

Dokumentation

für das Fach

WPFMW Moderne Webanwendungen mit HTML5, CSS3 und JavaScript

von

Andre Kasper (M.Sc.)

an der

Fachhochschule Köln
Campus Gummersbach
Fakultät für Informatik
und Ingenieurwissenschaften

erarbeitet von

Anatol Walger
(Matrikelnummer 11096667)

Tobias Szuminski
(Matrikelnummer 11096923)

Lukas Heblík
(Matrikelnummer 11098934)

31. August 2015

Inhaltsverzeichnis

1.	<u>EINLEITUNG</u>	<u>1</u>
2.	<u>PLANUNG</u>	<u>2</u>
3.	<u>DATENBANK</u>	<u>3</u>
3.1.	<u>DATENSTRUKTUR</u>	<u>3</u>
3.2.	<u>SQL-QUERY</u>	<u>4</u>
4.	<u>RESSOURCEN</u>	<u>5</u>
4.1.	<u>/INDEX</u>	<u>5</u>
4.2.	<u>/MANAGE</u>	<u>6</u>
4.3.	<u>/PROFILE</u>	<u>7</u>
4.4.	<u>/SETTINGS</u>	<u>7</u>
4.5.	<u>/SEARCH</u>	<u>7</u>
4.6.	<u>/LOGOUT</u>	<u>8</u>
4.7.	<u>CSS-FRAMEWORK BOOTSTRAP</u>	<u>8</u>
5.	<u>SERVERANFORDERUNGEN UND INSTALLATION</u>	<u>9</u>
5.1.	<u>SERVERANFORDERUNGEN</u>	<u>9</u>
5.2.	<u>INSTALLATION</u>	<u>9</u>
6.	<u>INFORMATION ÜBER DATEIEN DER GALERIE</u>	<u>10</u>
6.1.	<u>KLASSEN UND KLASSENSPEZIFISCHES</u>	<u>10</u>
6.2.	<u>SCHRIFTEN</u>	<u>10</u>
6.3.	<u>BILDER</u>	<u>10</u>
6.4.	<u>JQUERY, AJAX, JAVASCRIPT</u>	<u>10</u>
6.5.	<u>CSS, LAYOUT</u>	<u>10</u>
6.6.	<u>RESSOURCEN</u>	<u>11</u>
6.7.	<u>DATENBANK</u>	<u>11</u>
7.	<u>QUELLEN ZU VERWENDETEN HILFSMITTELN</u>	<u>12</u>

1. Einleitung

Diese Dokumentation wurde im Rahmen des Wahlpflichtfaches „WPFMW – Moderne Webanwendungen“ erstellt und enthält alle Überlegungen und Vorgehensweisen zur Realisierung einer modernen Bildergalerie, die folgende Anforderungen erfüllen sollte:

- mit Javascript realisiert
- Bilder werden beim Anklicken vergrößert und man kann in der Vergrößerung durch Pfeile von einem zum nächsten Bild wechseln
- Login-Bereich und Registrierung
- MySQL-Anbindung um die Nutzerdaten zu speichern und den Upload von neuen Bildern zu ermöglichen
- weitere Inhalte mit jQuery und AJAX

2. Planung

Bevor mit der Implementierung der Galerie begonnen werden konnte, musste zunächst eine angemessene Authentifizierungsmöglichkeit diskutiert werden.

Da die Projektanforderung ein Login-System voraussetzt, fiel die Wahl auf gleichberechtigte Benutzer, welche jeweils ihre eigenen Bilder im Login-Bereich verwalten und ansehen können.

3. Datenbank

Wie in der Projektanforderung beschrieben, wurde MySQL als Datenbank für die Galerie gewählt. In der Datenbank werden alle Benutzer und benutzergenerierten Inhalte, wie etwa hochgeladene Bilder und persönliche Daten, gespeichert. Die Datenbankabfragen erfolgen via PHP Data Objects (PDO).

3.1. Datenstruktur

Nachfolgend wird die Datenstruktur der Galerie beschrieben. Alle Felder sind mit dem UTF-8 Zeichensatz kodiert.

Benutzer: Ein Benutzer wird eindeutig durch seine ID identifiziert, die als Primärschlüssel für alle weiteren Daten des Benutzers dient.

Spalte	Typ
user_id	int(10)
username	varchar(25)
password	varchar(80)
email	varchar(60)
avatar	varchar(80)
timestamp	date
PRIMARY KEY	user_id
UNIQUE KEY	username
UNIQUE KEY	email

Galerie: Ein Bild wird eindeutig durch seine ID identifiziert, die als Primärschlüssel für alle weiteren Daten des Bildes dient. Um einem Benutzer alle seine Bilder zuzuordnen, wird die ID des Benutzers in der internen Repräsentation eines jeden Bildes gespeichert.

Spalte	Typ
id	int(10)
orig_path	varchar(80)
user_id	varchar(10)
thumb_path	varchar(80)
description	varchar(100)
title	varchar(60)
PRIMARY KEY	id

3.2. SQL-Query

Um die Datenstruktur der Galerie anzulegen, sind folgende SQL-Befehle in eine geeignete SQL-Datenbank zu importieren:

```
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
```

```
CREATE TABLE `gallery` (  
  `id` int(10) NOT NULL AUTO_INCREMENT,  
  `orig_path` varchar(80) DEFAULT NULL,  
  `user_id` int(10) NOT NULL,  
  `thumb_path` varchar(80) DEFAULT NULL,  
  `description` varchar(100) DEFAULT NULL,  
  `title` varchar(60) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `users` (  
  `user_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `username` varchar(25) COLLATE utf8_unicode_ci NOT NULL,  
  `password` varchar(80) COLLATE utf8_unicode_ci NOT NULL,  
  `email` varchar(60) COLLATE utf8_unicode_ci NOT NULL,  
  `avatar` varchar(80) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `timestamp` date NOT NULL,  
  PRIMARY KEY (`user_id`),  
  UNIQUE KEY `username` (`username`)  
  UNIQUE KEY `email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

4. Ressourcen

Nachfolgend werden alle für die Galerie notwendigen Ressourcen spezifiziert.

4.1. /index

Nicht eingeloggt

Sobald die Galerie in einem Web-Browser aufgerufen wird, gelangt man auf diese Ressource. Die Ressource stellt fest, ob eine aktive PHP-Session vorliegt und leitet den Benutzer bei erfolgreicher Prüfung in den Login-Bereich weiter. Fällt die Prüfung negativ aus, können die Registrierung und der Login durchgeführt werden.

Die Registrierung erfolgt durch Anklicken des Menüpunktes „Register“. In einem modalen Fenster werden „Username“, „E-Mail-Adresse“ und „Passwort“ abfragt und die Registrierung mit dem entsprechenden Button bestätigt, wobei die Methode „register_user“ der Klasse „user“ aufgerufen, das Passwort durch die PHP-Funktion „password_hash“ verschlüsselt wird und die Daten des Benutzers in der Datenbank gespeichert werden. Sobald die Registrierung abgeschlossen ist und keine Fehler aufgetreten sind, kann ein neuer Benutzer sich einloggen.

Das Login erfolgt durch Anklicken des Menüpunktes „Login“. In einem Dropdown-Feld füllt der Benutzer „Username“ und „Passwort“ aus, wobei der Login-Prozess durch den entsprechenden Button beginnt und die „Login“-Methode der Klasse „user“ aufgerufen wird.

Für alle Benutzer wird auf der Ressource durch eine AJAX-Abfrage eine Galerie mit den populärsten Bildern bei Flickr angezeigt. Durch Anklicken eines Bildes wird dieses in einer vergrößerten Ansicht angezeigt und es kann mit Pfeilen durch alle Bilder navigiert werden.

Eingeloggt

Nachdem der Login-Prozess abgeschlossen ist und eine aktive PHP-Session existiert, ändert sich die Ressource, um dem eingeloggten Benutzer weitere Interaktionsmöglichkeiten zu bieten und ihn von einem nicht eingeloggten Benutzer abzugrenzen.

Die Funktion zur Registrierung wird ausgeblendet und der Menüpunkt „Hallo (Name des Benutzers)“ bietet dem eingeloggten Benutzer die Möglichkeit, sein Profil anzusehen, seine Daten zu verwalten und sich auszuloggen. Außerdem wird die Suchfunktion aktiviert und der zusätzliche Menüpunkt „Bilder verwalten“ angezeigt.

Des Weiteren zeigt die Ressource oberhalb der Galerie mit den populärsten Flickr-Bildern alle vom Benutzer hochgeladenen Bilder als Thumbnails an. Durch Anklicken eines Bildes erscheint es als Overlay, durch welches mit Pfeilen navigiert werden kann.

4.2. /manage

In dem Bereich „Bilder verwalten“ kann der eingeloggte Benutzer Bilder zu seiner Galerie hinzufügen und bereits hochgeladene Bilder als Thumbnails und beim Anklicken in einem Overlay ansehen, durch das anhand von Pfeilen navigiert werden kann.

Der Upload eines Bildes funktioniert über eine AJAX-Abfrage an die Klasse „gallery“ durch den Aufruf der Methode „upload.“ Nachdem eine Datei ausgewählt wurde, ermöglicht der HTML5 FileReader eine Vorschau des jeweiligen Bildes, die anhand eines lokalen Pfades berechnet wird. Innerhalb der Vorschau können dem ausgewählten Bild ein Titel und optional eine Beschreibung hinzugefügt werden. Gibt der Benutzer keinen Titel an, wird der aktuelle Titel des Bildes gespeichert.

Durch das HTML5 „accept“ Attribut kann spezifiziert werden, dass der Upload von Dateien mit anderslautendem MIME-Type als „jpg“, „jpeg“, „gif“ oder „png“ blockiert wird. Um dennoch sicherzustellen, dass nur die angegebenen MIME-Types akzeptiert werden, erfolgt eine weitere Prüfung innerhalb der Klasse „gallery.“

Sobald der Benutzer den Upload-Button betätigt, wird das ausgewählte Bild im Erfolgsfall hochgeladen, auf dem Server gespeichert und der Verweis darauf in der Datenbank gesichert. Für jeden Benutzer wird auf dem Server ein neuer Ordner mit dessen Bildern angelegt. Ist der Upload beendet, wird das Bild via jQuery „Append“ an den Anfang des Arrays der aktuellen Bilder angefügt oder, falls der Upload fehlgeschlagen ist, eine Fehlermeldung ausgegeben.

Unterhalb des Upload-Formulars werden dem Benutzer die sich aktuell in seiner Galerie befindlichen Bilder angezeigt. Hier kann der Benutzer ein bestimmtes

Bild anhand des entsprechenden Symbols von dem Server und aus der Datenbank entfernen. Dazu wird ein AJAX Request an die Klasse „gallery“ gesendet, welche die entsprechende Methode besitzt, wobei in der Callback-Funktion das jeweilige Bild aus dem Array der aktuellen Bilder entfernt wird.

4.3. /profile

Diese Ressource ist für die Anzeige von benutzerspezifischen Informationen zuständig. Der Benutzer kann seine bei der Registrierung eingegebenen Daten und das Registrierungsdatum ansehen. Durch einen Verweis auf die Ressource „settings“ kann der Avatar bearbeitet oder gesetzt werden.

4.4. /settings

Diese Ressource ist für die Bearbeitung von benutzerspezifischen Informationen zuständig. Der Benutzer kann seine E-Mail-Adresse und sein Passwort ändern sowie einen Avatar hochladen oder bearbeiten, wobei die Klasse „user“ entsprechende Methoden besitzt.

Beim Upload eines Avatars wird, ähnlich der Ressource „manage“, ein jQuery AJAX-Request an die Klasse „user“ abgesetzt und, falls der Upload erfolgreich war, das Avatar-Bild dynamisch innerhalb der Callback-Funktion geändert, ansonsten eine Fehlermeldung ausgegeben. Auch hier wird von dem HTML5 „accept“ Attribut Gebrauch gemacht, welches sicherstellt, dass der Upload von Dateien mit anderslautendem MIME-Type als „jpg“, „jpeg“, „gif“ oder „png“ blockiert wird.

4.5. /search

Anhand dieser Ressource ist es möglich, nutzergenerierte Bilder nach einem bestimmten String zu durchsuchen. Dabei werden Titel und Beschreibung der Bilder berücksichtigt und die Ergebnisse als Thumbnails angezeigt. Zusätzlich erfolgt mit dem selben Suchbegriff eine Suche von Bildern bei Flickr, dessen Ergebnisse unterhalb der Ergebnisliste der eigenen Bilder des Benutzers angezeigt werden. Durch Anklicken eines der gefundenen Bilder wird es in einem Overlay geöffnet, durch das anhand von Pfeilen navigiert werden kann.

4.6. /logout

Diese Ressource ist für den Logout eines Benutzers zuständig und zerstört die aktive PHP-Session. Der Aufruf erfolgt durch den eingeloggten Benutzer im Menüpunkt „Hallo (Name des Benutzers)“, wobei die Methode „logout“ der Klasse „user“ aufgerufen wird.

4.7. CSS-Framework Bootstrap

Als Basis für die Galerie und zur Erleichterung des Entwicklungsprozesses wurde das Bootstrap 3 Framework von Twitter genutzt. Anhand dessen war es möglich, eine responsive Webseite mit dynamischer Struktur zu bauen, die plattformunabhängig auf jedem Endgerät funktioniert und korrekt angezeigt wird.

5. Serveranforderungen und Installation

5.1. Serveranforderungen

Um sicherzustellen, dass die Galerie korrekt installiert werden kann, muss der Server bestimmte Voraussetzungen erfüllen.

Die minimalen Anforderungen sind:

- PHP Version 5.5.14 oder höher
- MySQL Version 5.6.19 oder höher mit Unterstützung für PDO-Anfragen
- PHP GD Erweiterung
- Unterstützung von PHP Sessions

Die genannten Anforderungen sind verpflichtend, damit die Galerie korrekt installiert werden kann. Sollte eine der Anforderungen nicht erfüllt sein, können Fehler auftreten.

5.2. Installation

1. Download der neuesten Version der Galerie
(GitHub Link: <https://github.com/tolja/wpf-gallery>)
2. Hochladen der Dateien auf einen geeigneten Server
3. Erstellung einer geeigneten MySQL Datenbank mit benötigten Rechten
4. Anpassen der Datei „config.php“ im Verzeichnis „includes“ mit den Zugangsdaten zur MySQL Datenbank
5. Sicherstellung, dass der Ordner „images“ existiert und Vergabe von chmod „775“ Rechten auf diesen Ordner.

6. Information über Dateien der Galerie

6.1. Klassen und Klassenspezifisches

- classes/gallery.php
- classes/user.php
- process_delete.php
- process_upload.php

6.2. Schriften

- fonts

6.3. Bilder

- images

6.4. jQuery, AJAX, Javascript

- js/bootstrap.min.js
- js/flickr_search.js
- js/gallery.js
- js/gridify.js
- js/jquery.blueimp-gallery.min.js
- js/jquery.min.js
- js/script.js

6.5. CSS, Layout

- style/blueimp-gallery.min.css
- style/bootstrap-theme.min.css
- style/bootstrap.min.css
- style/main.css
- layout/header.php
- layout/footer.php
- layout/nav.php
- img

6.6. Ressourcen

- index.php
- includes/form_check.php
- logout.php
- manage.php
- profile.php
- search.php
- settings.php

6.7. Datenbank

- db.sql
- includes/config.php

7. Quellen zu verwendeten Hilfsmitteln

- Ausrichtung der Bilder anhand von Width und Height
(Miro Mannino: <http://miromannino.github.io/Justified-Gallery/>)
- Anzeige der Bilder in einem Overlay
(Sebastian Tschan: <https://github.com/blueimp/Gallery>)
- Script zur Anzeige der populärsten Flickr Bilder
(Sebastian Tschan: <https://github.com/blueimp/Gallery>)
- Bootstrap
(<http://getbootstrap.com>)