

DAY – 15

19 August 2025

PDF manual

The PISMS (Punjab Irrigation Support & Management System) Chatbot is developed to serve a diverse group of users who differ significantly in terms of educational background, technical knowledge, professional roles, and familiarity with digital platforms. Since the system is intended for statewide use, including both rural and urban users, special attention is given to simplicity, clarity, and accessibility. The chatbot interface and responses are designed in such a way that even first-time users can interact with it comfortably without feeling confused or dependent on external help.

The primary goal of defining user characteristics is to understand user expectations and limitations so that the chatbot can deliver accurate, relevant, and easy-to-understand information. The system avoids technical complexity and focuses on practical guidance, ensuring that users can complete their tasks efficiently through conversational interaction.

Types of Users

The users of the PISMS Chatbot can broadly be classified into three main categories:

1. Farmers

Farmers form the largest and most important user group of the PISMS chatbot system. Most of these users belong to rural areas and may have limited exposure to advanced digital technologies.

- Many farmers have basic or low digital literacy and may not be comfortable with complex mobile applications or web portals.
- They often rely on smartphones with limited internet connectivity and basic device configurations.
- Farmers usually seek information related to irrigation services, application procedures, registration steps, document requirements, and application status tracking.
- They expect the chatbot to guide them step by step rather than providing lengthy or technical explanations.

- Simple language, short sentences, and clear instructions are essential for this user group.
- Visual clarity and logically ordered responses help farmers understand the process better.
- Bilingual support in Punjabi and English is a critical requirement, as many farmers prefer Punjabi for better understanding and comfort.

The chatbot is designed to act as a digital assistant for farmers, reducing their dependency on physical office visits, phone calls, or intermediaries. By providing clear guidance in a conversational manner, the system empowers farmers to independently use the PISMS platform.

2. Officials (Patwari / SDO / JE / XEN / SE, etc.)

Government officials involved in irrigation management represent the second major user group. These users interact with the PISMS system as part of their daily administrative and field-related duties.

- Officials generally possess moderate technical knowledge and are familiar with government software systems.
- They require quick access to information related to specific modules such as Chakbandi, CO40, Warabandi, and other irrigation-related forms.
- Their primary expectation is accuracy and speed, as delays can affect official workflows and service delivery.
- Officials prefer task-oriented guidance that helps them complete processes efficiently without unnecessary explanations.
- The chatbot must provide precise instructions, document requirements, and process clarifications relevant to their roles.
- Time-saving and error reduction are key expectations from this user group.

For officials, the chatbot functions as a support tool that minimizes confusion, reduces dependency on manual help desks, and ensures uniform understanding of procedures across departments.

3. Admin / Super Admin

Admin and Super Admin users are responsible for managing and maintaining the PISMS platform and its operational structure.

- These users have high technical expertise and deep knowledge of system workflows and configurations.
- They are responsible for authority mappings, user role management, system monitoring, and overall platform governance.
- The chatbot is used as a quick-reference support system to address repeated or common queries.
- Admin users expect concise and direct responses that help reduce manual intervention and repetitive support tasks.
- Reliability and consistency of information are critical for this group.

For administrators, the chatbot improves operational efficiency by reducing the workload on technical teams and support staff.

Key Needs of Users

Across all user categories, certain common needs are identified that guide the chatbot design:

- Clear and simplified guidance without technical jargon
- Easy-to-understand conversational interaction
- Bilingual support in English and Punjabi for wider accessibility
- Availability of assistance at any time (24x7)
- Quick and accurate responses to reduce task completion time
- Reduced dependency on physical visits and human support

Actor	Role
User (Farmer / Official)	Interacts with chatbot
Chatbot System	Receives query, processes response & returns output
FAQ Dataset	Provides predefined Q/A
PDF Knowledge Base	Provides additional Q/A extracted from PRSC.pdf
ML Model (TF-IDF / Classifier)	Matches user question with dataset
Translation Service	Translates messages English ↔ Punjabi
Flask Server	Handles API request + Session Chat History

- User → Ask Question → Chatbot System
- Chatbot System → Check Conversation History
- Chatbot System → Determine Language
- Chatbot System → Match Q/A via ML Model
- Chatbot System → If not found → Search PDF Knowledge Base
- Chatbot System → Format Answer
- Chatbot System → Show Answer to User
- Chatbot System → Store Chat History in Session

Information of languages to used-

Along with the four major programming languages (HTML, CSS, JavaScript, and Python), this project also depends on several **supporting languages, libraries, frameworks and tools**. These technologies improve the performance, efficiency, usability and deployment of the chatbot. Supporting tools do not replace the main languages, but work together with them to enhance the overall system.

1. Flask Framework (Python-Based)

Flask is a lightweight web framework for Python that allows building backend servers easily.

In this project, Flask connects the chatbot interface with the backend logic. It helps in:

- Creating routes to receive and send messages

- Connecting the front-end (browser) with the Python logic
- Maintaining user session
- Responding to the user without reloading the page

Flask makes backend development fast, secure, and modular, which is ideal for chatbot projects.

2. PDF Reader Library (PyPDF2 or Similar)

A chatbot needs access to knowledge sources for providing answers. In this project, a PDF extraction library is used to read and extract text from the PRSC PDF file.

It allows the chatbot to:

- Scan the PDF
- Identify important topics
- Search for answers based on the user's question

This library helps the chatbot behave like a knowledge base system.

3. Dataset File (dataset.txt)

The dataset file is used as a training or lookup database for simple responses. It contains frequently asked questions and their answers in English and Punjabi. The chatbot first checks the dataset for a quick match and then searches the PDF if needed.

This improves:

- Response accuracy
- Speed of answer generation
- Language support

4. AJAX / Fetch API (JavaScript Technique)

The project uses the Fetch API through JavaScript to communicate with the backend without refreshing the page. It helps in:

- Sending user message to Python server
- Receiving chatbot reply instantly

- Displaying the message in chat area in real time

This technique gives the chatbot a smooth and seamless user experience.

5. Web Browser Developer Tools

Modern browsers (Chrome / Edge / Firefox) include developer tools for testing and debugging.

During development, they were used to:

- Inspect HTML/CSS elements
- Debug JavaScript responses
- Monitor network requests and API calls

These tools ensured that the chatbot runs error-free and fast.