AI Support Assistant

TEAM: TRAGICBYTES

Team Members: Vedang Wajge Bhagyesh Patil
Soham Satpute Laksh Sodhai

Objectives : Steps Implemented

•AI-Powered Intelligent Support
The chatbot uses advanced NLP (e.g., GPT, Gemini AI) for natural language understanding, contextual conversation handling, and intent recognition to resolve user queries efficiently.

•Self-Updating Knowledge Base
A dynamic, structured repository (e.g., MongoDB) enables easy content management, while the chatbot continuously learns from user interactions and feedback to improve accuracy over time.

•Real-Time ERP Data Access
Integrated via RESTful APIs with secure OAuth2 authentication, the chatbot fetches real-time data (Sales, HR, Finance) to provide users with instant, actionable insights directly within the chat.

•Multi-Channel and Scalable Access
Accessible across web, WhatsApp, Teams, Slack, and email, with future support for voice-based interactions, ensuring widespread adoption and user convenience.

Objectives : Steps Implemented

•**Efficient Escalation and Security**
Automatically escalates unresolved queries to human support, while role-based access ensures users only access data relevant to their permissions, maintaining security and compliance.

•**Insightful Analytics and Reporting**
A centralized dashboard (e.g., via Power BI/Tableau) offers performance monitoring, user query trends, and knowledge base gaps, enabling data-driven improvements and support optimization.

Challenges And Solutions:

1. **Complex Query Understanding**
   - **Challenge**: Handling multi-part, context-heavy queries from users.
   - **Solution**: Used fine-tuned NLP models, multi-step intent recognition, and context-aware conversation flow.

2. **Real-Time Data Integration**
   - **Challenge**: Securely and efficiently fetching ERP data without delays.
   - **Solution**: Used RESTful APIs with OAuth2, caching, and asynchronous data fetching for speed and security.

3. **Multi-Channel Support**
   - **Challenge**: Ensuring consistent chatbot behavior across platforms (e.g., WhatsApp, Teams).
   - **Solution**: Modular backend, platform-specific SDKs/webhooks, and extensive cross-platform testing.

Challenges And Solutions:

4.Knowledge Base Management

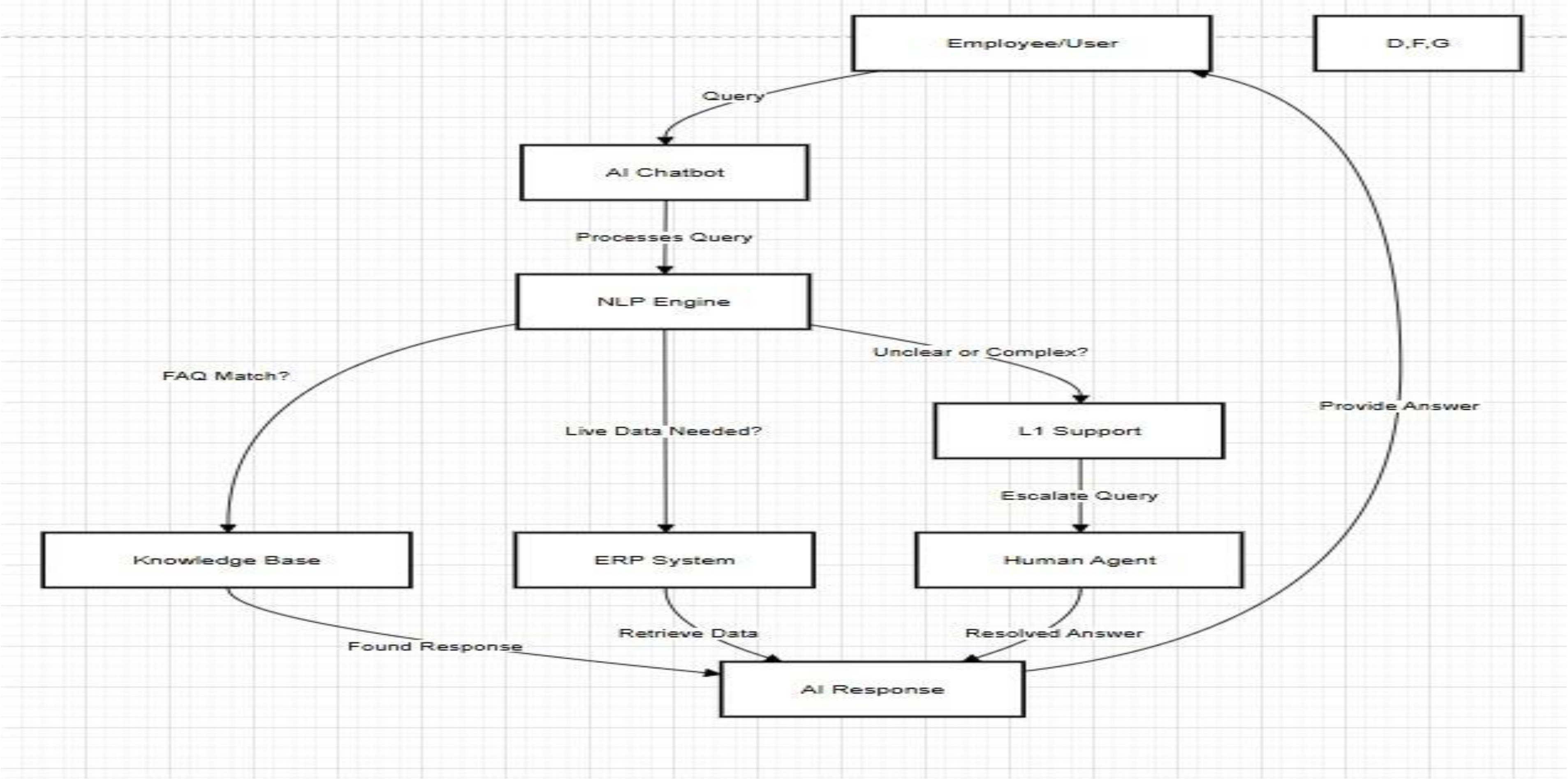
- **Challenge**: Updating content easily while maintaining relevance and accuracy.
- **Solution**: Built an admin dashboard, AI-driven content checks, and a feedback loop for knowledge updates.

5.Role-Based Access Control

- **Challenge**: Delivering secure, role-specific data to users.
- **Solution**: Integrated RBAC using ERP roles, with secure authentication and thorough testing.

Solution Architecture

DFD Diagram:



Tech Stack:

1.User Interface Layer

1. Platforms: Web (React.js), WhatsApp (Twilio API), Microsoft Teams (Bot Framework SDK), Slack (Slack SDK), Email

2.AI Chatbot Engine

1. Backend: Node.js with Express.js
2. NLP Models: OpenAI GPT-4 / Google Gemini AI, spaCy, HuggingFace Transformers

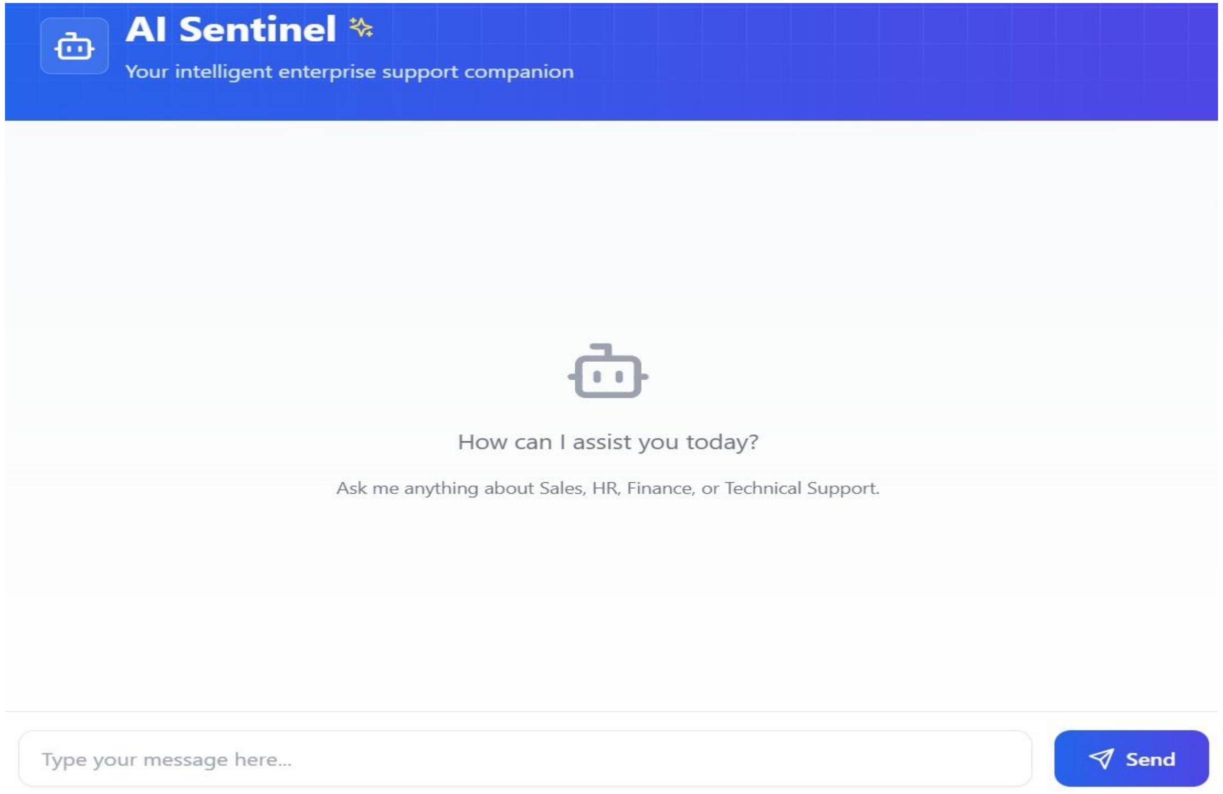
3.Knowledge Base Management

1. Database: MongoDB
2. Admin Dashboard: React.js (Frontend) + Node.js (Backend)

4.ERP System Integration

1. RESTful APIs with OAuth2 Authentication
2. Middleware/API Gateway: AWS API Gateway or Kong

Project Screenshot:



AI Sentinel

Your intelligent enterprise support companion

How can I assist you today?

Ask me anything about Sales, HR, Finance, or Technical Support.