

## Cíl:

Rozšířte aplikaci ToDo vzvinutou na lekci o níže uvedené mynosti

## 1. Obnovení smazaných úkolů

**Zadání:** Implementujte funkci pro vrácení smazaných úkolů zpět do aktivního seznamu.

- Vytvořte novou view `restore(request, pk)`
- Přidejte tlačítko "Obnovit" v historii smazaných úkolů
- Nastavte `deleted=False` a `deleted_at=None`
- Přidejte URL pattern `restore/<int:pk>/`

## 2. Trvalé smazání úkolů

**Zadání:** Umožněte definitivní smazání úkolů z databáze (pouze již smazané úkoly).

- Vytvořte view `permanent_delete(request, pk)`
- Ověřte, že úkol je již označen jako smazaný (`deleted=True`)
- Použijte `todo.delete()` pro trvalé odstranění z databáze
- Přidejte potvrzovací dialog před smazáním




## 3. Kategorie/štítky úkolů

**Zadání:** Rozšířte model o možnost přiřazení barevných kategorií úkolům.

- Přidejte pole `kategorie` do modelu (choices: Práce, Škola, Osobní, Nákupy, atd.)
- Přidejte filtrování úkolů podle kategorií na hlavní stránce
- Zobrazujte kategorie barevně odlišenými štítky
- Upravte formuláře pro přidání/editaci

## 4. Priorita úkolů

**Zadání:** Implementujte systém priorit pro úkoly.

- Přidejte pole priorit (Vysoká, Střední, Nízká)
- Seřaďte úkoly podle priority (vysoká priorita nahoře)
- Zobrazujte prioritu pomocí ikon nebo barev (    )
- Umožněte filtrování podle priority

## 5. Vyhledávání úkolů

**Zadání:** Přidejte vyhledávací funkci pro rychlé nalezení úkolů.

- Vytvořte vyhledávací formulář na hlavní stránce
- Implementujte vyhledávání v textu úkolu a poznámce
- Použijte Django `__icontains` pro case-insensitive vyhledávání
- Zvýrazněte nalezené výrazy ve výsledcích

---

Materiály z použité v lekci naleznete zde: [elsnoxx.github.io](https://elsnoxx.github.io)

Výslednou aplikaci odevzdejte do [MyStatu](#).