

Matt Nichols

Technical Deep-Dive: Good Wording

Overview & Stack

- **Good Wording:** building consumer software for introspective writing, and AI tools for deeper understanding
- Incorporated in late 2023 w/ co-founder Rishi
- We shipped an **alpha web app** for a handful of test users: <https://ink.goodword.ing>
- After ~6 months of full-time work, we decided to sunset Good Wording in mid-2024
- My ownership was primarily the **frontend**

Frontend

- SvelteKit
- TypeScript
- Tailwind
- TipTap
- Three.js
- shadcn-svelte

Backend

- Node
- TypeScript
- TSOA (Express)
- Freeplay (prompt eng)
- Supabase (Postgres)
- fly.io



Welcome Home

Ink is [Good Wording](#)'s foundational alpha app. This is a vibey space that can help you craft and sustain your intentional writing practice, and we're so glad you're here!

We care deeply about providing a comfortable and immersive space for deep work. All of your writing is encrypted and securely stored, and you can customize the app's appearance & behavior in [options](#).

Ink is an alpha app, so you may encounter bugs and unpolished details. If you have any feedback or suggestions, please [let us know!](#)

Oct 2 Capturing life through journaling, focusing on process
1:10 pm

Oct 2 This is a second demo entry!
1:09 pm

Oct 2 This is a demo entry!
1:09 pm

I am typing up a few little things here to demonstrate this app in dark mode; maybe we'll do something similar live, but I want to have a few images in place just in case. Let's try generating some text!

That's a great idea to have some images ready as a backup, just in case. As you continue exploring the app, I encourage you to use the journaling prompts or free-write to dive deeper into your thoughts and feelings. What aspects of your day or life are you hoping to capture through this journaling session? Reflecting on that could help provide a helpful starting point for generating meaningful text. Remember, the process is more important than the end result - allow yourself to be open and curious as you put pen to paper (or fingers to keyboard).

And maybe some bullet points

- ooh
- *very cool!*

Really what I'm hoping to do is just write out a little bit - just to make sure everything's online, and have a little bit more to show in an interview setting.

/

- Audio** Start an audio transcription
 - Prompt** Show prompts relevant to your writing
 - Review** Generate a review for this entry
 - Summon** Summon the AI to respond to your writing
- OPTIONS
- Fullscreen** Go big or go home
 - High Caps** Display everything in uppercase
 - Low Caps** Display everything in lowercase

- Jun 9** Pondering app tweaks, theme editor progress, and AI ambitions
9:47 am [Marriage and Relationships](#) • [Personal Projects](#) • [Product Lifecycle](#)
- Jun 7** Thinking through user settings vs editor settings vs document settings
9:17 am [Journaling](#) • [Self-Reflection Practices](#)
- Jun 6** Pondering gravity, sawdust, and the art of writing
8:50 pm [Audio Transcription](#) • [Journaling](#) • [Self-Reflection Practices](#)
- Jun 6** Exploring Tauri and iCloud sync for our multi-platform product
8:22 pm [Work](#)
- Jun 6** Pondering car purchase, craving new dimensions of growth
7:08 pm [Audio Transcription](#) • [Self-Awareness and Authenticity](#) • [Self-Reflection Practices](#)
- Jun 6** Rishi's busy weekend, now I seek human connection
10:39 am [Personal Projects](#) • [Supportive Friend](#) • [Journaling](#) • [Work](#) • [Self-Reflection Practices](#)
- Jun 3** Feeling Sick at Parents' House
6:41 pm [Personal Projects](#) • [Work](#)
- May 31** Productive week with frontend updates and theme editor ideas
6:32 pm [Work](#) • [Progress](#) • [Software Engineering](#)
- May 30** Focusing on the Present Amid a Busy Schedule
9:16 am [Marriage and Relationships](#) • [Personal Projects](#) • [Work](#) • [Software Engineering](#)
- May 28** Enjoying gaming with friends despite some anxieties
10:13 pm [Marriage and Relationships](#) • [Entrepreneurship Dynamics](#) • [Halo](#) • [Netflix](#)

Personal Projects

You have been exploring a variety of personal projects and ideas, including experimenting with visual AI, building a mobile web app, and working on your "Good Wording" startup with your co-founder Rishi. You've also been reflecting on your mindset and self-perception, trying to maintain a positive and optimistic outlook despite feeling overwhelmed at times. Additionally, you've been interested in the developments in artificial intelligence and its potential impact.

Entries

Updates

- Aug 8** Ok gotta real quick capture some ideas
6:48 pm [Personal Projects](#) • [Entrepreneurship Dynamics](#) • [Conversational AI Experience](#)
- Aug 5** Navigating career options, creative pursuits, and personal growth
9:56 am [Marriage and Relationships](#) • [Personal Projects](#) • [Product Lifecycle](#) • [Supportive Friend](#)
- Jul 22** Ran, shipped features, chatted with folks
10:14 pm [Marriage and Relationships](#) • [Personal Relationships and Life Milestones](#) • [Personal Projects](#)
- Jul 20** Dining delights, creative flow, and episode anticipation
11:25 pm [Marriage and Relationships](#) • [Personal Relationships and Life Milestones](#) • [Personal Projects](#)
- Jul 19** Adventurous day at Oakland museum with Laura
10:01 pm [Marriage and Relationships](#) • [Personal Relationships and Life Milestones](#) • [Personal Projects](#)
- Jul 18** Navigating the ups and downs of work and life
9:15 am [Personal Relationships and Life Milestones](#) • [Personal Projects](#) • [Supportive Friend](#)

☰ Options



Thank you for using our Alpha!

If you'd like to contribute feedback, no notes are too large or too small to share! Please use the [floaty feedback doodad](#), click the big button below, or press Cmd+Shift+F from anywhere in the app.

[Send Feedback](#)

Floaty Feedback Doodad



Configuration

Adjust settings to configure your Ink experience.

App Theme

Choose your preferred look and feel for the app.

Parchment ▾

Color Mode

Some themes support both light and dark mode, and can track your system settings.

Aquamarine

✓ Parchment

Astro [WIP]

Pizza [BAD]

Editor Typography

Choose the font style for the editor.

Auto-Scroll

Automatically scroll to keep your text cursor in the vertical center of the screen.



Spell Check

Use the system's spell check, auto-capitalize, and auto-completion features.



Visualizer Motion

Enable or disable animated graphics in Ink's background visualizer.



Enable Debug Features

See experimental settings, development tools, and assorted miscellany.



[Log Out](#)

Agenda

- Overview & Stack
- Screenshots
- Architecture Notes
- Document Model & UX
- Blocks & Jobs
- Visualizer
- Themes & User Config
- Improvements, Next Steps, Etc.
- Additional Discussion Topics

Architecture Notes

- Monorepo ✨
- **Frontend:** (owned by Matt)
 - SvelteKit file-system based routing + edge compute
 - State via Svelte stores; some remote initial values
 - Composite “blocks” for generated UI, inside & outside editor
 - Supabase used directly for auth (via SvelteKit request handling)
- **Backend:** (owned by Rishi)
 - Single production server instance (fly.io)
 - Supabase for data + auth + realtime updates
 - REST API to trigger/access async processes
 - “Jobs” framework for generated content

Document Model & UX

- **TipTap** is based on ProseMirror:
 - Extensions: add behavior, e.g. handle “on transaction” events
 - Commands: define actions that can be taken on an editor instance
 - Nodes: content within the editor, custom views, de/serialization, layout
- GW document format is series of HTML pages, each a serialized TipTap doc
- TipTap instance contained in a Svelte component, hydrated via a dynamic page route (SvelteKit)
- Editor supports “actions” (like Notion): type a slash, then an action keyword substring

Blocks & Jobs

- **Blocks** can render generated content within custom nodes in the editor, or more generally throughout the UI.
- Blocks are composed from sub-components:
 - Loading, header, text, quote, array, HTML, ...
 - We experimented w/ markdown sub-components for rich text, but moved to HTML to keep markdown rendering on the server, and to expand prototyping possibilities
- **Jobs** produce JSON content, including Blocks; “job manager” on the client enables subscriptions to job updates for live component refresh (via Supabase realtime)
- We can also subscribe to “nouns”, e.g. to pick up a changing title for a journal entry
- To-do: scaling considerations for realtime generative features

Visualizer

- A generalized system for building visualizers with parameter inputs (set by current theme config)
- FragShaderVisualizer renders ShaderToy-type generative art. Support for multi-pass rendering!
- VizCanvas, VizManager, VizRegistry (asset loading w/ Vite)

```
export abstract class Visualizer {  
    manager: VizManager;  
  
    /**  
     * Create a new visualizer.  
     * @param manager The manager that will render & control this visualizer.  
     */  
    constructor(manager: VizManager) {  
        this.manager = manager;  
    }  
  
    /**  
     * Render a single frame.  
     * Called by the manager, likely in a render loop.  
     * @param time The current time in milliseconds.  
     */  
    abstract render(time: number): void;  
  
    /**  
     * Set a parameter for the visualizer.  
     * @param name The name of the parameter to set.  
     * @param value The value to set the parameter to.  
     * @throws {Error} If the parameter name or value is invalid.  
     */  
    abstract setParam(name: string, value: VizParamType): void;  
  
    /**  
     * Clean up internal resources.  
     * Called by the manager, at the end of the visualizer's lifecycle.  
     */  
    abstract destroy(): void;  
  
    /**  
     * Resize internal resources.  
     * Called by the manager, after updating the canvas size.  
     */  
    abstract resize(size: [number, number]): void;  
}
```

Themes & User Config

- **Themes** are bundled as collections of design tokens (CSS vars), then selected with injected “data-theme” attribute as the request resolves
- Themes are also defined with a visualizer config: viz key + parameter values, which can be specific to light/dark modes, or always applied
- Typography CSS is also injected depending on the theme definition, using bundled CSS files, making sure we load only the fonts we need
- User Config is its own table in our postgres instance, keyed by user. We access this for each request, but cache it for reuse (via cookies)

Improvements, Next Steps, Etc.

- **Product directions:**
 - Daily ritual + structured journaling
 - Create & edit templates
 - Theme proliferation + marketplace
- Local-first & offline support (PWA and/or mobile)
- Improved mobile modality: audio emphasis!
- Streaming updates everywhere
- Intelligence: quality heuristics & test harness, avoid regenerating artifacts
- More tests, more polish, more shared models (or GraphQL) ...

Additional Discussion Topics

- **Voice features:** transcriptions, cleanup
- **Live-chat NUX:** find a daily question
- **Intelligence design:** entities, topics
- **Editor features:** auto-scroll, other UX
- **Editor actions:** sorting, grouping, etc
- **Deeper SvelteKit Discussion** (vs React)
- **Templates & flows:** multi-page entries
- **shadcn-svelte, tailwind,** other styling

Thank You!