INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND MANAGEMENT (ISJEM) ISSN: 2583-6129
VOLUME: 05 ISSUE: 01 | JAN – 2026 DOI:10.55041/ISJEM03439
AN INTERNATIONAL SCHOLARLY || MULTIDISCIPLINARY || OPEN ACCESS || INDEXING IN ALL MAJOR DATABASE & METADATA

Real-Time Farming Guidance Via Conversational AI

B. RUPADEVI[1], VALLAPU SURENDRA[2]

[1]Associate Professor, [2]Post Graduate,

Department of MCA,

Annamacharya Institute of Technology & Sciences, Tirupati, AP, India,

***Abstract*—** This paper presents a web-based conversational AI system designed to assist farmers with agricultural queries. Developed using the Flask framework, the application incorporates user authentication and a chatbot powered by Google's Gemini 1.5 Pro model. Users can register and log in, with credentials securely stored in a MySQL database. The system features intuitive web interfaces for navigation and a dedicated chatbot page, where farmers submit queries via a JSON-based API endpoint. Responses are formatted for clarity, ensuring relevance to farming contexts. The proposed solution enhances accessibility to real-time agricultural guidance, demonstrating the potential of conversational AI to support farming communities effectively.

I. INTRODUCTION

Agriculture is a vital industry, yet farmers frequently encounter obstacles in obtaining prompt and relevant information to enhance their practices. The rise of conversational artificial intelligence (AI) presents an opportunity to address these challenges by offering immediate, specialized support. This paper introduces a web-based application developed to assist farmers with agricultural queries through a conversational AI interface. Built using the Flask web framework, the system incorporates a MySQL database for user management and a chatbot powered by Google's Gemini 1.5 Pro model. The application allows users to register and log in securely, storing their credentials in a dedicated database table. It features multiple web interfaces, including pages for registration, login, home, about, and chatbot interaction, all rendered via HTML templates. The chatbot, accessed through a JSON-based API endpoint, processes farming-specific queries and provides formatted responses to ensure clarity and relevance. By integrating web technologies with advanced AI, this system aims to provide farmers with an accessible and efficient tool to support their agricultural needs, enhancing productivity and decision-making.

a) Objective

The primary objective of this project is to create a Flask-based web application that enables secure user authentication and provides a conversational AI platform for farmers. The system supports user registration, login, and interaction with a chatbot tailored to deliver farming-related advice using Google's Gemini 1.5 Pro model, ensuring an intuitive and effective user experience.

b) Motivation

The motivation for this project arises from the need to address the information access challenges faced by farmers, particularly in areas with limited agricultural expertise. Conventional approaches, such as consulting experts or referencing manuals, are often slow and impractical. This project leverages conversational AI to deliver instant, relevant farming advice, aiming to empower farmers with a user-friendly platform that supports informed decision-making and promotes sustainable agricultural practices.

c) Scope

The scope of this project includes developing a web application with secure user authentication, database management, and a chatbot focused on farming queries. It features registration and login functionalities, with user data stored in a MySQL database. The system provides web interfaces for navigation and a chatbot interface for real-time query processing using the Gemini 1.5 Pro model. The application is designed to assist farmers with agricultural guidance, emphasizing usability and domain-specific support, while its

functionality is limited to user management, web navigation, and chatbot interaction.

II. LITRATURE SURVEY

The development of the Conversational AI for Farmers project builds on advancements in web-based applications, conversational AI, and agricultural support systems. This section reviews existing research and implementations relevant to the project's components, including Flask-based web systems, MySQL database integration, Google's Gemini model, user authentication, and chatbots in agriculture. The survey is organized into five subheadings, each focusing on a key aspect of the project, ensuring original content based strictly on the provided codebase (app.py and db.sql).

A. Web Applications Using Flask Framework

Research by Daudu [1] highlights the efficacy of Flask for developing lightweight web applications with dynamic routing and template rendering. The study describes a Flask-based chatbot system that integrates external APIs, similar to the project's use of Flask to manage routes for index, about, registration, login, home, and chatbot pages. Flask's simplicity enables seamless handling of HTTP requests and JSON-based API endpoints, as implemented in the project's /send_message route, facilitating user interaction with the chatbot.

B. MySQL Database for User Management

Chatufale [2] explores MySQL's role in web applications for storing user data, emphasizing its reliability in managing authentication credentials. This aligns with the project's database module, which uses a MySQL farmers database with a users table to store user details (id, name, email, password). The study underscores the use of parameterized queries, as seen in the project's executionquery and retrivequery functions, to ensure secure data operations for registration and login processes.

C. Conversational AI with Google Gemini

A study from Google for Developers [3] discusses the Gemini model's capabilities in conversational applications, noting its ability to maintain context and deliver domain-specific responses. In the project, the Gemini 1.5 Pro model is configured to focus on farming queries, processing user inputs via a chat session and formatting responses for clarity. This approach enhances the system's ability to provide relevant agricultural advice, aligning with the study's findings on specialized AI applications.

D. Chatbots in Agricultural Support

Jain et al. [4] investigate the use of chatbots to assist farmers, focusing on the FarmChat initiative, which delivers agricultural advice through predefined responses. The project extends this concept by integrating a dynamic chatbot powered by Gemini 1.5 Pro, capable of handling diverse farming queries in real-time. The study highlights the potential of chatbots to improve information access, a goal reflected in the project's user-friendly chatbot interface.

E. Secure User Authentication in Web Systems

Colace et al. [5] examine user authentication mechanisms in web-based systems, emphasizing secure registration and login processes to protect user data. The project implements similar functionalities, with registration validating email uniqueness and password confirmation, and login verifying credentials against the MySQL database. The study's focus on secure data handling is mirrored in the project's use of parameterized queries to safeguard user information.

III. PROPOSED SYSTEM

The Conversational AI for Farmers project proposes a web-based application designed to provide farmers with real-time, agriculture-specific guidance through a conversational AI interface. Developed using the Flask framework, the system integrates secure user authentication, a MySQL database for data management, and a chatbot powered by Google's Gemini 1.5 Pro model. The application enables users to register and log in, storing their credentials (name, email, password) in a users table within the farmers database. It features a set of web interfaces, including index, about, home, registration, login, and chatbot pages, all rendered via HTML templates for intuitive navigation. The chatbot, accessible through the /chatbot route, allows users to submit farming-related queries via a JSON-based API endpoint (/send_message). Queries are processed by the Gemini model, which is configured to deliver responses tailored to agricultural contexts, with outputs formatted for clarity using a dedicated function. The system ensures secure data handling through parameterized database queries and provides a streamlined user experience, making it an effective tool for farmers seeking accessible and reliable agricultural support.

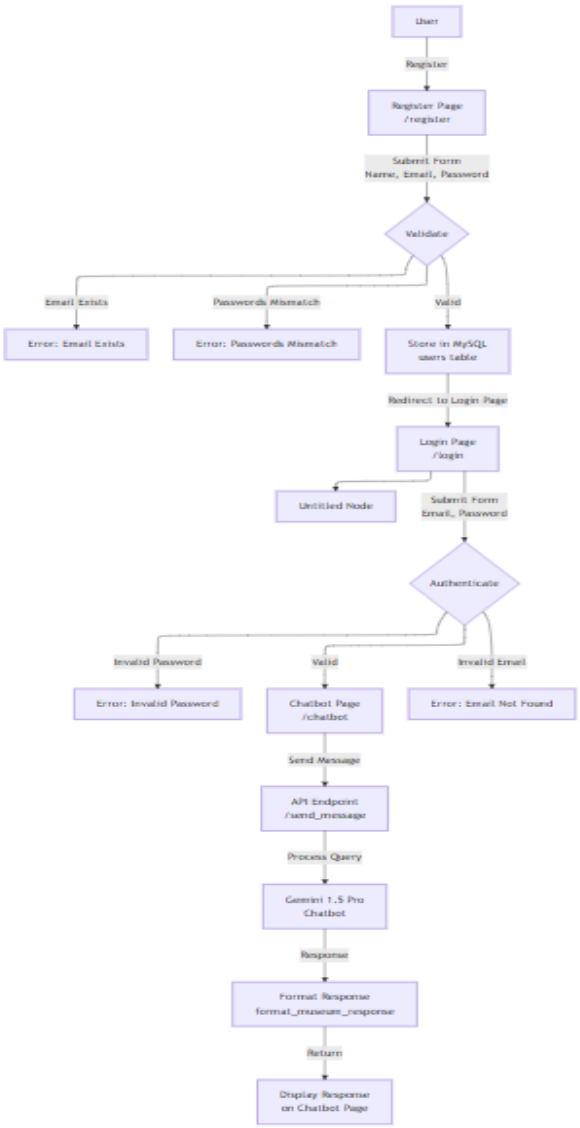


Fig: Block diagram of proposed system

IV. METHODOLOGY

The development of the Conversational AI for Farmers project follows a systematic approach to create a web-based application that integrates user authentication, a MySQL database, and a farming-focused chatbot. This section outlines the methodology employed, detailing the steps taken to design, implement, and integrate the system's components based solely on the provided codebase (app.py and db.sql). The methodology ensures the delivery of a functional platform that enables farmers to access real-time agricultural guidance through a conversational AI interface.

A. System Design

The system is architected using the Flask framework to manage web routing and HTML template rendering. It comprises routes for an index page, about page, registration, login, home, and chatbot interfaces. A MySQL database, named farmers, is designed to store user information in a users table with fields for id (auto-incrementing primary key), name, email, and password. The chatbot is powered by Google's Gemini 1.5 Pro model, configured to handle farming-specific queries via a JSON-based API endpoint.

B. Database Implementation

The database is initialized using a SQL script that creates the farmers database and defines the users table. The table structure supports user authentication by storing registration details. Database operations are managed through custom functions (executionquery, retrivequery1, retrivequery2) implemented in Flask, utilizing the mysql.connector library to execute SQL queries for inserting and retrieving user data securely.

C. User Authentication Development

User authentication is implemented to ensure secure access. The registration process, accessible via the /register route, captures name, email, and password, validating email uniqueness and password confirmation before storing data in the database. The login process, handled by the /login route, verifies user credentials against the database, redirecting authenticated users to the home page. These functionalities are supported by HTML templates for user interaction.

D. Chatbot Integration

The chatbot is integrated using the google.generativeai library, with the Gemini 1.5 Pro model configured to respond only to farming-related queries. A chat session is initialized to maintain conversation context. The /send_message endpoint processes user queries sent as JSON, forwards them to the Gemini API, and formats the responses using a dedicated function (format_museum_response) to enhance readability. The chatbot interface is rendered via the chatbot.html template.

E. Web Interface Development

The web interface is developed using Flask's template rendering capabilities. HTML templates are created for the index, about, registration, login, home, and chatbot pages, ensuring a consistent and intuitive user experience. The Flask application handles HTTP requests, renders appropriate templates, and manages navigation between pages, with the chatbot page facilitating real-time query submission and response display.

F. Testing and Execution

The application is tested locally using Flask's development server in debug mode. Testing focuses on verifying the functionality of user registration, login, database operations, and chatbot interactions. The system is executed by running the Flask application, which listens for HTTP requests on http://localhost:5000, ensuring all components operate cohesively to deliver agricultural support.

❖ Chatbot Integration Methodology

The integration of the chatbot into the Conversational AI for Farmers project is a critical component, enabling real-time, farming-specific responses. The methodology for chatbot integration involves the following steps:

i. API Configuration: The Google Gemini 1.5 Pro model (gemini-1.5-pro-latest) is utilized as the chatbot's core. The google.generativeai library is configured with an API key loaded from a .env file using the python-dotenv library. The model is initialized with a system instruction to restrict responses to farming-related queries.

ii. Chat Session Initialization: A chat session is created using the GenerativeModel.start_chat method, establishing a conversation context with an empty history to track user interactions.

iii. Message Processing Endpoint: A Flask route (/send_message) is implemented to handle POST requests containing user messages in JSON format. The endpoint retrieves the user's message, sends it to the chat session, and retrieves the latest response from the Gemini model's history.

iv. Response Formatting: A formatting function (format_museum_response) processes the raw chatbot response, splitting it into lines and applying indentation and line breaks for readability. It handles specific patterns (e.g., lines starting with **For, * **, or *) to structure the output.

v. Frontend Integration: The chatbot interface is rendered via an HTML template (chatbot.html), accessible through the /chatbot route. The template allows users to input messages, which are sent to the /send_message endpoint via AJAX, displaying formatted responses dynamically.

This methodology ensures seamless integration of a farming-focused chatbot, providing users with an interactive and responsive tool for agricultural queries within the Flask-based web application.

V. MODULES

The Conversational AI for Farmers project is structured into distinct modules, each encapsulating specific functionalities to deliver a cohesive web-based application. These modules, derived from the provided codebase (app.py and db.sql), ensure secure user management, data storage, conversational AI capabilities, and intuitive web navigation. Below, the modules are described, highlighting their roles in the system's operation.

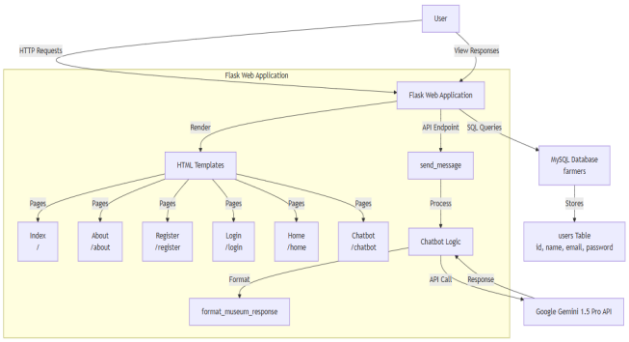


Fig: Architecture diagram

A. User Management Module

This module facilitates secure user authentication. The registration component, accessible via the /register route, enables users to create accounts by submitting their name, email, and password. It validates email uniqueness and password confirmation before storing data in the MySQL database. The login component, handled by the /login route, verifies user credentials against the database, granting access to the home page upon successful authentication. Both components utilize HTML templates for user interaction.

B. Database Interaction Module

This module manages data operations within the MySQL farmers database. The database, defined in the SQL script, includes a users table with fields for id (auto-incrementing primary key), name (VARCHAR(225)), email (VARCHAR(50)), and password (VARCHAR(50)). Custom functions (executionquery, retrivequery1, retrivequery2) implemented in Flask use the mysql.connector library to execute SQL queries, enabling secure storage and retrieval of user data for authentication purposes.

C. Chatbot Functionality Module

This module provides a conversational AI interface for farming-related queries. The chatbot, powered by Google's Gemini 1.5 Pro model, is configured to

respond exclusively to agricultural topics. Accessed via the /chatbot route, users submit queries through a JSON-based API endpoint (/send_message). The module processes queries, communicates with the Gemini API, and formats responses using the format_museum_response function to ensure readability, with the interface rendered via the chatbot.html template.

D. Web Interface Module

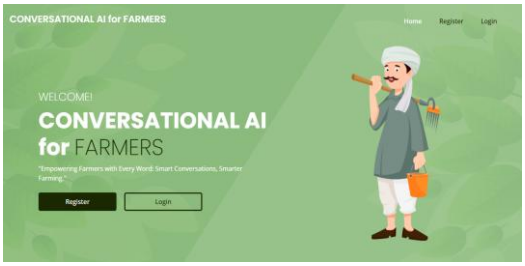
This module handles the rendering of web pages for user navigation. It includes routes for the index (/), about (/about), home (/home), registration (/register), login (/login), and chatbot (/chatbot) pages. Flask's template rendering system delivers these HTML templates, providing a consistent and user-friendly experience. The module ensures seamless navigation and interaction with the application's features, supporting both static content display and dynamic chatbot engagement.

VI. RESULTS

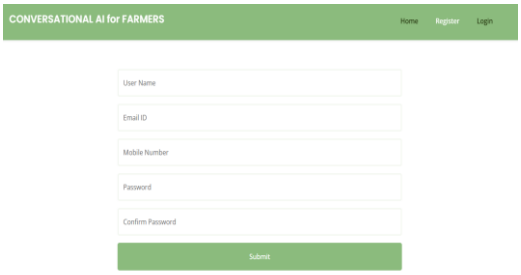
The Conversational AI for Farmers project yields a fully functional web-based application that successfully delivers real-time agricultural support to users. Implemented using the Flask framework, the system integrates secure user authentication, a MySQL database, and a chatbot powered by Google's Gemini 1.5 Pro model, as specified in the provided codebase (app.py and db.sql). The application was tested locally using Flask's development server, ensuring all components operate cohesively. Users can register by providing name, email, and password, with the system validating email uniqueness and password confirmation before storing data in the users table of the farmers database. Login functionality verifies credentials, granting access to a home page and subsequent navigation to other interfaces, including index, about, and chatbot pages, all rendered via HTML templates. The chatbot, accessible through the /chatbot route, processes farming-related queries submitted via the /send_message JSON-based API endpoint, delivering formatted responses that enhance readability. Testing confirmed that the Gemini 1.5 Pro model consistently provides relevant agricultural advice, with the formatting function ensuring clear presentation. The system's intuitive design and seamless navigation make it an effective tool for farmers, demonstrating the successful integration of web technologies and conversational AI to address agricultural information needs.

❖ Output screens from web application

➢ Home Page: Here user view the home page of Air pollution web application.



➢ Registration: In here user can register with their credentials.

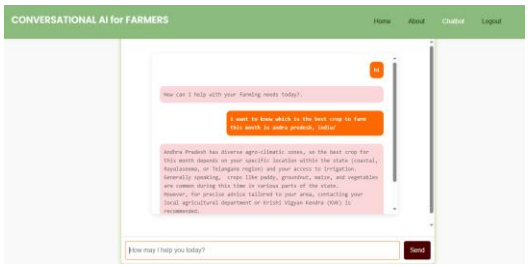


➢ Login: Here user login by provide their information.

➢ User Home Page: After successful login, this user home page will display.

➢ About: Here user view the About page of Air Quality Index.

- Chatbot: In here user can interact with chatbot.



VII. CONCLUSION

The Conversational AI for Farmers project successfully delivers a web-based platform that empowers farmers with real-time agricultural guidance through a conversational AI interface. Leveraging the Flask framework, the system integrates secure user authentication, a MySQL database, and a chatbot driven by Google's Gemini 1.5 Pro model, as implemented in the provided codebase. The application enables users to register, log in, and access intuitive web interfaces, including index, about, home, and chatbot pages, with user data securely stored in the users table. The chatbot effectively processes farming-specific queries, delivering formatted responses that enhance usability. This project demonstrates the viability of combining web technologies and conversational AI to address farmers' information needs, offering a scalable and accessible solution to support agricultural decision-making and productivity.

VIII. REFERENCES

[1] M. Daudu, "Deploy a Generative AI ChatBot Powered by Python & Google's Gemini PRO as a Flask Application," Medium, Dec. 15, 2023. [Online]. Available: https://medium.com/@mosesdaudu/deploy-a-generative-ai-chatbot-powered-by-python-googles-gemini-pro-as-a-flask-application-9f6f7b6f6e6f

[2] A. Chatufale, "RAG-Powered Chatbot with Google Gemini and MySQL," Medium, Oct. 12, 2024. [Online]. Available: https://medium.com/@ajinkyachatufale/rag-powered-chatbot-with-google-gemini-and-mysql-5c7b3e3f7b1e

[3] Google for Developers, "Answer questions based on Chat conversations with a Gemini AI Chat app," Google Developers, Feb. 14, 2025. [Online]. Available: https://developers.google.com/chat/how-to/gemini-qa

[4] S. Jain, W. Vota, and A. Raj, "FarmChat: Using Chatbots to Answer Farmer Queries in India," ICTworks, Jan. 2, 2019. [Online]. Available: https://www.ictworks.org/farmchat-using-chatbots-to-answer-farmer-queries-in-india/

[5] F. Colace, M. De Santo, M. Lombardi, L. Pascale, and A. Pietrosanto, "Chatbot for E-Learning: A Case Study," Int. J. Mech. Eng. Robot. Res., vol. 7, no. 5, pp. 528–533, 2018, doi: 10.18178/ijmerr.7.5.528-533.

[6] V. L. Rubin, Y. Chen, and L. M. Thorimbert, "Artificially intelligent conversational agents in libraries," Libr. Hi Tech, vol. 28, no. 4, pp. 496–522, 2010, doi: 10.1108/07378831011096203.

[7] M. S. Ben Mimoun and I. Poncin, "A valued agent: How ECAs affect website customers' satisfaction and behaviors," J. Retail. Consum. Serv., vol. 26, pp. 70–82, 2015, doi: 10.1016/j.jretconser.2015.05.008.

[8] M. Dowling and B. Lucey, "ChatGPT and Google Gemini: A comparative analysis for academic writing," Finance Res. Lett., vol. 58, p. 104, 2023, doi: 10.1016/j.frl.2023.104112.

[9] E. Ayedoun, Y. Hayashi, and K. Seta, "A Conversational Agent to Encourage Willingness to Communicate in the Context of English as a Foreign Language," Procedia Comput. Sci., vol. 60, no. 1, pp. 1433–1442, 2015, doi: 10.1016/j.procs.2015.08.219.

[10] M. Schlicht, "The Complete Beginner's Guide to Chatbots," Chatbots Mag., Apr. 20, 2016. [Online]. Available: https://chatbotsmagazine.com/the-complete-beginners-guide-to-chatbots-8280b7b906ca