**Problem Statement / Description**

A retail company named ShopEase Ltd. is in the business of selling consumer products across multiple cities through both physical stores and an e-commerce website. As the business grows, the company faces challenges in managing customer support effectively, since customers contact the company with issues that may involve describing defective items, wrong deliveries, or billing concerns.

To improve customer satisfaction, the company wants a web-based software solution (with a companion mobile app) that integrates multimodal AI support for handling customer service requests.

The system will allow a Customer to register and log in to their account. Customers can create Support Tickets by:

* Typing a description of their issue (text),
* Speaking their issue using voice (converted to text automatically), or
* Uploading an image (e.g., damaged product, incorrect shipment, or receipt).

An AI-powered assistant (LLM) will process the multimodal input, analyze the problem, and suggest potential solutions in real-time. If the AI cannot resolve the issue, the ticket will be escalated to a Support Agent, who can view the full ticket details, including AI suggestions, customer history, and uploaded images.

Each support ticket will have a status (Open, Pending, Resolved, Escalated), and customers will be notified of updates by email or in-app alerts. The system will also prevent a customer from opening more than three unresolved tickets at a time.

Support Agents should be able to log in, view assigned tickets, and respond to customers directly through the system. The company’s Customer Support Manager can monitor all tickets, track resolution time, and view analytics reports on the performance of AI vs. human resolutions.

The system will thus reduce response times, improve customer satisfaction, and lower support costs by combining multimodal AI with human support workflows.

## **✅ Functional Requirements**

1. **Customer Management**
   * The system must allow customers to register an account with their name, email, phone, and shipping address.
   * Customers must be able to log in securely and manage their profile information.
2. **Support Ticket Creation**
   * Customers must be able to create support tickets by:
     + Typing a description of the issue (text input).
     + Recording their voice (converted to text automatically).
     + Uploading an image (e.g., broken item, wrong delivery, or receipt).
   * Each support ticket must have a unique ticket ID, issue description, creation date, and current status.
   * A customer must not be able to create more than **three unresolved tickets** at any given time.
3. **AI Assistant Integration**
   * The system must use an AI assistant (LLM) to analyze multimodal inputs (text, image, voice-transcribed text).
   * The AI must provide an automated response suggesting possible solutions.
   * If the AI cannot resolve the issue, the ticket must be automatically escalated to a human Support Agent.
4. **Support Agent Functionality**
   * Support Agents must be able to log in to the system.
   * Support Agents must be able to view all escalated/assigned tickets, including the customer’s description, uploaded images, and AI suggestions.
   * Support Agents must be able to respond to customers through the system and update the ticket status.
5. **Ticket Management**
   * Each ticket must have one of the following statuses: **Open, Pending, Resolved, or Escalated**.
   * Customers must be able to view the status of all their tickets.
   * Customers must be able to request ticket closure once they are satisfied with the resolution.
6. **Notification System**
   * The system must notify customers (via email and/or in-app alerts) when:
     + A ticket is created.
     + An AI or Support Agent has responded.
     + A ticket status changes.
7. **Customer Support Manager Features**
   * A Manager must be able to log in and view all tickets across the system.
   * The Manager must have access to reports showing:
     + Number of tickets created per day/week.
     + Resolution times (AI vs. human).
     + Percentage of AI-resolved vs. escalated tickets.

## **✅ Software Solution Architecture Diagram**

This diagram provides a high-level view of the software solution's architecture for the ShopEase customer service assistant. It follows a layered architectural pattern, separating the application into distinct tiers for clarity and maintainability.

### Physical and Logical Tiers

* **Client Tier:** This is the user-facing part of the system. It includes:
  + **Browser:** The web-based solution is an **HTML** application. It uses rich HTML, CSS, and **React** for a responsive user interface.
* **Middle Tier:** This tier contains the core application logic and services. It is hosted on an **Application Server** within a **Web Container**.
  + **Web Services:** The system uses **Python Uvicorn** to serve **FastAPI** **RESTful requests** from the client applications. This layer handles all communication and routing for the business logic.
  + **AI Service:** This is a crucial, distinct component for the system. It processes **multimodal input** (text, voice, and images) and leverages a large language model (LLM) to analyze issues and suggest solutions in real-time.
  + **Business Service Layer (POJO):** This layer contains the business logic, such as managing the status of tickets (Open, Pending, Resolved, Escalated), preventing customers from having more than three unresolved tickets, and handling the escalation of tickets to a Support Agent.
  + **Data Access Object (DAO) & Hibernate ORM:** This is the layer responsible for communicating with the database. It uses an **Object-Relational Mapping (ORM)** tool like Hibernate to manage the interaction between the application's objects and the database tables.
* **Data Tier:** This tier is where all the system's persistent data is stored.
  + **Database:** A relational database stores all the system's data, including customer accounts, support tickets, agent information, AI responses, and attachments.
  + **Enterprise Information Services:** The system integrates with external services for functionalities such as sending email notifications. This could also include legacy systems for customer history.

This architecture separates the presentation, business logic, and data layers, making the system modular, scalable, and easier to maintain.