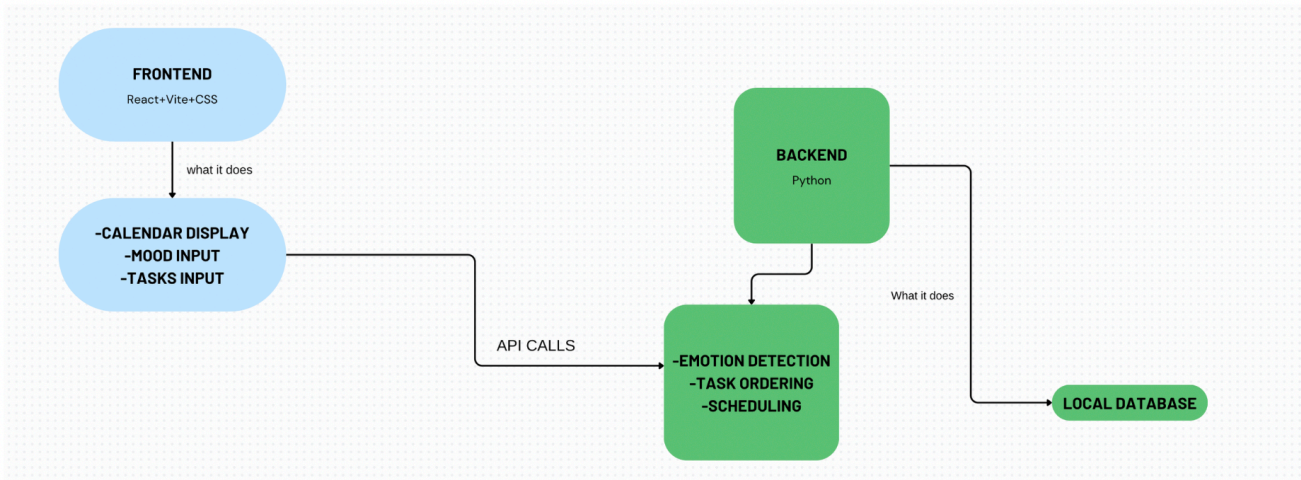


MINDSYNC

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

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

6. Chosen Technologies & Justification

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