Konzeptdokument: Schnupfspruch-Website

1. Projektübersicht

Ziel dieses Projekts ist die Erstellung einer minimalistischen und schnellen Website, die zufällige Schnupfsprüche anzeigt. Nutzer sollen zudem die Möglichkeit haben, eigene Sprüche einzureichen. Die Seite soll durch ihr klares Design und ihre einfache Bedienbarkeit überzeugen.

2. Zielgruppe

Die Zielgruppe umfasst Personen, die an Schnupftabak und den damit verbundenen Traditionen und Sprüchen interessiert sind und eine schnelle und unkomplizierte Unterhaltung suchen.

3. Design und User Experience (UX)

- **Clean und Simpel:** Das Design soll bewusst reduziert und übersichtlich sein. Keine ablenkenden Elemente.
- **Fokus auf den Inhalt:** Der Schnupfspruch steht im Mittelpunkt.
- **Mobile-First Ansatz:** Die Seite muss auf allen Endgeräten (Desktop, Tablet, Smartphone) optimal dargestellt werden und bedienbar sein.
- **Schnelle Ladezeiten:** Optimierung für Geschwindigkeit ist ein Kernaspekt.

4. Kernfunktionen

4.1. Startseite (Homepage)

- **Layout:**
    - Minimalistisches Design.
    - Zentrierter Button mit der Aufschrift "**Priis**".
    - Keine weiteren sichtbaren Elemente initial.
- **Funktionalität "Priis"-Button:**
    - Bei Klick auf den "Priis"-Button erscheint oberhalb des Buttons ein Textfeld.
    - In diesem Textfeld wird ein zufällig ausgewählter Schnupfspruch aus der Datenbank angezeigt.
    - Jeder Klick auf den "Priis"-Button lädt einen neuen, zufälligen Spruch.
    - Das Textfeld mit dem Spruch bleibt sichtbar, bis ein neuer Spruch geladen wird oder die Seite neu geladen wird.
- **Button "Spruch einreichen":**
    - Ein weiterer, klar erkennbarer Button (z.B. unterhalb des "Priis"-Buttons oder in einer dezenten Navigationsleiste) mit der Aufschrift "**Spruch einreichen**".
    - Bei Klick auf diesen Button wird der Nutzer auf eine separate Seite zur Sprucheinreichung weitergeleitet.

4.2. Seite "Spruch einreichen"

- **Layout:**
    - Einfaches und klares Formular.
    - Titel (z.B. "Reiche deinen Schnupfspruch ein").
    - Kurze Anleitung oder Hinweis zum Einreichen.
- **Formularfelder:**
    - **Textfeld für den Schnupfspruch:** Mehrzeiliges Textfeld (Textarea) zur Eingabe des Spruchs.
    - **(Optional) Feld für Name/Nickname:** Einzeiliges Textfeld, um dem Einreichenden die Möglichkeit zu geben, einen Namen anzugeben (kein Pflichtfeld).
    - **(Optional) Captcha/Spamschutz:** Um Spam-Einreichungen zu minimieren.
- **Button "Senden" / "Einreichen":**
    - Button zum Absenden des Formulars.
- **Funktionalität:**
    - Nach Absenden des Formulars werden die Daten an das Backend übermittelt.
    - Der Nutzer erhält eine Bestätigungsnachricht (z.B. "Vielen Dank, dein Spruch wurde eingereicht und wird geprüft.")
    - Die eingereichten Sprüche werden in der Datenbank gespeichert, idealerweise mit einem Status "zur Prüfung", um eine redaktionelle Kontrolle vor der Veröffentlichung zu ermöglichen.

5. Technische Anforderungen

5.1. Frontend

- **Technologie:** Empfehlung für ein leichtgewichtiges Framework oder Vanilla JavaScript, um die Ladezeiten gering zu halten (z.B. Preact, Svelte oder optimiertes React/Vue). Statische Seitengenerierung (SSG) oder serverseitiges Rendering (SSR) für die initiale Ansicht könnten ebenfalls zur Geschwindigkeit beitragen.
- **Styling:** CSS (ggf. mit einem schlanken Framework wie Tailwind CSS Utility-First oder custom CSS).
- **Responsivität:** Die Seite muss auf allen Bildschirmgrößen gut aussehen und funktionieren.

5.2. Backend

- **Datenbank:**
    - Effiziente Speicherung von über 2000 Sprüchen (und zukünftig mehr).
    - Schneller, zufälliger Zugriff auf die Sprüche.

- **Programmiersprache:** Wahl der Programmiersprache für das Backend ist flexibel (z.B. Node.js, Python (Flask/Django), PHP (Laravel/Symfony), Go, Ruby on Rails). Wichtig sind Effizienz und Skalierbarkeit.
- **Spruchverwaltung (Admin-Bereich - optional, aber empfohlen):**
    - Ein einfacher Admin-Bereich zur Verwaltung der Sprüche (Ansehen, Bearbeiten, Löschen, Freischalten von eingereichten Sprüchen) wäre sinnvoll. Dies muss nicht Teil der initialen öffentlichen Seite sein, aber die Backend-Struktur sollte es ermöglichen.

5.3. Performance und Skalierbarkeit

- **Schnelle Datenbankabfragen:** Optimierung der Datenbankstruktur und Abfragen für schnellen, zufälligen Zugriff.
- **Caching:** Einsatz von Caching-Mechanismen sowohl serverseitig (für Datenbankabfragen) als auch clientseitig (Browser-Caching für statische Assets).
- **Minimierung von Requests:** Reduzierung der Anzahl von HTTP-Requests.
- **Code-Splitting und Lazy Loading:** Laden von JavaScript und anderen Ressourcen nur dann, wenn sie benötigt werden.
- **Optimierung von Bildern und Assets:** Falls Bilder oder andere Assets verwendet werden (aktuell nicht im Fokus), müssen diese komprimiert werden.
- **CDN (Content Delivery Network):** Für eine globale Auslieferung und schnellere Ladezeiten kann ein CDN in Betracht gezogen werden.

6. Sprüche-Datenbank

- Die initiale Befüllung mit über 2000 Sprüchen muss vorbereitet werden (z.B. als CSV-Datei oder JSON-Array zum Import).
- Struktur pro Spruch:
    - `id` (eindeutige Identifikationsnummer)
    - `text` (der Schnupfspruch selbst)
    - `eingereicht_von` (optional, Name des Einreichenden)
    - `status` (z.B. 'freigegeben', 'zur_pruefung')
    - `erstellt_am` (Zeitstempel der Erstellung/Einreichung)

7. Datenschutz

- Falls Namen bei der Sprucheinreichung erfasst werden, ist eine Datenschutzerklärung notwendig, die über die Verwendung dieser Daten informiert.

- Minimierung der Datenerfassung.

8. Zukünftige Erweiterungen (Optional)

- Bewertung von Sprüchen
- Kategorisierung von Sprüchen
- Suchfunktion
- Nutzer-Accounts zum Speichern von Lieblingssprüchen

9. Zusammenfassung der wichtigsten Punkte für die Entwickler

- **Priorität 1: Geschwindigkeit.** Alle technologischen Entscheidungen sollten diesem Ziel untergeordnet sein.
- **Priorität 2: Einfachheit.** Sowohl im Design als auch in der Code-Struktur.
- **Frontend:**
    - Startseite: "Priis"-Button, darüber Textfeld für zufälligen Spruch.
    - Zweiter Button "Spruch einreichen" führt zu neuer Seite.
- **Backend:**
    - API für zufälligen Spruch.
    - API für Sprucheinreichung.
    - Effiziente Datenbank für >2000 Sprüche mit schnellem Random-Zugriff.
    - Mechanismus zur Moderation/Freischaltung eingereichter Sprüche.
- **Mobile-First und responsives Design.**

Dieses Dokument dient als Grundlage für die Entwicklung. Detailliertere Spezifikationen (z.B. genaue API-Definitionen, Datenbank-Schema) können in weiteren Schritten erarbeitet werden.