Benutzerdaten suchen

# Problemstellung

### Deine Rolle

Du bist Webentwickler bei der Firma WebDesign GmbH in Imst.

### Die Situation

Für ein bekanntes Open-Source CMS soll eine Filterfunktion für Benutzerdaten entwickelt werden. Deine Arbeitskollegin aus der Graphikabteilung hat bereits zwei Mockups erstellt.

### Dein Ziel

Dein Chef hat dich beauftragt einen funktionsfähigen Prototyp zum Filtern und Anzeigen von Benutzern zu erstellen. Die Daten werden derzeit in einem Array zur Verfügung gestellt, später werden diese direkt aus der Datenbank geladen.

### Deine Zielgruppe

Die Filterfunktion soll nach erfolgreichen Tests in das CMS integriert und der Community kostenlos zur Verfügung gestellt werden. Die Anwender sind somit alle Backend-Administratoren welche das CMS einsetzen. Da die Entwickler-Community vor der Freigabe einen kritischen Blick auf den Code werfen wird, muss dieser allen aktuellen Regeln und Standards entsprechen.

### Das erwartete Produkt

#### Funktionen

Der Filterfunktion für Benutzerdaten sollte mindestens folgendes umfassen:

* Darstellung der wichtigsten Benutzerdaten in Tabellenform (<table>)
  + Vorname und Nachname in gemeinsamer Spalte
  + Link auf die Detailseite mit Hilfe der Datensatz-ID z.B. href=“details.php?id=1“
  + Datumsformat DD.MM.YYYY
  + Zeilen farblich abwechselnd gestalten
* Eingabefeld zum Filtern nach Name und E-Mail
  + Tabellenzeilen werden entsprechend dem Suchfeld herausgefiltert
  + Eine Teilübereinstimmung des Suchtextes wird als Treffer gewertet
  + Buttons zum Suchen und Leeren (Tipp: zwei Formulare, kein JavaScript)
* Ausgabe einer Fehlermeldung falls keine Einträge gefunden wurden
* Responsive Design
* Verwendung eines HTML/CSS Frameworks, z.B. Bootstrap

Die Detailseite für Benutzerdaten sollte mindestens folgendes umfassen:

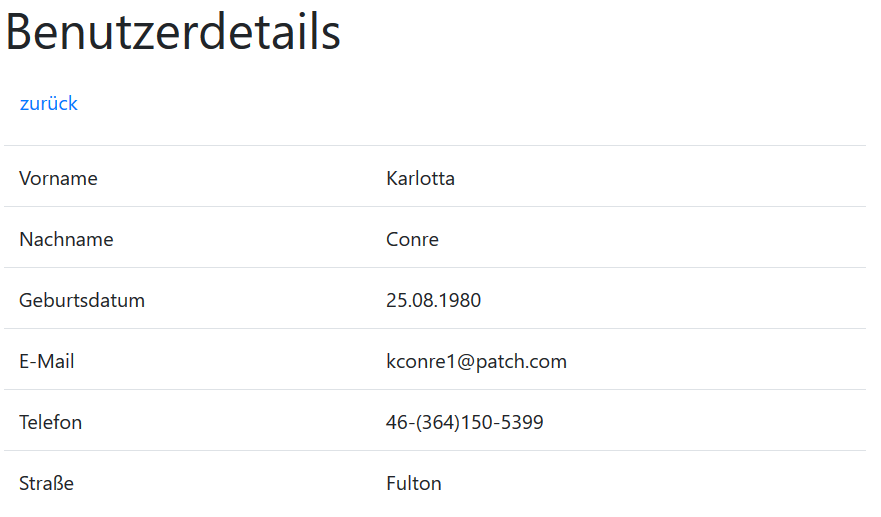
* Übersichtliche Darstellung aller Benutzerdaten
* Über einen zurück-Link wird der User wieder zur Übersichtsseite geleitet

Benutzerdaten:

* Die Benutzerdaten sollen über entsprechende Funktionen zur Verfügung gestellt werden
  + function getAllData()
  + function getDataPerId($id)
  + function getFilteredData($filter)

#### Mockup:





# Zusatzaufgabe 1 (Link zurück)

Die Suchparameter sollten beim Öffnen der Detailseite, bzw. nach Rückkehr auf die Startseite erhalten bleiben. Verwende dazu GET-Parameter und KEINE Cookies.

# Zusatzaufgabe 2 (jQuery)

Erweitere das Eingabefeld um einen Keylistener, durchsuche die Tabelle mittels jQuery bei Eingabeänderungen (vor Absenden des Formulars, Unabhängig vom Such-Button) und filtere die Zeilen entsprechend.

Hinweise:

* Selector: <https://api.jquery.com/element-selector/>
* foreach-Schleife: <http://api.jquery.com/jquery.each/>
* Kindelement auswählen: <https://api.jquery.com/nth-child-selector/>

# Zusatzaufgabe 3 (SQL)

Erweitere das Programm so, dass die angezeigten Daten aus der Datenbank geladen werden. Verwende einen Online-Datengenerator z.B. <https://mockaroo.com/> um die Daten zu generieren. Tausche das data.php Skript gegen db.data.php aus und verwende dort dieselben Funktionen.

DB-Schema:

CREATE DATABASE IF NOT EXISTS `php13` DEFAULT CHARACTER SET latin1 COLLATE latin1\_swedish\_ci;

USE `php13`;

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

`id` int(11) DEFAULT NULL,

`firstname` varchar(50) DEFAULT NULL,

`lastname` varchar(50) DEFAULT NULL,

`email` varchar(50) DEFAULT NULL,

`phone` varchar(50) DEFAULT NULL,

`birthdate` date DEFAULT NULL,

`street` varchar(50) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `user` (`id`, `firstname`, `lastname`, `email`, `phone`, `birthdate`, `street`) VALUES

(1, 'Erma', 'Almond', 'ealmond0@ihg.com', '913-783-2500', '2018-02-07', '6904 Rigney Place');