Trial: Extension Design Document

1. Executive Summary

Project Name: Trial (formerly GhostPass)
Target Platform: Google Chrome (Manifest V3)
Core Value: Privacy, Speed, and Ephemerality.
Mission: "Your free trial for the web." To allow users to generate disposable identities and functional temporary emails instantly, bypassing invasive sign-up forms without compromising their real personal data.

2. User Stories & Core Features

Feature	**User Story**	**Priority**
Identity Generation	As a user, I want to click one button to generate a realistic fake name, username, and password so I don't have to think of one.	**P0 (MVP)**
Disposable Email	As a user, I want a working email address that can actually receive verification codes.	**P0 (MVP)**
Form Autofill	As a user, I want to right-click an input field and select "Fill with Trial" to populate the form automatically.	**P0 (MVP)**
Smart Inbox	As a user, I want to see incoming emails inside the extension popup without leaving the current tab.	**P1**
OTP Extraction	As a user, I want the extension to find the verification code in the email and let me copy it with one click.	**P1 (Wow Factor)**
Input Injection	As a user, I want a small "Trial Icon" (T) to appear automatically inside email inputs for faster access.	**P2 (Bonus)**

3. Technical Architecture (Manifest V3)

3.1 Component Overview

The extension follows the strict **Manifest V3** architecture, separating concerns between the UI, the Background Service, and the Webpage Context.

- **Popup (UI):** The control center. It displays the current identity, the inbox, and the controls to regenerate data.
    - *Style:* Industrial Minimalist (Geist Mono).
    - *Behavior:* Delegates all API calls to the Service Worker.
- **Service Worker (background.js):** The brain.
    - Handles all HTTP requests to 1secmail.com.
    - Manages the "Inbox Polling" alarm (checks for new mail every 10s when active).
    - Listens for Context Menu clicks.
- **Content Script (content.js):** The hands.
    - Runs on <all_urls>.
    - Listens for messages from the Service Worker to fill inputs.
    - (Phase 2) Injects the "Trial Icon" into DOM elements using Shadow DOM to prevent style leakage.
- **Storage (chrome.storage.local):** The memory.
    - Persists the current "Active Identity" so it survives browser restarts.

3.2 Data Flow Diagram

1. **Generation:** User clicks "New Identity" (Popup) $\rightarrow$ Message to Service Worker $\rightarrow$ SW calls API $\rightarrow$ SW saves to Storage $\rightarrow$ Popup updates UI.
2. **Autofill:** User Right-Clicks (Page) $\rightarrow$ Context Menu Event (SW) $\rightarrow$ SW reads Storage $\rightarrow$ Message to Content Script $\rightarrow$ DOM Manipulation.
3. **Email Check:** Alarm triggers (SW) $\rightarrow$ SW calls API $\rightarrow$ New Mail found? $\rightarrow$ Badge Text Update / Popup Notification.

4. API Specification (1secmail)

We utilize the **1secmail** free API. No API key is required.

4.1 Base URL

https://www.1secmail.com/api/v1/

4.2 Endpoints Used

Action	**Method**	**Endpoint Structure**	**Purpose**
Generate Email	GET	?action=genRandomM	Get a random address

		ailbox&count=1	(e.g., x7s8d@1secmail.com).
Check Inbox	GET	?action=getMessages&login={user}&domain={domain}	Get list of emails (ID, Subject, Sender).
Read Email	GET	?action=readMessage&login={user}&domain={domain}&id={id}	Get full HTML body of a specific email.

5. Implementation Details

5.1 Storage Schema (chrome.storage.local)

```
{
  "currentIdentity": {
    "firstName": "John",
    "lastName": "Doe",
    "username": "johndoe99",
    "password": "SecurePassword123!",
    "email": "x7s8d@1secmail.com",
    "login": "x7s8d",
    "domain": "1secmail.com",
    "created_at": 1715000000
  },
  "settings": {
    "theme": "industrial",
    "autoCopy": true
  }
}
```

5.2 The Smart OTP Regex

To extract verification codes from email bodies, we use this specific Regex pattern in src/utils/parser.js:

```
// Matches 4 to 8 digits that are isolated (not part of a phone number or date)
const otpRegex = /(?<!\d)\d{4,8}(?!\d)/;
```

6. UI/UX Design

6.1 Visual Language: "Industrial Minimalist"

We adopt a raw, functional aesthetic similar to Vercel/Next.js or linear command lines.

- **Typography: Geist Mono** (or generic monospace fallback). All text is treated as data.
- **Color Palette:**
  - Background: #000000 (Pure Black)
  - Surface: #111111 (Dark Gray)
  - Text: #EDEDED (Off-white)
  - Accent: #333333 (Borders/Separators)
  - Action: #FFFFFF (Buttons - Inverted high contrast)
  - Success: #00FF94 (Terminal Green - for "Copied" or "Verified" states)
- **Components:** Sharp corners (border-radius: 0px or 4px), 1px borders, subtle noise textures.

6.2 The Popup Layout

- **Header:** "TRIAL" (Bold, Monospace, Tracking +2).
- **Data Grid:**
  - Displays Identity fields as a key-value list (like a JSON object).
  - Hovering a row inverts the colors to indicate "Copyable".
- **Inbox Section:**
  - A simplified list.
  - When an OTP is detected, it renders as a large, green, glitch-effect number block.

6.3 The Context Menu

- **Label:** "Trial: Auto-fill Form"
- **Behavior:** Intelligent detection of name, email, password attributes.

7. Security & Privacy

7.1 Permissions Justification

- storage: To save the temporary identity during the session.
- scripting: Required to insert values into web forms securely.
- contextMenus: To provide the "Right-click to fill" functionality.
- host_permissions: Strictly limited to https://www.1secmail.com/* to fetch emails.

7.2 Data Handling

- **No Remote Telemetry:** The extension does not track user activity.
- **Local Only:** All identity data is stored locally in the browser.
- **Ephemeral:** Data can be wiped by the user or regenerated instantly.

8. Development Roadmap & Git Strategy

To maximize points for methodology, the development will follow this branch strategy:

Phase 1: The Core (Branch: feature/core-logic)

- [ ] Setup Manifest V3.
- [ ] Implement generator.js (Random names/passwords).
- [ ] Implement storage handling.

Phase 2: API Integration (Branch: feature/email-api)

- [ ] Implement service-worker.js fetch handlers.
- [ ] Connect "Generate" button to 1secmail API.
- [ ] Implement Inbox polling.

Phase 3: DOM Interaction (Branch: feature/autofill)

- [ ] Add Context Menu listener.
- [ ] Specific logic to detect <input> types and fill correctly.
- [ ] React/Vue event dispatching (fixing the "empty value" bug).

Phase 4: Polish (Branch: feature/ui-polish)

- [ ] Import Geist Mono font.
- [ ] Apply "Industrial" CSS (Grid layouts, 1px borders).
- [ ] The "Smart OTP" display.