

Arbeitsanweisung für excelähnliche Tabellen auf Webseiten

Diese PDF ist als technische Grundlage für einen KI-Chat gedacht. Sie beschreibt bewährte Regeln für verstellbare Webtabellen. Der konkrete Aufbau, die Spalten und gewünschte Zusatzfunktionen werden der KI zusätzlich im Chat beschrieben.

Wichtig: Prüfe zuerst, ob im Projekt bereits eine gut funktionierende Tabelle als technische Referenz vorhanden ist. Wenn ja, verwende deren bewährte Mechanik. **Wenn keine geeignete Referenztable existiert, baue die Tabelle anhand der folgenden Grundregeln vollständig neu.** Das Fehlen einer vorhandenen Tabelle ist kein Grund, die Aufgabe abzubrechen oder Funktionen wegzulassen.

1. Zuerst den Bedarf klären

- Bevor Du programmierst, kläre anhand der Beschreibung des Nutzers, welche Spalten die Tabelle benötigt und welche Komfortfunktionen tatsächlich gewünscht sind.
- Die Grundidee ist excelähnlich, nicht Excel selbst: Spalten können beispielsweise an ihrer Grenze mit der Maus breiter oder schmaler gezogen werden. Weitere Funktionen wie Sortierung, Filter, Suche, Ausblenden von Spalten oder Zeilen, Formeln oder frei verschiebbare Spalten gehören nur dann dazu, wenn sie ausdrücklich gewünscht oder für den Anwendungsfall sinnvoll sind.
- Erfinde keine fehlenden Anforderungen. Wenn eine wesentliche Entscheidung nicht aus Projekt oder Nutzerwunsch hervorgeht, frage nach.

2. Referenz vorhanden - oder neu bauen

- Prüfe den vorhandenen Quellcode auf eine bereits praktisch funktionierende Tabelle mit vergleichbarem Verhalten. Falls eine passende Referenz existiert, untersuche deren Tabellenstruktur, Resize-Griffe, Mindestbreiten, Gesamtbreite, Scroll-Wrapper, Speicherung und das Zusammenspiel mit Sortierung oder Eingabefeldern.
- Existiert keine passende Referenz, verwende die Regeln dieser PDF als Ausgangspunkt für eine neue robuste Lösung. Verwende keine erfundene Projektabhängigkeit und keine hart codierte Spaltenzahl.

3. Grundmechanik für verstellbare Spalten

- Für eine neu gebaute Tabelle ist table-layout: fixed ein bewährter Ausgangspunkt. Jede echte Kopfspalte erhält genau einen Resize-Griff an ihrer rechten Kante.
- Beim Ziehen wird nur die Breite dieser einen Spalte verändert. Danach wird die Gesamtbreite der Tabelle als Summe der aktuellen Spaltenbreiten gesetzt. Wird die Tabelle breiter als ihr sichtbarer Bereich, bleibt horizontales Scrollen möglich.
- Die letzte Spalte wird wie jede andere behandelt. Ihr rechter Griff ist zugleich die rechte Außenkante der Tabelle. Keine Sonderlogik einbauen, die beim Ziehen einer Grenze alle Spalten gleichzeitig verändert.

4. Mindestbreiten - schmal, aber nicht kaputt

- Spalten sollen weit zusammengeschoben werden können, aber nicht so weit, dass Überschrift, notwendiger Zellinhalt oder Bedienung unbrauchbar werden. Verwende deshalb sinnvolle, möglichst inhaltsabhängige Mindestbreiten statt einer einzigen starren Mindestbreite für alle Spalten.
- Für die Messung nur den tatsächlich sichtbaren relevanten Inhalt berücksichtigen. Versteckte Details, Untertabellen oder große Bedienelemente dürfen die Mindestbreite nicht versehentlich aufblasen.
- Wenn der Nutzer eine Spalte bis fast auf null reduzieren oder vollständig ausblenden möchte, ist das eine zusätzliche Funktion und muss ausdrücklich als solche umgesetzt werden.

5. Direkte Zellen und verschachtelte Tabellen

- Die Resize-Logik darf für eine Tabelle nur deren direkte Kopf- und Datenzellen verwenden. Unbegrenzte Selektoren können Zellen verschachtelter Untertabellen versehentlich mitzählen.
- Jede Untertabelle ist technisch eine eigene Tabelle und wird - falls sie ebenfalls verstellbar sein soll - separat initialisiert. Die Zahl der von der Resize-Logik erkannten Spalten muss zur sichtbaren Haupttabelle passen.

6. Dynamische oder zunächst versteckte Tabellen

- Prüfe, ob die Tabelle beim Laden bereits vorhanden und sichtbar ist oder erst später per JavaScript/AJAX erzeugt bzw. in einem geschlossenen Detailbereich angezeigt wird.
- Dynamisch erzeugte Tabellen erst nach ihrer Erzeugung initialisieren. Bei zunächst versteckten Tabellen keine ungeprüften Breitenmessungen verwenden; gegebenenfalls erst nach dem Sichtbarwerden messen oder robuste textbasierte Mindestbreiten verwenden.

7. Breiten speichern

- Die vom Nutzer eingestellten Spaltenbreiten sollen nach einem Neuladen wiederhergestellt werden, wenn dies für die Anwendung sinnvoll ist.
- Bei Anwendungen mit Benutzerkonten ist eine serverseitige Speicherung pro Benutzer und Tabelle vorzuziehen. Ohne Benutzerkonten kann localStorage eine passende einfache Lösung sein. Die Speichertechnik muss zur vorhandenen Anwendung passen - nicht blind PHP, eine bestimmte API oder einen bestimmten Dateipfad voraussetzen.
- Gespeicherte Werte immer gegen die aktuelle Spaltenzahl und gültige Mindestbreiten prüfen. Ein stabiler, bei Strukturänderungen versionierbarer Layout-Schlüssel verhindert, dass alte Breiten auf eine geänderte Tabelle angewendet werden.

8. Sortierung, Eingaben und weitere Funktionen

- Wenn die Tabelle sortierbar ist, darf ein Drag am Resize-Griff keine Sortierung auslösen. Resize- und Sortierereignisse sauber voneinander trennen.
- Vorhandene Eingabefelder, Selects, Links, Buttons, Filter, Suche, Detailbereiche und Summen dürfen durch die Tabellenmechanik nicht beschädigt werden.
- Bei einer neuen Tabelle nur die Funktionen ergänzen, die gewünscht oder für den konkreten Zweck sinnvoll sind. Spalten/Zeilen ausblenden, Spalten verschieben, Formeln, Zellberechnungen oder Excel-ähnliche Bearbeitung sind keine automatische Grundfunktion.

9. Praktische Abnahme

- Teste mindestens: jede Spaltengrenze ziehen; letzte Spalte ziehen; sehr schmale und breite Werte; horizontales Scrollen; Neuladen und Wiederherstellen; vorhandene Sortierung; Eingaben und Buttons; mobile bzw. schmale Ansicht; gegebenenfalls verschachtelte oder dynamische Tabellen.
- Bei einer vorhandenen Anwendung dürfen angrenzende Funktionen und das übrige Design nicht nebenbei verändert werden.

10. Wenn es nicht funktioniert: messen statt raten

- Nach wiederholten Fehlversuchen nicht immer neue Varianten auf Verdacht bauen. Ermittle den tatsächlichen Browserzustand.
- Prüfe insbesondere: erwartete und erkannte Spaltenzahl, Anzahl der Resize-Griffe, `style.width`, berechnete Breite und DOM-Breite je Spalte, Tabellenbreite, `min-width`, `scrollWidth/clientWidth`, Wrapper-Breite, gespeicherte Breiten, Layout-Schlüssel, Initialisierungszeitpunkt und mögliche Überlagerungen der Resize-Griffe.
- Erst wenn die konkrete Ursache belegt ist, den nächsten Änderungsschritt durchführen.

Arbeitsauftrag für den KI-Chat

Diesen Abschnitt kann der Nutzer zusammen mit seiner eigenen Beschreibung der gewünschten Tabelle verwenden:

Lies die beigefügte PDF vollständig und verwende sie als erste technische Anleitung und Regelwerk für den Tabellenbau.

Ich beschreibe Dir anschließend, welche Tabelle ich für meine Webseite benötige und welche Funktionen sie haben soll.

Prüfe zuerst den vorhandenen Projektstand. Wenn bereits eine gut funktionierende, vergleichbare Tabelle existiert, verwende ihre bewährte Technik als Referenz, soweit sie zu meinem Wunsch passt.

Wenn keine geeignete Referenztabelle existiert, erstelle die Tabelle anhand der Grundregeln aus dieser PDF vollständig neu. Das Fehlen einer vorhandenen Tabelle ist kein Grund, die Aufgabe abzubrechen.

Setze nur Funktionen um, die ich wünsche oder die für die gewünschte Tabelle technisch notwendig sind. Nimm keine unnötigen Änderungen an anderen Bereichen der Webseite vor. Wenn eine wesentliche Anforderung unklar ist und sich nicht aus meinem Projekt oder meiner Beschreibung ergibt, frage nach, statt sie zu erfinden.

Merksatz: Erst verstehen, was gebraucht wird. Vorhandene bewährte Technik nutzen, wenn sie existiert. Wenn nicht, nach den Grundregeln sauber neu bauen. Bei Fehlern messen statt raten.