# Project Title:

Banking Customer Support AI Agent using Multi-Agent Architecture

# Problem Scenario:

Modern digital banking platforms manage a high volume of customer service interactions, often through fragmented systems that struggle to personalize responses or provide timely status updates. There is a growing need for scalable, intelligent support systems that can understand user sentiment, manage service records efficiently, and handle real-time queries.

This project aims to develop a multi-agent Generative AI system tailored for banking customer support workflows to reduce manual effort, improve customer satisfaction, and ensure timely responses.

# Project Objective:

The objective is to design and implement a multi-agent AI assistant that can effectively classify incoming user messages into feedback (positive or negative) or queries and respond appropriately.

Key features include:

* Classification of user messages into positive feedback, negative feedback, or queries.
* Personalized responses generated based on classification and user sentiment.
* Ticket tracking and updates integrated with a support database for unresolved issues.

# Project Components:

1. **Multi-Agent Design and Execution Logic:**
   * **Classifier Agent:**  
     Takes unstructured user messages as input and categorizes them as positive feedback, negative feedback, or query, then routes the message to the respective downstream agent.
   * **Feedback Handler Agent:**  
     Activates upon classification as feedback. For positive feedback, generates warm, personalized thank-you messages. For negative feedback, generates a unique ticket number, inserts a new unresolved ticket into the support database, and returns an empathetic acknowledgment with the ticket number.
   * **Query Handler Agent:**  
     Activates when the input is a query. Extracts ticket numbers from messages, queries the ticket database, and returns the current ticket status to the user.
   * **Sample Use Case Flow:**
     + User: “Thanks for sorting out my net banking login issue.”  
       Response: “Thank you for your kind words! We’re happy to support you.”
     + User: “My debit card replacement still hasn’t arrived.”  
       Response: “We apologize for the inconvenience. A new ticket #784521 has been created. Our support team will look into this promptly.”
     + User: “Could you check the status of ticket 650932?”  
       Response: “Your ticket #650932 is currently marked as: Resolved.”
2. **LLMOps Evaluation, Logging, and Interface Design:**
   * **Model Evaluation:**  
     Assess the quality of generated responses focusing on feedback accuracy, empathy, and clarity of updates. Use QA scoring, test coverage for classification, and evaluate agent routing success.
   * **Streamlit UI Design:**  
     Develop an interactive dashboard to accept user inputs, simulate agent routing, display classification and responses, view historical queries with logs, and test various agent scenarios.
   * **Logs and Debugging View:**  
     Display prompt traces, classification outputs, ticket actions, and maintain logs of ticket IDs and agent success rates. Optionally, integrate user feedback into the improvement process.

This summary covers the project background, objectives, implementation plan, and evaluation aspects.