

در این سری تمرینات از شما می خواهیم تا یک وبسایت مشابه [IMDB](#) پیاده سازی کنید. این وب سایت شامل حداقل موارد زیر است:

- صفحه Login و Signup
- صفحه اصلی سایت (Home page)
- شامل فیلم و سریال ها مرتب شده بر اساس تاریخ Release و...
- صفحه فیلم و سریال
- شامل اطلاعات کامل فیلم و سریال
- صفحه پروفایل کاربر
- اطلاعات کاربر (نام کاربری و ...)
- فیلم و سریال هایی که کاربر به آنها امتیاز داده
- لیست فیلم و سریال هایی که کاربر به watch list اضافه کرده
- صفحه افراد مشهور
- بیوگرافی
- فیلم/سریال های مرتبط با شخص

در این تمرین قصد داریم تا مدل های دیتابیزی این وب سایت را ایجاد کنیم. حداقل مدل های موردنظر برای پیاده سازی به شرح زیر است:

مدل ها

1. مدل User

Description	Data Type	Field Name
شناسه یکتا	INT (PK)	id
نام کاربری	VARCHAR	username
ایمیل کاربر	VARCHAR	email
رمز عبور (hashed)	VARCHAR	password
تاریخ ایجاد حساب	TIMESTAMP	created_at
تاریخ آخرین به روزرسانی	TIMESTAMP	updated_at

2. مدل Movie

Description	Data Type	Field Name
شناسه یکتا	INT (PK)	id
عنوان فیلم	VARCHAR	title
توضیحات فیلم	TEXT	description
تاریخ انتشار	DATE	release_date
نمره کلی فیلم (میانگین)	FLOAT	rating
مدت زمان (دقیقه)	INT	duration
تاریخ اضافه شدن فیلم	TIMESTAMP	created_at
تاریخ آخرین به‌روزرسانی	TIMESTAMP	updated_at

3. مدل Genre

Description	Data Type	Field Name
شناسه یکتا	INT (PK)	id
نام ژانر	VARCHAR	name

4. مدل Comment

Description	Data Type	Field Name
شناسه یکتا	INT (PK)	id
متن نظر	TEXT	content
تاریخ ارسال نظر	TIMESTAMP	created_at
کلید خارجی از جدول User	INT (FK)	user_id
کلید خارجی از جدول Movie	INT (FK)	movie_id

5. مدل Trailer

Description	Data Type	Field Name
شناسه یکتا	INT (PK)	id
لینک ویدیو تریلر	VARCHAR	url
کلید خارجی از جدول Movie	INT (FK)	movie_id

6. مدل UserMovie (جدول واسط برای علاقه‌مندی‌ها)

Description	Data Type	Field Name
شناسه یکتا	INT (PK)	id
کلید خارجی از جدول User	INT (FK)	user_id
کلید خارجی از جدول Movie	INT (FK)	movie_id
نمره کاربر به فیلم	FLOAT	rating
وضعیت مشاهده (دید شده/نشده)	BOOLEAN	watched_status
تاریخ اضافه شدن به علاقه‌مندی‌ها	TIMESTAMP	created_at

7. مدل GenreMovie (جدول واسط برای ژانرها)

Description	Data Type	Field Name
کلید خارجی از جدول Genre	INT (FK)	genre_id
کلید خارجی از جدول Movie	INT (FK)	movie_id

ارتباطات

ارتباطات زیر را بین مدل‌ها برقرار کنید) تشخیص این که این ارتباطات با کدام روش *one-to-one*, *one-to-many*, ... خواهد بود بر عهده شماست)

- ارتباط بین مدل User و Movie برای پیاده‌سازی بخش علاقه‌مندی‌ها: این ارتباط صرفاً یک جدول pivot خالی نیست و نیاز به attribute‌هایی مثل نمره کاربر به یک فیلم و وضعیت مشاهده و... را دارا است.

- ارتباط بین مدل Comment و Movie

- ارتباط بین مدل User و Comment

- ارتباط بین مدل Genre و Movie

- ارتباط بین مدل Movie و Trailer

نکات

- نیازی به اتصال تمرین به پروژه فرانت نیست و صرفاً مدل های خواسته شده را در سمت بکند پیاده سازی کنید

- نیازی به استفاده از serializer نیست

- همانطور که گفته شد، فیلدهای مربوط به اتصالات (مانند foreign key ها و ...) در مدل ها نیامده است. طبیعتاً این روابط را باید خودتان اضافه کرده و فیلد(های) موردنیاز را به مدل ها اضافه کنید.

- در صورت نیاز می توانید فیلدها و یا مدل های دیگری به تمرین اضافه کنید.

-در ادامه ی تمرین قبلی بکند از شما می خواهیم که به وسیله ی شخصی سازی ادمین پنل جنگو، امکانات مدیریتی سایت را پیاده سازی کنید. خواسته های این تمرین به شرح زیر است:

- در پنل باید یک صفحه وجود داشته باشد که در آن بتوانیم لیست ویدئوهای علاقه مندی یک کاربر را ببینیم و براساس نمره و تاریخ اضافه شدن، قابل فیلتر باشد.

- در پنل باید یک صفحه وجود داشته باشد که بتوانیم یک فیلم را با تمام اطلاعات موردنیاز، به لیست علاقه مندی های یک کاربر اضافه کنیم.

- در یک صفحه باید لیست کامنت هایی که برای یک فیلم ثبت شده است را بتوانیم مشاهده کنیم. کامنت ها باید قابل حذف باشند. کامنت ها باید براساس تاریخ ارسال قابل فیلتر باشند.

- یک صفحه برای نمایش لیست کاربران ایجاد کنید. این لیست باید براساس نام کاربر قابل جستجو باشد.

- در یک صفحه مخصوص یک فیلم، باید بتوانیم یک ژانر به آن فیلم اضافه کنیم.

-در ادامه ی تمرین قبلی بکند از شما می خواهیم که سه بخش URL, view, template سایت را پیاده سازی کنید. خواسته های این تمرین به شرح زیر است:

بخش Urls

حداقل url های زیر برای پروژه موردنیاز است:

1. (بخش لاگین و ثبت نام) /auth

2. /titles (نمایش همه عناوین)

3. /title/<imdb_id> (tt نمایش یک عنوان با آیدی، آیدی باید حتما با)

4. /top-movies/ (Top 250 movies)

5. /top-series/ (Top 250 series)

6. /profile/ (پروفایل کاربر لاگین کرده)

7. /profile/<user_id> (پروفایل کاربر مشخص)

- در صفحاتی که نیاز به دسترسی خاصی وجود دارد (مثلا آدرس /profile که فقط کاربر ثبت نام شده است) باید دسترسی مورد نظر بررسی شود.
- هنگامی که کاربر در برنامه لاگین است، نباید بتواند وارد صفحات لاگین و ثبت نام شود.
- اگر آدرس(های) دیگری نیاز بود، می توانید به لیست url ها اضافه کنید و محدودیتی در این زمینه ندارید.

بخش View و Template

در این بخش از شما انتظار می رود که صفحات html ای که در تمرینات سمت کلاینت تولید شده اند را به پروژه ی Django تان اضافه کنید. موارد زیر را برای این بخش در نظر بگیرید:

- فایل های CSS و JS باید در پروژه Django وارد شده و در فولدرهای assets موردنظر قرار گیرند.
- به ازای هر صفحه view های موردنظر را در فایل(های) views.py ایجاد کنید.
- احراز هویت و امنیت فرم ها را با csrf token تضمین کنید.
- دقت کنید اگر جایی نیاز به نمایش یک بخش از صفحه به صورت شرطی (conditional) و نمایش یک سری المان به صورت تعدادی بود، باید از ابزارهای جنگو برای نمایش آن ها استفاده کنید و کدهای JS مربوط به آن ها را پاک کنید. به طور مثال اگر می خواهید لیست کارت ها را در صفحه جستجو نمایش دهید، نباید تعداد و داده های لیست ها را به وسیله DOM آپدیت کنید؛ بلکه نمایش و آپدیت کارت ها باید به وسیله کدهای django داخل فایل های html مدیریت شوند.
- توابعی که مربوط به CRUD مدل های پیاده سازی شده هستند هم باید به view اضافه شوند.

نکات:

- حالا که پروژه سمت بکند ایجاد شده است، داده ها باید کاملا واقعی باشند و دیگر نباید داده ها را شبیه سازی یا به اصطلاح mock کنید. تنها داده هایی را نمایش دهید که واقعا در جداول دیتابیس موجود باشند و به وسیله ادمین پنل داده تولید کرده اید.

- برای بخش ثبت نام، از ساختارهای authentication استفاده نکنید. صرفا اطلاعات کاربر لاگین شده را در جایی ذخیره کرده و هر جایی که نیاز بود، از همان جا داده ها را خوانده و نمایش دهید.

-در این تمرین می خواهیم یک RestAPI برای مینی پروژه سایت IMDB توسعه دهیم.

موارد endpoint ای که باید پیاده سازی کنید به صورت زیر است: پیاده سازی endpoint های بیشتر با تایید حل تمرین، شامل نمره اضافه می شود)

دقت کنید که برای هر endpoint باید HTTP Method مناسب استفاده کنید و تشخیص method مناسب به عهده خودتان است.

- `api/user`: دریافت لیست تمام کاربران (فقط توسط ادمین)
- `api/user/<id>`: دریافت مشخصات کاربر با شناسه `id` فقط توسط ادمین)
- `api/user/<id>`: پاک کردن کاربر (فقط توسط ادمین)
- `api/user/<id>`: آپدیت مشخصات کاربر (حتما توسط خود کاربر یا ادمین)
- `api/auth/login`: ورود کاربر
- `api/auth/signup`: ثبت نام کاربر
- `api/user/<id>/list/`: نمایش لیست فیلم و سریال های موردعلاقه کاربر
- `api/user/<id>/list/add`: اضافه کردن فیلم به لیست فیلم و سریال های مورد علاقه
- `api/title/`: نمایش لیست تمام عناوین (با صفحه بندی)

نکته: این endpoint دو ورودی `page` و `count` به صورت `query string` دارد.

مثال `api/title/?page=1&count=10`:

نکات

- حتما از [django rest framework](https://www.djangoproject.com/) استفاده کنید.
- برای ثبت نام باید از روال های احراز هویت معتبر مانند `jwt token` و `oauth token` و ... استفاده کنید.
- ورودی و خروجی های هر endpoint به عهده ی خودتان می باشد.

- پیشنهاد می شود در تمرین از app های مختلف برای مرتب تر شدن کدها استفاده کنید.
- برای api هایی که نیاز به لاگین بودن کاربر دارند، middleware موردنظر را پیاده سازی کنید.
- برای api هایی که نیاز به لاگین بودن ادمین دارند، middleware موردنظر را پیاده سازی کنید.
- حتما از serializer ها استفاده شود.
- در صورت نیاز به تغییر endpoint های پیشنهادی، حتما با حل تمرین هماهنگ کنید.
- دیپلوی کردن برنامه روی سرورهای رایگان/شخصی نمره اضافه دارد. دقت کنید در این حالت باید حالت CORS را مدیریت کنید.