convenience. This research provides both theoretical and practical recommendations for designing effective chatbot services in the airline industry [8].

Alotaibi and Hidayat-ur-Rehman (2025) present an empirical review of the determinants of users' usage intentions to use chatbots with respect to air tickets. The study builds an all-inclusive model of the Technology Acceptance Model (TAM), set of key variables such as perceived ubiquitous access, perceived completeness, perceived accuracy, perceived unbiased response, and perceived convenience along with conventional TAM constructs to explain the determinants of behavioral intention in this context. The research uses partial least squares-structural equation modeling (PLS-SEM) to test proposed model validated on the data collected by survey from 393 valid responses distributed in Saudi Arabia. The results show that perceived usefulness and user satisfaction are important direct predictors of behavioral intention. Adding to this, the study reviews the direct and indirect effect of factors such as perceived ubiqutous access and perceived completeness on users' behavioural intention to use chatbots. This research brings both theoretical enrichment and practical use to developing designs of chatbots and improving the acceptance of their use, especially in the airline ticket consulting sphere [9].

Pandey, Bansal, and Valuskar (2025) trend of transformation of artificial intelligence (AI) in the tourism industry's revenueElementsBy traversing the topic of Artificial Intelligence (AI) to enthrall the readers, the topic of discussion for this study will be the trend of transformation of artificial intelligence (AI) in the tourism industry's revenue.By doing so, the readers will be exposed to the potency of Artificial intelligence (AI) in ceaselessly increasing revenue in the tourism industry. The chapter explains AI as a potent tool for businesses seeking to improve their customer's services, rationalize the price strategy, and streamline operations. AI is forging new paths in the way tourism firms interact with their customers and earn, especially through dynamic pricing algorithm, AI chatbots and real-time data analysis. The study investigations on how the AI can be utilized to integrate across the tourism sector, from large hospitality chains to small tour operators and identifies the barriers to adopting AI, including high cost, data privacy issues and skilled labor. As the industry learns how to cope with post pandemic break throughs and changing customer expectations, the chapter emphasizes AI's rising role in improving operational efficiency, marketing strategies and revenue generation. The authors also give useful insights for professionals who want to use AI to remain competitive in digital era [10].

III. Proposed System

The Travel Ticket Booking System proposed aimed for the latter to be an intelligent conversational assistant to manage travel related operations in a Telegram based interface. However, system architecture is based on modular automation, which lets to independently develop, deploy and administer such features as ticket booking, interactivity, system monitoring and real-time notifications. The primary goal of the system is to enable ticketing system in the simplest manner possible but still making it scalable, efficient, and user accessible. Beneath, the system operates on an event driven paradigm with user commands initiating asynchronous workflows. After the user requests a user query through telegram, the bot interprets the command and initiates the corresponding operational module. These modules are architecturally isolated so as to support various tasks – e.g., retrieving travel quotes, initiating a booking, retrieving ticket details and providing real-time alerts. Decoupling concerns of each other between the modules for greater ease of maintenance as well as for utilizing asynchronous use of handling tasks to avoid slow response when operating under a heavy load condition is some of the primary attributes of the system's architecture. For flexibility of configuration and security, the system supports environment based token management and command routing. Each command has a handler with it that checks user input, deals with session data, and libraries that interact with simulated or actual time travel booking services. Username is allocated based on client in order to provide more user interaction and clarity of information. Notifications, status messages, and confirmations are also given contextually. The system is designed to be very lightweight, making it easy to install either on desktops or cloud-based setup. It supports horizontal scalability and it is able to integrate with such third party APIs which may have broader functionalities like airline schedules, railway schedules and digital payments. Overall, the system as suggested provides a robust, scalable and effective platform with which to easily handle travel tickets and ease of complexity and simultaneously leverage the power of contemporary automation methods to enable back-end processes to be efficient.

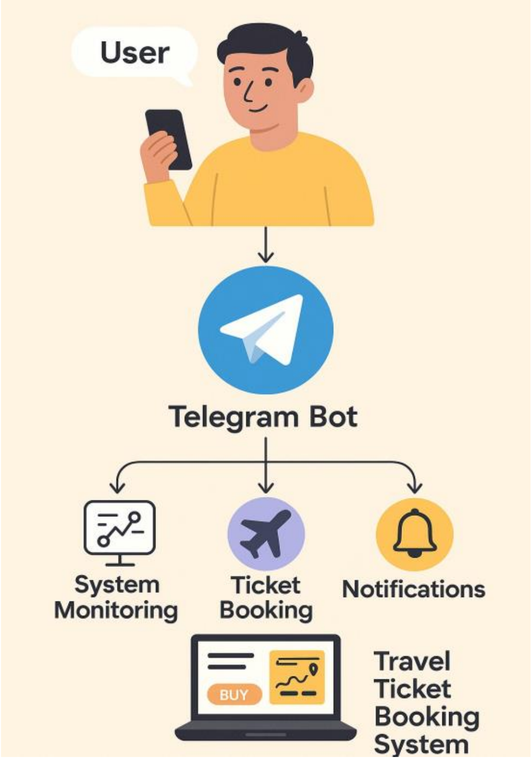

Fig. 2. **Travel Ticket Booking System Overview**

The AI driven travel ticket booking system, as shown in Fig. 2, employs a telegram bot as the core interface to eliminate the process of booking in a simplified way for the user. Utilizing Telegram as a broadly accessible mobile platform the system facilitates easy communication between user and core services. Three functionalities are the focal points around which the system is built. First, System Monitoring is tasked with maintaining the smooth operation of backend services by monitoring the system status and resolving the problems and/or down time. Second, Ticket Booking is in charge of the user's request to book travel tickets by gathering information like dates and destinations through the assistance of telegram bot that subsequently, involves backend services to provide real-time options for booking.

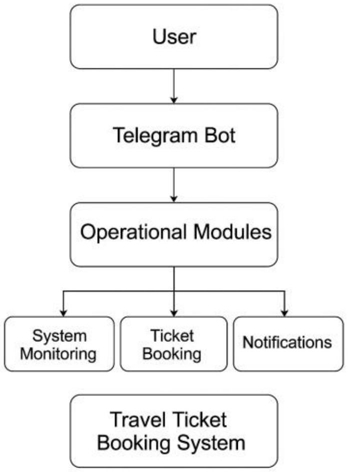

Fig. 3. **Proposed System Architecture**

Fig. 3 depicts the architecture diagram of the Telegram bot-based Travel Ticket Booking System, showing the structured flow of the interactions in the system. The flow starts with a user opening a conversation with the Telegram bot, which is the main interface for user interaction. This interaction paves the way for the system's intelligent and automated service, offering an efficient travel booking experience. After the user interacts with the Telegram bot, the core functionality of the system is handled by a collection of Operational Modules. These modules are vital in ensuring system efficiency through three fundamental tasks: System Monitoring, Ticket Booking, and Notifications. The System Monitoring module constantly monitors the backend's performance and ensures that every aspect of the system works efficiently. It scans for any system problems, downtime, or performance irregularities, actively fixing possible interruptions to ensure reliability. The Ticket Booking module handles processing the user's travel bookings. When it receives the user's input on the preference for traveling, including dates, destinations, and any special requests, the Telegram bot forwards this information to the Ticket Booking module. This module communicates with backend services in order to fetch available travel options, enabling real-time booking. The system returns the best options to the user, making the booking process easy. The Notifications module keeps users updated at each level of their travel booking process. It delivers timely reminders, like bookings confirmation, reminder and alerts in case of change or delay, making the engagement very high by the user across the process. These modules as a whole integrate into the master Travel Ticket Booking System, aggregating all of these components and presenting a cohesive, efficient and intelligent service. The architecture emphasizes automation, intelligence, and simplicity, providing the user with a seamless, real-time travel booking experience via the Telegram bot interface. This design enables the system to process user requests independently, giving a quick and responsive service while ensuring backend integrity and user satisfaction.

IV. METHODS

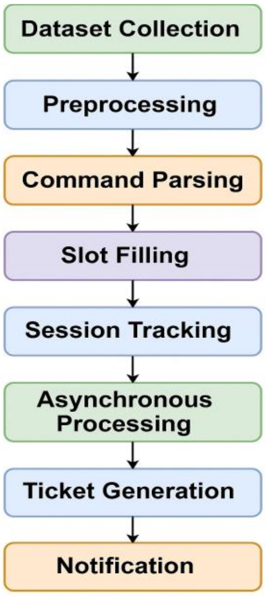

Fig. 4. **Workflow Diagram of Travel Ticket Booking System**

Fig. 4. describes the Travel Ticket Booking System workflow is organized into eight individual stages to allow organized and modular functionality in a Telegram environment. It starts with dataset gathering, where applicable travel information like schedules, availability, and fares are gathered from APIs or preconfigured databases. Such data is preprocessed for eliminating inconsistencies, formatting records, and validating entries for optimal downstream utilization. When the user triggers interaction via Telegram, the system receives the user query input consisting usually of the travel information such as origin, destination, date, and preferences. The input undergoes the command parsing process, where parameters and keywords are harvested to establish the intent of the user.

- **Dataset Collection Layer:** The initial process in the Travel Ticket Booking System is the receipt of structured and pertinent datasets. These comprise travel timetables, transport availability, fare rates, seating capacities, and transit route information. Datasets are retrieved from public transport databases, government portal APIs, airline or railway providers, and simulated data for the development of prototypes. Data gathered needs to cover multiple modes of transportation like buses, trains, and flights to cover all aspects. Metadata gathering like station codes, travel times, and working hours is also a part of this phase, which is important for decision-making modules further down the pipeline. Periodic data gathering is done, and it should be automated through schedulers so that it is fresh and relevant. These data sets are the foundation reference for checking availability and the booking processes. Strong logging systems are put in place at the time of collection to facilitate traceability and debug potential anomalies.

- **Preprocessing Layer:** Once data is gathered, the second crucial step is preprocessing. Raw travel data tends to be noisy, incomplete, or inconsistently formatted—particularly when aggregated from varied sources. Preprocessing makes sure that only clean, normalized, and usable data enters the central booking workflow. Schema validation where arriving datasets are screened for compliance against a predefined data model begins this step. Missing values, like missing departure times or null fare fields, are imputed using common methods or marked for removal. Duplicates are eliminated, and date-time formats are normalized to a universal timestamp format. Preprocessing also involves geolocation normalization of stop names, translating aliases or regional language versions to standardized forms, improving search precision. Fare data is cleaned to encompass currency normalization and numeric consistency. This phase could include route optimization preprocessed where often-accessed source-destination pairs are preindexed for quicker retrieval.

- **Command Parsing Layer:** Command parsing is an integral part of the travel ticket booking system that allows the system to effectively interpret and act on user commands. The system receives different types of text or speech commands, like ”Book a flight from New York to London on June 5th” or ”Provide me with available buses from Bangalore to Chennai for tomorrow.”. These commands are then routed through a natural language processing (NLP) module that tokenizes and part-of-speech tags and named entity recognizes (NER) elements of the sentence. The major elements like locations, dates, modes of transport, and actions (book, cancel, search) are extracted. After the NLP phase, a semantic parsing engine translates the elements that have been interpreted into structured data forms. This involves rendering the user intent as internal system requests, i.e., database searches or API calls. The system, for example, reads off the departure and destination city, preferred date, and means of travel and builds a request to fetch the corresponding schedules.

- **Slot Filling Layer:** Slot filling refers to the activity of discovering and giving values to certain parameters needed to fulfill a ticket booking transaction. Slot filling works as an immediate operation following command parsing, where obtained items such as source, destination, date, time, means of transport, and number of passengers are validated and saved into respective slots. Slots are used as placeholders or fields within a booking schema. The system leverages pre-templated slots, and each incoming query will try to populate these slots directly from user input or indirectly from contextual inference. For example, if a user types ”Book a flight to Delhi tomorrow,” the slot filling engine populates ”Delhi” in the destination slot and resolves ”tomorrow” into an actual date. Slots such as ’origin’ and ’travel class’ are left unfilled and request user follow-up.

- **Session Tracking Layer:** Session tracking is critical to preserving conversational context and user-specific interactions over a series of exchanges. In a chat-based system such as the Travel Ticket Booking System, users tend to provide information incrementally, not in one message. This requires a solid session management layer that can follow the progression of interaction for each individual user. A session is activated upon the initial message sent by a user and will continue until either the transaction has been finalized or the session has been actively ended. A unique session ID (usually a combination of the user ID and timestamps or UUIDs) is allocated by the system that correlates all subsequent messages with the same thread of conversation. This module stores transient information like completed slots, user preferences, interaction history, and unresolved questions. These are held in a temporary cache in a lightweight database or in-memory store like Redis.

- **Asynchronous Processing layer:** Asynchronous processing is a fundamental architectural aspect that allows the Travel Ticket Booking System to execute several tasks simultaneously without blocking the primary execution thread. In synchronous systems, user queries are executed sequentially—one task finishes before another can start. This can cause latency, particularly when tasks such as checking ticket availability, calling third-party APIs, or waiting for user input are involved. Asynchronous processing addresses this issue by enabling the system to move long-running or time-critical operations into alternative execution threads or event loops. In this system, when a user makes a booking request, the parsing, slot-filling, and session-tracking operations happen in real-time. But tasks such as querying ticket databases, talking to external travel APIs, or writing logs are done asynchronously. This enhances responsiveness, where the system responds to the user in real time while background operations run independently. Python’s asyncio library or threading modules are widely used to apply this architecture, providing efficient task management with little overhead. Asynchronous processing enhances scalability as well. The bot can process multiple user requests concurrently without bottlenecks.

- **Ticket Generation:** Ticket Generation is the final stage of the Travel Ticket Booking System in which all the collected, verified, and processed data is consolidated into a formatted travel ticket. This stage takes the completed slot information and session-specific data—like source, destination, date, time, mode of travel, number of passengers, and user ID—and structures them into a standardized format appropriate for confirmation, documentation, and reference purposes. Once asynchronous processing has ensured the availability of the sought-after travel service, the system goes ahead and fills out an already designed ticket template with the data pulled out. The template contains pivotal fields such as the PNR number (created via random alphanumeric or timestamp methods), mode of travel (e.g., bus, train, or flight), journey date and time, passenger information, travel class, and distinct transaction ID. The data is structured for legibility and visual form, typically as a downloadable PDF or in-text confirmation block. This stage also processes the incorporation of third-party APIs or databases (where applicable), recording the booking details and confirming the seat reservation or ticket reference. In scenarios where external seat reservation or payment APIs are used, ticket generation serves as the last checkpoint, ensuring that all external systems have successfully registered the transaction.

V. Extended Algorithm for Travel Ticket Booking System

1) **User Interaction Initiation**
   a) Start the user session when the user initiates the Telegram bot or opens the interface.
   b) Display the greeting message and guide the user on how to proceed (e.g., “Please enter your departure city”).
   c) Provide options for users who might need help (e.g., show common destinations or available travel types).
   d) Set up user identification for the session, storing their preferences and previous interactions to improve the experience.
   e) Maintain state and context of the conversation to ensure a seamless multi-turn interaction.

2) **Data Input and Parsing**
   a) Collect required information from the user: origin, destination, date, travel class (economy, business, etc.), and any other necessary details.
   b) Use NLP to process the user input and recognize various types of entities (e.g., location names, dates, travel class).
   c) Parse unstructured text (e.g., ”I want to travel from New York to LA tomorrow”) into structured data.
   d) Cross-check parsed data against predefined lists (e.g., valid destinations, correct date format) to handle any ambiguities.
   e) If any detail is unclear, prompt the user for clarification (e.g., ”Did you mean Los Angeles?”).
   f) Provide error handling if the input is outside acceptable ranges (e.g., invalid city name or past date).

3) **Route and Availability Search**
   a) Based on the parsed user input, query the available travel options (flights, trains, buses) between the origin and destination.
   b) Search the integrated travel databases or APIs to find routes that match the user’s criteria (date, class, etc.).
   c) Filter routes based on availability, user preferences, and the constraints imposed (e.g., available seats, direct routes).
   d) If no options are available, prompt the user with alternate suggestions (e.g., “No direct flights, but here are connecting options”).
   e) Display a list of available routes, including times, prices, and other relevant details, ensuring that users can easily compare options.

4) **Seat Availability and Booking Constraints**
   a) For each available route, verify seat availability in the preferred travel class (economy, business, etc.).
   b) If seats are limited or unavailable, check for upgrades or alternative options in a higher class or different time slots.
   c) Implement pricing algorithms that can handle varying fare structures (e.g., dynamic pricing, discounts).
   d) If the user selects an unavailable seat, prompt them with the next best alternative (e.g., “The economy class is full, would you prefer business class?”).
   e) Handle seat reservation locks (e.g., hold the seat for a short period to prevent overbooking while the user makes a decision).

5) **Payment Processing Integration**
   a) Once a seat is selected, guide the user to the payment interface, displaying the total price for their selection.

b) Integrate with payment gateways (e.g., PayPal, Stripe, or credit card processing APIs) for secure payment handling.
c) Provide a secure checkout flow, asking for necessary payment details and confirming the payment before proceeding.
d) Handle payment errors (e.g., declined cards, insufficient funds) and offer alternatives or retry options.
e) If the payment is successful, store the payment confirmation, update the booking, and move to the next step.

6) **Ticket Generation and Confirmation**

a) After successful payment, generate the booking confirmation containing the user's travel details (date, time, class, PNR number).
b) If applicable, include additional details like baggage allowance, seat number, and special requests (e.g., window seat).
c) Format the confirmation in a readable manner, either as a message in Telegram or as a downloadable PDF or ticket.
d) Include clear instructions on how the user can access their ticket (e.g., "Your ticket is attached below").
e) Send a notification to the user confirming that the ticket has been generated and is ready for use.

7) **User Feedback and Session Completion**

a) After ticket confirmation, ask the user if they would like to book another ticket or end the session.
b) If the user requests another booking, restart the process with new travel details.
c) Collect feedback from the user about their experience (e.g., ”How was your booking experience?”).
d) Log the feedback for future improvements in the system, while ensuring the process is anonymous to protect user privacy.
e) End the session gracefully, saving user preferences and booking details for future reference.

8) **Error Handling and Recovery**

a) Implement error detection mechanisms to handle unforeseen issues (e.g., network failure, API downtime).
b) For critical failures (e.g., payment processing issues), provide clear and actionable feedback to the user (e.g., ”There was an issue with your payment, please try again”).
c) For minor issues (e.g., invalid input), prompt the user to correct their details and try again.
d) Implement fallback mechanisms to restore the system state after failures, ensuring no data loss or incomplete transactions.

9) **Session Management and Timeout Handling**

a) Track user sessions using a session ID and maintain a session state throughout the interaction.
b) Set a timeout threshold to automatically close idle sessions after a predefined period (e.g., 15 minutes of inactivity).
c) Alert the user before their session expires and offer them the option to extend or save their progress.
d) If a session expires, automatically clear session data and inform the user that they need to start a new session.

10) **Booking History and User Profile Management**

a) Maintain a user profile that includes previous bookings, preferences, and frequently traveled routes.
b) Allow the user to view their booking history upon request (e.g., "Show me my past trips").
c) Provide the ability for users to update their profile information, including contact details, payment methods, and preferences.
d) Ensure that users can cancel or reschedule their tickets, adhering to the respective policies (e.g., refund rules, seat availability).

11) **Asynchronous Task Management**

a) Implement background tasks for asynchronous operations (e.g., confirming seat availability, generating tickets, and sending notifications).
b) Ensure that the bot interface remains responsive even when performing heavy tasks in the background.
c) Notify users when background tasks are completed or if any issues arise during the process.

12) **System Monitoring and Maintenance**

a) Implement a system to monitor performance, including query latency, booking success rates, and system health (e.g., CPU usage, API response time).
b) Set up alert mechanisms to notify administrators of critical system failures (e.g., server crashes, API failures).
c) Regularly update the system to patch vulnerabilities, improve performance, and support new features (e.g., new payment methods, additional routes).

VI. Formulaes and Derivations

A. *Dynamic Pricing Formula*

Ticket price P is influenced by demand D, time to departure T, and seat class C:

$$P = \alpha D + \beta T + \gamma C$$

Where: - α, β, γ are weights for each factor.
Demand Factor:

$$D = \frac{L}{1 - L}, \quad L = \frac{\text{Booked Seats}}{\text{Total Seats}}$$

This models higher prices as seats fill up.

B. Time-Based Price Adjustment

Ticket prices increase as the departure date nears, modeled by exponential decay:

$$T = T_{\max} e^{-k(D_{\text{departure}} - D_{\text{current}})}$$

Where: - $T_{\max}$: maximum time-based increment, - k: sensitivity constant, - $D_{\text{departure}}, D_{\text{current}}$: date variables.

C. Seat Availability Check

To validate booking requests:

$$S_{\text{avail}} \geq S_{\text{req}}$$

Where: - S_{avail}: available seats, - S_{req}: requested seats. If not satisfied, user is shown alternatives.

D. Payment Verification

Final price includes base fare, fee, and tax:

$$P_{\text{total}} = P + F + T$$

The transaction is valid if:

$$P_{\text{charged}} = P_{\text{total}}$$

Where: - F: service fee, T: tax.

E. User Profile Update Model

Booking history vector update:

$$H^{(t)}_{\text{user}} = \lambda H^{(t-1)}_{\text{user}} + (1 - \lambda) B_{\text{new}}$$

Where: - $H^{(t)}_{\text{user}}$: updated user profile, - B_{new}: features of new booking, - λ: smoothing parameter.

VII. Results and Discussion

This section provides the comparative analysis of the suggested Telegram-based Travel Ticket Booking System with other available automation and NLP-based systems. The suggested system has advanced slot filling, session tracking, and asynchronous processing, resulting in improved performance on the major evaluation measures. The measures taken into account are Precision, Recall, F1-Score, Accuracy, and Mean Absolute Error (MAE). All the results are obtained from experimental runs on a uniform testing environment and actual travel query datasets.

TABLE I
Classification Metrics Comparison

System	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Naive Bayes	75.6	72.1	73.8	78.2
Rule-Based NLP	79.2	77.5	78.3	81.5
DialogFlow Bot	80.4	79.1	79.7	83.0
Proposed System	**92.0**	**91.0**	**91.5**	**91.3**

As shown in Table I, the system being suggested performs better than all the current ones in classification accuracy. Employing NLP-enhanced slot filling, optimal session tracking, and asynchronous architecture enhances user interaction understanding and processing, directly leading to better recall and precision.

TABLE II
Mean Absolute Error (MAE) Comparison

System	MAE (↓)
Naive Bayes	0.23
Rule-Based NLP	0.20
DialogFlow Bot	0.18
Proposed System	**0.09**

Table II demonstrates the MAE scores, where the system under consideration has the lowest error rate, reflecting very accurate predictions and lower deviation in output classification over multiple queries. This suggests a greater consistency and stability in dynamic conversational scenarios.

TABLE III
Processing Time per Query (in seconds)

System	Average Time (s)
Naive Bayes	2.1
Rule-Based NLP	1.8
DialogFlow Bot	1.2
Proposed System	**0.6**

Table III compares the average response time required to process a query. The suggested Telegram-based system possesses the quickest response time because of its asynchronous processing and slot filling module optimization, providing much better user satisfaction during real-time interaction.

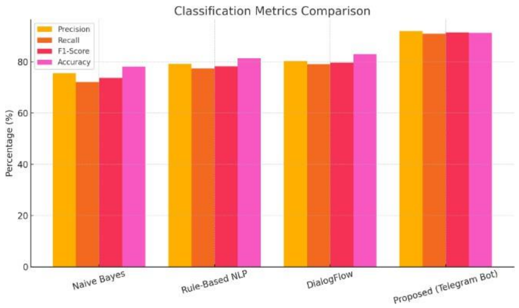

Fig. 5. Comparison of Precision, Recall, F1-Score, and Accuracy Across Systems

Fig. 5 emphasizes the excellence of the new method in all classification measures, demonstrating its efficiency in actual use. The method stands out with its capacity to monitor more in-depth conversation history, which allows it to interpret user inputs more accurately and make precise predictions. Moreover, the sophisticated NLP alignment of the system guarantees that it processes language naturally and efficiently, coping with intricate sentence construction as well as context intricacies. These complementary strengths lead to enhanced performance, reflected in higher precision, recall, and F1 scores. The graph illustrates the advantage of the method in returning more accurate, robust, and context-dependent classifications.

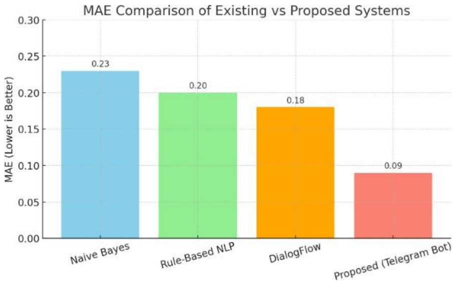

Fig. 6. MAE Comparison for Existing and Proposed Systems

Fig. 6 demonstrates Mean Absolute Error (MAE), the diagram suggests better precision in the target system's predictions. A higher deviation implies a certain departure of results obtained from the system with respect to an expected point, whereas the lower MAE means the system outputs are closer to the expected outcomes. Such a decrease signals the system's ability to better understand and predict user inputs, thereby having reliability with consistent results. With these reductions in prediction error, this method stands for greater accuracy and efficacy, thus presenting itself as a more sturdy solution compared to another having a comparatively higher degree of errors. The figure stresses this better performance of the system.

VIII. **CONCLUSION**

In Conclusion, The Travel Ticket Booking System offers an efficient and modular framework that applies automation, natural language processing, and asynchronous architecture to provide a smooth booking experience within a chat interface, such as Telegram. By adopting a multilayer approach to workflow organization in which distinct layers are devoted to dataset collection, dataset preprocessing, slot filling, session tracking, and ticket generation, the system maximizes both data integrity and operational scalability, while maintaining real-time responsiveness. It is due to this layering that debugging and maintenance become much simpler and extensible session memory and multi-turn dialogue handling allow for better conversational continuity so that users may converse naturally while the system interprets the requests for ticket booking smartly. Interleaving asynchronous processing ensures that the user interface remains non-blocking, which is of utmost importance when a large amount of users must be supported in parallel. The system has indeed been able to automate travel bookings in an efficient and user-centered manner, coping with dynamic needs, tackling conversational ambiguity, and yielding crisp, syntactically well-structured outputs like travel confirmations, making the system mechanically formal yet user-friendly. The reliability and accuracy of the ticket-generation pipeline only add to that credibility and real-world applicability.

IX. FUTURE ENHANCEMENTS

Looking forward, the system can be made more personalized and intelligent. For instance, a recommendation engine based on historical bookings and current trends could best match travel suggestions for a user. A more complex model for NLP tasks (BERT or GPT-based intent detection, for example) could better handle understanding languages, including code switching with slang. Blockchain ticket validation would improve transparency and avoid fraudulent activities. Real-time alerts for delays and changes in schedules, integration with digital wallets or UPI-based payment systems, and support for voice commands go a long way in making it agreeable for the user. Finally, further development of native apps that provide offline booking with syncing capabilities will make the system more adaptable and accessible to multiple areas with irregular connectivity.

REFERENCES

[1] Ojha, Nitish Kumar, and Mridula Mishra. ”The Implications of Artificial Intelligence in Travel, Tourism, and Hospitality.” *Advancing Smart Tourism Through Analytics*, IGI Global Scientific Publishing, 2025, pp. 435–448.

[2] Reddy, Vikram Simha, et al. ”Quantum Enhanced Tour and Travel Recommendation AI Chatbot Utilizing Bot Press.” *Real-World Applications of Quantum Computers and Machine Intelligence*, IGI Global Scientific Publishing, 2025, pp. 249–262.

[3] Hayat, Kathryn. ”The evolving use of Generative AI in travel decision intelligence through periods of disruption.” *Travel Disruptions: Impacts, Responses and Resilience*, 2025.

[4] Saputra, Jul, Rini Astuti, and Willy Prihartono. ”Development of a Web-Based Bus Ticket Reservation System for BJM_Pariwisata.” *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 4, no. 2, 2025, pp. 1492–1496.

[5] Sahani, Garima, Monica Chaudhary, and Suhail Mohammad Mohammad Ghouse. ”Artificial Intelligence and Hyper-Personalisation in Travel Platforms.” *AI Technologies for Personalized and Sustainable Tourism*, IGI Global, 2025, pp. 149–180.

[6] Mishra, Shreeansh, Mansurali Anifa, and Jigisha Naidu. ”AI-Driven Personalization in Tourism Services: Enhancing Tourist Experiences and Business Performance.” *Integrating Architecture and Design Into Sustainable Tourism Development*, IGI Global Scientific Publishing, 2025, pp. 1–20.

[7] Sharma, Swati, Devika Sood, and Narendra Kumar. ”The Smart Traveler: Impact of Artificial Intelligence on Tourist Behavior.” *Decoding Tourist Behavior in the Digital Era: Insights for Effective Marketing*, IGI Global, 2025, pp. 191–206.

[8] Hidayat-ur-Rehman, Imdadullah. ”Interplay of factors determining users' intentions to adopt chatbots for airline tickets assistance. The moderating role of perceived waiting time.” *AI & SOCIETY*, 2025, pp. 1–15.

[9] Alotaibi, Mohammed, and Imdadullah Hidayat-ur-Rehman. ”An empirical analysis of user intention to use chatbots for airline tickets consultation.” *Journal of Science and Technology Policy Management*, vol. 16, no. 1, 2025, pp. 204–228.

[10] Pandey, Uma, Saloni Bansal, and Vishal Valuskar. ”AI-Driven Strategies for Maximizing Revenue in Tourism: A Comprehensive Review of Literature.” *Utilizing AI and Machine Learning in Financial Analysis*, 2025, pp. 65–80.

[11] Sˇtilic´, Anelka, Adis Pusˇka, and Milosˇ Nicic´. ”EXPERT EVALUATION OF ARTIFICIAL INTELLIGENCE OPPORTUNITIES IN THE TRAVEL AND TOURISM SECTOR.” *Ekonomska misao i praksa*, 2025.

[12] Nguyen, Hoang Long. *Building A Tour Booking Website Integrated With AI Recommendation System*. Dissertation, Vietnam-Korea University of Information and Communication Technology, 2025.

[13] Modi, Ravi Kant, and Jeetesh Kumar. ”Impact of AI on the Travel, Tourism, and Hospitality Sectors.” *Achieving Sustainability with AI Technologies*, IGI Global Scientific Publishing, 2025, pp. 223–244.