

Suche:

Beliebte Filme

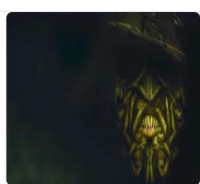


Bewertung

73%

Black Adam

2022-10-19



Bewertung

58%

Jeepers Creepers: Reborn

2022-09-15



Bewertung

67%

Halloween Ends

2022-10-12



Bewertung

68%

Orphan: First Kill

2022-07-27



Bewertung

55%

Project Gemini

2022-01-06



Bewertung

73%

Fall

2022-08-11

FILME & SERIEN WEBSEITE

Projektdokumentation: 3. Lehrjahr

Exposee

Für das dritte Jahr in der Berufsschule erstellen wir im Rahmen der Projektarbeit eine Webseite für Filme & Serien mit verschiedenen Funktionen

Kenan Cakici, Martin Dang, Lars Tröster

Klasse: E3FI – Lehrer: Herr Huber – 23.10.2022

Inhalt

Abkürzungsverzeichnis	2
1. Information.....	3
1.1 Ausgangslage	3
1.2 Projektbegründung.....	3
2. Projektziel / Soll-Zustand.....	3
3. Projektplanung	4
3.1 Machbarkeitsanalyse.....	4
3.2 Verwendete Sprachen und Anwendungen	4
3.2.1 HTML	4
3.2.2 PHP	4
3.2.3 CSS	5
3.2.4 Bootstrap	5
3.2.5 JavaScript.....	5
3.2.6 XAMPP	5
3.2.7 MySQL.....	6
3.3 Zeitplanung.....	6
4. Durchführung & Projektverlauf	7
4.1 Woche 1 13.09.2022 – 18.09.2022	7
4.2 Woche 2 19.09.2022 – 25.09.2022	8
4.2.1 API-Dokumentation von TMDB (Auszug)	9
4.3 Woche 3 26.09.2022 – 02.10.2022	10
4.4 Woche 4 03.10.2022 – 09.10.2022	12
4.5 Woche 5 10.10.2022 – 16.10.2022	14
4.6 Woche 6 17.10.2022 – 23.10.2022	16
5. Fazit	17
5.1 Soll-/Ist-Vergleich	17
5.2 Tätigkeitsnachweis / Zeitaufwand.....	17
5.3 Was haben wir aus dem Projekt gelernt?	17
5.4 Ausblick.....	18
6. Eidesstattliche Erklärung	19
Quellenverzeichnis	20

Abkürzungsverzeichnis

API	
Application Programming Interface	4
CSS	
Cascading Style Sheets	5
cURL	
Curl URL Request Library	10
HTML	
Hypertext Markup Language	4
IMDb	
Internet Movie Database	4
JSON	
JavaScript Object Notation	14
MySQL	
Vornamen My, den die Tochter des MySQL-AB-Mitbegründers Michael Widenius trägt und Structured Query Language.....	6
PHP	
Hypertext Preprocessor.....	4
TMDB	
The Movie Database.....	4
XAMPP	
Abkürzung für Apache, MySQL, Perl und PHP. Das X steht für verschiedene Betriebssysteme	5

1. Information

1.1 Ausgangslage

Im Rahmen der Projektarbeit für das dritte Lehrjahr in der Berufsschule, mussten wir uns am 13.09.2022 zum Start des Schuljahres, ein Thema überlegen. Dafür hatten wir den ganzen Tag zur Verfügung. Nach gründlichem Nachdenken haben wir uns schlussendlich entschieden, eine Webseite für Filme und Serien zu erstellen, da Martin bereits Erfahrungen durch seinen Ausbildungsbetrieb in dem Bereich sammeln konnte und Lars sich schon privat damit auseinandergesetzt hat.

Angefangen hat das Ganze mit einer Kurzbeschreibung des Projekts, bevor das Thema überhaupt von Herr Huber akzeptiert wurde. Die Kurzbeschreibung bestand aus ein paar Sätzen zum Vorhaben und den W-Fragen. Der grobe Zeitplan für das ganze Projekt beträgt 40 Stunden. Dankbarerweise standen uns dafür die kompletten Schultage vom 13.09.2022 bis zum 18.10.2022 zur Verfügung. Der 25.10.2022 stand als Tag der Präsentation und der 24.10.2022 als der letzte Abgabetermin für die Projektdokumentation fest.

1.2 Projektbegründung

Wieso und wozu braucht man so eine Seite? Aus persönlichen Erfahrungen, sowie von Freunden und Familien aus unserem Umfeld wissen wir, dass bei dem Überangebot an Streaming-Diensten schnell die Übersicht verloren werden kann. Besonders, weil die Anbieter sich die Lizenzen nur für einen bestimmten Zeitraum holen und die angebotenen Inhalte nach dessen Ablauf nicht mehr zum Abruf bereitstehen. Es wäre also überaus praktisch alle Filme und TV-Serien auf einer Webseite vereint zu haben und zu sehen, welche/r Serie/Film wo und wie lange verfügbar ist. Wenn zusätzlich dazu noch die Möglichkeit besteht sich anzumelden und eine eigene Watchlist zusammenzutragen, hilft das enorm den Überblick zu behalten.

2. Projektziel / Soll-Zustand

In unseren Köpfen hatten wir uns die Funktionen der Webseite dementsprechend wie folgt ausgemalt:

- Der Anwender soll durch verschiedene Filter- und Sortieroptionen nach Filmen und Serien suchen können
- Die Anwender sollen sich registrieren und somit einen eigenen Benutzer anlegen können
- Der Benutzer soll sich Filme und Serien in eine Watchlist hinzufügen können
- Der Benutzer soll Filme und Serien bewerten können (5 Sterne System)
- Die Seite soll ein Suchfeld haben, dass als Ergebnisse alle Treffer liefert, welche das eingegebene Wort beinhalten
- Die Seite soll den Anwendern zeigen, auf welchen Streaming-Plattformen (Netflix, Amazon Prime, Disney+ etc.) die Inhalte verfügbar sind
- Auf der Startseite soll eine bestimmte Auswahl an Filmen angezeigt werden (z.B. populär, am besten bewertet etc.)
- Die Filme und Serien sollen eine Detailansicht mit mehr Informationen wie eine Beschreibung und Mitwirkende bekommen, wenn der Anwender draufklickt

3. Projektplanung

3.1 Machbarkeitsanalyse

Die erste und wahrscheinlich wichtigste Frage, die wir uns stellen mussten, war, von wo bekommen wir all die Informationen über die unzähligen Filme und Serien? Es existieren bereits riesige Datenbanken wie das IMDb, welche sämtlichen Informationen beinhaltet. Allerdings konnten wir in diesem Fall nicht ohne weiteres auf diese Infos zugreifen und waren gezwungen nach einer Alternative zu suchen. Am Ende der Suche fanden wir die Antwort mit TMDb. Diese Datenbank bzw. Seite bietet nämlich eine API an, die genau die Funktionen liefert, die unsere Seite haben soll.

Eine API ist eine Schnittstelle, welche u.a. aus einem Satz von Befehlen und Funktionen besteht, die wir als Programmierer verwenden können, um eine Software oder wie in diesem Projekt eine Webseite zu erstellen. Sie verbindet Soft- und Hardwarekomponenten, Anwendungen und Benutzeroberflächen. Durch sie kann man mit externen Systemen interagieren, sprich der Inhalt zwischen verschiedenen Programmen oder Webseiten kann ausgetauscht werden. Somit stehen uns die rund 955.000 Einträge an Filmen und Serien auf TMDb zum Abruf bereit. TMDb stellt zur ihrer API auch eine Dokumentation mit ihren Methoden bereit, welche wir für unsere Webseite benutzt haben. Durch die vorhandenen Methoden können wir uns Arbeit und Zeit sparen, weil sie zum Großteil schon die gewünschten Funktionen beinhalten. Für den kostenlosen Zugriff für den Key war lediglich ein Benutzerkonto notwendig und die Bedingung die API nicht kommerziell zu nutzen. Den Key brauchen wir, um den Zugriff von uns zu verifizieren und die Daten von TMDb zu erhalten.

3.2 Verwendete Sprachen und Anwendungen

3.2.1 HTML

HTML ist eine textbasierte Auszeichnungssprache. Darunter versteht man eine Sprache, die für Maschinen lesbar ist und für die Formatierung von Texten und anderen Daten dient. Sie besteht aus öffnenden und schließenden „Tags“, wie z.B. `Beispiel`. HTML macht es für Browser möglich, den „Code“ zu interpretieren, anzuzeigen, sowie Webseiten zu verknüpfen. Wir benutzen den aktuellen Standard HTML 5. Das eignet sich besonders für suchmaschinenoptimierte und userorientierte Webseiten.

3.2.2 PHP

PHP ist eine weit verbreitete Skriptsprache, die sich speziell für das Programmieren von Webseiten eignet. Sie ist open source und kann somit frei verwendet werden.

Grundsätzlich wird die Sprache so benutzt, dass Anweisungen geschrieben werden und in HTML eingebettet werden. Der Code steht zwischen den Anweisungen „`<?php`“ und „`?>`“, welche zum Starten und Beenden dienen.

PHP ist auf serverseitige Skripte fokussiert, d.h. der Code wird auf dem Server ausgeführt generiert dort eine HTML-Ausgabe, die an den Client gesendet wird. Der Client erhält somit nur das Ergebnis und es ist nicht möglich den eigentlichen Code herauszufinden.

Die Skriptsprache eignet sich trotz großem Umfang an Funktionen für Neueinsteiger, weil die Verwendung recht einfach ist.

3.2.3 CSS

Bei CSS handelt es sich um eine Gestaltungs- und Formatierungssprache, die es einem ermöglicht das Design von HTML-Dokumenten zu gestalten. Mit ihr lässt sich also das Layout anpassen und nicht der Inhalt. Beispiele: Schriftart, Farben, Linien etc. CSS und HTML existieren getrennt voneinander, sprich ein HTML-Dokument lässt sich auch ohne CSS lesen. CSS ist ermöglicht es Browsern die Inhalte optisch ansprechender darzustellen.

Eine Anweisung kann beispielsweise so aussehen: `h1 {color:red; font-size:11px;}`

Es ist möglich CSS durch interne und externe Stylesheets einzubinden. Bei externen Stylesheets wird es in der HTML-Datei in einen „link-Tag“ geschrieben und bei internen Stylesheets wird ein „style-Element“ eingefügt.

Dadurch das mit HTML die Struktur vorgegeben wird und mit CSS die Gestaltung angepasst wird, ergänzen sich die beiden perfekt.

3.2.4 Bootstrap

Bootstrap ist ein Frontend-Framework, mit der verschiedene Arten von Webseiten erstellt werden können. Es ist ein Open-Source-Projekt und bietet unterschiedliche Gestaltungsvorlagen an, welche auf HTML und CSS basieren.

Wir haben es dazu verwendet, um das Design für den Login zu bauen.

3.2.5 JavaScript

JavaScript ist eine Skriptsprache, welche clientseitig arbeitet. Mit ihr lassen sich Funktionalitäten von Webseiten erhöhen, indem die Inhalte verändert oder generiert werden. JavaScript kann in HTML eingebunden werden und ist daher nützlich, um HTML und CSS zu ergänzen. Dafür kann es zwischen den „Tags“ geschrieben werden, in einer externen Datei abgelegt werden oder als ein HTML-Link eingebaut werden. Beim Anbinden muss darauf geachtet werden, dass das Skript fehlerfrei ist, damit Bugs vermieden werden.

3.2.6 XAMPP

XAMPP bietet uns als Komplettlösung eine Testumgebung mit allen wichtigen Komponenten an. Dazu gehört ein Apache-Server und eine MYSQL Datenbank. Durch diese Komponenten brauchen wir keinen Webserver, da XAMPP uns damit einen lokalen Server zur Verfügung stellt, auf dem wir testen können. Das Programm dient auch zur Kontrolle einer lokalen Server-Umgebung.

3.2.7 MySQL

MySQL ist eines der verbreitetsten relationalen Open-Source-Datenbankverwaltungssysteme. Es ist auch als kommerzielle Version verfügbar und bildet die Grundlage für dynamische Webseiten. Damit lassen sich Daten in Tabellen verwalten, anzeigen, speichern und ändern.

Eine relationale Datenbank teilt Daten in mehrere Tabellen (Speicherbereiche) auf, anstatt alles in einer Speichereinheit zusammenzufassen, weil bei einer Einheit Probleme wie doppelte Daten oder eine schlechte Organisation auftreten können. MySQL ist zudem schnell, stabil und benutzerfreundlich.

3.3 Zeitplanung

Wir haben die Aufgaben untereinander in drei ungefähr ähnlich zeitaufwendige Hauptaufgaben aufgeteilt:

- Webseite anhand der API bauen
- Benutzerprofile und Login bauen
- Projektorganisation & Dokumentation schreiben

Das heißt nicht, dass jeder nur für sich gearbeitet hat, aber dadurch hatte jeder immer eine Aufgabe und Beschäftigung.

Projektphase	Geplant (Stunden)
1. Aussuchen des Projekts & Kurzbeschreibung	4
2. Zielsetzung & Anforderungen	3
3. API suchen und Suchanfrage bauen	4
4. Profil & Benutzerlogin mit PHP bauen	8
5. Suchfeld einbauen	2
6. Suchergebnisse auf mehrere Seiten aufteilen	3
7. Detailansicht einbauen	4
8. Bewertungsmöglichkeit für Benutzer bauen	5
9. Header für die Webseite gebaut	1
10. Registrierung, Anmeldung & Profil integriert	2
11. Optische Gestaltung mit CSS/Bootstrap	3
12. Planung & Organisation	4
13. Projektdokumentation	10
Arbeitszeit gesamt	53

4. Durchführung & Projektverlauf

Das waren die Aufgaben, die wir in den sechs Wochen angegangen sind:

4.1 Woche 1 | 13.09.2022 – 18.09.2022

Am ersten Berufsschultag des dritten Jahres stand nach dem organisatorischen Kickoff die Projektarbeit auf dem Plan. Wir haben uns schnell als Gruppe gefunden und überlegt, was für ein Thema sich für das 40-Stündige Projekt eignet. Durch die Erfahrungen von Martin in der Webentwicklung in seinem Ausbildungsbetrieb und mit dem Hintergrund, dass sich Lars bereits privat damit beschäftigt hat, fiel die Entscheidung eine Webseite zu entwickeln.

In der Kurzbeschreibung hatten wir folgende Ideen aufgeschrieben, die uns direkt in den Sinn gekommen sind:

- Anhand einer API sollen Filme/Serien auf einer Webseite aufgelistet werden
- Es soll einen Benutzerlogin geben
- Der Benutzer soll Bewerten können
- Der Anwender soll durch verschiedene Filter nach Filmen/Serien suchen können
- Es soll angezeigt werden wie lange die Filme/Serien auf den Streaming-Plattformen zum Abruf bereitstehen bzw. auf welchen Plattformen sie verfügbar sind

Dementsprechend hatten wir uns die Ziele und Anforderungen gesetzt, die unter dem Projektziel / Soll-Zustand aufgelistet sind.

4.2 Woche 2 | 19.09.2022 – 25.09.2022

Zu Beginn haben wir nach einer API gesucht und erst bei IMDb geprüft, ob wir auf ihre API zugreifen können, weil das die größte Filmdatenbank ist. Dafür ist allerdings ein AWS (Amazon Web Service) Account notwendig und für diesen Account muss eine Kreditkarte hinterlegt werden, die wir nicht haben. Aus diesem Grund haben wir diese Möglichkeit gestrichen und nach einer anderen API umgesehen. Die zweite Anlaufstelle war die Seite <https://rapidapi.com/>, dessen APIs waren aber stark begrenzt und haben nicht die gewünschten Funktionen oder Dokumentationen geliefert. Schließlich fanden wir TMDb, welche eine API anbietet, für die man lediglich einen Account benötigt und die auch gut dokumentiert ist.

Die ersten Versuche mit der TMDb API waren, Filme ausgeben zu lassen und uns vertraut zu machen. Somit konnten wir testen und entscheiden, ob es für das Projekt von Nutzen ist.

Für das Suchfeld haben wir die Seite www.w3schools.com zu Rate gezogen. Dort haben wir uns an dem Formular für das Suchfeld und den Button orientiert. Hierbei hat das Suchfeld ein Label mit dem Text Suche bekommen. Das Textfeld hat einen placeholder mit einer angepassten Größe von 50. Der hidden-Wert ist dafür da, um bei der Suche die erste Seite mitauszugeben. Das alles wurde in ein div-Tag gepackt, damit es mit CSS designed werden kann.

```
print
<div class="searchbar">
<form>
<label for="suche">Suche:</label>
<input type="text" id="suche2" name="suche2" placeholder="Suche nach TV Show, Filme und Personen" size="50">
<input type="hidden" value="1" name="page">
<input type="submit" value="suchen">
</form>
</div>
';
}
```

Die Streamingdienste bei den einzelnen Filmen und Serien anzeigen zu lassen hat leider nicht funktioniert, weil keine Antwort vom Server kam.

4.2.1 API-Dokumentation von TMDb (Auszug)

THE MOVIE DATABASE The Movie Database API 3
<https://api.themoviedb.org/3> OAS RAML Support

Select a different version
Filter sections...

GETTING STARTED

- Introduction
- Authentication
- Daily File Exports
- Languages
- Images
- Image Languages
- Regions
- External IDs
- Popularity
- Request Rate Limiting
- JSON & JSONP
- Append To Response
- Search & Query For Details

ACCOUNT

AUTHENTICATION

CERTIFICATIONS

CHANGES

COLLECTIONS

COMPANIES

CONFIGURATION

CREDITS

DISCOVER

FIND

GENRES

Getting Started

Introduction

Welcome to version 3 of The Movie Database (TMDb) API. Below you will find a current list of the available methods on our movie, tv, actor and image API. If you need help or support, please head over to our [API support forum](#).

To register for an API key, click the [API link](#) from within your account settings page. You can also view the screenshots below for help:

1. Click on your avatar or initials in the main navigation ([screenshot](#))
2. Click the "Settings" link ([screenshot](#))
3. Click the "API" link in the left sidebar ([screenshot](#))
4. Click "Create" or "click here" on the API page ([screenshot](#))

Please note that the API registration process is *not optimized* for mobile devices so you should access these pages on a desktop computer and browser.

Before being issued an API key you will have to agree to our terms of use. You can read that [here](#).

A few useful tips...

- The [configuration methods](#) are useful to get the static lists of data we use throughout the database. You can find things like the languages, countries, timezones and translations that we use. The configuration method also holds useful image information.
- Understanding the basics of our authentication is useful. You can read about this [here](#).
- We enforce rate limiting on the API. You can read about that [here](#).

API-Dokumentation: <https://developers.themoviedb.org/3>

4.3 Woche 3 | 26.09.2022 – 02.10.2022

Um weitere Informationen zu einem Film oder zu einer Serie zu bekommen, wollten wir eine Detailansicht bauen. Diese soll sich öffnen, sobald auf einen Film oder eine Serie geklickt wird. Mit der Ansicht werden weitere Details wie das Genre oder das Erscheinungsdatum angezeigt. Der Benutzer wird dann auch die Möglichkeit haben eine Bewertung in Form von Sternen (1 bis 5) abzugeben.



Pokémon Heroes

2002-07-13 - Animation Adventure Action Fantasy

Benutzerbewertungen:



Handlung

Ash, Pikachu and his friends try and stop a pair of thieves hiding out in the canals and alleyways of Altomare, the age-old water capital. Joining the adventure are two new legendary Pokémon, a pair of siblings named Latias and Latios, who serve as peacekeepers and protectors of the Soul Dew - a priceless treasure with a mysterious power.

Die API-Anfrage mit cURL sieht wie folgt aus:

```
if (isset($_REQUEST["tvdetails"])) {
    $curl = curl_init();
    curl_setopt_array($curl, [
        CURLOPT_URL => 'https://api.themoviedb.org/3/tv/' . $_REQUEST["tvdetails"] . '?&language=de',
        CURLOPT_RETURNTRANSFER => true,
        CURLOPT_FOLLOWLOCATION => true,
        CURLOPT_ENCODING => "",
        CURLOPT_MAXREDIRS => 10,
        CURLOPT_TIMEOUT => 30,
        CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
        CURLOPT_CUSTOMREQUEST => "GET",
        CURLOPT_HTTPHEADER => [
            "Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiIyNmQ4NWUyZWZjNDY1IiwiaXNjaWkiOiJyZmJk40WmNkZG9yOTQ1MDA3YmFjYTNlZCIsInNjb3JlcyI6WyJhcGlfcmlhZGFuZCJd",
            "Content-Type: application/json; charset=utf-8"
        ],
    ]);
    $response = curl_exec($curl);
    $err = curl_error($curl);

    curl_close($curl);

    if ($err) {
        echo "cURL Error #: " . $err;
    } else {
        $test = json_decode($response, associative: true);
    }
}
```

cURL ist eine Bibliothek, durch welche man eine Verbindung zu verschiedensten Servern herstellen kann. Diese Verbindung ist dazu da, um Daten von aus dem Server abzufragen und somit im Code zu Verwenden. Das Tool ermöglicht neben der Dateiübertragung die Fernsteuerung und das Testen von Websites.

In diesem HTML-Teil werden die Daten, wie das Bild und der Text aus der API-Anfrage ausgegeben:

```

$mediatype="tv";
print '<div class="container" id="zweite_spalte">
<div class="row">
<div class="col-4">

</div>
<div class="col-8">

<h3><strong>.$test["name"].</strong></h3>

<div class="unten">.$test["first_air_date"];print' - ' ; foreach($test["genres"] as $value){print $value["name"]. " ;"; print}</div>

<div>Benutzerbewertungen: </div>
<div><div class="progress">

<div class="progress-bar" role="progressbar" style="width: '.substr($test["vote_average"], offset: 0, length: 1).substr($test["vote_average"], offset: 2, length: 1).%" aria-valuenow="25" aria-valuemin="0

<div><span class="fa fa-star" id="1" onClick="checked(1)" onMouseover="hover(1)" onMouseleave="hoverout(1)"></span>
<span class="fa fa-star" id="2" onClick="checked(2)" onMouseover="hover(2)" onMouseleave="hoverout(2)"></span>
<span class="fa fa-star" id="3" onClick="checked(3)" onMouseover="hover(3)" onMouseleave="hoverout(3)"></span>
<span class="fa fa-star" id="4" onClick="checked(4)" onMouseover="hover(4)" onMouseleave="hoverout(4)"></span>
<span class="fa fa-star" id="5" onClick="checked(5)" onMouseover="hover(5)" onMouseleave="hoverout(5)"></span></div>
<div class="texttrun"><h5>Handlung</h5></div>
<div>.$test["overview"].</div>
<div></div>
</div>
</div>
</div>';

```

Weil das Suchergebnis nur eine lange Seite mit allen Ergebnissen geliefert hat, mussten wir eine Anzeige mit einer Seitenanzahl einbauen.

Die aktive Seite wird als active angezeigt und leuchtet blau und wenn die Seitenanzahl am Anfang ist, kann nicht auf Previous geklickt werden, weil der Button ausgeblendet wird. Das gleiche für die letzte Seite und dem Next Button.

```
print '<nav aria-label="Page navigation example">'
<ul class="pagination justify-content-center">
    print '<li class="page-item">;if($_REQUEST["page"] == 1){print "disabled";} print "<a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . $ _REQUEST["suche2"]' .
    print $_REQUEST["page"] = $_REQUEST["page"] - 1;
    print "< >Previous</a ></li>";
    $_REQUEST["page"] = $_REQUEST["page"] + 1;
    print "<li class="page-item">";
    if ($_REQUEST["page"] == 1) {
        print "active";
    }
    print "<a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . ' &page=1">1</a></li>";
    if ($_REQUEST["page"]==1 and $_test["total_pages"]>1){
        print "<li class="page-item"><a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . ' &page=2">2</a></li>";
    }elseif($_REQUEST["page"]==2){
        print "<li class="page-item"><a class="page-link active" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . ' &page=2">2</a></li>";
        print "<li class="page-item"><a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . ' &page=3">3</a></li>";
    }elseif($_REQUEST["page"]>2){
    }
    print "<a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . ' &page=3">3</a></li>";
    <li class="page-item">";
    if ($_REQUEST["page"] == 4) {
        print "active";
    }
    print "<a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_REQUEST["suche2"]' . ' &page=' . $_test["total_pages"]' . '&#x27; . $_test["total_pages"]' . '</a></li>";
    print '<li class="page-item">;if($_REQUEST["page"] == $_test["total_pages"]){print "disabled";} print "<a class="page-link" href="http://localhost/FilmSerienWebsite/index.php?suche2=' . $_
    print $_REQUEST["page"] = $_REQUEST["page"] + 1;
    print "< >Next</a ></li>";
    $_REQUEST["page"] = $_REQUEST["page"] - 1;
    print' </li>
</ul>
</nav>
```

Previous **1** 2 3 4 Next

4.4 Woche 4 | 03.10.2022 – 09.10.2022

Damit Bewertungen abgegeben werden und eine Watchlist erstellt werden kann, benötigen wir einen Benutzer bzw. ein Profil. Deshalb soll sich der Anwender, falls er noch keinen Benutzer hat, registrieren können und ansonsten anmelden können.

Anmelden

Benutzername
Passwort

Absenden

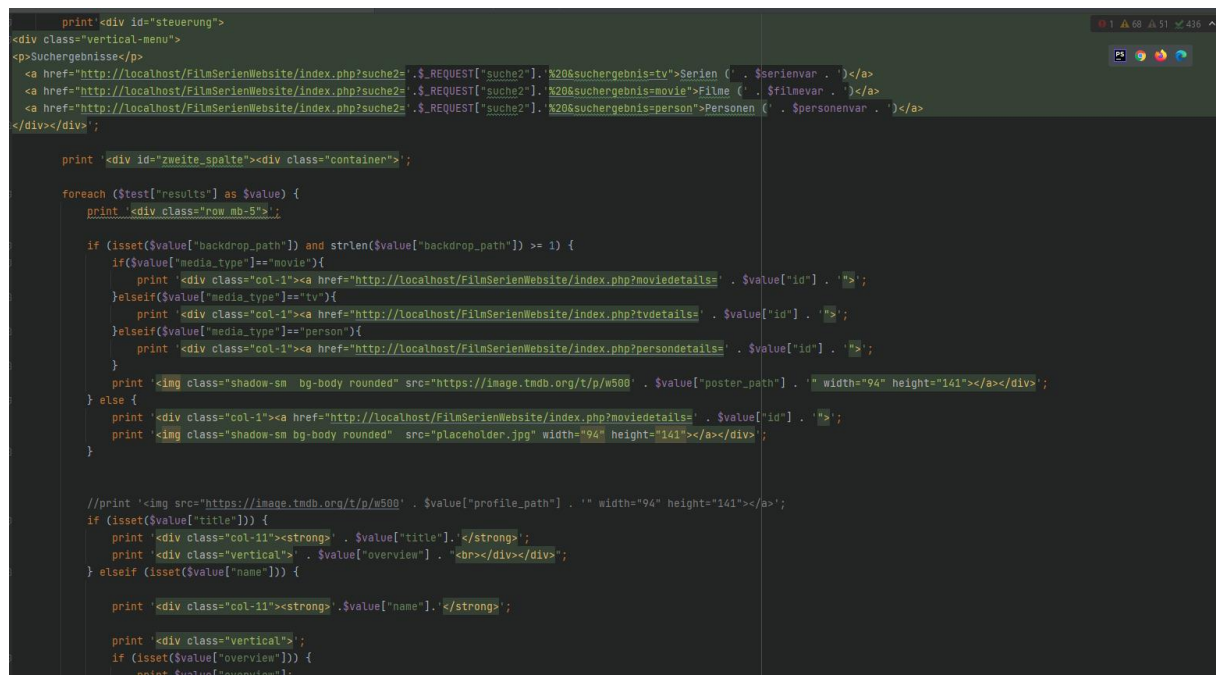
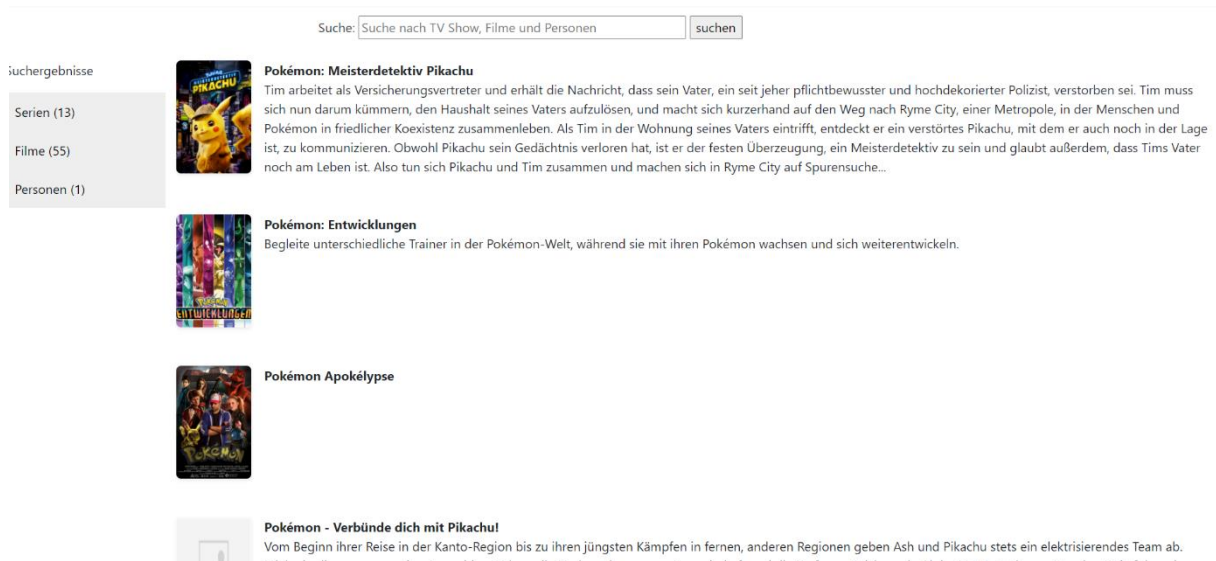
Doch [registrieren](#)?

Den Benutzerlogin haben wir mit der Skriptsprache PHP gebaut. Die Benutzerdaten haben wir in einer MySQL Datenbank abgespeichert. Auf diese konnten wir mit der PHP-Erweiterung mysqli im Code zugreifen.

Beispiel-Select, Überprüfung der Anmeldedaten:

```
$vorhanden = $mysqli->prepare("SELECT id from benutzer where benutzername =? and password = ?");  
$vorhanden->bind_param('ss', $benutzername, $password);  
$vorhanden->execute();  
$search_result = $vorhanden->get_result();
```

Nach einer Suche wurden die Bilder anfangs viel zu groß angezeigt. Aus diesem Grund musste die Auflösung angepasst. Gleichzeitig wurde links ein Filter nach Filmen, Serien und Personen eingebaut, welches nach der Suche angezeigt wird.



4.5 Woche 5 | 10.10.2022 – 16.10.2022

Die Bewertungsmöglichkeit und die Watchlist haben wir durch die Arbeitsaufteilung separat gebaut. Beim Integrieren in die Webseite hat es aber zu Problemen geführt und die abgegebene Bewertung wird nicht in die Datenbank übergeben, da JavaScript die Variable nicht an PHP übergibt.



Pokémon: Meisterdetektiv Pikachu

2019-05-03 - Action Abenteuer Fantasy

Benutzerbewertungen:



69%

Handlung

Tim arbeitet als Versicherungsvertreter und erhält die Nachricht, dass sein Vater, ein seit jeher pflichtbewusster und hochdekoriertes Polizist, verstorben sei. Tim muss sich nun darum kümmern, den Haushalt seines Vaters aufzulösen, und macht sich kurzerhand auf den Weg nach Ryme City, einer Metropole, in der Menschen und Pokémon in friedlicher Koexistenz zusammenleben. Als Tim in der Wohnung seines Vaters eintrifft, entdeckt er ein verstörtes Pikachu, mit dem er auch noch in der Lage ist, zu kommunizieren. Obwohl Pikachu sein Gedächtnis verloren hat, ist er der festen Überzeugung, ein Meisterdetektiv zu sein und glaubt außerdem, dass Tims Vater noch am Leben ist. Also tun sich Pikachu und Tim zusammen und machen sich in Ryme City auf Spurensuche...

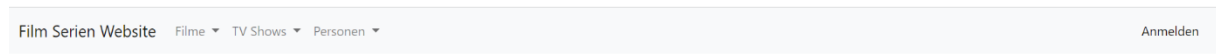
```
<span class="fa fa-star" id="1" onclick="checked(1)" onmouseover="hover(1)" onmouseleave="hoverout(1)"></span>
<span class="fa fa-star" id="2" onclick="checked(2)" onmouseover="hover(2)" onmouseleave="hoverout(2)"></span>
<span class="fa fa-star" id="3" onclick="checked(3)" onmouseover="hover(3)" onmouseleave="hoverout(3)"></span>
<span class="fa fa-star" id="4" onclick="checked(4)" onmouseover="hover(4)" onmouseleave="hoverout(4)"></span>
<span class="fa fa-star" id="5" onclick="checked(5)" onmouseover="hover(5)" onmouseleave="hoverout(5)"></span>
```

Die Daten werden über die fetch-Methode in JavaScript an die PHP-Datei übergeben:

```
fetch("bewertung.php", {
  body: "data=1",
  headers: {
    "Content-Type": "application/x-www-form-urlencoded",
  },
  method: "post"
})
.then(response => response.text())
.then(text => console.log(text))
break;
```

Die fetch-Methode fragt Daten von einem Server an oder sendet diese dem Server zu. Es kann auch Text- oder JSON-Dateien wie CSV-Dateien auslesen.

Für die Webseite wurde ein Header eingebaut, welches für die Navigation notwendig ist. Damit kann der Anwender auch schneller nach Filmen/Serien suchen, die z.B. nach der Popularität sortiert sind.



Für den Header haben wir eine Bootstrap 5 Navigationsbar als Vorlage genommen und für unsere Zwecke angepasst:

```
<title>Film Serien Website</title>
</head>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="http://localhost/FilmSerienWebsite/index.php">Film Serien Website</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Filme
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="http://localhost/FilmSerienWebsite/index.php?popularmovies">Beliebte Filme</a></li>
          </ul>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            TV Shows
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="http://localhost/FilmSerienWebsite/index.php?popularartv">Beliebte TV Shows</a></li>
          </ul>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Personen
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="http://localhost/FilmSerienWebsite/index.php?popularperson">Beliebte Personen</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

```
if(isset($_SESSION['benutzer'])){
  print '<li class="nav-item dropdown">';
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
    //Select für benutzernamen
    $benutzerid = $_SESSION['benutzer'];
    $search_username = $mysqli->prepare("SELECT benutzernamen from benutzer where id =?");
    $search_username->bind_param('s', $benutzerid);
    $search_username->execute();
    $username_result = $search_username->get_result();

    while ($row = $username_result->fetch_assoc()) {
      $username = $row['benutzernamen'];
    }
    print $username;

    print '</a>';
    <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
      <li><a class="dropdown-item" href="http://localhost/FilmSerienWebsite/index.php?profil=true">Profil anzeigen</a></li>
      <li><a class="dropdown-item" href="http://localhost/FilmSerienWebsite/index.php?einstellung">Einstellungen</a></li>
      <li><a class="dropdown-item" href="http://localhost/FilmSerienWebsite/index.php?abmelden=true">Abmelden</a></li>
    </ul>
  </li>';
} else {
  print '<a href="http://localhost/FilmSerienWebsite/index.php?anmelden=true"><button type="button" class="btn btn-light">Anmelden</button></a>';
}
print '</div>';
```


4.6 Woche 6 | 17.10.2022 – 23.10.2022

Für die optische Gestaltung der Webseite haben wir mit Bootstrap 5 Vorlagen gearbeitet. Zu den jeweiligen HTML-Objekten mussten wir dazu Klassen hinzufügen.

Wie in diesem Beispiel vom Registrieren:

```
<body class="text-center">
  <main class="form-signin">
    <form action="" method="post">
      <h1 class="h3 mb-3 fw-normal">Registrieren</h1>

      <div class="form-floating">
        <input class="form-control" type="text" id="benutzername1" name="benutzername" placeholder="Benutzername">
        <label for="benutzername1">Benutzername</label>
      </div>

      <div class="form-floating">
        <input class="form-control" type="password" id="password1" name="password1" placeholder="Passwort">
        <label for="password1">Passwort</label>
      </div>

      <div class="form-floating">
        <input class="form-control" type="password" id="password2" name="password2" placeholder="Passwort wiederholen">
        <label for="password1">Passwort wiederholen</label>
      </div>

      <input class="w-100 btn btn-lg btn-primary" type="submit" name="absenden" value="Absenden">
      <p class="mt-5 mb-3 text-muted">Doch <a href="anmelden.php">anmelden</a>?</p>
    </form>
  </main>
</body>
</html>
```

Wir mussten diese aber auch mit CSS individueller gestalten.

Da der Benutzerlogin separat gebaut wurde, haben wir das nach der Fertigstellung in den Code der Webseite erfolgreich integriert.

An der Projektdokumentation wurde jede Woche erweitert und in der letzten Woche vollendet. Da wir noch lange am Projekt gearbeitet haben, war die Dokumentation das letzte, was fertig gestellt wurde.

5. Fazit

5.1 Soll-/Ist-Vergleich

Am Ende kommt das Ergebnis unseren anfänglichen Vorstellungen sehr nahe, auch wenn wir es optisch gerne noch schöner gestaltet hätten und noch mehr Funktionen bzw. Inhalte eingebaut hätten.

Ein paar Ziele, wie das Anzeigen der Streamingdienste, sowie das „Speichern“ der Bewertungen und der Watchlist haben wir leider nicht wie gewünscht funktioniert.

5.2 Tätigkeitsnachweis / Zeitaufwand

Wir haben am Ende 7 Stunden mehr Zeitaufwand gehabt, als wir anfangs geplant hatten. Dabei waren ein paar Aufgaben weniger zeitintensiv als gedacht, andere dafür umso mehr. Insgesamt haben wir 60 Stunden am Projekt gearbeitet. Das heißt die Zeit in der Berufsschule hat uns nicht komplett gereicht und wir haben uns online getroffen, um daran weiterzuarbeiten.

Projektphase	Geplant (Stunden)	Stunden	Differenz
1. Ausschuchen des Projekts & Kurzbeschreibung	4	4	0
2. Zielsetzung & Anforderungen	3	4	-1
3. API suchen und Suchanfrage bauen	4	7	-3
4. Profil & Benutzerlogin mit PHP bauen	8	10	-2
5. Suchfeld einbauen	2	1	1
6. Suchergebnisse auf mehrere Seiten aufteilen	3	3	0
7. Detailansicht einbauen	4	2	2
8. Bewertungsmöglichkeit für Benutzer bauen	5	3	2
9. Header für die Webseite gebaut	1	1	0
10. Registrierung, Anmeldung & Profil integriert	2	2	0
11. Optische Gestaltung mit CSS/Bootstrap	3	3	0
12. Planung & Organisation	4	5	-1
13. Projektdokumentation	10	15	-5
Arbeitszeit gesamt	53	60	-7

5.3 Was haben wir aus dem Projekt gelernt?

Bei der Projektorganisation haben wir gemerkt, dass der Arbeitsaufwand deutlich reduziert werden kann, wenn zumindest eine Person sich darum kümmert und den Überblick der anzugehenden Punkte hat, was wir dann auch so gemacht haben. Wir haben uns oft untereinander ausgetauscht, damit jeder auf dem laufenden bleibt. Mit voranschreitender Zeit mussten wir Aufgaben priorisieren, welche die Seite unbedingt gebraucht hat und welche eher optional sind. Was wir hier gelernt haben, vor allem in Hinsicht auf unseren Abschluss, ist, dass eine Planung, Organisation und Priorisierung der Aufgaben die Grundvoraussetzung für ein aufwendigeres Projekt sind.

5.4 Ausblick

In der Zukunft würden wir noch viel mehr Zeit in das Design investieren, weil die optische Gestaltung einer Webseite heutzutage wichtig ist. Die Seite soll beim ersten Einblick bereits einladend aussehen und die Navigation sollte intuitiv sein.

Das Benutzerprofil würden wir gerne erweitern, so dass die User Beiträge erstellen können, unter Filmen und Serien Kommentare hinterlassen können und ihr Profil anpassen können (z.B. Profilbild oder Beschreibung).

Die Webseite soll den Benutzern noch mehr Optionen und Möglichkeiten bieten, wie z.B. eigene Listen erstellen, Ranglisten, einbinden von Film-Trailern und mehr Filter- / Sortieroptionen.

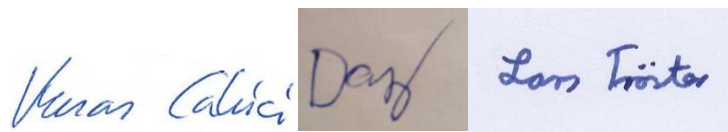
6. Eidesstattliche Erklärung

Wir, Kenan Cakici, Martin Dang, Lars Tröster, versichern hiermit, dass wir unsere Dokumentation zur schulischen Projektarbeit mit dem Thema

Filme & Serien Webseite

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben, wobei wir alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet haben. Die Arbeit wurde bisher keiner anderen Prüfungsinstanz vorgelegt und auch nicht veröffentlicht.

Reutlingen, den 23.10.2022

The image shows three handwritten signatures in blue ink. From left to right: 'Kenan Cakici', 'Dang', and 'Lars Tröster'. The signatures are written on a light-colored background.

Kenan Cakici, Martin Dang, Lars Tröster

Quellenverzeichnis

<https://developers.themoviedb.org/3/getting-started/introduction>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://getbootstrap.com/>

<https://www.php.net/manual/de/intro-what-is.php>

<https://www.php.net/manual/de/intro-what-cando.php>

<https://www.seo-kueche.de/lexikon/html/>

<https://de.ryte.com/wiki/JavaScript#:~:text=Bei%20JavaScript%20handelt%20es%20sich,CSS%20und%20HTML%20zu%20erg%C3%A4nzen.>

<https://www.ionos.de/digitalguide/websites/webdesign/was-ist-css/>

<https://www.checkdomain.de/hosting/lexikon/css/>

<https://www.ionos.de/digitalguide/websites/web-entwicklung/bootstrap-tutorial-die-ersten-schritte/#:~:text=Bootstrap%20ist%20ein%20Frontend%2DFramework,%2C%20sowie%20optionale%20JavaScript%2DERweiterungen.>

https://praxistipps.chip.de/was-ist-xampp-einfach-erklart_41910

<https://www.checkdomain.de/hosting/lexikon/mysql/>

<https://kinsta.com/de/wissensdatenbank/was-ist-mysql/>