

Projektdokumentation KauansAllesDatenbank

Autoren: Kimi Müller, David Matheja, Simon Hauck

Matrikelnummer: 7544003 (Kimi), 9809395 (David), 7182169 (Simon)

Abgabe: 22.06.2025

1. Technische Dokumentation

Das Projekt "KauansAllesDatenbank" wurde im Rahmen des Moduls Web Engineering 2 entwickelt und verfolgt das Ziel, eine benutzerfreundliche, intuitive Webanwendung, zur Einsicht, Erschaffung und Verwaltung von Datenbanken bereitzustellen. Benutzer sollen in der Lage sein, eigene Datenbanken (Sei es eine Datenbank, in der man seine gelesenen Bücher sortiert oder eine, bei der man seine Ausgaben einträgt) zu erschaffen, einzusehen und zu bearbeiten.

Die Anwendung berücksichtigt grundlegende Aspekte der Webentwicklung, darunter saubere HTML-Struktur, Clientseitige Logik mit Javascript, serverseitige Logik mit PHP sowie Anbindung an eine relationale MySQL-Datenbank.

Projektstruktur

Das Projekt ist klar strukturiert, um eine Trennung von **Darstellung**, **Datenanbindung** und **deployment** zu gewährleisten:

```
|— docker-compose.yml
|— .env
└─ php_app
    |— Dockerfile.php
    |— css
    |   |— dark.css
    |   |— light.css
    |   |— style.css
    |   └─ themes.css
    |— display_table.php
    |— edit-entry.php
    |— include
    |   |— db_config.php
    |   └─ footer.php
```

```

|   |— functions.php
|   └— header.php
|— index.php
|— js
|   |— oneko
|   |   |— oneko.gif
|   |   └— oneko.js
|   └— table-features.js
|   └— theme.js
|— login.php
└— logout.php

```

- **/ (Projekt-Root)**
 - docker-compose.yml: Definiert die Anwendungsdienste, einschließlich des PHP-Webserver und der MySQL-Datenbank, und konfiguriert deren Zusammenspiel.
- **/php_app**
 - Dockerfile.php: Spezifiziert die Konfiguration des Docker-Images für den Apache/PHP-Server, einschließlich der PHP-Version und der erforderlichen Erweiterungen (pdo, pdo_mysql)
 - index.php: Die Hauptseite nach dem Login, die dem Benutzer eine Liste seiner Tabellen anzeigt und die Erstellung neuer Tabellen.
 - display_table.php: Zeigt den Inhalt einer ausgewählten Tabelle an und ermöglicht CRUD-Operationen (Erstellen, Lesen, Aktualisieren, Löschen) für die Einträge.
 - login.php, logout.php: Verantwortlich für die Benutzerauthentifizierung und die Verwaltung von Sitzungen. login.php übernimmt das theme, das zuletzt gesetzt wurde.
 - **/include:**
 - db_config.php: Stellt die PDO-Datenbankverbindung her. Initialisiert bei der ersten Ausführung die users-Tabelle und legt einen Standard-Admin-Benutzer an.
 - header.php, footer.php: Enthalten den Kopf- und Fußbereich des HTML-Layouts, einschließlich Navigation und Skript-Einbindungen.
 - functions.php: Enthält zentrale PHP-Funktionen, die anwendungsweit wiederverwendet werden, z. B. zur Erstellung von SQL-Abfragen und zum Rendern von Formularfeldern.
 - **/css:**
 - style.css, dark.css, light.css: Drei verschiedene Stylesheets, die unterschiedliche visuelle Themes für die Anwendung bereitstellen.

- **/js:**
 - **theme.js:** JavaScript zur Verwaltung des Theme-Wechsels, speichert die Auswahl des Benutzers lokal.
 - **table-features.js:** Fügt den Datentabellen clientseitige Sortier- und Filterfunktionen hinzu. Bei großen Datenmengen ineffizient, SQL ist im Filtern und Sortieren besser als js.
 - **/oneko:** Enthält oneko.js, eine animierte Katze, die der Maus folgt. ¹³

Verwendete Technologien

Technologie	Verwendung im Projekt
HTML5	Struktureller Aufbau der Benutzeroberfläche.
CSS3	Gestaltung der Anwendung über externe Stylesheets. Es werden drei verschiedene Themes (Pink, Hell, Dunkel) angeboten.
JavaScript	Implementierung clientseitiger Funktionen, einschließlich eines Theme-Wählers, Tabellensortierung/-filterung und der Katzen-Animation.
PHP 8.2	Serverseitige Logik, Verarbeitung von Formulardaten, Session-Handling und Datenbankkommunikation via PHP Data Objects
MySQL 8.0	Relationale Datenbank zur Speicherung der Benutzer- und der dynamisch erstellten Tabellen.
Docker	Containerisierung der Anwendung und ihrer Abhängigkeiten (Apache, PHP, MySQL) für eine portable und konsistente Ausführung.

Serverseitige Logik

Die serverseitige Logik wird vollständig mit PHP 8.2 abgewickelt. Die Datenbankverbindung

erfolgt über die PDO-Erweiterung, was eine Abstraktionsschicht und Schnittstelle für Datenbankoperationen gewährleistet. Zum Schutz vor SQL-Injection werden Prepared Statements verwendet, insbesondere bei der Verarbeitung von Benutzereingaben in Formularen. Auch Bobby Tables kann so keinen Schaden anrichten (xkcd.com/327).

Das Session-Management regelt den Zugriff auf geschützte Bereiche. Nach erfolgreicher Anmeldung wird eine Benutzersitzung erstellt, die die `user_id`, den `username` und die `role` speichert. Diese betrifft allerdings nur elementarstes RBAC, eine Unterscheidung zwischen Nutzer und Admin.

Datenbankstruktur

Die Anwendung nutzt eine relationale multi-tenant Datenbank.

1. **users-Tabelle:** Eine statische Tabelle zur Verwaltung der Benutzer. Sie wird beim ersten Start der Anwendung automatisch erstellt, falls sie nicht existiert. ²⁵
 - `id` (INT, Primary Key, Auto-Increment)
 - `username` (VARCHAR, UNIQUE)
 - `password` (VARCHAR, gehasht)
 - `role` (ENUM: 'user', 'admin')
 - `created_at` (TIMESTAMP)
2. **Dynamische Benutzertabellen:** Benutzer können eigene Tabellen erstellen. Um Datenkollisionen und unbefugten Zugriff zu verhindern, erhält jede vom Benutzer erstellte Tabelle ein Präfix, das sich aus seiner Benutzer-ID zusammensetzt (z.B. `u1_meine_tabelle`). Die Spalten und deren Datentypen (TEXT, INT, DATE, BOOLEAN) können vom Benutzer bei der Erstellung definiert werden.

Die Beziehung zwischen Benutzer und Tabelle ist 1:n, jede Tabelle kann nur zu einem Nutzer gehören.

2. Benutzerdokumentation

Systemanforderungen

Zur Ausführung des Projekts wird eine funktionierende Installation von **Docker** und **Docker Compose** benötigt. Es sind keine weiteren Abhängigkeiten erforderlich, da diese durch die Docker-Konfiguration bereitgestellt werden.

Installation und Inbetriebnahme

1. **Projektdateien beziehen:** Klonen des Repository oder Download der ZIP.
2. **Umgebungsvariablen konfigurieren:** Bearbeitung der `.env`, mit eigenen Werten.
3. **Anwendung starten:**

in einer shell (Ich nutze zsh):

`docker-compose up -d --build` für die alte Docker-Compose Version.

`docker compose up -d --build` für die neue Docker Compose Version.

Docker Desktop:

1. Image bauen, 2. compose starten.

4. **Anwendung aufrufen:** Die Webanwendung ist nun über die URL `http://localhost:8080` in einem beliebigen Webbrowser erreichbar, falls kein anderer Port spezifiziert wurde.

Bedienung der Anwendung

- **Login und Registrierung:**

- Beim ersten Start wird automatisch ein Administrator-Konto mit dem via Umgebungsvariablen definierten Passwort angelegt. Standard-zugangsdaten sind `admin/admin1`
- Benutzer werden bei nicht vorhandener Sitzung zur `login.php`-Seite weitergeleitet.

- **Tabellenverwaltung (Startseite):**

- Nach dem Login sehen Benutzer eine Liste ihrer erstellten Tabellen.
- Über das Formular "Neue Tabelle erstellen" können neue Tabellen angelegt werden, indem ein Name und bis zu fünf Spalten mit zugehörigen Datentypen definiert werden.
- Bestehende Tabellen können direkt von dieser Seite gelöscht werden.

- **Datenverwaltung:**

- Durch Klick auf einen Tabellennamen in der Liste gelangen Benutzer zur Detailansicht (`display_table.php`).
- Hier werden alle Einträge der Tabelle angezeigt.
- Die Schaltflächen **Bearbeiten** und **Löschen** können einzelne Datensätze bearbeiten
- Ein Formular am unteren Rand der Seite ermöglicht das Hinzufügen neuer Einträge.
- Die Tabelle verfügt über clientseitige Filter- und Sortierfunktionen pro Spalte für eine verbesserte Benutzerfreundlichkeit.

- **Admin-Funktionen:**

- Benutzer mit der Rolle `admin` haben erweiterte Rechte.
- Diese können die spezielle `users`-Tabelle einsehen und verwalten, um Benutzerdaten zu bearbeiten oder Benutzer zu löschen.
- Das Löschen eines Benutzers führt auch zur unwiderruflichen Löschung aller Tabellen, die dieser Benutzer erstellt hat. Dies geschieht über ein separates Statement, womöglich wäre ein **ON DELETE CASCADE** auch eine Möglichkeit gewesen.

- **Theming:**

- Im Header der Seite befindet sich ein Dropdown-Menü, mit dem zwischen einem Standard-, einem hellen und einem dunklen Design gewechselt werden kann.
- Die Auswahl wird im Browser des Benutzers gespeichert und bei zukünftigen Besuchen automatisch angewendet, auch im Login-Screen.