

2022

Robert Göpfert,
Kai Rohrpasser und
Dennis Schneeweiß

[BUECHERLADEN PHP PROJEKT]

Ein Datenbank Kompagnon der Extravagante

Inhalt

1. Beschreibung des Programms	2
2. Anforderungen	2
3. Anleitung	3
3.1 Landing Page	3
3.2 Datenansicht	4
3.3 Edit-Pop-Up Fenster	5
3.4 SQL-Injection	6
3.5 Datensatz hinzufügen	7
4 Exemplarische Code-Beispiele	8
4.1 Sortierfunktion	8
4.2 Abfrage für Referenzen beim Bearbeiten von Daten	9

1. Beschreibung und Bedeutung des Programms

Das Programm stellt eine Benutzerfreundliche Schnittstelle zu der Datenbank Buecherladen zur Verfügung. Durch verschiedene Funktionen wird der Umgang mit den Daten vereinfacht. Unter anderem ist es dem User möglich die Daten aus den Tabellen anzuzeigen, zu verändern und zu löschen. Durch ein Ansprechendes User Interface und zahlreichen Funktions-Buttons werden das Navigieren und das Arbeiten an den Datensätzen vereinfacht. Zusätzlich besteht, für SQL-Kundige, die Möglichkeit, selbst SQL Befehle zu schreiben und auszuführen, falls die angebotenen Funktionen nicht ausreichen.

2. Anforderungen

Die Anforderungen sind in vier Teilabschnitten unterteilt.

Teil 1: Benutzeroberfläche

- Alle Tabellen sollen per Knopfdruck angezeigt werden.
- Die Möglichkeit selbst einen SQL-Befehl zu schreiben und auszuführen soll gegeben sein.
- Die Oberfläche soll optisch ansprechend und übersichtlich gestaltet sein.

Teil 2: Bearbeitung der Daten

- Es soll dem Benutzer möglich sein Die Daten zu verändern, zu löschen oder neue Datensätze einzufügen.

Teil 3: Sortierung

- Dem Benutzer soll es möglich sein Die ausgegebenen Tabellen nach Belieben zu sortieren.

Teil 4: Dokumentation

- Eine Dokumentation und ein Benutzerhandbuch sollen geschrieben werden.
- Der Nutzer soll mit einer Schritt für Schritt Anleitung durch das Projekt geführt werden
- Die Bedeutung und Funktionalität des Projektes soll erfasst werden.
- Optisch soll sie nutzerfreundlich und ansprechend gestaltet sein.

3. Anleitung

3.1. Landing Page:

Auf der Landing Page sind je ein Feld für jede Tabelle (1) der Datenbank zu sehen.

In gelb wird der Name der Tabelle angezeigt (2), in grün die Anzahl der Attribute und Entitäten (3) und in lila ein Button der zur Datenansicht der jeweiligen Tabelle führt (4).

Um sich eine der Tabellen anzeigen zu lassen drückt man auf dem entsprechenden Feld auf Tabelle Anzeigen (4).

Mit dem Button Datenbank zurücksetzen (5) setzt man die Datenbank auf ihren Ursprünglichen Zustand zurück.

Mit dem Button SQL Injection (6) lassen sich selbst erstellte SQL Befehle ausführen.



3.2. Datenansicht:

Mit dem Zurück-Button (1) navigiert man zurück zur Landing Page.

Mit dem Dropdown Menü (2) kann zwischen Tabellen gewechselt werden.

Die Spaltenüberschrift (3) kann angeklickt werden um die Datensätze zu sortieren. Wenn die Spaltenüberschrift einmal angeklickt wird sind die Datensätze von A-Z sortiert und wenn sie zweimal geklickt wird, von Z-A. Neben der Spaltenüberschrift erscheint ein Pfeil der die Sortierrichtung anzeigt.

Mit dem Edit Button (4) öffnet sich ein Pop-Up Fenster in dem der Datensatz bearbeitet werden kann.

Der Del Button (5) löscht den Datensatz.

Mit der Projekt-Überschrift (6) kann man zu jeder Zeit zurück zur Landing Page navigieren.

Der Button mit der Aufschrift Eintrag einfügen (7) öffnet ein Popup-Fenster mit dem ein neuer Datensatz erstellt werden kann.

buecher_id	titel	verkaufspreis	erkaufspreis	erscheinungsjahr	verlegerverleger_id	
2	Hess	9.95	6.45	1977	1	<div>Edit 4</div> <div>Del 5</div>
3	Mein Hähnchen, mein Bruder und ich	11.90	5.12	1965	2	<div>Edit</div> <div>Del</div>
5	Wilde Jagd auf John Smith Die Rache	15.90	7.11	1995	1	<div>Edit</div> <div>Del</div>
6	Wilde Jagd auf John Smith Die Vergeltung	29.95	15.95	2012	3	<div>Edit</div> <div>Del</div>
7	Der letzte Kuss	6.00	2.40	2000	5	<div>Edit</div> <div>Del</div>
8	Mondschein	7.90	6.23	2008	4	<div>Edit</div> <div>Del</div>
9	Wagen 1331	15.95	9.62	2008	6	<div>Edit</div> <div>Del</div>
10	Lass ihn doch singen!	36.80	23.95	2007	4	<div>Edit</div> <div>Del</div>
11	Es war einmal	29.95	18.83	2006	5	<div>Edit</div> <div>Del</div>
12	Father and Son	16.00	8.00	2011	6	<div>Edit</div> <div>Del</div>

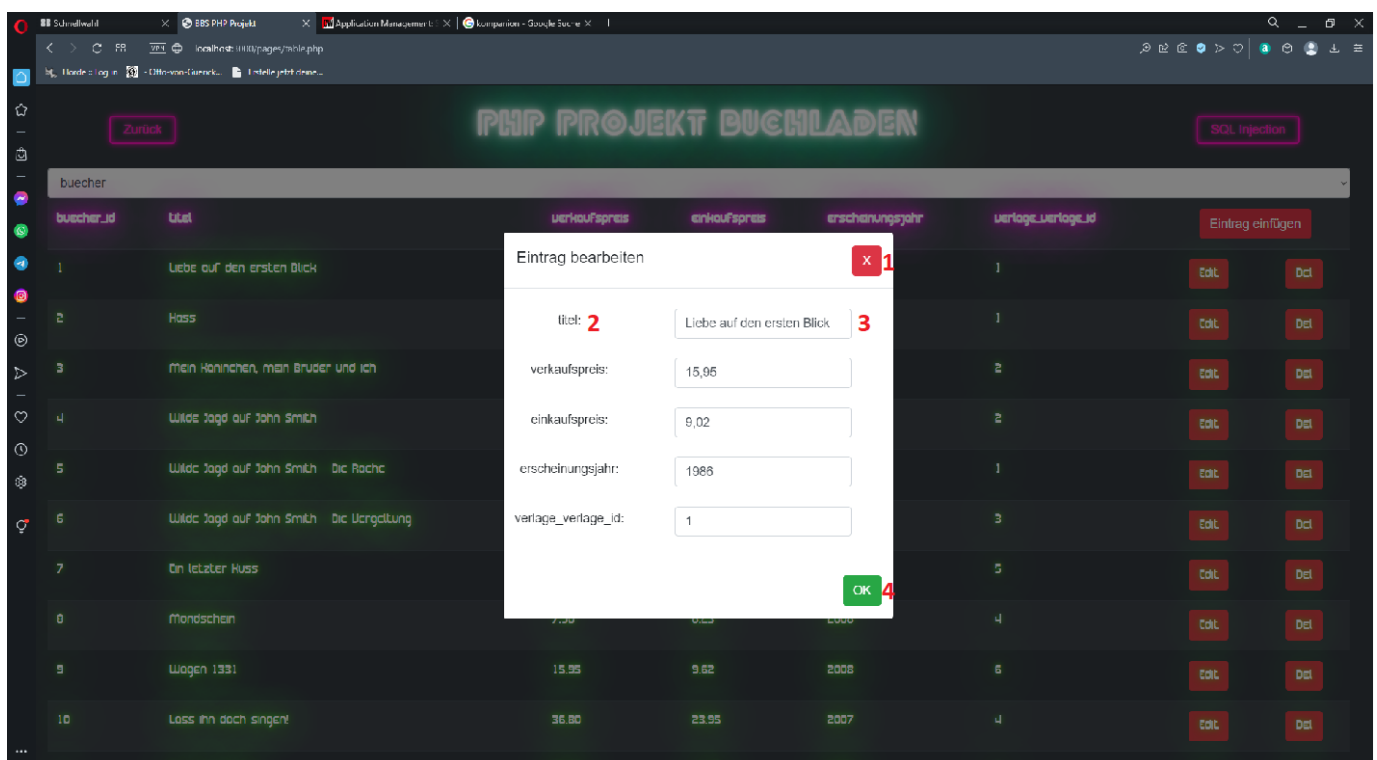
3.3. Edit-Pop-Up Fenster:

Mit dem roten X-Button (1) lässt sich der Bearbeitungsvorgang abbrechen.

Für jedes zu bearbeitende Feld wird der Name (2) angezeigt.

Neben dem Namen des Feldes ist ein Textfeld (3) in dem die Bearbeitungen vorgenommen werden.

Mit dem grünen OK-Button (4) werden die vorgenommenen Veränderungen übernommen.



3.4. SQL-Injektion

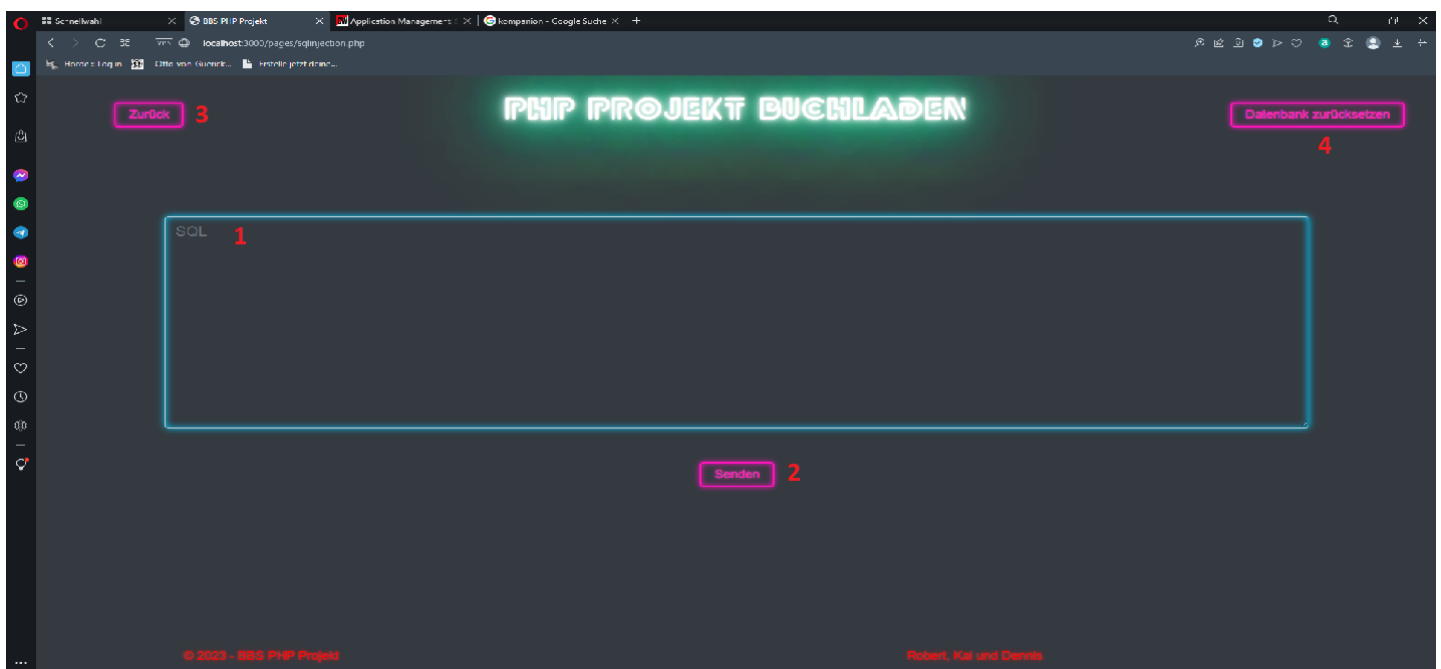
In dem Textfeld (1) hat man die Möglichkeit selbst einen SQL-Befehl zu erstellen um die Daten in der Datenbank zu verändern bzw. anzuzeigen.

Mit dem Senden Button (2) schickt man sein SQL-Statement los zum Verarbeiten.

Der Zurück-Button (3) lässt einen zu der Landing Page navigieren.

Der Button mit der Aufschrift Datenbank zurücksetzen (4) bringt die Datenbank auf den Urzustand.

Sollte eine Select abfrage gesendet werden wird die Ausgabe unter dem Senden Button angezeigt.



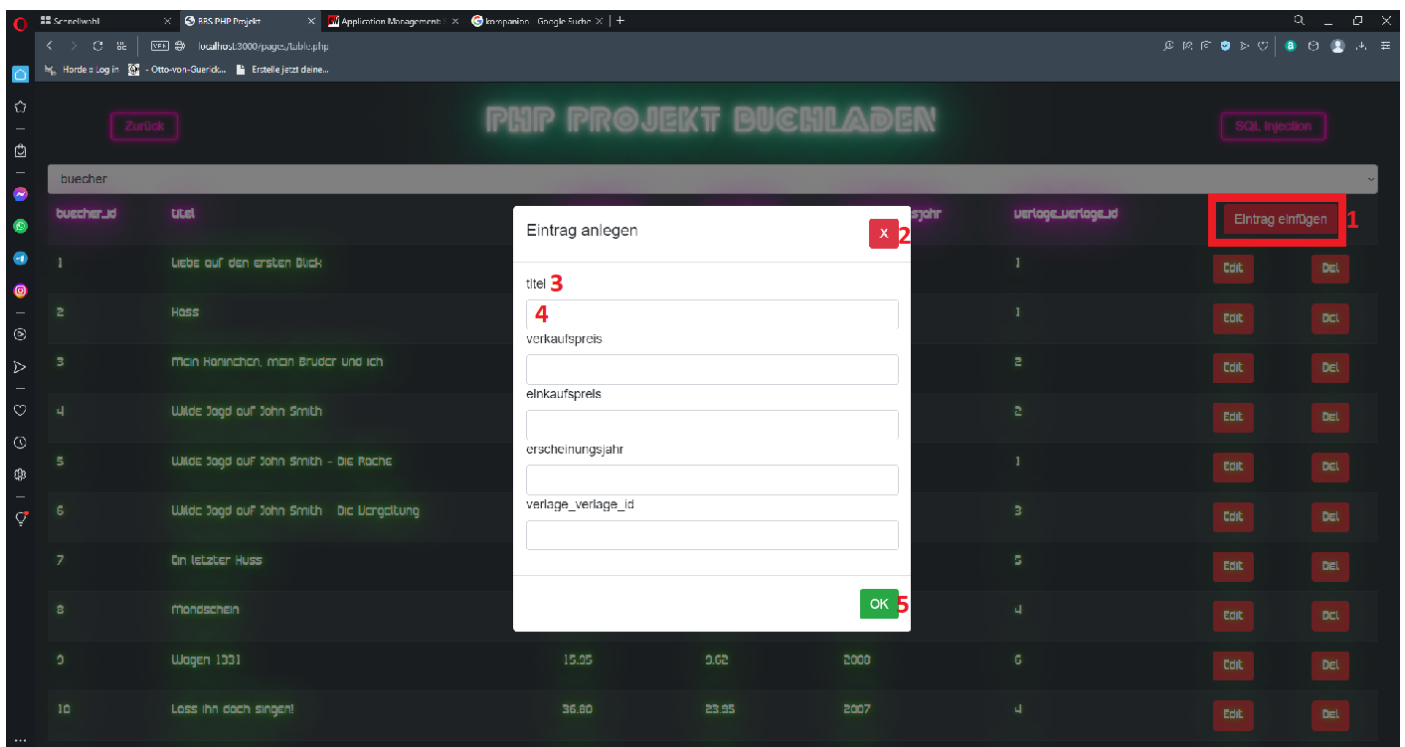
3.5. Eintrag hinzufügen

Nachdem der User auf den Button Eintrag einfügen (1) klickt öffnet sich ein Pop-Up Fenster in dem Ein neuer Eintrag erstellt werden kann.

Mit dem roten X (2) lässt sich der Vorgang abbrechen.

Für jeden Wert eines Datensatzes werden der Name des Attributes (3) und ein Textfeld (4) angezeigt in dem der gewünschte Wert eingetragen werden kann.

Mit dem grünen OK Button (5) schließt man den Vorgang ab und der neue Datensatz wird erstellt.



4. Exemplarische Code-Beispiele

4.1. Sortierfunktion

Zum Sortieren haben wir einen einfachen Bubble-Sort benutzt bei dem jede Zeile einer Tabelle mit der nächsten verglichen wird (Zeile 87/91) und je nach Sortier-Reihenfolge die Datensätze vertauscht oder übersprungen werden (Zeile 88/92). So werden nach und nach die Datensätze in die richtige Reihenfolge gebracht. Zum Schluss wird die fertig sortierte Tabelle zurückgegeben (Zeile 97).

Um die richtige Reihenfolge zu bestimmen klickt man auf den Attributs Namen und je nachdem welche Reihenfolge gerade aktiv ist (Zeile 107) wird sie in die entgegengesetzte Richtung gebracht (Zeile 108/111).

```
82  function orderTableData($tableData, $orderBy, $orderDirection) {
83      $n = count($tableData);
84      for ($i = 0; $i < $n - 1; $i++) {
85          for ($j = 0; $j < $n - $i - 1; $j++) {
86              if ($orderDirection === 'ASC') {
87                  if ($tableData[$j][$orderBy] > $tableData[$j + 1][$orderBy]) {
88                      swap($tableData[$j], $tableData[$j + 1]);
89                  }
90              } else {
91                  if ($tableData[$j][$orderBy] < $tableData[$j + 1][$orderBy]) {
92                      swap($tableData[$j], $tableData[$j + 1]);
93                  }
94              }
95          }
96      }
97      return $tableData;
98  }
99
100 function swap(&$a, &$b) {
101     $temp = $a;
102     $a = $b;
103     $b = $temp;
104 }
105
106 function ascORdesc($value) {
107     if ($value == "ASC") {
108         return "DESC";
109     }
110     else {
111         return "ASC";
112     }
113 }
```

4.2. Abfrage für Referenzen beim Bearbeiten von Daten

Um zu erkennen wann das Pop-Up Fenster angezeigt wird gibt es eine Abfrage (Zeile 8) die kontrolliert ob der Edit Button gedrückt wurde.

Durch die Abfrage, ob die Tabelle in der die Werte stehen Referenzen hat (Zeile 11) wird sichergestellt, dass auch Fremdschlüssel geändert werden falls nötig.

Durch die Funktionen updateTable (Zeile 14) und updateRefTable (Zeile 19) werden dann die Datensätze in der Datenbank überschrieben.

```
8  if (isset($_POST['btnEdit'])) {
9      $table = $_POST['table'];
10     $values = $_POST['edit'];
11     if (isRefTable($table)) {
12         if (isset($_POST['oldValues'])) {
13             $oldValues = $_POST['oldValues'];
14             updateRefTable($table, $oldValues, $values);
15         }
16     } else {
17         if (isset($_POST['id'])) {
18             $id = $_POST['id'];
19             updateTable($table, $id, $values);
20         }
21     }
22 }
23
```