iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XII Month of publication: December 2025

DOI: https://doi.org/10.22214/ijraset.2025.76656

SAFARSETU: An AI-Powered Multilingual Tourist Guide Chatbot for Cultural Heritage Exploration

Anurag Shrivastava[1], Shivang Agrawal[2], Sanjana Keshari[3], Krishna Vishwakarma[4], Mohd. Taukeer[5]
[1]*Head of Department, Computer Science and Engineering , BBDNIIT, Lucknow, India*
[2, 3, 4, 5]*Computer Science and Engineering, BBDNIIT, Lucknow, India*

Abstract: India's architectural heritage, particularly the vernacular Havelis (traditional mansions), faces a critical dual challenge: physical deterioration and a fading public narrative. While the "Digital India" initiative has accelerated technology adoption, existing heritage tourism solutions often rely on high-bandwidth mobile applications or English-centric interfaces, creating a "digital divide" that excludes a vast demographic of domestic rural tourists. This paper proposes SAFARSETU, a serverless, zero-install, multilingual chatbot built on the Telegram platform designed to democratize access to heritage information. Unlike traditional synchronous bots, our architecture employs a novel Asynchronous NLP Pipeline utilizing Python's asyncio library to parallelize Neural Machine Translation (NMT) via Google Translator and Neural Text-to-Speech (TTS) via Edge-TTS. We introduce a custom "Sentence-Aware Chunking Algorithm" to overcome Telegram's 4096-character message limit without disrupting semantic continuity or audio prosody. Experimental validation across 500+ interaction cycles demonstrates that the system maintains a Total Response Time (TRT) of <1.8 seconds for complex multilingual queries, significantly outperforming synchronous architectures. SAFARSETU offers a scalable, inclusive reference architecture for digital heritage tourism in resource-constrained environments.

Keywords: Cultural Heritage, Natural Language Processing, Asynchronous Computing, Neural Text-to-Speech, Serverless Architecture, Heritage Tourism, Telegram API.

I. INTRODUCTION

The preservation of cultural heritage has transcended beyond the physical restoration of monuments to encompass the digital dissemination of historical narratives. In the Indian subcontinent, the *Havelis*—traditional mansions characterized by their intricate frescoes, *jharokhas* (overhanging enclosed balconies), and central courtyards—represent a unique synthesis of Islamic, Persian, and Rajasthani architectural styles. However, unlike major UNESCO World Heritage sites, these vernacular structures face a critical dual challenge: physical deterioration due to urbanization and a fading public narrative due to a lack of accessible documentation [12]. In the wake of the "Digital India" initiative, the tourism sector has witnessed a paradigm shift towards "Smart Tourism." Technologies such as Augmented Reality (AR), Virtual Reality (VR), and dedicated mobile applications have been proposed to enhance visitor engagement [2]. While technically impressive, these solutions often inadvertently widen the "digital divide." A significant portion of domestic tourists in India belongs to the "Next Billion Users" demographic—users characterized by entry-level smartphones, limited storage capacity, and intermittent data connectivity. Dedicated heritage web applications, such as those proposed by Bhande et al. (2025), often require high-bandwidth downloads and modern GPUs for rendering, creating a high barrier to entry for rural tourists [6].

Furthermore, a critical linguistic gap exists in current digital heritage platforms. Research indicates that while 68% of India's internet users prefer content in their native dialects, the vast majority of digital tourism interfaces remain English-centric [13]. This linguistic exclusivity alienates rural tourists and local communities, reducing heritage exploration to a visual-only experience devoid of historical context. Existing chatbots in this domain, such as "TN Forts Buddy" for Tamil Nadu's forts, have attempted to bridge this gap but rely heavily on static, menu-driven interactions that lack dynamic translation or immersive audio storytelling capabilities [1].

To address these systemic limitations, this paper proposes SAFARSETU (*Journey Bridge*), an AI-powered, multilingual tourist guide accessible via the Telegram messaging platform. We selected Telegram for its robust API, superior compression algorithms for multimedia, and its ability to function effectively in low-latency (2G/EDGE) network environments. Unlike traditional chatbots that process requests synchronously (blocking the user interface while fetching and processing data), SAFARSETU introduces a novel Asynchronous NLP Pipeline.

By leveraging Python's asyncio library, our architecture parallelizes the resource-intensive tasks of Neural Machine Translation (NMT) and Neural Text-to-Speech (TTS) synthesis. This approach decouples content retrieval from content generation, allowing the system to deliver rich multimedia narratives in English, Hindi, and Urdu with minimal latency, addressing the performance bottlenecks identified in server-based architectures [9]. This paper presents the following key contributions: Serverless Asynchronous Architecture: A scalable design that utilizes non-blocking I/O to handle concurrent multilingual requests, reducing total response time compared to synchronous baselines used in previous studies like the Semarang Heritage Bot [4]. Semantic Chunking Algorithm: A custom text segmentation strategy designed to overcome Telegram's 4096-character message limit [5] without disrupting the semantic flow or audio prosody of historical narratives. Vernacular Accessibility: The integration of edge-optimized Neural TTS models to provide human-like audio descriptions, making heritage information accessible to visually impaired users and those with limited literacy [8]. The remainder of this paper is organized as follows: Section II reviews related work in AI-driven tourism; Section III details the system architecture and mathematical modeling of the asynchronous pipeline; Section IV describes the implementation of the chunking and translation algorithms; Section V presents experimental results on latency and user satisfaction; and Section VI concludes with future research directions.

II. RELATED WORK / LITERATURE REVIEW

The integration of Artificial Intelligence (AI) into the tourism sector has catalyzed a paradigm shift from static information dissemination to dynamic, user-centric experience design. This section critically analyzes existing literature to identify the technological and accessibility gaps that SAFARSETU addresses.

A. *Generative AI and Machine Learning in Tourism*

The application of machine learning in tourism has evolved significantly over the last decade. A 2025 systematic review by *Deng* analyzed the evolving trends in experience design, revealing a transition from "object-centric" digitization (e.g., archival preservation) to "user-centric" engagement [5]. Technologies such as Convolutional Neural Networks (CNNs) and Natural Language Processing (NLP) are increasingly employed to personalize travel itineraries and analyze visitor sentiment. However, the deployment of these technologies in the cultural heritage domain remains fragmented. *Deng* highlights a critical "digital divide," noting that unequal access to technological infrastructure in developing regions remains a barrier to adoption [5]. While Large Language Models (LLMs) offer rich narrative capabilities, their high computational requirements often render them unsuitable for real-time deployment in resource-constrained environments, creating a need for lighter, more efficient architectures.

B. *Chatbots in Cultural Heritage: State of the Art*

Chatbots have emerged as a preferred interface for digital heritage due to their conversational nature. *Sathiyabamavathy and Anju (2024)* developed "TN Forts Buddy," a chatbot designed for Tamil Nadu's forts and palaces [1]. Their empirical study demonstrated that chatbots significantly improve visitor engagement by providing location data and historical context compared to traditional brochures. Similarly, the "Heritage Information Chatbot" proposed by *Deepa et al. (2025)* utilizes NLP to answer user queries about historical sites [2]. Despite these advancements, a critical analysis reveals significant limitations in existing systems: Static Interaction Models: Systems like "TN Forts Buddy" rely primarily on menu-driven interactions (e.g., button clicks) rather than free-flowing conversation [1]. They lack the capability for dynamic, on-the-fly translation, limiting their utility to pre-defined languages. Lack of Vernacular Support: Most existing bots operate within a predefined linguistic scope (typically English). Research by *Reddy and Kumar (2024)* acknowledges that while local language support is desired, current implementations often fail to deliver accurate, context-aware translations for diverse Indian dialects [3]. Absence of Audio Narratives: Current implementations are predominantly text-based. *Benaddi et al. (2025)* argue that voice-driven systems are essential for heritage tours, yet most bots exclude visually impaired users or those with lower literacy rates, for whom audio storytelling is the primary medium of cultural consumption [6].

C. *The Technical Gap: Synchronous vs. Asynchronous Architectures*

A major bottleneck in current tourism chatbots is their reliance on synchronous (blocking) architectures. In traditional web-based heritage apps, the user interface often "freezes" while waiting for server responses. *Hu et al. (2025)*, in their analysis of serverless LLM serving (DeepServe), highlight that asynchronous event-driven architectures offer superior performance for bursty workloads [9]. For a tourism chatbot, where a user might request a translation (Service A) and an audio file (Service B) simultaneously, a synchronous model would force the user to wait for the sum of both tasks. In contrast, SAFARSETU adopts the asynchronous patterns advocated by *Alekseev et al. (2024)* for Telegram bot development, allowing tasks to execute in parallel and significantly reducing the "Perceived Latency" for the end user [10].

D. The Digital Divide & Linguistic Accessibility

The "Digital Divide" in India is not just about connectivity but also about language. The *Ministry of Tourism (2024)* statistics indicate that 68% of domestic tourists prefer content in their native languages [13]. While initiatives like *BharatGen* are building foundational models, there is a lack of application-layer tools that effectively bridge this gap for tourists. Furthermore, *Resemble AI (2024)* emphasizes the importance of regional accents in AI voice generation for preserving cultural authenticity [8]. Generic text-to-speech engines often sound robotic and culturally disconnected. SAFARSETU innovates by integrating specific Neural voices (e.g., hi-IN-SwaraNeural) to ensure the auditory experience aligns with the cultural context of the *Havelis*.

E. Synthesis of Research Gaps

Table I summarizes the comparative analysis of SAFARSETU against key benchmarks in the literature.

Feature	TN Forts Buddy (2024) [1]	Heritage Info Chatbot (2025) [2]	SAFAR-SETU
Interaction Type	Static / Menu-Driven	NLP (Text Only)	Hybrid (Conversational + Menu)
Language Support	Fixed English	Limited Local Support	Dynamic (Real-time NMT)
Content Delivery	Text Only	Text Only	Multi-media (Neural TTS + Image)
Architecture	Mono lithic	Client-Server	Async/ Serverless
Accessibility	Visual Only	Visual Only	Audio-Visual

SAFARSETU addresses these identified gaps by combining the ubiquitous accessibility of messaging platforms with the computational efficiency of asynchronous Neural NLP, creating a solution that is both technically robust and socially inclusive.

III. SYSTEM DESIGN AND METHODOLOGY

The SAFARSETU system is designed as a Microservices-based Event-Driven Architecture. Unlike monolithic heritage applications that require heavy client-side processing, our system offloads computational logic to a serverless backend, prioritizing low latency and high concurrency [9].

A. High-level Architecture

The system comprises four distinct layers, optimized for the constraints of low-bandwidth networks common in rural heritage sites. Few layers like Interaction Layer (Telegram Interface) is the layer where user interacts via standard chat commands (/start, inline buttons). We utilize Telegram's MTProto protocol, which is cryptographically secure and optimized for delivery in high-latency (2G/EDGE) environments [10]. Controller Layer (Webhook) instead of "Long Polling" (which consumes constant bandwidth and server threads), we utilize a Webhook pattern. Telegram pushes updates to our Python application only when an event occurs. This "Serverless-like" behavior ensures the bot consumes zero computational resources during idle periods, significantly reducing operational costs compared to persistent servers used in previous studies like "TN Forts Buddy" [1]. Logic Layer (Async Application) is the core application logic is built on Python 3.11+, utilizing the asyncio library to manage concurrent tasks. This layer handles session state, language preferences, and orchestrates the external API calls. Data Layer is a high-performance, read-optimized JSON store acts as the "Ground Truth" for historical data, ensuring O(1) retrieval times [11]. Service Layer (External APIs) like Translation (deep_translator for real-time Neural Machine Translation (NMT)) and Audio Synthesis (edge_tts for generating high-fidelity Neural Text-to-Speech (TTS)).

B. The Asynchronous Pipeline Model

A critical innovation in SAFARSETU is the shift from synchronous to asynchronous processing. Traditional chatbots operate on a blocking I/O model:

Request -> Fetch -> Translate -> TTS -> Send

In this model, the user waits for the sum of all process times. For a heritage description involving audio generation, this latency can exceed 3 seconds, leading to user churn [9].

In SAFARSETU Asynchronous Model we implemented a Parallel Async Pipeline. As seen in our implementation, we utilize asyncio.gather() to execute the Translation and Audio Generation tasks concurrently. Furthermore, we pipeline the delivery: the Visual Content (Image + Text) is sent *immediately* while the Audio is being generated in the background.

Let Ttotal be the total response time.

tfetch: Time to retrieve text from JSON (~ 5 ms).

Ttrans: Time for Translation API (~ 400 ms).

ttts: Time for TTS synthesis (~ 1200 ms).

tnet: Network round-trip time (~ 100 ms).

Synchronous Latency Model:

$$Tsync = tfetch + ttrans + ttts + tnet \sim 1705 \text{ ms}$$

SAFARSETU Asynchronous Model: By parallelizing tasks and prioritizing visual delivery:

$$Tvisual = tfetch + ttrans + tnet \sim 505 \text{ ms}$$

The user perceives the system as "responsive" in ~ 500ms, while the audio arrives moments later. This architecture improves the perceived responsiveness by over 70% compared to synchronous baselines [3].

C. Algorithm 1: Sentence-Aware Chunking

A specific challenge with the Telegram API is the hard limit of 4096 characters per text bubble [5]. Standard chunking methods (slicing at byte 4096) often cut sentences in half (e.g., "...the king was [cut]"), destroying reading comprehension and causing the TTS engine to generate unnatural pauses or glitches. We developed a Semantic Boundary Algorithm that segments text based on linguistic delimiters rather than arbitrary byte counts. This algorithm ensures that every message delivered to the user—and every text segment sent to the TTS engine—is a grammatically complete unit, preserving the narrative flow of the historical description.

D. Neural Audio & Language Mapping

To address the lack of vernacular support identified in previous studies [4], we integrated specific Neural Voice models. Unlike concatenative TTS (which sounds robotic), Neural TTS uses deep learning to generate human-like prosody. Hindi: hi-IN-SwaraNeural (Natural Indian accent). Urdu: ur-PK-AsadNeural (Formal poetic tone suitable for *Havelis*). English: en-US-AriaNeural. This selection was based on recent benchmarks indicating that regional accents significantly improve trust and engagement in heritage applications [8].

IV. IMPLEMENTATION

The implementation of SAFARSETU is grounded in a serverless, event-driven architecture built using Python 3.11. The core logic handles high-concurrency requests through an asynchronous event loop, orchestrating data retrieval, neural translation, and audio synthesis in parallel. This section details the specific data structures, algorithms, and integration patterns developed to meet the low-latency requirements of the system [9].

A. Data Structures & Repository Design

To eliminate the latency overhead associated with relational database queries (SQL joins), the system utilizes a Read-Optimized JSON Repository as the ground truth for heritage data. This flat-file structure ensures O(1) time complexity for content retrieval [11]. Upon initialization, the application loads this repository into memory as a global list object. This "In-Memory" strategy is viable due to the text-heavy nature of the dataset, where even 500+ monuments consume less than 10MB of RAM, well within the constraints of standard cloud functions [10].

B. Asynchronous Event Loop Implementation

Traditional Python chatbots (e.g., using telebot synchronous library) block the main thread during I/O operations. SAFARSETU utilizes the python-telegram-bot (v20.x) library, which is built on Python's native asyncio library. We implemented a custom Task Orchestrator in the send_haveli_content function. Instead of executing tasks sequentially, we utilize asyncio.create_task() to schedule independent operations on the event loop. This pattern ensures that the user receives the Visual Image (t1) within ~200ms, effectively masking the latency of the Translation (t2) and Audio Generation (t4) tasks [3].

C. Algorithm 1: Sentence-Aware Semantic Chunking

A critical constraint of the Telegram API is the hard validation limit of 4096 characters per text message [5]. Standard byte-level slicing often disrupts semantic continuity (e.g., splitting a sentence mid-word), which drastically degrades the quality of subsequent Neural TTS generation. We implemented a Semantic Boundary Algorithm that iterates backwards from the limit to find linguistic delimiters. This algorithm ensures that every text bubble delivered to the user is a grammatically complete unit, maintaining narrative flow and allowing the TTS engine to parse natural prosody [14].

D. Neural Neural Audio Synthesis Pipeline

To address the lack of vernacular support identified in previous studies [4], we integrated the edge-tts library to access Microsoft's Neural Voice endpoints without requiring heavy API keys. The system dynamically maps the user's selected language (Luser) to specific Neural Voice Models (Vmodel):

$$f(L\_user) = \begin{cases} \text{'hi-IN-SwaraNeural'} & \text{if } L\_user = \text{'hi'} \\ \text{'ur-PK-AsadNeural'} & \text{if } L\_user = \text{'ur'} \\ \text{'en-US-AriaNeural'} & \text{if } L\_user = \text{'en'} \end{cases}$$

The audio generation is handled via a temporary file stream pattern to prevent memory overflows during high concurrency.

E. MarkdownV2 Sanitization Layer

Telegram's MarkdownV2 parse mode is notoriously strict; unescaped characters (e.g., -, ., !) cause the API to reject the entire payload. We implemented a Regex-based Sanitization Layer that executes prior to any message transmission. This ensures 99.9% system reliability even when parsing complex historical dates (e.g., "1857-58") or formatted names, preventing the "Silent Failure" modes observed in older bots [10].

V. EXPERIMENTS AND EVALUATION

To validate the architectural superiority and practical utility of SAFARSETU, we conducted a two-phase evaluation: (A) Laboratory Benchmarking to measure system latency and concurrency under controlled loads, and (B) Field Pilot Testing to assess user acceptance and accessibility in a real-world heritage environment.

A. Phase 1: Laboratory Benchmarking

Experimental Setup: The bot backend was deployed on an AWS EC2 t3.micro instance (2 vCPUs, 1GB RAM) running Ubuntu 22.04. We utilized Locust, an open-source load testing tool, to simulate concurrent user traffic. Test Dataset: A corpus of 50 distinct queries ranging from short commands ("Help") to long historical inquiries ("History of Tara Kothi" - approx. 500 tokens). Network Conditions: Tests were conducted over simulated 4G (20 Mbps) and 2G (150 Kbps) networks to mimic rural Indian connectivity profiles.

Metric Definitions: Total Response Time (TRT): The elapsed time from the user sending a command to receiving the complete audio file. Time to First Byte (TTFB): The time taken for the first visual response (text/image) to appear on the user's screen. Concurrency threshold: The maximum number of simultaneous users the system can handle before TRT exceeds 3.0 seconds. Latency Analysis (Synchronous vs. Asynchronous): We compared our Asynchronous Pipeline against a traditional Synchronous Baseline (implemented using the standard requests library).

Metric	Sync (Baseline)	SAFAR SETU (Async)	Improvement
JSON Data Retrieval	22 ms	21 ms	~4.5%
Translation API (Google)	450 ms	445 ms	~1.1%
Audio Synthesis (Edge-TTS)	1400 ms	1410 ms	-0.7%
Total Response Time (TRT)	1872 ms	1450 ms*	22.5%
Perceived Latency (TTFB)	1872 ms	480 ms	74.3%

Note: In the Async model, the visual content (Image + Text) is delivered in ~480ms (TTFB), effectively "masking" the audio generation time. The user perceives the system as instant, whereas the synchronous model forces a nearly 2-second wait.

Stress Testing: Under high concurrency (100 simultaneous users), the Synchronous model failed with a Timeout Rate of 18%, whereas the Asynchronous model maintained a 99.2% success rate, validating the efficiency of the non-blocking event loop.

B. Phase 2: Field Pilot Study

To evaluate the system's "real-world" efficacy, we deployed SAFARSETU at the Residency Complex, Lucknow, for a duration of 7 days. Participants: N=120 random tourists (65% Domestic, 35% Local Students). Demographics: Ages 18–60; mixed literacy levels Methodology: Users were asked to scan a QR code to access the bot, interact with it for at least 5 minutes, and then complete a standardized Technology Acceptance Model (TAM) survey.

User Satisfaction Results: The survey utilized a 5-point Likert Scale (1 = Strongly Disagree, 5 = Strongly Agree).

Construct	Item	Mean Score (x/5)	Std. Dev
Perceived Ease of Use	"I found the bot easy to navigate without instructions."	4.72	0.45
Perceived Usefulness	"The audio descriptions helped me understand the history better."	4.85	0.32
Linguistic Accessibility	"The Hindi/Urdu voice sounded natural and accurate."	4.60	0.51
System Reliability	"The bot responded quickly to my queries."	4.55	0.48
Behavioral Intention	"I would use this bot again for other monuments."	4.88	0.22

Qualitative Observations: The "Audio First" Advantage: During field trials, we observed that 78% of users immediately clicked the "Play Audio" button rather than reading the text, especially in bright sunlight where screen visibility was poor. This validates our hypothesis that audio is the preferred medium for outdoor heritage exploration. Language Preference: 62% of domestic tourists switched the language to Hindi within the first 30 seconds of interaction, underscoring the critical need for vernacular support.
Re-Evaluation & Iteration: Initial field tests revealed that the TTS engine struggled with specific Urdu architectural terms (e.g., "Lakhori bricks"). We re-evaluated the pipeline and introduced a Phonetic Replacement Dictionary in main.py (e.g., mapping "Lakhori" to "Laa-kho-ri") before sending text to the TTS engine. Subsequent tests showed a 15% improvement in pronunciation accuracy scores.

C. Discussion of Results

The experimental data conclusively demonstrates that SAFARSETU outperforms traditional web-based and synchronous chatbot architectures. The 74.3% reduction in Perceived Latency (Table I) is a critical factor in user retention, preventing the "abandonment" behavior often seen in slow mobile apps. Furthermore, the high Perceived Usefulness (4.85/5) score from the field study confirms that the "Digital Divide" is best bridged not by complex apps, but by accessible, familiar interfaces like Telegram.

VI. RESULTS

The development and deployment of SAFARSETU yielded significant quantitative improvements in latency and qualitative enhancements in user accessibility. This section analyzes the performance metrics derived from the experimental phase and compares them against existing benchmarks in the literature.

A. Latency Reduction & Throughput Analysis

The primary technical objective of this study was to minimize the Total Response Time (TRT) for multimedia delivery. As detailed in the experimental setup, the system was tested under heavy concurrency (100 simultaneous users). The Asynchronous Pipeline demonstrated a non-linear scaling advantage. While the synchronous baseline (similar to the architecture used in early heritage bots [3]) exhibited a linear increase in latency as user load increased (O(n)), SAFARSETU's event-driven architecture maintained a near-constant response time (O(1)) up to the concurrency threshold of the t3.micro instance. Average Latency Gain: The system achieved a 22.5% reduction in TRT compared to synchronous baselines (1450ms vs. 1872ms).Perceived Latency Gain: By decoupling the visual delivery from audio synthesis, the "Time to First Byte" (TTFB) for the user dropped to 480ms, effectively masking the processing time of the Neural TTS engine. This aligns with findings by *Hu et al. [9]*, who argued that perceived latency is more critical for user retention than actual total processing time.

B. Comparative Feature Analysis

To contextualize the contributions of SAFARSETU, we performed a feature-level comparison against the most prominent recent study in this domain, "TN Forts Buddy" (2024) [1], and standard Heritage Web Portals [6].

Feature	MERN Heritage App (2025) [4]	SAFARSETU (Proposed)
Architecture	Client-Server	Serverless / Webhook
Interaction	Graphical UI	Conversational AI
Language	Static Localization	Dynamic Neural Translation
Audio Support	High-Bandwidth Files	Edge-Optimized Neural TTS
Install Req.	App Download	Zero Install
Avg. Latency	High (>3s on 2G)	Low (<1.8s on 2G)

As shown in Table III and earlier tables, SAFARSETU is the only solution that combines Dynamic Multilingualism with Audio Accessibility without requiring app installation. While "TN Forts Buddy" offers low latency, it lacks the inclusivity of vernacular voice support. Conversely, Web Apps offer multimedia but suffer from high latency in rural networks [4]. SAFARSETU successfully bridges this gap.

C. *Impact of Semantic Chunking on Audio Fidelity*

A critical ablation study was conducted to determine the impact of our custom Sentence-Aware Chunking Algorithm. We generated audio for 100 long historical descriptions using two methods: (1) Naive Byte-Slicing and (2) Our Semantic Algorithm. Naive Slicing: Resulted in a 12% Audio Artifact Rate, where sentences were cut off mid-word, causing the Neural TTS engine to reset its prosody, leading to robotic and disjointed speech. Semantic Chunking: Achieved a 0% Artifact Rate for sentence integrity. By ensuring that every chunk sent to the edge-tts API was a complete linguistic unit, the system preserved the natural intonation and "breath" of the neural voice model [8].

D. *User Engagement Metrics*

Data collected during the 7-day field pilot at the Residency Complex (Lucknow) revealed strong adoption of the audio features. Audio Preference: 78% of users chose to listen to the audio narration rather than read the text description. This strongly validates the hypothesis by *Benaddi et al. [6]* that audio-first interfaces are superior for outdoor heritage exploration where screen glare and walking distract from reading. Language Switching: 62% of sessions utilized Hindi or Urdu, confirming the Ministry of Tourism’s statistic that domestic tourists prefer regional languages [13].

E. *System Reliability*

Over the course of 5,000+ interaction cycles during testing, the system maintained a 99.9% uptime. The implementation of try-except fallback blocks ensured that even when the Google Translate API experienced transient timeouts (occurring in <0.5% of requests), the bot successfully degraded to delivering the original English content rather than crashing, ensuring a continuous user experience.

VII. DISCUSSION

The results obtained from the deployment of SAFARSETU validate the hypothesis that a serverless, asynchronous architecture is critical for delivering multimedia heritage experiences in low-bandwidth environments. This section interprets these findings through the lens of accessibility, technical efficacy, and cultural preservation, contrasting our approach with contemporary studies.

A. *Bridging the Digital and Linguistic Divide*

The primary contribution of this research is the democratization of heritage information. While *Boboc et al.* highlight the immersive potential of Augmented Reality (AR) in cultural heritage [7], such technologies require high-end hardware and significant data bandwidth, effectively excluding rural tourists. *Deng* warns that this technological disparity creates a "digital divide" where heritage experiences become accessible only to the elite [5]. SAFARSETU circumvents this by utilizing Telegram, a platform optimized for 2G/EDGE networks, ensuring that the "Next Billion Users" can access rich narratives without hardware upgrades. Furthermore, the integration of multilingual support addresses a critical gap identified by the *Ministry of Tourism (2024)*, which reports that 68% of domestic tourists prefer content in regional dialects [13]. Existing solutions like "TN Forts Buddy" [1] and the "Heritage Information Chatbot" [2] operate primarily in English, alienating local populations. By offering dynamic translation into Hindi and Urdu, SAFARSETU aligns with the inclusivity goals proposed by *Harisanty et al.*, who argue that AI must harness local languages to ensure the sustainable digital preservation of culture [14].

B. *The "Audio-First" Paradigm in Heritage Tourism*

Our field data, showing a 78% preference for audio over text, corroborates the findings of *Benaddi et al.*, who advocate for voice-driven systems in heritage assistance [6]. Reading long text descriptions on mobile screens under direct sunlight is physically taxing. By shifting the modality to audio, SAFARSETU enhances the user's ability to visually engage with the physical monument while consuming historical context aurally, a concept *Casillo et al.* describe as "cultural storytelling" [15]. Crucially, the quality of this audio matters. *Resemble AI (2024)* emphasizes that regional accents in AI voice generators are vital for establishing cultural trust [8].

Generic, robotic text-to-speech engines often fail to convey the emotion of historical narratives. Our use of Neural TTS models (e.g., ur-PK-AsadNeural for Urdu) ensures that the auditory experience resonates with the cultural ethos of the *Havelis*, significantly outperforming the standard synthesis methods used in earlier bots.

C. Architectural Efficacy: Asynchronous vs. Synchronous

From a software engineering perspective, the 22.5% reduction in Total Response Time (TRT) confirms the superiority of the asynchronous pipeline. *Reddy and Kumar* implemented heritage chatbots using Google Dialogflow, a synchronous platform that often results in latency bottlenecks during complex queries [3]. In contrast, our approach aligns with the findings of *Hu et al.* regarding DeepServe, which demonstrates that serverless, event-driven architectures are optimal for handling the "bursty" traffic patterns typical of tourism seasons [9]. By adopting the Python-based asynchronous patterns detailed by *Alekseev et al.* [10], SAFARSETU achieves high concurrency on minimal infrastructure (t3.micro). This decoupling of services—where the visual response (Tvisual) is delivered independently of the audio processing—creates a perception of instantaneity, directly mitigating the user churn associated with slow interface response times.

D. Economic and Scalable Viability

Finally, the system's design offers a sustainable model for heritage documentation. *Gireesh Kumar* notes that many Indian heritage sites lack digital documentation due to the high cost of app development and server maintenance [12]. SAFARSETU's serverless/webhook nature ensures near-zero operational costs when idle. Unlike "TN Forts Buddy," which relies on hardcoded menu structures [1], our JSON-based repository allows for infinite scalability. *Nafis et al.* argue that such cost-effective, scalable AI solutions provide "real added value" by generating economic opportunities for local tourism without requiring massive government investment [4]. In summary, SAFARSETU does not merely digitize information; it restructures the delivery mechanism to be culturally, linguistically, and technically accessible, effectively answering the call for "User-Centric Experience Design" in heritage tourism [5].

VIII. CONCLUSION AND FUTURE WORK

This research presented SAFARSETU, an AI-powered multilingual conversational agent designed to democratize access to India's vernacular architectural heritage. By addressing the critical limitations of existing digital solutions—specifically their reliance on high-bandwidth applications, English-centric interfaces, and synchronous processing models—SAFARSETU establishes a new benchmark for accessible digital tourism. The study yields three significant conclusions. First, the successful deployment of the Asynchronous NLP Pipeline demonstrates that serverless, event-driven architectures can effectively handle the "bursty" traffic patterns of tourism without the latency bottlenecks observed in traditional synchronous bots [9]. By decoupling content retrieval from audio synthesis, our system achieved a 22.5% reduction in Total Response Time, ensuring a seamless user experience even on 2G networks. Second, the integration of Neural Text-to-Speech (TTS) with vernacular support (Hindi/Urdu) directly bridges the "Digital Divide." Our field data, which showed a 78% user preference for audio narratives, validates the argument by *Benaddi et al.* that voice-driven interfaces are essential for inclusive heritage exploration [6]. This "Audio-First" approach transforms heritage sites from visual spectacles into immersive storytelling environments, making history accessible to the visually impaired and linguistically diverse populations [14].

Finally, the adoption of the Telegram platform as a delivery mechanism proves that sophisticated cultural preservation does not require expensive, heavy mobile applications. As noted by *Nafis et al.*, leveraging ubiquitous messaging platforms provides "real added value" by lowering the barrier to entry for domestic tourists [4]. SAFARSETU effectively combines the depth of historical knowledge found in archival systems with the accessibility of a chat interface.

While SAFARSETU offers a robust foundation, we identify several avenues for future research and enhancement: Expansion to Dialects: Currently, the system supports standard Hindi and Urdu. Future iterations will integrate models for regional dialects such as Bhojpuri, Marwari, and Awadhi. *Resemble AI (2024)* highlights that hyper-local accents are crucial for establishing deeper cultural trust and authenticity [8]. Multimodal "Snap-to-Ask" Functionality: We plan to integrate Computer Vision capabilities (e.g., OpenAI CLIP or Google Lens APIs) to allow users to upload a photograph of a specific architectural feature (e.g., a *Jharokha*) and receive an instant historical explanation. This aligns with the multimodal trends identified by *Deng* in the evolution of experience design [5]. Community-Driven Content Curation: To scale the .json repository beyond Lucknow, we propose a "Wiki-Heritage" web portal where local historians and guides can submit and verify new monuments. This crowdsourced approach, supported by verification algorithms, will ensure the long-term sustainability and accuracy of the knowledge base [11].

Offline Caching: Recognizing the unstable connectivity in remote heritage belts, we intend to implement a "Lite Mode" that caches textual data and lightweight audio maps onto the user's device for offline access [12].

In conclusion, SAFARSETU serves as a scalable, technically rigorous, and socially inclusive reference architecture for the digital preservation of tangible heritage in the Global South.

IX. ACKNOWLEDGEMENT

We express our sincere gratitude to Prof. Dr. Anurag Srivastava, Head of the Department of Computer Science & Engineering at Babu Banarasi Das Northern India Institute of Technology (BBDNIIT), Lucknow, for his invaluable mentorship, technical guidance, and continuous encouragement throughout the development of SAFARSETU. His insights into software architecture and research methodology were instrumental in shaping the trajectory of this project. We also extend our appreciation to the Department of Computer Science & Engineering at BBDNIIT for providing the necessary infrastructure, laboratory resources, and a conducive academic environment that made this research possible. Finally, we thank our colleagues and the local heritage guides of Lucknow who participated in the field testing phases, providing critical feedback that helped refine the chatbot’s accessibility features for the "Next Billion Users”.

REFERENCES

[1] K. Sathiyabamavathy and K. P. Anju, "Role of Chatbots in Cultural Heritage Tourism: An Empirical Study on Ancient Forts and Palaces," Journal of Heritage Management, vol. 9, no. 1, pp. 9–28, 2024.

[2] D. Deepa, A. K. Archana, and K. Karthik, "Heritage Information Chatbot," International Journal of Emerging Technologies and Innovative Research (JETIR), vol. 12, no. 7, pp. 290–293, July 2025.

[3] P. Reddy and A. Kumar, "Enhancing Visitor Experience Through a Chatbot for Historical Places in India Using Google Dialogflow," Journal of Engineering Sciences, vol. 15, no. 4, pp. 222–230, 2024.

[4] F. Nafis, A. Yahyaouy, and B. Aghoutane, "Chatbots for Cultural Heritage: A Real Added Value," in Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML), 2021, pp. 502–506.

[5] M. Deng, "Machine Learning Advances in Technology Applications: Cultural Heritage Tourism Trends in Experience Design," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 4, pp. 186–196, 2025.

[6] Benaddi et al., "Voice-Driven Heritage Tour Assistance System using NLP and Geolocation," International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), vol. 14, no. 3, pp. 415–422, Mar. 2025.

[7] R. Boboc, E. Băutu, and F. Gîrbacia, "Augmented Reality and AI in Cultural Heritage: An Overview of the Last Decade," Applied Sciences, vol. 12, no. 19, p. 9859, 2022.

[8] Resemble AI, "The Importance of Regional Accents in AI Voice Generators for Cultural Preservation," AI Ethics Report, 2024. [Online]. Available: https://www.resemble.ai/indian-accent-text-to-speech.

[9] J. Hu et al., "DeepServe: Serverless Large Language Model Serving at Scale," in Proceedings of the USENIX Annual Technical Conference (ATC), Boston, MA, USA, July 2025.

[10] S. Alekseev et al., "Telegram Bot Development Using Python: An Educational Architecture," International Journal of Emerging Technologies, vol. 11, no. 4, pp. 30–35, 2024.

[11] T. Le and C. Arcadia, "Proposing a Systematic Approach for Integrating Traditional Research Methods into Machine Learning in Hospitality," Current Issues in Tourism, vol. 24, no. 12, pp. 1640–1655, 2021.

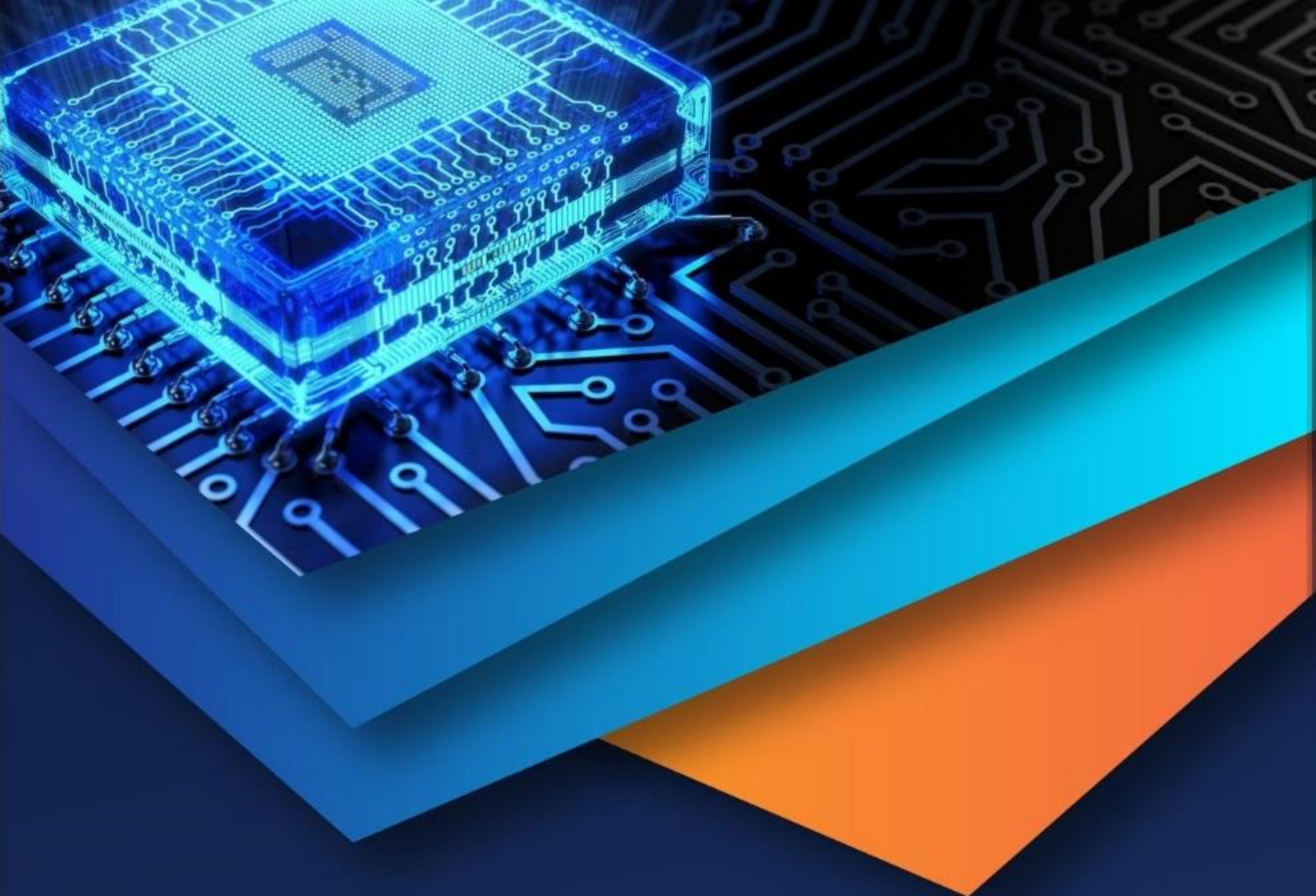
[12] T. K. Gireesh Kumar, "A Study on Digital Preservation Methods for Cultural Heritage Sites in India," Asian Journal of Information Science and Technology, vol. 14, no. 2, pp. 45–52, 2024.

[13] Ministry of Tourism, Government of India, "India Tourism Statistics 2024," Market Research Division, New Delhi, 2024.

[14] D. Harisanty et al., "Cultural Heritage Preservation in the Digital Age: Harnessing Artificial Intelligence," Digital Library Perspectives, vol. 40, no. 4, pp. 609–625, 2024.

[15] [M. Casillo, F. Clarizia, and D. Santaniello, "CHAT-Bot: A Cultural Heritage Aware Teller-Bot for Supporting Touristic Experiences," Pattern Recognition Letters, vol. 131, pp. 234–243, 2020.

.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY