

# VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management

Rida K. Nakhuda <sup>1,✉</sup>, Bilal N. Shaikh <sup>1</sup>, Zainab Mirza <sup>2</sup>

1. Information Technology, Somaiya Vidyavihar University, Mumbai, IND

2. Information Technology, M. H. Saboo Siddik College of Engineering, Mumbai, IND

Received: April 29, 2026 | Review began: June 07, 2026 | Review ended: July 31, 2026 | Published: August 03, 2026

© Copyright 2026

This is an open access article distributed under the terms of the Creative Commons Attribution License CC-BY 4.0., which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## Abstract

Many knowledge workers and students often find it challenging to manage personal productivity across digital tools. Most productivity applications available today only cater to one aspect of workflow management: task tracking, time management, or analytics, which causes context switching. Moreover, most productivity software has a manual one-to-one GUI approach that does not leverage the power of voice-based natural language interfaces to be more efficient. VoicePro is a comprehensive AI-powered productivity tool that is built on the advanced capabilities of GPT-4o-mini. It merges task management, Pomodoro-style focus timers, calendar scheduling, reminders, and productivity analytics in one single web app with the help of a voice assistant. VoicePro, powered by the advanced Whisper STT model of OpenAI, processes speech-to-text transcription, and the advanced GPT-4o-mini model provides natural language intent recognition. Users can perform a range of productivity operations through simple conversational voice commands. The system adopts a hybrid voice processing framework using a deterministic fallback parser to ensure reliability in case of unavailability of AI services. The design is a three-tier client-server architecture employing a React 18 single-page application as the front end, a REST API by Flask with JWT authentication, and Google OAuth 2.0 as the back end. We study VoicePro in three ways. We compare the time taken to complete tasks with voice and manual input modalities. Then, we measure the accuracy of command recognition from test utterances. According to the preliminary results, we found that users take 42% less time to create tasks through voice versus filling forms. Moreover, the percentage accuracy of the AI intent parser was 88% across six command categories.

**Categories:** AI applications, HCI Appliances and applications, Natural Language Processing (NLP)

**Keywords:** voice user interface, productivity management, natural language processing, speech recognition, task management, web application, intent parsing, pomodoro technique

## Introduction

Managing personal productivity is becoming increasingly complicated in the digital age. Knowledge workers and students interact with multiple applications daily for task management, time tracking, calendar scheduling, and performance monitoring. Research shows that switching between tasks leaves a cognitive “attention residue” that significantly impairs performance [1]. The fragmentation of productivity tools creates substantial context-switching overhead, which studies indicate can reduce productive output by as much as 40% [2]. Switching between different interfaces further increases cognitive overload, ultimately defeating the purpose of these tools.

### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

Voice user interfaces are an emerging modality that can reduce interaction friction. The rapid adoption of AI has been driven by the widespread availability of voice assistants such as Apple Siri, Google Assistant, and Amazon Alexa. Today, voice assistants have become popular, with millions of users relying on them daily to perform everyday tasks such as searching for information and controlling media content [3]. However, despite being designed for versatility, these assistants struggle to adapt to productivity-focused workflows. They can set reminders and calendar events but cannot manage priority-based task systems, initiate structured focus sessions, or provide longitudinal productivity analytics.

While applications like Todoist [4], Notion, and Asana are more advanced productivity applications with task features, they do not offer voice features. On the contrary, focus applications like Forest and Pomofocus offer a Pomodoro timer but do not provide task management or analytics. This creates a gap: to the best of our knowledge, no ecosystem exists that integrates a comprehensive productivity tool with an AI-driven conversational voice interface that understands natural human language commands - with context to boot.

VoicePro is a full-stack web application that bridges this gap through a combination of five core productivity modules, namely task management, focus timers, calendar scheduling, reminders and analytics, and a voice assistant for taking natural commands.

Through natural voice commands, users can perform multiple productivity operations with VoicePro using the OpenAI GPT-4o-mini model for intent recognition and Whisper for speech transcription.

Modern task management apps have become more than just to-dos, offering advanced task management with priority assignment, project organization, and productivity visualization. Microsoft To Do [5] integrates with the Microsoft 365 ecosystem. Notion [6] provides task management and knowledge management services using a block-based system. Operations involving “tasks” are not supported through voice by any of these platforms.

Mainstream adoption has been achieved by general-purpose voice assistants. Siri, Google Assistant, and Alexa are capable of answering a diverse set of questions, but they do not integrate into structured productivity workflows in any deeper sense [7]. Studies on task-oriented dialogue systems have investigated domain-specialized conversational agents [8], yet these ideas have rarely been applied to personal productivity management. In recent times, state-of-the-art large language models like GPT-4 [9] have significantly improved natural language understanding capabilities [10-12]. In particular, they are capable of flexible command parsing that takes context into consideration. This is different from traditional intent classification methods used in standard dialogue systems [13].

The Pomodoro technique was developed by Cirillo [14]. It allows users to work for 25 minutes and take a break. Apps like Pomofocus and Forest use this technique, but instead of being integrated into task apps, they function only as timers. According to Newport’s concept of deep work [15], individuals must engage in long sessions of focused, undistracted work. The work done by Mark et al. [16] has highlighted that, on average, interruptions to focused work require individuals 23 minutes to return to the original task. This shows the importance of integrated systems that do not require context switching.

There has been a growing interest in personal analytics tools [17]. These tools report productivity app usage and time spent on activities. Toggl affords manual time tracking and project-based reporting. Although these tools have become very popular, they do not interact with task management and focus systems, which means that they cannot tie productivity data to either tasks or work styles. According to the research conducted by Li et al. [18], users are able to identify and change their productivity patterns more efficiently when multiple dimensions of self-tracking are presented in one view.

Of all tools we evaluated, only VoicePro merged these capabilities into one system that does not have to switch.

---

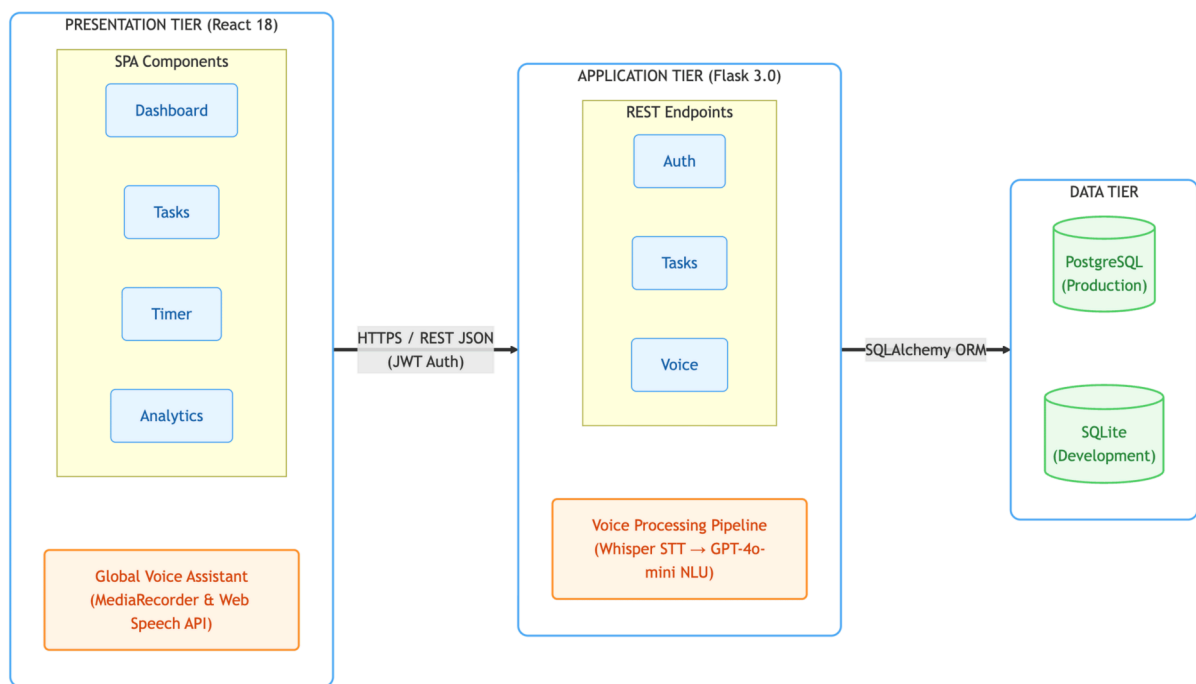
### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. *Cureus J Comput Sci* 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

## Materials And Methods

### High-level architecture

VoicePro comprises a three-tier client-server architecture featuring a presentation tier built on a React single-page application, an application tier built on a Flask REST API, and a data tier built on a SQL database. The front end is developed as a single-page React 18 application (SPA) [19]. It leverages Framer Motion for animations reliant on physics and page transitions, Recharts for interactive data visualization, and Axios for HTTP communication with the backend. A centralized AppContext provider uses React's Context API to manage the states of the authenticated user, task list, timer sessions, and reminders. The ChatbotAssistant is rendered globally, which means the user is able to use voice from any page without navigating. The backend uses Flask 3.0 [20] and follows the app factory pattern. The application utilizes SQLAlchemy as its ORM for the database, using SQLite in development and PostgreSQL in production with Render; it must support ACID guarantees and concurrency. As shown in Figure 7, VoicePro follows a three-tier architecture.



**FIGURE 1: VoicePro's Three-Tier Architecture**

The three-tier structure of VoicePro depicts its React 18 frontend, Flask REST API backend, and SQL database. Original figure by the authors.

### Voice processing pipeline

The pipeline consists of two phases:

Stage 1: Voice-to-Text: The MediaRecorder API is used to record audio in the browser. Audio blobs are sent to the server, where they are decoded using a Whisper model [21] to create text transcriptions. There is a backup SpeechRecognition library available if the OpenAI API is down.

Stage 2: Natural Language Understanding: The text produced from Stage 1, along with a default prompt indicating that the input is from user speech, is supplied with additional setting data such as commands, conversation history (up to 12 previous turns), current tasks, current reminders, and anaphora resolution (e.g., identifying who 'he' refers to in the given

### How to cite this article:

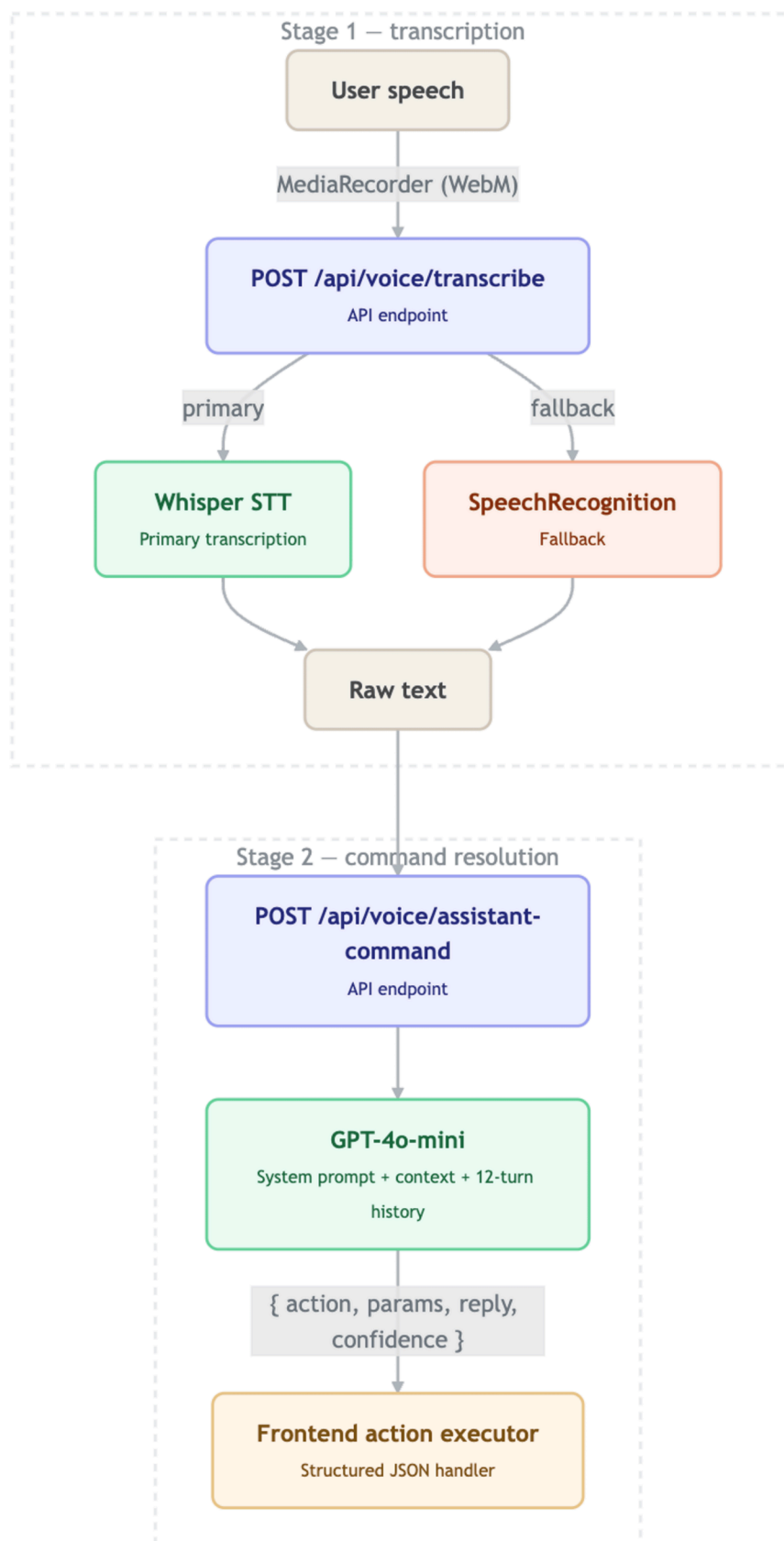
Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

context). The reply generated by GPT-4o-mini is returned as a JSON object. Each command can have a maximum of three actions, each containing a description, completion status, and confidence level. This JSON outlines an interface for intricate user commands generated from voice. The flow from audio capture to intent parsing through the two-stage voice processing pipeline is shown in Figure 2.

---

### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>



#### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

### FIGURE 2: Two-Stage Voice Processing Pipeline

The processing of voice can occur in two stages. The first stage performs speech-to-text conversion via Whisper. The second stage performs intent parsing via GPT-4o-mini with the help of conversation context.  
Original figure by the authors.

### Database schema

The system uses a relational data model with five entities that capture various aspects of user interaction with tasks. The users table contains all user credentials and related metadata. The tasks table includes 12 fields that can be used to record details about tasks. These include details such as title and category, as well as more complex attributes like priority levels and expected deadlines. The timers table is specifically designed for focus sessions. It keeps a detailed log of all sessions, which includes the type of session, duration in minutes, and start and end times. It also defines a real-world relationship with the associated work through a foreign key constraint. The reminders table keeps track of upcoming alerts that notify users about deadlines and when to follow up with others.

### Authentication architecture

VoicePro uses Google OAuth 2.0 [22] as its only authentication system. Users authenticate by going through Google's authorization page, and the application exchanges the authorization code for the user's information on the server. If there is no account associated with the user's Google email, one is created. Upon successful authentication, the server issues a 24-hour JSON Web Token (JWT) that is used for sessions. All protected APIs verify the JWT through Flask-JWT-Extended.

### Methodology

VoicePro was assessed on three parameters in this study. First, we compared the time needed to accomplish tasks by means of voice or keyboard. Second, we compared the accuracy of voice commands. Third, we compared system metrics.

### Participants

Five people took part in the preliminary evaluation: three men and two women between 20 and 24 years of age. These participants were recruited from the computer science department of the university. All of them were native speakers or proficient in English. They had used productivity applications before but had not used VoicePro. The study was reviewed by the Institutional Ethics Committee of Somaiya School of Basic and Applied Sciences, Somaiya Vidyavihar University, and granted an exemption from full ethics review (Ref: SSBAS/2026-27/MISC/43), as it involved non-invasive software usability testing with no collection of personal or sensitive data. All participants provided informed verbal consent prior to participation.

### Procedure

All participants attended one evaluation session that lasted 45 minutes. At first, each participant performed five common productivity tasks using the manual graphical user interface. Afterward, each participant performed the same tasks using voice. The order of tasks was randomized for each method.

### Implementation

#### *Frontend*

The frontend consists of 15 React components across six main pages. The Dashboard contains a real-time live view of all elements, four stat cards, a task triage matrix, a list of tasks in progress, and a timeline. The Tasks page contains a list of tasks that users can filter and sort, and it also uses color coding for priorities. The Timer page displays a circular countdown that can be used to implement the Pomodoro technique. The Analytics page displays nine different types of charts.

---

### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

The ChatbotAssistant component runs the global voice assistant, supporting live audio recording, a scrollable conversation history, rich response cards, and a voice synthesizer using the Web Speech API. The Web Speech API translates text into speech, allowing users to hear responses. ChatbotAssistant is integrated with a deterministic action executor.

### *Backend*

The AI intent parser technically forms the most important component of the project, which lies in the backend and consists of the voice\_processing module. This component uses a well-designed system prompt for instructing GPT-4o-mini. The prompt provides the current date and time in the ISO 8601 format so that the model can understand relative timing references, and it provides a buffer consisting of conversation history so that it can reference previous words in the conversation. In the event that the OpenAI API becomes unavailable, the project instead utilizes a fallback parser that is deterministic and can be trained on 15 different compiled regular expression patterns.

### *Analytics*

The analytics section of the application collects data from the 'Tasks' and 'Timers' tables. It then creates nine datasets that are then used to create visualizations. These visualizations include the Focus Time Trend, Focus Time Breakdown, Time-by-Priority, Tasks Status Distribution, Hourly Productivity Pattern, Day-of-Week Pattern, Estimated vs. Actual Time Pattern, Activity Heatmap, and Burndown Chart.

## Results And Discussion

### Experimental setup

This example was tested on a MacBook Air with an Apple M3 chip, 16 GB RAM, and a 512 GB SSD. The local server was set up with Vite, listening on port 5173, and a Flask backend, listening on port 5000. The MediaRecorder API records audio at 48 kHz and sends it to the server as WebM. Speech-to-text is performed using the OpenAI API's whisper-1 model, and natural language processing is performed using OpenAI's GPT-4o-mini model through the OpenAI Chat Completions API. Network latency is measured using Chrome DevTools. The five participants used for testing all had the same configuration: a MacBook Air, an Apple M3 chip, and the Google Chrome browser.

### Task completion time

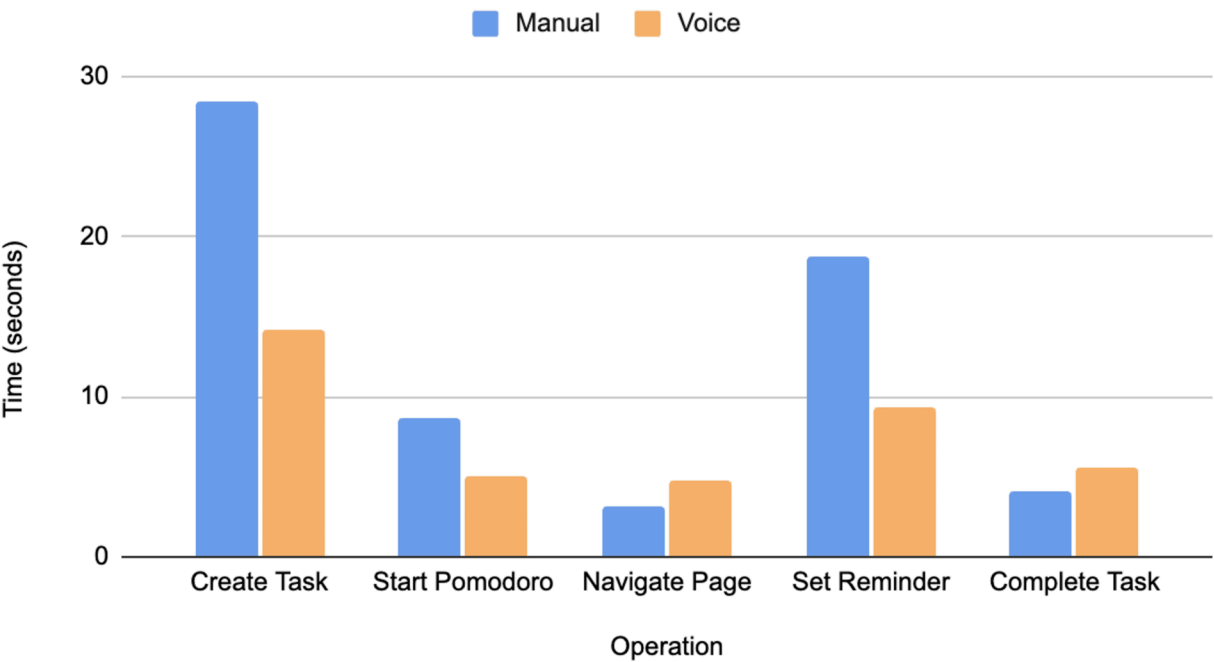
Complex multi-field operations, such as creating a task (50.0%) or setting a reminder (50.3%), save substantially more time through voice input than through typing. The manual graphical user interface is quicker for simple single-click operations, such as navigation and task completion, because of the intrinsic voice processing latency. This gives an overall weighted average reduction of 42% for multi-step workflows. A paired-samples t-test on the combined multi-step task completion times confirmed that voice input was significantly faster than manual GUI input ( $t(4) = 15.13$ ,  $p = 0.0001$ , 95% CI [23.27, 33.73]). In the t-test, we used paired completion times for multi-step tasks from the participants. Task completion time is compared across the two modalities in Figure 3.

---

### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

### Task completion time: manual vs voice input



**FIGURE 3: Task Completion Time Comparison**

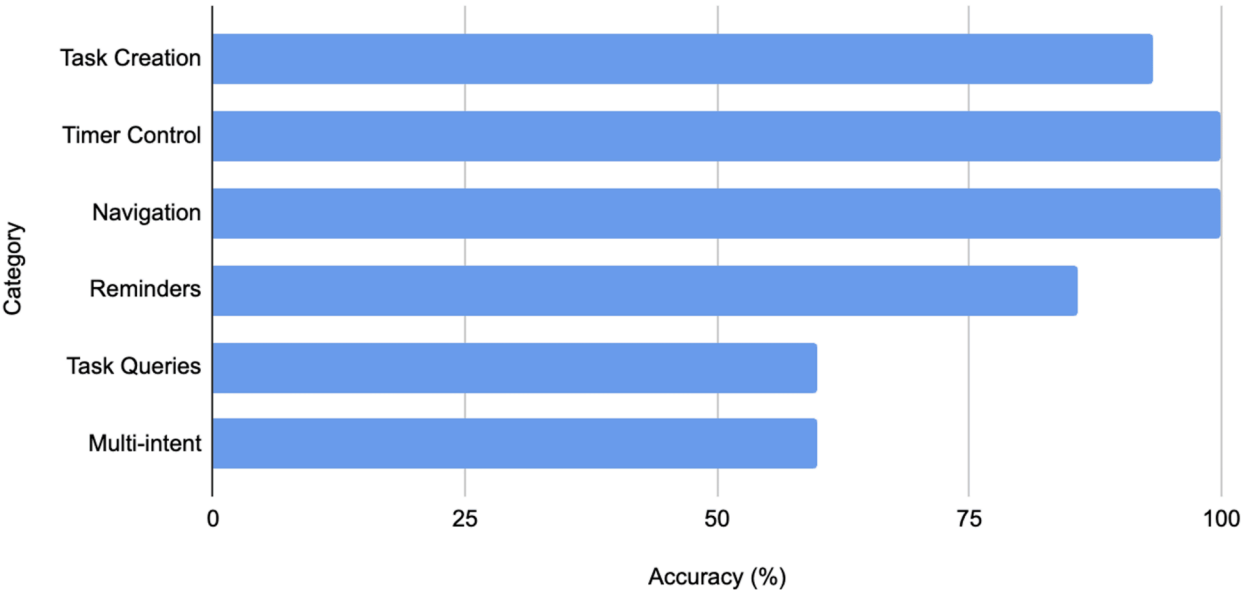
The time it takes to complete each operation using voice input and manual graphical user interface input. Standard deviation is represented on error bars.  
Original figure by the authors.

### Voice command accuracy

The system achieved 93-100% accuracy on well-defined single-intent commands, such as task creation, timer functions, and navigation. Task updates that required contextual reference resolution (60%) and complex multi-intent commands (60%) resulted in lower accuracy. Figure 4 breaks down the accuracy across all six command types. Specifically, the system achieved 93.3% accuracy for task creation (14/15 commands), 100% for timer control (10/10) and navigation (5/5), 90% for reminders (9/10), and 60% for both task queries (3/5) and multi-intent commands (3/5).

**How to cite this article:**

Voice command recognition accuracy by category



**FIGURE 4: Voice Command Recognition Accuracy**  
The overall average for voice command recognition accuracy is 88%, indicated by the dashed line.  
Original figure by the authors.

System performance

Latency: Overall latency for the complete end-to-end voice command pipeline (recording, sending audio over the network, Whisper transcription, parsing with GPT-4o-mini, action execution on the device, and UI update) is ~2.5 s. Whisper transcription: 1,240 ms. GPT-4o-mini intent parsing: 890 ms. Task CRUD operations: ~45 ms. The non-voice API queries have a consistent response time of <50 ms for CRUD operations and an average latency of <400 ms for complex analytics queries, enabling cross-year dataset analysis.

The Whisper model processes audio in WebM format at 48 kHz, sampled via the MediaRecorder API.

Key findings

The assessment demonstrates a clear advantage pattern unique to each modality. Using voice is significantly faster for complex tasks that involve multiple parameters. Creating new tasks was found to be 50% faster when using voice. Setting a reminder saw an improvement of 50.3% when using voice. One main reason for this is that voice allows users to specify multiple fields in a single natural utterance. In contrast, for simple single-action tasks, such as going to a page or performing an on-screen action, the graphical user interface has an advantage because it requires only a single click, whereas voice command input has a non-zero and irreducible latency floor of ~2.5 seconds. This leads to a desirable interaction strategy: a hybrid input paradigm that uses voice for multi-step workflows and the graphical user interface for atomic operations [23]. Recent studies on direct manipulation interfaces extend these ideas to voice-controlled large language models (LLMs), enabling advanced editing operations, such as rich task management and navigation, using conversational voice commands [24].

### **Intent parsing based on LLM and traditional natural language understanding**

GPT-4o-mini can be a practical choice for intent detection, allowing a more flexible approach than traditional task-oriented dialogue systems [13]. Manual system training often fails with out-of-vocabulary phrases because it requires explicitly labeled training data. In contrast, the LLM-based approach works with just a structured system prompt, naturally handles linguistic variation, and supports zero-shot generalization to phrasings not explicitly programmed. In addition, our zero-shot LLM approach yields a recognition accuracy of 88%, which comes very close to that of conventional natural language understanding systems requiring extensive domain-specific training. This is consistent with the IntentGrasp study [25], which showed that although frontier LLMs excel at zero-shot intent comprehension, they often underperform when handling more complicated multi-field classifications. Notably, the observed average accuracy of 60% for contextual reference resolution and multi-intent scenarios aligns with these challenges.

Nonetheless, LLM-based parsing has trade-offs, including per-request latency, API costs, as well as dependence on the uptime of the auxiliary service. The availability risk is mitigated by the deterministic fallback parser in the hybrid architecture.

The GPT-4o-mini model is instructed via a structured system prompt that includes the current date and time in ISO 8601 format, a list of the supported action types that can be performed by the model (`create_task`, `start_timer`, `set_reminder`, `complete_task`, `navigate`, and `general_query`), and a schema that includes the output fields required by the model to function properly. The prompt also includes a rolling buffer of up to 12 previous conversation turns between the model and the user in order to perform anaphora resolution. Each call to the API returns a JSON object that contains up to three actions that can be performed by the model. In addition, there is a fallback parser that uses 15 compiled regular expressions corresponding to the same action schema to provide functionality in the event that the API is unavailable.

### **Error evaluation**

The six mistakes in the accuracy of voice commands (12% failure rate) are of two types. The first type is ambiguous reference failures (three cases), where users attempted to refer to utterances by using pronouns (e.g., change that one). However, models were unable to resolve the referent due to low context. When a user issued multi-command messages such as 'create a meeting task and start a 30-minute timer for it', the model correctly detected the first command but dropped the second. This happened in three cases.

### **Restrictions**

There are some limitations to note. To begin with, the sample size of the evaluation ( $n = 5$ ) is small and exclusively sourced from computer science students. To transcribe using Whisper and process it using GPT-4o-mini, the system needs an internet connection. At present, the voice interface only supports English. The fourth limitation was that only short-term task efficiency was evaluated; longitudinal studies would determine whether the increased performance is representative of an actual increase in productivity.

## **Conclusions**

VoicePro is a voice-enabled suite for productivity based on AI. The full-stack web application couples functions like task management with Pomodoro-style focus timers, calendar scheduling, reminders, and nine-chart analytics. It also comes with a voice assistant. VoicePro empowers users to efficiently perform complex productivity operations using their natural voice through the utilization of GPT-4o-mini for natural language intent recognition and OpenAI Whisper for speech transcription. The study showed that using voice to create multi-step tasks takes less time than using the graphical user interface to create task steps. The AI intent parser achieves high accuracy across six types of commands.

In the future, we will enhance a toolbox for multilingual voice, offline voice, team-level features, and formal user studies with more ( $n \geq 30$ ), diverse, and longitudinal user studies. This will allow us to measure productivity impact and more, with a focus on exploring different avenues to alleviate current shortcomings.

---

### **How to cite this article:**

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. *Cureus J Comput Sci* 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

## Additional Information

### Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

**Concept and design:** Rida K. Nakhuda, Bilal N. Shaikh

**Acquisition, analysis, or interpretation of data:** Rida K. Nakhuda, Zainab Mirza

**Drafting of the manuscript:** Rida K. Nakhuda

**Critical review of the manuscript for important intellectual content:** Bilal N. Shaikh, Zainab Mirza

**Supervision:** Bilal N. Shaikh

### Disclosures

**Human subjects:** Consent was obtained or waived by all participants in this study. Institutional Ethics Committee (IEC), Somaiya School of Basic and Applied Sciences, Somaiya Vidyavihar University, Mumbai issued approval SSBAS/2026-27/MISC/43 (Exempt). The study was reviewed by the Institutional Ethics Committee (IEC) of Somaiya School of Basic and Applied Sciences, Somaiya Vidyavihar University, and granted an exemption from full ethics review (Ref: SSBAS/2026-27/MISC/43, dated 30.06.2026). The research involved a non-invasive software usability evaluation in which five voluntary adult participants performed timed productivity tasks on a web application. No personally identifiable information, health data, or sensitive personal data was collected. All participants provided informed verbal consent prior to participation. The study posed no physical, psychological, or social risk to the participants. **Animal subjects:** All authors have confirmed that this study did not involve animal subjects or tissue. **Conflicts of interest:** In compliance with the ICMJE uniform disclosure form, all authors declare the following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

### Data Availability Statements

All data generated or analyzed during this study are included in this published article and/or its appendices.

## References

1. Leroy S: [Why is it so hard to do my work? The challenge of attention residue when switching between work tasks](#). Organizational Behavior and Human Decision Processes. 2009, 109:168-181. [10.1016/j.obhdp.2009.04.002](#)
2. Mark G, Gudith D, Klocke U: [The cost of interrupted work: more speed and stress](#). Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2008, 107-110. [10.1145/1357054.1357072](#)
3. Ammari T, Kaye J, Tsai JY, Bentley F: [Music, search, and IoT: How people \(really\) use voice assistants](#). ACM Transactions on Computer-Human Interaction (TOCHI). 2019, 26:1-28. [10.1145/3311956](#)
4. Doist Inc: [Todoist: the to-do list to organize work and life](#). Accessed: April 15, 2026: <https://todoist.com>.
5. Microsoft Corporation: [Microsoft To Do: lists, tasks & reminders](#). Accessed: April 15, 2026: <https://www.microsoft.com/en-us/microsoft-365/microsoft-to-do-list-app>.
6. Notion Labs Inc: [Notion: your connected workspace](#). Accessed: April 15, 2026: <https://notion.so>.
7. Lopatovska I, Rink K, Knight I, et al.: [Talk to me: Exploring user interactions with the Amazon Alexa](#). Journal of Librarianship and Information Science. 2018, 51:984-997. [10.1177/0961000618759414](#)

---

### How to cite this article:

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>

8. Young T, Hazarika D, Poria S, Cambria E: [Recent trends in deep learning based natural language processing](#). IEEE Computational Intelligence Magazine. 2018, 13:55-75. [10.1109/mci.2018.2840738](#)
9. Gallifant J, Fiske A, Levites Strekalova YA, et al.: [Peer review of GPT-4 technical report and systems card](#). PLOS Digital Health. 2024, 3:e0000417. [10.1371/journal.pdig.0000417](#)
10. Wang L, Ma C, Feng X, et al.: [A survey on large language model based autonomous agents](#). Frontiers of Computer Science. 2024, 18:186345. [10.1007/s11704-024-40231-1](#)
11. Bhattacharjee A, Zeng Y, Xu SY, et al.: [Understanding the role of large language models in personalizing and scaffolding strategies to combat academic procrastination](#). Proceedings of the ACM CHI Conference on Human Factors in Computing Systems. 2024, 1-18. [10.1145/3613904.3642081](#)
12. Min B, Ross H, Sulem E, et al.: [Recent advances in natural language processing via large pre-trained language models: a survey](#). ACM Computing Surveys. 2023, 56:1-40. [10.1145/3605943](#)
13. Chen H, Liu X, Yin D, Tang J: [A survey on dialogue systems: recent advances and new frontiers](#). ACM SIGKDD Explorations Newsletter. 2017, 19:25-35. [10.1145/3166054.3166058](#)
14. Cirillo F: [The Pomodoro Technique: The Acclaimed Time-Management System That Has Transformed How We Work](#). Crown Currency, New York; 2018.
15. Newport C: [Deep Work: Rules for Focused Success in a Distracted World](#). Grand Central Publishing, New York; 2016.
16. Mark G, Gonzalez VM, Harris J: [No task left behind? Examining the nature of fragmented work](#). Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2005, 321-330. [10.1145/1054972.1055017](#)
17. Li I, Dey A, Forlizzi J: [A stage-based model of personal informatics systems](#). Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2010, 557-566. [10.1145/1753326.1753409](#)
18. Li I, Dey AK, Forlizzi J: [Understanding my data, myself: supporting self-reflection with ubicomp technologies](#). Proceedings of the ACM Conference on Ubiquitous Computing. 2011, 405-414. [10.1145/2030112.2030166](#)
19. [React Documentation: React: a JavaScript library for building user interfaces](#). Accessed: April 15, 2026: <https://react.dev>.
20. [Pallets Projects: Flask: a Python micro web framework](#). Accessed: April 15, 2026: <https://flask.palletsprojects.com>.
21. Radford A, Kim JW, Xu T, Brockman G, McLeavey C, Sutskever I: [Robust speech recognition via large-scale weak supervision](#). arXiv. 2022, [10.48550/arXiv.2212.04356](#)
22. [Google: Google OAuth 2.0 documentation](#). Accessed: April 15, 2026: <https://developers.google.com/identity/protocols/oauth2>.
23. Shneiderman B, Plaisant C, Cohen M, Jacobs S, Elmqvist N, Diakopoulos N: [Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th Edition](#). Pearson, London; 2016.
24. Colazzo L, Matera M: [Enabling voice-based direct manipulation in intent-oriented interaction paradigms](#). CHIItaly '25: Proceedings of the 16th Biannual Conference of the Italian SIGCHI Chapter. 2025, 1-2. [10.1145/3750069.3755951](#)
25. Xu L, Sanders J, Li L, et al.: [IntentGrasp: a comprehensive benchmark for intent understanding](#). Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. 2025, [10.18653/v1/2025.emnlp-main.511](#)

---

**How to cite this article:**

Nakhuda R K, Shaikh B N, Mirza Z (August 03, 2026) VoicePro: An Artificial Intelligence Voice Assistant for Integrated Personal Productivity Management. Cureus J Comput Sci 3 : es44389-026-00203-x. DOI <https://doi.org/10.7759/s44389-026-00203-x>