ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec156

Meet Mate: AI Powered Meeting Scheduler

Pragati Arya[1]; Shruti Singh[2]; Virendra Kumar Ojha[3]; Saurabh Singh[4]

[1;2;3;4]Department of Computer Science Engineering (Data Science) Greater Noida Institute of Technology

Publication Date: 2025/12/11

Abstract: This research paper is prepared for presenting MEET MATE, an AI powered meeting scheduler to manage the meeting more efficiently, this app integrates many features i.e.; face authentication, password protection, meeting scheduler and rescheduler, jitsi link generator, sends automatic invite through WhatsApp, hand gesture-based volume control using AI and python. It is different from other such apps as this app provides meeting scheduling and rescheduling as well as deleting through GUI. This app generates a dynamic meeting room which is Jitsi where participants get automatic invite through whatsapp. this also controls volume using hand gesture recognition making the system more effortless using Mediapipe. facial recognition makes the app more secure and authenticated and this feature is created using OpenCV face detection. all the meeting details are saved in JSON storage. Lastly, the most vivid part is that it used Tkinter for creating UI. It also gives voice feedbacks using pyttsx3 after every action successful execution

Keywords: *MeetMate AI, Face Authentication, Gesture Recognition, WhatsApp Automation, Voice Assistance, Tkinter, MediaPipe, OpenCV.*

How to Cite: Pragati Arya; Shruti Singh; Virendra Kumar Ojha; Saurabh Singh (2025) Meet Mate: AI Powered Meeting Scheduler. *International Journal of Innovative Science and Research Technology*, 10(12), 352-356. https://doi.org/10.38124/ijisrt/25dec156

I. INTRODUCTION

This is an ideal app for students, teachers, and professional workers as in today's busy world, time management plays a crucial role in improving productivity. This app provides facial recognition [1], [2], [3], [4] which makes the app unique from others, and it also integrates hand gesture-based volume control [5], [6], [7]. Traditional meeting platforms do not include all these features in one application and lack artificial intelligence, so this app is designed to combine everything in a single system and make human work smarter, more efficient, and accessible through AI-enabled Human-Computer Interaction (HCI) [7], [15], even for differently abled users, without compromising security and privacy of personal details [13].

After authenticating the face and entering the password [13], the user just needs to give a brief input like "Physics class at 3 PM" and provide the WhatsApp contact, and MeetMate AI is ready to schedule the meeting at 3 PM, create a Jitsi room, and share the link using WhatsApp web-based automation [10], [11], or simply copy the link to clipboard for manual sharing. Invitees can join by tapping the shared link. The volume can be controlled using hand gestures detected by MediaPipe [5], [6], [7] where lifting the hand increases the volume and lowering it reduces it. It also allows voice-based interaction using pyttsx3 [8].

This tool bridges the gap between traditional and AI-based touchless control for modern Human-Computer Interaction (HCI) [7], [15]. It does not require cloud-based storage, as all meeting records are stored locally using JSON-based persistent storage [16]. Tkinter provides the GUI interface platform [12], and the overall application is implemented using Python integrating OpenCV [1], MediaPipe [5], and automation.

As digital communication tools have become more central in the post-pandemic era, the demand for intelligent and consolidated meeting management platforms has grown significantly. MeetMate AI responds to these evolving needs by bringing several essential features together within one system, minimizing manual effort and removing the need to rely on multiple separate applications. The platform enhances overall productivity while supporting inclusive interaction through gesture-based controls and voice-guided responses.

With its blend of automation, AI-driven functionalities, and secure user verification, MeetMate AI provides an efficient and dependable solution suited for contemporary academic and professional settings that require streamlined and smart workflow management.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec156

Fig 1 MeetMate AI

II. LITERATURE REVIEW

Basic event planning, reminders, and time management features are available in various meeting scheduling and collaboration platforms like Google Calendar and Microsoft Outlook. These tools do not emerge AI integration, gesture-based interaction, face authentication, or messaging services for automatic communication [6]. Biometric-based face recognition systems have been explored to enhance security and reliability. OpenCV is commonly used in computer vision applications because of its well-implemented image processing features [1]. Face recognition has been taken to the next step which result in better accuracy and computational efficiency [2], [11], [12].

Gesture-based Human-Computer Interaction represents the direction of development for touchless control using hand motion and vision-based tracking. MediaPipe, developed by Google, provides real-time hand landmark detection using machine learning models suitable for volume control and other forms of interactive input processing [3, 5]. In various research studies, this approach was seen to provide better accessibility, thus offering a more intuitive interaction model for differently abled users and so increasing overall usability in contemporary systems [5, 14].

Voice-based automation and interaction have also played an important role in intelligent applications. Offline text-to-speech technologies such as pyttsx3 are in use for providing voice feedback seamlessly, while making AI systems more conversational and user-friendly [9]. Simultaneously, natural language processing systems are being integrated into meeting automation for the processing of user commands and for scheduling different tasks more intelligently [8]. WhatsApp Business API is emerging as a popular platform for automated communication, and is being used by applications for invitation sending, sending meeting details, and reminders with the help of web-based automation and link encoding techniques [6], [10].

Python, on the software development level, provides a flexible environment for GUI applications combined with AI modules. Tkinter, Python's default GUI framework, provides an easy way to create lightweight graphical interfaces in AI-enabled desktop applications [4]. Local JSON data storage allows for offline access, privacy, and security without any dependence on cloud services [13]. Facial recognition, gesture control, speech synthesis, and automation of meetings have been addressed individually in different research works. Very few systems integrate face authentication, gesture detection, voice interaction, AI-assisted scheduling, WhatsApp communication, and GUI-based persistent calendar management into one integrated AI-based application for meeting management. MeetMate AI thus hopes to fill this gap.

III. METHODOLOGY

The MeetMate AI methodology consists of several integrated modules: facial authentication, gesture-based volume control, WhatsApp meeting automation, voice interaction, scheduling, and GUI-based interface. These work together to provide a smart and efficient way of managing meetings. The overall architecture is implemented in Python, integrating OpenCV, MediaPipe, Tkinter, pyttsx3, and JSON for local data storage.

- *Face Authentication*

Face authentication provides basic access security for starting the meeting scheduler. Face detection is done in real-time using OpenCV through deep learning-based models [1]. A fast facial detection-based algorithm uses classifiers to find

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec156

features [11]. An eigenface approach in face recognition helps to further validate the recognized facial features against a database that has already been trained [12]. Biometric authentication boosts security and blocks unauthorized access [2], [15].

➢ *Gesture-Based Volume Control*

MediaPipe is used for the detection of hand gestures to track coordinates in real time for the key finger points. Comparing the landmarks of the thumb and pinky finger, if the movement is upward, the volume increases; with downward movement, the volume decreases. Human-Computer Interaction based on gestures improves accessibility and provides touchless control over the system for differently abled users and real-time operations.

➢ *WhatsApp Meeting Automation*

Once a meeting is scheduled, the application automatically generates a dynamic meeting room URL using timestamp-based encoding. It sends invitations to participants on WhatsApp through API-based message automation [6], [10]. The system encodes the meeting name, scheduled time, and link with URL parsing techniques and opens WhatsApp Web for direct sharing.

➢ *Voice Interaction (Text-to-Speech)*

The pyttsx3 text-to-speech engine is used to implement voice feedback, allowing for offline speech synthesis, thereby providing system-based audio responses [9]. Users receive verbal notifications for scheduled, rescheduled, or canceled meetings, enhancing system interactivity. This module makes the application more user-friendly and accessible to visually impaired or elderly users.

➢ *Meeting Scheduling and Calendar Storage*

This scheduling mechanism lets users input natural language commands like "Physics class at 3 PM," which the system then parses with string extraction methods and assigns to the predefined working hours. It will store all the scheduling information, such as the name of the meeting, the time, and the list of participants, locally using JSON, with no dependency on cloud storage for privacy [13].

➢ *Graphical User Interface Development*

The GUI of MeetMate AI is developed on Tkinter, which is one of the lightest and most flexible environments for a desktop Python-based application [4]. Also, it displays the interface with the calendar, time slots, meeting name, and status in real-time (Booked/Free). It also supports options like Scheduling, Reschedule, delete meetings, Face authentication enable option, and Gesture-based control.

➢ *System Integration*

The complete system architecture integrates all modules into one workflow:

- Log in → Face authentication → Voice welcome
- Scheduling Meetings (Manual/Voice Input) → Save to JSON → Display in GUI
- WhatsApp-Based Link Sharing → Voice Confirmation
- Gesture-Based Volume Control → TTS Feedback When integrated, all these AI-based technologies create a seamless, intelligent meeting management experience, filling up the gap between conventional scheduling tools and modern automation technologies.

IV. SYSTEM ARCHITECTURE

The architecture of MeetMate AI is modular and includes several interconnected components, such as the User Interface Layer, AI and Interaction Layer, Communication Layer, and Data Storage Layer. It works with these modules to provide secure, intelligent, and touchless meeting management using computer vision, gesture recognition, voice interaction, and messaging automation [1], [3], [4], [5], [7], [8], [10], [13], and [14].

A. Overall Architectural Design:

The system architecture consists of:

➢ *User Interface Layer (Tkinter GUI) Provides the user with all input and output screens via Tkinter [4].*

Provides User with:

- Enter meeting name and time
- Provide WhatsApp numbers of participants
- Trigger scheduling, rescheduling, and deletion
- Start face authentication and gesture volume control
- Also displays the calendar view showing the time slots as either Free or Booked.

➢ *AI and Interaction Layer*

This is the core logic layer that integrates all AI-based modules:

- Face Detection and Authentication Module

Uses OpenCV and face recognition concepts, such as Viola–Jones and Eigenfaces for detection and user presence verification.

➢ *Hand Gesture Recognition Module*

- Built using MediaPipe-based real-time hand landmark tracking [3].
- Interpret hand poses for volume increase/decrease using the principles of gesture-based HCI [5], [14].

➢ *Voice Interaction Module*

- Uses pyttsx3 for offline text-to-speech in order to provide audio feedback in the case of a meeting being scheduled, rescheduled, or deleted.

➢ *Meeting Logic Module*

- Maps user input such as "Physics class at 3 PM" to a valid time slot.
- Handles conflict checking, slot updates, and status changes in the calendar.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec156

➢ *Communication Layer: WhatsApp Automation and Link Generation*

- Generates unique Jitsi meeting room links using meeting name and timestamps.
- Encodes a message containing meeting title, time, and room link.
- Utilizes web-based automation and WhatsApp Business API concepts for opening WhatsApp Web and pre-filling the messages to participants [6], [10].
- This is the layer in charge of communication outside and delivering invitations.

➢ *Data Storage Layer: JSON-Based Calendar Management*

- All details of meetings, including time, meeting name, and participants, are stored in a local JSON file.
- This avoids dependencies on cloud services and enhances both privacy and security [13].
- The system loads the JSON at startup and refreshes it whenever a change is made in the schedule.
- The same data is utilized to render the GUI calendar table.

B. Data Flow

The general data flow of MeetMate AI is as follows:

➢ *User Input Phase*

- User opens an application → face detected using camera [1], [11].
- Password entered → basic security ensured [13].
- User inputs the details of the meeting through GUI, such as "Physics class at 3 PM" along with WhatsApp numbers [4].

➢ *Processing Phase*

➢ *The Meeting Logic Module Parses the Time and Checks it Against Working Hours.*

- Calendar entry is created or updated in JSON.
- A dynamic Jitsi link is created for the meeting.

➢ *Communication Phase*

- The link and message are encoded into a WhatsApp Web URL.
- WhatsApp Web opens with pre-filled text for participants [6], [10].
- Alternatively, the meeting link is copied to clipboard for manual sharing.

➢ *Interaction Phase (Real-Time Control)*

- Voice feedback announces: "Meeting Physics class scheduled at 3 PM" using pyttsx3 [9].
- Gesture module remains active in order to control the system volume using MediaPipe hand tracking [3], [5], [14].

➢ *Persistence Phase*

- All changes are written back to the JSON calendar file for future sessions [13].
- When the application launches again, it reloads the same data and displays it.

V. RESULTS

Testing showed successful meeting scheduling, deletion, link generation, WhatsApp sharing, real-time gesture volume adjustment, and accurate face detection in varied lighting conditions. The GUI is lightweight and user-friendly. It successfully manages meeting and performs all the integrated features efficiently making the work effortless. The application was tested on a windows-based desktop with python 3.10, 8GB RAM, having a webcam and microphone support.

➢ *The Results are Categorised Below:*

Table 1 Time Consumption

Criterion	Observed Result
Average Meeting Setup Time	8-10 Seconds
WhatsApp Invitation Delivery Time	3-4 Seconds
JSON Reading/Writing Time	<1 Seconds
System Startup Time	4 Seconds
Gesture Response Delay	0.5 Seconds

Table 2 Functional Testing

Feature	Outcome	Status
Face Detection (OpenCV)	Successfully detected and validated user face in real-time	✓ Successful
Gesture-Based Volume Control (MediaPipe)	Controlled voice output accurately using thumb-pinky finger tracking	✓ Successful
Voice Feedback (pyttsx3)	Announced meeting schedules, confirmations, and deletion alerts accurately	✓ Successful
WhatsApp Automation	Generated dynamic Jitsi link and opened WhatsApp with pre-filled invitation	✓ Successful
Meeting Scheduling & Rescheduling	Correctly allocated times, updated calendar, and saved data into JSON	✓ Successful
GUI Rendering (Tkinter)	Displayed real-time calendar, time slots, and status (Free/Booked)	✓ Successful

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25dec156

➢ *Key Observations:*

- The face authentication works great in normal lighting but struggles in low-light conditions.
- The gesture-based volume control is highly accurate but it requires good visibility of the hands.
- WhatsApp automation reliably opens the browser and shares meeting invites with encoded messages.
- Local JSON storage allowed the system to be privacy-friendly, independent of any internet-based cloud systems.
- Enhanced accessibility through TTS voice feedback for the visually impaired and elderly.

VI. CONCLUSION

MeetMate AI successfully integrates various AI-driven technologies, such as face authentication, hand gesture recognition, WhatsApp automation, voice interaction, and intelligent scheduling, into a single desktop-based smart meeting management system. It offers touchless control, real-time generation of meeting links, the automatic sharing of invitations, and safe access through AI-based computer vision techniques, unlike traditional scheduling tools. Correct facial and hand landmark detection was achieved by using libraries like OpenCV and MediaPipe, while pyttsx3 allowed for the development of speech-based interaction within the application. The use of Tkinter-based GUI and JSON-based offline data persistence further empowered the application in usability while eliminating dependence on cloud-based services.

These experimental results proved the efficiency, accuracy, and real-time responsiveness of the system, showing the applicability of the system for students, teachers, professionals, and differently abled users. MeetMate AI bridges the gap between conventional meeting organizers and AI-enabled human-computer interaction by offering a secure, intelligent, and user-friendly solution.

REFERENCES

[1]. G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.
[2]. S. Z. Li and A. K. Jain, Handbook of Face Recognition, 2nd ed. New York, NY, USA: Springer, 2011.
[3]. P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), vol.1, pp. 511–518, 2001.
[4]. Turk and A. Pentland, “Eigenfaces for Recognition,” Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71–86, 1991.
[5]. Google AI, “MediaPipe: Real-Time Hand and Face Tracking,” 2023.
[6]. MediaPipe Team, “MediaPipe Hands Solution,” 2024.
[7]. Microsoft Research, “Gesture-Based Human-Computer Interaction,” IEEE Transactions on Human-Machine Systems, vol. 50, no. 3, pp. 235–244, 2023.
[8]. Pyttsx3 Documentation, “Offline Text-to-Speech Engine for Python,” 2024.
[9]. Google Cloud, “Cloud Text-to-Speech API Documentation,” 2024.
[10]. WhatsApp Inc., “WhatsApp Business API Documentation,” 2024.
[11]. A. S. Mittal and R. Kumar, “Smart Meeting Scheduler using WhatsApp API Automation,” International Journal of Advanced Computer Science and Applications, vol. 14, no. 2, pp. 92–99, 2023.
[12]. Python Software Foundation, “Tkinter GUI Documentation,” 2024.
[13]. W. Stallings, Cryptography and Network Security: Principles and Practice, 7th ed. Pearson, 2017.
[14]. B. R. Chandavarkar and V. N. Bharadi, “AI-powered meeting automation using natural language processing,” IEEE Access, vol. 10, pp. 55102–55112, Apr. 2022.
[15]. IEEE Standards Association, “Standards for Artificial Intelligence-Based Human-Computer Interaction Systems,” 2023.
[16]. JSON.org, “JSON Data Interchange Standard,” 2024.