# Mind Mate — Project Starter Specification for GitHub Copilot

This document is designed so you can paste it into VS Code and let **GitHub Copilot** generate the full project for you. It includes: - Full project definition - Functional requirements - Non-functional requirements - File/folder structure - Backend (Node.js) expectations - Frontend (React) expectations - Component descriptions - API specifications - Styling theme - Sample prompts for Copilot to expand - Additional tasks Copilot should complete

Paste this entire document into your VS Code workspace as `PROJECT_STARTER.md` or similar. Copilot will understand the architecture and begin scaffolding files as you start creating them.

---

# 1. PROJECT OVERVIEW

**Mind Mate** is an AI-powered wellness companion for workplace mood tracking, stress detection, wellness nudges, and empathetic conversational support.

This project consists of: - A **Node.js (Express)** backend - A **React frontend** - A minimal LLM integration (OpenAI) - A simple JSON-file data store (for hackathon simplicity)

The end goal is to have a working prototype where: - Users submit mood logs - They chat with an empathetic AI - Stress patterns are detected - Mood trends are visualized - Wellness summaries are generated - A breathing animation is accessible

---

# 2. HIGH-LEVEL REQUIREMENTS

## Functional Requirements

1. **Mood Logging**
2. User selects an emoji (😀🙂😐😟😢) and a stress level (0–5).
3. Data is stored in `backend/src/data/data.json`.

4. Backend returns last 10 entries.

5. **Empathetic Chat**

6. User sends a message.

7. Backend calls LLM wrapper → responds with <70 word supportive response.

8. **Stress Detection**

9. If 2 of last 3 stress entries ≥4 → trigger.

10. If mood drops ≥2 points across last 4 entries → trigger.

11. **Wellness Summary**

12. User requests summary.
13. Backend compiles last 12 entries.

14. LLM produces 4-part summary.

15. **Mood Trend Chart**

16. Use Chart.js in React.

17. Display last 10 entries.

18. **Breathing Animation**

19. A page/component contains the animated breathing exercise.

---

# 3. NON-FUNCTIONAL REQUIREMENTS

- Keep code simple and readable.
- No database required — JSON file storage only.
- UI must be calming and minimalistic.
- Empathy-focused tone.
- No medical or diagnostic language.
- Responses deterministic for demo.

---

# 4. RECOMMENDED FOLDER STRUCTURE

```
mind-mate/
├── PROJECT_STARTER.md
├── backend/
│   ├── package.json
│   ├── .env
│   └── src/
│       ├── server.js
│       ├── routes/
```

```
│       │   ├── mood.js
│       │   ├── chat.js
│       │   └── summary.js
│       ├── services/
│       │   ├── llm.js
│       │   └── stressDetector.js
│       ├── data/
│       │   └── data.json
│       └── utils/
├── frontend/
│   ├── package.json
│   └── src/
│       ├── App.jsx
│       ├── index.jsx
│       ├── components/
│       │   ├── ChatPanel.jsx
│       │   ├── MoodInput.jsx
│       │   ├── TrendChart.jsx
│       │   ├── SummaryPanel.jsx
│       │   ├── Avatar.jsx
│       │   └── BreathingExercise.jsx
│       └── styles/main.css
└── demo-assets/
    └── breathing-animation.html
```

---

# 5. BACKEND SPECIFICATION (Node.js + Express)

Copilot should generate the following files using these definitions.

## server.js

- Initialize Express
- CORS enabled
- Load routes: `/mood`, `/chat`, `/summary`
- JSON parser
- Port 4000

## Routes

### POST /mood

- Input: `{ mood: number, stress: number }`
- Append entry to JSON file

- Run stress detection
- Return detection + last 10 entries

### POST /chat

- Input: `{ message: string }`
- Uses OpenAI wrapper to generate empathetic response

### GET /summary

- Compiles last 12 entries
- Passes into summary LLM prompt

## Services

### stressDetector.js

Rules: - If 2 of last 3 entries have stress ≥4 → trigger - If mood drops ≥2 over last 4 entries → trigger

### llm.js

Methods: - `generateChatReply(message)` → empathetic <70 words - `generateSummary(trendValues)` → 4-part summary

---

# 6. FRONTEND SPECIFICATION (React)

Copilot should scaffold the components using these instructions.

## App.jsx behavior

- Layout: Left column (mood input + activity), center (chat), right (trend chart)
- State hooks for chat history, mood entries, summary data

## Components

### ChatPanel.jsx

- Chat history list
- Input box
- Sends POST `/chat`
- Shows response bubbles

### MoodInput.jsx

- Emoji selection

- Stress slider
- POST `/mood`

### TrendChart.jsx

- Line graph using Chart.js
- X-axis: timestamps
- Y-axis: mood scores

### SummaryPanel.jsx

- Shows summary text when user requests it

### BreathingExercise.jsx

- Uses the CSS animation from demo-assets

### Avatar.jsx

- Displays small friendly face icon

---

# 7. STYLING GUIDELINES

## Colors

- Primary Blue: `#6FA8F1`
- Teal: `#4FD1C5`
- Lavender: `#C6B9FF`
- Background Beige: `#FBFBF9`
- Gray Text: `#586069`

## Fonts

- Inter (body)
- Poppins (headers)

## CSS Notes

- Soft shadows: `0 4px 12px rgba(0,0,0,0.08)`
- Rounded corners: `border-radius: 12px`
- Use light gradients for cards

---

# 8. BREATHING ANIMATION (HTML/CSS)

Copilot should generate a React version of this.

```
<div class="circle">Breathe</div>
```

CSS:

```
.circle {
  animation: breathe 6s ease-in-out infinite;
  width: 120px;
  height: 120px;
  border-radius: 50%;
  background: #6FA8F1;
}

@keyframes breathe {
  0% { transform: scale(0.85); opacity: .8 }
  50% { transform: scale(1.15); opacity: 1 }
  100% { transform: scale(0.85); opacity: .8 }
}
```

---

# 9. COPILOT GENERATION PROMPTS

Paste these into VS Code alongside components to prompt Copilot effectively.

## Prompt: Generate Backend Route

```
// Copilot: create an Express route for POST /mood that:
// - Reads data.json
// - Appends new mood entry
// - Runs detectStress()
// - Returns entries and detection
```

## Prompt: Generate Empathetic Chat Handler

```
// Copilot: create a function generateChatReply(message) using OpenAI.
// Tone: gentle, empathetic, non-medical. Under 70 words.
```

### Prompt: Generate React Chat Component

```
// Copilot: build ChatPanel.jsx with:
// - chat bubbles
// - text input
// - POST /chat on send
// - auto-scroll to bottom
```

### Prompt: Generate Trend Chart

```
// Copilot: create TrendChart.jsx using Chart.js
// Input: mood entries array
// Output: line graph: timestamp vs mood
```

---

# 10. NEXT STEPS FOR COPILOT

- Generate React pages based on wireframes
- Generate CSS modules from style guide
- Scaffold backend files from folder structure
- Add placeholder responses if OpenAI key missing
- Add mock data to improve demo stability

---

# 11. FINAL NOTES FOR COPILOT

- Keep everything minimal and hackathon-ready
- Quick iteration > perfection
- Prioritize working demo path
- UI clarity and emotional tone are more important than features

---

This is the complete Job Spec for **Copilot**. Begin creating files according to the folder structure and follow behavior described above.

**End of Project Starter Document**