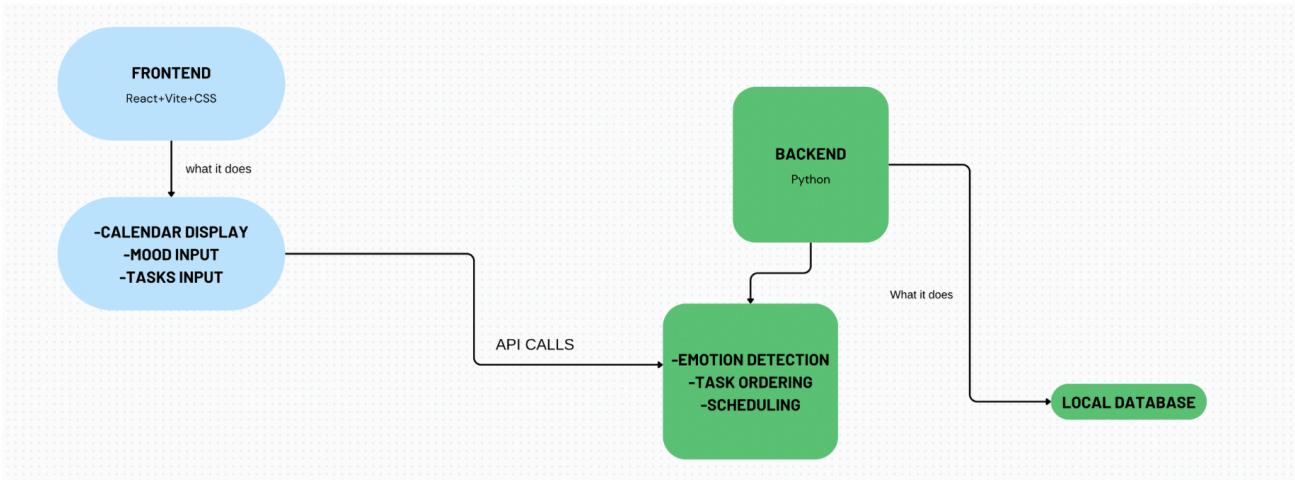


MINDSYNC

SYSTEM DESIGN  
**REPORT**

RAMANU RISHITA

# ARCHITECTURE DIAGRAM



## 3. Layer Breakdown

### Frontend Layer (React)

- Provides an intuitive, responsive interface for:
  - Mood description input
  - Adding/editing tasks
  - Choosing schedule settings (start time, break duration, type of calendar)
  - Viewing generated schedule in a dynamic calendar
- Uses **FullCalendar** in the preview section for schedule visualization
- Implements smooth UI animations + alternating layout design

### Backend Layer (FastAPI in Python)

- Handles API endpoints:
  - `/detect_mood` → returns detected emotion + confidence + emoji
  - `/generate_schedule` → produces a time-aware ordered schedule
- Emotion detection handled using a lightweight sentiment/emotion model
- Scheduling logic respects:

- Energy level mapping
- Task duration
- Break intervals
- Chronological start time
- CORS enabled for frontend integration

### **Data Layer (Lightweight Local Storage / Python Data Models)**

- Since MindSync runs client-side / session based, task & schedule data is stored in memory

## **5. Component Breakdown**

<b>Component</b>	<b>Location</b>	<b>Responsibility</b>
<b>Features.jsx</b>	<b>frontend/src/pages/</b>	Main workflow: mood input → task input → settings → generate schedule
<b>CalendarView.jsx</b>	<b>frontend/src/components/</b>	Renders calendar UI dynamically based on generated schedule
<b>backend_api.py</b>	<b>backend/</b>	Defines FastAPI endpoints for mood detection + schedule creation
<b>emotion_model.py</b>	<b>backend/</b>	Emotion classification logic
<b>planner.py / tasks_db.py</b>	<b>backend/</b>	Task arrangement & event time calculation

## 6. Chosen Technologies & Justification

Technology	Reason
<b>React</b>	Component-based architecture, fast rendering, easy UI animations & state handling
<b>Vite</b>	Faster dev build system than CRA, lightweight & production-ready
<b>FastAPI</b>	Modern Python backend framework, async, auto-docs, clean routing
<b>Python Emotion Model</b>	Easy integration with NLP tools, no external API costs, works offline
<b>FullCalendar.js</b>	Professional-grade calendar UI with drag, week/day/month view
<b>Local Storage / Lightweight Data</b>	No heavy DB needed for single-user task planning, reduces complexity