به نام خدا



دانشگاه فنی و حرفه ای

دانشکده فنی و حرفه ای شهید منتظری

پایان‌نامه کاردانی

رشته کامپیوتر گرایش نرم افزار

استاد راهنما:  
مهندس ملک فر

نگارش:

مهدیار اعتماد  
  
دی ماه سال ۱۴۰۳

فصل ۱:

مقدمه

انگیزه:

در دنیای امروز، صنعت فیلم و سریال به یکی از محبوب‌ترین و پرطرفدارترین بخش‌های سرگرمی در سراسر جهان تبدیل شده است. با افزایش روزافزون تولید محتواهای تصویری و علاقه‌ی کاربران به تجربه‌های متنوع، نیاز به ابزارهایی که اطلاعات دقیق، ساختارمند و جذاب را درباره فیلم‌ها و سریال‌ها ارائه کنند، بیش از پیش احساس می‌شود.

ایده اصلی پروژه ReelRave از این نیاز نشأت گرفته است؛ ایجاد بستری که کاربران بتوانند به سادگی اطلاعات مربوط به فیلم‌ها و سریال‌ها، از جمله خلاصه داستان، بازیگران، ژانرها، امتیازها و نقدها را دریافت کنند. این پروژه با هدف ساده‌سازی فرایند جستجو و ارائه اطلاعات شفاف و کاربردی طراحی شده است.

یکی از دلایل اصلی انتخاب این پروژه، علاقه شخصی به دنیای فیلم و سریال و مشاهده چالش‌هایی بود که افراد هنگام یافتن اطلاعات جامع و دقیق با آن روبه‌رو هستند. بسیاری از وب‌سایت‌ها یا فاقد رابط کاربری مناسب هستند یا اطلاعاتشان ناقص است. ReelRave تلاش می‌کند با بهره‌گیری از تکنولوژی‌های پیشرفته‌ای مانند **Django REST Framework** در بخش بک‌اند و **Vue.js**  به همراه **Vuetify**  در بخش فرانت‌اند، تجربه کاربری متمایزی ارائه دهد.

این پروژه نه تنها به ارتقای دانش فنی در زمینه برنامه‌نویسی و توسعه وب کمک می‌کند، بلکه نشان‌دهنده تعهد به حل مشکلات واقعی کاربران است. ReelRave می‌تواند به عنوان ابزاری مفید برای دوست‌داران فیلم و سریال و حتی منتقدان و تحلیل‌گران حوزه سینما عمل کند.

در نهایت، این پروژه فرصتی برای توسعه مهارت‌های شخصی، خلاقیت و تحقق علاقه‌ها در مسیری علمی و کاربردی است.

مروری بر پیشینه و کار های مشابه:

با گسترش فناوری و افزایش علاقه کاربران به صنعت فیلم و سریال، پلتفرم‌های متعددی برای ارائه اطلاعات در این حوزه توسعه یافته‌اند. وب‌سایت‌هایی مانند **IMDb**  و **Rotten Tomatoes**  از پیشگامان این زمینه هستند که اطلاعاتی جامع در مورد فیلم‌ها، سریال‌ها، امتیازات، نقدها و بازیگران ارائه می‌دهند.

با این حال، این پلتفرم‌ها با وجود کارایی بالا، اغلب رابط کاربری پیچیده‌ای دارند یا نیازمند اشتراک ویژه برای دسترسی به برخی اطلاعات هستند. علاوه بر این، برخی از اپلیکیشن‌ها، مانند **Letterboxd**، بیشتر بر جنبه‌های اجتماعی مانند ثبت نظر کاربران متمرکز هستند و برای کاربران عادی که فقط به دنبال اطلاعات سریع و دقیق هستند، ممکن است کارآمد نباشند.

پروژه **ReelRave**  تلاش می‌کند تا از نقاط قوت این پلتفرم‌ها الهام بگیرد و با ترکیب سادگی در طراحی، دسترسی آزاد به اطلاعات و ارائه جزئیات کلیدی، تجربه کاربری بهتری ایجاد کند.

هدف:

هدف اصلی پروژه **ReelRave**  ایجاد بستری کارآمد و کاربرپسند برای دسترسی به اطلاعات فیلم‌ها و سریال‌ها است. این پروژه قصد دارد با ارائه جزئیات جامع، مانند خلاصه داستان، امتیازات، بازیگران و ژانرها، تجربه‌ای ساده و لذت‌بخش برای کاربران ایجاد کند. همچنین، این پلتفرم فرصتی برای توسعه مهارت‌های برنامه‌نویسی و ترکیب دانش فنی با علاقه شخصی به سینما فراهم می‌کند.

فصل ۲:

تجزیه و تحلیل نیازمندی ها

مقدمه:

در این فصل، نیازمندی‌های پروژه **ReelRave**  به‌طور دقیق بررسی و تحلیل می‌شوند. ابتدا نیازمندی‌های کارکردی و غیرکارکردی تعریف شده و سپس محدودیت‌ها و پیش‌نیازهای فنی مشخص خواهند شد. این تجزیه‌وتحلیل، پایه‌ای محکم برای طراحی و پیاده‌سازی سیستم فراهم می‌کند.

تعریف و بررسی مسئله:

با گسترش صنعت سینما و سریال‌سازی، کاربران برای یافتن اطلاعات دقیق و جامع درباره فیلم‌ها و سریال‌ها به پلتفرم‌های آنلاین روی آورده‌اند. با این حال، بسیاری از وب‌سایت‌ها یا دارای رابط کاربری پیچیده هستند یا اطلاعات ناقصی ارائه می‌دهند. علاوه بر این، دسترسی به پلتفرم‌های بزرگ مانند **IMDb**  یا **Rotten Tomatoes**  ممکن است برای همه کاربران آسان نباشد، به‌خصوص به دلیل نیاز به اشتراک یا محدودیت‌های زبانی.

مسئله دیگر عدم وجود پلتفرمی است که اطلاعات را به شکلی ساده و بدون پیچیدگی‌های اضافی ارائه دهد. کاربران به دنبال ابزاری هستند که بتوانند به سرعت و بدون سردرگمی اطلاعات کلیدی مانند امتیازات، بازیگران، ژانر و خلاصه داستان را پیدا کنند.

هدف از تعریف این مسئله، شناسایی این کمبودها و ایجاد سیستمی است که با طراحی مناسب و فناوری‌های پیشرفته، تجربه‌ای کاربرپسند و دسترسی آزاد به اطلاعات ارائه دهد. ReelRave به‌عنوان پاسخی به این چالش‌ها طراحی شده است تا نیازهای کاربران به ساده‌ترین و کارآمدترین شکل برآورده شود.

نحوه تجزیه و تحلیل:

برای تجزیه و تحلیل این سیستم، از متدولوژی‌های مختلفی استفاده شده است که شامل مراحل زیر می‌شود:

**جمع‌آوری اطلاعات:**

اطلاعات مورد نیاز از بررسی نشریه های دانشگاه ها و سیستم‌های مشابه جمع‌آوری شده است.

**تعریف نیازمندی‌ها:**

شناسایی نیازهای نویسندگان و دانشجویان و الزامات فنی برای توسعه یک بستر کارآمد و موثر.

**مدل‌سازی سیستم:**

ایجاد مدل‌های گرافیکی برای نمایش ساختار و تعاملات بین اجزا.

**تجزیه و تحلیل داده‌ها:**

بررسی داده‌های جمع‌آوری‌شده برای شناسایی الگوها و نیازهای خاص.

معرفی اجزا و ساختار:

کاربران: فیلم ها را بازدید و در مورد ان نظر می‌دهند.

محتوا: فیلم ها که توسط ادمین ها به سایت اضافه می‌شود.

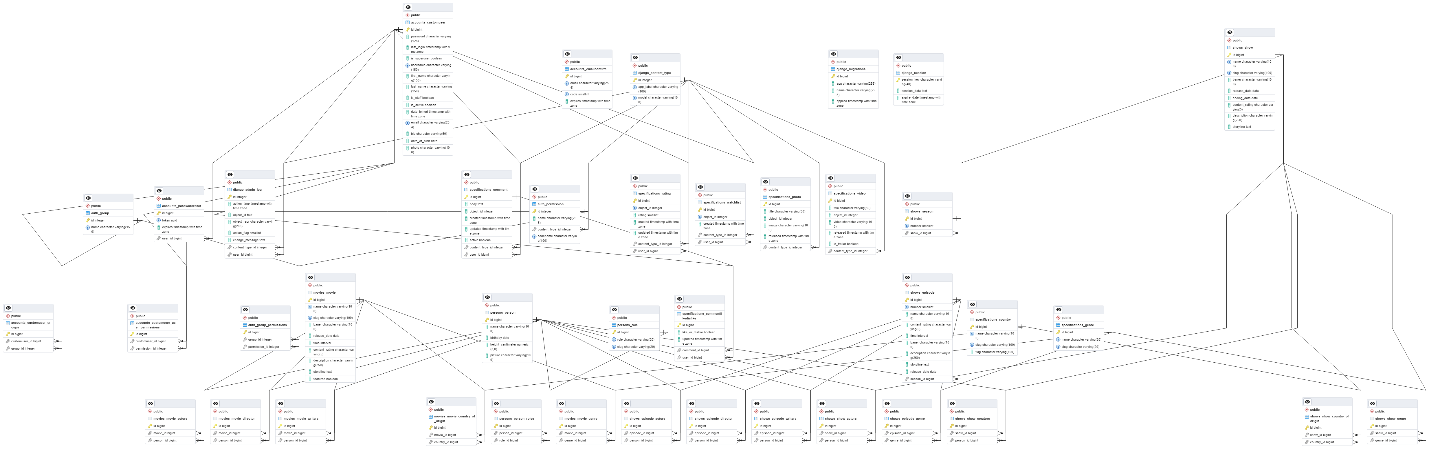
پلتفرم: وبسایت و نرم‌افزارهای مرتبط که امکان اشتراک‌گذاری و مدیریت محتوا را فراهم می‌آورند.

مدیریت: ادمین‌ها و ناظران که نظارت بر کیفیت محتوا و فعالیت‌های کاربران را بر عهده دارند.

توضیح ارتباطات:

ارتباطات بین اجزای مختلف سیستم با استفاده از نمودارهای جریان داده و نمودارهای روابط در زیر نمایش داده شده است.‌

نمودار (ERD) روابط بین موجودیت ها:



نتیجه گیری:

تحلیل دقیق نیازمندی‌ها برای طراحی و پیاده‌سازی پروژه ReelRave ضروری است. این تحلیل نه تنها به شفاف‌سازی اهداف پروژه کمک می‌کند، بلکه مسیر دقیقی را برای توسعه سیستم فراهم می‌آورد. نیازمندی‌ها به صورت فنی و عملکردی شفاف‌سازی شده‌اند و این اطلاعات پایه‌ای برای مراحل بعدی طراحی و پیاده‌سازی خواهند بود.

فصل ۳:

ساختار داده و بانک اطلاعات

مقدمه:

در این فصل، به بررسی ساختار داده‌ها و بانک اطلاعاتی سیستم بلاگ دانشجویی پرداخته خواهد شد. هدف از این فصل، ارائه یک نمای جامع از نحوه سازماندهی داده‌ها، مدل‌های داده‌ای استفاده شده و جزئیات مربوط به طراحی بانک اطلاعاتی است.

ساختار داده ها:

در این بخش، ساختار داده‌ها و جداول بانک اطلاعاتی پروژه بلاگ دانشجویی به تفصیل بررسی خواهد شد. این ساختار شامل جداول مربوط به مدل‌های تعریف‌شده در پروژه و جداول پیش‌فرضی است که توسط Django ایجاد می‌شوند.

جداول و موجودیت های تعریف شده:

**جدول accounts\_customuser:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | bigint | 20 | Not Null | شناسه کاربری بصورت خودکار(کلید اصلی) |
| 2 | Password | varchar | 128 | Not Null | رمز عبور (فعال برای ادمین ها) |
| 3 | Last\_login | tinyint | 1 | - | آخرین ورود کاربر |
| 4 | Is\_superuser | varchar | 150 | Not Null | مشخص کننده ادمین بودن کابر |
| 5 | Username | varchar | 150 | Not Null | نام کاربری(ایندکس شده) |
| 6 | First\_name | varchar | 150 | Not Null | نام |
| 7 | Last\_name | varchar | 150 | Not Null | نام خانوادگی |
| 8 | Email | Varchar | 200 | Not Null | ایمیل کاربر برای احراز هویت(ایندکس) |
| 9 | Is\_staff | tinyint | 1 | Not Null | دسترسی مدیریتی کاربر را مشخص میکند |
| 10 | Is\_active | tinyint | 1 | Not Null | فعال بودن اکانت را مشخص میکند |
| 11 | Date\_joined | datetime | 6 | Not Null | تاریخ ثبت نام |
| 12 | date\_of\_birth | int | 2 | - | سن |
| 13 | photo | varchar | 100 | Not Null | عکس کاربری |

**جدول accounts\_emailconfirm:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | bigint | 20 | Not Null | شناسه رمز عبور تولید شده(کلید اصلی) |
| 2 | email | varchar | 128 | Not Null | ایمیل کاربر |
| 3 | Code | Varchar | 6 | Not Null | کد ارسال شده ب |
| 4 | expires | Datetime | 6 | Not Null | زمان انقضا |

**جدول auth\_permission**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | Bigint | 20 | Not Null | شناسه بصورت خودکار(کلید اصلی) |
| 2 | name | varchar | 255 | Not Null | نام مجوز |
| 3 | Content\_type\_id | int | 11 | Not Null | کلید خارجی جدول django\_content\_type |
| 4 | codename | varchar | 100 | Not Null | کدنام مجوز (ایندکس) |

این جدول مجوزهای مربوط به مدل‌های مختلف را ذخیره می‌کند.

**جدول auth\_group**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | Bigint | 20 | Not Null | شناسه بصورت خودکار(کلید اصلی) |
| 2 | name | varchar | 150 | Not Null | نام گروه |

این جدول گروه‌های کاربری را ذخیره می‌کند که برای مدیریت سطوح دسترسی مختلف به کار می‌رود.

**جدول auth\_group\_permissions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | Bigint | 20 | Not Null | شناسه بصورت خودکار(کلید اصلی) |
| 2 | Group\_id | int | 11 | Not Null | شناسه گروه (کلید خارجی جدول auth\_group ) |
| 3 | Permission\_id | int | 11 | Not Null | شناسه مجوز ( کلید خارجی جدول auth\_permission ) |

این جدول ارتباط بین گروه‌ها و مجوزها را نگهداری می‌کند و نقش کلیدی در سیستم احراز هویت و مدیریت دسترسی‌های Django ایفا می‌کند.

**جدول django\_migrations**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | Bigint | 20 | Not Null | شناسه بصورت خودکار(کلید اصلی) |
| 2 | app | varchar | 255 | Not Null | نام برنامه (مانند مقاله یا کامنت) |
| 3 | name | varchar | 255 | Not Null | نام مایگرشن |
| 4 | applied | datetime | 6 | Not Null | تاریخ اعمال مایگریشن |

این جدول تغییرات و به‌روزرسانی‌های مایگریشن را ثبت می‌کند.

**جداول django\_content\_type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ردیف | نام فیلد | نوع فیلد | طول | مجوز | شرح |
| ۱ | Id | Bigint | 20 | Not Null | شناسه بصورت خودکار(کلید اصلی) |
| 2 | App\_label | varchar | 100 | Not Null | نام برنامه ای که مدل متعلق به آن است |
| 3 | model | varchar | 100 | Not Null | نام مدل |

طراحی:

در این بخش، به توضیح نحوه طراحی دیتابیس پروژه بلاگ دانشجویی با استفاده از فریمورک Django پرداخته خواهد شد. Django یکی از فریمورک‌های محبوب و قدرتمند برای توسعه وب است که امکانات بسیاری را برای طراحی و مدیریت دیتابیس‌ها فراهم می‌آورد. در این پروژه، از قابلیت‌های داخلی Django برای تعریف مدل‌ها و مدیریت دیتابیس استفاده شده است.

تعریف مدل ها در Django:

مدل‌ها در Django، نشان‌دهنده ساختار داده‌ها و نحوه ذخیره‌سازی آن‌ها در دیتابیس هستند. هر مدل به یک جدول در دیتابیس معادل است و فیلدهای مدل به ستون‌های جدول مربوطه تبدیل می‌شوند. برای تعریف مدل‌ها در Django، از ماژول django.db.models استفاده می‌شود.

تصاویر زیر پیاده سازی مدل اکانت و فیلم و کامنت و ... را نشان می دهد.



شکل 1 مدل کاربر



شکل 2 مدل ایمیل کد تایید



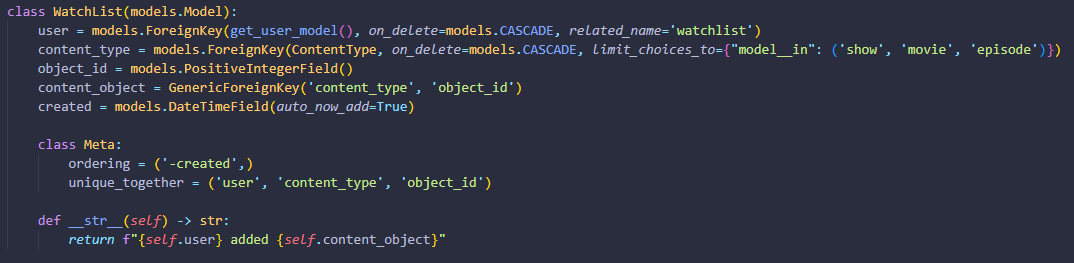
شکل 3 مدل فیلم



شکل 4 مدل کامنت



شکل 5 مدل لایک کامنت



شکل 6 مدل لیست تماشا

فصل ۴:

پیاده سازی

مقدمه:

در این فصل، به بررسی و توضیح جزئیات پیاده‌سازی سیستم بلاگ دانشجویی پرداخته خواهد شد. هدف این فصل ارائه یک دیدگاه جامع از ابزارها و تکنولوژی‌های مورد استفاده، ساختار و نحوه عملکرد بخش‌های مختلف سایت است.

پایگاه داده:

postgresql: برای ذخیره‌سازی داده‌های مرتبط با کاربران، مقالات، نظرات و سایر اطلاعات ضروری از postgresql استفاده شده است. این دیتابیس به دلیل پایداری و کارایی بالا برای پروژه‌های مشابه مناسب است.

برنامه نویسی سمت سرور:

Django 5.0: به عنوان فریمورک اصلی برای توسعه سمت سرور انتخاب شده است. Django به دلیل قابلیت‌های گسترده و امنیت بالا، امکان پیاده‌سازی سریع و مؤثر را فراهم می‌آورد.

کتابخانه های بک‌اند:

djangorestframework==3.15.2 : مهم ترین کتابخانه ای که نصب شده است برای ارسال api با فرمت json به فرانت‌اند.

redis==5.2.0 : برای کش کردن صفحات پر بازدید استفاده شده است.

pillow==11.0.0 : این کتابخانه امکان کار با عکس ها را در پروژه فراهم کرده است.

celery==5.4.0 : برای انجام تسک هایی مانند ارسال ایمیل خوش امد گویی استفاده شده است.

برنامه نویسی سمت کاربر:

Vue: برای راحتی کار از این فریمورک استفاده شده است.

محیط توسعه نرم افزار:

Visual Studio Code (VSCode): به عنوان محیط توسعه یکپارچه (IDE) انتخاب شده است. VSCode با افزونه‌های متنوع، به کد نویسی تمیز و رفع مشکلات در این پروژه کمک میکند.

در این بخش، مراحل مختلف پیاده‌سازی پروژه بلاگ دانشجویی به تفصیل شرح داده خواهد شد. هدف از این بخش، ارائه یک نمای کلی از فرآیند توسعه پروژه، ابزارها و تکنولوژی‌های مورد استفاده و گام‌های کلیدی است که برای ساخت این سیستم انجام شده است.

روش پیاده سازی:

در این بخش، مراحل مختلف پیاده‌سازی پروژه به تفصیل شرح داده خواهد شد. هدف از این بخش، ارائه یک نمای کلی از فرآیند توسعه پروژه، ابزارها و تکنولوژی‌های مورد استفاده و گام‌های کلیدی است که برای ساخت این سیستم انجام شده است.

تعریف برنامه ها و مدل ها:

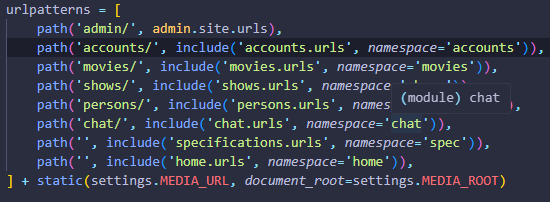
با توجه به تجزیه و تحلیل انجام شده در بخش قبلی، نیازمندی‌های سیستم به دقت مشخص شد. بر اساس این نیازمندی‌ها، نمودارهای جریان داده و ارتباطات موجودیت‌ها تهیه شدند.

در این مرحله، برنامه‌های مورد نیاز با استفاده از دستور python manag.py startapp appname در Django ایجاد شدند. هر برنامه (app) به یک بخش خاص از سیستم اختصاص دارد و وظایف مشخصی را انجام می‌دهد

**تعریف و تنظیم آدرس ها UrlConf:**

در Django، یکی از بخش‌های مهم و کلیدی هر پروژه، مدیریت آدرس‌ها و مسیرها است. این بخش به وسیله سیستم URLconf (URL configuration) انجام می‌شود که شامل تعریف الگوهای URL و ارتباط آنها با نماها است. URLconf به شما امکان می‌دهد تا درخواست‌های ورودی به برنامه خود را به درستی پردازش و پاسخ‌های مناسب را ارسال کنید.

هر برنامه (app) در پروژه Django معمولاً یک فایل urls.py دارد که الگوهای URL مربوط به آن برنامه را تعریف می‌کند. این فایل شامل یک لیست از الگوهای URL است که به نماهای مختلف متصل می‌شوند.

****

شکل 7 این فایل معمولاً به عنوان نقطه ورودی اصلی برای مدیریت مسیرها عمل می‌کند



شکل 8 مسیر های مربوط به ورود و ثبت نام



شکل 9 مسیر های مربوط به کامنت و لیست تماشا



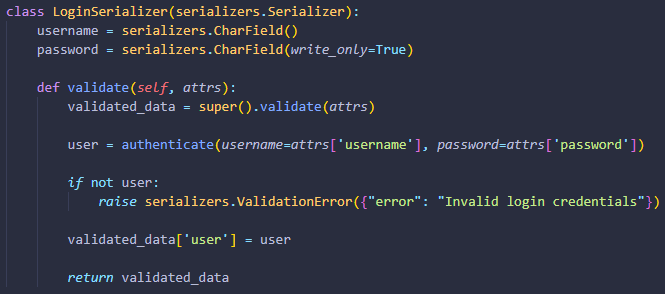
شکل 10 مسیر های مربوط به فیلم

**تعریف serializer ها:**

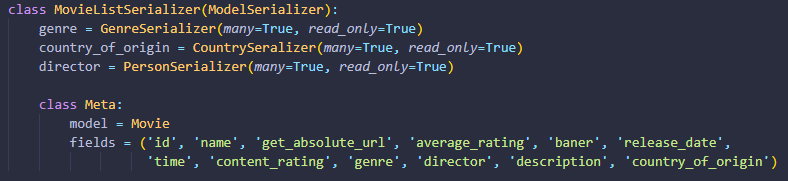
برای تبدیل ابجکت های python به فایل json و ازسال ان در نماها



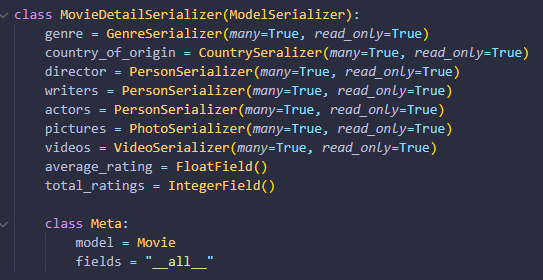
شکل 11 ثبت نام serializer



شکل 12 ورود seiralizer



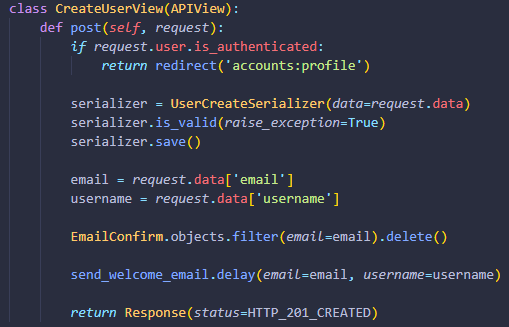
شکل 13 لیست فیلم ها serializer



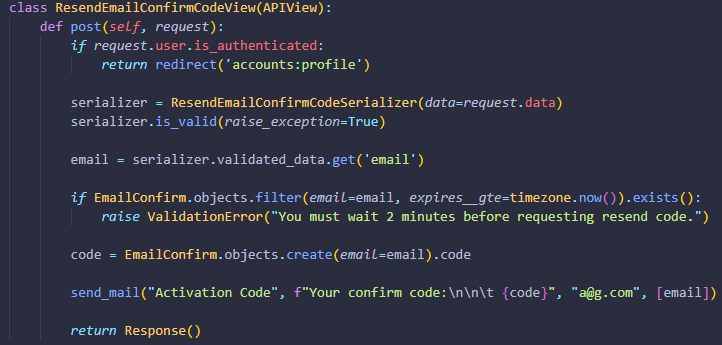
شکل 14 جزئیات فیلم serializer

**تعریف نماها:**

بعد از تعریف مسیر ها و تنظیم آنها ، نوبت به ایجاد نما برای مسیر های مربوطه است. نماها توابع یا کلاس‌هایی هستند که درخواست‌های HTTP را پردازش کرده و پاسخ‌های مناسب را برمی‌گردانند. هر الگوی URL به یک نمای مشخص متصل است. نماها معمولاً در فایل views.py تعریف می‌شوند.



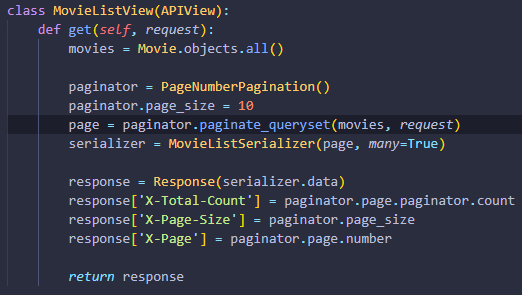
شکل 15 نمای مربوط به ثبت نام



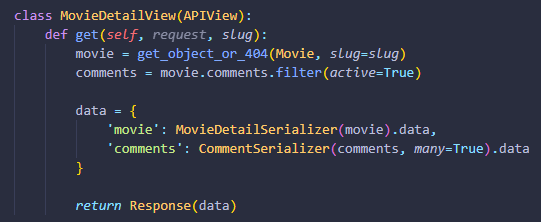
شکل 16 نمای مربوط به ارسال کد تایید



شکل 17 نمای مربوط به ورود



شکل 18 نمای مربوط به لیست فیلم ها



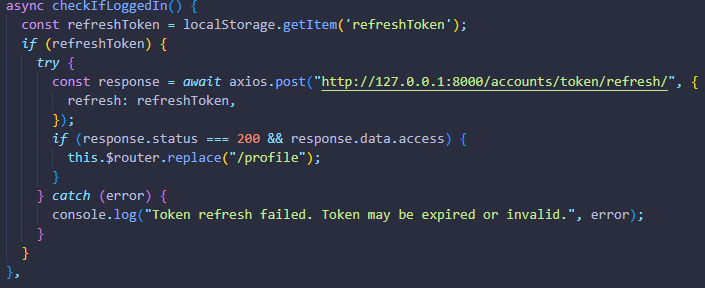
شکل 19 نمای مربوط به جزئیات فیلم

متوذ های مربوط به vue:

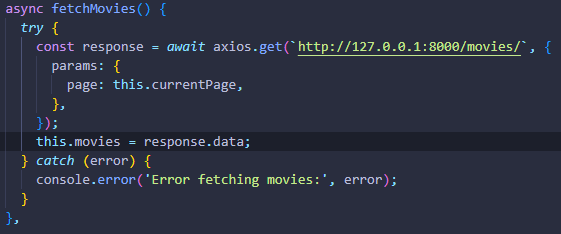
در این بخش متود های مربوط به فرانت اند که درخواست api میدهند بررسی میشود



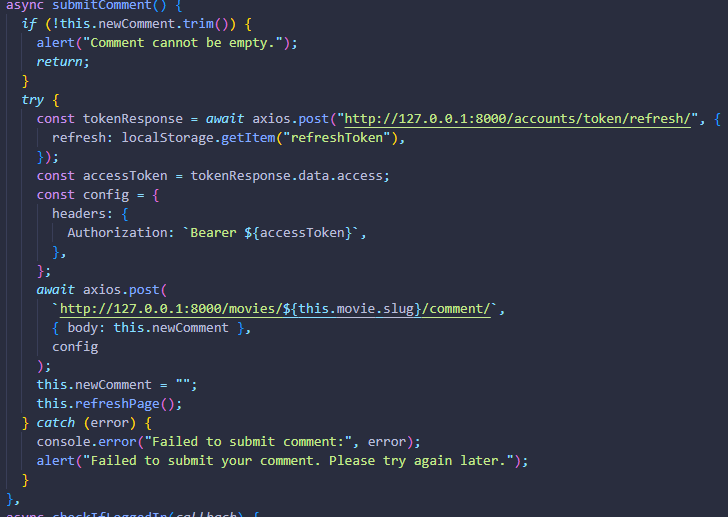
شکل 20 متود مربوط به لاگین



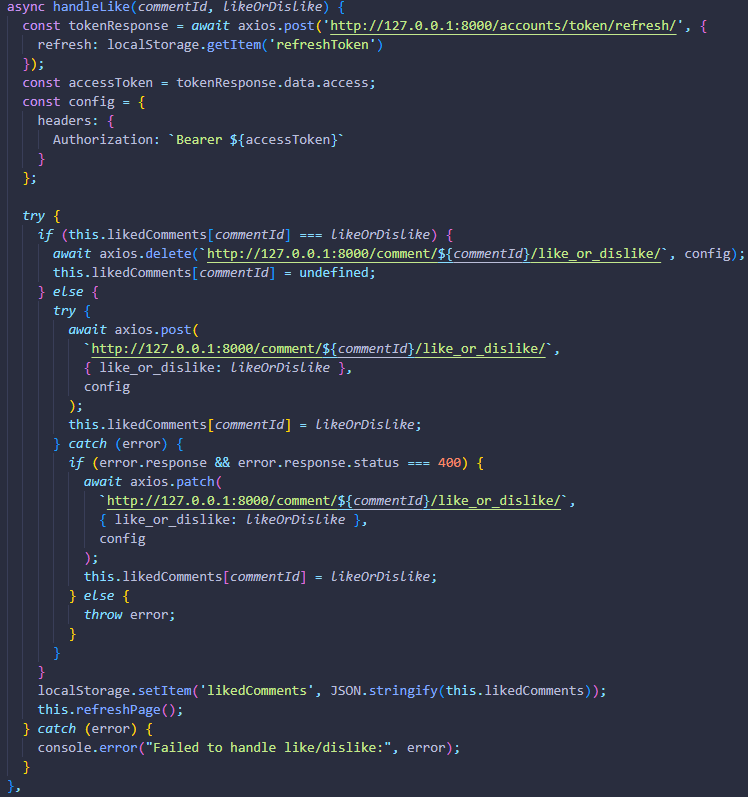
شکل 21 متود بررسی لاگین



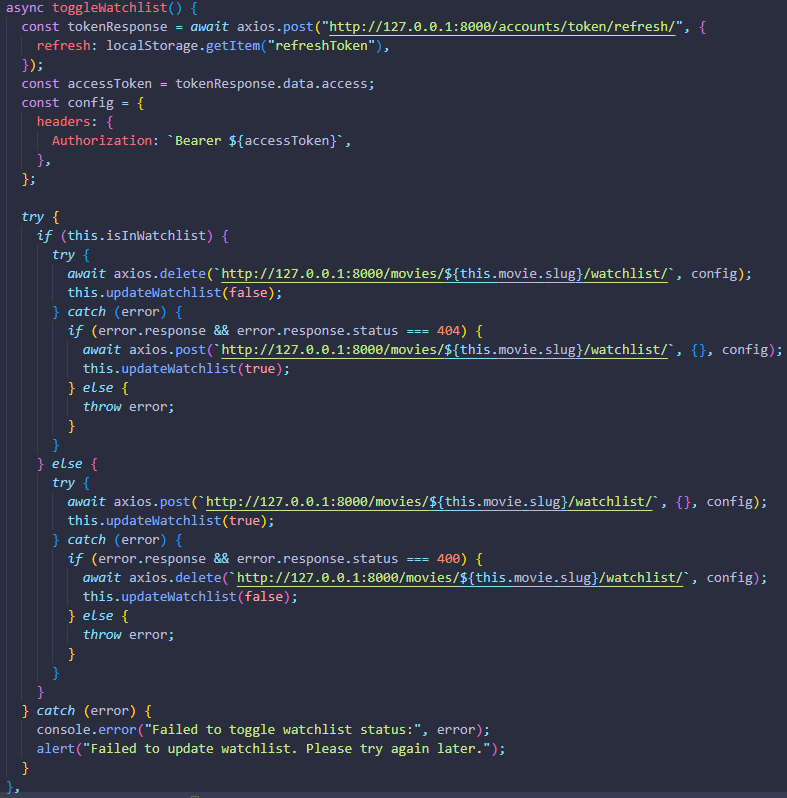
شکل 22 متود دریافت فیلم



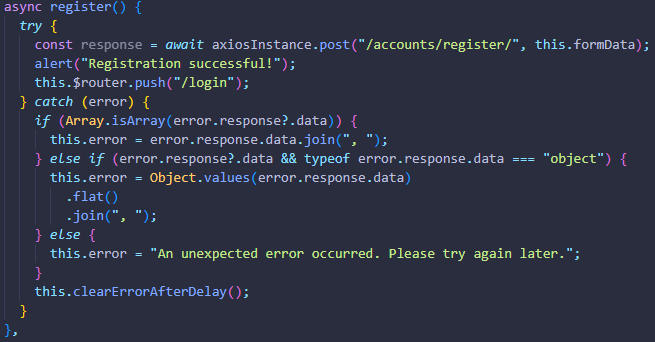
شکل 23 متود ثبت کامنت



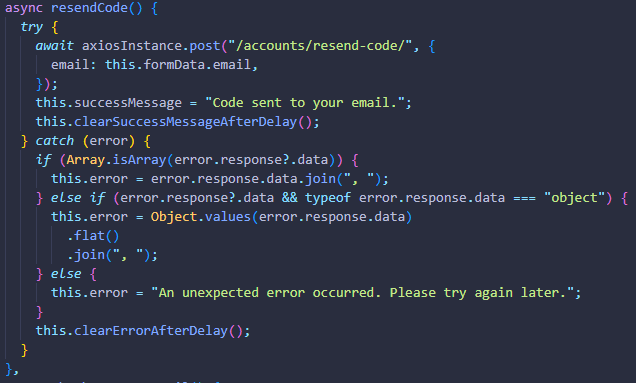
شکل 24 متود لایک کامنت



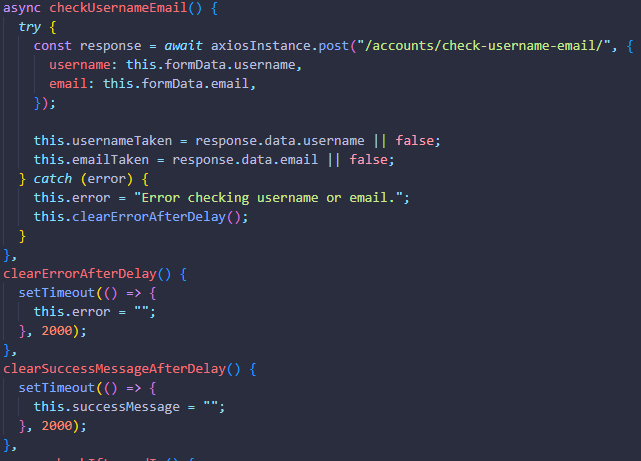
شکل 25 متود افزودن به لیست تماشا



شکل 26 متود ثبت نام



شکل 27 متود درخواست کد تایید



شکل 28 متود بررسی نام کاربری و ایمیل

فصل ۵:

جمع بندی

نتیجه گیری:

پروژه **ReelRave**  با هدف ایجاد یک پلتفرم کارآمد و کاربرپسند برای دسترسی به اطلاعات فیلم‌ها و سریال‌ها طراحی و توسعه یافت. در دنیای امروز که اطلاعات به بخش جدایی‌ناپذیر زندگی انسان‌ها تبدیل شده، نیاز به سیستم‌هایی که اطلاعات دقیق، جامع و به‌روز را به ساده‌ترین شکل ممکن ارائه دهند، بیش از پیش احساس می‌شود. ReelRave با بهره‌گیری از فناوری‌های پیشرفته مانند **Django REST Framework**  برای بک‌اند و **Vue.js**  و **Vuetify**  برای فرانت‌اند، تلاشی برای پاسخ به این نیازها بوده است.

در مراحل توسعه پروژه، نیازمندی‌های کاربران به دقت مورد تجزیه و تحلیل قرار گرفت و بر اساس آن‌ها، طراحی و پیاده‌سازی سیستم انجام شد. در بخش نیازمندی‌های کارکردی، تمرکز بر ارائه اطلاعات کلیدی مانند امتیازها، خلاصه داستان، بازیگران و ژانر بود. همچنین، نیازمندی‌های غیرکارکردی همچون سرعت پاسخ‌دهی، رابط کاربری ساده و تجربه کاربری جذاب نیز به‌طور ویژه در نظر گرفته شد.

این پروژه نه تنها فرصتی برای ترکیب دانش فنی و علاقه شخصی به صنعت سینما فراهم کرد، بلکه مهارت‌های عملی مانند توسعه وب، مدیریت پروژه و طراحی سیستم را نیز تقویت کرد. چالش‌هایی مانند انتخاب ابزارهای مناسب و اطمینان از هماهنگی بین بخش‌های مختلف سیستم به کمک تحقیق و آزمون و خطا حل شدند.

در نهایت، ReelRave نه تنها به عنوان یک پروژه آکادمیک، بلکه به‌عنوان نمونه‌ای واقعی از کاربرد فناوری برای حل مشکلات روزمره، موفقیت‌آمیز بوده است. با ادامه توسعه و اضافه کردن امکانات جدید، این پلتفرم می‌تواند به ابزاری کاربردی برای کاربران علاقه‌مند به فیلم و سریال تبدیل شود و نقش مؤثری در بهبود تجربه آن‌ها ایفا کند.

پیشنهاد هایی برای کار های آتی:

با توجه به موفقیت اولیه پروژه **ReelRave** در ارائه اطلاعات جامع و کاربرپسند درباره فیلم‌ها و سریال‌ها، گام‌های آینده می‌توانند به ارتقای قابلیت‌ها و بهبود تجربه کاربری این پلتفرم کمک کنند. برخی از پیشنهادهای مهم برای توسعه‌های آتی به شرح زیر است:

1. **افزودن سیستم ثبت‌نام و ورود کاربران**: با ایجاد پروفایل‌های شخصی، کاربران می‌توانند لیست علاقه‌مندی‌ها، نقدها و امتیازات خود را ذخیره کنند. این قابلیت باعث ایجاد حس تعامل و شخصی‌سازی خواهد شد.
2. **پشتیبانی از چند زبان**: اضافه کردن زبان‌های دیگر به پلتفرم می‌تواند دسترسی کاربران جهانی را افزایش دهد و تجربه‌ای بهتر برای کاربران غیرانگلیسی‌زبان ایجاد کند.
3. **یکپارچگی با APIهای خارجی**: با اتصال به APIهای معتبر مانند TMDb یا IMDb، امکان ارائه اطلاعات به‌روزتر و دقیق‌تر درباره فیلم‌ها و سریال‌ها فراهم خواهد شد.
4. **افزودن قابلیت جستجوی پیشرفته**: ارائه فیلترهای پیشرفته مانند ژانر، سال تولید، بازیگران و امتیاز می‌تواند جستجوی کاربران را بهینه کند.
5. **توسعه اپلیکیشن موبایل**: با طراحی یک اپلیکیشن موبایل برای پلتفرم، دسترسی کاربران در هر زمان و مکان ساده‌تر خواهد شد.
6. **افزودن سیستم پیشنهاد هوشمند**: با استفاده از الگوریتم‌های یادگیری ماشین، می‌توان فیلم‌ها و سریال‌هایی را که با علاقه کاربران هماهنگی دارند پیشنهاد داد.

این اقدامات نه تنها قابلیت‌های ReelRave را گسترش می‌دهند، بلکه می‌توانند آن را به یک پلتفرم پیشرو در این حوزه تبدیل کنند

جمع بندی:

پروژه **ReelRave**  با هدف ارائه بستری ساده، کارآمد و جامع برای دسترسی به اطلاعات فیلم‌ها و سریال‌ها طراحی و پیاده‌سازی شده است. این پلتفرم با بهره‌گیری از تکنولوژی‌های مدرن مانند **Django REST Framework**  و **Vue.js**، تجربه‌ای کاربرپسند فراهم می‌کند. تحلیل نیازمندی‌ها، طراحی اصولی و تمرکز بر حل مشکلات واقعی کاربران، این پروژه را به نمونه‌ای ارزشمند از کاربرد دانش برنامه‌نویسی در دنیای واقعی تبدیل کرده است. ReelRave می‌تواند به نیازهای کاربران علاقه‌مند به سینما پاسخ دهد.