IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15159

AVI-AN AI VIRTUAL ASSISTANT

Akhila B[1], Suma N R[2]

Department of MCA, BIT,

K.R. Road, V.V. Pura, Bangalore, India[1,2]

Abstract: Managing incoming calls during busy or unavailable periods remains a significant challenge in modern organizational environments, often leading to missed important calls and inefficient follow-up scheduling. This paper presents AVI – an AI Virtual Assistant that automates call handling and meeting scheduling by integrating Twilio telephony services, a Fast API backend, Python logic, Google Calendar API, and a React-based dashboard. Incoming calls are processed through Twilio webhooks, where digit-based user inputs are forwarded to the backend for deterministic decision-making and real-time calendar availability checks. Scheduling requests are displayed on the dashboard with caller details and approve or reject options, enabling user-controlled meeting management. Upon approval, a meeting link is automatically generated and updated on the dashboard, while rejected requests are declined without further notification. Experimental results indicate that AVI offers reliable request handling, smooth backend-to-dashboard synchronization, and efficient automated scheduling, making it a scalable and practical solution for intelligent call automation.

Keywords: AI Virtual Assistant, Call Automation, Twilio Webhooks, Fast API Backend, Automated Meeting Scheduling, Google Calendar Integration, React Dashboard, Telephony Automation.

I. INTRODUCTION

In modern organizations, managing incoming calls efficiently is critical, especially when professionals are frequently engaged in meetings or other responsibilities. Missed calls often lead to delayed communication and reduced service quality, while conventional call-handling approaches such as manual reception or traditional IVR systems either require constant human involvement or frustrate callers with complex navigation. Although AI-based voice assistants have been introduced, their reliance on speech recognition limits reliability over telephony networks due to noise, compression, and inconsistent audio quality. This paper presents **AVI – an AI Virtual Assistant**, an intelligent call-handling and meeting-scheduling system developed using Twilio, Fast API, Python, Google Calendar API, and a React-based dashboard to overcome these limitations. AVI replaces voice recognition with a DTMF-based digit input mechanism, enabling dependable caller interaction through keypad inputs. Incoming calls are routed via Twilio webhooks, and digit inputs are processed by a Fast API backend to determine actions such as scheduling meetings or recording voice messages. Upon approval, the system updates the calendar by creating a meeting event and generating a Google Meet link, clearly marked as scheduled by the AI assistant. The dashboard provides real-time visibility of caller requests and approval status, while rejected requests are handled without scheduling. Due to geographic restrictions on Twilio inbound routing in India, system validation was performed using simulated webhook triggers, demonstrating reliable request handling and seamless system integration.

1.1 Project Description

AVI – An AI Virtual Assistant is an automated call-handling and meeting-scheduling system designed to manage incoming calls during busy or unavailable periods. The system integrates Twilio telephony services with a FastAPI backend, Python-based logic, Google Calendar API, and a React-based dashboard, while using a DTMF-based digit input mechanism instead of speech recognition to ensure reliable interaction over telephone networks. Incoming calls are processed through Twilio webhooks, and caller inputs are handled by the backend to perform actions such as scheduling meetings or recording voice messages. Scheduling requests are displayed on the dashboard for user approval, and upon approval, the system checks calendar availability, creates a meeting event, generates a Google Meet link, and updates the dashboard in real time. The project was validated using simulated webhook triggers and demonstrates efficient call automation, seamless system integration, and dependable scheduling.

1.2 Motivation

In today's fast-paced professional environment, individuals often miss important calls due to busy schedules, meetings, or unavailability, leading to delayed communication and reduced efficiency. Traditional call-handling methods such as manual reception and conventional IVR systems are either resource-intensive or provide a poor user experience. Although voice-based AI assistants exist, their dependence on speech recognition makes them unreliable over telephone networks. The motivation behind this project is to develop a reliable, user-friendly, and automated call-handling system that

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026
DOI: 10.17148/IJARCCE.2026.15159

overcomes these limitations by using digit-based interaction and intelligent scheduling, thereby improving communication efficiency and reducing manual effort.

II. RELATED WORK

Paper [1], Traditional IVR systems have been widely used in organizations to handle incoming calls through predefined menu-based options. While they reduce the need for human operators, these systems often provide limited flexibility and result in poor user experience due to complex navigation and lack of intelligent decision-making.
Paper [2], Cloud-based telephony solutions such as Twilio enable programmable voice services using APIs and webhooks. These platforms support automated call routing and data exchange with backend applications; however, they rely on external logic for decision-making and do not inherently provide intelligent scheduling or approval-based workflows.
Paper [3], Several AI-powered virtual assistants use speech recognition and natural language processing to interpret caller intent. Although effective in controlled environments, their performance over telephone networks is degraded due to background noise, signal compression, and language variations, making them less reliable for critical call handling.
Paper [4], Existing scheduling tools integrate with calendar platforms to automate meeting creation and availability checks. However, these systems typically operate independently of call-handling mechanisms and lack real-time caller interaction and user-controlled approval processes.
Paper [5], Event-driven architectures using webhooks have been explored for real-time data exchange between telephony services and backend systems. While efficient, most implementations focus on data transfer rather than providing a complete call-handling and scheduling solution.

III. METHODOLOGY

A. System Environment

The proposed AVI system operates in a cloud-based and web-enabled environment designed for reliable call handling and automated scheduling. The backend environment is built using Python with the Fast API framework and runs on a Uvicorn ASGI server for handling concurrent webhook requests efficiently. Twilio's cloud telephony platform is used to receive and manage incoming calls, while Ngrok is employed to expose the local backend server securely for webhook communication during development and testing.

The scheduling environment integrates Google OAuth and Google Calendar API to access calendar availability and generate Google Meet links. The frontend environment consists of a React-based dashboard that provides real-time visibility of caller requests, approval status, and scheduled meetings. The system is platform-independent and can be accessed through any modern web browser, requiring only an active internet connection for operation.

Twilio's cloud telephony environment serves as the core communication layer, handling incoming calls, digit-based user inputs (DTMF), and voice prompts. The system operates independently of speech recognition, ensuring reliable performance over telephone networks. Google OAuth is employed to securely authenticate user access to calendar services, while the Google Calendar API enables real-time availability checks, event creation, and automatic generation of Google Meet links.

The frontend environment consists of a React-based web dashboard that runs on standard web browsers. The dashboard provides real-time visibility of incoming caller requests, approval and rejection controls, meeting status updates, and generated meeting links. The system follows an event-driven architecture, allowing smooth synchronization between the telephony layer, backend logic, and frontend interface. Overall, the system environment is scalable, platform-independent, and suitable for deployment in organizational settings with minimal hardware requirements and reliable internet connectivity.

The AVI system supports both development and production deployment environments. During development, the backend runs locally and is exposed using Ngrok to enable communication with Twilio webhooks. This setup allows effective testing of call flows and backend logic. For production, the system can be deployed on cloud servers to ensure scalability, availability, and reliable handling of concurrent call requests.

The system is tested through unit and integration testing to verify correct processing of digit inputs, backend logic, calendar scheduling, and dashboard updates. Due to geographic restrictions on Twilio inbound routing in India, simulated webhook triggers are used to emulate real call events. Testing confirms reliable system behavior, accurate scheduling, and smooth end-to-end integration.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15159

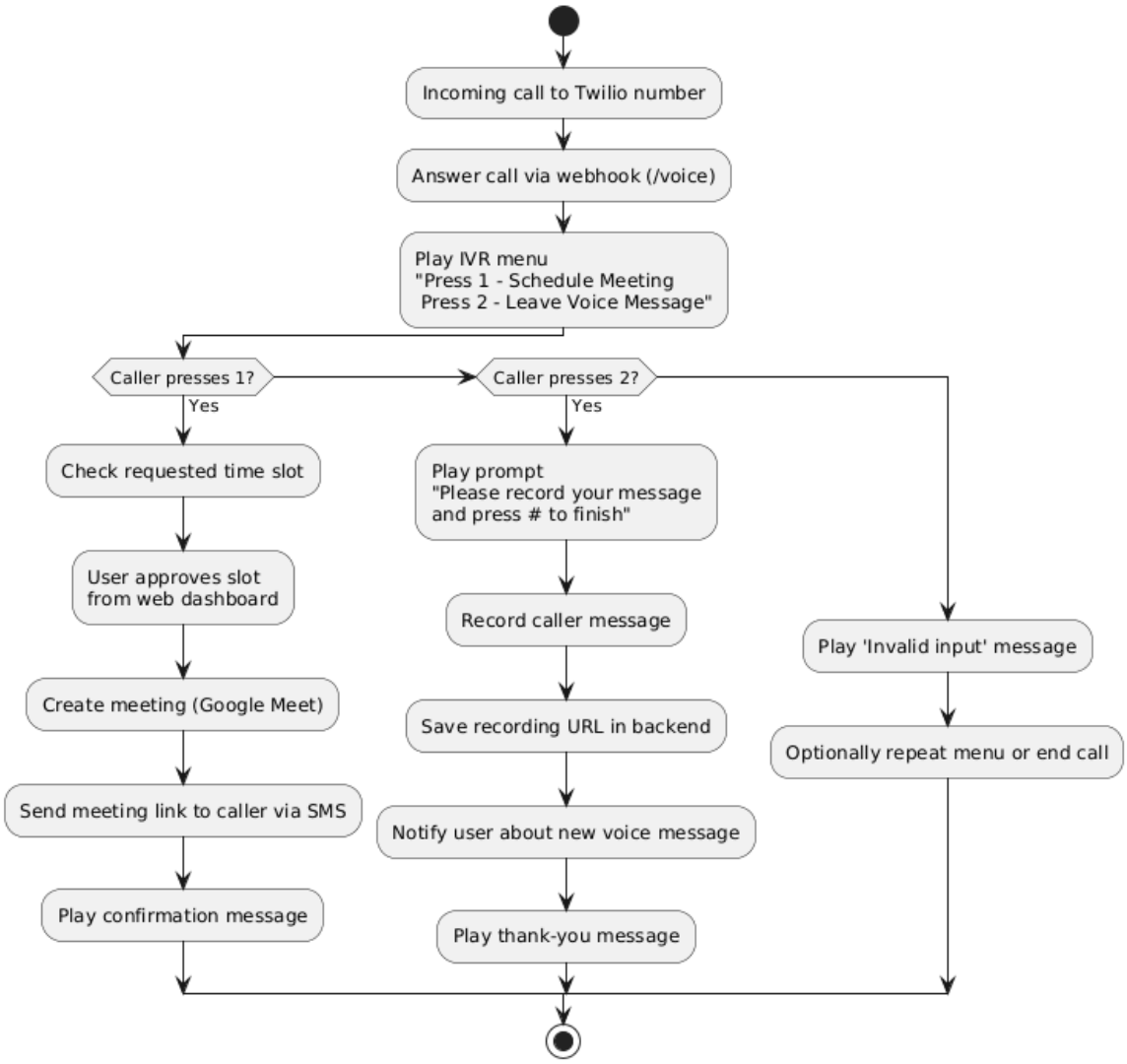

Fig.1.Flowchart of methodology

B. Call Handling Architecture

Client-Side Training:
Each client instance of the AVI system processes local call interaction data such as call frequency, user responses, approval patterns, and scheduling behaviour. The learning process is performed locally without sharing raw call data, ensuring privacy of user communication and personal information.

Server-Side Aggregation:
Instead of collecting raw call logs, the central server receives only model updates or learned parameters from multiple AVI client instances. These updates are securely aggregated to improve a global decision model, which can enhance call-handling efficiency and scheduling accuracy across the system.

C. Call Handling and Decision Logic

The global model can be periodically updated through iterative learning rounds to adapt to changing call patterns and user preferences. This adaptive mechanism enables AVI to improve decision-making for call routing, meeting scheduling, and response handling over time. By continuously refining the shared model without accessing raw data, the system enhances performance while maintaining data privacy and scalability.

DOI: 10.17148/IJARCCE.2026.15159

D. Implementation Flow

1. Initialize the Fast API backend server and configure Twilio webhook endpoints.
2. Receive incoming calls through the Twilio telephony platform.
3. Capture caller digit inputs using the DTMF-based interaction mechanism.
4. Forward digit inputs to the Fast API backend for deterministic decision-making.
5. Display caller details and request status on the React-based dashboard.
6. Upon approval, check calendar availability and schedule the meeting using Google Calendar API.
7. Generate a Google Meet link and update the dashboard in real time; repeat the process for subsequent calls.

E. Hardware and Software Requirements

- **Hardware:** Standard desktop or laptop system with a minimum of 8 GB RAM, multi-core processor, and stable internet connectivity.
- **Software:** Python 3.7 Fast API framework, Uvicorn server, Twilio cloud telephony platform, Ngrok, Google Calendar API with OAuth, React.js.

IV. SYSTEM EVALUATION FRAMEWORK

This section describes the overall system design, simulation process, and evaluation strategy adopted for the AVI – AI Virtual Assistant system. The framework integrates cloud telephony, backend automation, and calendar scheduling to enable intelligent call handling and meeting management in organizational environments. The system is implemented using Python as the primary control and orchestration layer, coordinating interactions between Twilio webhooks, the Fast API backend, Google Calendar API, and the React-based dashboard. The evaluation focuses on validating reliable call processing, accurate scheduling, and seamless end-to-end system integration.

A. System Architecture and Workflow

The architecture of **AVI – An AI Virtual Assistant** follows a modular design that integrates telephony services, backend processing, and calendar automation. Incoming calls are managed through the Twilio voice platform, where caller intent is captured using a DTMF-based digit input mechanism. Call events are transmitted to a Fast API backend via webhooks for structured decision-making. The backend communicates with the Google Calendar API to schedule meetings and generate Google Meet links. All system updates are reflected on a React-based dashboard, enabling real-time monitoring and administrative control.

B. Testing and Validation

The AVI system was tested and validated to ensure reliable call handling, accurate scheduling, and seamless integration between all components. Since inbound call routing through Twilio is geographically restricted in India, testing was performed using simulated webhook triggers to emulate real incoming call events and digit-based user interactions. These simulated inputs allowed verification of backend decision logic, DTMF input handling, and call flow execution.

Integration testing was carried out to validate communication between the Twilio platform, Fast API backend, Google Calendar API, and the React-based dashboard. The system was evaluated by monitoring dashboard updates, approval workflows, meeting scheduling accuracy, and successful generation of Google Meet links. The testing results confirm consistent system behaviour, real-time synchronization across modules, and dependable end-to-end operation of the AVI virtual assistant.

C. Call Automation and Control Flow

The AVI system follows a structured request handling and decision-making process to manage incoming calls efficiently. When a call is received through the Twilio platform, caller inputs are captured using a DTMF-based digit input mechanism. These inputs are forwarded to the Fast API backend via webhooks, where deterministic logic is applied to identify the requested action. Based on the input, the system either initiates a meeting scheduling request or records a voice message. Approved scheduling requests trigger calendar availability checks and automatic meeting creation using

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15159

the Google Calendar API, while rejected requests are safely discarded. This rule-based process ensures reliable execution, predictable outcomes, and secure handling of caller interactions.

Results and Observations

The AVI system successfully handles inbound calls using a DTMF-based digit input mechanism, eliminating dependence on speech recognition and reducing interpretation errors. Call events are reliably processed through Twilio webhooks, ensuring a structured and consistent flow of data to the backend.

The system automates meeting scheduling through seamless integration with Google OAuth and the Google Calendar API, accurately identifying available time slots and generating Google Meet links without manual intervention.

A React-based dashboard provides real-time visualization of caller requests, scheduling status, and meeting details, enabling effective administrative control. Overall communication efficiency is improved by minimizing missed calls and reducing administrative workload. System functionality was validated using simulated webhook triggers, confirming backend stability and reliable operation despite telephony routing limitations.

The Meeting Approval Dashboard serves as the central control interface for managing incoming meeting requests generated by the AVI system. It displays real-time details of callers, including contact information, request status, and preferred scheduling time. Authorized users can review each request and take immediate action by approving or rejecting the meeting. Upon approval, the dashboard triggers automatic calendar scheduling and displays the generated meeting link, while rejected requests are closed without further processing.

This dashboard enables efficient decision-making, ensures transparency in call management, and reduces manual coordination efforts.

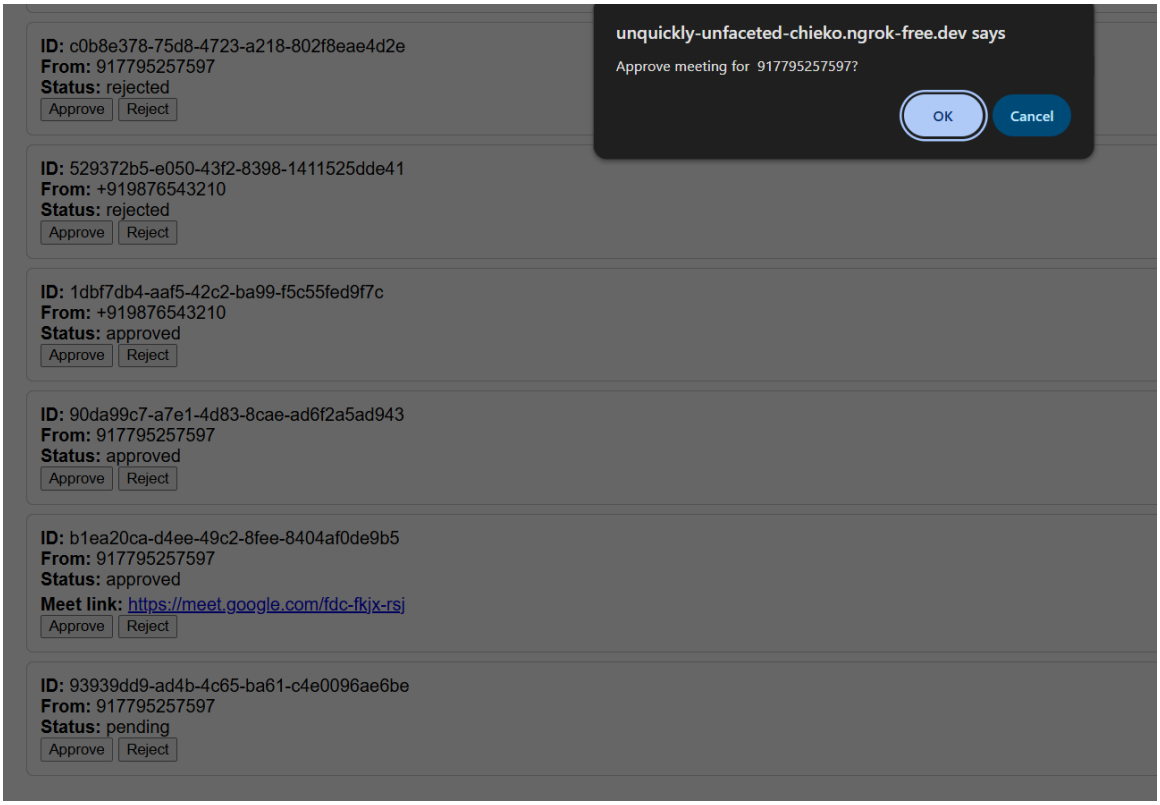

Fig. 1. Meeting approval dashboard

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026

DOI: 10.17148/IJARCCE.2026.15159

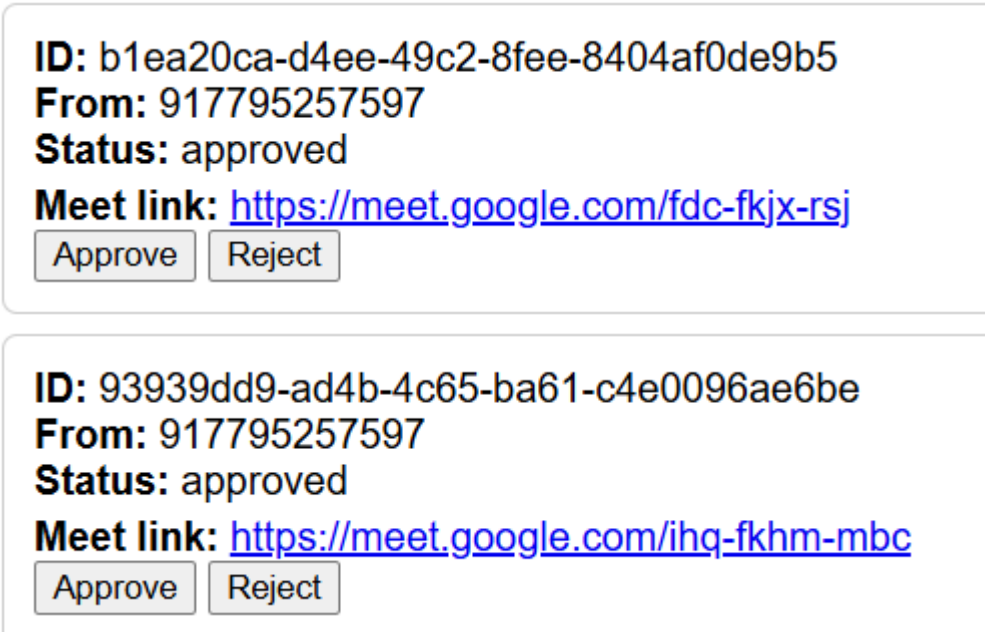

Fig. 2. Automatic meeting link generated after approved

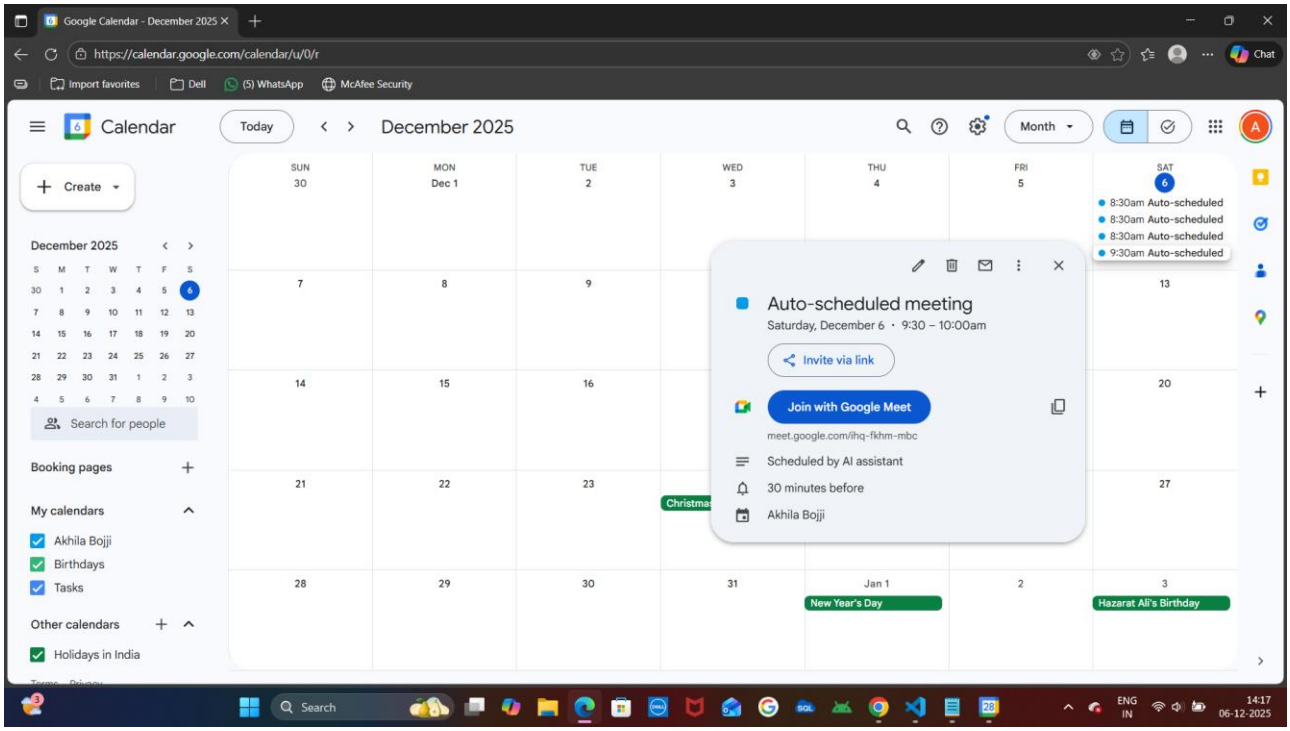

Fig. 3. AI Assistant scheduled a meeting by checking free slot

V. RESULTS AND DISCUSSION

The implementation of AVI – An AI Virtual Assistant demonstrates effective automation of call handling and meeting scheduling in busy organizational environments. The system successfully processed inbound calls using a DTMF-based digit input mechanism, ensuring reliable interaction without the limitations of speech recognition. Call events were consistently handled through Twilio webhooks, enabling structured communication between the telephony layer and the Fast API backend.

Automated scheduling through integration with Google Calendar API accurately identified available time slots and generated Google Meet links with minimal delay. The React-based dashboard provided real-time visibility of caller

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 1, January 2026
DOI: 10.17148/IJARCCE.2026.15159

requests, approval status, and scheduled meetings, improving administrative control and response efficiency. Testing using simulated webhook triggers validated system stability and reliable backend performance despite telephony routing constraints. Overall, the results indicate that the proposed system reduces missed calls, minimizes manual coordination, and enhances communication efficiency, confirming its suitability as a scalable solution for intelligent call automation.

VI. CONCLUSION

AVI – An AI Virtual Assistant was presented as a practical solution for automating inbound call handling and meeting scheduling in professional environments. The system employs a deterministic DTMF-based interaction model, eliminating the limitations of speech recognition and ensuring reliable user input over telephony networks. By integrating Twilio telephony services with a Fast API backend, Google Calendar API, and a React-based dashboard, AVI enables seamless coordination between call management and automated scheduling. Due to regional restrictions on direct inbound call routing for Indian numbers, the system was tested using simulated webhook interactions, which successfully validated backend workflows, scheduling logic, and dashboard synchronization. The results demonstrate that AVI provides a scalable, efficient, and dependable alternative to conventional voice-based virtual assistants. Its modular architecture further supports future enhancements, including extended telephony support, advanced analytics, and integration with additional communication platforms.

VI. FUTURE WORK

Future development of the AVI system will focus on enabling full real-time inbound call handling by extending telephony support to regions where direct voice routing is currently restricted. Once geographic limitations associated with Twilio services are resolved, the system can be deployed and tested using live call traffic to further validate performance under real-world conditions. Additional enhancements may include improved monitoring of call interactions, optimization of backend workflows, and extended dashboard analytics. These developments will strengthen system reliability and enable large-scale deployment while maintaining the existing architecture, scheduling logic, and dashboard integration.

REFERENCES

[1]. Twilio Inc., “Twilio Voice API Documentation,” *Twilio*, 2024. [Online]. Available: https://www.twilio.com/docs/voice
[2]. Google, “Google Calendar API Documentation,” *Google Developers*, 2024. [Online]. Available: https://developers.google.com/calendar
[3]. R. T. Fielding, *Architectural Styles and the Design of Network-Based Software Architectures*, Ph.D. dissertation, University of California, Irvine, 2000.
[4]. J. Klein and R. Dabbish, “Scheduling and coordination in collaborative systems,” in *Proceedings of the ACM Conference on Intelligent User Interfaces*, 2018.
[5]. S. Patel and R. Shah, “Challenges of speech recognition in telephony networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 28, 2020.
[6]. A. Gupta and D. Kumar, “DTMF-based interaction systems for telecommunication applications,” *International Journal of Embedded Systems*, vol. 12, no. 3, 2021.
[7]. React Team, “React Documentation,” *Meta Platforms*, 2024. [Online]. Available: https://react.dev
[8]. FastAPI Team, “FastAPI Documentation,” 2024. [Online]. Available