

Laborationsrapport

Laboration / Projekt

Kurskod: DT211G, Frontend-baserad-webbutveckling

Författare: Mohamed Abokashef, moab2302@student.miun.se

Termin, år: 01, 2024



Mittuniversitetet
MID SWEDEN UNIVERSITY

1 Sammanfattning

Jag planerar skapa en webbplats som fokuserar på att visa de senaste filmerna och TV-serierna. Där kommer användare att kunna söka och utforska olika titlar. En viktig funktion på sidan kommer vara möjligheten att översätta beskrivningarna av filmerna och tv-serierna till flera olika språk. Webbplatsen kommer att ha ett anpassat gränssnitt som är lättanvänt och visuellt tilltalande. Det kommer att inkludera alternativ för sortering, filtrering efter namn. För att garantera en positiv upplevelse för användarna på olika enheter, planeras webbplatsen att utvecklas med responsiv design. Det innebär att den kommer att anpassa sig smidigt för både stationära datorer och mobila enheter.

2 Innehållsförteckning

1	Sammanfattning.....
2	Innehållsförteckning.....
3	Introduktion
4	Teori.....
5	Metod
6	Konstruktion.....
7	Slutsatser.....
8	Källförteckning

Introduktion

Målet med denna projektet är att skapa en onlineplattform där användare kan upptäcka och kolla på de senaste filmerna och TV-serierna. Dessutom kommer plattformen att erbjuda översättningar av beskrivningar för att göra innehållet mer tillgängligt för en bredare publik. Webbplatsen kommer att använda information från externa källor som filmdatabaser och TV-serier, samt genom att använda Google Translate API för att erbjuda innehåll på olika språk.

Problemmotivering

1. Problem: Använd två olika API:er från olika webbtjänster och kombinera dem för att visa innehållet.:
 - För att skapa en lätt och bra upplevelse för användarna behöver jag integrera innehåll från två olika webbtjänster genom att använda deras API:er. Genom detta kan jag visa varierat innehåll på min webbapplikation och skapa en mer intressant användarupplevelse.
2. Visa senaste filmer och TV-serier:
 - Användare vill kunna smidigt upptäcka och få information om de senaste filmerna och TV-serierna.
 - Det krävs att detaljerad information som titel, beskrivning, betyg och bilder för varje titel hämtas och presenteras.
3. Sökmöjligheter:
 - en sökfunktion på sidan för att kunna hitta särskilda filmer eller TV-serier kan det göra webbplatsen mer användbar och det händer genom att använda titlar eller nyckelord för att söka en specifik film eller tv-serier. För att kunna visa passande resultat snabbt behöver webbsidan att koppla samman med en sökmotor för filtrering resultat och sortering data för att visa fler filmer och tv-serier på webbsidan.
4. Översättning av beskrivningar:
 - Användare ska kunna översätta film- och TV-seriebeskrivningar till olika språk.

- Det behövs en översättningstjänst API för att översätta innehållet utifrån användarens val av språk.

5. Användarupplevelse och gränssnittsdesign:

Användarna förväntar sig en vacker design som gör det lätt att navigera och utforska innehållet på webbplatsen. Det är viktigt att skapa en responsiv layout så att webbapplikationen fungerar smidigt på olika enheter och skärmar.

Teori

Parcel: Parcel är ett verktyg för att hantera paket och byggande av webbplatser. Det används för att automatiskt paketera och bygga webbapplikationer genom att behandla beroenden, kompilera filer och optimera resurser. [2]

Sass: För att förbättra hanteringen av CSS-filer och göra dem mer modulära, beslutade jag att använda Sass, en preprocessor för CSS. Jag använde Sass i mitt byggsystem med hjälp av Parcel, vDet gjorde det möjligt att effektivt strukturera och organisera stilmallar. Det gav mig även fördelen att använda variabler, nesting och återanvändbara mixins för att förbättra min CSS-kodbas. [6]

API: vilket är en förkortning för Application Programming Interface, utgör en samling av definitioner och protokoll som möjliggör kommunikation mellan olika programvaror. Inom detta projekt används API:er för att hämta information om filmer och TV-serier från externa webbtjänster.

async och await: är termer som används i JavaScript för att effektivt hantera asynkrona operationer på ett sätt som är enklare att förstå och hantera. [3]

clip-path: clip-path i CSS används för att skapa ett område där endast vissa delar av ett element visas, medan resten av elementet är dolt. Detta gör det möjligt att visa en specifik region och dölja allt utanför den [1].

Transform: Det ändrar dimensionerna, utseendet och placeringen av en låda och dess innehåll genom att rotera, skeva, skala. [4]

Transition: är ett stilattribut som används för att skapa mjuka övergångar mellan två olika tillstånd av ett element. Det kan användas för att animera förändringar i egenskaper som färg, storlek, position och mycket annat. [5]

- **egenskaper som kombinerar:**

- transition-property
- transition-duration
- transition-timing-function
- transition-delay

Metod

1. Jag valde att använda Parcel som verktyg för att skapa min webbsida. Parcel är en aktuell och effektiv lösning för byggverktyg, som automatiserar många av de uppgifter som krävs för att förbereda en webbplats inför publicering. Jag kunde enkelt att hantera olika typer av filer såsom HTML, CSS och JavaScript med Parcel, samtidigt som jag utnyttjade dess inbyggda funktioner som live reload och automatisk optimering av filstorlekar. Genom att använda Parcel kunde jag förbättra min process för utveckling och fokusera mer på att skapa innehåll och design för min webbplats utan att behöva oroa mig för tekniska detaljer kring byggprocessen.

Jag använder "Fetch" API för att göra HTTP-förfrågningar till "TMDb API för att hämta data om filmer och TV-serier, som jag sedan använder på min webbplats.

2. Async/await används för att göra kodblocket synkront och hantera de asynkrona HTTP-anropen på ett smidigt och lättläst sätt. Detta bidrar till att undvika komplexa callback-problem och möjliggör väntan på HTTP-svaret innan datan bearbetas vidare.
3. Google Cloud Translation API: Det här verktyget används för att konvertera text från ett språk till ett annat. Det är till för att översätta de originalbeskrivningarna av filmer till det önskade språket.

4. `window.onscroll`-event: Denna kodrad ställer in en händelsehanterare för när fönstret bläddras, vilket innebär att funktionen `scrollFunction()` kommer att aktiveras varje gång användaren skrollar på sidan.[figure 2]
5. CSS-transitioner: Används för att underlätta smidiga övergångar mellan olika tillstånd hos DOM-elementet. Detta uppnås genom att ändra värdena för CSS-egenskaperna och ange övergångseffekter (`transform = "scale(1)";`) för att förstora nav när användaren scrolla ner. [figure 2]

Konstruktion

1. För att lösa detta problem måste vi utföra följande steg:

- Identifiera lämpliga API:er: Först måste jag hitta två olika online-tjänster som erbjuder API:er för att hämta önskat material för detta har jag valt (TMDB) webbsida för filmer och TV-serier och (google translate API) för att översätta beskrivning av filmer och TV-serier.
- Användning av API:er: för att skicka förfrågningar och få information från olika webbtjänster. Det är viktigt att följa dokumentation för varje API för att kunna skicka korrekta förfrågningar och hantera svar från servern på rätt sätt.
- Kombinera och formatera data: Efter att jag har samlat in information från båda API:erna behöver vi sammanfoga dem på ett meningsfullt sätt och anpassa dem efter våra specifika krav. Det kan inkludera att kombinera information från olika källor, filtrera eller bearbeta den för att göra den lämplig för vår webbapplikation.
- Presentera innehållet på användargränssnittet: till sist behöver vi visa det samlade materialet på användargränssnittet på ett sätt som är lätt att använda.

2. För att lösa detta problem:

- från föregående steget och efter att jag har Hämtat information från en databas för filmer och TV-program genom att använda dess API.

- Funktionen `fetch()` används för att göra en GET-förfrågan till TMDb API med en specifik URL som innehåller API-nyckeln (`apiKey`). Jag använde `await` för att vänta på att HTTP-svaret ska returneras från API:et innan koden fortsätter att exekveras. Detta gör koden asynkron, vilket innebär att den inte hindrar hela programmet från att köra medan den inväntar svar från API. När svaret har returnerats från API, använde `await` på nytt för att vänta på att JSON-data ska hämtas från svaret med hjälp av `json()`-funktionen. Detta resulterar i ett JavaScript-objekt (`movieData`) som innehåller detaljer om den film eller tv-serie som efterfrågades.
- Sedan kompletterade jag js koden för att visa informationen på ett sätt som är lättförståeligt för användare genom att visa titel, beskrivning, betyg och bilder för varje titel.

3. För att lösa detta problem:

- Jag lade till en sökfunktion på webbplatsen så att besökare kan söka efter filmer eller TV-serier genom titlar eller nyckelord. Använd en sökmotor för att sortera och visa relevanta resultat direkt när användaren skriver eller klickar på val som jag erbjuder det och växla mellan TV-serier eller filmer.

4. För att lösa detta problem:

- jag använde google översättningstjänst API för att tillåta att beskrivningar översätts till olika språk.
- samtidigt Låter jag användare välja det språk de föredrar och översätt beskrivningarna automatiskt baserat på deras val när de klickar på knappen översätta brevskrivning.

5. För att lösa detta problem:

- Utvecklade en användarvänlig och snygg design för att förhöja användarupplevelsen och det händer genom att skapa en animering som visar beskrivning när användaren hover på film eller tv-serier bild, för att kunna göra det behövde jag använda komponering av `clip-path` med CSS transitions [1]. samtidigt använde jag js för att ändra nav storlek när användaren scrolla ner genom att använda `transform` med `scale`. Jag använde `transition: all .5s ease-in-out;` detta resulterar i en övergångseffekt för alla attribut på elementet. Övergången tar 0,5 sekunder och använder en accelererande och

decelerande effekt (ease-in-out) för att skapa en smidig övergång när något attribut ändras. "transform: scale(1.6) rotate(20deg);" det är Ett sätt att ändra elementet är att förstora det med en faktor på 1,6 (skala(1,6)) och sedan rotera det 20 grader(rotera(20 grader)). Som vi ser i figure 1



Figur 1

- jag Skapade en responsiv layout som anpassar sig efter olika enheter och skärmar för att säkerställa en smidig användarupplevelse oavsett enhet.

```
"use strict";

window.onscroll = function () { scrollFunction() };

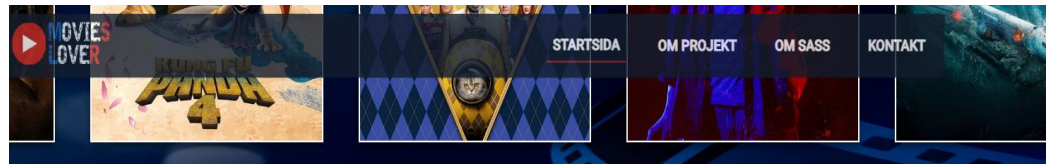
function scrollFunction() {
  if (document.body.scrollTop > 0 || document.documentElement.scrollTop > 0) {
    document.getElementById("scroll_function").style.backgroundColor = "darken(#395886, 25%)";
    document.getElementById("scroll_function").style.transition = "all .5s !important";
    document.getElementById("scroll_function").style.transitionDuration = "0.6s";
    document.getElementById("scroll_function").style.opacity = ".8";
    document.getElementById("scroll_function").style.borderRadius = "10px";
    document.getElementById("scroll_function").style.transform = "scale(.7)";
  } else {
    document.getElementById("scroll_function").style.opacity = "1";
    document.getElementById("scroll_function").style.transform = "scale(1)";
    document.getElementById("scroll_function").style.transitionDuration = "0.4s";
  }
}
```

Figur 2

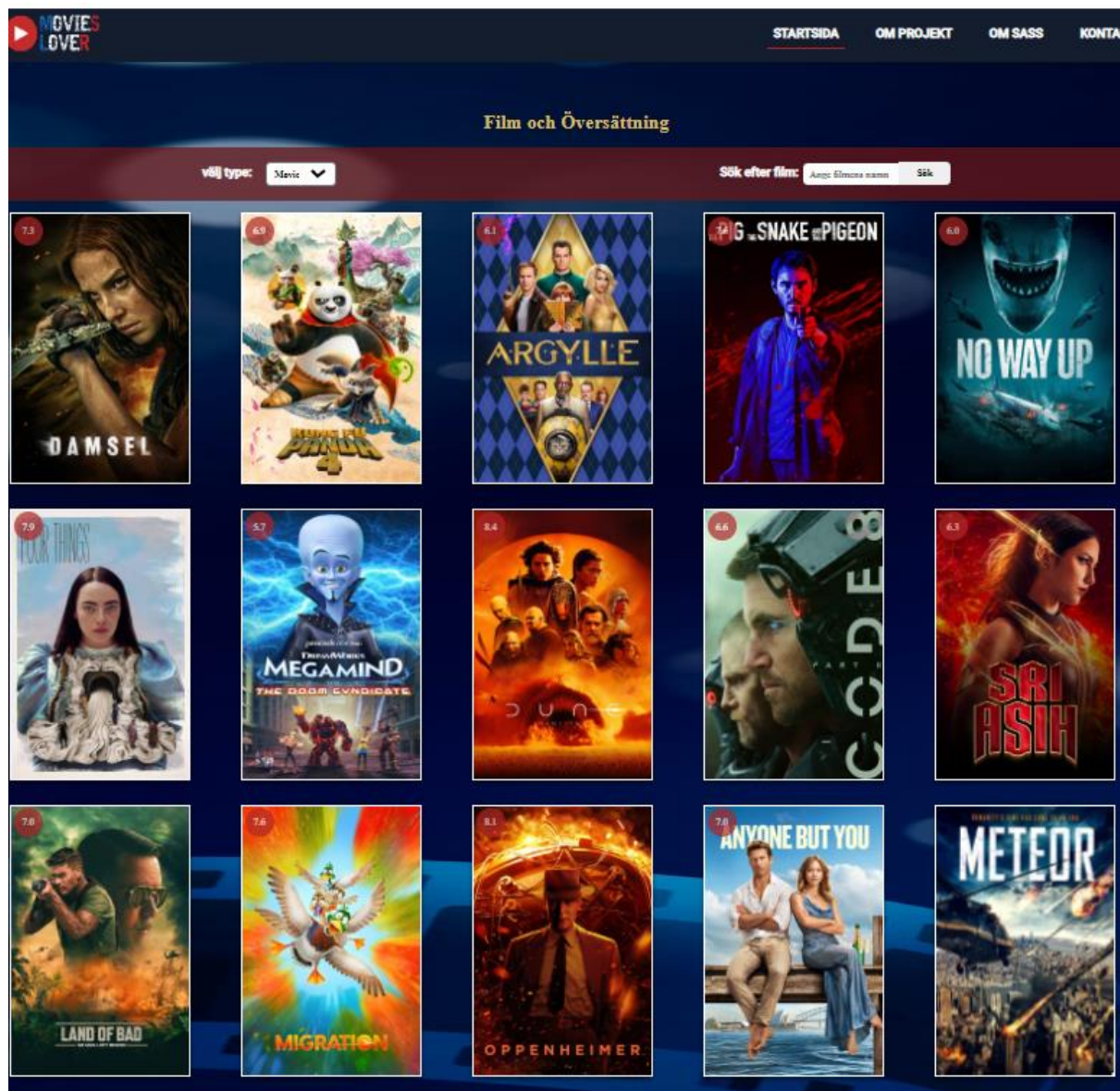
- Som vi ser i figure 2 använde jag JS koden för att skapa en effekt som ändrar utseendet på ett navelement när besökaren scrollar på sidan. En if-sats används för att kolla om användaren har rullat ner på sidan. Om detta är fallet ändras olika CSS-egenskaper för det specifika elementet för att ge en visuell effekt när användaren rör sig ner på sidan. När en användare rullar uppåt återställer elementet till sitt ursprungliga utseende som vi ser i figure 3 och 4.



Figur 3



Figur 4



Figur 5

Resultat

Projektet ledde till att skapa en responsiv och lättanvänd webbplats för filmer och TV-serier. Genom att ta till vara på moderna metoder och verktyg, såsom Sass för CSS-styling och Parcel för byggprocessen, jag kunde förbättra processen för att utveckla webbplatsen och göra den mer anpassningsbar och lätt att underhålla på lång sikt. Dessutom, genom att integrera Google Translate API, kunde jag erbjuda flerspråkigt stöd vilket förbättrade tillgängligheten och användarupplevelsen för besökarna på webbplatsen. Se figure 5

Källförteckning

1. Almand, T. (2019, July 9). Animating with clip-path. *CSS-Tricks*. <https://css-tricks.com/animating-with-clip-path/>
2. Parcel documentation. (n.d.). Parceljs.org. Retrieved March 16, 2024, from <https://parceljs.org/docs/>
3. Async/await. (n.d.). Javascript.Info. Retrieved March 16, 2024, from <https://javascript.info/async-await>
4. CSS property: Transform. (n.d.). Htmldog.com. Retrieved March 16, 2024, from <https://www.htmldog.com/references/css/properties/transform/>
5. CSS property: Transition. (n.d.). Htmldog.com. Retrieved March 16, 2024, from <https://www.htmldog.com/references/css/properties/transition/>
6. @use. (n.d.). Sass-lang.com. Retrieved March 17, 2024, from <https://sass-lang.com/documentation/at-rules/use/>