

مدیریت پروژه

در این تمرین از شما می‌خواهیم چند مدل ساده را پیاده‌سازی کنید.

پروژه‌ی اولیه

پروژه‌ی اولیه را از [این لینک](#) دانلود کنید. ساختار فایل‌های این پروژه به‌صورت زیر است.

quera_challenge/

Copy

Plain text

```
├─ manage.py
├─ requirements.txt
├─ app
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations
│   │   ├── __init__.py
│   │   └─ 0001_initial.py
│   ├── > models.py<
│   ├── > serializers.py<
│   ├── tests.py
│   ├── urls.py
│   └─ > views.py <
├─ config
│   ├── __init__.py
│   ├── asgi.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └─ auth.py
```

پایگاه داده و مدل‌هایی که باید تکمیل شوند

پروژه (Project)

- عنوان (رشته‌ی title): عنوان پروژه
- شرح (رشته‌ی description): توضیحات پروژه
- اعضاء (رابطه چندبه‌چند (ManyToMany) با User از طریق مدل شخصی‌سازی‌شده‌ی Membership): اعضای شرکت‌کننده در پروژه
- دسته‌بندی (کلید خارجی به Category): دسته‌بندی مرتبط با پروژه
- تاریخ ایجاد (زمان created_at): زمان ایجاد پروژه
- تاریخ آخرین به‌روزرسانی (زمان updated_at): زمان آخرین به‌روزرسانی پروژه

دسته‌بندی (Category)

- نام (رشته name): نام دسته‌بندی
- والد (این کلید با نام parent به خود مدل Category وصل می‌شود)

نقش (RoleType)

- این مدل به صورت IntegerChoices پیاده‌سازی می‌شود.
- "MEMBER" = 0, "member": عضوی که دسترسی معمولی دارد و تنها به‌عنوان شرکت‌کننده در پروژه حضور دارد.
- "OWNER" = 1, "owner": مالک پروژه که دسترسی مدیریتی و تصمیم‌گیری در پروژه را دارد.

عضویت (Membership)

- **person (کلید خارجی به User):** فرد شرکت‌کننده در پروژه
- **project (کلید خارجی به Project):** پروژه‌ای که فرد در آن شرکت دارد
- **نقش (عدد صحیح role):** نقش فرد در پروژه (عضو یا مالک) که از بین گزینه‌های مدل Roletype انتخاب می‌شود.

سریالایزر و ویو

سریالایزر ProjectSerializer

• فیلدها:

- **title :** عنوان پروژه
- **description :** توضیحات پروژه
- **category :** دسته‌بندی پروژه
- **members :** فهرستی از *id* اعضای پروژه
- **created_at :** تاریخ ایجاد پروژه
- **updated_at :** تاریخ آخرین به‌روزرسانی پروژه

ویوی ProjectListView

این کلاس یک ویوی لیستی از پروژه‌ها را فراهم می‌کند که در آن کاربران احراز هویت شده می‌توانند پروژه‌های موجود را مشاهده کنند.

- از `ListAPIView` از جنگو رست فریم‌ورک برای ارائه یک لیست از اشیاء مدل‌ها استفاده می‌شود.
- تنها کاربران احراز هویت شده با استفاده از کلاس `IsAuthenticated` اجازه دسترسی به این ویو را دارند.

- تمامی پروژه‌های ذخیره‌شده در دیتابیس بازگردانده می‌شوند.
- این ویو از ProjectSerializer برای تبدیل داده‌های مدل به فرمت مناسب JSON استفاده می‌کند.

نکات مهم

- شما تنها مجوز ایجاد تغییرات در فایل‌های serializers.py ، views.py ، models.py و را دارید و تمامی تغییرات دیگر شما در فایل‌های پروژه نادیده گرفته خواهند شد.
- فیلدهای project و person در مدل MemberShip باید unique باشند.
- رکوردها در مدل Project براساس تاریخ آخرین تغییر order می‌شوند.
- مدل‌های User و Project از طریق مدل MemberShip رابطه‌ی ManyToMany برقرار می‌کنند.

آنچه باید آپلود کنید

شما باید یک فایل ZIP با ساختار مشابه با پروژه اولیه را آپلود کنید.