**Pilot Project: ChatGPT × Linear × Obsidian-**

**Style Knowledge Tool**

**Goal**

Build a MVP that feels like a mix of **ChatGPT, Linear, and Obsidian** — collaborative problem

solving/tracking for knowledge work.

The goal is not to replicate these tools fully but to show how fast you can ship and polish similar

functionality. Simply get as far as you can given the allotted time.

**Core Features**

1. **Omnibox Input**

o Single text field where user can type, submit entries, attach files/images, and

begin chatting with AI (similar to ChatGPT/Claude/Gemini/Grok omnibox).

2. **Problems and Other Objects**

o Main object is **Problems** (like Linear’s “issues”), each with status, priority, and

owner.

o Each new chat from the omnibox either creates a new problem or attaches to an

existing one.

o Problems can have multiple chats.

o Problems are named automatically as clearly and concisely as possible (user can

edit).

o Other objects:

▪ chats

▪ artifacts (files, images, etc.)

▪ conjectures (possible solutions to problems, text-based)

▪ criticisms (data, evidence, tests, or other refutations of conjectures, text-

based).

o All other objects must link to at least one problem. Criticisms must also link to a

conjecture.

3. **Dashboard + Knowledge Graph**

o **Dashboard:** Manage problems with a Linear-like dashboard and views.

o **Knowledge Graph:** Visualize problems and linked objects in an Obsidian-style

graph.

**Stack (suggested, flexible)**

• Frontend: Next.js (React + TypeScript)

• Backend: NestJS (TypeScript)

• Database: Postgres

• Deployment: Vercel + Railway/Render

Above is only a suggestion. Use whatever stack lets you work fastest.

**Deliverables**

1. Working deployed app (link).

2. GitHub repo with code.

3. README with:

o Setup instructions

o Design decisions

o AI tools used and most impactful prompts

**Evaluation Criteria**

• **Velocity:** Speed of shipping usable features.

• **AI-first workflow:** Effective AI use for speed and polish.

• **Code quality:** Structure, readability, maintainability.

• **Polish:** Deployment, clean UX, minimal bugs.

Color theme :

* **Landing Page Hero** → Gradient background (Indigo → Violet → Blue), white text
* **Buttons** → Primary = gradient; Secondary = white with gradient border
* **Sidebar** → Light gray (#F3F4F6) or Dark gray (#1F2937)
* **Problem Cards** → White card with colored status pill

Tailwind palette:

{

primary: {

from: '#6366F1', // Indigo

via: '#8B5CF6', // Violet

to: '#3B82F6' // Blue

},

background: '#F9FAFB',

text: '#111827',

muted: '#6B7280',

success: '#10B981',

warning: '#F59E0B',

danger: '#EF4444',

}

**Website Pages**

**(A) Landing Page (Website)**

**Goal**: Explain what the tool is, why it’s useful, and get users to try it.

**Sections**:

1. **Hero Section**
   * Logo + App Name (e.g., *Synapse*)
   * Tagline: *“AI-powered problem-solving meets connected knowledge.”*
   * CTA Button: **Try Now** / **Get Started Free**
   * Illustration: Graph nodes connected (Obsidian-style)
2. **Features Section**
   * **Omnibox** → “One box to ask, create, and solve”
   * **Problem Tracker** → “Organize issues like Linear, add priority, status and owner”
   * **Knowledge Graph** → “See how everything connects”
   * **AI Chat** → “Collaborate with AI directly in context”
   * **Profile** → “Add your socials, connect with people, add friends”
3. **Demo Screenshot / Animation**
   * Show sample dashboard with problems + chat + graph
4. **Testimonials / Use Cases** (optional MVP)
   * “Great for researchers, startups, and problem solvers.”
5. **Footer**
   * Links: GitHub, Docs, Contact, Twitter

**(B) App Dashboard (Core UX)**

Think **ChatGPT** + **Linear + Obsidian mashup**.

**Layout**:

* **Sidebar (left)**:
  + Problems
  + Conjectures
  + Criticisms
  + Artifacts
  + Knowledge Graph (button)
* **Topbar (center)**:
  + Omnibox search/command input
* **Main Panel (center)**:
  + List of Problems (cards with title, status, priority, owner)
  + Clicking a Problem → opens detail view (chat + related conjectures/criticisms)
* **Right Panel (optional)**:
  + Chat with AI
  + Suggestions / summaries

**(C) Problem Detail Page**

Like Linear’s “issue detail” + AI chat.

* Title (auto-generated, editable)
* Status / Priority dropdowns
* Owner assignment
* Tabs:
  + **Chat** (discussion with AI + teammates)
  + **Conjectures** (possible solutions)
  + **Criticisms** (linked to conjectures)
  + **Artifacts** (files, screenshots, images)

**(D) Knowledge Graph Page**

Like Obsidian’s graph view.

* Nodes: Problems (main)
* Linked Nodes: Conjectures, Criticisms, Artifacts
* Hover → shows preview
* Click → opens detail panel

Name of the tool/ website:Synapse

Design a complete website MVP based on these above given features and instructions, to give a complete, user-friendly MVP. Make sure you write the code based on the coding standards, maintain code readability, and velocity.