

Факултет за информатички науки и компјутерско
инженерство (ФИНКИ)



Проект по предметот - Напреден Веб Дизајн

Тема: Movie App Application

Изработиле:

Бојана Здравковска 203022
Надица Димитровска 203013

Датум 14.09.2023

➤ Вовед во проектот

Во последните неколку години бевме сведоци на тоа колкаво влијае може да има филмската индустрија врз нас, како корисници на интернет. А за тоа, доказ ни е и самата пандемија, во која бевме изолирани во нашите домови и отуѓени од луѓето од нашиот мал свет.

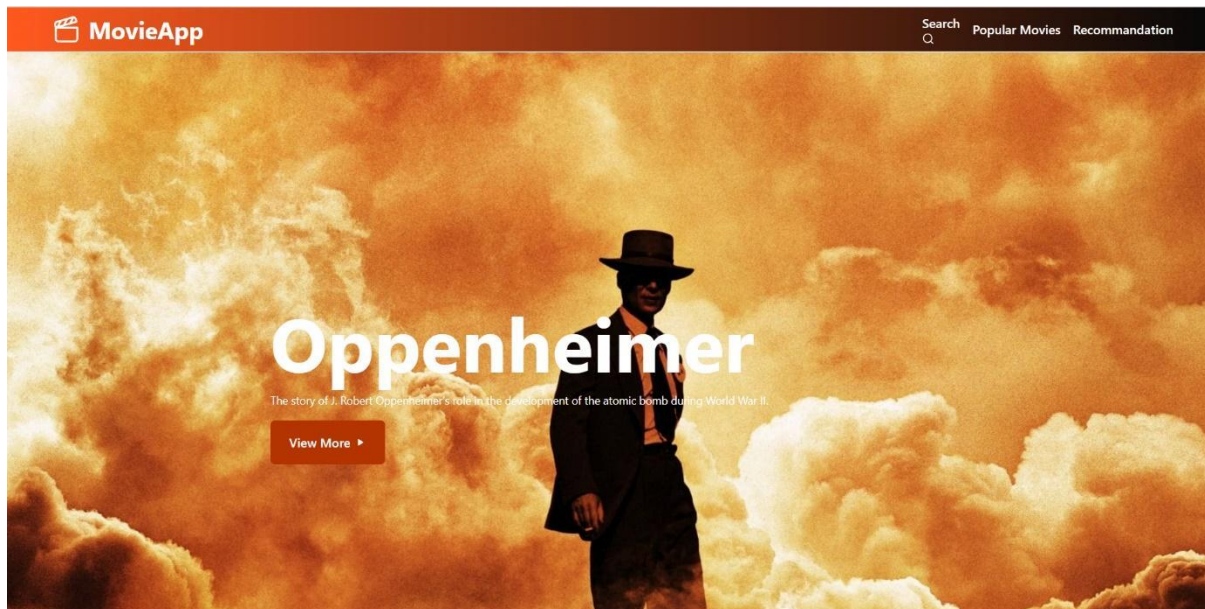
Преку таргетираните реклами, високобуџетен маркетинг, разновиден merchandise и огромно социјално влијание филмската индустрија ја покажа својата моќ. Инспирирани од тоа, и со самиот напредок на развојот на веб апликации, креирањето на апликации наменети за филм фанатаците станува се полесно и подостапно за девелопери да имаат простор да креираат едукативно и интересно искуство за корисниците. За целите на овој предмет, изработка на една таква апликација беше овозможена со помош на Vue.js

Vue.js е прогресивна JavaScript рамка која овозможува развој на апликации на една страница со архитектура базирана на компоненти. Неговата едноставност и флексибилност го направи идеален избор за изградба на тривијална апликација која нуди беспрекорно корисничко искуство.

➤ Процесот на креација на веб апликацијата

Првото прашање при креирање на апликацијата кое си го поставивме е како да го привлечеме вниманието на корисникот. За таа цел ни требаше впечатлива почетна страна односно фокусна точка. Фокусната точка е привлечниот центар на страницата; се издвојува и се разликува од другите компоненти. Кога корисникот ќе пристапи кон апликацијата тој најпрво е пречекан од екран каде се наоѓа постер од моментално најактуелниот филм на репертоарот и е искористена комбинација на бои и фонтови кои го зграпчуваат вниманието. Дополнително, следејќи го правилото на конзистентност и препознатливост, во горниот лев агол е испишано името на апликацијата која му дава до знаење дека е на точното место. Логото во комбинацијата со името се користи и како escape button за корисникот да има можност лесно да се врати на почетниот екран.

Бидејќи проектот е креиран со Vue.js предноста е тоа што во овој проект е овозможено рутирање така што при клик на одреден линк со опцијата Router Link и доделување на соодветно мапирање корисникот е пренасочен на нова адреса. За да се овозможи соодветно рутирање низ целата апликација, креиравме соодветни компоненти како и соодветни views каде се вметнати тие компоненти. Во index.js фајлот се наведени сите патеки кои се достапни во овој проект, патека кон домашен екран, патека кон екран за пребарување на филмови, патеки кон најпопуларните филмови, патека кон предлог филмови.



Друг важен дел со кој се сретнува корисникот на почетниот екран е навигациското мени. Навигациското мени служи за корисникот лесно да може да се пренесочува кон различни делови од апликацијата. На почетокот од навигациското мени се наоѓа самото име на апликацијата, а останите делови го носат кон различни страни. За се оствари овој лесен дизајн на навигациско мени е искористен Tailwindcss.



Навигациското мени се состои од три линкови кои претставуваат главни компоненти во веб апликацијата: Popular Movies, Search и Recommendation.

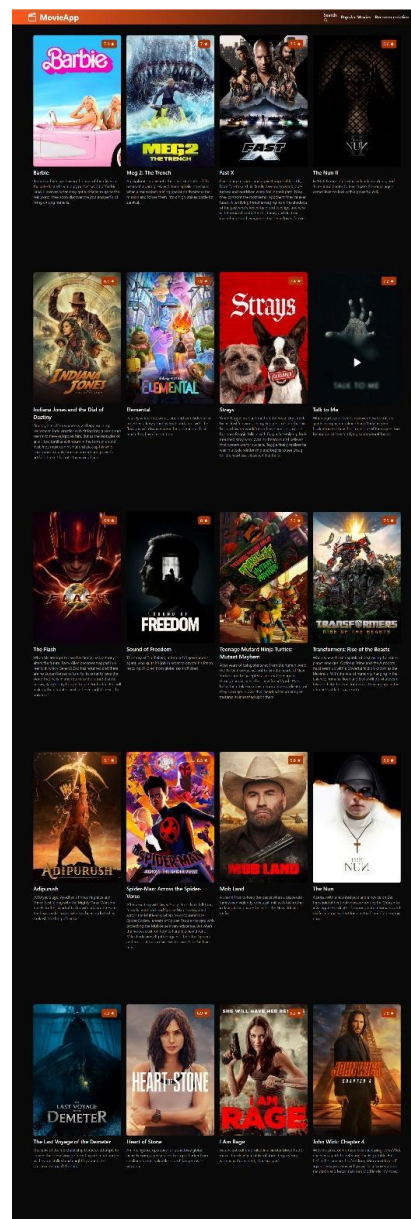
За самата апликација да се креира ги користевме следниве команди:

- ***npm create vite@latest*** . - создавање на проект со vite
- ***npm install*** - инсталација на сите зависимости во проектот дефинирани во package.json
- ***npm install -D tailwindcss postcss autoprefixer*** - инсталација на библиотеки и модули потребни за tailwindcss
- ***npx tailwindcss init -p*** - инсталација на конфигурациски фајл за tailwindcss
- ***npm install flowbite*** - инсталација на оваа библиотека за дизајн на компоненти
- ***npm install axios*** - за превземање на апи
- ***npm install --save-dev @iconify/vue*** - библиотека за превземање на икони
- ***npm install vue-router@4*** - за управување со рути

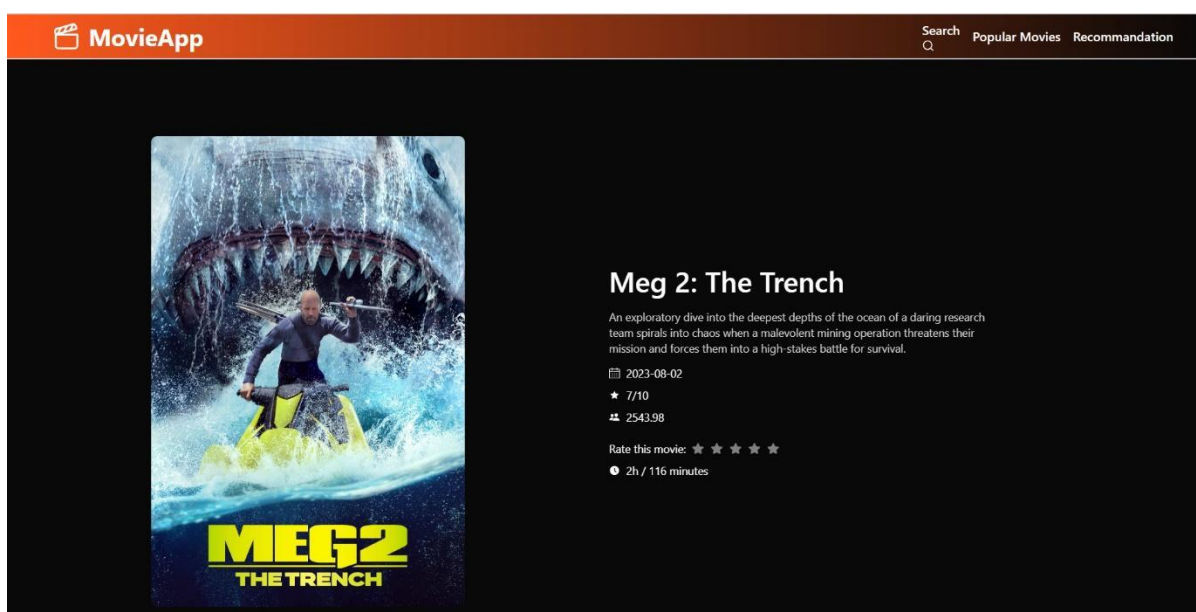
➤ Popular movies

Низ текот на целиот проект податоците се преземени преку испраќање повици кон отворено API на The Movie Database (TMDb) достапно на линкот

<https://developer.themoviedb.org/reference/intro/getting-started> . Ова апи нуди огромен број на можности и подесување на параметрите за да се кастмзира веб апликацијата според различни вкусови. Интеграцијата вклучува HTTP барања до крајните точки на API користејќи ја вградената библиотека Axios на Vue.js. Конкретно, за компонентата Popular movies, во кодот е направен повик кон https://api.themoviedb.org/3/movie/popular?&api_key=c55957c0a5d4a32160fa6b032a436368&language=en-US&page=1 кој враќа одговор низа од најпопуларните филмови во json формат. Потоа тие филмови се прикажени како картички поделени во grid систем каде имаат конзистентност 3 колони.



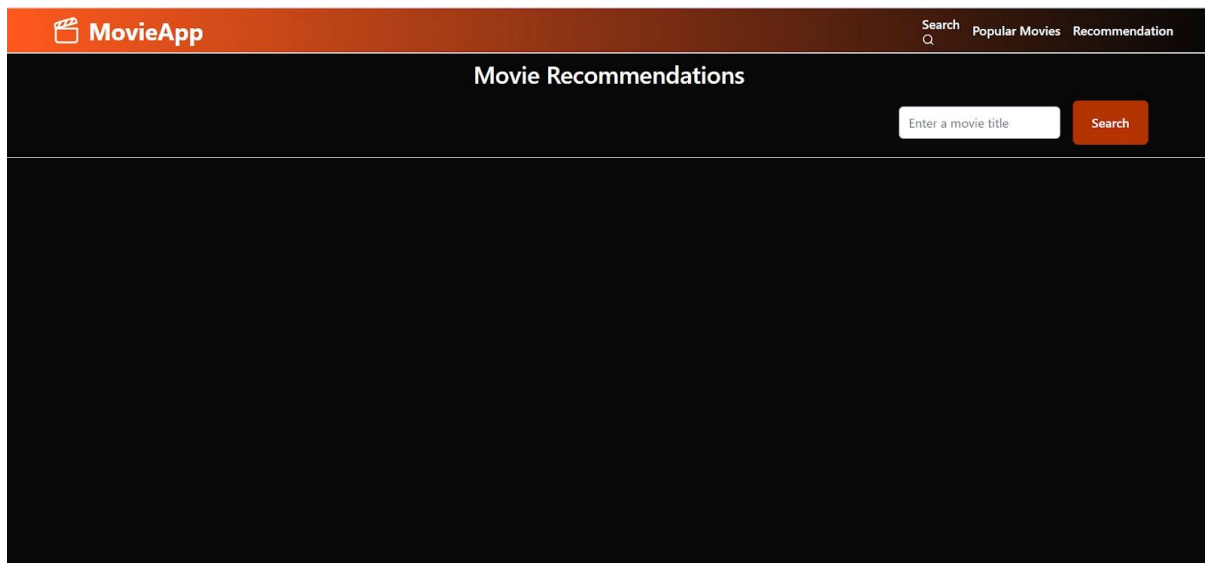
Секоја картичка е индивидуална компонентна која претставува различен филм. Доколку корисникот застане со глумчето на некоја од картичките (hover движење на кое е додадено да се заматува сликата) е овозможен клик кој води кон детален опис на филмот. Податоците за филмот се преземени така што одговорот кој ни го враќа апи повикот односно листата од филмови во себе содржи уникатни key:value вредности и од нив се естрахирани оние кои се од најголем интерес на корисниците како rating, година на издавање, режисер... Дополнително е овозможена интеракција на корисникот со системот каде може да остави свој rating доколку го има изгледано филмот и да го зачува истиот за понатамошно користење на апликацијата.



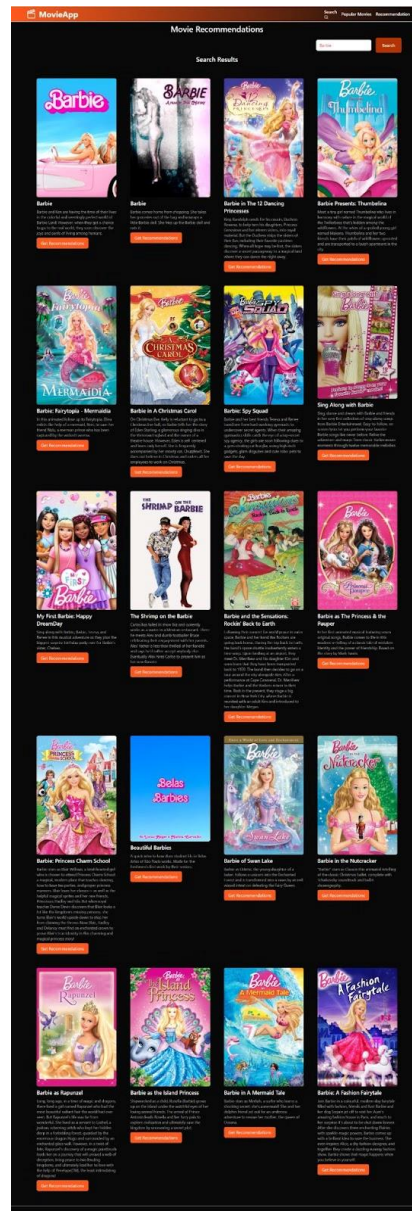
Функционалноста за rating е во форма на ѕвезди, кои доколку не е оценет филмот ѕвездите се во сива боја а при клик на нив, тие се обојуваат во жолта боја. Со самото кликање на ѕвездите се променува и бројката на rating и истата се променува кај Popular во десниот агол при приказот на филмовите.

➤ Recommendation

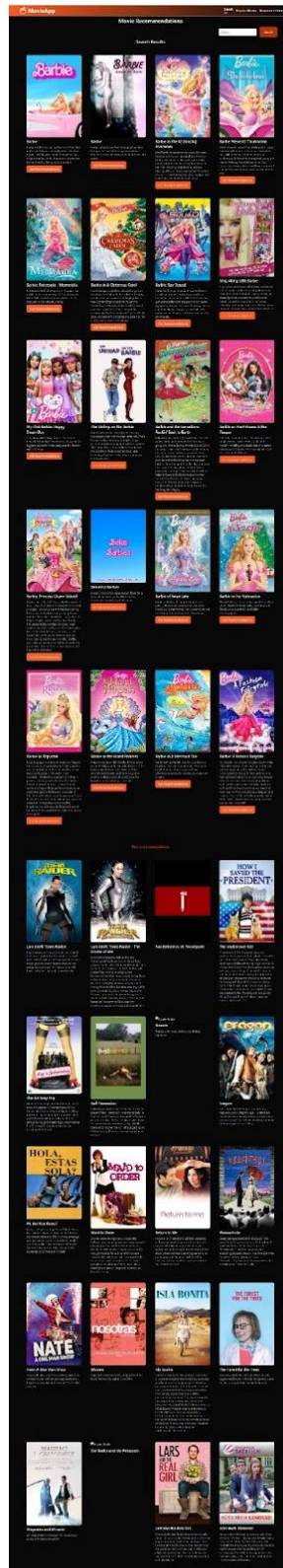
Со клик на компонентата за Recommendation, ви се прикажува следниот интерфејс, со цел да овозможи на корисникот да го најде препорачаниот филм сличен на тој што го пребарува и да добие детални информации за него. Изгледот на интерфејсот изгледа како на приложената слика.



На самиот интерфејс е прикажано наслов на компонентата и текст инпут поле со копче за пребарување, корисникот во инпут полето го внесува филмот кој што сака да го најде за да добие предлог како филмот што го гледал, па за таа цел најпрво се листаат филмовите од пребарувањето.



Резултатот од пребарувањето е прикажан во вид на сет картички, каде секоја картичка претставува различен филм, каде што картичките се поделени во grid систем каде имаат конзистентност 3 колони. За секој филм на картичката, корисникот ќе ги види следниве информации: слика од филмот, кој може да го види во горниот дел на картичката, наслов на филмот, кој се наоѓа под сликата, како и краток опис на филмот, и под секој опис има копче Get Recommendation, со самото притискање на копчето, го зима ид-то на тој филм и ги прикажува прелдлог филмовите како на тој филм на кој што е притиснато.



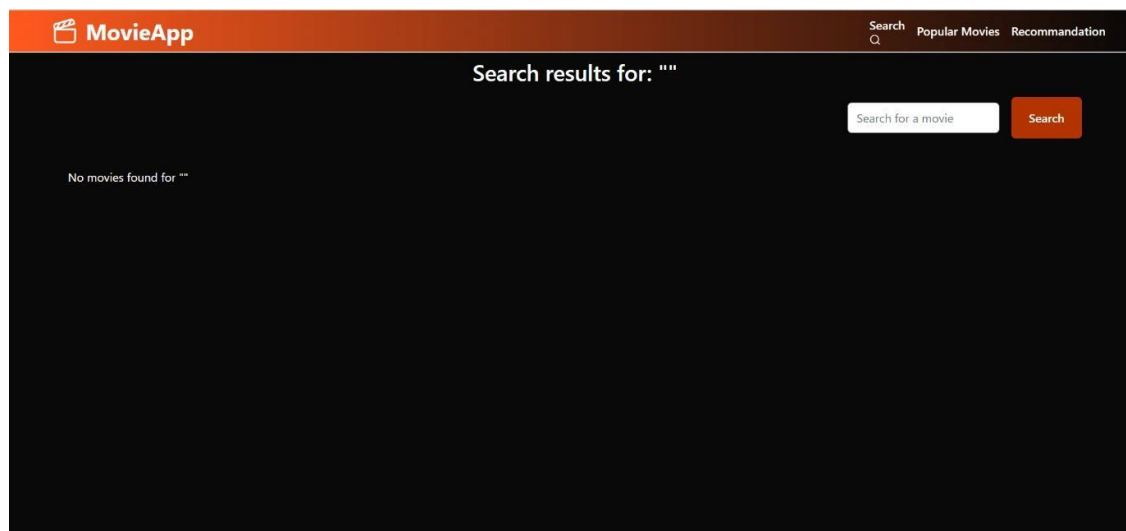
Доколку корисникот застане со глумчето на некоја од препорачаните картичките (hover движење на кое е додадено да се заматува сликата) е овозможен клик кој води кон детален опис на филмот. Податоците за филмот се преземени така што одговорот кој ни

го враќа апи повикот односно листата од филмови во себе содржи уникатни key:value вредности и од нив се естрахирани оние кои се од најголем интерес на корисниците како rating, година на издавање, режисер...

Функционалноста за оваа компонента е направена со најпрво во инпут полето да се внесе посакуваниот филм и да се притисне на копчето за пребарување, кое од апи-то <https://api.themoviedb.org/3/search/movie> ќе ни ги даде сите филмови со пребараниот збор. Под секој филм има копче Get Recommendation, со самото притискање на копчето, го зима ид-то на тој филм и ги прикажува преледлог филмовите како на тој филм на кој што е притиснато. Ид-то го пренесува во самото апи <https://api.themoviedb.org/3/movie/{movieId}/similar> на кои потоа се додаваат параметрите од клучот, страната и јазикот.

➤ Search

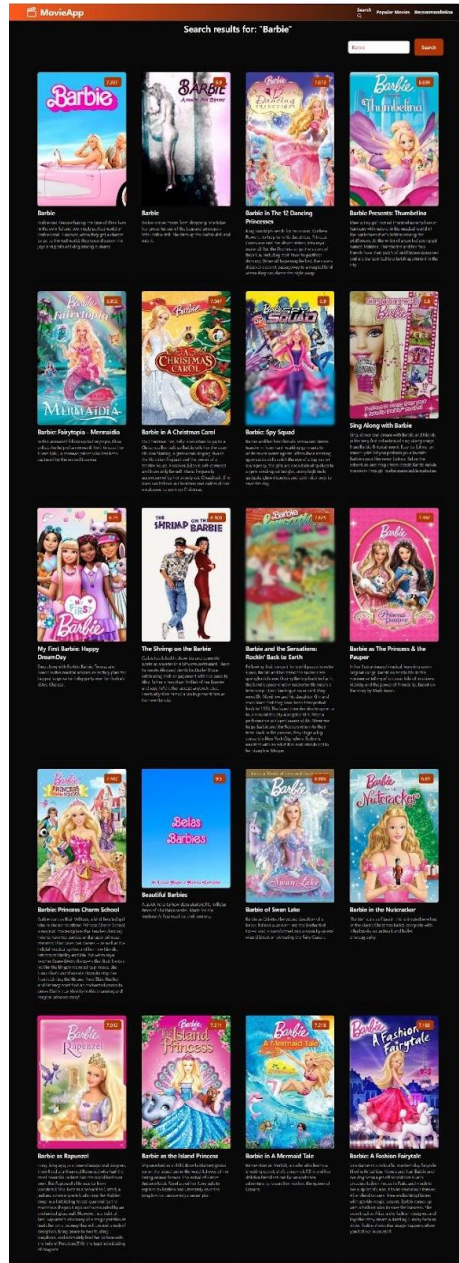
Со клик на компонентата за Search, се прикажува следниот интерфејс, со цел да овозможи на корисникот да го најде истражуваниот филм и да добие детални информации за него. Изгледот на интерфејсот изгледа како на приложената слика,



се со цел корисникот да го побара посакуваниот филм, за кој што сака да добие детални информации. На прво отварање, се гледа дека имаме наслов за пребараниот филм (во моментот при клик на самата компонента нема), порака доколку таков филм со тоа име не е пронајден и инпут поле со копче за пребарување. Со самото внесување на инпут полето се листаат повеќе филмови со истото име (keywords на насловот) кое го содржат и во насловот Search results for: се зима зборот од инпут полето и се прикажува, додека наместо порака дека не е најден филмот кој го пребарува корисникот, добиваме листа од филмови. Корисникот може да го менува внесениот пребарувачки збор во текст инпутот и да притисне "Search" копчето повторно за да направи ново пребарување. За оваа функционалност е користен data објект, кој ги чува состојбите на самата компонента, во овој случај зборот кој се пребарува и резултати од пребарувањето, а исто така има и

метод, кој е асинхрон за пребарувањето на филмови од апито <https://developer.themoviedb.org/reference/search-movie>.

Листањето на филмови е земено од истото апи што се користи и во претходните компоненти, но со само мапирање на /search (https://api.themoviedb.org/3/search/movie?include_adult=false&language=en-US&page=1&api_key=c55957c0a5d4a32160fa6b032a436368).



Исто така приказот на резултатот од пребарувањето е прикажан во вид на сет картички, каде секоја картичка претставува различен филм, каде што картичките се поделени во grid систем каде имаат конзистентност 3 колони. За секој филм на картичката, корисникот ќе ги види следниве информации: слика од филмот, кој може да го види во горниот дел на

картичката, наслов на филмот, кој се наоѓа под сликата, како и краток опис на филмот, оценка на филмот во форма на бројка со звезда во горниот десен агол со црвена позадинска боја. Секоја слика на филмот е кликабилна и ако корисникот кликне на неа, ќе биде пренасочен на детална страница за селектираниот филм.

➤ **Заклучок**

Vue.js одигра клучна улога во правењето на Movie App апликацијата да биде интерактивна и динамична. Неговиот систем овозможува да се ажурира корисничкиот интерфејс во реално време врз основа на корисничките дејства, создавајќи течно и привлечно искуство. Апликацијата е поделена на 4 компоненти користејќи Vue.js. Секоја компонента има своја функционалност и одговорности, како што е самиот почетен екран односно влезна точка, излитување на листата од филмови и детален опис за секој филм. Оваа архитектура базирана на компоненти го направи кодот помодуларен и полесен за одржување. Вградени се привлечни графики, бои и фонтови за да се подобри ангажманот и да биде апликацијата попријатна за користење. Дизајнот има цел да биде впечатлив, но во исто време и едноставен за да не направи на корисниците да се чувствуваат пренатрупано со информации. Создавањето на оваа апликација со користење на Vue.js ни беше возбудлив потфат што ни овозможи да го истражime потенцијалот на веб-технологиите во развојот на интерактивни апликации. Комбинацијата на Vue.js, интеграција на API, внимателен UI/UX дизајн и дополнителни функции и преку самиот процес на дизајнирање и имплементација, добивме вредни сознанија за моќта и флексибилноста на Vue.js како JavaScript рамка за градење интерактивни веб- апликации. Свкупно, проектот ни обезбеди практично искуство во веб-дизајн, развој на предниот дел, интеграција на API и важноста на корисничкото искуство. Тоа ни овозможи да ги изостриме нашите вештини и да го продлабочиме нашето разбирање за современите веб технологии, на крајот подготвувајќи не за идните напори во областа на веб-развојот.

Линк до проектот: <https://github.com/bojanazdravkovska/nvd-proekt>