
بخش ۱ URL — View

❏

یک view به نام `about_view` بساز که یک قالب `about.html` را رندر کند.
برای آن یک URL با نام `about` تعریف کن و در قالب `home` لینکی به آن بده.

❏

یک URL با پارامتر `post_id` بساز (مثلاً `/posts/۵/`) و مقدار آن را در صفحه نمایش بده.

❏

یک view بساز که لیست ۵ نام را داخل `context` بفرستد و در قالب با `for` نمایش بده.

❏

یک لینک در قالب بساز که با استفاده از `{% url %}` به صفحه جزئیات یک محصول با `id=۱۰` برود.

❏

یک view بنویس که اگر کاربر پارامتر `?name=ali` در URL ارسال کرد، نام را در صفحه نمایش دهد
(با استفاده از `request.GET`).

بخش ۲ Django Admin —

❏

مدل Category با فیلدهای name و created_at بساز و آن را در admin ثبت کن.

۷

برای مدل Product، در پنل ادمین این موارد را تنظیم کن:

- نمایش name, price, stock
 - امکان جستجو بر اساس name
 - فیلتر بر اساس created_at
-

۸

برای مدل Order یک ModelAdmin بنویس که:

- مرتب‌سازی پیش‌فرض بر اساس created_at باشد
 - date_hierarchy فعال باشد
-

۹

یک اکشن سفارشی بساز که محصولات انتخاب‌شده را غیرفعال کند. (is_active=False)

10

برای مدل Book و Chapter که رابطه ForeignKey دارند، Chapter را به صورت Inline داخل BookAdmin نمایش بده.

11

یک فرم ساده بساز با فیلدهای:

- username
- email
- age

و آن را در قالب نمایش بده.

12

فرمی بساز که اگر سن کمتر از ۱۸ بود، خطای سفارشی بدهد (اعتبارسنجی دستی).

13

یک فرم تماس بساز و بعد از ارسال موفق، کاربر را به صفحه `thank_you` ریدایرکت کن.

14

یک `ModelForm` برای مدل `Article` بساز که فقط فیلدهای `title` و `content` را داشته باشد و بعد از ذخیره به لیست مقالات برگردد.

15

یک فرم آپلود فایل بساز که یک تصویر بگیرد و آن را در دیتابیس ذخیره کند (با استفاده از `ImageField` و تنظیم `MEDIA_ROOT`).

چالش

یک سیستم ساده بساز که:

- کاربر بتواند پست ایجاد کند (ModelForm)
- پست‌ها در صفحه لیست نمایش داده شوند
- هر پست صفحه جزئیات داشته باشد
- همه چیز از Named URL استفاده کند
- مدل در admin سفارشی شده باشد