



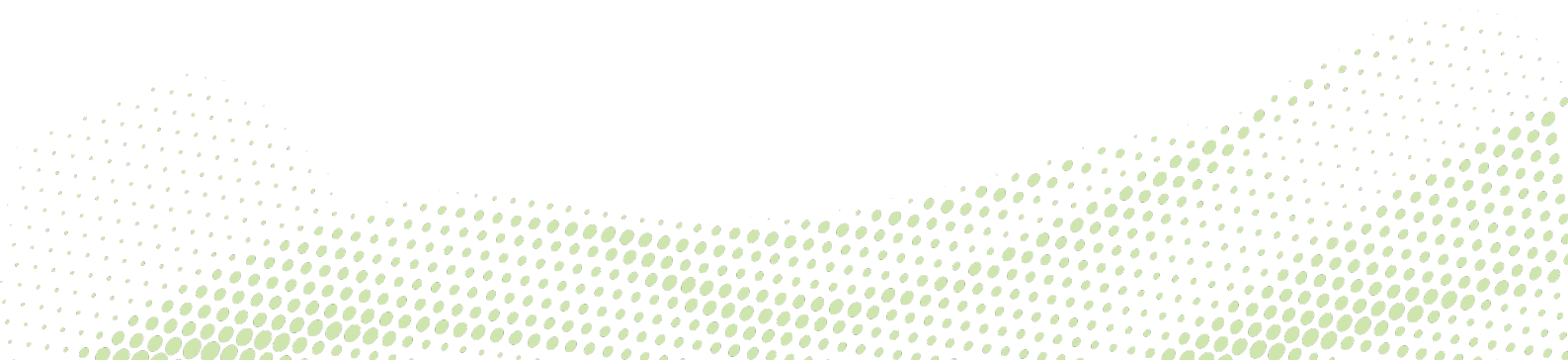
Projektarbeit JavaScript



JavaScript Projektarbeit

Ausgangssituation

Du arbeitest als Webentwickler in einem mittelständischen Unternehmen, das beschlossen hat, seine interne Mitarbeiterverwaltungssoftware zu überarbeiten. Die aktuelle Version läuft noch auf einem veralteten System, das weder benutzerfreundlich noch effizient ist. Deine Aufgabe ist es, mit reinem JavaScript eine neue, dynamische und interaktive Benutzeroberfläche zu entwickeln. Diese soll Funktionen wie die Anzeige von Mitarbeiterlisten, das Hinzufügen neuer Mitarbeiter, die Suche und Filterung von Mitarbeiterdaten sowie die Bearbeitung und Löschung vorhandener Einträge beinhalten.



Aufgabenpakete

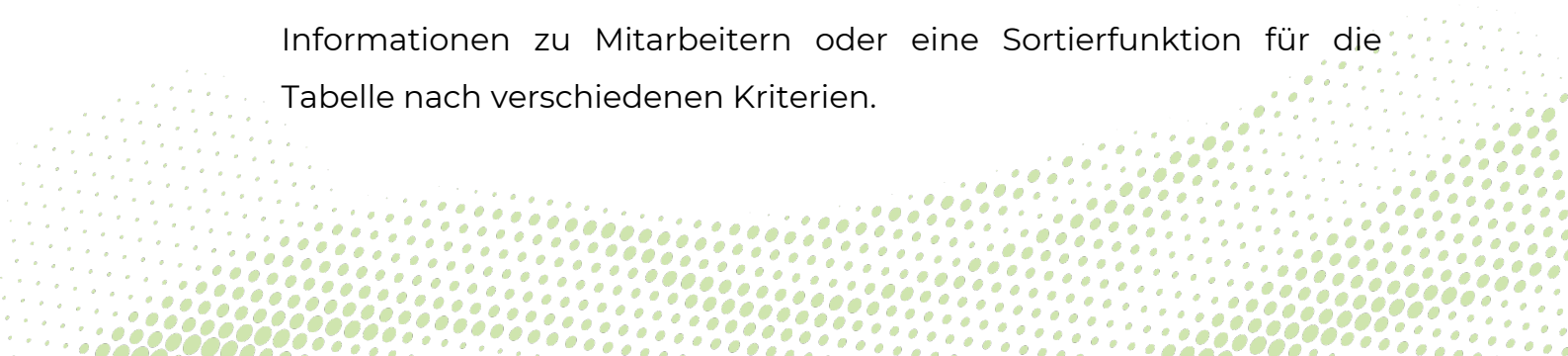
Aufgabenpaket 1: Grundlegende Projektstruktur und DOM-Manipulation

1. Erstelle eine HTML-Datei als Basis für die Anwendung und binde eine JavaScript-Datei ein.
2. Entwickle eine statische Benutzeroberfläche mit HTML, die Bereiche für Mitarbeiterlisten und ein Formular zum Hinzufügen neuer Mitarbeiter und eine Suchleiste enthält.
3. Nutze DOM-Manipulation in JavaScript, um dynamisch Mitarbeiterdaten in der Liste anzuzeigen. Verwende dafür vorerst hart codierte Daten in einem Array von Objekten.

Aufgabenpaket 2: Dynamische Interaktion und Datenhandling

1. Implementiere die Funktionalität, um über das Formular neue Mitarbeiterdaten zu erfassen und zur Liste hinzuzufügen.
2. Füge eine Suchfunktion hinzu, die es ermöglicht, Mitarbeiter anhand ihres Namens in der Liste zu finden. Die Suche soll direkt beim Tippen Ergebnisse liefern (Live-Suche).
3. Erstelle Funktionen zum Bearbeiten und Löschen von Mitarbeitereinträgen direkt in der Liste. Nutze dazu Modal-Dialoge oder ein ähnliches UI-Element für die Bearbeitung.

Aufgabenpaket 3: Erweiterte Funktionen und Benutzerinteraktion

1. Implementiere eine Paginierung oder Infinite Scrolling für die Mitarbeiterliste, um die Performance bei einer großen Anzahl von Einträgen zu verbessern.
 2. Entwickle eine Filterfunktion, die es erlaubt, Mitarbeiter nach verschiedenen Kriterien (z.B. Abteilung, Position) zu filtern.
 3. Füge interaktive Elemente hinzu, wie z.B. Tooltips für zusätzliche Informationen zu Mitarbeitern oder eine Sortierfunktion für die Tabelle nach verschiedenen Kriterien.
- 

Aufgabenpaket 4: Validierung und Fehlerbehandlung

1. Implementiere Client-seitige Validierung für das Formular zum Hinzufügen und Bearbeiten von Mitarbeitern. Stelle sicher, dass alle Eingaben korrekt sind, bevor sie verarbeitet werden.
2. Entwickle eine Fehlerbehandlung, die benutzerfreundliche Fehlermeldungen anzeigt, falls beim Hinzufügen, Bearbeiten oder Löschen von Mitarbeitern Probleme auftreten.
3. Stelle sicher, dass die Anwendung auch bei fehlerhaften Daten oder Bedienungsfehlern stabil läuft und entsprechendes Feedback an den Benutzer gibt.

Aufgabenpaket 5: Lokale Datenspeicherung und -wiederherstellung

1. Nutze den Local Storage des Browsers, um die Mitarbeiterdaten zu speichern, damit diese auch nach einem Neuladen der Seite erhalten bleiben.
2. Implementiere eine Funktion zum Exportieren und Importieren der Mitarbeiterdaten im JSON-Format, um ein einfaches Backup und Wiederherstellung zu ermöglichen.
3. Füge eine Option hinzu, um alle Daten zurückzusetzen und den Local Storage zu leeren, falls benötigt.

Anforderungen

- Verwende ausschließlich Vanilla JavaScript, ohne externe Bibliotheken oder Frameworks.
 - Achte auf sauberen, gut strukturierten und kommentierten Code.
 - Die Benutzeroberfläche soll rein mit HTML und CSS gestaltet sein, ohne Verwendung von CSS-Frameworks.
 - Die Anwendung muss in allen modernen Browsern korrekt funktionieren.
 - Implementiere responsive Design-Prinzipien, damit die Anwendung auf verschiedenen Geräten nutzbar ist.
- 