Promptly

1. Project Naming & Branding

App Name: Promptly

Tagline: Your micro-reminder, instantly.

This name and tagline are short, memorable, and directly communicate the app's core purpose. "Promptly" implies both speed and immediacy, which are central to the app's value proposition.

---

2. Technology Stack & Configuration

The technology stack for this project is designed for maximum simplicity and zero cost, making it perfect for a solo developer.

- **Frontend:**
  - **HTML:** The core structure of the single web page.
  - **CSS:** A minimal stylesheet to create a clean, modern look. The UI should be dominated by a single, large button.
  - **JavaScript:** The entire logic of the application will be written in vanilla JavaScript. There is no need for any frameworks (like React or Vue.js).
- **APIs:**
  - **Web Speech API:** This is a native browser API that provides the voice-to-text functionality. It's free and readily available in all modern browsers. You'll use it to capture the user's spoken reminder.
  - **Web Notifications API:** This API is used to display the final reminder notification to the user, even if the browser is minimized or the user has switched to another tab.
- **Hosting:**
  - **GitHub Pages:** A free and easy way to host a static HTML, CSS, and JS file. You simply push your code to a GitHub repository, and it's live on the web in minutes. This completely eliminates any server or hosting costs.
- **No Backend:** As previously stated, there is absolutely no need for a server, database, or API keys. All of the application's logic runs directly in the user's browser.

---

3. Core Features & Logic

Your project will have a single user flow with the following steps.

1. **Start State:** The user sees a single button on a screen. The button's text is something like, "Tap and Speak Your Reminder."
2. **Voice Input:** The user taps the button. A JavaScript function is triggered to access the Web Speech API. The browser's native microphone interface appears.
3. **Reminder Processing:** The user says their reminder (e.g., "Take out the trash"). The Web Speech API listens for the input and returns the text string "Take out the trash."
4. **Timer Activation:** The JavaScript code takes the captured text and immediately starts a timer using `setTimeout`. The duration is set to a fixed 180,000 milliseconds (3 minutes).
5. **Notification Display:** After exactly 3 minutes, the `setTimeout` function fires. It triggers a function that uses the Web Notifications API to display a notification on the user's screen with the text they spoke earlier. The notification will simply say, "Promptly: Take out the trash."

4. Implementation Steps (Simplified)

1. **Create the File:** Start with a single `index.html` file. Include a basic `button` element and an empty `<script>` tag.
2. **Add CSS:** Use a `<style>` tag within the HTML or a separate `.css` file to make the button large and centered for a clean user experience.
3. **Write the JavaScript:**
    - Declare a variable for the button element.
    - Add an `onclick` event listener to the button.
    - Inside the event listener, write the code to use the `webkitSpeechRecognition` object (the Web Speech API) to start listening.
    - Implement the `onresult` event handler for the speech recognition object to capture the spoken text.
    - Use `setTimeout` to set the 3-minute timer.
    - Inside the `setTimeout` callback function, use `Notification.requestPermission()` and then `new Notification()` to display the reminder.
4. **Test and Deploy:** Open the `index.html` file in your browser to test it locally. Once it's working, upload the file to a GitHub repository and enable GitHub Pages in the settings. Your app is now live!

Phase 1: Problem Understanding & Industry Analysis

Goal: Understand what we're building and why.

---

1. Requirement Gathering

The core problem we are solving is the "ephemeral task" – a micro-task that is too fleeting for a traditional to-do list app, but too important to forget. The current process for capturing these tasks is high-friction, involving unlocking a phone, finding a specific app, and typing or navigating menus. This often leads to the thought being lost to distraction.

Our solution is a single-purpose, "digital sticky note" that serves a single reminder, then disappears. The key requirements are:

- **Frictionless Capture:** Capture a thought with a single tap and voice command.
- **Time-Sensitive:** The reminder should be triggered after a fixed, short duration (e.g., 3 minutes).
- **Zero-Overhead:** The app should require no setup, no account creation, and no data storage on a server.

2. Stakeholder Analysis

As a solo developer, I am the sole stakeholder, but I represent the end user.

- **Admin (the developer):** Primary need is simplicity and ease of implementation. The chosen stack of HTML, CSS, and JavaScript with browser APIs meets this need perfectly.
- **End User:** The user wants an incredibly simple, fast, and reliable tool. They need a solution that works without cognitive load and doesn't get in the way of their primary task.

3. Business Process Mapping

The user flow is a simple, linear process with a clear beginning and end.

- **Customer requests reminder:** User has a fleeting thought (e.g., "Take the laundry out").
- **App listens for input:** The user taps the button. The app automatically starts listening for voice input.
- **App sets timer:** The app captures the reminder text and starts a 3-minute countdown.
- **Notification is triggered:** After 3 minutes, a notification appears on the user's screen with the reminder. The process is complete.

4. Industry-Specific Use Case Analysis

While this project is simple, it's solving a real-world problem. Traditional to-do apps are designed for long-term planning, not for in-the-moment tasks. The user doesn't want to add "Take out the laundry" to a long list of other items. They want an immediate, temporary reminder that serves its purpose and then vanishes, leaving no digital clutter.

This is a use case that traditional apps fail to address, as they are built for persistent data and complex organization. The "Promptly" app is a pure solution for this specific user need.

5. AppExchange Exploration

There are no existing AppExchange solutions that fit this specific, simple use case. While a "reminders" app exists in various forms, none are built with the unique, zero-friction, voice-only approach of "Promptly." This confirms the app's unique position in the market.