

FINAL PROJECT JULIA PODLIPENSKY - VIBE CODING CLASS

PROMPT DOCK

A MODULAR PROMPT BUILDING LIBRARY, RIGHT AT YOUR FINGERTIPS

A local desktop tool for creating, reusing,
and assembling AI prompts

Focused on speed, clarity, and everyday
usability

Designed for students and non-
developers, not prompt engineers



“EVERYTHING IS EVERYWHERE”

CURRENT SITUATION

Prompts are scattered across:
Notion, Word documents, Notes,
Figma Boards, Old chat histories,...

→ **Finding the “right” prompt breaks focus and flow**

ADDITIONAL FRICTION

- New tasks still require writing prompts from scratch
- Reusing roles, output formats or styles means copy-pasting from multiple sources
- Learning new prompting techniques creates even more fragments

RESULT

- **Inconvenience**
- **Lost time**
- **Inconsistent quality**
- **Frustration, especially for beginners**

TARGET GROUP



MOTIVATION

Observed in student discussions and group work:

- So many good templates, but scattered in word documents and not modular
- Reusing effective prompts or sentences mostly by long search through old chats

LEARNING ANGLE

Need a central place to:

- Save what works
- Reuse effective patterns
- Experiment safely and quickly

MY SOLUTION

A MODULAR PROMPT

LIBRARY

KEY IDEA

Prompts are built interactively from reusable “Lego” blocks:

- Actor
- Task
- Context Format
- Output Format
- Style / Constraints

- Fast prompt creation without starting from zero
- Consistent structure across different tasks
- Easy reuse and iteration

ACCESS

- Always available via a global keyboard shortcut
- Works regardless of browser or AI tool used

FROM HOTKEY TO PROMPT

ACCESS

User presses a global hotkey on their keyboard (mouse hotkey is OS sensitive)

POP UP

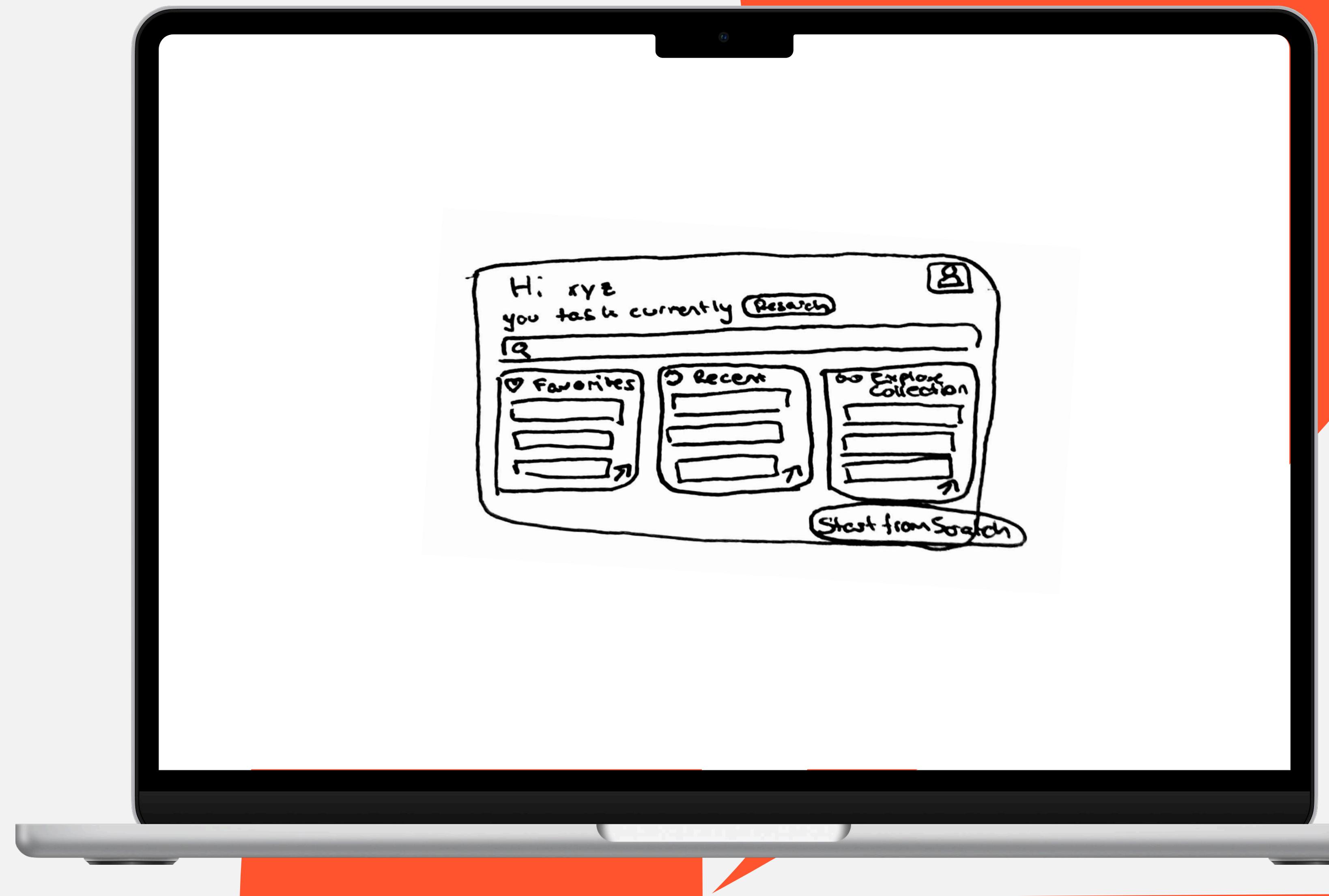
A small menu overlay opens instantly at your cursor

INVITE

User sees Current Task (e.g Coding, Research,...), Fuzzy Search, Favorites, Recent prompts, Collections

EDIT

User selects either a full prompt or starts building one modularly



BUILDING A PROMPT

PROMPT EDITOR

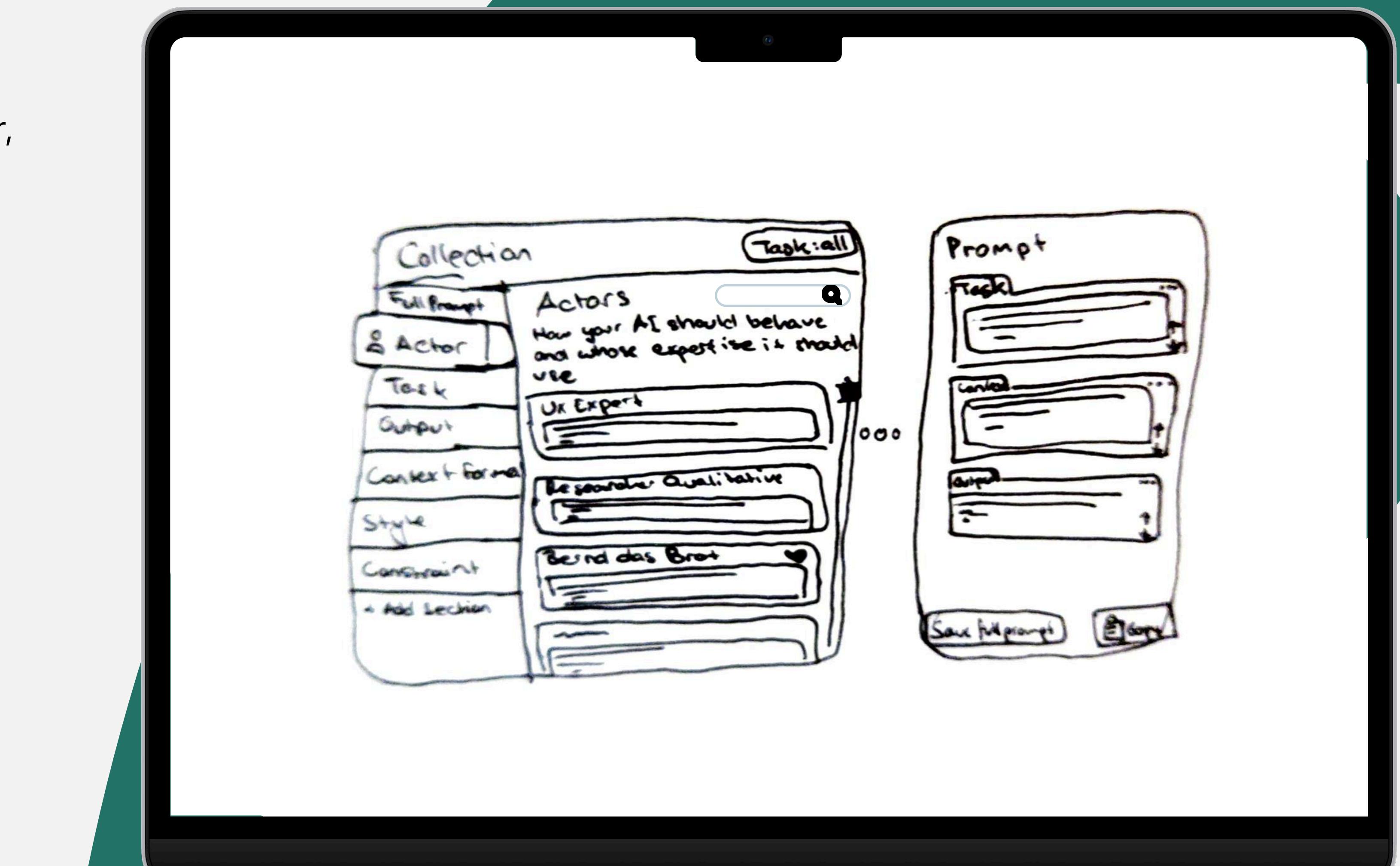
- Prompt is shown as editable blocks (Actor, Task, Output, Constraints,...)
- Each block has:
 - A clear headline
 - Editable text
 - Optional variables

INTERACTION

- Blocks can be:
 - Selected
 - Added
 - Removed
 - Reordered (via drag & drop & buttons)

FINISHING

- Live preview of the full prompt
- One-click “Copy to Clipboard”
- Paste into any AI tool



LOOK N FEEL

The screenshot shows the buildprompt.ai interface. On the left, a sidebar lists categories: Collection, Actors, Task, Output, Context / Format, Style, Constraint, and Add Section. The 'Actors' section is selected, displaying a card for 'UX Expert' with the sub-instruction: 'Act as a seasoned UX specialist with 10 years of experience...'. Below it is another card for 'Researcher Qualitative' with the same instruction. A third card, 'Bernd das Brot', is partially visible. At the bottom of the sidebar are buttons for '+ Add Section' and '+ Add actors'. The main area shows a 'Prompt' card for 'Bernd das Brot' with sections for Task, Context, and Output, along with Save full prompt and Copy buttons. A large orange callout highlights this card. At the bottom is a 'New Research Prompt' modal window containing a tree view of components like People, Situation, Development, etc., and a detailed description of the prompt for Henry Kravis.

VISUAL DIRECTION

- Clean
- Playful but focused
- Guided Tool-like, not “developer dashboard”

WHAT TO USE

- Cards and movable blocks
- Plain language
- Progressive disclosure (advanced options stay hidden)

REFERENCES

- Found and tried existing tool [buildprompts.ai](#) is functional, but feels too technical
- This new tool aims to be approachable for non-developers

TECH STACK & PLATTFORM

PLATFORM DECISION

Desktop app instead of browser extension, as you could trigger it reliably right from your fingertips in whichever browser or agent you are.

→ should work for: Windows, macOS

TECH STACK

UI: React

Desktop shell: Electron (seen it once before) or Tauri (more lightweight but never tried)

Search: Fuse.js

State: Zustand (lightweight global UI + builder state)

Storage: local JSON, optionally later SQLite

Native capabilities: global hotkey & clipboard, overlay window

PROJECT SCOPE

NOW IN SCOPE

- Global hotkey to open the app
- Modular prompt builder Interface
- Favorites, recents, collections
- Interactability and Adding
- Local-only storage
- Copy to clipboard of Prompt

LATER AND OUT

- Cloud sync for multiple devices and sharing prompt blocks (if there is time, this is the first I'll start with)
- Voice-based prompt building
- AI-Assisted prompt improvement
- Advanced context capture (Screen or computer-vision)