AI-Powered Meeting Scheduler

Project Overview

The AI-Powered Meeting Scheduler is an intelligent calendar management system that combines natural language processing with Google Calendar integration to automate meeting scheduling. The system uses voice and text input to understand user requests and automatically finds available time slots, creates calendar events, and generates meeting links.

Introduction

In today's fast-paced business environment, scheduling meetings efficiently is crucial for productivity. Traditional calendar management requires manual checking of availability, coordinating with multiple participants, and creating calendar events. This project addresses these challenges by creating an AI agent that can understand natural language requests and automatically handle the entire meeting scheduling process.

The system acts as a personal scheduling assistant that can process requests like "Schedule a 30-minute meeting on Monday at 2 PM" or "I need a meeting before Friday at 10 AM" and handle the complete workflow from availability checking to calendar event creation.

Technology Stack

Core Technologies

- **Python 3.x** - Primary programming language
- **Google Calendar API** - Calendar integration and event management
- **OpenAI GPT-4** - Natural language processing and intent recognition
- **Speech Recognition** - Voice input processing
- **Text-to-Speech (pyttsx3)** - Voice output generation
- **Html** CSS java scripts for (Frontend)

Key Libraries and Dependencies

- **google-auth & google-auth-oauthlib** - OAuth2 authentication with Google services
- **googleapiclient** - Google API client library
- **openai** - OpenAI API integration
- **speech_recognition** - Audio input processing
- **pyttsx3** - Text-to-speech conversion
- **dateutil** - Advanced date/time parsing
- **datetime & calendar** - Date and time manipulation

External Services

- **Google Calendar API** - Calendar data access and event creation
- **OpenAI API** - Language model for natural language understanding

- **Google Cloud Console** - OAuth2 credentials management

Features

1. Natural Language Processing

- **Intelligent Request Parsing**: Understands complex meeting requests in natural language
- **Context Awareness**: Maintains conversation state across multiple interactions
- **Flexible Date Recognition**: Processes various date formats including day names, relative dates, and specific dates
- **Time Preference Handling**: Interprets different time formats and preferences

2. Voice Interface

- **Speech-to-Text**: Converts voice commands to text using Google Speech Recognition
- **Text-to-Speech**: Provides audio feedback and confirmations
- **Multimodal Input**: Supports both voice and text input modes
- **Interactive Conversations**: Maintains natural dialogue flow

3. Calendar Integration

- **Real-time Availability Checking**: Queries Google Calendar for busy/free time slots
- **Automatic Event Creation**: Creates calendar events with appropriate details
- **Google Meet Integration**: Automatically generates video conference links
- **Reminder Setup**: Configures email and popup reminders for scheduled meetings

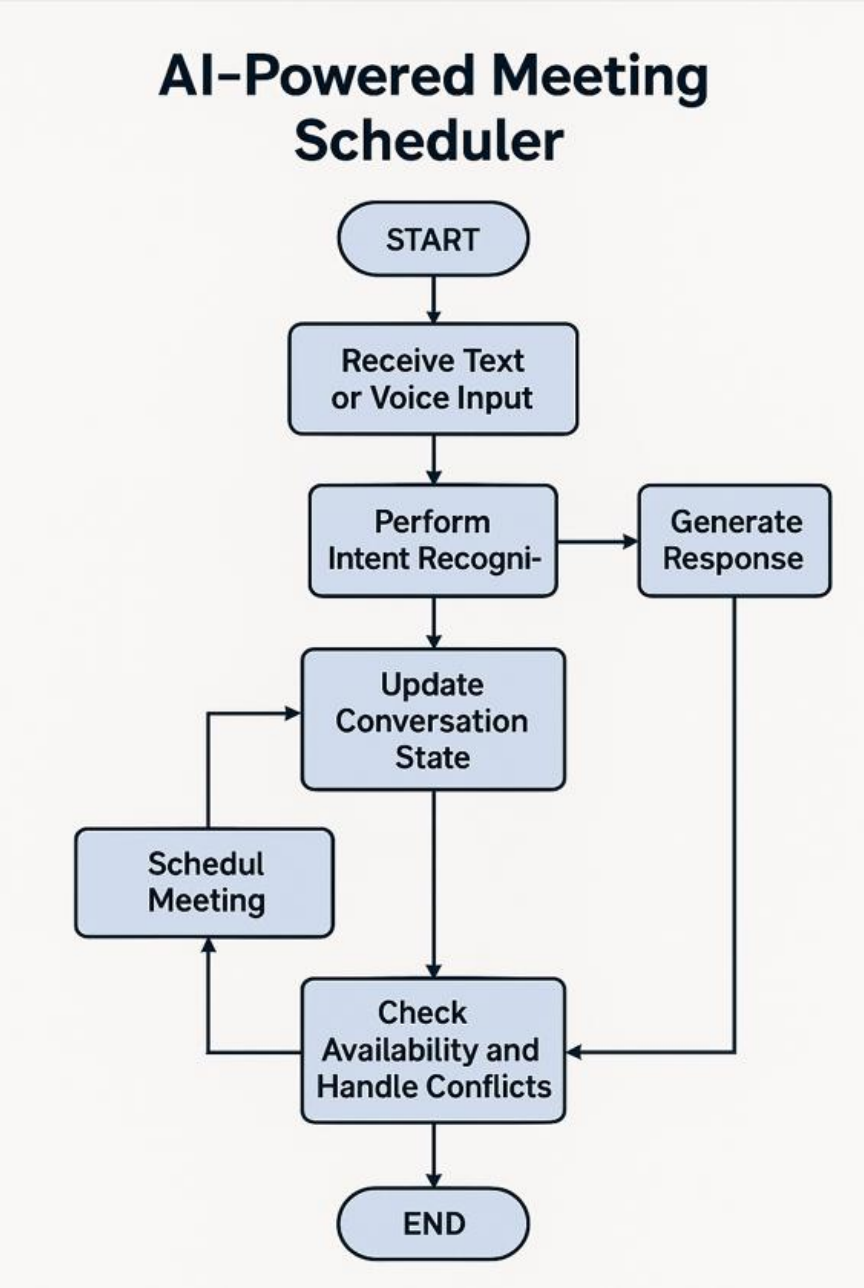
4. Smart Scheduling Logic

- **Conflict Detection**: Identifies scheduling conflicts and suggests alternatives
- **Deadline Awareness**: Respects user-specified deadlines for meeting scheduling
- **Multi-day Availability Search**: Searches across multiple days to find suitable slots
- **Duration-based Slot Finding**: Matches available time slots with required meeting duration

5. User Experience Features

- **Conversation State Management**: Remembers meeting requirements across interactions
- **Suggestion Engine**: Provides alternative dates when preferred times are unavailable
- **Greeting and Casual Conversation Handling**: Responds appropriately to greetings and casual inputs
- **Error Handling and Recovery**: Gracefully handles errors and guides users to successful scheduling

Design Choices and Architecture



1. Modular Architecture

The system is designed with clear separation of concerns across multiple modules:

- **google_calendar.py**: Handles all Google Calendar API interactions
- **llm.py**: Manages OpenAI API communication and response processing
- **voice.py**: Provides speech recognition and text-to-speech functionality
- **main.py**: Orchestrates the overall workflow and user interaction

This modular approach ensures maintainability, testability, and easy feature extension.

2. State Management Strategy

The system employs a persistent state accumulator (MEETING_STATE) that tracks:

- Meeting duration requirements
- Date preferences and deadlines
- Time preferences
- Request context (new request vs. date selection)

This design choice allows for natural, multi-turn conversations where users can provide information incrementally.

3. Natural Language Processing Pipeline

The NLP pipeline follows a structured approach:

1. **Intent Classification**: Determines if input is a greeting, new request, or date selection
2. **Information Extraction**: Uses GPT-4 to extract structured meeting details from natural language
3. **State Updates**: Intelligently merges new information with existing state
4. **Validation**: Ensures all required information is present and valid

4. Error Handling and User Guidance

The system implements comprehensive error handling:

- **Authentication Recovery**: Automatically refreshes expired tokens
- **API Error Management**: Handles Google Calendar API failures gracefully
- **User Input Validation**: Provides clear feedback for invalid or incomplete requests
- **Alternative Suggestions**: Offers alternatives when preferred times are unavailable

How the Agent Works

1. Initialization Phase

1. **Service Authentication**: Establishes connection with Google Calendar API using OAuth2
2. **Voice Engine Setup**: Initializes speech recognition and text-to-speech engines
3. **LLM Connection**: Establishes connection with OpenAI API

2. User Interaction Loop

1. **Input Processing**:
    - Accepts voice or text input from user
    - Converts speech to text if needed
2. **Intent Recognition**:

    - Classifies input as greeting, meeting request, or date selection
    - Extracts structured information using GPT-4
3. **State Management**:
    - Updates conversation state with new information
    - Maintains context across multiple interactions
4. **Business Logic Processing**:
    - Validates meeting requirements
    - Checks calendar availability
    - Handles conflicts and suggests alternatives
5. **Action Execution**:
    - Creates calendar events when all requirements are met
    - Provides feedback and confirmation to user

3. Calendar Operations Workflow

1. **Availability Query**: Uses Google Calendar freebusy API to check time slots
2. **Slot Analysis**: Compares busy periods with requested duration and time preferences
3. **Conflict Resolution**: Identifies overlaps and calculates free time slots
4. **Event Creation**: Creates calendar event with meeting details and Google Meet link
5. **Confirmation**: Provides user with meeting details and access links

4. Conversation Flow Management

The agent handles different conversation scenarios:

- **Complete Requests**: "Schedule a 30-minute meeting on Monday at 2 PM"
- **Incremental Requests**: User provides information across multiple turns
- **Date Selection**: User chooses from suggested available dates
- **Conflict Resolution**: Agent suggests alternatives when conflicts arise

5. Intelligence and Adaptability

The system demonstrates intelligence through:

- **Context Preservation**: Remembers user preferences throughout the conversation
- **Flexible Parsing**: Handles various date and time formats naturally
- **Proactive Suggestions**: Offers alternatives when initial preferences aren't available
- **Deadline Awareness**: Prioritizes suggestions based on user deadlines

Technical Implementation Highlights

OAuth2 Authentication Flow

The system implements a robust authentication mechanism that handles token refresh, expiration, and re-authentication automatically, ensuring seamless user experience.

Date and Time Processing

Advanced date parsing using dateutil library enables the system to understand relative dates ("next Monday"), day names ("Friday"), and various date formats.

API Integration Strategy

The integration with Google Calendar API and OpenAI API includes proper error handling, rate limiting consideration, and fallback mechanisms.

Voice Processing Pipeline

The voice interface uses Google's speech recognition service with proper audio preprocessing and error handling for accurate speech-to-text conversion.

Future Enhancement Opportunities

1. **Multi-user Support**: Extend to handle scheduling across multiple participants
2. **Integration Expansion**: Add support for other calendar platforms (Outlook, Apple Calendar)
3. **Advanced AI Features**: Implement meeting agenda generation and participant preparation
4. **Mobile Application**: Develop mobile app interface for enhanced accessibility
5. **Analytics Dashboard**: Add scheduling analytics and productivity insights

Conclusion

The AI-Powered Meeting Scheduler demonstrates the potential of combining natural language processing with calendar automation to create intelligent productivity tools. The system's modular architecture, robust error handling, and natural conversation flow make it a practical solution for automating meeting scheduling tasks while maintaining user-friendly interaction patterns.