# AI-Powered Scheduling Assistant

Team: SyncWave
Vinay: ullivinaybabu@gmail.com
Kushwanth Goutham: tkngoutham@live.com

The Problem

- ❌ Manual scheduling across time zones

- ❌ Multiple back-and-forth messages

- ❌ No intelligent conflict resolution

- ❌ Existing tools lack automation

- 💡 Need: Seamless, intelligent assistant that minimizes manual effort

Our AI Solution Architecture

- 🤖 LangGraph Agent Framework
- • React Agent with Tool Calling
- • Qwen2.5-14B-Instruct LLM
- • Structured Output Parsing

- 📅 Google Calendar Integration
- • OAuth2 Authentication
- • Real-time Calendar Scanning
- • Multi-user Event Retrieval

- ⚡ Intelligent Conflict Resolution
- • Priority-based Meeting Classification
- • Automatic Conflict Detection
- • Smart Rescheduling Logic

Key Features & Priority System

- 🔴 P0 - OUTAGE MEETINGS (CRITICAL)
- • Keywords: emergency, urgent, outage, crisis
- • Override ALL conflicts
- • Schedule immediately

- 🟡 P1 - CLIENT/CUSTOMER MEETINGS (HIGH)
- • Keywords: client, demo, feedback, requirements
- • Override internal meetings only
- • No weekends/off-hours

- 🟢 P2 - INTERNAL MEETINGS (STANDARD)
- • Keywords: team, sprint, planning, 1-on-1
- • Can be rescheduled for higher priority
- • Business hours only

Priority Resolution Matrix

- 📊 INTELLIGENT CONFLICT RESOLUTION RULES

- 🔴 OUTAGE vs ANYTHING = OVERRIDE
- • Emergency meetings take absolute priority
- • Can override weekends, off-hours, all meetings
- • Immediate scheduling within 30 minutes

- 🟡 CLIENT vs INTERNAL = OVERRIDE
- • Client meetings override internal meetings
- • Cannot override outages or external events
- • Business hours only

- 🟢 INTERNAL vs INTERNAL = DEFER
- • Internal meetings can be rescheduled
- • Respects working hours and availability
- • Flexible scheduling with 2-hour buffers

React Pattern: Why It's Revolutionary

- 🧠 REASONING + ACTING + OBSERVING

- 💭 **THINKING PHASE**
- • Analyzes meeting request context
- • Determines priority level (P0/P1/P2)
- • Plans required actions

- 🔧 **ACTING PHASE**
- • Calls calendar API to fetch events
- • Scans participant availability
- • Identifies conflicts and solutions

- 👁️ **OBSERVING PHASE**
- • Processes calendar data
- • Evaluates conflict resolution options
- • Refines strategy based on findings

Technical Implementation & Results

- 🛠 Core Technologies
- • LangChain + LangGraph
- • Google Calendar API v3
- • Pydantic for structured output
- • Local vLLM inference server

- 📊 Smart Features
- • Natural language meeting requests
- • Automatic timezone handling (IST)
- • Conflict detection & resolution
- • Event rescheduling with 2-hour buffer

- ⚡ Performance
- • Sub-second execution time
- • Real-time calendar scanning
- • JSON-structured output
- • Scalable multi-user support

Thank you