TITILE:- Gen-AI Restaurant Table Booking Chatbot

Gen-AI Powered Restaurant Table Booking Chatbot for Gen-Z

Description:-

*This project implements a **Gen-AI powered dining reservation chatbot** designed specifically for **Gen-Z users**, focusing on **fast response times, simple interaction**, and **night-only restaurant bookings**.*

The chatbot assists users with:

- *Table booking*
- *Table availability*
- *Booking cancellation*
- *Late arrival communication*

*The system combines **deterministic Python logic** for reliability with an **LLM (Google Gemini)** for generating **natural, human-like responses**.*

*This project was developed and tested using **Google Colab** and is suitable for **hackathons, academic submissions, and Gen-AI demos**.*

Problem Statement:-

Create a dining reservation chatbot using **compressed data structures** and **availability schedules**, streamlining bookings with improved response times while maintaining a conversational Gen-AI experience.

Tech Stack:-

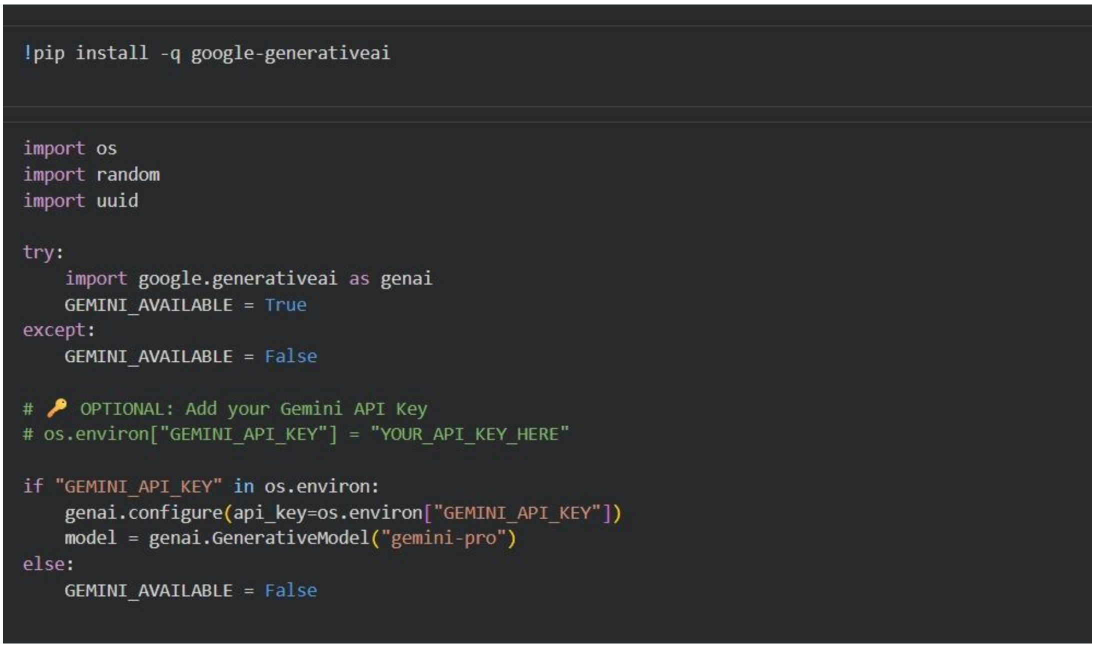
- **Language:** Python
- **LLM:** Google Gemini (Generative AI)
- **Environment:** Google Colab
- **Libraries:**
    1. google-generativeai
    2. uuid
    3. random

ALGORITHM:-

1. Display welcome message using LLM
2. Show available table types
3. Ask user to choose **book** or **cancel**
4. If booking:
    - Take table type
    - Take booking date (night-only)
    - Validate time (5 PM – 11 PM)
    - Take booking name
    - Generate unique booking ID
    - Store booking details
    - Display booking summary + branch contact
5. If cancellation:
    - Ask for booking ID
    - Validate booking ID
    - Cancel booking (non-refundable)
6. Generate all responses using LLM (fallback to normal text if LLM unavailable).

THE VIEW CODE:-

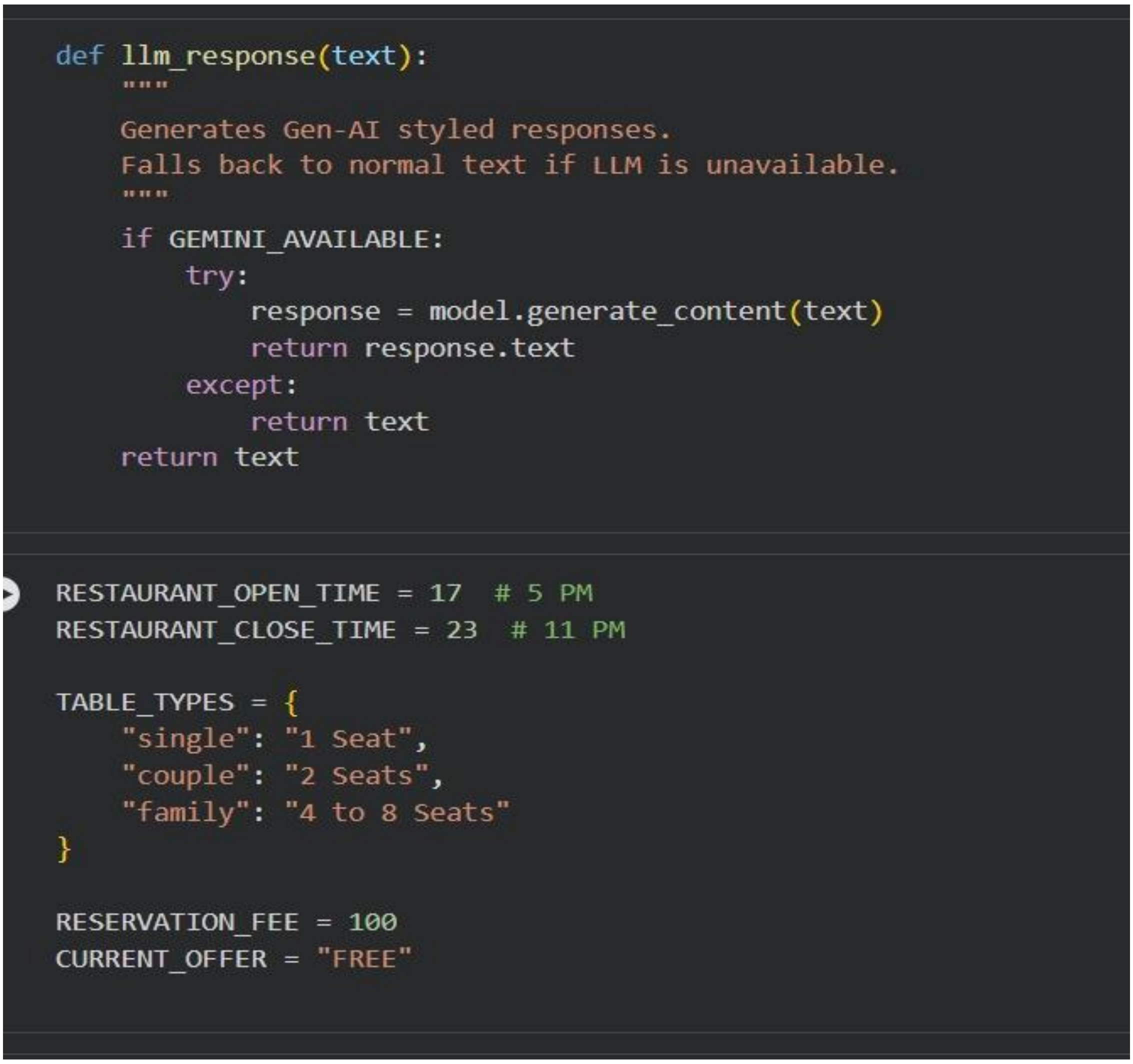
Step 1:- Here implemented the libraries..



```
!pip install -q google-generativeai
```

```
import os
import random
import uuid

try:
    import google.generativeai as genai
    GEMINI_AVAILABLE = True
except:
    GEMINI_AVAILABLE = False

# 🔑 OPTIONAL: Add your Gemini API Key
# os.environ["GEMINI_API_KEY"] = "YOUR_API_KEY_HERE"

if "GEMINI_API_KEY" in os.environ:
    genai.configure(api_key=os.environ["GEMINI_API_KEY"])
    model = genai.GenerativeModel("gemini-pro")
else:
    GEMINI_AVAILABLE = False
```

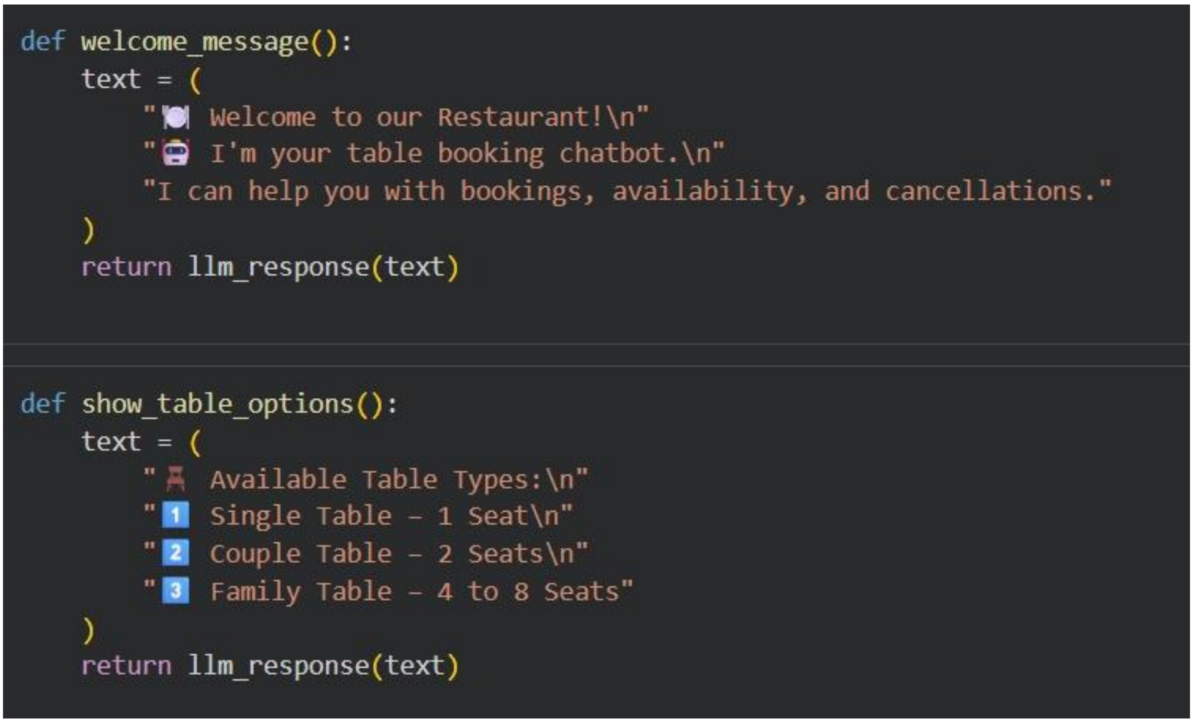
Step 2:- Data Structures are used..



```python
def llm_response(text):
    """
    Generates Gen-AI styled responses.
    Falls back to normal text if LLM is unavailable.
    """
    if GEMINI_AVAILABLE:
        try:
            response = model.generate_content(text)
            return response.text
        except:
            return text
    return text
```

```python
RESTAURANT_OPEN_TIME = 17   # 5 PM
RESTAURANT_CLOSE_TIME = 23   # 11 PM

TABLE_TYPES = {
    "single": "1 Seat",
    "couple": "2 Seats",
    "family": "4 to 8 Seats"
}

RESERVATION_FEE = 100
CURRENT_OFFER = "FREE"
```

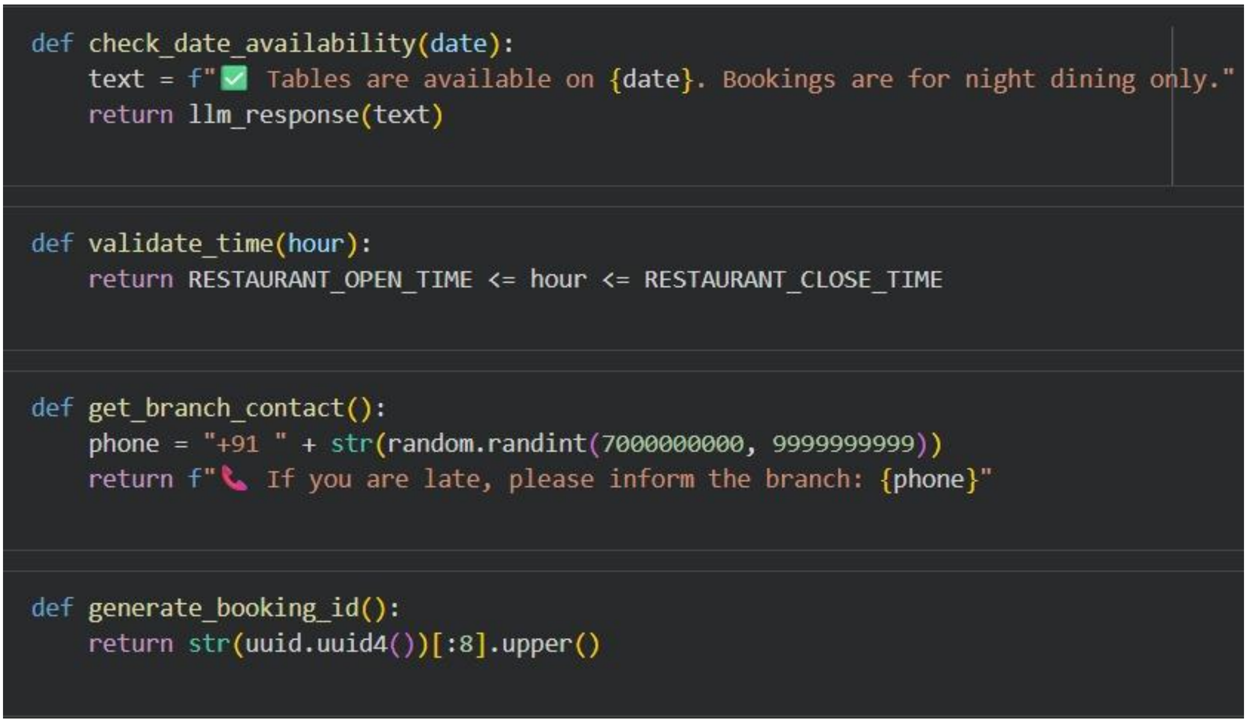
STEP 3:- LLM response are used..



```python
def welcome_message():
    text = (
        "🍽️ Welcome to our Restaurant!\n"
        "🤖 I'm your table booking chatbot.\n"
        "I can help you with bookings, availability, and cancellations."
    )
    return llm_response(text)
```

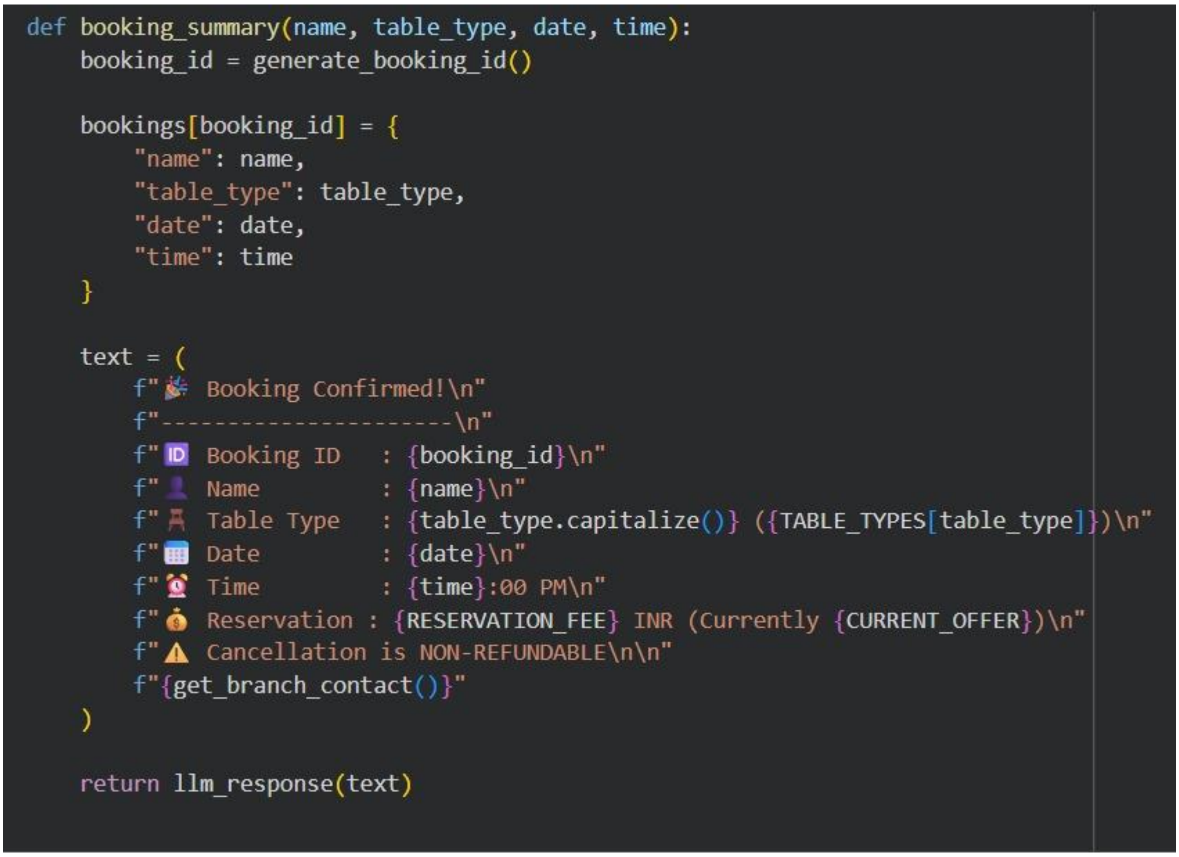
```python
def show_table_options():
    text = (
        "🪑 Available Table Types:\n"
        "1️⃣ Single Table – 1 Seat\n"
        "2️⃣ Couple Table – 2 Seats\n"
        "3️⃣ Family Table – 4 to 8 Seats"
    )
    return llm_response(text)
```

STEP 4:- Availability checking and Time..



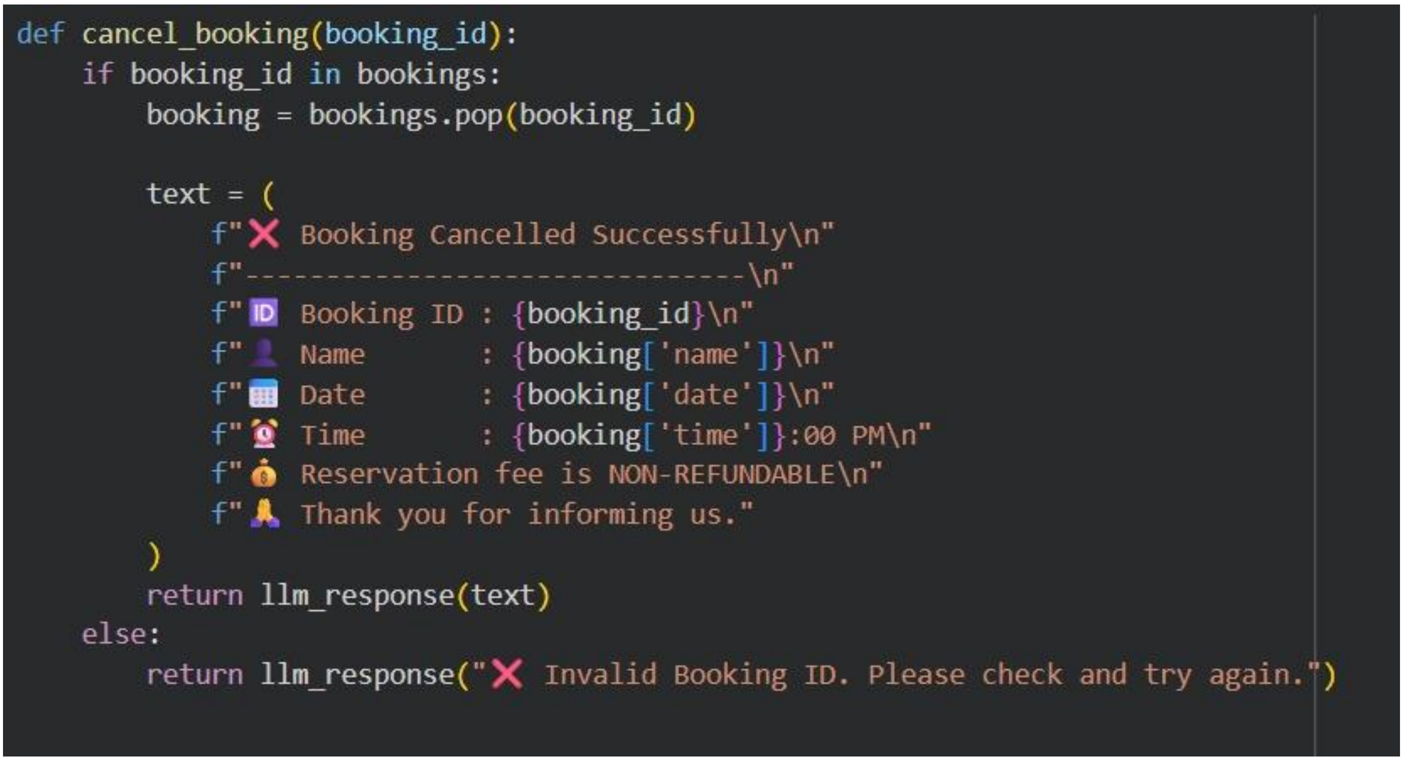
```python
def check_date_availability(date):
    text = f"✅ Tables are available on {date}. Bookings are for night dining only."
    return llm_response(text)


def validate_time(hour):
    return RESTAURANT_OPEN_TIME <= hour <= RESTAURANT_CLOSE_TIME


def get_branch_contact():
    phone = "+91 " + str(random.randint(7000000000, 9999999999))
    return f"📞 If you are late, please inform the branch: {phone}"


def generate_booking_id():
    return str(uuid.uuid4())[:8].upper()
```

STEP 5:- PROVIDING BOOKING ID AN BOOKING SUMMARY..



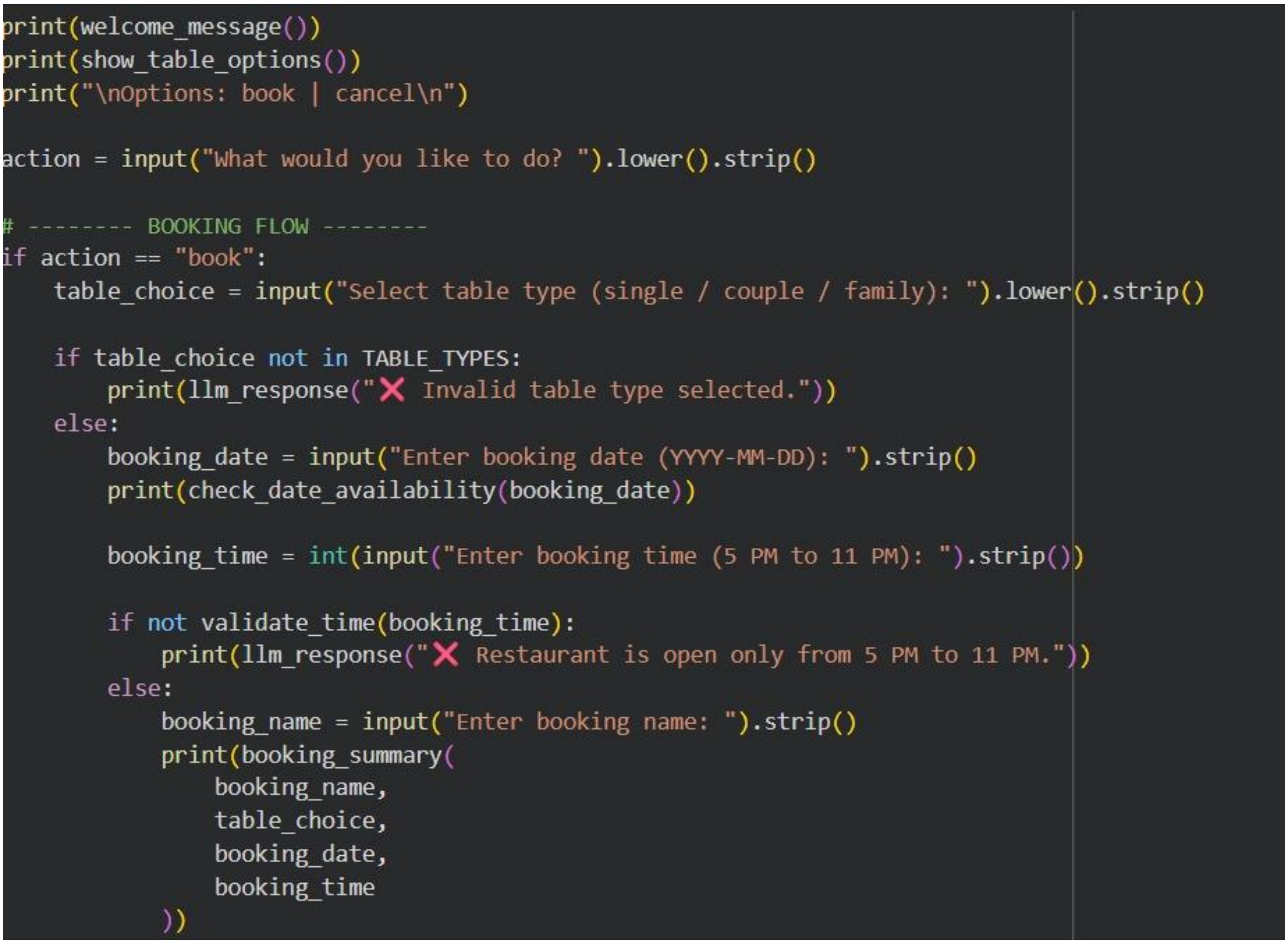
```python
def booking_summary(name, table_type, date, time):
    booking_id = generate_booking_id()

    bookings[booking_id] = {
        "name": name,
        "table_type": table_type,
        "date": date,
        "time": time
    }

    text = (
        f"🎉 Booking Confirmed!\n"
        f"----------------------\n"
        f"🆔 Booking ID   : {booking_id}\n"
        f"👤 Name         : {name}\n"
        f"🪑 Table Type   : {table_type.capitalize()} ({TABLE_TYPES[table_type]})\n"
        f"📅 Date         : {date}\n"
        f"⏰ Time         : {time}:00 PM\n"
        f"💰 Reservation : {RESERVATION_FEE} INR (Currently {CURRENT_OFFER})\n"
        f"⚠️ Cancellation is NON-REFUNDABLE\n\n"
        f"{get_branch_contact()}"
    )

    return llm_response(text)
```

STEP 6:- FOR CANCELATION USED THIS BLOCK..



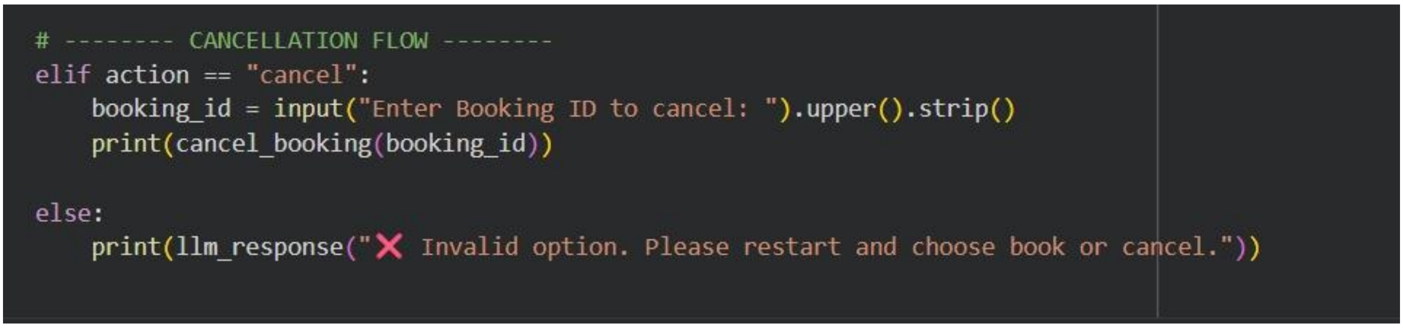
```python
def cancel_booking(booking_id):
    if booking_id in bookings:
        booking = bookings.pop(booking_id)

        text = (
            f"❌ Booking Cancelled Successfully\n"
            f"--------------------------------\n"
            f"🆔 Booking ID : {booking_id}\n"
            f"👤 Name       : {booking['name']}\n"
            f"📅 Date       : {booking['date']}\n"
            f"⏰ Time       : {booking['time']}:00 PM\n"
            f"💰 Reservation fee is NON-REFUNDABLE\n"
            f"🙏 Thank you for informing us."
        )
        return llm_response(text)
    else:
        return llm_response("❌ Invalid Booking ID. Please check and try again.")
```

STEP 7:- BRINGING ALL THE FUNCTION AND CALLING..



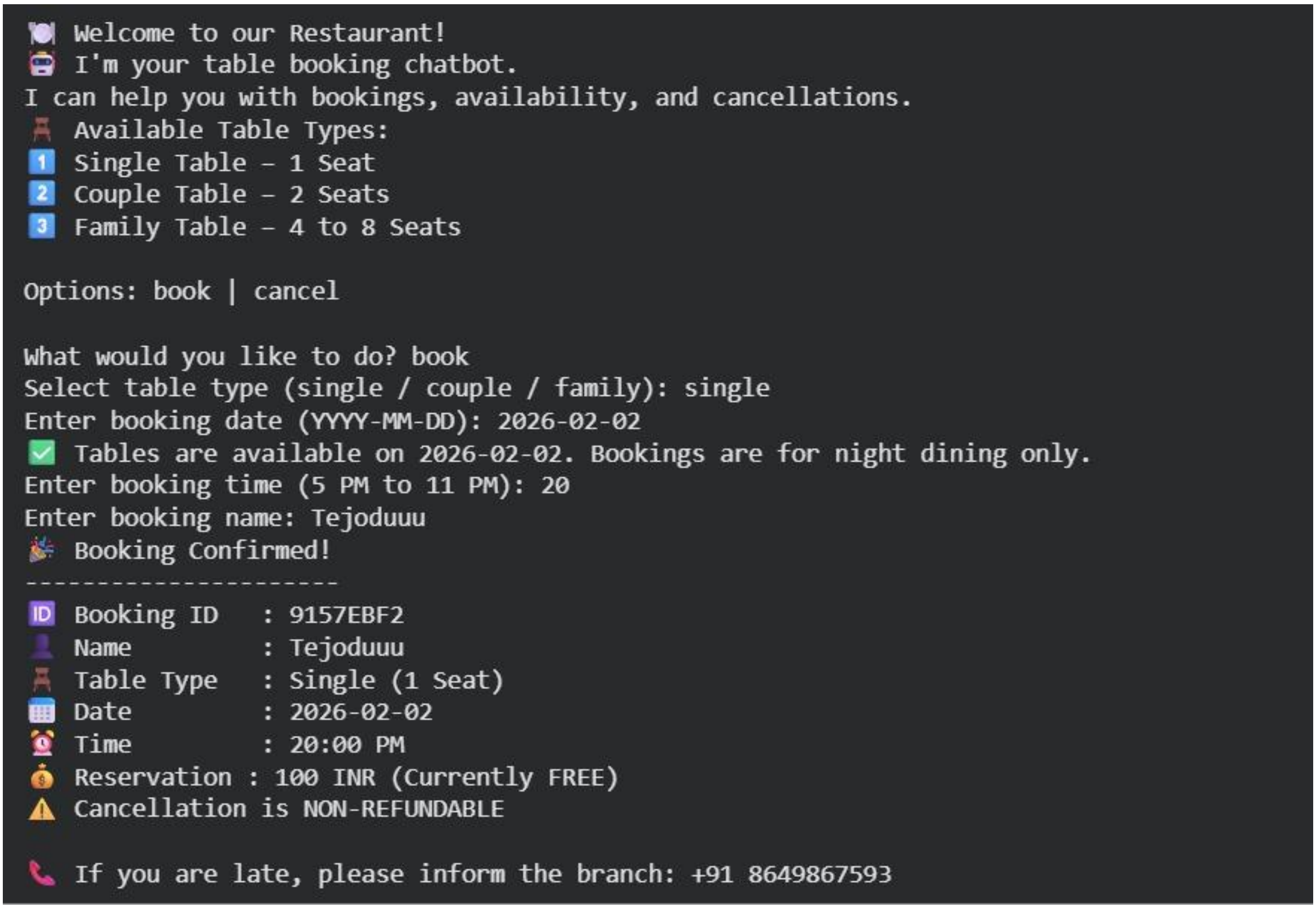
```python
print(welcome_message())
print(show_table_options())
print("\nOptions: book | cancel\n")

action = input("What would you like to do? ").lower().strip()

# -------- BOOKING FLOW --------
if action == "book":
    table_choice = input("Select table type (single / couple / family): ").lower().strip()

    if table_choice not in TABLE_TYPES:
        print(llm_response("❌ Invalid table type selected."))
    else:
        booking_date = input("Enter booking date (YYYY-MM-DD): ").strip()
        print(check_date_availability(booking_date))

        booking_time = int(input("Enter booking time (5 PM to 11 PM): ").strip())

        if not validate_time(booking_time):
            print(llm_response("❌ Restaurant is open only from 5 PM to 11 PM."))
        else:
            booking_name = input("Enter booking name: ").strip()
            print(booking_summary(
                booking_name,
                table_choice,
                booking_date,
                booking_time
            ))
```

STEP 8:- FINALLY FOR CANCELLATION FLOW AND CONTACT NUMBER..



```
# -------- CANCELLATION FLOW --------
elif action == "cancel":
    booking_id = input("Enter Booking ID to cancel: ").upper().strip()
    print(cancel_booking(booking_id))

else:
    print(llm_response("❌ Invalid option. Please restart and choose book or cancel."))
```

OUT PUT:-



```
🍽️ Welcome to our Restaurant!
🤖 I'm your table booking chatbot.
I can help you with bookings, availability, and cancellations.
🪑 Available Table Types:
1️⃣ Single Table – 1 Seat
2️⃣ Couple Table – 2 Seats
3️⃣ Family Table – 4 to 8 Seats

Options: book | cancel

What would you like to do? book
Select table type (single / couple / family): single
Enter booking date (YYYY-MM-DD): 2026-02-02
✅ Tables are available on 2026-02-02. Bookings are for night dining only.
Enter booking time (5 PM to 11 PM): 20
Enter booking name: Tejoduuu
🎉 Booking Confirmed!
----------------------
🆔 Booking ID   : 9157EBF2
👤 Name         : Tejoduuu
🪑 Table Type   : Single (1 Seat)
📅 Date         : 2026-02-02
⏰ Time         : 20:00 PM
💰 Reservation : 100 INR (Currently FREE)
⚠️ Cancellation is NON-REFUNDABLE

📞 If you are late, please inform the branch: +91 8649867593
```

Performance Optimization:-

- Compressed data structures
- No heavy ML inference for logic
- LLM used only for response generation
- Minimal API calls
- Fast execution in Google Colab

Why Gen-AI?:-

- Enhances user experience with natural conversation
- Makes chatbot feel human-like
- Suitable for Gen-Z interaction patterns
- Reduces rigid, rule-based responses

Error Handling:-

- Invalid table type detection
- Invalid time validation
- Invalid booking ID handling
- LLM fallback if API key is missing

Future Enhancements:-

- Streamlit / Web UI
- Database integration (Firebase / SQLite)
- OTP-based booking confirmation
- WhatsApp or Telegram bot integration
- Multi-branch support
- Dynamic pricing

How to Run:-

1. Open **Google Colab**
2. Copy code cells step-by-step
3. (Optional) Add Gemini API Key
4. Run the final chatbot cell
5. Start booking or cancelling tables

THANK YOU..