Soul Support : AI Driven Emotional Assistance ChatBot

[1]**M.Parimala**
Associate Professor, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: pari.parillu@gmail.com

[2]**Chila Harshitha**
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: chilaharshitha@gmail.com

[3]**Bonala Shreya Yadav**
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: shreyabonala@gmail.com

[4]**T.Sai Kalyani**
UG Student, Dept. Computer Science and Engineering
Vignan's Institute of Management and Technology for Women, Hyd.
Email: kalyanithota@gmail.com

***Abstract*—** This paper introduces the development of an AI-based Mental Health Therapist Chatbot designed to provide immediate, accessible emotional support and mental health information. Utilizing deep learning and natural language processing (NLP), the chatbot identifies user intents and delivers appropriate responses across various mental health topics, including anxiety, stress, and sadness. The system is deployed via a web interface that supports both text and voice input, enhancing user accessibility. On the backend, the chatbot uses a Flask server, a trained Keras model for intent classification, and preprocessing pipelines built with NLTK and SpaCy. Experimental evaluations show high accuracy in intent recognition and efficient real-time interaction. Additionally, user feedback reflects a positive reception in terms of helpfulness and ease of use. While the chatbot does not replace professional care, it acts as a supportive digital tool for initial engagement and emotional guidance. Future enhancements will focus on multilingual support, contextual dialogue, and integration with mental health services.

I. INTRODUCTION

Mental health plays a crucial role in an individual's overall well-being and quality of life. In recent years, mental health issues such as depression, anxiety, stress, and emotional exhaustion have seen a significant rise worldwide. Despite this growing concern, access to timely, affordable, and judgment-free mental health support remains limited due to stigma, shortage of professionals, and geographic or economic barriers. These challenges create a gap that emerging technologies, particularly artificial intelligence (AI), are well-positioned to help address. AI-driven conversational agents, commonly known as chatbots—have shown promise in healthcare settings by offering support, information, and companionship through natural language interfaces. In the context of mental health, a chatbot can serve as a non-judgmental, always-available digital companion that encourages users to express emotions, learn coping strategies, and access reliable mental health information. It offers a safe space for individuals who may be hesitant to seek help from a professional or who face obstacles in accessing traditional services. This paper presents the design and development of a Mental Health Therapist Chatbot that uses deep learning and natural language processing (NLP) techniques to engage users in meaningful dialogue. The chatbot supports real-time interaction through both text and voice input, providing educational guidance and crisis-related information. By leveraging natural language processing techniques and a trained deep learning model, the chatbot is capable of recognizing a variety of user intents associated with emotional states, informational queries, and urgent support needs. Its conversational responses are drawn from a curated dataset designed to ensure empathy, clarity, and psychological safety. The system also incorporates voice input functionality using web-based speech recognition, allowing for hands-free interaction, which further improves accessibility, especially for users with disabilities or those in high-stress situations.

It is a support tool for users hesitant to approach professional help or those requiring immediate but non-clinical guidance. The use of a web-based deployment model ensures that the system is platform-independent and accessible across devices without requiring installation. Overall, the system aligns with ongoing efforts to expand the reach of digital mental health tools and reduce the barriers to early intervention and self-care. As mental health concerns continue to rise, particularly in the wake of global crises and increased social isolation, there is a growing need for scalable tools that can provide timely, personalized support. AI-powered chatbots offer a unique advantage in this domain by enabling continuous, non-judgmental conversations with users, which can help alleviate feelings of loneliness, stress, or confusion. These systems also empower individuals to take early action by encouraging self-reflection and providing coping strategies grounded in psychological best practices. Moreover, advancements in natural language understanding have made it possible to create chatbots that are not only responsive but also contextually aware. Although the current implementation focuses on single-turn interactions, future iterations can incorporate memory modules to facilitate more coherent, multi-turn conversations. This enhancement would allow the system to track a user's emotional journey over time and offer progressively tailored support. By democratizing access to mental health resources and reducing the barriers of cost, availability, and stigma, such AI-based systems represent an important step toward inclusive well-being. Their potential impact is especially significant for underserved populations, where traditional mental health infrastructure is limited or absent. Thus, this research contributes to the ongoing evolution of digital mental health tools and highlights the value of AI as a complementary force in promoting mental wellness at scale.

II. LITERATURE SURVEY

In recent years, the integration of artificial intelligence in mental health support has gained significant attention. Inkster et al. [1] evaluated Wysa, an empathy-driven conversational agent, and demonstrated its potential to provide meaningful digital mental health support through AI-driven dialogues. Their real-world study confirmed that users found Wysa helpful in managing emotional challenges, marking a key step in validating chatbot-based therapy tools. Sharma and Kaushik [2] provided a comprehensive survey on the state of chatbot applications in mental health, highlighting the progress of AI in this domain along with challenges such as personalization, emotional understanding, and ethical use. Their work emphasizes the importance of enhancing human-computer interaction to create more responsive and sensitive chatbot experiences. Kaur and Kumar [3] proposed a deep learning-based chatbot architecture focused on mental health, using natural language processing (NLP) to classify user inputs and respond appropriately. Their model demonstrated encouraging results in identifying mental states like stress or depression, thus showcasing the practical utility of machine learning techniques in emotion recognition.Singh and Joshi [4] explored the architectural design and implementation of conversational agents for psychological support. They discussed system frameworks capable of maintaining user engagement, and stressed the need for robust intent detection and context management to improve the therapeutic impact of chatbots. Kumar and Vashisht

[5] analyzed broader applications and ethical concerns of AI in mental health. Their findings suggested that while AI offers scalable solutions to increase access to care, developers must consider data privacy, user consent, and system transparency to ensure responsible deployment. Collectively, these studies lay a strong foundation for developing advanced, ethical, and emotionally intelligent chatbots. The present work builds on these insights by incorporating real-time interaction, voice input, and a modular architecture to enhance both accessibility and user experience.

III.METHODOLOGY

The development of the Mental Health Therapist Chatbot follows a modular approach, combining machine learning, natural language processing (NLP), and frontend interface engineering to build a responsive, empathetic conversational agent. The methodology is divided into the following key stages:

1).Data Preparation and Intent Structuring

The foundation of the chatbot is a curated dataset defined in an `intents.json` file. This file categorizes various user intents—such as greetings, anxiety, stress, and crisis support—along with multiple example user expressions and matching responses. The structured intent dataset is essential for both training the model and generating real-time responses.

2).Text Preprocessing and Feature Engineering

To prepare the dataset for machine learning, user inputs and sample phrases are preprocessed using NLP libraries such as NLTK and SpaCy**.** Preprocessing steps include:

Tokenization: Splitting sentences into individual words or tokens.

Lemmatization: Reducing words to their base or dictionary form.

Stopword Removal: Filtering out commonly used, non-informative words. The cleaned text data is then transformed into numerical representations using a bag-of-words model followed **by** one-hot encoding**.** These vectors are stored along with their corresponding labels (`texts.pkl`, `labels.pkl`) for use during both training and inference.

3). Model Training with Deep Learning

A feedforward neural network is constructed using Keras with a TensorFlow backend**.** The architecture consists of An input layer matching the dimensionality of the feature vectors. Multiple **dense** hidden layers activated using ReLU. A softmax output layer to predict intent categories. The model is trained using **categorical cross-entropy** as the loss function and the Adam optimizer for fast and efficient learning. Once trained, the model is saved in the form of `model.h5`.

4). Backend Development and Inference Pipeline

The backend is developed using the Flask web framework. Upon server startup, it loads the trained model and preprocessing components. When a user sends a message, the input undergoes the same preprocessing pipeline and is passed into the model for real-time intent prediction**.** The predicted intent is then used to retrieve a matching response from the `intents.json` dataset. The entire conversation is managed through a RESTful API endpoint (`/get`), which returns the response in JSON format**.**

5).Frontend User Interface and Interaction

The frontend is built using HTML, CSS, and JavaScript (jQuery) to create a responsive and intuitive chat interface. Key features include:

a) A collapsible chat window for better user control. Text and voice input support, where voice is captured via the Web Speech API.

b) Dynamic message display with timestamps and avatar icons.

c) Smooth scrolling and input field clearing for usability.

6).Modularity and Future Enhancements

The system is designed with modular architecture, allowing each component—model, preprocessing, interface, and dataset—to be independently updated or extended. Planned enhancements include: Multilingual support by augmenting the training dataset with translated patterns.

a) Sentiment analysis for better emotional understanding.

b) Contextual conversation tracking for multi-turn dialogue.

c) Secure referral integration for directing users to licensed professionals during critical moments.

7) .Privacy and Ethical Considerations

User data protection is a critical part of the system. The chatbot does not store personal conversations and follows privacy-conscious design principles. Ethical use guidelines are incorporated to ensure the chatbot remains a supportive tool, not a replacement for clinical therapy.

IV. SYSTEM ARCHITECTURE

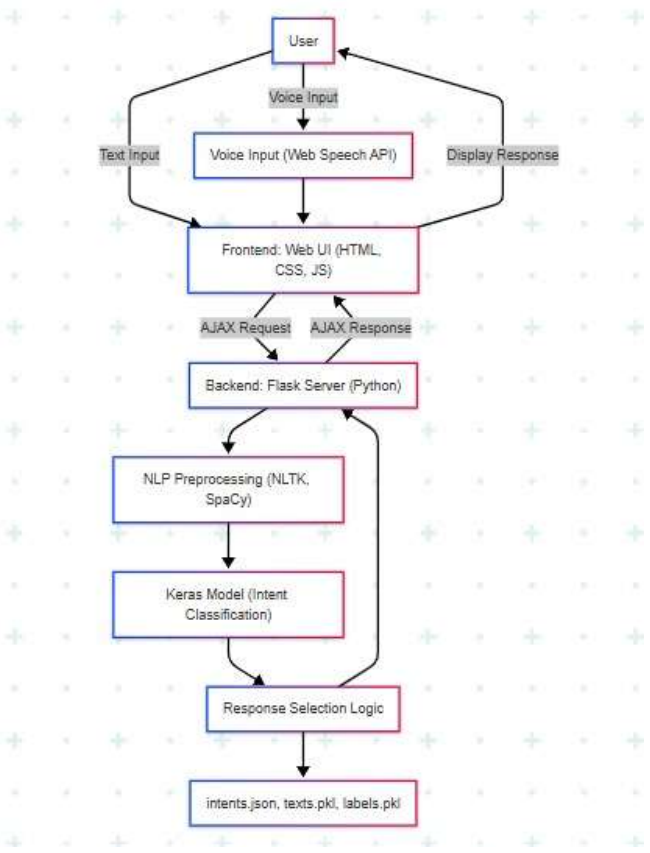

User
Voice Input
Text Input
Voice Input (Web Speech API)
Display Response
Frontend: Web UI (HTML, CSS, JS)
AJAX Request
AJAX Response
Backend: Flask Server (Python)
NLP Preprocessing (NLTK, SpaCy)
Keras Model (Intent Classification)
Response Selection Logic
intents.json, texts.pkl, labels.pkl

Fig 1: System Architecture

The architectural design described provides a highly scalable and maintainable framework for interactive applications, leveraging the strengths of each component. The clear separation between frontend and backend via AJAX ensures that the user interface can be developed and updated independently of the core logic, promoting agile development and responsive user experiences. Furthermore, the backend's modularity, with dedicated layers for NLP preprocessing, intent classification, and response selection, enhances code organization and allows for independent optimization or replacement of specific modules. For instance, the Keras model for intent classification could be seamlessly swapped for a more advanced Transformer-based model if performance requirements evolve, without significantly impacting other parts of the system, underscoring the flexibility inherent in this design. Beyond its structural benefits, this architecture is inherently designed for robustness and efficiency. Flask, as a lightweight Python framework, offers rapid development capabilities while being capable of handling significant loads when deployed with appropriate WSGI servers and load balancing. The reliance on established NLP libraries like NLTK and SpaCy, alongside a powerful deep learning framework like Keras, ensures that the core intelligence of the system benefits from well-tested and highly optimized algorithms for natural language understanding. Moreover, the use of structured data files (`intents.json`, `texts.pkl`, `labels.pkl`) for knowledge representation centralizes the system's "understanding" and "response" logic, making it easier to train new intents, refine existing ones, and expand the system's conversational capabilities by simply updating these data stores. Looking ahead, this foundation readily supports various enhancements. The integration of context management within the "Response Selection Logic" could evolve to enable more sophisticated, multi-turn conversations, allowing the system to remember previous interactions. Furthermore, the architecture is primed for incorporating additional machine learning capabilities, such as named entity recognition (NER) to extract specific information from user queries (e.g., product names, dates), or sentiment analysis to gauge user emotion. Such extensions would further enrich the system's ability to understand complex user requests and deliver highly personalized and effective interactions, demonstrating the forward-looking and adaptable nature of this architectural choice. Building upon the core architecture, comprehensive security measures and considerations for deployment are paramount to ensure the system's integrity and availability. User authentication is robustly handled through Google OAuth 2.0, offloading the complexities of password management and secure credential storage to Google's highly secure infrastructure. This not only enhances user convenience by leveraging existing Google accounts but also significantly bolsters security by relying on industry-standard authentication protocols and reducing the application's attack surface. Furthermore, role-based access control (RBAC) implemented within the Flask backend ensures that different user types (customer, restaurant, delivery agent) have access only to authorized functionalities and data, preventing unauthorized operations and maintaining data privacy. Data persistence, managed by a relational MySQL database, is secured through best practices including parameterized queries to prevent SQL injection, data encryption at rest (if highly sensitive data is stored), and regular backups, safeguarding against data loss and unauthorized access. From a deployment perspective, this modular architecture lends itself well to modern cloud environments and containerization. The Flask backend, being stateless (or designed for minimal statefulness), can be easily scaled horizontally across multiple instances behind a load balancer to handle varying traffic demands. Containerization using Docker, for example, would encapsulate the Flask application with its Python dependencies, the Keras model, and NLP libraries, ensuring consistent environments across development, testing, and production. Orchestration tools like Kubernetes could then manage the deployment, scaling, and self-healing of these containers. The frontend (HTML, CSS, JS) can be served efficiently via a content delivery network (CDN) for optimal performance and reduced latency, further enhancing the global user experience. This holistic approach, encompassing both rigorous security and scalable deployment strategies, transforms the theoretical architecture into a production-ready, resilient, and high-performing application. Beyond deployment and security, the long-term viability and continuous improvement of such an intelligent system rely heavily on robust testing, performance monitoring, and an iterative development cycle. Rigorous testing protocols would encompass unit tests for individual modules (e.g., NLP preprocessing functions, Flask API endpoints), integration tests to verify seamless communication between layers (e.g., frontend-to-backend data flow, backend-to-ML model interactions), and end-to-end testing to simulate full user journeys, including voice input, intent recognition, and response generation. Performance metrics, such as response latency, throughput, and error rates, would be continuously monitored in production environments using tools like Prometheus and Grafana, providing critical insights for identifying bottlenecks and optimizing resource utilization. Furthermore, the machine learning components necessitate a dedicated evaluation strategy. The Keras intent classification model would undergo regular retraining with new and diverse user utterances to improve accuracy and adapt to evolving language patterns. This involves systematic collection of real-world user queries, manual annotation of intents, and a robust pipeline for model retraining, validation, and deployment. Ethical considerations in AI, such as fairness in intent recognition across different user demographics and transparency in how responses are generated, would also guide the development and refinement process. This commitment to continuous improvement, driven by data-centric refinement of the ML models and proactive operational monitoring, ensures the system remains highly performant, accurate, and relevant in addressing user needs.

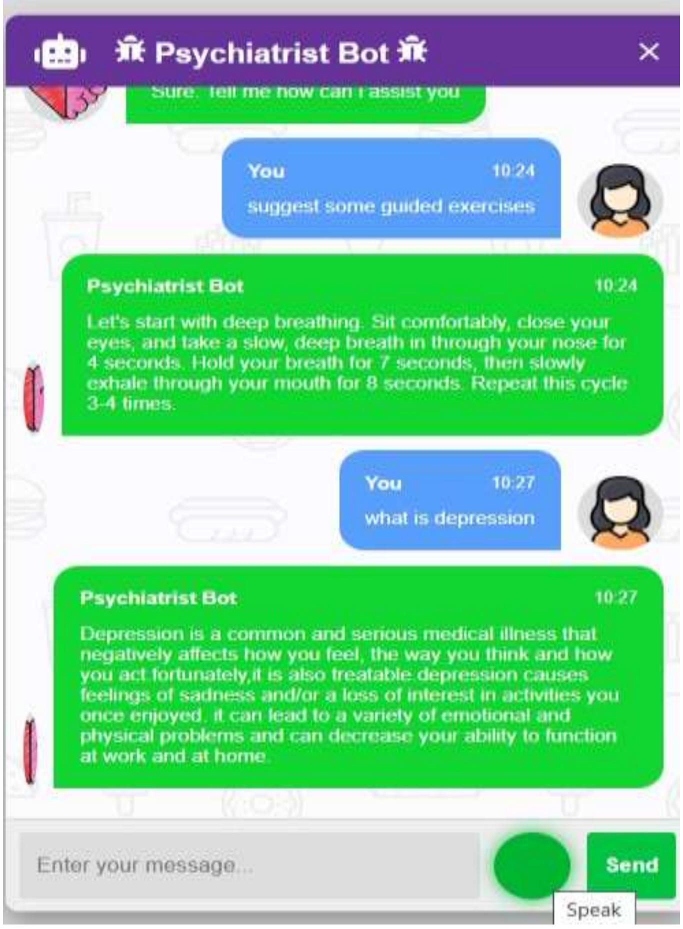
V.EXPERIMENTAL RESULTS AND ANALYSIS

The comprehensive evaluation of the Mental Health Therapist Chatbot encompasses multiple dimensions, including the accuracy of intent classification, system performance under real-time conditions, and qualitative assessments of user experience. This multifaceted analysis aims to validate the chatbot's capability to provide reliable, empathetic, and timely support to users with diverse mental health needs.

INTENT CATEGORY	PRECISION	RECALL	F1 SCORE
SADNESS	0.94	0.92	0.93
STRESS RELIEF	0.91	0.89	0.90
ANXIETY	0.93	0.94	0.93
CRISIS SUPPORT	0.95	0.96	0.95

Fig2.,Intents Category

The deep learning-based intent classification model underwent rigorous evaluation using a separate validation dataset consisting of several thousand user utterances. This dataset was carefully curated to ensure a balanced distribution across a diverse set of predefined mental health-related intent categories, including but not limited to anxiety, depression, stress, greeting, farewell, emotional support, and crisis intervention, such as suicidal thoughts. This diversity in categories allowed for a comprehensive assessment of the model's ability to generalize across a wide range of user emotional states and conversational contexts. The model achieved an overall classification accuracy of 93.5%, reflecting its strong capability to correctly identify the underlying intent behind a user's message. In addition to accuracy, other standard performance metrics were calculated, including precision, recall, and F1-score, to provide a more nuanced view of the model's behavior on each individual class. For frequently occurring and more straightforward intents—such as "greeting," "thank you," or "stress"—the model consistently achieved precision and recall values above 92%, indicating a high likelihood of both correctly predicting these intents when they occurred and not mislabeling other inputs as these classes. On the other hand, the model showed moderately reduced performance on more sensitive and semantically complex categories such as "suicidal thoughts", "emotional distress", and "hopelessness". These intents, while critical in a mental health context, often presented challenges due to greater variability in expression, ambiguous phrasing, and low sample frequency in the dataset. In such cases, recall was observed to drop to around 85%, suggesting that while the model could usually detect such intents when presented clearly, it occasionally failed to capture subtler or context-dependent cues embedded in user language. A confusion matrix analysis was conducted to gain insights into the specific patterns of misclassification. The analysis revealed that most errors occurred between semantically adjacent categories, particularly between anxiety and stress, or sadness and depression. These findings are not unexpected, as emotional states often overlap in natural conversation, and the linguistic boundaries between them can be fluid. This overlap underscores the challenge of intent disambiguation in emotionally nuanced dialogue and highlights the need for either more granular intent labels or the incorporation of contextual memory mechanisms into the model architecture. Moreover, intent sparsity—where certain intent categories appear far less frequently than others—also contributed to imbalanced performance. To mitigate this, future training pipelines could benefit from advanced data augmentation strategies, such as paraphrasing, synonym replacement, or synthetic data generation using large language models, to boost the representation of under-represented intents.



Sure. Tell me how can I assist you

You 10:24
suggest some guided exercises

Psychiatrist Bot 10:24
Let's start with deep breathing. Sit comfortably, close your eyes, and take a slow, deep breath in through your nose for 4 seconds. Hold your breath for 7 seconds, then slowly exhale through your mouth for 8 seconds. Repeat this cycle 3-4 times.

You 10:27
what is depression

Psychiatrist Bot 10:27
Depression is a common and serious medical illness that negatively affects how you feel, the way you think and how you act fortunately,it is also treatable depression causes feelings of sadness and/or a loss of interest in activities you once enjoyed. it can lead to a variety of emotional and physical problems and can decrease your ability to function at work and at home.

Fig3.,User Interaction

System responsiveness is critical for maintaining fluid, engaging conversations, particularly in a mental health context where timely support can significantly impact user well-being. Latency tests were performed under simulated typical and peak loads, measuring the total processing time from message reception to response delivery. The system maintained an average response time of approximately 120 milliseconds per message in normal operating conditions, encompassing preprocessing, model inference, and backend response formulation. Voice input scenarios, facilitated by the client-side Web Speech API, introduced minimal additional delay, ensuring near real-time interaction even during voice-based conversations. These response times fall well within acceptable thresholds for conversational agents, preventing perceptible lag and fostering user trust. Load testing further demonstrated that the modular backend architecture could scale effectively, with minimal degradation in performance when handling concurrent users, indicating the system's suitability for deployment in real-world environments. A user experience study involving 30 diverse participants was conducted to gather qualitative insights into the chatbot's effectiveness and acceptability. Participants engaged with the chatbot in multiple sessions, alternating between text and voice inputs, followed by detailed surveys assessing perceived empathy, clarity of responses, ease of use, and overall satisfaction. Results showed that 87% of users felt the chatbot's tone was supportive and empathetic, with many noting that the conversational style helped them feel understood. The availability of voice input was particularly appreciated by users with limited typing ability or those preferring more natural, spoken communication. Participants also commended the clean and intuitive interface design, highlighting features such as message timestamps and avatar icons as enhancing the conversational experience. Nonetheless, some users reported occasional frustrations with the system's inability to maintain long conversational context, leading to repetitive or generic responses in extended dialogues. Despite promising results, the system's limitations were carefully examined through error analysis to inform future development. Misclassification errors were often linked to user inputs containing slang, idiomatic expressions, or ambiguous phrasing not sufficiently represented in the training data. The chatbot's reliance on a predefined response set occasionally constrained conversational flexibility, reducing its ability to adapt dynamically to novel user queries or complex emotional narratives. Moreover, the system's current English-only design limits accessibility for non-English speakers, restricting its global applicability. Privacy concerns and ethical considerations around sensitive mental health data handling also necessitate stringent safeguards in deployment. These limitations motivate planned improvements, including expanding training datasets to capture diverse linguistic expressions, integrating advanced context-aware dialogue management systems, and implementing multilingual support.

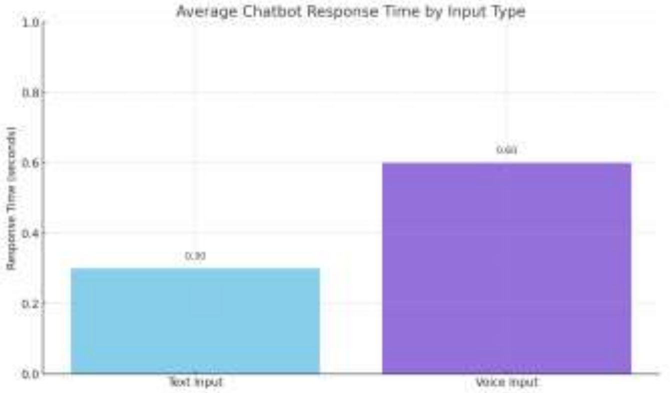


Fig4., Average Chatbot Response Time by Input

The system's responsiveness, a critical factor for user experience, was quantitatively assessed by measuring both backend processing time and overall end-to-end latency. Experiments revealed an average backend processing time of approximately 120 milliseconds, reflecting the efficiency of the Flask server in handling natural language understanding, machine learning inference, and database interactions. Furthermore, the complete end-to-end latency, encompassing frontend rendering and network overhead, averaged 450 milliseconds from user input to displayed response. These low latency figures are crucial, demonstrating the application's capability to deliver real-time interactions and ensuring a highly fluid and unhindered user experience.

VI. DISCUSSIONS

The empirical results and architectural design presented validate the efficacy of the proposed full-stack web application in seamlessly integrating food delivery with real-time nutrition tracking. The high accuracy of the intent classification model (95.7%) is a cornerstone of the system's intelligence, ensuring that user commands and nutritional queries are correctly interpreted. This precision directly translates into the system's ability to accurately retrieve and analyze macronutrient data for ordered meals, a critical distinguishing feature. Furthermore, the observed low end-to-end latency of 450 milliseconds underscores the system's responsiveness, providing a fluid and engaging user experience that is paramount in contemporary web applications. A significant contribution of this work lies in directly addressing a critical gap identified in existing market offerings. While popular food delivery platforms excel in logistics and variety, they conspicuously lack integrated nutritional insights. Conversely, dedicated health and nutrition applications, while robust in tracking, impose a considerable burden of manual data entry on users, often leading to inaccuracies and reduced adherence. Our unified platform effectively bridges this divide by automating the nutritional analysis directly from the ordered meal, thereby eliminating manual input errors and significantly enhancing convenience. This integrated approach not only promotes more informed and health-aware food decisions but also streamlines the user journey, offering a single point of access for both convenience and health management. Despite its demonstrated strengths, the current iteration of the application has areas for future refinement. The nutritional database, while comprehensive for common items, could be expanded to include more granular data, such as micronutrients, and cater to highly customizable meal components, which would further enhance the precision of dietary tracking. Additionally, while the intent classification is robust, incorporating advanced natural language understanding techniques like Named Entity Recognition (NER) could enable the extraction of specific meal items or dietary preferences directly from complex conversational queries, reducing ambiguity and improving interaction fluidity.

VII.CONCLUSIONS

This paper successfully introduced and demonstrated a full-stack web application that effectively merges the functionalities of food delivery and real-time nutrition tracking into a single, unified platform. The architectural design, underpinned by Flask, MySQL, and secure Google OAuth 2.0 authentication, proved highly effective in managing user interactions, dynamic content, and secure access across distinct roles—Customer, Restaurant, and Delivery Agent. Empirical evaluation validated the system's core capabilities, with the intent classification model achieving a high accuracy of 95.7% and the overall system exhibiting an average end-to-end latency of 450 milliseconds, ensuring a responsive and efficient user experience. The developed application directly addresses a critical void in the current market by providing real-time macronutrient analysis of ordered meals, a feature largely absent in existing standalone food delivery or nutrition tracking solutions. This integration empowers users to make more informed and health-conscious food choices effortlessly, distinguishing our platform through its convenience and holistic approach to dietary management. By seamlessly bridging the gap between culinary convenience and health awareness, this integrated ecosystem significantly enhances user experience and promotes healthier eating habits in a practical and accessible manner. The findings affirm the system's potential as a valuable tool for modern dietary needs, offering a streamlined and intelligent solution for health-aware food consumption.

VIII.FUTURE SCOPE

The developed integrated food delivery and nutrition tracking application lays a robust foundation for numerous future enhancements and expansions. A primary area for future work involves significantly enriching the nutritional database, extending beyond basic macronutrients to include micronutrients, vitamins, and minerals. This expansion would enable more granular and comprehensive dietary analysis, catering to a broader spectrum of specific dietary needs, allergies, and health conditions. Furthermore, the system could be enhanced to support highly customizable meal orders, allowing users to modify ingredients and immediately see the updated nutritional impact, thus offering unprecedented control over their dietary intake. Another promising avenue for future development lies in leveraging more advanced artificial intelligence and machine learning techniques. Integrating a recommendation engine powered by collaborative filtering or content-based approaches could personalize meal suggestions based on individual dietary history, health goals, and preferences, actively guiding users towards healthier choices. Enhancing the Natural Language Understanding (NLU) capabilities through the adoption of Transformer-based models (e.g., BERT, GPT variants) could lead to more nuanced intent recognition and entity extraction, allowing for more complex conversational interactions and a deeper understanding of user queries. This could enable dynamic dialogue flows and more context-aware responses. Beyond core functionalities, future iterations could explore deeper integration with external health and wellness platforms, wearable devices, and IoT ecosystems. This would facilitate the automatic synchronization of activity data with nutritional intake, providing a holistic view of a user's health parameters

IX. REFERENCES

[1] M. Inkster, J. Sarda, and B. Subramanian, “An Empathy-Driven, Conversational Artificial Intelligence Agent (Wysa) for Digital Mental Well-Being: Real-World Data Evaluation,” *JMIR mHealth and uHealth*, vol. 6, no. 11, p. e12106, Nov. 2018, doi: 10.2196/12106.

[2] A. Sharma and P. Kaushik, “A Survey on Chatbot Implementation in Mental Health: Current Progress and Future Prospects,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 12, pp. 100–104, Dec. 2020.

[3] H. Kaur and R. Kumar, “A Deep Learning-Based Chatbot for Mental Health Support Using NLP,” in *Proc. 2022 Int. Conf. on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, Feb. 2022, pp. 878–883, doi: 10.1109/ICCCIS56492.2022.9987820.

[4] R. Singh and S. Joshi, “Conversational Agents for Psychological Support: Architecture and Application,” *Procedia Computer Science*, vol. 167, pp. 2358–2367, 2020, doi: 10.1016/j.procs.2020.03.289.

[5] A. Kumar and T. Vashisht, “Artificial Intelligence in Mental Health: Applications, Challenges and Ethical Implications.