

Dokumentation Sport

Von Karin Achtnich

Als Übung für das Modul <Datenbank in einer Webseite einbinden>

30.12.2018

Inhalt

1	Management Summary	3
2	Information.....	3
2.1	Auftrag.....	3
3	Planung.....	3
3.1	Funktionen.....	3
3.1.1	Must-haves	3
3.2	Datenbank	3
3.2.1	Dropdownliste	4
3.2.2	User und deren Rechte.....	4
4	Realisation	5
4.1	Grundgerüst	5
4.2	Datenbank	5
4.2.1	Login	5
4.2.2	Hochladen.....	5
4.2.3	Suchen	5
4.2.4	Löschen.....	6
4.2.5	Registrieren	6
4.3	Videos.....	6
4.3.1	Allgemein.....	6
5	Kontrolle.....	7
5.1	Testdokument	7
5.1.1	Konstruktiv	7
5.1.2	Destruktiv	7
6	Auswertung	8
6.1	Zusammenfassung des Tests	8
6.2	Finale Worte	8
7	Verzeichnis	9
7.1	Bilder	9
7.2	Tabellen	9
7.3	Quellen	9

1 Management Summary

Dieses Projekt ist für die Fachschaft Sport von der Kantonsschule Frauenfeld. Es ist ein Portal, um Videos zu verwalten. Für die Datenbank soll man sqlite3 brauchen. Es hat die Funktionen Login, Datenbank Verwaltung, Videoübersicht und Slow Motion Wiedergabe. Dazu auch ist das Design Responsive. die Sprache ist Deutsch. Dieses Projekt ist eine Übung für ein Modul und wird nicht eingesetzt.

2 Information

2.1 Auftrag

Von der Fachschaft Sport wurde der Auftrag gegeben, um ein Portal zu erstellen, um Videos zu verwalten. Es sollte folgendes können:

- Responsive Design
- Login Verwaltung
- Hochladen Clips
- Verschlagwortung der Clips
- Verwaltung der Clips
- Suchen nach Clips
- Slow Motion – Wiedergabe

Dazu gib es auch noch ein Github-Archiv und eine Dokumentation

Das Github-Archiv ist: <https://github.com/KarinAchnich/Sport>

3 Planung

3.1 Funktionen

3.1.1 Must-haves

- Responsive Design
- Login Verwaltung
- Hochladen Clips
- Verschlagwortung der Clips
- Verwaltung der Clips
- Suchen nach Clips
- Slow Motion – Wiedergabe

3.2 Datenbank

Die Datenbank ist eine SQLite3 Datenbank. Diese hat 2 Tabellen darin, welche **nicht** miteinander verknüpft sind. Die Tabelle «Personen» brauche ich für das Login und die andere Tabelle «Tvideo» zum Speichern von Videos.

Tabelle 1 personen

personen			
Attribute	Type	Schlüssel	Beispiel Wert
username	Varchar(45)	Primary key	Gast
passwd	Varchar(45)		1
online	Char(1)		0

Tabelle 3 Tvideo

Tvideo			
Attribute	Type	Schlüssel	Beispiel Wert
id	integer	Primary key	1
filename	Varchar(45)		Karin
Sportart	Varchar(30)		Aikido
Abteilung	Varchar(10)		IMS
Klasse	char(1)		3
Jahr	year		2019
Semester	Char(2)		FS
Path	Varchar(45)		«Download/». \$filename.»mp4»

Tabelle 2 Sportart

Sportart
Vergessene
Akrobatik
Aikido
Badminton
Baseball
Beachvolleyball
Diskens Wurf
Eishockey
Federball
Fussball
Geräteturnen
Gymnastik
Handball
Hindernislauf
Hochsprung
Klettern
Kugelstoss
Leichtathletik
Orientierungslauf
Pilates
Quidditch
Reiten
Run and Bike
Schwimmen
Smolball
Staffellauf
Tanzen
Unihockey
Völkerball
Volleyball

3.2.1 Dropdownliste

Ich habe bei Sportart, Abteilung, Klasse, Jahr, Semester eine Dropdownliste erstellt. Dadurch kann ich mir ein Enum mit allen Werten ersparen. Bei der Sportart habe ich so 30 aufgelistet, welche man am meisten im Turnunterricht machen würde nach meiner Erfahrung nach. Es habe insgesamt etwa 500 verschiedene Sportarten gefunden. Da musste ich mich einschränken.

Tabelle 5 Jahr

2019
2020
2021
2022
2023
2024
2025

Tabelle 6 Semester

FS
FS

Tabelle 4 Abteilung

Abteilung
FMS
HMS
GMS
IMS
Andere

Tabelle 7 Klasse

1
2
3
4

3.2.2 User und deren Rechte

User	Password
Gast	1
Sudo su (=Student)	1
Lehrer	1

Es gibt bei den Personen genau drei User.

Tabelle 8 User

Diese haben folgende Rechte.

Tabelle 9 Rechte der User

X = erlaubt

User	Suchen & Anschauen	Hochladen	Löschen
Gast	x		
Sudo su	x	x	
Lehrer	x	x	x

4 Realisation

4.1 Grundgerüst

Das Grundgerüst für die Webseite (html, css) machte ich zuerst. Dort hatte ich den Fehler gemacht, dass ich den Input nicht in eine Form getan habe. Nachdem das gelöst war, fing ich mit den Funktionen an.

4.2 Datenbank

4.2.1 Login

Beim Login hatte ich folgende Probleme. Zuerst erstellte ich eine SQLite3 Datenbank. Die Datenbank hat in der Tabelle <Personen> drei Datensätze darin aka drei Benutzer (Gast, Lehrer, Sudo su). Für die Implementierung war es einfacher jedem das gleiche Passwort <1> zu geben. Auch in der Datenbank gespeichert ist, wer Online ist. Dies wird dargestellt mit <0> und <1>. Dort wo in der Datenbank <1> im Attribut <Online> steht, ist man angemeldet. Der Nachteil an dieser Methode ist, dass nur eine einzige Person angemeldet sein kann. Für mich funktioniert dieses System sehr gut. Ich werde es noch überarbeiten.

Es gab noch ein Problem mit Chrome. Chrome konnte die Seite nicht mehr finden beim Laden, da der <Charset> nicht definiert ist. mit viel herumtüfteln, konnte ich das Problem lösen.

4.2.2 Hochladen

Beim Hochladen musste ich zuerst zusätzliche Attribute auslesen und Tabellen erstellen. Diese habe ich dann in einem Dropdown implementiert. Zuerst habe ich den Test, wer eingeloggt ist von Login kopiert und entsprechend angepasst. Dies findet man unter <testlog()>.

Danach arbeite ich an <getotherInfo()>. Diese Funktion speichert alle zusätzlichen Attribute zum Video. Beim Namen ist der «Trigger», ob das Feld leer ist. Bei dem Dropdown teste ich, ob es gesetzt wurde. Ich gebe dann den Input zurück, damit ich sicher alles gespeichert habe.

Als nächstes speicherte ich die Info in die Datenbank mit der Funktion <intoDB(Attribute)>. Dort hatte ich das Problem, dass schon beim Laden der Seite den Eintrag <int, null,null....> gemacht wurde. Mit einer if-Schleife auf dem Namen konnte ich dies verhindern.

Nun kommt eines der Probleme, welches ich noch nicht gelöst habe. Die Funktion <getFile()> sollte das Video herunterladen. Jedoch konnte es das File nicht finden. Ich habe unter dem Ordner <upload>, <uploadtest2> und <useless test files> verschiedene Lösungen angeschaut und angepasst.

Da ich den Upload das File und die Datenbank getrennt habe und die Datenbank funktioniert, ging ich zur Suche hinüber.

4.2.3 Suchen

Bei der Suche habe ich die Aufgabe in verschiedene Abschnitte getrennt.

Zuerst einmal habe ich die Funktion <getotherInfo()> wieder gebraucht um den Input zu speichern. Mit der Funktion <searchDB()> habe ich zuerst alle Datensätze ausgegeben.

Der nächste Schritt war es die Where bedingung zu erstellen mit den Eingaben. Diese Funktion habe ich <factors()> genannt. Dabei habe ich 2 Arrays definiert. Einer hat alle Where-Bedingungen und der andere die Werte. Mit einem negativen Zähler konnte ich die Anzahl der ausgewählten Felder herausfinden. Danach addiere ich die Where-Bedingungen zusammen und nehme das letzte And weg. Diese füge ich am Ende in «searchDB()» ein.

4.2.4 Löschen

Beim Löschen habe ich zuerst den <testlog(>) wieder eingefügt und angepasst. Man kann nur ein Video löschen pro mal. Da ich beim Suchen auch die ID des Datensatzes ausgabe, weiss der User, welche Nummer man eingeben muss.

Danach habe ich den Input erhalten gemacht und die Funktion<outDB(>) erstellt. Diese löscht die Daten von der Datenbank. Jedoch geht der Knopf nicht so ganz und man muss nun per Enter die Funktion aufrufen.

4.2.5 Registrieren

Ich habe die Registrierfunktion beim Login gemacht. Ich habe ein neues Feld<gruppe> eingefügt in meiner Login Funktion. Somit kann man beim Registrieren Username, Gruppe und Passwort eingeben. Der Username kann nur einmal gebraucht werden. Ich habe alle andere <testlog(>) Funktionen entsprechend angepasst. Mit dem neuen System kann sich immer noch nur einer einloggen. Das hat sich nicht geändert. Ich dachte noch eine Profildarstellung oder eine <Passwort vergessen> Seite zu machen. Jedoch fehlt mir dazu die Zeit.

4.3 Videos

4.3.1 Allgemein

Ich habe mit Video js ein Video eingefügt. Jedoch zeigt der Browser nicht das ganze Bild an. Das Video läuft und man sogar das Video slow Motion abgeben. Der Ton kommt auch. Manchmal zeigt sich das Video von selbst. Ich habe nicht genug Zeit, um den Fehler zu finden. Ich ignorierte das Problem und ging weiter.

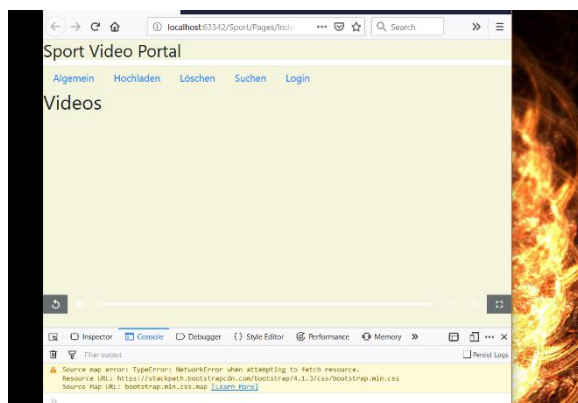


Abbildung 1 Normale Video anzeige

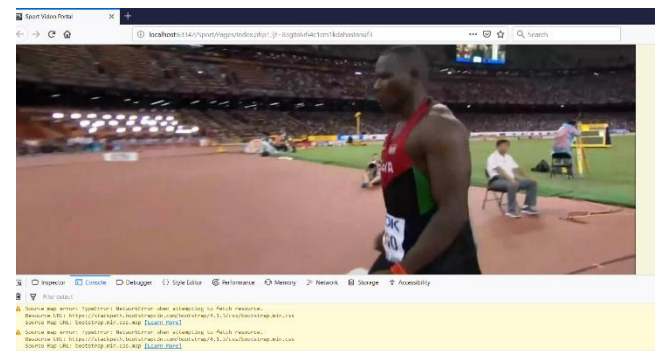


Abbildung 2 Video anzeige beim Fehler im Laden

Damit ich die Daten besser managen kann habe ich die Suchfunktion kopiert und hineingetan. Da ich aber die Videos nicht hochladen kann, bringt es mir aber sehr wenig nach Videos zu suchen. Der Pfad sollte ein Video laden und habe mal den Code dafür reingeschrieben. Jedoch habe ich es auskommentiert, damit nicht immer eine Fehlermeldung kommt.

5 Kontrolle

5.1 Testdokument

5.1.1 Konstruktiv

0	Testfall	Resultat	Funktion
1	Gehe vom Browser vom Vollbildschirm Modus zum Kleinbildschirm Modus auf der Seite Index.	Das Design verkleinert sich entsprechend dem Rahmen des Browsers.	Responsive Design
2	Kreiere den Benutzer: Username: Lehrer3, Gruppe: Lehrer, Passwort: 1 in der Registration und Loge dich anschliessend ein im Login.	Man kann sich einloggen im Portal.	Login Verwaltung
3	Lade auf der Seite Hochladen ein Video rauf mit Folgenden Angaben: Name: Aikido Sport: Aikido Abteilung: IMS Klasse: 3 Jahr: 2019 Semester: HS Gehe auf dem ersten Submit Knopf, um die Sachen zu speichern.	Der Datensatz wird hochgeladen. Doch das Video kann nicht hochgeladen werden.	Hochladen Clips
4	Gehe auf die Seite Suchen und gebe den Sport Aikido ein und drücke Suchen.	Das Video Aikido wird als Datensatz angezeigt im unteren Feld.	Verschlagwortung der Clips
5	Gehe auf Allgemein und gebe den Sport Aikido ein.	Der Datensatz wird angezeigt für das fiktionale Video.	Verwaltung der Clips
6	Gehe auf Suchen und suche nach Abteilung IMS und Jahr 2019.	Der Datensatz Aikido wird angezeigt.	Suchen nach Clips
7	Spiele das Testvideo ab und verlangsame das Video.	Das Bild wird nicht angezeigt, aber man kann das Video slow Motion anzeigen lassen.	Slow Motion – Wiedergabe

Tabelle 10 Konstruktive Testfälle

5.1.2 Destruktiv

0	Testfall	Resultat	Funktion
1	Kreiere den gleichen Account im Login wie bei Fall 2 im Konstruktiven Fall.	Es gibt eine Fehlermeldung, dass der Name schon vergeben ist.	Login Verwaltung
2	Lösche den Datensatz (Nummer vom Datenbankeintrag von Aikido) zwei Mal hintereinander.	Das Video kann nicht gelöscht werden	Verwaltung der Clips
3	Gebe beim Löschen ein Buchstabe ein.	Es geht nicht, da der Input nur Zahlen akzeptiert	Verwaltung der Clips

Tabelle 11 Destruktive Testfälle

6 Auswertung

6.1 Zusammenfassung des Tests

Tabelle 12 Zusammenfassung des Tests

Funktion	Geht	Problem	Lösung
Responsive Design	Ja/Nein		
Hochladen Clips	Ja/Nein	Das Video kann nicht hochgeladen werden	Beispiel von Hochladefile besser anschauen
Login Verwaltung	Ja/Nein	Kein Löschen, und kein Verändern von Profil. Nur eine Person kann online sein.	Löschen und Verändern einfach zu lösen mit mehr Zeit. Online Problem via Session Variablen Lösen.
Verschlagwortung der Clips	Ja		
Verwaltung der Clips	Ja/Nein	Die Clips werden nicht vollständig im Bild angezeigt.	Ich habe keine Ahnung.
Suchen nach Clips	Ja		
Slow Motion – Wiedergabe	Ja		

6.2 Finale Worte

Die Video-Anzeige hat mich vollkommen aus dem Zeitplan geschleudert und habe deswegen mich auf die Datenbank Funktionen konzentriert. Auch das Hochladen von Clips hat mir sehr Mühe gegeben. Ich habe einen Ansatz gefunden, das Problem zu beheben. Hätte ich mehr Zeit würde ich das noch machen. Ich bin so weit gekommen wie ich konnte in dieser Zeit. Jedoch bin ich nicht zufrieden mit dem Resultat.

Würde ich das Projekt weiterziehen würde ich die Login Verwaltung zuerst fertig machen und danach das Hochladen bereinigen. Die richtige Videoanzeige habe ich ehrlich gesagt aufgegeben.

Dies war definitiv nicht eine gute Leistung.

7 Verzeichnis

7.1 Bilder

Abbildung 1 Normale Video anzeige	6
Abbildung 2 Video anzeige beim Fehler im Laden	6

7.2 Tabellen

Tabelle 1 personen	3
Tabelle 3 Tvideo	4
Tabelle 2 Sportart	4
Tabelle 4 Abteilung	4
Tabelle 5 Jahr	4
Tabelle 6 Semester	4
Tabelle 7 Klasse	4
Tabelle 8 User	4
Tabelle 9 Rechte der User	4
Tabelle 10 Konstruktive Testfälle	7
Tabelle 11 Destruktive Testfälle	7
Tabelle 12 Zusammenfassung des Tests	8

7.3 Quellen

<http://www.w3programmers.com/file-upload-and-download-with-php/> (30.12.2018, 23.00)

<https://www.php-einfach.de/experte/php-codebeispiele/kontaktformular/> (30.12.2018, 23.00)

<https://stackoverflow.com/questions/21271060/file-upload-php-files-undefined-index-error>
(30.12.2018, 23.00)

<https://videojs.com/getting-started/#download-cdn> (30.12.2018, 23.00)

https://www.w3schools.com/html/html5_video.asp (30.12.2018, 23.00)