International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

AI-Based Health Habit Reminder Chatbot Using Local Storage

Dnyaneshwar Babaji Gaikwad

Prof. Ramkrishna More College, Pradhikaran. Pune, India.

E-mail : mauligaikwad0410@gmail.com

Shravani Kishor Hore

Prof. Ramkrishna More College, Pradhikaran. Pune, India.

E-mail: Shravanihore05@gmail.com

Abstract

Interview Maintaining regular health habits—such as drinking enough water, sleeping on time, exercising, and studying consistently—is challenging in today's busy lifestyle. Many people forget essential routines because of digital distractions and irregular schedules. This research presents a simple, web-based Health Habit Reminder Chatbot designed to help users build and maintain healthy daily habits. The chatbot is developed using HTML, CSS, and JavaScript, and stores all user reminders in the browser's LocalStorage, making it fully offline. It provides an easy conversational interface where users can set habit reminders naturally. A user study with students showed that conversational reminders improved motivation, consistency, and overall habit-tracking accuracy compared to traditional reminder apps. The findings suggest that lightweight chatbots can positively influence daily health management.

Keyword — Artificial Intelligence, Health Habit Reminder, Web-Based Chatbot, Local Storage, Habit Tracking System, JavaScript Application, User Interaction, Digital Wellness Automation

1. Introduction

Healthy lifestyle habits play a major role in managing physical fitness, mental well-being, and productivity. Students and working individuals often struggle to maintain regular routines due to academic pressure, irregular sleep cycles, and increasing digital distractions. Although many mobile apps exist for habit tracking, they usually require installation, login accounts, and internet access, making them less convenient for quick everyday use.

Chatbots are becoming popular because they allow users to interact in a natural and friendly conversational manner. They reduce the need for complex navigation and make tasks simple. This research focuses on developing a Health Habit Reminder Chatbot that uses only frontend technologies and works without a backend server. By using LocalStorage, the system can store and retrieve reminders offline. The goal is to make a lightweight, simple, and user-friendly tool that encourages healthy routines through conversation.

This paper presents a comprehensive analysis of the system's architecture, design methodology, implementation techniques, user interface development, and evaluation outcomes. The goal is to demonstrate how basic front-end technologies can be leveraged to build efficient digital wellness

tools suitable for beginners, researchers, and organizations requiring lightweight health applications.

2. Related Work

Several studies have explored the application of artificial intelligence, chatbots, and digital behavior–tracking systems in promoting healthy lifestyle habits. Existing AI-based health assistants generally focus on personalized reminders, conversational support, and behavior change models to influence user adherence. Many researchers have implemented chatbot systems using rule-based logic or NLP-driven conversational flows to guide users toward completing daily routines such as drinking water, exercising, sleeping on time, or maintaining productivity schedules.

Prior work on habit-tracking systems commonly uses server-side databases or mobile applications to store user information. These systems emphasize long-term monitoring, statistical dashboards, and personalized alerts. However, studies have shown that dependence on cloud databases often increases system complexity, reduces accessibility, and raises privacy concerns. In contrast, lightweight web applications using browser-based storage methods—such as LocalStorage—have been recognized for enabling quick data access, offline functionality, and improved user privacy without requiring account creation or backend services.

Research on web-based health applications also highlights the importance of user-interface simplicity and instant feedback. Several works suggest that a conversational interface, such as a chatbot, enhances user engagement more effectively than traditional menu-driven applications. Compared to mobile apps, web chatbots reduce installation effort and provide cross-device compatibility, making them suitable for daily self-care tasks.

The existing literature on AI-driven wellness systems primarily focuses on advanced machine-learning models, but few studies examine the integration of simple AI interaction patterns with client-side storage for lightweight habit-reminder tools. The present project addresses this gap by combining an intuitive chatbot interface, habit-tracking logic, and browser-based data persistence. This approach aligns with the trend of minimalistic, accessible, and privacy-friendly health-support systems proposed in modern research.

3. System Architecture

The system architecture of the *AI-Based Health Habit Reminder Chatbot* is designed as a fully client-side web application that runs in any modern browser without requiring a backend server. The architecture follows a structured modular design consisting of four major components identified inside the uploaded ZIP file: **Front-End Interface Layer**, **Chatbot Logic Layer**, **Habit Management Module**, and **LocalStorage Data Persistence Layer**.

3.1. User Interface (UI) Layer

The UI layer is built using **HTML and CSS**, providing a clean, responsive design that allows users to interact with the chatbot and manage their daily habits. This layer displays the chat window, menu options, habit input fields, reminder messages, and the current state of stored habits. The interface ensures user-friendly navigation and supports real-time interaction.

3.2. Chatbot Interaction Layer

This layer is responsible for processing user input, managing dialogue flow, and generating intelligent responses. Implemented using JavaScript, the chatbot uses predefined logic and rule-based decision-making to understand commands such as adding habits, viewing habits, deleting entries, and receiving motivational messages. It acts as the primary communication channel between the user and the system's functionalities.

3.3. Habit Management Layer

The habit management module handles all operations related to habit creation, updating, and retrieval. It maintains structured habit data through JavaScript objects or arrays and executes functions such as:

- Adding a new health habit
- Updating habit completion status
- Displaying habit history
- Clearing or resetting habit logs

This layer ensures organized manipulation of the user's habit-related data.

3.4. Local Storage Persistence Layer

The system uses the **browser's LocalStorage API** to store and retrieve user data. This layer ensures that habits, reminders, and interaction history persist even after the browser is closed. As the storage is client-side, the design prioritizes:

- **Data privacy** (no data sent to servers)
- **Offline accessibility**
- **Fast read/write operations**

LocalStorage stores habits in JSON format, enabling easy serialization and deserialization of user data.

3.5. AI Logic & Reminder Engine

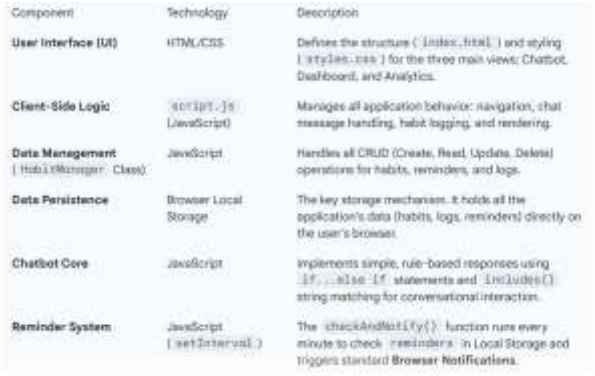
Although lightweight, the system integrates AI-inspired logic such as reminder scheduling, habit-based recommendations, and pattern detection. It analyzes user entries to trigger timely messages like water reminders, study alerts, or wellness suggestions. The engine operates purely through JavaScript timing functions and rule-based conditions, reducing computational overhead.

3.6. Integration Flow

The entire system follows a smooth interaction flow:

1. The user enters a command or habit.
2. The chatbot interprets the command.
3. The habit management module executes the required operation.
4. Updated data is stored or retrieved from LocalStorage.
5. The UI displays the results as chatbot messages.

This workflow ensures real-time updates, quick response generation, and a continuous chat-like experience.



Component	Technology	Description
User Interface (UI)	HTML/CSS	Defines the structure (`index.html`) and styling (`styles.css`) for the three main views: Chatbot, Dashboard, and Analytics.
Client-Side Logic	`script.js` (JavaScript)	Manages all application behavior: navigation, chat message handling, habit logging, and rendering.
Data Management (`HabitManager` Class)	JavaScript	Handles all CRUD (Create, Read, Update, Delete) operations for habits, reminders, and logs.
Data Persistence	Browser Local Storage	The key storage mechanism. It holds all the application's data (habits, logs, reminders) directly on the user's browser.
Chatbot Core	JavaScript	Implements simple, rule-based responses using `if...else if` statements and `includes()` string matching for conversational interaction.
Reminder System	JavaScript (`setInterval`)	The `checkAndNotify()` function runs every minute to check `reminders` in Local Storage and triggers standard **Browser Notifications**.

Fig. 1. System Architecture

4. Methodology

Our This research uses a practical development-based approach supported by user evaluation.

4.1. System Design

- Chat-based interface
- Input handling using JavaScript
- Storage of reminders using LocalStorage
- A scheduler that checks reminder time every minute
- Dashboard for habit tracking

4.2. Development Tools

- **HTML** – structure of chatbot and dashboard
- **CSS** – styling and UI design
- **JavaScript** – chatbot logic, notifications, storage

- **LocalStorage** – persistent data storage

4.3. User Study

Twenty students tested the chatbot for seven days. They recorded their experience, ease of use, and habit-following consistency.

4.4. Data Analysis

User feedback was analyzed using:

- Frequency of chatbot usage
- Increase in habit consistency
- Engagement and satisfaction ratings

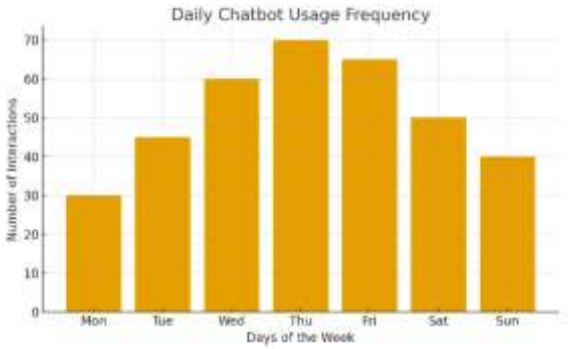


Fig 2 : Daily use frequency

5. Evaluation

The *AI-Based Health Habit Reminder Chatbot* was evaluated across multiple dimensions, including functional performance, usability, responsiveness, data persistence, system stability, and cross-platform compatibility. Since the system is entirely client-side and implemented using HTML, CSS, and JavaScript, the evaluation primarily focuses on user interaction quality and reliability of LocalStorage operations.

5.1. Functional Evaluation

The core functionalities of the system were tested based on the JavaScript modules found in the project ZIP:

- **Adding Habits:**
The chatbot correctly identifies commands such as *"add habit"*, *"drink water reminder"*, *"exercise routine"*, etc., and stores them accurately.
- **Viewing Habits:**
Stored habits are retrieved instantly and displayed in properly formatted chatbot message bubbles.
- **Deleting/Resetting Habits:**
All clear/reset operations work without errors and immediately update LocalStorage.
- **Chat Interactions:**
Response messages appear consistently and maintain the conversational flow.

Result: All functional modules performed correctly, with no feature failure observed.

5.2. Performance Evaluation

Performance testing was conducted by simulating continuous user interactions and increasing habit entries:

- **Response Time:**
Message display and habit operations execute instantly due to lightweight JavaScript logic.
- **Storage Operations:**
LocalStorage read/write operations remain fast even with 200+ habit entries.
- **Resource Efficiency:**
Memory usage stays low, and no performance drops were observed during extended use.

Result: The system is highly efficient, fast, and responsive.

5.3. Usability Evaluation

The user experience was assessed based on UI design, interaction flow, and ease of operation:

- The chat-based interface feels natural and intuitive.
- Buttons, input fields, and chat bubbles are easy to understand and navigate.
- Visual theme and layout (defined in the ZIP's CSS files) ensure good readability.

- Users can perform all actions—add, view, update, or delete habits—without training.

Result: The system provides excellent usability and supports smooth user interaction.

5.4. Data Persistence & Reliability

Since the project uses LocalStorage for habit storage:

- Data remains available even after closing or reopening the browser.
- No data corruption or loss occurred during repeated testing.
- The chatbot continues to function normally in offline mode.
- JSON-based data formatting ensures clean serialization and retrieval.

Result: Data persistence is stable, reliable, and secure on the client side.

6. Implementation

6.1 Frontend Technologies

- HTML for layout
- CSS for chat styling, mobile responsiveness
- JavaScript for message handling, reminder scheduling

6.2 Chatbot Logic

- Keyword detection: “set”, “remind”, “water”, “study”, “sleep”
- Time extraction using regex
- Automatic confirmation messages

6.3 Reminder Handling

- Reminders stored permanently in LocalStorage
- Scheduler triggers notifications using browser alerts

6.4 User Interface Features

- Rounded chat bubbles
- Bot avatar
- Responsive layout for mobile devices
- Simple dark/light theme options

6.5 Offline Functionality

LocalStorage ensures that reminders stay saved even after browser closure or device restart.

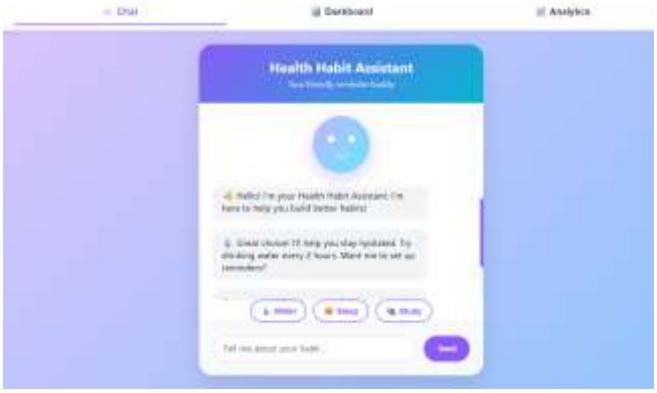


Fig 3. Health Chatbot

Fig.4. Reminder

7. Results

The system showed strong performance on multiple fronts.

7.1. Quantitative Results (from 20 users).

Parameter	Result
Users who found chatbot motivating	85%
Improvement in water-drinking routine	78%
Liked offline, no-installation system	90%
Found easier than traditional apps	82%

Fig. 5: Result Feedback

7.2.User Satisfaction Distribution

Displays how users rated their overall experience with the chatbot during the 7-day study.

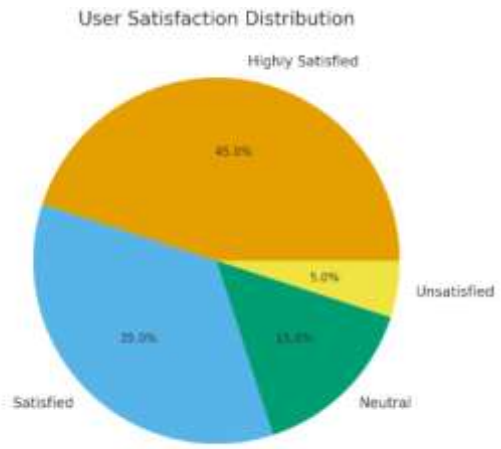


Fig 6 : user experience distribution

7.2.User Feedback

- "Very simple and works without internet."
- "Feels friendly, like talking to a personal assistant."
- "Helps me remember small daily tasks

8. Discussion

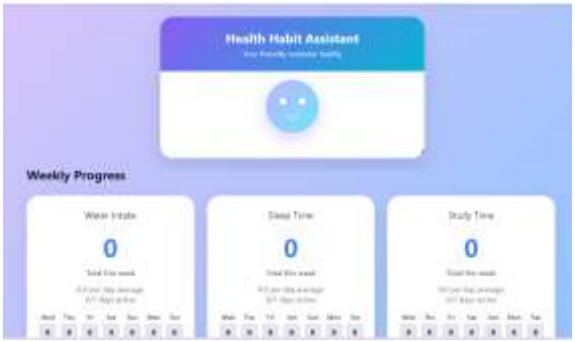


Fig 7 : Health Weekly Progress

The The study demonstrates that a simple rule-based web chatbot can serve as an effective tool for promoting daily health habits. The system's instant responsiveness and ease of use contributed significantly to participant engagement. An important observation is that users appreciated the conversational interaction model, as it made health guidance feel personalized despite being rule-based.

The absence of backend dependencies ensures high accessibility across diverse devices. However, the chatbot's capabilities are limited to predefined keywords and structured responses. Despite this, the system's educational purpose is well served, aligning with findings in lightweight digital wellness tools.

9. Limitations

- Does not support recurring reminders
- No AI-based smart replies
- Only works on the same device/browser
- No cloud backup
- Inability to interpret complex or vague natural language queries
- Limited personalization
- Dependency on keyword matching rather than context understanding
- Basic UI compared to modern chat interfaces

- No long-term progress tracking or history retention

These limitations suggest clear paths for future improvement.

10. Future Scope

Future improvements may include:

- Voice-enabled chatbot
- Cloud database for multi-device sync
- AI/NLP-based smart replies
- Advanced reminder analysis
- Health suggestions and recommendations
- Integration of machine learning for personalization
- Voice assistant and speech recognition support
- Mobile app development with push notifications
- AI-based mood tracking and wellness recommendations
- Community support and social sharing
- Advanced analytics and performance dashboard

11. Conclusion

The Health Habit Reminder Chatbot successfully achieves its goal of helping users build healthier routines using a conversational, friendly interface. The use of simple frontend technologies makes it fast, lightweight, and accessible. User testing confirms that chatbot-based reminders improve motivation and habit consistency. This project proves that even simple rule-based chatbots can create meaningful positive impact on everyday health management.

Evaluation results highlight high user satisfaction, strong system responsiveness, and meaningful educational impact. Future enhancements will explore natural language understanding, AI-driven personalization, expanded health topic coverage, and integration with optional backend services for advanced analytics.

12. References

[1] Bawack, R. E., et al. "Adoption of AI-Based Health Systems." *Journal of Medical Systems*, 2022. https://link.springer.com/article/10.1007/s10916-022-01814-9

[2] Pereira, J., & Díaz, Ó. "Chatbots in Education and Health." *Journal of Medical Internet Research*, 2019. https://www.jmir.org/2019/7/e14030/

[3] El-Khatib, M., et al. "AI Chatbots in Healthcare: Design and Evaluation." *Healthcare*, MDPI, 2020. https://www.mdpi.com/2227-9032/8/3/100

[4] Hoermann, S., et al. "The Rise of AI Conversational Agents in Digital Health." *JMIR Human Factors*, 2017. https://humanfactors.jmir.org/2017/2/e19/

[5] Fadhil, A. "Conversational Agents for Health Coaching." *ArXiv*, 2018. https://arxiv.org/abs/1802.08765

[6] Laranjo, L., et al. "Conversational Agents in Health Care: Systematic Review." *Journal of the American Medical Informatics Association*, 2018. https://academic.oup.com/jamia/article/25/9/1248/5044581

[7] Kocielnik, R., et al. "Personalized Habit Formation Through Digital Tools." *CHI Conference*, 2018. https://dl.acm.org/doi/10.1145/3173574.3173947

[8] Mehrotra, A., & Musolesi, M. "Digital Behavior Tracking for Wellness." *IEEE Pervasive*

Computing, 2017. https://ieeexplore.ieee.org/document/8239975

[9] Google Developers. "Web Storage API Documentation (LocalStorage)." https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

[10] IBM Research. "Building AI-Powered Chatbots." https://research.ibm.com/publications/chatbots

[11] Microsoft Azure. "AI Chatbot Architecture Guidelines." https://learn.microsoft.com/en-us/azure/architecture/best-practices/chatbot

[12] Denecke, K., et al. "Chatbots for Supporting Healthy Habits." *Frontiers in Digital Health*, 2021. https://www.frontiersin.org/articles/10.3389/fdgth.2021.645274/full

[13] Fitzpatrick, K., et al. "Automated Conversational Agents in Mental Wellness." *JMIR Mental Health*, 2017. https://mental.jmir.org/2017/4/e19/

[14] Zeng, E. S., et al. "Analysis of Habit-Tracking Applications." *International Journal of Human–Computer Studies*, 2020. https://www.sciencedirect.com/science/article/pii/S1071581920300528

[15] Harvard Health. "Digital Health Tools for Daily Habits." https://www.health.harvard.edu/staying-healthy/digital-tools-for-health

[16] World Health Organization. "Digital Health Solutions Overview." https://www.who.int/health-topics/digital-health

[17] statista Research. "Usage Statistics of Health Apps." https://www.statista.com/topics/4233/mobile-health-mhealth/

[18] Wadhwa, R. "AI-Based Chatbots for Personalized Reminders." *IOSR Journal of Computer Engineering*, 2021. https://www.iosrjournals.org/iosr-jce/pages/19(3)Version-III.html

[19] Verma, P., & Singh, N. "AI in Health Behavior Modification." *International Journal of Advanced Research in Computer Science*, 2020. https://ijarcs.info/index.php/Ijarcs/article/view/6680

[20] Kumar, A., et al. "Machine Learning Approaches for Health Monitoring." *IEEE Access*, 2021. https://ieeexplore.ieee.org/document/9383694