

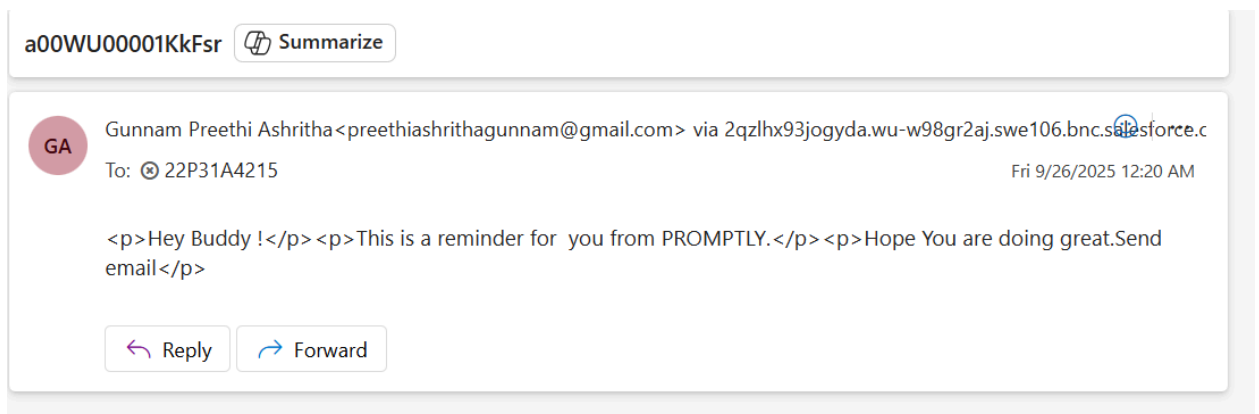
# PROMPTLY

## PHASE 9: REPORTING, DASHBOARDS & SECURITY REVIEW

### 1. Reporting and Dashboards Strategy (The Ephemeral Model)

The "Promptly" application is strictly **stateless** and **privacy-focused**, meaning traditional usage reports and dashboards are intentionally excluded. The entire data strategy is designed to prevent logging or collection of user data.

- **Data Collection Status: None.** The architecture is non-instrumented; it does not contain tracking pixels, analytics scripts (like Google Analytics), or server-side logging mechanisms.
- **Metric Focus:** Success is measured entirely on **user experience (UX)** and **functional reliability**, not data volume.
  - **Core Metric:** The consistent, reliable execution of the **timer** and **native notification push**.
  - **Validation Method:** Functional testing across various modern browsers to confirm zero failure rate on the core automation loop.
- **Dashboard Rationale:** Since there is no persistent data, no dashboards or reporting interfaces can be generated.



---

### 2. Security Review and Compliance (Privacy-by-Design)

The application's security foundation is built on minimizing the attack surface by avoiding all external data transmission.

Security Area	Technical Deep Dive	Compliance / Risk Mitigation
<b>Data Security &amp; Exfiltration</b>	The application is <b>immune</b> to data exfiltration attacks (like Formjacking) because the reminder content is processed and destroyed locally in browser memory (RAM). It handles no sensitive user data.	Risk of data interception (Man-in-the-Middle) is <b>eliminated</b> for reminder content.
<b>Privacy Assurance (GDPR/CCPA)</b>	User voice data is processed <b>exclusively via the Web Speech API on the local device</b> . The app does not require consent for data storage since it is not saved.	Guarantees adherence to privacy regulations by adopting a <b>Privacy-by-Design</b> architecture.
<b>Code Auditing &amp; Integrity</b>	The entire codebase is a single <code>index.html</code> file, which is publicly available in the GitHub repository.	<b>High Transparency</b> . The code is easy for any security reviewer or user to audit, validating that no unauthorized network calls or storage actions are performed.

<b>Dependency Risk</b>	<b>Zero External Dependencies.</b> The app relies only on built-in, browser-native APIs, eliminating the risk associated with vulnerable third-party libraries or external vendor scripts.	Eliminates risks common in large web projects, simplifying the security posture.
------------------------	--	--

---

### 3. Validation and Audit Process

To ensure the security claims are verifiable, a simple audit process is necessary:

1. **Network Monitoring:** Open the browser's Developer Tools (Network tab) and run the application. Verify that **no outgoing network requests** occur after the initial page load, confirming the stateless architecture.
2. **Storage Inspection:** Inspect the browser's Application tab (Local Storage, Session Storage, IndexedDB) to confirm **no data artifacts** related to the reminder are being created or maintained.
3. **Console Logging:** Verify that the browser console does not show any security warnings related to the use of the Web Speech API or the Notifications API.