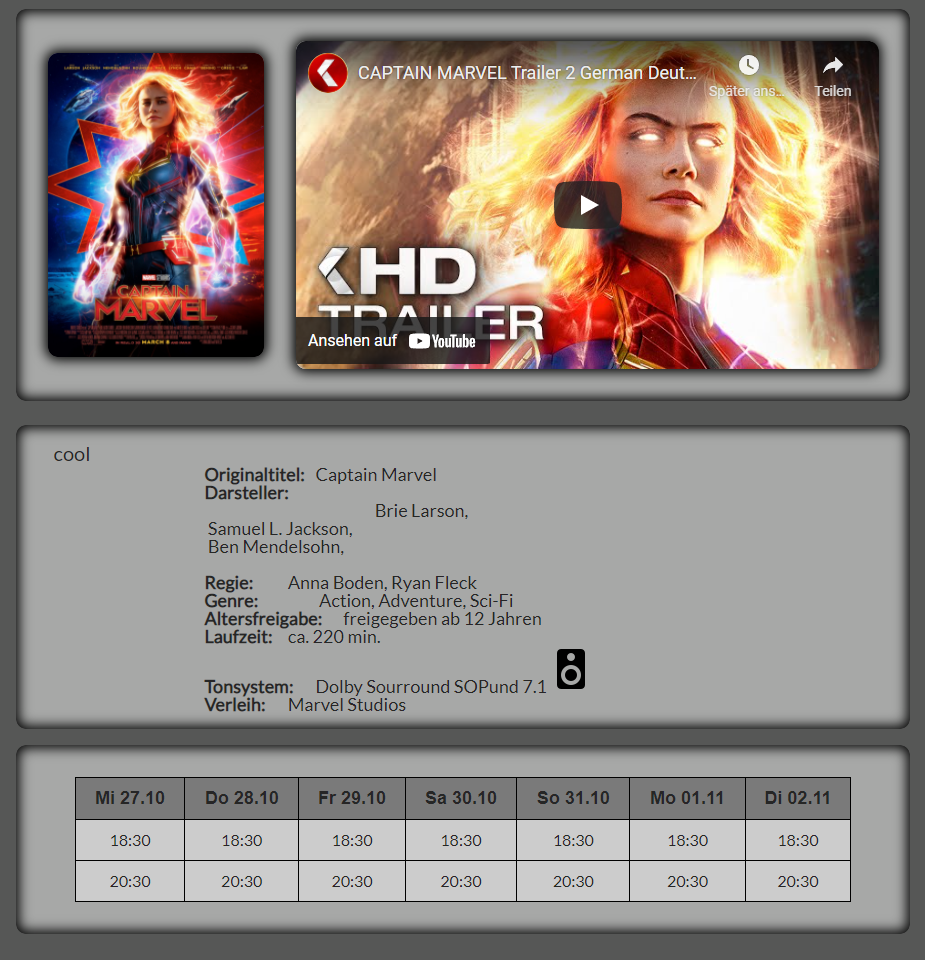
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung



**Kino Webseite (aus Besuchersicht)**

**14.12.2021**

Lukas Haiber, Jakob Meyer

Inhaltsverzeichnis

[Projektauftrag 3](#_Toc90279538)

[Zusammenfassung 3](#_Toc90279539)

[Details 3](#_Toc90279540)

[Termin Vorgaben 3](#_Toc90279541)

[Projektstrukturplan 4](#_Toc90279542)

[Organisationschart 4](#_Toc90279543)

[Planung 5](#_Toc90279544)

[Zielsetzung 5](#_Toc90279545)

[Beispiele existierender Ansätze 5](#_Toc90279546)

[Ergebnisse 5](#_Toc90279547)

[Strukturlayout / Featureset 7](#_Toc90279548)

[Nutzen des Strukturplans 8](#_Toc90279549)

[Aufgabenverteilung 8](#_Toc90279550)

[Aufgabenblöcke 8](#_Toc90279551)

[Verteilung 8](#_Toc90279552)

[Durchführung 9](#_Toc90279553)

[Genutzte Technologien 9](#_Toc90279554)

[Repository 9](#_Toc90279555)

[Inhalte 9](#_Toc90279556)

[Verwendete Tags/Elemente 10](#_Toc90279557)

[Design 11](#_Toc90279558)

[Color Scheme 11](#_Toc90279559)

[Flexbox 11](#_Toc90279560)

[Datenbank Anbindung 12](#_Toc90279561)

[Test-Server 13](#_Toc90279562)

[Entwicklungsumgebung 13](#_Toc90279563)

[Codeausschnitte 13](#_Toc90279564)

[Aktivitäten 13](#_Toc90279565)

[Anhänge 16](#_Toc90279566)

[Glossar 16](#_Toc90279567)

[Quellen 16](#_Toc90279568)

# Projektauftrag

## Zusammenfassung

Das Projekt (HTML, CSS, PHP) soll in einer Zeitspanne von ca. 12-16 SW Stunden realisiert werden. Eine Erarbeitung auch außerhalb des Unterrichts wird angestrebt. Die Aufgabe umfasst das Projekt selbst, eine abschließende Präsentation mit Fragen sowie eine ca. 10 bis 15-seitge Ausarbeitung, die einen wissenschaftlichen Charakter haben soll. Die Struktur und der Aufbau obliegen den jeweiligen Gruppen.

## Details

## Termin Vorgaben

30.11.2021 Labor, Erarbeitung in den Projektgruppen

01.12.2021 Labor, Erarbeitung in den Projektgruppen

14.12.2021 Labor, Letzter Termin zur Erarbeitung

Abgabe von Dokumentation und Projekt (bis 18:00 Uhr)

15.12.2021 Halten der Präsentation (13:45 bis 15:30)

# Projektstrukturplan

# Organisationschart

# Planung

## Zielsetzung

Das Ziel unserer Gruppe für dieses Projekt ist es, eine Kino-Webseite zu erstellen. Sie soll die Ansicht des Kunden darstellen und diesem ermöglichen das Kino-Programm einzusehen, sowie Karten zu buchen. Des Weiteren soll zu jedem Film eine Detail-Ansicht aufrufbar sein, welche dem Kunden weitere Informationen wie Darsteller, Altersfreigabe und Laufzeit bieten kann. Während des Buchungsprozesses gibt der Kunde seine Daten an und wählt den Sitzplatz aus.

Für die gesamte Webseite gilt, dass die individuellen Daten per PHP von einer Datenbank geladen werden sollen. Dies jedoch nur statisch, da es sich nicht um ein reelles Projekt und somit um Testdaten handelt.

## Beispiele existierender Ansätze

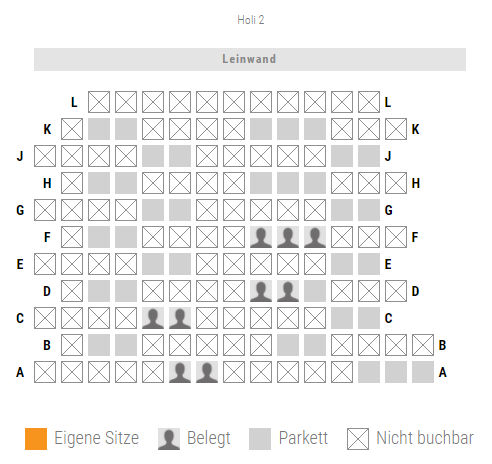
### Ergebnisse

Dank der Analyse hatten wir eine genauere Vorstellung der Abläufe auf einer typischen Kino-Webseite. Dadurch konnten wir besser Planen welche unter Seiten wir benötigen und welche Komponenten diese beinhalten sollen. Außerdem gaben uns die Beispiele Ideen für das Design und Layout unserer Webseite.

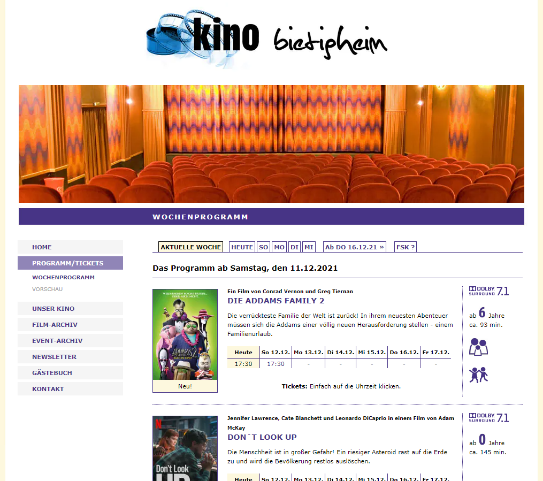
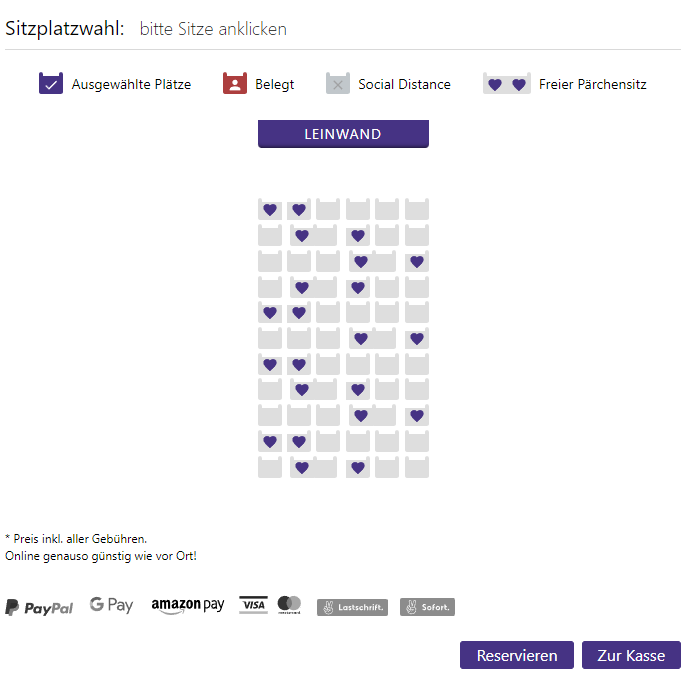
Somit haben wir uns zum Beispiel dazu entschieden den Kunden nicht direkt mit den Informationen zu den Filmen zu überhäufen, sondern erst nach Auswahl und der damit verbundenen Interessenbekundung. Des Weiteren haben wir beschlossen das Programm direkt auf der Startseite anzuzeigen, da dies der Fokus bei Kino Webseiten sein sollte.

Ein weiterer Bereich, bei dem uns das vorherige Betrachten von „Konkurrenz“ Webseiten geholfen hat, ist bei der Wahl der Interessanten Informationen wie beispielsweise auf der Detailseite die Daten zum Film.

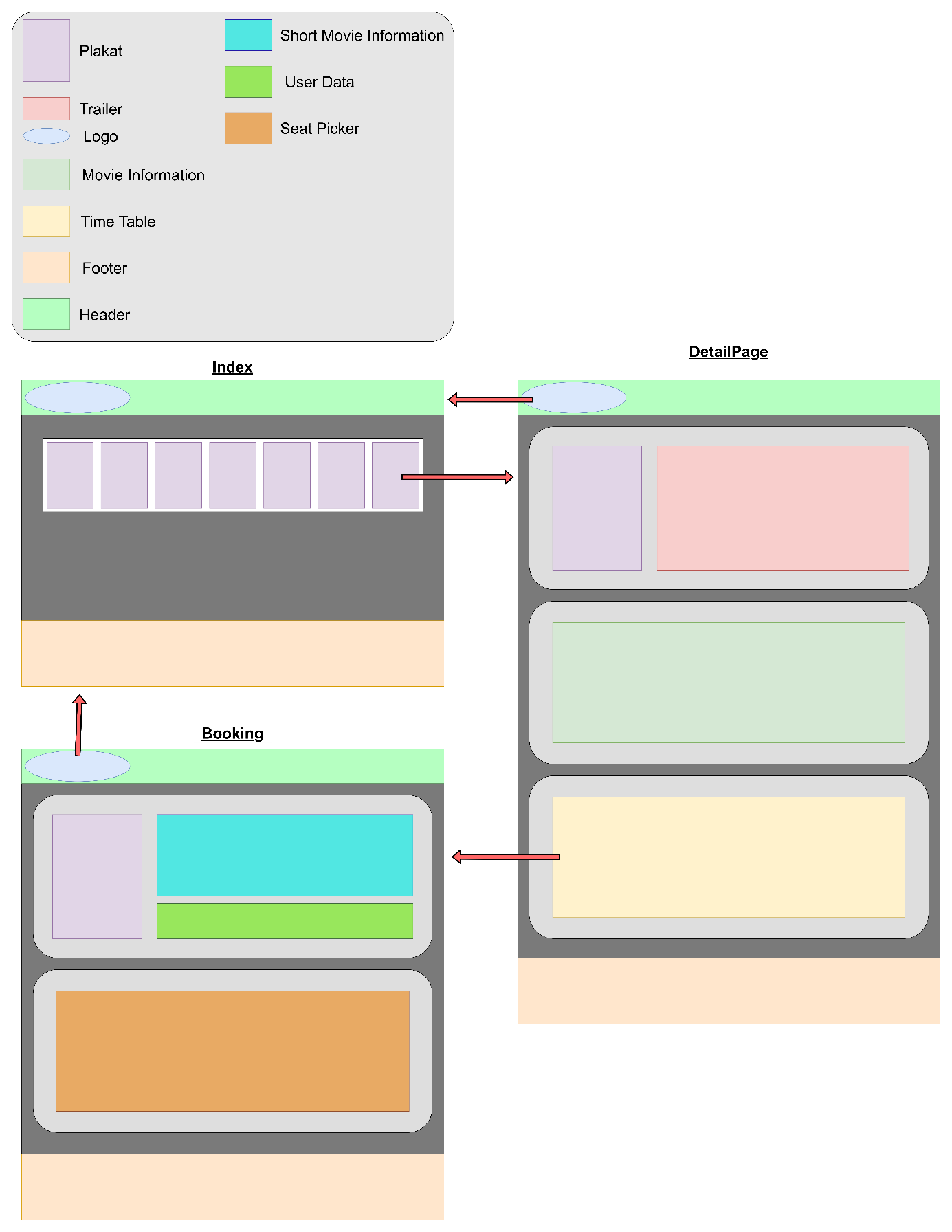
Für die Buchungsseite haben wir uns für ein Abbild des reellen Kinosaals durch Buttons in Sitz form entschieden. Dies gibt dem Kunden eine klare Vorstellung davon, wie der Saal aufgebaut ist und dadurch, wo er sitzen möchte.

Ein Bild, das Text, drinnen enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, verschieden, mehrere enthält.

Automatisch generierte Beschreibung

## Strukturlayout / Featureset



### Nutzen des Strukturplans

Durch das Erstellen eines Strukturplans konnten wir schon im Voraus erarbeiten und festlegen wie unsere Webseite am Ende aufgebaut sein soll. Dies ist wichtig da man ohne Planung nicht strukturiert und effektiv an einem Projekt arbeiten kann. Außerdem kann für ein Teamprojekt die Aufgabenverteilung nur stattfinden, wenn vorher Schritte und Aufgaben definiert wurden.

## Aufgabenverteilung

### Aufgabenblöcke

Als es an die Aufgaben Verteilung ging haben wir erst einmal entsprechend dem Strukturplan das Projekt in die Seiten Index, Detail, Booking und Contact unterteilt. Zusätzlich haben wir dann noch größere Komponenten wie die Platzauswahl als extra Aufgabe definiert. Des Weiteren galt es zwischen Design und Funktionalität zu unterscheiden. Hier galt für uns, dass Design die HTML und CSS und Funktion die PHP und Datenbank umfasst. Zu guter Letzt gab es noch die Dokumentation und Präsentation. Das Festhalten des eigenen Fortschrittes haben wir nicht als eigenen Aufgabenblock definiert, da sich hier jeder selbstständig kümmern muss.

### Verteilung

Zuerst haben wir das Erstellen der einzelnen Seiten verteilt. Dadurch hat Lukas sich um Index und Booking gekümmert, während Jakob die Seiten Detail und Contact übernommen hat. Außerdem haben wir die Arbeit etwas nach Interessen verteilt, weshalb sich Lukas mehr um die Funktionalität also um Datenbank und PHP und Jakob um Design und Inhalt gekümmert hat. Des Weiteren hat Lukas den Seitenübergreifenden Header und Footer angelegt während Jakob die Platzauswahl umgesetzt hat. Im weiteren Verlauf des Projekts hat Jakob das Design der einzelnen Bestandteile der Webseite überarbeitet und Dynamik sowie Responsiveness hinzugefügt. Im gleichen Zuge hat Lukas die Daten aus der Datenbank in die Webseite eingelesen und angezeigt. Hierfür waren an manchen Stellen Änderungen am Aufbau nötig damit die PHP-Anbindung besser funktioniert. Im Ausgleich für die Datenbank hat Jakob größere Teile der Dokumentation übernommen. Insgesamt ist es uns auf diese Weise gelungen die Aufgaben gleichmäßig zu verteilen und somit gemeinsam den Gesamtarbeitsaufwand zu handhaben.

# Durchführung

## Genutzte Technologien

### Repository

Wir haben uns dazu entschieden **GitHub** für unser Projekt zu verwenden. Somit konnten wir unsere Daten sicher speichern ohne die Gefahr, dass sie aufgrund von fehlerhaften Endgeräten, Problemen der Datenträger oder anderweitigem Versagen verloren gehen. Des Weiteren konnten wir so unser Projekt ohne Probleme auf unsere Entwicklungsgeräte übertragen und aktualisieren. Der größte Vorteil, war jedoch die Versionskontrolle, durch die wir Zwischenstände erstellen und gleichzeitig am Projekt arbeiten konnten.



### Ein Bild, das Text, Schild enthält. Automatisch generierte BeschreibungInhalte

Um die Struktur, Verteilung und den Aufbau der Inhalte unserer Webseite zu definieren haben wir die Hypertext Markup Language (kurz **HTML**) verwendet. Wie der Name schon erraten lässt, handelt es sich hier nicht um eine Programmier-, sondern um eine Auszeichnungssprache.

Der Code dieser Sprache ist aus einer Reihe von Tags in der Form <öffnend> </schließend> aufgebaut. Inhalte wie Text stehen dann zwischen dem öffnenden und dem schließenden Tag. Falls es keinen Inhalt gibt, kann der schließende Tag weggelassen werden und der öffnende am Ende um einen / ergänzt werden.

Des Weiteren können diesen Tags Parameter mitgegeben werden. Dies geschieht in der Form „parameterName = wert“ und wird in den öffnenden Tag hinter dessen Namen geschrieben.

#### Verwendete Tags/Elemente (Beispiele)

##### <div>

* + Dies ist ein Container mit fließendem Inhalt und ohne genauere Bedeutung. Er wird oft zur Strukturierung und in Kombination mit CSS-Styles verwendet.

##### <h1>

* + H1 beziehungsweise H1 bis H6 sind Überschriftelemente. Wie schon gesagt gibt es die Level 1 bis 6 wobei 1 die höchste Gewichtung trägt und somit größer und deutlicher hervorgehoben wird als die nachfolgenden. Dank der Level ist es außerdem möglich automatisch ein Inhaltsverzeichnis zu generieren.

##### <img>

* + Mit diesem Tag lassen sich Grafiken in ein HTML-Dokument einbetten. Ein praktischer Parameter hierfür ist „alt“, da sich hiermit ein Alternativtext definieren lässt. Dieser wird nur dann angezeigt, wenn das Bild nicht geladen werden konnte. Außerdem ist es dieser Text, welcher vorgelesen wird, sollte ein Benutzer gebrauch der Barrierefreiheit machen (wie zum Beispiel Blinde). Dies ist mit einer der Gründe, weshalb rein dekorative Grafiken mit einem leeren „alt“ versehen werden, da diese dann beim Vorlesen übersprungen werden.

##### <iframe>

* + Der Inlineframe wird dazu verwendet eine weitere HTML-Seite in die aktuelle einzubetten. Mit dem Parameter „src“ kann die URL dieser Webseite definiert werden. In unserem Projekt verwenden wir iframe dazu um ein YouTube Video als Trailer anzuzeigen. Dank der Einbettung kann dieses direkt auf unserer Webseite abgespielt werden und ist kein Link zu YouTube.

##### <i>

* + Der i Tag soll ein Wort wie zum Beispiel einen Fachbegriff vom restlichen Text abgrenzen. Dies geschieht dadurch, dass der vom i Tag umschlossene Inhalt Kursiv dargestellt wird.

##### <b>

* + Dieser Tag hebt einen Text hervor, indem dieser Fett geschrieben wird. Er hat ansonsten keine weiteren Informationen wie es bei <strong> oder <em> der Fall wäre.

##### <p>

* + Der Paragraf (kurz p) stellt einen Absatz und somit einen zusammenhängenden Abschnitt eines Textes dar. Da er Hauptsächlich eine gruppierende Funktion hat, kann er auch für die Gruppierung anderer Elemente als Text, wie zum Beispiel Bilder verwendet werden.

##### <table>

* + Mit dem Table Tag können meistens zwei dimensionale Tabellen erstellt werden. Innerhalb können mit <thead>, <tfoot>, und <tbody> der erste, letzte und alle anderen Abschnitte angelegt werden. Jeder dieser Abschnitte kann Zeilen enthalten welche mit <tr> für Table Row definiert werden. Danach kann pro Zeile noch eine oder mehrere Spalten mit <td> definiert werden.

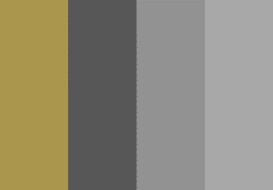
### Design



Das Cascading Style Sheet (kurz CSS) wird dafür verwendet, eine Webseite zu gestalten. Mit ihm können sämtliche Komponenten und Tags von HTML formatiert werden. Des weiteren können im HTML die Parameter is und class dazu verwendet werden zusätzliche Komponenten zu definieren. Somit kann beispielsweise eine einzelne Überschrift rot gefärbt werden indem nicht der <h1> sondern der „.überschrift“ Identifizierer verwendet wird.

#### Color Scheme

##### Base



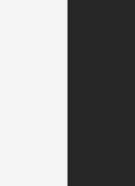
#AA964C Accent

#565756 Main Background

#929292 Element Hover

#A7A8A7 Element

##### Text



#F5F5F5 Main

#262626 Element

#### Flex box

as

### Datenbank Anbindung

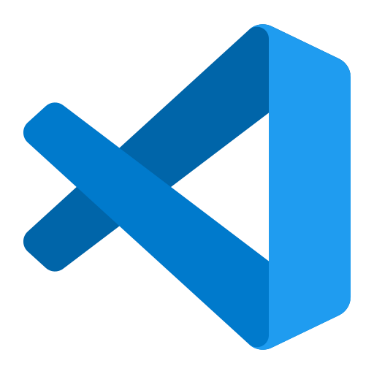
PHP

### Test-Server



XAMPP

### Entwicklungsumgebung



VisualStudio Code

## Codeausschnitte

A

## Aktivitäten

|  |  |  |
| --- | --- | --- |
| 26.10.21 | Beide | Uns wurde die Aufgabe gegeben eine Kino-Webseite aus Benutzer Sicht zu erstellen.  Wir haben das Git-Repository aufgesetzt und jeweils eine lokale Kopie unserer Branches erstellt. Des Weiteren haben wir uns Gedanken über Funktion und Aufbau unserer Webseite gemacht. Nach Betrachtung von Beispielen konnten wir unser Grund Layout und Feature Set definieren. |
| Lukas | Lukas hat die Index.php angelegt und mit Informationen gefüllt. Außerdem hat er das basisdesign umgesetzt für die Index Page umgesetzt. Des Weiteren hat er einen Header und Footer erstellt und in der Startseite eingebunden. Danach hat er zum Test, Daten in einem Array angelegt. |
| Jakob | Jakob hat die Detail Page angelegt und mit Beispiel Informationen gefüllt. Zu diesen gehören unter anderem Titel, Darsteller Altersfreigabe und Laufzeit. Des Weiteren hat er ein Plakat und Trailer eingebunden. Danach hat er das Basis Design erstellt und dafür das Color Scheme definiert. |
| 30.10.21 | Lukas | Lukas hat die Datenbank angelegt und per PHP an die Webseite angebunden. Dann hat er den Header und Footer noch mal überarbeitet. Außerdem hat er die Detail Page zu PHP umgewandelt damit die Daten der Datenbank eingebunden werden können. |
| Jakob | Jakob hat weitere Änderungen am Design vorgenommen. Danach hat er durch die Verwendung von Schatten den Elementen einen 3D Effekt verliehen. Zu guter Letzt hat er Style Variablen angelegt damit diese konsistent im restlichen Stylesheet angewandt werden können. |
| 16.11.21 | Lukas | Lukas hat dafür gesorgt, dass auf der Detail Seite Darsteller und weitere Informationen dynamisch aus der Datenbank geladen werden. Außerdem hat er die Buchungsseite erstellt und angefangen diese einzurichten. Des Weiteren hat er eine Admin Seite angelegt, welche zum Löschen von Filmen und Buchungen verwendet werden kann. Für die Buchungsseite hat er die Spielplan PHP-Parameter angelegt. |
| Jakob | Jakob hat die Styles der Detailpage so angepasst, dass die Elemente dynamisch skalieren und sich so an die aktuelle Größe des Browserfensters anpassen. Außerdem hat er dafür gesorgt, dass die Website responsive ist und dem Benutzer verdeutlicht, wenn er mit ihr interagiert. Danach hat er noch die Time Table mit den buchbaren Film Zeiten auf der Detailview eingefügt. |
| 20.11.21 | Jakob | Jakob hat die Sitzauswahl für die Buchungsseite angelegt |
| 30.11.21 | Lukas |  |
| Jakob | Jakob hat die Styles des Headers an das zuvor erstellte Designschema angepasst. außerdem hat er dafür gesorgt, dass sich die Sitzauswahl automatisch zentriert und in manchen Punkten dynamisch skaliert. |
| 01.12.21 | Lukas | Lukas hat die Sitzauswahl in die Buchungsseite eingebunden. |
|  | Jakob | Jakob hat eine Kontaktpage angelegt und diese mit Beispieldaten gefüllt. Danach hat er dafür gesorgt, dass sie dynamisch skaliert. zu guter Letzt hat er ein Kontaktformular erstellt. |
| 11.12.21 | Lukas | Lukas hat weitere Anpassungen an der Buchungsseite vorgenommen. |
| Jakob | Jakob hat sich um die Ausarbeitung der Dokumentation gekümmert und kleinere Korrekturen an der Webseite vorgenommen. |
| 14.12.21 | Beide | Wir haben unser Projekt fertiggestellt und zusammengeführt. Vor 18:00 Uhr haben wir unser Projekt selbst und dessen Dokumentation abgegeben. |
| 15.12.21 | Beide | Wir haben unsere Präsentation gehalten und damit diese Projektarbeit abgeschlossen. |

|  |  |  |
| --- | --- | --- |
| 26.10.21 | Lukas | * Git Repo Aufsetzen * Abstimmung Featureset * Index PHP * Header Footer * Testdaten als Array |
| Jakob | * Git Repo aufsetzen * Abstimmung Featureset * Detail Page * Color Scheme * Base Design |
| 30.10.21 | Lukas | * Datenbank Anbindung * Footer Header Refactoring * Detail Seite zu PHP |
| Jakob | * Design Anpassungen * Schatten * Style variablen anlegen |
| 16.11.21 | Lukas | * Detailseite, Darsteller dynamisch aus Datenbank * Buchungsseite * Admin Seite, löschen von Filmen und Buchungen * Spielplan PHP Parameter |
| Jakob | * Dynamische Skalierung der Detail Page * Responsive Design * Time Table in Detail |
| 30.11.21 | Lukas |  |
| Jakob | * Header Styling |
| 01.12.21 | Lukas | * Buchungsseite |
| Jakob | * Contact Page |
| 11.12.21 | Lukas | * Anpassungen der Buchungsseite |
| Jakob | * Ausarbeitung Dokumentation * Korrekturen auf der Webseite |
| 14.12.21 | Lukas | * Fertigstellung, Zusammenführung und Abgabe des Projekts und dessen Dokumentation |
| Jakob |
| 15.12.21 | Lukas | * Halten der Präsentation |
| Jakob |

# Anhänge

## Glossar

A

## Quellen

* <https://www.kino-bietigheim.de/programm>
* <https://www.cinemaxx.de/film/ammonite>
* <https://www.cineplex.de/neckarsulm/>
* <https://www.toptal.com/designers/colourcode/>
* <https://www.freeiconspng.com/downloadimg/15142>
* <https://github.com/logos>
* <https://freeiconshop.com/icon/html-icon-flat/#google_vignette>
* <https://freeiconshop.com/icon/css-icon-flat/>
* <https://developer.mozilla.org/de/docs/Learn/Getting_started_with_the_web/HTML_basics>
* <https://www.w3schools.com/html/>
* <https://www.w3schools.com/css/>
* <https://www.w3schools.com/php/>
* <https://developer.mozilla.org/de/docs/Web/CSS/box-shadow>
* <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
* <https://developer.mozilla.org/de/docs/Web/CSS/CSS_Flexible_Box_Layout/Basic_Concepts_of_Flexbox>
* <https://www.w3schools.com/css/css3_flexbox.asp>
* <https://icon-icons.com/de/download/94513/PNG/512/>
* <https://icon-icons.com/de/download/130084/PNG/512/>
* <https://icon-icons.com/de/download/130266/PNG/512/>