# Homepage Welcome Screen



**1.**

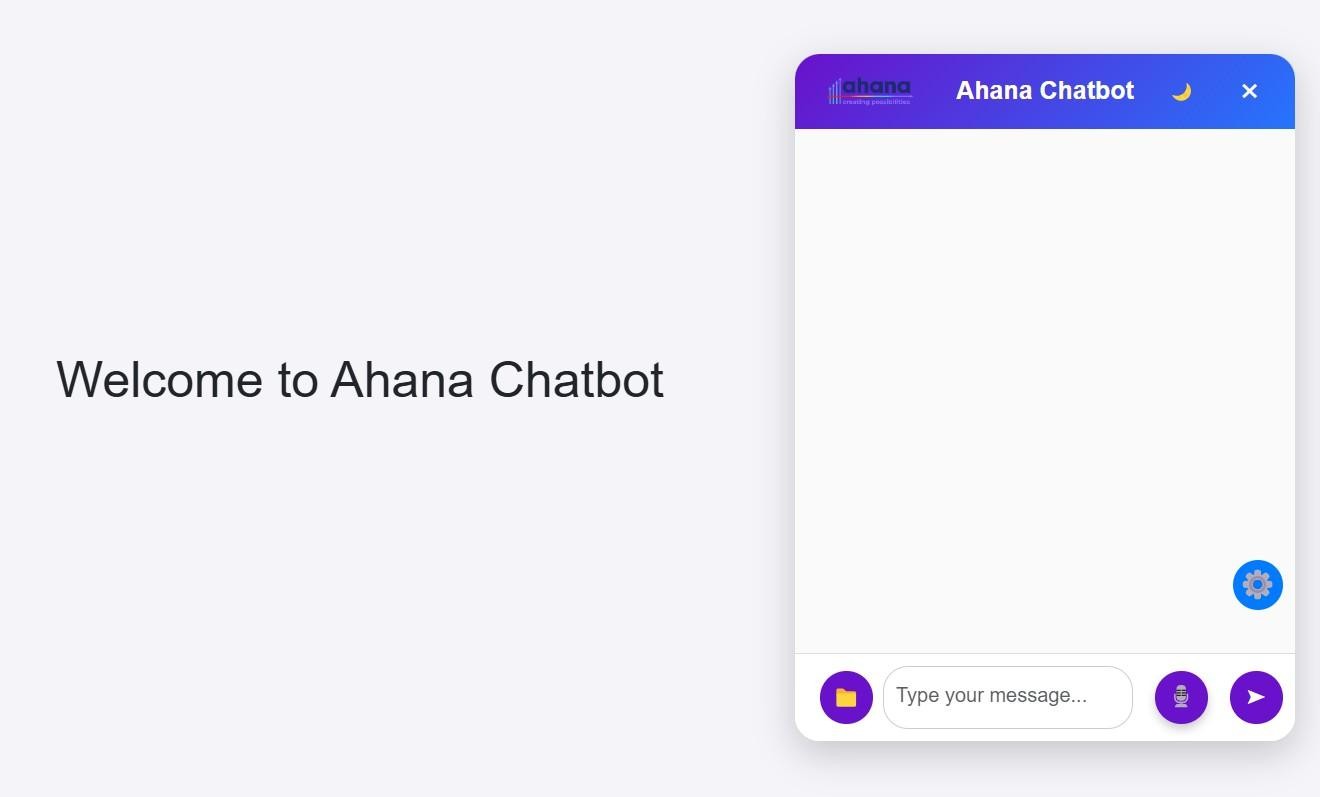
****

Fig 1 : Homepage Welcome Screen

# Chatbot Initial Interface:

* The welcome screen is the user's first interaction with the Ahana Chatbot.
* It features a clean and simple layout with a greeting message.
* A text input box is provided for user interaction.
* Users can type queries, upload files, or extract scraped website content.
* It encourages engagement and sets the tone for an interactive experience.

# Dynamic Question and Answering Flow

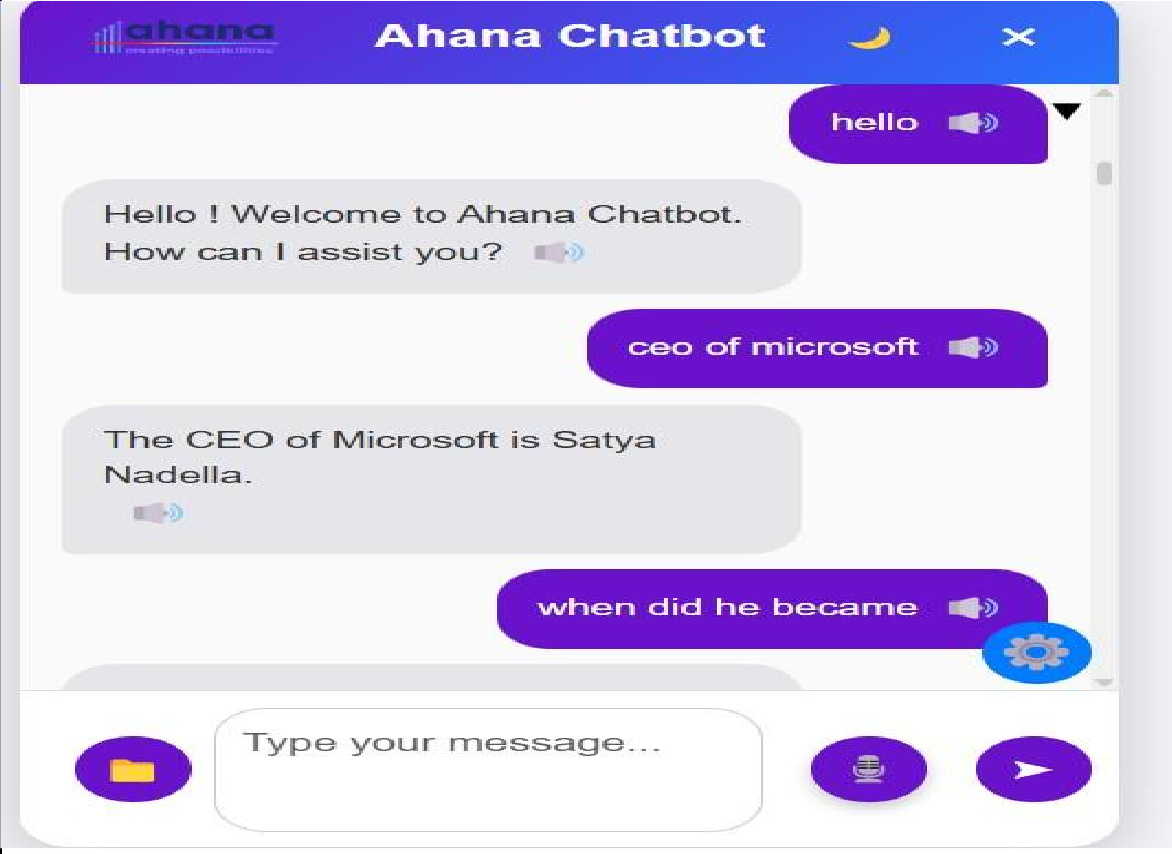
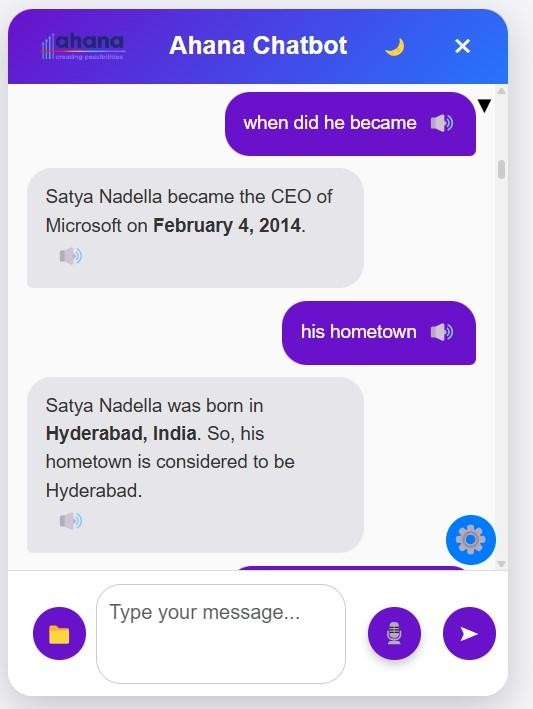
.

Fig 2 : Dynamic Question and Answering Flow

* + Demonstrates the chatbot’s ability to answer factual questions.
  + Uses Gemini 2.0 Flash’s generative AI.
  + Answers questions like Satya Nadella’s CEO appointment date or hometown.
  + Retrieves and formats accurate information.
  + Displays key details (e.g., dates, locations) in bold for clarity.

# Dynamic Question and Answering Flow

****

F ig 3 : Dynamic Question and Answering Flow

* + Demonstrates the chatbot’s ability to answer factual questions using Gemini 2.0 Flash’s generative AI.
  + The chatbot maintains a conversation history, which allows it to recall previous questions in sequence.
  + This history is stored in a session-based storage or memory structure, capturing each interaction with timestamps, user queries, and bot responses to ensure continuity throughout the conversation.
  + By tracking context, the chatbot can dynamically reference earlier exchanges to refine its follow-up answers.
  + The conversation history is stored either temporarily within the session for short-term recall or persistently in a database for long-term tracking of user interactions.
  + When interacting with Gemini 2.0 Flash, past exchanges are organized into structured prompts, enabling the AI model to generate context-aware responses and facilitate smooth, ongoing dialogue.

# File Content Extraction: Summary

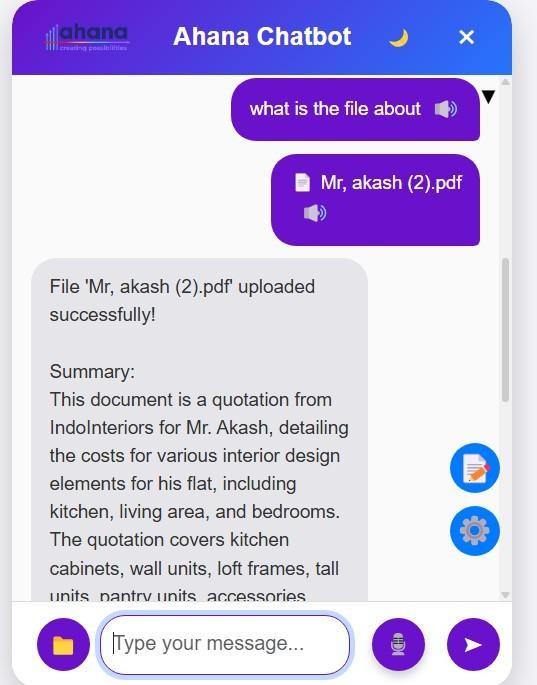
****

Fig 4: File Content Extraction: Summary

* + When a user uploads a file, the system first validates its extension, allowing only supported types like PDF, DOCX, TXT, CSV, XLSX, JPG, and PNG.
  + Once validated, the file is saved to the server.
  + Based on the file type, content extraction is performed:
    - PDFs are read using **PdfReader**; if no text is found, OCR is applied using **pytesseract** after converting pages to images.
    - DOCX files are processed with the **python-docx** library, extracting text from paragraphs, tables, headers, and footers.
    - TXT files are read directly as plain text.
    - CSV and Excel files are loaded using **pandas** and converted to strings.
    - Image files are processed using **PIL** and **pytesseract** for OCR text extraction.
  + The extracted text is stored in the session.
  + The **Gemini model** is prompted to summarize the content.
  + This summary is returned to the user and saved for future use, enabling accurate responses to questions or comparisons involving the uploaded file.

# File Content Extraction: Question And Answer Functionality

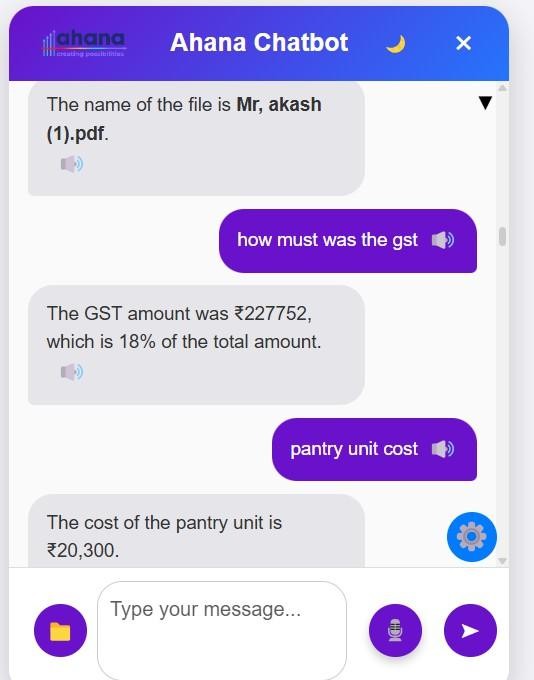
****

Fig 5: File Content Extraction: Question And Answer Functionality

* Once a file is uploaded and its text content is extracted, it is stored in the session.
* The system enables question-and-answer interactions based on the uploaded file.
* When a user asks a question related to a specific file, the app retrieves the extracted text using the filename as a key.
* Gemini processes the input contextually and returns a relevant and coherent answer.
* This allows users to ask specific or general questions about any uploaded document and receive accurate, file-based responses in real time.

# Comparison of 2 files

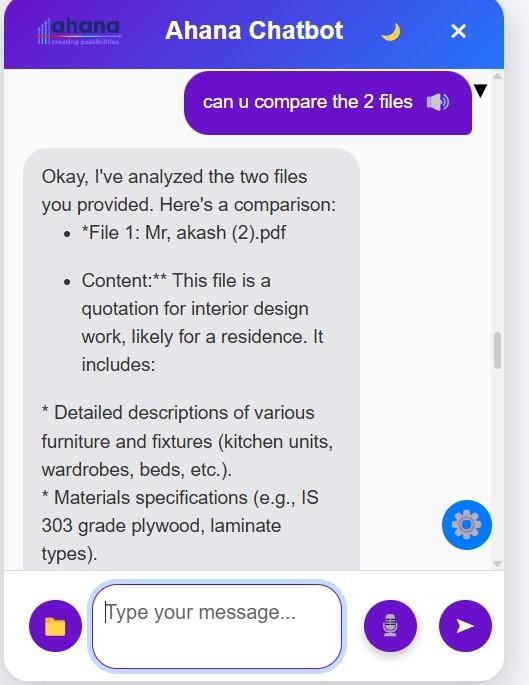
****

Fig 6.1 : Comparison of 2 files

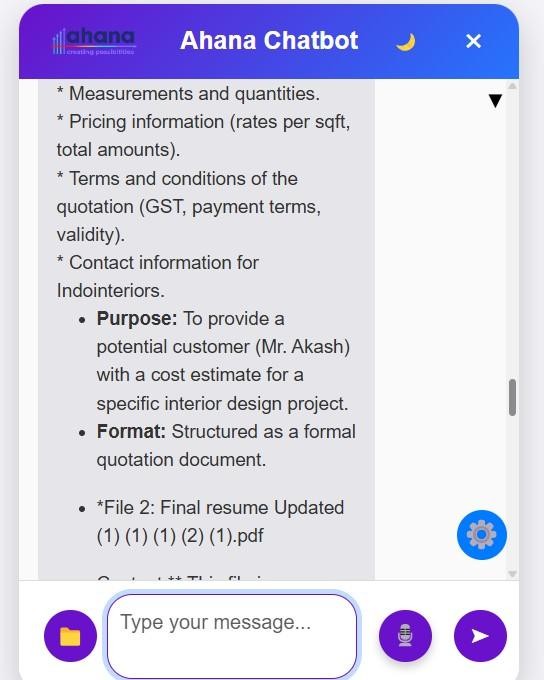


Fig 6.2 : Comparison of 2 files

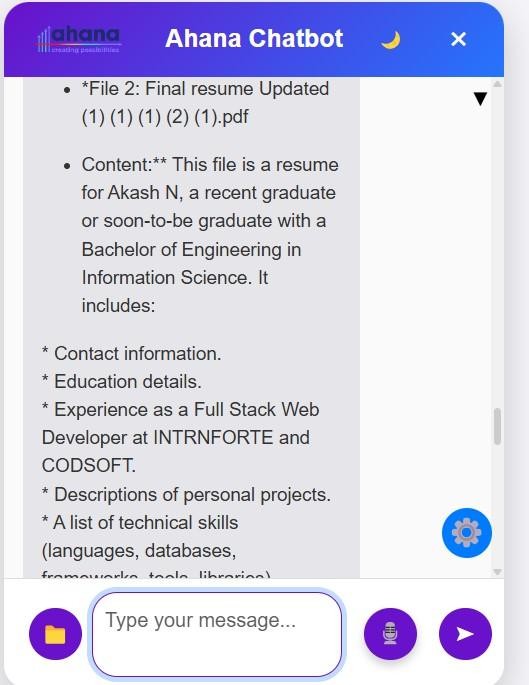


Fig 6.3 : Comparison of 2 files

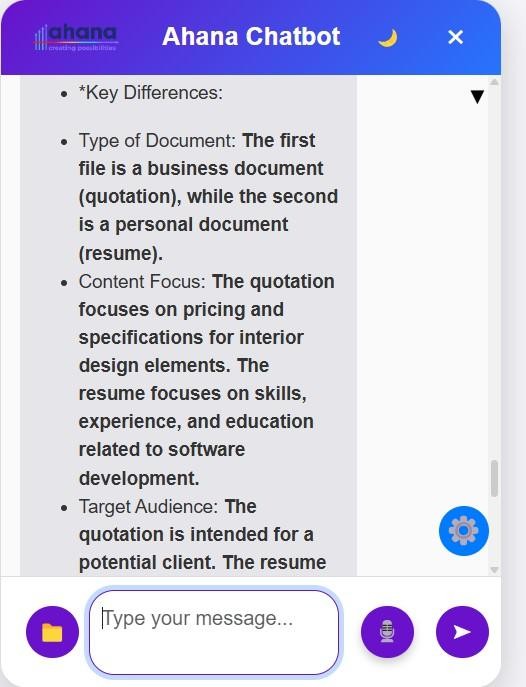


Fig 6.4 : Comparison of 2 files

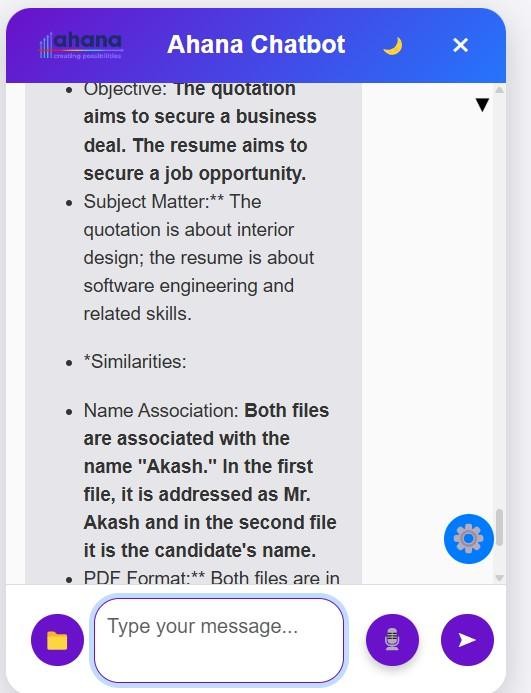


Fig 6.5 : Comparison of 2 files

* When a user asks for a comparison between two files, the backend checks if the names of two previously uploaded files are mentioned in the query.
* If both filenames are found, the system retrieves the corresponding extracted texts from the session data.
* A well-formatted prompt is constructed, including both documents’ text and comparison instructions focusing on objectives, structure, content, language, and overall similarities.
* This prompt is sent to the Gemini model using chat.send\_message(), and the model returns a detailed, human-readable comparison.
* The comparison result is displayed to the user and optionally stored in the chat history for future reference, enabling document analysis within the same session.

# Chatbot Code Retrieval Functionality

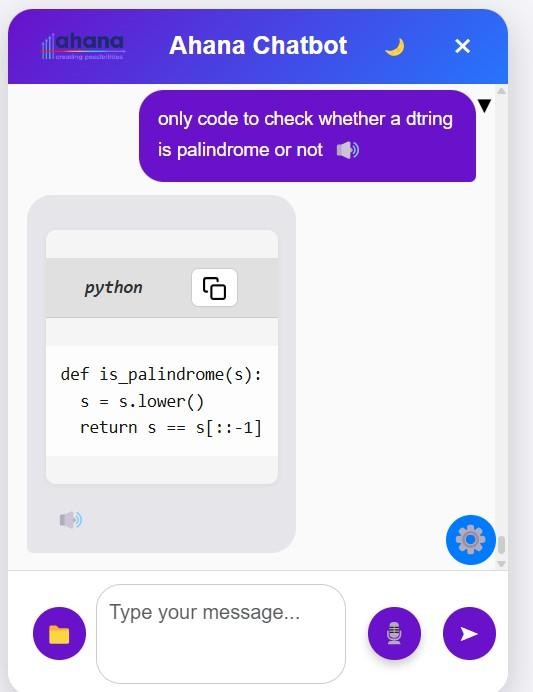
****

Fig 7 :Chatbot Code Retrieval Functionality

* Demonstrates the Ahana Chatbot’s ability to intelligently respond to a user’s query.
* When asked for a Python function to check whether a string is a palindrome, the chatbot provides a relevant and concise solution.
* The solution uses string manipulation and slicing for accuracy.
* This showcases the chatbot's proficiency in understanding programming- related questions.
* The chatbot delivers accurate, minimal code responses instantly.

# : Saving Retrieved Code in the Integrated Notepad

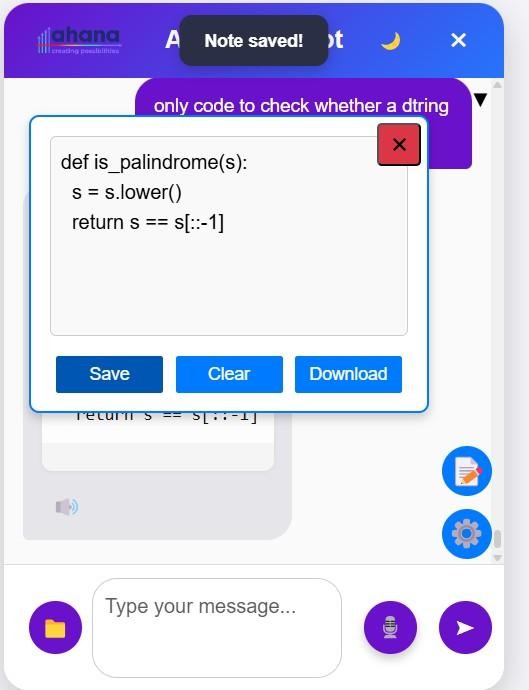
****

Fig 7.1: Saving Retrieved Code in the Integrated Notepad

* + - The chatbot allows users to save retrieved responses, such as code snippets, directly into an integrated notepad interface.
    - The notepad includes options like **Save**, **Clear**, and **Download** for managing and preserving notes.
    - A "Note saved!" confirmation message
    - is shown, indicating the code has been successfully stored within the session.
    - The saved code can be used for future reference or exported.
    - Users have full control over managing their notes within the notepad interface.

# : Download Functionality of Notes.

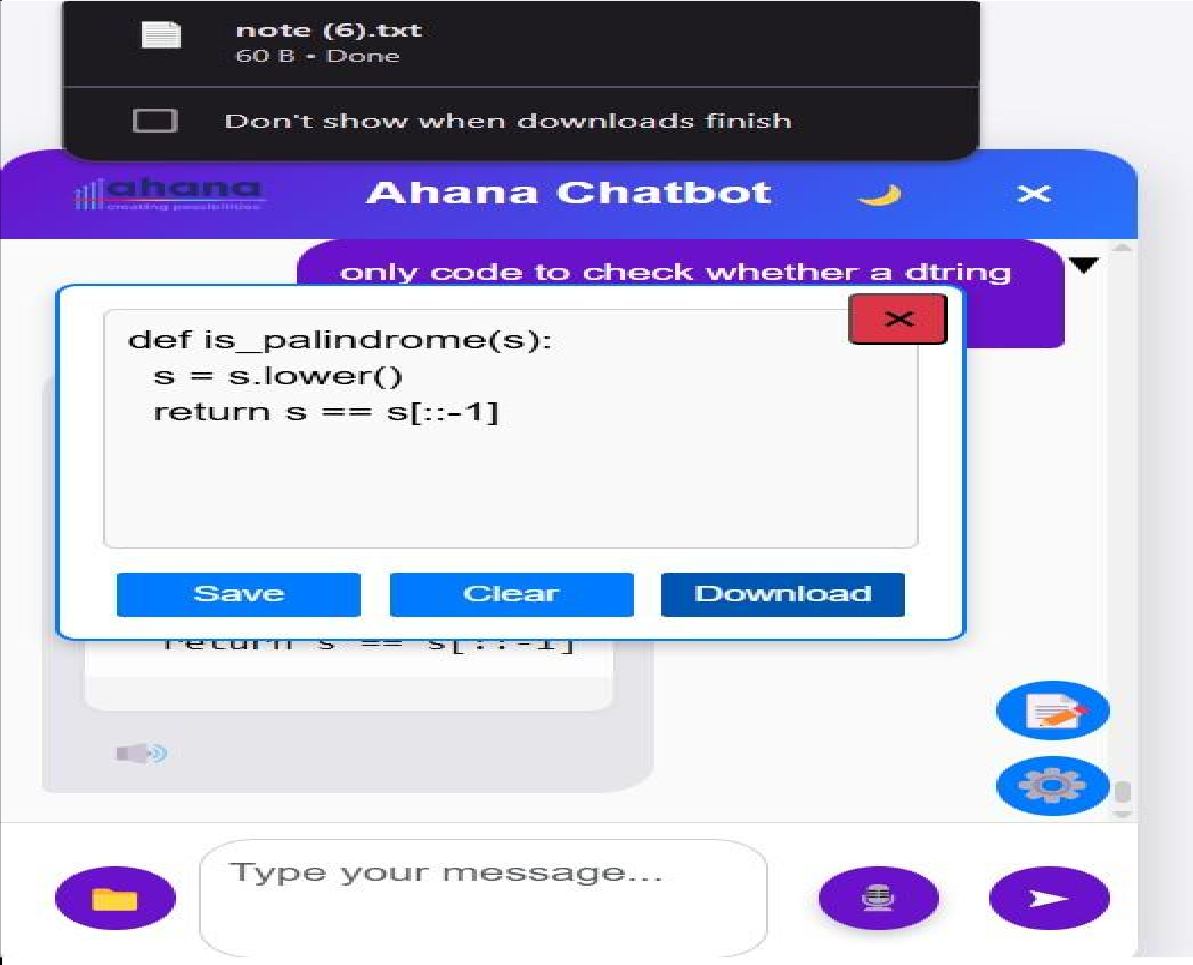
****

Fig 7.2: Download Functionality of Notes.

* + - Users can download saved code as a .txt file from the chatbot’s notepad interface.
    - Upon clicking the **Download** button, a confirmation message appears at the top of the screen.
    - The message indicates that the file (e.g., **note (6).txt**) has been successfully downloaded.
    - This feature allows users to conveniently export chatbot-generated content.
    - It provides an easy way for users to use the content outside the application environment.

# Clearing Notes from the Notepad Interfacejm

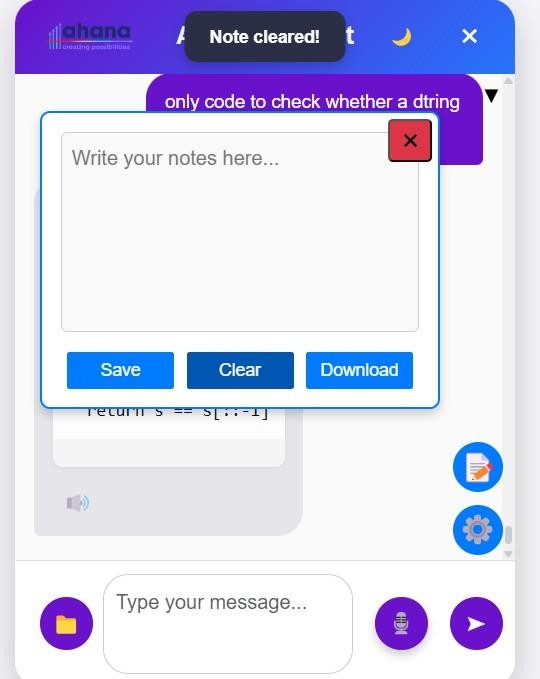
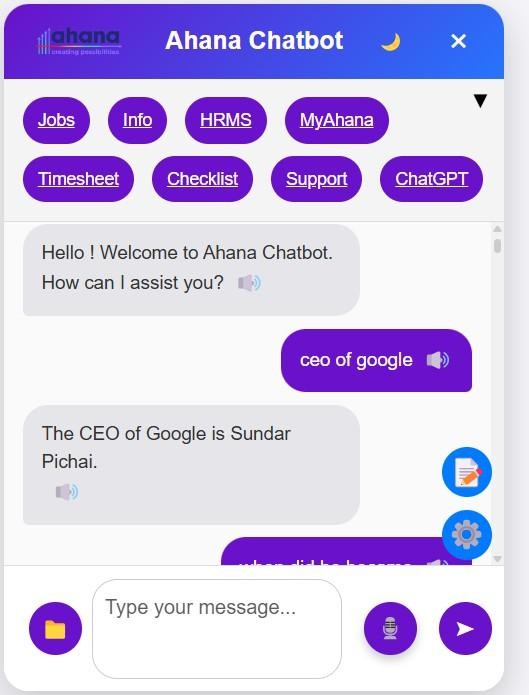


Fig 8 : Clearing Notes from the Notepad Interface.

* The notepad interface appears cleared with a notification stating **“Note cleared!”**
* This reflects the user’s ability to reset or remove saved content with a single action.
* It ensures users can maintain a clean workspace, edit existing content, or start fresh.
* The clear functionality enhances flexibility and usability of the chatbot’s built-in note-taking system.
* Users can easily manage their notes and control the content within the interface.

# Smart Navigation System

****

**Fig 9 :Smart Navigation System**

* The top panel of the chatbot features quick-access navigation buttons.
* These buttons link to key sections of the Ahana platform, including **Jobs**, **Info**, **HRMS**, **MyAhana**, **Timesheet**, **Checklist**, **Support**, and **ChatGPT**.
* The navigation buttons allow users to effortlessly move between important areas without leaving the chatbot.
* This improves efficiency and enhances the overall user experience.
* Users can quickly access and navigate essential features directly from the chatbot interface.

# Dark Mode

Fig 10 : Dark Mode Toggle

* The chatbot includes a **Dark Mode** option for enhanced visual comfort.
* Users can toggle between light and dark themes based on preference.
* Dark Mode reduces eye strain, especially during low-light usage.
* The interface elements, including notepad, chat area, and navigation panel, adapt seamlessly to the dark theme.
* This feature enhances accessibility and improves the overall user experience.

# Voice Interaction Support

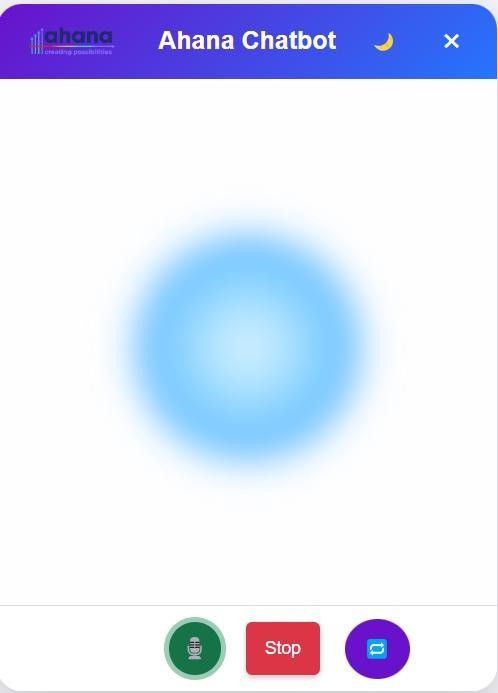
****

Fig 11:Voice Interaction Support

* The Ahana Chatbot includes voice functionality with **play**, **stop**, and **reset**

controls.

* When a user asks a question, the chatbot can **speak the answer aloud sequentially**.
* Users can **stop the voice response** at any point if it is too long.
* The **reset** option allows restarting the conversation from the beginning.
* This feature makes interactions more **dynamic and accessible** for users.