🛠️ Project: AI-Powered Calendar Assistant

📥 Backlog / Future Ideas

(*Not needed for MVP, but useful for V2+*)

- Energy-aware scheduling (e.g. low-energy tasks in afternoons)
- “Focus mode” toggle that blocks out uninterrupted time
- Natural language GPT-style command interface
- Integration with other calendars (Google, Outlook)
- Shared/team calendar planning
- AI-driven rescheduling suggestions (“Looks like you missed this — here’s when you could do it next”)
- Analytics dashboard (time spent, task completion rate, etc.)

🧪 Research & Planning

(*Start here*)

- Define user personas and scheduling pain points
- Research Apple Calendar integration options (CalDAV vs EventKit)
- Sketch basic UX wireframes for:
  - Task input form
  - Calendar view with AI-inserted items
- Review AWS services for hosting (EC2 vs Lambda for scheduler logic)
- Design initial data model for tasks, users, events
- Decide where you’ll store user data (PostgreSQL, DynamoDB, etc.)

🧱 Phase 1: Core Infrastructure & Setup

(*Foundation layer*)

- Set up AWS EC2 instance for backend/API

- Set up basic backend server (FastAPI or Flask)
- Set up database (PostgreSQL or DynamoDB)
- Create API endpoints for:
  - Add task
  - Get all tasks
  - Sync calendar data
  - Schedule tasks
- Write backend logic to store and retrieve user tasks

✅ Phase 2: Task Input System

(*MVP: ability to input tasks with metadata*)

- Create task input form (web or CLI)
- Support fields:
  - Task title
  - Priority (high/med/low)
  - Estimated duration
  - Deadline
  - Type (focus/admin/flex)
- Validate and store tasks in database
- Basic task list dashboard (table or simple UI)

📅 Phase 3: Apple Calendar Integration

(*Read & write events to user's calendar*)

- Authenticate user with Apple Calendar (iCloud / CalDAV / EventKit)
- Read user events and availability
- Identify free/busy slots
- Write custom events into Apple Calendar
- Mark AI-generated events as such (e.g., use a prefix or color)

🤖 Phase 4: Scheduling Engine (v1 - Rule Based)

(*Core intelligence for inserting tasks into calendar*)

- Analyze current calendar availability
- Build logic to prioritize tasks by:
    - Priority
    - Deadline urgency
    - Time of day preference
- Fill in tasks into free slots
- Insert new events into calendar
- Add buffer time between scheduled items
- Avoid scheduling in Do Not Disturb / blocked time

🔁 Phase 5: Rescheduling Logic

(*If a task is missed, find a new slot*)

- Detect if a scheduled task was skipped (past time, not marked done)
- Mark it as “incomplete”
- Attempt to find the next best time slot
- Push lower priority items if needed
- Notify user of rescheduled items

🔔 Phase 6: Notifications & Feedback Loop

(*Let users know what's coming, what's changed*)

- Daily or weekly email summary of tasks
- Alert when tasks are rescheduled
- Option to confirm/reject AI changes
- Mark tasks as complete/incomplete
- Capture feedback for future improvements

🧠 Phase 7: AI Enhancement (v2)

(*Smarter logic and personalization*)

- Add user preferences (e.g. preferred work hours, focus time)
- Weight slots based on time of day, energy level, etc.
- Start collecting data for ML (task completion, preferred times)

- Implement scoring system for optimal task placement
- Optional: Integrate GPT-based scheduling assistant (“Find 2 hours for writing this week”)

🚀 Launch / Testing

- Internal testing of scheduling and rescheduling
- Test Apple Calendar sync under different scenarios
- Add user onboarding flow
- Deploy to production
- Collect early user feedback
- Monitor AWS performance and optimize costs

🧰 Tools & Tech Notes (Side Cards)

- Apple Calendar integration: CalDAV docs, EventKit (iOS/Mac)
- AWS: Use EC2 for backend/API, S3 for logs, IAM roles for security
- Frontend: Could be React (for web), SwiftUI (if native Apple app)
- Calendar UI ideas: FullCalendar.js, React Big Calendar, or native iOS calendar views

---

Would you like a **pre-filled Trello or Notion template** you can copy and use? I can also generate a Gantt-style diagram next if you prefer that visualization.