Task: Time Zone-Based Appointment Scheduling System

Objective

To build a backend function that facilitates booking appointments across multiple U.S. time zones while maintaining a standardized booking system in Indian Standard Time (IST). The system will fetch available time slots, convert them based on user preferences, and finalize bookings accordingly.

Workflow Overview

1. **User Request for Availability**
   - The user provides their preferred time zone (e.g., Pacific Time Zone, Central Time Zone, Eastern Time Zone).
   - This time zone is sent to a webhook that fetches free/busy slots from a Google Calendar API.
2. **Fetching Available Time Slots**
   - The webhook retrieves available slots from Google Calendar.
   - These slots are initially stored in IST (Indian Standard Time).
3. **Converting and Presenting Time Slots**
   - Convert the available time slots from IST to the user's requested time zone.
   - Organize these into one-hour windows.
   - The formatted list is sent to Retell AI.
   - Retell AI selects 2-3 suitable options and presents them to the user in a conversational manner.
4. **User Selection & Booking Confirmation**
   - The user selects a preferred date and time.
   - This selection, in the user's time zone, is sent back to a webhook under the function call `save_booking`.
   - The selected date and time are converted from the user's time zone back to IST.
   - The final booking is made in Google Calendar in IST.

Detailed Steps for Implementation

1. Check Availability Function

- Accepts:
  - User's preferred time zone
  - Start and end date for available slots (default: from today to the next two weeks)
- Fetches:

- ○ Free/busy slots from Google Calendar API in IST
- ● Converts:
  - ○ Available slots from IST to the user's requested time zone
- ● Sends:
  - ○ Formatted list of one-hour slots to Retell AI

2. Retell AI Interaction

- ● Receives available slots
- ● Selects 2-3 suitable options
- ● Presents options in a natural, conversational way to the user

3. Save Booking Function

- ● Accepts:
  - ○ User's selected date and time in their time zone
- ● Converts:
  - ○ Selected time back to IST
- ● Books:
  - ○ Finalized appointment in Google Calendar in IST

Additional Considerations

1. **Handling Edge Cases**
   - ○ Ensure correct conversion across daylight savings adjustments in U.S. time zones.
   - ○ Validate user-provided time zones before processing requests.
   - ○ Handle time conflicts by checking Google Calendar before confirming bookings.
2. **API/Webhook Integrations**
   - ○ **Google Calendar API** for fetching availability and finalizing bookings
   - ○ **Retell AI API** for conversational interactions and appointment selection
3. **Scalability & Future Enhancements**
   - ○ Support for additional time zones outside the U.S.
   - ○ Ability to integrate multiple calendars for better availability tracking.
   - ○ AI-based suggestion system to recommend optimal time slots based on user behavior.

Important Note

All functionality should be implemented using fully coded solutions. No-code or zero-code tools should not be used. Developers must create their own webhooks, process API calls independently, and handle all business logic programmatically.

Technologies Recommended

- **Backend:** Node.js (Express.js), Python (FastAPI, Flask, or Django)
- **Database:** PostgreSQL, MongoDB, MySQL
- **APIs:** Google Calendar API, Retell AI API
- **Time Zone Handling:** Moment.js (JavaScript), dateutil & pytz (Python)
- **Hosting & Deployment:** AWS Lambda, Google Cloud Functions, Docker, Kubernetes

---

Final Deliverables

- A backend service with two core functions:
  - `check_availability`: Fetches, converts, and sends available time slots
  - `save_booking`: Receives user selection, converts back to IST, and finalizes booking
- Seamless integration with Google Calendar and Retell AI
- Well-documented API/webhook endpoints for easy integration