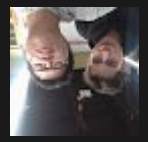
# CLERK

Computer Science Senior Capstone | MATES C/O 2026

# THE CREATORS



## MICHAEL THOMAS

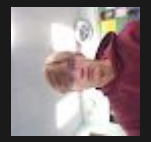
**Lead Coder / Full-Stack**

Architecting the multi-stage pipeline and logic feedback loops.

## JOHN SEDORIOSA

**Backend Specialist**

Managing SQLite databases, FastAPI integrations, and API normalization.



## CHRIS NOLAN

**Frontend Specialist**

Designing the React-based user interface for schedule visualization.

# The PROBLEM

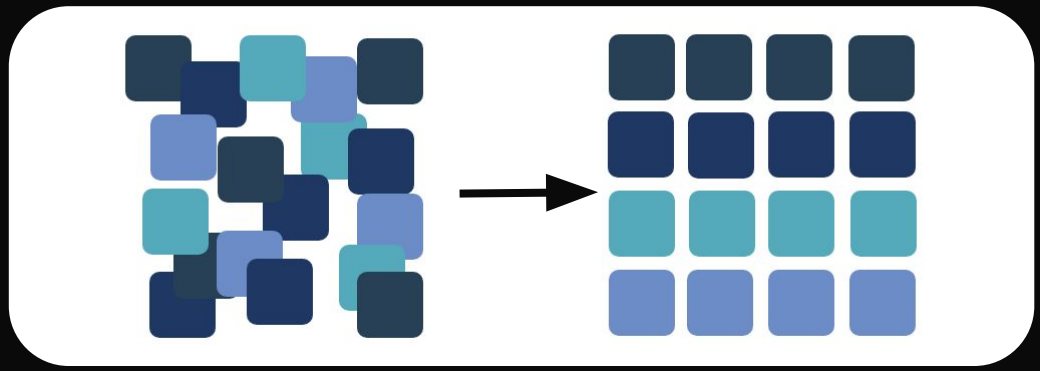
- Our daily lives generate vast amounts of unstructured data, voice notes, emails, and quick texts, that get lost because they require manual effort to organize
  - Fragmented notes/tasks on various formats
  - Users must manually set reminders and cross-reference availability
  - Traditional calendars don't learn user preferences or habit patterns

# OUR SOLUTION: CLERK

**A SEAMLESS PIPELINE**

Clerk uses a **multi-stage pipeline** with feedback loops to process raw data.

- **Ingestion:** Text and Audio
- **Normalization:** Converting all messy inputs into plain text.
- **Intelligence:** GPT-4o Mini extracts actionable JSON objects.

# CORE INTELLIGENCE

## TEMPORAL RESOLUTION

Resolves relative phrases like "tomorrow" or "by next Tuesday" into hard date-time stamps using current user context.

## CONFIDENCE SCORING

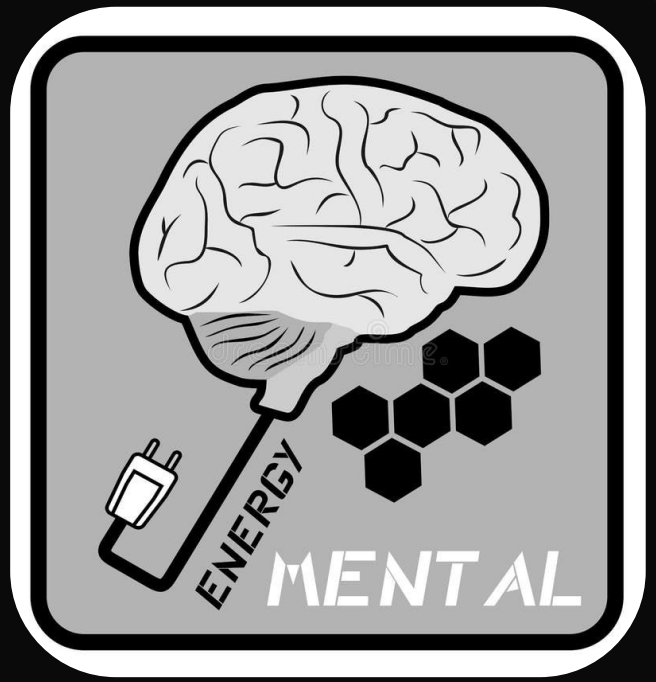
Calculates certainty based on language clarity. Vague items are flagged as "Inferred" for user review rather than guessed.

## ADAPTIVE BEHAVIOR

Uses statistical aggregation to learn user habits—like when you prefer to work—to improve future schedule suggestions.

# WHY IT MATTERS

- Automates translating notes  and voice notes into calendar events
- Learns user behavior and adapts to it, giving time to the user to focus on more important tasks
- Manual scheduling is prone to human error
- Clerk acts as a layer of verification ensuring user's schedule is accurate
- Saves mental energy



# TECHNICAL INFRASTRUCTURE

## BACKEND & APIS

- **Languages:** Python, SQL
- **Framework:** FastAPI, Pydantic
- **Intelligence:** GPT-4o Mini, Whisper
- **Database:** SQLite

## FRONTEND & DEVOPS

- **Web:** React, HTML5, CSS3
- **Deployment:** Render
- **Tools:** VS Code, GitHub
- **Integrations:** Gmail, Google Calendar

# ROAD MAP

### 01: Foundation

FastAPI setup, SQL Schema design, and environment initialization.

### 02: Extraction

Text paste interface, AI extraction layer with GPT-4o mini.

### 03: Logic

Confidence scoring system and converting relative phrases

### 04: Scheduling

Order tasks by urgency and identify available time slots, build dashboard to view, edit, and approve suggestions .

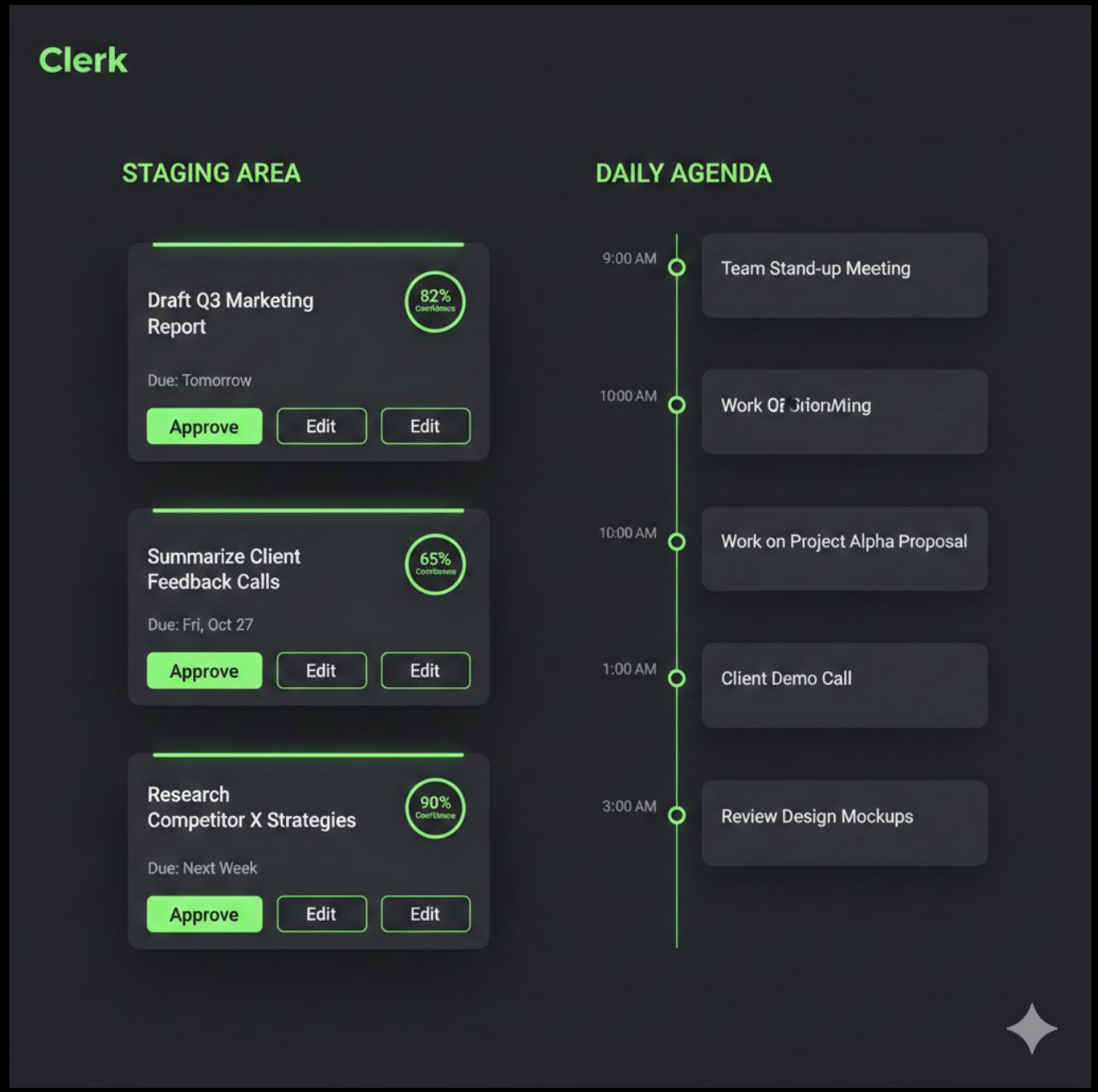
### 05: Audio

OpenAI whisper for audio note transcription and Media Recorder API for browser based recording .

### 06: Refinement

Implement light statistical aggregation to learn user behavior, and attempt Google Calendar API integration.

AI Image of what the Daily Agenda could look like



# PROJECT BOUNDARIES

## IN SCOPE

- Processing direct text inputs, emails, and audio files
- Using GPT-4o Mini to transform unstructured data into actionable JSON schemas
- Confidence based scoring system
- Web based UI for task approval and scheduling

## OUT OF SCOPE

- Fully autonomous scheduling without user review
- Negotiating meetings between user's calendars
- Interactive Grid Calendar
- Push notifications

# FEEDBACK?

Thank you for your attention.