IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2026

DOI: 10.17148/IARJSET.2026.13616

MnemoX: An AI-Powered Personalized Memory Assistant with Mood Tracking, Goal Monitoring, and Multi-Modal Interaction

T. Amalraj Victoire[1], A. Gopinath[2]

Associate Professor, Department of Master Computer Application,

Sri Manakula Vinayagar Engineering College, Pondicherry, India[1]

Student, Department of Master Computer Application,

Sri Manakula Vinayagar Engineering College, Pondicherry, India[2]

Abstract: This paper presents MnemoX, a web-based AI-powered personal memory assistant designed to help users store, recall, and analyze their daily activities, goals, and cognitive patterns. Built on the Django framework and integrated with the Google Gemini 2.0 API, MnemoX provides a conversational interface that supports natural language memory storage, mood detection, goal tracking, image and PDF analysis, multi-language translation, Wikipedia-backed knowledge retrieval, real-time news aggregation, and voice interaction. The system employs a structured relational database via Django ORM to persist user memories, daily logs, reminders, and goals, while leveraging large language model (LLM) capabilities for intelligent summarization and contextual responses. Experimental observations demonstrate that MnemoX effectively bridges personal productivity tools and conversational AI, offering a unified, user-friendly platform for cognitive augmentation. The system addresses limitations of existing memory tools by combining multi-modal input (text, voice, image, PDF), automatic mood detection, and AI-generated reminders within a single coherent framework.

Keywords: AI memory assistant, natural language processing, mood detection, goal tracking, Django, Google Gemini API, multi-modal interaction, cognitive augmentation, conversational AI

1. INTRODUCTION

The human memory is inherently fallible and context-dependent. Research in cognitive science confirms that individuals forget approximately 50% of new information within an hour, and up to 70% within a day (Ebbinghaus, 1885). In an era of information overload, professionals, students, and individuals increasingly rely on digital tools to augment human memory. While traditional note-taking applications such as Evernote, Notion, and Google Keep offer structured storage, they lack the intelligence to analyze patterns, detect emotional states, provide proactive reminders, or engage in natural conversational recall.

Large language models (LLMs) have emerged as a transformative technology for building conversational agents capable of understanding context, summarizing documents, and responding to complex queries. Systems like ChatGPT and Google Bard demonstrate the potential of LLMs for general-purpose assistance; however, these tools are not designed for persistent personalized memory across sessions, nor do they integrate mood monitoring or goal-tracking within a unified framework.

This paper introduces MnemoX (Memory + X, denoting extensibility), a full-stack web application that addresses these limitations. MnemoX provides a Django-based backend with a SQLite/MySQL database for persistent user memory, integrates the Google Gemini 2.0 Flash API for intelligent language processing, and offers a rich web interface supporting text, voice, image, and PDF inputs. The system automatically detects the user's mood from conversational input, tracks user-defined goals by correlating them with daily activity logs, and generates proactive reminders based on behavioral patterns.

The primary contributions of this work are:

- A personalized, session-persistent conversational AI memory assistant with multi-modal input support.
- An automated mood detection module using Gemini API that classifies user emotional states from natural language.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2026

DOI: 10.17148/IARJSET.2026.13616

- A goal-tracking subsystem that correlates user-defined objectives with daily activity logs and generates AI-powered progress reports.
- An intelligent reminder generation engine using LLM inference with keyword-based fallback.
- Multi-language translation, Wikipedia knowledge retrieval, PDF/image summarization, and real-time news integration within a unified conversational interface.
- A multi-user Django web application with authentication, admin dashboard, export capabilities, and voice interaction.

2.RELATED WORK

Personal AI assistants and memory augmentation systems have been an active area of research. Early work by Lamming and Flynn (1994) on the Forget-Me-Not system explored continuous capture of personal interactions to aid memory recall. More recently, lifelogging systems such as MyLifeBits (Bell and Gemmell, 2007) proposed capturing all digital experiences; however, such approaches raise significant privacy concerns and offer limited intelligent analysis.

Conversational agents have been widely studied. ALICE (Wallace, 2009) used pattern-matching AIML rules, while modern systems leverage transformer-based models. Zhao et al. (2020) proposed personalizing dialogue agents using explicit memory networks, demonstrating improved response quality with stored user facts. Xu et al. (2022) extended this with long-term memory in open-domain chatbots, a problem MnemoX addresses through database-backed memory with LLM retrieval.

Mood detection from text has been approached using lexicon-based methods (Liu, 2012), SVM classifiers (Pang and Lee, 2008), and more recently transformer models (Devlin et al., 2019). MnemoX leverages Gemini's zero-shot classification capability for mood detection, avoiding the need for labeled training data.

Goal-tracking systems have been explored in behavior change research. Fogg (2009) proposed a behavior model emphasizing motivation and ability triggers. Digital goal-tracking applications like Habitica and Strides provide manual tracking; however, none integrate AI-driven progress analysis from unstructured daily activity logs, which is a distinguishing feature of MnemoX.

Multi-modal AI systems combining text, image, and document analysis have been enabled by foundation models such as GPT-4V (OpenAI, 2023) and Gemini (Google DeepMind, 2023). MnemoX specifically leverages Gemini Vision for image analysis and Gemini text models for PDF summarization, integrating these capabilities within a persistent personal assistant context.

3. SYSTEM ARCHITECTURE

MnemoX is implemented as a three-tier web application comprising a presentation layer (HTML/CSS/JavaScript templates), a business logic layer (Django views and AI modules), and a data persistence layer (Django ORM with SQLite/MySQL). Figure 1 illustrates the high-level system architecture.

3.1 Technology Stack

Component	Technology	Purpose
Backend Framework	Django 5.2	Web server, ORM, authentication, routing
AI / LLM	Google Gemini 2.0 Flash	NLP, mood detection, summarization, translation
Database	SQLite / MySQL	Persistent memory, goals, reminders, profiles
Knowledge Base	Wikipedia REST API	Fallback knowledge retrieval
Voice Input	SpeechRecognition	Browser-based voice-to-text
Text-to-Speech	pyttsx3	Assistant voice output
PDF Processing	PyPDF2	Text extraction from PDF files
Export	Pandas + ReportLab	CSV and PDF history export
Frontend	Bootstrap 5 / HTML5	Responsive UI, charts, mood visualizations

Table 1: MnemoX Technology Stack

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2026

DOI: 10.17148/IARJSET.2026.13616

3.2 Data Models

The system defines five core Django ORM models that constitute the persistent memory layer:

- AssistantMemory: Stores all user query–response pairs, with fields for memory_key, memory_value (for structured facts), and mood (detected emotional state).
- DailyMemory: Records date-tagged activity logs used for date-based recall, weekly summaries, goal correlation, and reminder generation.
- Reminder: Stores AI-generated or user-defined reminders with completion tracking.
- Goal: Tracks user-defined objectives with status (active/completed/abandoned), last progress check timestamp, and AI-generated progress reports.
- Profile: Extends the Django User model with a profile picture using a one-to-one relationship, automatically created via Django signals.

3.3 AI Processing Pipeline

The central AI processing pipeline is implemented in logic.py and follows a priority-based command dispatch architecture. When a user submits a query, the process_command() function evaluates it against a sequence of pattern matchers in the following priority order:

1. Explicit website/system commands (open Google, open Notepad, etc.)
2. Mathematical expression detection and evaluation using Python's safe eval()
3. Dictionary lookup via the Free Dictionary API
4. Language translation via Gemini API
5. Entertainment (jokes, motivational quotes) using local response pools
6. To-do list management using database-backed memory keys
7. Goal creation, progress checking, and status updates
8. Mood self-inquiry commands
9. Memory save/recall patterns using regex ("my X is Y", "remember that", "what is my X")
10. Date-based activity logging and recall via DailyMemory model
11. Today/weekly summary generation via Gemini
12. URL analysis and webpage summarization
13. News headline aggregation from Google News RSS
14. Countdown timer computation against stored date memories
15. Study helper and quiz generation via Gemini
16. Programming code generation via Gemini with local fallback solutions
17. General conversational fallback to Gemini with last 6-message chat history

The gemini_reply() function manages all Gemini API calls with automatic retry on HTTP 429 (rate limit) and transparent fallback to Wikipedia-based retrieval via a two-step search+summary algorithm. This ensures availability even when API quotas are exceeded.

4.KEY SYSTEM MODULES

4.1 Mood Detection Module

The mood detection module uses zero-shot classification via the Gemini API. For each user query, the detect_mood() function constructs a constrained prompt instructing Gemini to classify the emotional content into one of ten categories: happy, sad, angry, anxious, excited, stressed, calm, frustrated, grateful, or neutral. The returned mood label is validated against the known set, defaulting to 'neutral' on unrecognized outputs. Detected moods are persisted with each AssistantMemory record and visualized as a time-series mood chart using Chart.js on the frontend. Users can query their mood trend using natural language commands such as 'what is my mood today?'

4.2 Goal Tracking and Progress Analysis

Users define goals using natural language patterns ('set goal: exercise 5 days a week'). The system creates a Goal model instance and subsequently correlates it with DailyMemory events over rolling 7-day windows. The check_goal_progress() function assembles a structured prompt containing the goal title and a chronological list of recent activities, then requests a 2–3 sentence progress analysis with an estimated completion percentage from Gemini. When the API is unavailable, a keyword-matching fallback counts DailyMemory events containing goal-related terms and generates a simple frequency-based report.

4.3 Intelligent Reminder Generation

After each daily activity is logged, the generate_reminder() function analyzes the 10 most recent DailyMemory entries

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2026

DOI: 10.17148/IARJSET.2026.13616

within the past 7 days and constructs a Gemini prompt requesting a short actionable follow-up reminder (max 12 words). A two-tier fallback system is employed: first, a keyword-to-reminder dictionary (20 keywords covering gym, study, hospital, meeting, exam, etc.) is checked against the most recent event; if no match is found, a generic encouragement message is returned. Generated reminders are appended to the assistant's response with a ' REMINDER__' marker for frontend extraction and display.

4.4 Multi-Modal Input Processing

MnemoX accepts three types of non-text input:

4.4.1 Image Analysis: Uploaded images are encoded as Base64 and submitted to the Gemini Vision API (gemini-2.0-flash-lite) with a user-provided or default question. The response is stored as an AssistantMemory record with a camera emoji prefix.

4.4.2 PDF Summarization: PDF files are processed using PyPDF2 to extract up to 3,000 characters of text, which is then summarized by Gemini in 5–6 sentences covering the main topic, key points, and conclusions.

4.4.3 Voice Input: Browser-based Web Speech API captures voice queries, which are transcribed client-side and submitted as text. The assistant response is optionally spoken back using pyttsx3 TTS on the server side.

4.5 Memory Architecture

MnemoX distinguishes between two types of memory storage. Semantic memory stores structured facts using key-value pairs in AssistantMemory (e.g., 'girlfriend → Priya', 'college → ABC University'). Episodic memory stores date-tagged events in DailyMemory. Retrieval uses both direct database queries (exact and partial key matching) and pattern-based natural language extraction. Chat history context is maintained by fetching the 6 most recent AssistantMemory records and formatting them as Gemini conversation turns, enabling contextually aware multi-turn dialogue.

5. IMPLEMENTATION DETAILS

5.1 API Integration

All Gemini API calls are performed using Python's standard urllib library without external HTTP dependencies, ensuring minimal deployment requirements. The system uses the gemini-2.0-flash-lite model endpoint for all text generation tasks. Each request includes a system instruction defining the assistant's persona as a memory assistant. Generation parameters are tuned per task: temperature 0.7 and maxOutputTokens 500 for conversational responses; temperature 0.2 and maxOutputTokens 800 for code generation; temperature 0.1 and maxOutputTokens 200 for translation tasks.

5.2 User Interface

The frontend is implemented using Django templates with Bootstrap 5 for responsive design. The dashboard presents a chat-style interface with messages ordered chronologically (oldest at top). Reminders are displayed as a collapsible sidebar panel. The mood chart page renders emotion distributions using Chart.js. The goals page provides cards for each active goal with a 'Check Progress' button that triggers an AJAX-style page reload with fresh AI analysis. The admin dashboard provides aggregate statistics including total users, total memory records, most active users, and recent activity.

5.3 Export and Data Portability

MnemoX provides two export formats for conversation history. CSV export uses pandas DataFrame to serialize AssistantMemory records into a downloadable file. PDF export uses ReportLab's canvas API to generate a formatted A4 report with user name, date, and paginated query–response pairs. Both exports are restricted to authenticated users and scoped to their own data.

6. EXPERIMENTAL OBSERVATIONS

The system was tested with multiple user accounts over a two-week observational period covering a range of conversational scenarios. The following table summarizes key functional observations:

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2026

DOI: 10.17148/IARJSET.2026.13616

Feature	Result	Notes
Memory save & recall	Accurate	Pattern matching correctly extracts key-value facts
Mood detection accuracy	~87%	Tested on 30 sample inputs vs. human labels
Goal progress correlation	High	Gemini correctly identifies related activities
Gemini API response time	1.2–3.4s	Average over 50 queries under normal load
Wikipedia fallback trigger rate	~12%	Occurs on API rate limit (429) responses
Image analysis (Gemini Vision)	Functional	Accurate description for standard JPEG/PNG inputs
PDF summarization	Functional	Text-based PDFs only; scanned PDFs not supported
Voice input (Web Speech API)	Browser-dependent	Requires Chrome/Edge; not supported in Firefox
Translation (20 languages)	Accurate	Tamil, Hindi, French, German, Japanese tested
Multi-user isolation	Enforced	All queries filtered by user FK in ORM queries

Table 2: System Functional Observations

7. DISCUSSION

7.1 Strengths

MnemoX demonstrates that a lightweight Django application can provide LLM-backed memory augmentation without requiring expensive infrastructure. The priority-based command dispatcher efficiently handles diverse query types with deterministic rule-based routing for common patterns, reserving API calls for genuinely open-ended queries. The dual-layer fallback strategy (API → Wikipedia → local rules) ensures consistent system availability even under rate-limiting conditions.

The combination of episodic (DailyMemory) and semantic (key-value AssistantMemory) storage models aligns with cognitive science frameworks of human memory organization, enabling both factual recall ('what is my girlfriend's name?') and event-based queries ('what did I do on March 15?'). The automatic mood detection provides passive emotional monitoring without requiring explicit user input, potentially serving therapeutic and productivity applications.

7.2 Limitations

The current implementation has several limitations. First, the system depends on the Gemini API's free tier, which imposes rate limits that trigger fallback responses under heavy usage. Second, the voice interaction is browser-dependent and relies on the Web Speech API, which is not universally supported. Third, PDF processing is limited to text-based documents; scanned or image-based PDFs cannot be processed without OCR integration. Fourth, the mood detection relies entirely on Gemini's zero-shot capabilities without domain-specific fine-tuning, which may reduce accuracy on ambiguous or sarcastic text. Fifth, the goal tracking is based solely on textual keyword correlation rather than structured behavioral metrics.

7.3 Comparison with Existing Systems

Compared to general-purpose AI assistants (ChatGPT, Gemini Chat), MnemoX provides persistent cross-session memory, mood tracking, and goal management within a private, user-controlled environment.

Compared to traditional note-taking apps (Notion, Evernote), MnemoX offers conversational recall, AI-driven summarization, and proactive reminders. The combination of these features in a unified open-source Django application represents a distinct contribution in the personal AI assistant space.

8. FUTURE WORK

Several directions are identified for future development:

- OCR Integration: Incorporating Tesseract OCR to enable analysis of scanned PDF documents and images containing text.
- Fine-tuned Mood Model: Training a lightweight BERT-based classifier on emotion datasets (e.g., SemEval, GoEmotions) to improve mood detection accuracy and reduce API dependency.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2026

DOI: 10.17148/IARJSET.2026.13616

- Mobile Application: Developing a React Native or Flutter mobile frontend with offline voice capture and background reminder notifications.
- Retrieval-Augmented Generation (RAG): Implementing vector embeddings (using FAISS or pgvector) of stored memories to enable semantic similarity-based recall beyond keyword matching.
- Wearable Integration: Connecting with smartwatch health APIs to automatically log physical activity data and correlate it with goal progress.
- Privacy-Preserving Mode: Adding end-to-end encryption for memory records and an optional local LLM backend (e.g., Ollama + LLaMA 3) to eliminate API dependency for privacy-sensitive users.
- Calendar Integration: Synchronizing with Google Calendar API to automatically log events and trigger countdown reminders for scheduled activities.

9. CONCLUSION

This paper presented MnemoX, a full-stack AI-powered personal memory assistant that integrates Google Gemini 2.0 LLM capabilities within a Django web application. MnemoX addresses the gap between passive note-taking tools and intelligent conversational assistants by providing persistent, user-specific memory across sessions, automated mood detection from natural language, AI-driven goal tracking, and multi-modal input processing including text, voice, image, and PDF.

The system's priority-based command dispatch architecture efficiently routes queries to specialized handlers, while its dual-layer API fallback ensures availability under rate-limiting conditions. Experimental observations confirm the functional correctness of all major modules, with mood detection achieving approximately 87% accuracy on tested samples.

MnemoX demonstrates that conversational AI memory augmentation is achievable with lightweight, accessible technology stacks, opening pathways for further research in personalized cognitive assistance, behavioral analytics, and privacy-preserving AI memory systems. The source code is available for academic and research purposes.

REFERENCES

[1] Bell, G., & Gemmell, J. (2007). A digital life. Scientific American, 296(3), 58–65.
[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT, 4171–4186.
[3] Django Software Foundation. (2024). Django documentation (v5.2). https://docs.djangoproject.com/
[4] Ebbinghaus, H. (1885). Memory: A contribution to experimental psychology. Teachers College, Columbia University.
[5] Fogg, B. J. (2009). A behavior model for persuasive design. Proceedings of the 4th International Conference on Persuasive Technology (Persuasive '09). ACM.
[6] Google DeepMind. (2023). Gemini: A family of highly capable multimodal models. Technical Report. Google LLC.
[7] Lamming, M., & Flynn, M. (1994). Forget-me-not: Intimate computing in support of human memory. Proceedings of FRIEND21, 4th International Symposium on Next Generation Human Interface.
[8] Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1–167.
[9] OpenAI. (2023). GPT-4 technical report. arXiv preprint arXiv:2303.08774.
[10] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2), 1–135.
[11] Wallace, R. S. (2009). The anatomy of ALICE. In Parsing the Turing Test (pp. 181–210). Springer, Dordrecht.
[12] Xu, J., Szlam, A., & Weston, J. (2022). Beyond goldfish memory: Long-term open-domain conversation. Proceedings of ACL 2022, 5180–5197.
[13] Zhao, W. X., et al. (2020). A survey of personalized dialogue systems. arXiv preprint arXiv:2007.04523.