
بخش ۱ — URL و View

۱

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.

۲

یک URL با پارامتر `post_id` بساز (مثلاً `/posts/5/`) و مقدار آن را در صفحه نمایش بده.

۳

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

۴

یک لینک در قالب بساز که با استفاده از `{% url %}` به صفحه جزئیات یک محصول با `id=10` برود.

۵

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

— Django Admin ۲ بخش

۶

مدل Category با فیلد های name و created_at و آن را در admin ثبت کن.

□

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
-

□

برای مدل Order یک ModelAdmin بنویس که:

- مرتب سازی پیش فرض بر اساس created_at باشد
 - date_hierarchy باشد
-

□

یک اکشن سفارشی بساز که محصولات انتخاب شده 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.

— Forms ۳ بخش

10

یک فرم ساده بساز با فیلدهای:

username •

email •

age •

و آن را در قالب نمایش بده.

11

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

12

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

13

یک فرم تماس بساز که فقط فیلدهای title و content را داشته باشد و بعد از ذخیره به لیست مقالات برگردد.

14

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

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