Automated Ticket Booking System using Agentic AI Workflow

# 1. Introduction

The goal of this project is to develop a fully automated flight booking system that uses Agentic AI for natural language interaction. The customer interacts with the system through a chatbot interface, which retrieves available flight itineraries from third-party APIs and handles booking confirmation. The process is entirely automated, and the user receives confirmation via email.

**Team Information**

Team Name: Major Cage

Team Members:

# Prasanndh Raaju M R ([prasanndhraaju.25@gmail.com](mailto:prasanndhraaju.25@gmail.com))

# Sachinsandron S ([sachinsandron12345@gmail.com](mailto:sachinsandron12345@gmail.com))

# Nivethika S ([nivethikasakthivel24@gmail.com](mailto:nivethikasakthivel24@gmail.com))

# 2. Key Design Decisions

## 2.1 System Architecture

- Language Model (LLM): The chosen LLM is OpenAI’s GPT, used for handling natural language processing (NLP) interactions with customers.  
- Agentic Framework: We have utilized AutoGen to manage the agentic workflow. This framework allows for automating tasks such as flight searches and booking confirmations.  
- Third-Party API Integration: Although real APIs for flights and booking are ideal, we simulate these interactions using mock APIs created with Flask. This allows for testing without requiring external API services.

The system is divided into the following components:  
- Frontend (Gradio): Chat-based interface to receive user inputs and display responses.  
- Backend (Flask APIs): Simulates the third-party API response for flight search and booking.  
- Email Service: Automatically sends confirmation emails using SMTP once the booking is completed.

## 2.2 Natural Language Interaction

The system uses GPT to manage conversation flow. The chatbot understands the user's input through keyword extraction (origin, destination, date) and uses that information to search flights.

## 2.3 Security Considerations

Security is a major concern in ticket booking systems, as it involves user data and payments. Even though this is a simulation:  
- Data Privacy: No sensitive data like passwords or credit card details are stored. Emails are sent via secure SMTP with encryption.  
- API Security: In a real-world scenario, secure API keys and HTTPS should be used.

# 3. Workflow

The overall workflow is outlined below:  
1. User Input: The user starts by entering their flight preferences in natural language (e.g., 'New York to Los Angeles on October 15th').  
2. Flight Search: The chatbot extracts origin, destination, and date from the input and makes a request to the flight search API.  
3. Flight Selection: Available flight options are shown, and the user selects their preferred flight by entering the flight number.  
4. Booking Confirmation: The system confirms the selected flight, books it, and sends a confirmation email with booking details.  
5. Post-Booking Interaction: If users have any follow-up questions, they can interact with the chatbot for further assistance.

# 4. Chat Interface

We are using Gradio to create the chatbot interface. It handles user input and maintains the state of conversation. The system interacts with the backend to fetch flight details and process booking confirmations.

# 5. Mock API Integration

For simulating real-world interactions, we developed mock APIs in Flask:  
- search\_flights: Returns available flight itineraries based on origin, destination, and date.  
- book\_flight: Confirms flight booking and generates a booking reference.

# 6. Email Service

Once the booking is complete, the user receives a confirmation email with details. We’ve used Python’s SMTP library for sending these emails.

# 7. Future Enhancements

Future developments could include:  
- Real-time flight availability: Integration with real-world flight APIs like Amadeus or Skyscanner.  
- Payment Gateway: Integration of secure payment gateways for handling real bookings.  
- Additional Travel Services: The system could be expanded to include hotel and car rental bookings.

# Conclusion

This project showcases a simple yet effective Agentic AI-based flight booking system using natural language processing and mock APIs. It demonstrates the potential of LLMs in automating user interactions for complex processes like ticket booking.