

# Projektaufgabe PHP-Grundlagen

## Einleitung

Zu erstellen ist ein Mini-Blog. Die Seite basiert auf PHP mit einer MariaDB-Datenbank. Die Seite kann mit CSS gestaltet sein. Wer mag kann auch ein Framework wie Bootstrap benutzen. **Sie muss nicht responsiv optimiert sein!**

Legen Sie für Ihr Projekt ein separates Projektverzeichnis an. Dort sind alle relevanten Dateien und Verzeichnisse unterzubringen! **Dateien, welche nicht zum Projekt gehören sind vor Abgabe aus diesem Verzeichnis zu entfernen!**

Abzugeben sind:

- Der komplette Projektordner (mit eigenem Namen bzw. Team-Namen benennen z.B. `max-mustermann-php.zip`) als Zip-Archiv
- Die Projektplanung

## Planung

In der Planungsphase machen Sie sich bitte Gedanken, welche Dateien Sie anlegen müssen, welche davon Ausgabe-Dokumente sind und welche davon nur inkludiert werden. Dateien, die inkludiert werden, sind mit der Endung `".inc.php"` (z.B. `functions.inc.php`) zu kennzeichnen und können in einem Extra-Verzeichnis abgelegt werden. Bei vielen Projekten hat es sich etabliert dem Bezeichner dieser Dateien einen Unterstrich voranzustellen (`_functions.inc.php`). **Die Startseite muss `index.php` heißen!**

Planen Sie eine zentrale Datei, in welcher sie Funktionen für ihr Projekt unterbringen und ein zentrales Dokument, welches die Verbindung zum Datenbankserver aufbaut. Die Angabe der Datenbank kann hier fest eingebaut werden (also nicht wie in unseren Beispiel-Datei über eine Konstante). Diese Verbindungs-Anweisungen können auch gut in einer `"init.inc.php"`-Datei mit weiteren zentralen Elementen (Start der Session, Festlegung der Funktionen zur Fehler- bzw. Ausnahmebehandlung etc.) untergebracht werden.

Planen Sie für jeden Anwendungsfall ein Ausgabe-Dokument. Mehrfach zu verwendende Elemente wie z.B. die Navigation, Header und Footer sollten Sie ebenfalls in je ein Include-Dokument zu platzieren, welches in die jeweiligen Ausgabe-Dokumente eingebunden wird. Machen Sie sich dazu gegebenenfalls eine Skizze aus welcher hervorgeht wie die Einbindungs-Hierarchie aufgebaut ist.

Planen Sie weiterhin die Syntax der Bezeichner von Dateien, Variablen, Konstanten, Funktionen etc. und halten Sie sich im Projekt strikt daran.

Planen Sie grob, wie die einzelnen Seiten auszusehen haben, z.B. wo das Bild bzw. die Meta-Daten bei einem Artikel ausgegeben werden sollen.

Planen Sie ebenfalls, welche Ausgabe-Seiten Sie benötigen (z.B. ob Sie in dem PHP-Dokument zum "Artikel bearbeiten" gleich noch die Option zum Löschen dieses Artikels unterbringen, oder dafür ein extra PHP-Dokument anlegen wollen.

Die Planung kann digital (Word-, PowerPoint-, Text-Datei, etc.) oder analog (Papier und Stift) abgegeben werden. Bitte daran denken, die Dokumente mit Ihrem Namen zu versehen. Die digitale Variante kann in das Zip-Archiv aufgenommen werden.

Es dürfen Teams gebildet werden. Team-Stärke höchstens vier Team-Mitglieder. Bei Team-Arbeit bekommt jedes Mitglied eine Aufgabe, die es umsetzt. Planungs-Konventionen sind von jedem Team-Mitglied zu beachten!

## Die Datenbank

Legen Sie eine Datenbank "miniblog" über PhpMyAdmin an. Sie benötigen 3 Datenbank-Tabellen, welche zueinander in Beziehung stehen.

### tbl\_users

<u>users_id</u>	int	Primary	A_I
users_forename	varchar(50)	NULL*	
users_lastname	varchar(50)		
users_sal	varchar(10)	(Anrede)	
users_email	varchar(100)	Unique	
users_password	varchar(255)		
users_create	timestamp	current_timestamp	

### tbl\_posts

<u>posts_id</u>	int	Primary	A_I
posts_users_id_ref	int	(Fremdschlüssel für Users-Tabelle)	
posts_categ_id_ref	int	(Fremdschlüssel für Kategorien-Tabelle)	
posts_header	varchar(50)	(Überschrift / Titel)	
posts_content	text		
posts_image	varchar(100)	NULL*	Pfadangabe zum Bilderordner und der Bild-Datei (z.B. /images/waldkasino.webp)
posts_created_at	timestamp	current_timestamp	
posts_updated_at	timestamp	on update current_timestamp	

### tbl\_categories

<u>categ_id</u>	int	Primary	A_I
categ_name	varchar(50)		
categ_desc	varchar(255)		

\* Das angegebene Feld darf NULL-Werte enthalten. Standard für diese Felder ist NULL.

## Aufbau des Blogs

### Allgemein

Jede Ausgabe-Seite besteht aus Header, Navigation, Hauptteil und Footer. Die Navigation kann dabei im Header untergebracht sein. Die vier Teile sollen dabei in je einer Datei untergebracht sein. Diese Dateien werden dann in einer zentralen Datei inkludiert.

### Startseite

Auf der Startseite sollen **die Titel** der einzelnen Posts **inkl. Angabe der Kategorie** aufgelistet werden. Das kann in einer Tabelle oder einer Liste umgesetzt werden. Die Titel sind als Links auszugeben. Das Link-Ziel ist hierbei die Seite für die Artikel-Ansicht eines einzelnen Artikels inkl. der Artikel-ID als URL-Query (siehe folgender Abschnitt, z.B. `post_single.php?post=4`).

### Artikel-Ansicht

Auf dieser Seite soll ein einzelner Artikel mit allen Infos angezeigt werden. Ist dem Artikel ein Bild zugeordnet, soll der HTML-Code für das Bild ausgegeben werden, ist kein Bild zugeordnet, soll dieser Code nicht ausgegeben werden. Im **Header-Bereich des Artikels** sollen die Meta-Daten ausgegeben werden (Autor, Kategorie, wann erstellt, wann geändert).

### User-Verwaltung

Erzeugen Sie eine Seite mit einem Registrierungs-Formular (Benutzername ist die E-Mail-Adresse). Diese Seite soll in der Navigation angezeigt werden, wenn der Nutzer **nicht** eingeloggt ist. Möchte sich ein User mit einer Mail-Adresse registrieren, welche schon vergeben ist, soll eine entsprechende Meldung ausgegeben werden. Beachten Sie dabei, dass die Mail-Adresse in der Datenbank den Unique-Index besitzt.

Erzeugen Sie ein Login-Formular in einem weiteren PHP-Dokument. Dieses ist in der Navigation ebenfalls sichtbar, wenn der Nutzer **nicht** eingeloggt ist. Hierbei vergleichen Sie die E-Mail-Adresse und das Passwort des Users. Wenn beides übereinstimmt, geben Sie eine entsprechende Meldung aus und speichern Sie die relevanten User-Daten (E-Mail, Vorname und Nachname) als Array im `$_SESSION`-Array ab. Außerdem werden die Menüpunkte für **Login und Registrierung** in der Navigation **durch** einen Menüpunkt auf die **Logout-Seite ersetzt**.

War der Login nicht erfolgreich wird ebenfalls eine Meldung ausgegeben, aber nichts in dem `$_SESSION`-Array gespeichert.

Denken Sie daran, dass in allen PHP-Dokumenten dazu am Anfang die Funktion `'session_start()'` eingefügt werden muss. Dies kann über ein zentrales `_init`-Dokument geschehen, welches in alle ausgebenden PHP-Seiten eingebunden wird.

In einer Logout-Seite soll die Session zerstört und eine Meldung über das erfolgreiche Logout ausgegeben werden. Eingeloggte User dürfen neue Artikel anlegen und bestehende Artikel bearbeiten oder löschen. Dazu gibt es in der auf der Startseite neben der Auflistung der Artikel je einen Link zum Bearbeiten und einen zum Löschen eines Artikels. Dieser Link führt zur Datei zum Artikel bearbeiten bzw. löscht den Artikel aus der Datenbank. Im Link-Ziel ist hier ebenfalls als URL-Query die Artikel-ID und die Aktion anzugeben (z.B.

`post_edit.php?post=3&action=edit`).



## Zusatzaufgabe (optional)

Bauen Sie auf der Startseite einen Filter nach Kategorien ein. Je nach ausgewählter Kategorie sollen nach der Filterung in der Artikel-Liste nur jene Artikel aufgelistet sein, die zu dieser Kategorie gehören. Legen Sie den Filter als Drop-Down-Liste an. **Die dort angegebenen Kategorien sollen aus der Datenbank ausgelesen und in der Liste ausgegeben werden.**

## Abgabe

Abgabetermin ist **Freitag, der 12.01.2024, 15:00 Uhr**. Das Projekt bitte als Zip-Archiv (Benennung `vorname-nachname-php.zip`) abgeben! Projektplanung nicht vergessen!

Viel Spaß!

## Hinweise

Wenn von URL-Query die Rede ist, dann meine ich damit eine URL, wie sie z.B. bei einem per `GET` abgesendetem Formular generiert wird. Die Query wird mit einem `"?"` von der URL getrennt und die einzelnen Schlüssel-Werte-Paare werden durch ein `"&"` getrennt. Diese Schlüssel-Werte-Paare stehen im Auswerte-Skript über das superglobale `$_GET`-Array zur Verfügung. Dieses funktioniert wie das `$_POST`-Array. Bei der URL `post_edit.php?post=3&action=edit` z.B. ist der Query-String `post=3&action=edit`. Im `$_GET`-Array gibt es dann den Schlüssel `post` mit dem Wert `3` und den Schlüssel `action` mit dem Wert `edit`. Auch hier ist zu beachten, dass alle Werte grundsätzlich als `String` vorliegen.

Auch bei diesem Projekt gilt: Qualität vor Quantität. Wenn ihr also einen Aspekt des Projektes nicht schafft, schreibt eine kurze Begründung **WAS** ihr und **WARUM** ihr es nicht geschafft habt. Das gilt allerdings nicht für die Planung. **Diese muss abgegeben werden.** Und **eine Planung wird am Anfang vor Beginn der Programmierarbeit fertiggestellt** und nicht am Ende, wenn das Projekt fertig ist!!!

**Projekt-Teile, die nicht fertig gestellt worden sind und deshalb nicht funktionieren bitte vor der Abgabe aus dem Projekt entfernen!**