

الجمهورية اليمنية

جامعة إب

كلية العلوم

قسم علوم الحاسوب وتقنية المعلومات



## تكاليف مقرر

هندسة برمجيات - عملى

Software Engineering

المحاضرة الخامسة

عمل الطالب :

أسامي سعيد محمد حمود سعيد - مجموعة A

إشراف :

م. مالك المصنف

2025 - 2026

## 1- أنواع الـ Query Set في Django

النوع	الوصف	المثال
<code>all()</code>	جلب جميع السجلات	<code>Student.objects.all()</code>
<code>filter()</code>	فلترة حسب شرط	<code>Student.objects.filter(age__gte=18)</code>
<code>exclude()</code>	استثناء سجلات	<code>Student.objects.exclude(stage="first")</code>
<code>get()</code>	جلب سجل واحد فقط	<code>Student.objects.get(id=1)</code>
<code>order_by()</code>	ترتيب النتائج	<code>Student.objects.order_by("-age")</code>
<code>values()</code>	جلب حقول محددة كـ dict	<code>Student.objects.values("first_name", "age")</code>
<code>values_list()</code>	جلب حقول كـ tuple	<code>Student.objects.values_list("first_name", "age")</code>
<code>aggregate()</code>	عمليات إحصائية على الكل	<code>Student.objects.aggregate(Avg("age"))</code>
<code>annotate()</code>	إضافة عمود محسوب	<code>Stage.objects.annotate(num=Count("student"))</code>
<code>count()</code>	عدد السجلات	<code>Student.objects.count()</code>
<code>first() / last()</code>	أول / آخر سجل	<code>Student.objects.first()</code>
<code>exists()</code>	هل يوجد سجل؟	<code>Student.objects.filter(age=20).exists()</code>
<code>distinct()</code>	جلب القيم المميزة	<code>Student.objects.values("stage").distinct()</code>
<code>create()</code>	إنشاء سجل جديد	<code>Student.objects.create(first_name="Osama")</code>
<code>update()</code>	تحديث مجموعة سجلات	<code>Student.objects.filter(stage="first").update(stage="second")</code>
<code>delete()</code>	حذف مجموعة سجلات	<code>Student.objects.filter(age__lt=18).delete()</code>
<code>get_or_create()</code>	جلب أو إنشاء سجل	<code>Student.objects.get_or_create(first_name="Osama")</code>
<code>bulk_create()</code>	إنشاء عدة سجلات دفعـة واحدة	<code>Student.objects.bulk_create([Student(first_name="Osama", age=20), Student(first_name="Osama", age=21)])</code>
<code>in_bulk()</code>	جلب عدة سجلات كـقاموس	<code>Student.objects.in_bulk([1,2,3])</code>
<code>update_or_create()</code>	تحديث أو إنشاء	<code>Student.objects.update_or_create(first_name="Osama", defaults={"age":22})</code>
<code>order_by("?)</code>	ترتيب عشوائي	<code>Student.objects.order_by("?)</code>
<code>defer()</code>	استبعـاد حقول	<code>Student.objects.defer("age")</code>

النوع	الوصف	المثال
<b>prefetch_related()</b>	جلب العلاقات ManyToMany باستعلام منفصل وتحسين الأداء	<code>Student.objects.prefetch_related("courses")</code>
<b>select_for_update()</b>	قفل الصورف أثناء الترانزكشن	<code>Student.objects.select_for_update().get(id=1)</code>
<b>raw()</b>	استعلام SQL مباشر	<code>Student.objects.raw("SELECT * FROM student_student")</code>
<b>using()</b>	تحديد قاعدة بيانات معينة	<code>Student.objects.using("default").all()</code>
<b>iterator()</b>	استدعاء النتائج واحد واحد (لذاكرة أقل)	<code>for s in Student.objects.iterator(): ...</code>
<b>select_related()</b>	جلب العلاقات ForeignKey/OneToOne بنفس الاستعلام	<code>Student.objects.select_related("stage")</code>
<b>only()</b>	جلب حقول محددة	<code>Student.objects.only("first_name")</code>

## -2 الـ **Django** في **Form**

هناك طريقتان رئيسيتان :

### -1 العاديّة (**مكتوبة يدوياً بالبايثون**) :

- هنا نكتب كل الحقول يدوياً.
- نقدر نضيف تخصيص (**Widgets**) عشان نضيف CSS مباشرة من البايثون .

### -2 مربوط مباشرة بالمدل ( **ModelForm** ) :

- يربط الفورم بالـ **model** مباشرة .
- أقل تكرار كود، لو تغير الـ **model** الفورم يتغير مباشرة .
- نقدر نضيف Widgets عشان نتحكم بالشكل والتنسيق زي الطريقة الأولى .

## 3- التطبيق باستخدام طريقة **ModelForm**

### -4 ملف **models.py**

```
from django.db import models

class Student(models.Model):
    first_name = models.CharField(max_length=100, verbose_name="الاسم الأول")
    last_name = models.CharField(max_length=100, verbose_name="الاسم الأخير")
    age = models.IntegerField(verbose_name="العمر")
    stage = models.CharField(
        max_length=20,
        choices=[("first", "الأولى"), ("second", "الثانية"), ("third", "الثالثة")],
        verbose_name="المرحلة"
    )

    def __str__(self):
        return f"{self.first_name} {self.last_name}"
```

### -5 ملف **forms.py**

```
from django import forms
from .models import Student

class StudentForm(forms.ModelForm):
```

```

class Meta:
    model = Student
    fields = ['first_name', 'last_name', 'age', 'stage']
    widgets = {
        'first_name': forms.TextInput(attrs={'class': 'form-control',
'placeholder': 'أدخل الاسم الأول '}),
        'last_name': forms.TextInput(attrs={'class': 'form-control',
'placeholder': 'أدخل الاسم الأخير '}),
        'age': forms.NumberInput(attrs={'class': 'form-control',
'placeholder': 'أدخل العمر '}),
        'stage': forms.Select(attrs={'class': 'form-select'}),
    }

```

## : views.py -6

```

from django.shortcuts import render, redirect
from .forms import StudentForm
from .models import Student

def student_list(request):
    students = Student.objects.all()
    return render(request, "list.html", {"students": students})

def add_student(request):
    if request.method == "POST":
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect("student_list")
    else:
        form = StudentForm()
    return render(request, "add.html", {"form": form})

```

## : urls.py -7

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.student_list, name="student_list"),
    path('add/', views.add_student, name="add_student"),
]

```

## 8 - ملف add.html

```
{% extends "base.html" %}

{% block content %}
<div class="container mt-5">
    <h2 class="mb-4">إضافة طالب جديد</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit" class="btn btn-success">حفظ</button>
        <a href="{% url 'student_list' %}" class="btn btn-secondary">رجوع</a>
    </form>
</div>
{% endblock %}
```

## 9 - ملف base.html

```
10-<!DOCTYPE html>
11-<html lang="ar" dir="rtl">
12-<head>
13-    <meta charset="UTF-8">
14-    <title>{{ block title }} نظام إدارة المدرسة</title>
15-    <!-- Bootstrap -->
16-    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
17-</head>
18-<body>
19-
20-<!-- نافبار -->
21-<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
22-    <div class="container-fluid">
23-        <a class="navbar-brand" href="/">نظام المدرسة</a>
24-        <div class="collapse navbar-collapse">
25-            <ul class="navbar-nav me-auto">
26-                <li class="nav-item"><a class="nav-link" href="{% url 'student_list' %}">الطلاب</a></li>
27-                <li class="nav-item"><a class="nav-link" href="{% url 'add_student' %}">إضافة طالب</a></li>
28-            </ul>
29-        </div>
30-    </div>
31-</nav>
32-
33-<!-- مكان المحتوى -->
```

```

34-<div class="container mt-4">
35-     {% block content %}{% endblock %}
36-</div>
37-
38-<!-- Footer --&gt;
39-&lt;footer class="bg-dark text-white text-center p-3 mt-5"&gt;
40-     &lt;p&gt;جميع الحقوق محفوظة © 2025&lt;/p&gt;
41-&lt;/footer&gt;
42-
43-<!-- Bootstrap JS --&gt;
44-&lt;script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
45-&lt;/body&gt;
46-&lt;/html&gt;
47-</pre>

```

: list.html ملـف -48

```

{% extends "base.html" %}

{% block content %}
<div class="container mt-5">
    <h2 class="mb-4">قائمة الطلاب</h2>
    <a href="{% url 'add_student' %}" class="btn btn-primary mb-3">+ إضافة طالب</a>
    <table class="table table-striped">
        <thead>
            <tr>
                <th>الاسم الأول</th>
                <th>الاسم الأخير</th>
                <th>العمر</th>
                <th>المرحلة</th>
            </tr>
        </thead>
        <tbody>
            {% for student in students %}
            <tr>
                <td>{{ student.first_name }}</td>
                <td>{{ student.last_name }}</td>
                <td>{{ student.age }}</td>
                <td>{{ student.get_stage_display }}</td>
            </tr>
            {% empty %}
            <tr>
                <td colspan="4" class="text-center">لا يوجد طلاب مسجلين</td>
            </tr>
            {% endfor %}
        </tbody>
```

```
</table>
</div>
{% endblock %}
```

## - صور التنفيذ :

## قائمة الطلاب

+ إضافة طالب

المرحلة	العمر	الاسم الأخير	الاسم الأول
الثالثة	23	الجibli	اسامة سعيد
الثالثة	22	فمحان	ايمن محمد
الثالثة	21	الحضرمي	ضياء فضل
الثالثة	21	العمري	طارق فضل
الثالثة	20	العمري	حازم هزام
الثالثة	22	القواس	علي محمد
الثالثة	22	العتابي	عبدالله حسان

جميع الحقوق محفوظة © 2025

## إضافة طالب جديد

الاسم الأول:

أدخل الاسم الأول

الاسم الأخير:

أدخل الاسم الأخير

العمر:

أدخل العمر

المرحلة:

-----

رجوع حفظ

جميع الحقوق محفوظة © 2025

## إضافة طالب جديد

الاسم الأول:  
حسام طلعتالاسم الأخير:  
الحدادالعمر:  
20المرحلة:  
الثالثة 

## قائمة الطلاب

المرحلة	الاسم الأول	الاسم الأخير	العمر
الثالثة	اسامة سعيد	الجلبي	23
الثالثة	ايمن محمد	قمحان	22
الثالثة	ضياء فضل	الحضرمي	21
الثالثة	طارق فضل	العمري	21
الثالثة	حازم هزام	العمري	20
الثالثة	علي محمد	القواس	22
الثالثة	عبد الله حسان	العتابي	22
الثالثة	حسام طلعت	الحداد	20

## - مقارنة بين الأنواع الثلاثة :

وجه المقارنة	ModelForm	عادي Form	عادي HTML
سهولة الكتابة	الأسهل	متوسط	صعب (كل شيء يدوي)
التحقق من القيم (Validation)	تلقائي	لازم تكتب حقول + يدوي validation	لازم تكتب كل بنفسك validation
التحديث مع المودل	يتحدث تلقائياً	لازم تعدل يدوياً	لازم تعدل يدوياً
التنسيق (CSS)	عبر widgets	عبر widgets	عادي HTML+CSS
المرونة	متوسط	عالي (تحكم كامل)	عالي جدًا (لكن الشغل كثير)