

# Django Student Management Mini Project

---

## 1. Agenda & Steps to Create Django Project

Objective: Create a Django app to display student data from the database using ORM queries, Admin Panel, Views, and HTML templates.

### Steps:

Step	Command / Action	Description
1	<code>django-admin startproject studentproject</code>	Create main project
2	<code>cd studentproject</code>	Navigate into project folder
3	<code>python manage.py startapp testapp</code>	Create app named testapp
4	Add 'testapp' in <code>INSTALLED_APPS</code>	Enable app in settings.py
5	Define Student model in <code>models.py</code>	Define DB structure
6	<code>python manage.py makemigrations</code>	Generate migration file
7	<code>python manage.py migrate</code>	Create tables in DB
8	<code>python manage.py createsuperuser</code>	Login to admin panel
9	Register model in <code>admin.py</code>	Manage student data from admin
10	Create view in <code>views.py</code>	Fetch and send data to template
11	Add app & project-level URLs	Route to views
12	Create template <code>std.html</code>	Show student data

13	python manage.py runserver	Start development server
----	-------------------------------	--------------------------

## 2. Views (HTML + JSON Response)

*testapp/views.py*

---

```

from django.shortcuts import render
from django.http import JsonResponse
from testapp.models import Student

def student_view(request):
    student_list = Student.objects.all().order_by('-marks')
    return render(request, 'testapp/std.html', {'student_list': student_list})

def student_json_view(request):
    data = list(Student.objects.values())
    return JsonResponse(data, safe=False)

```

## 3. URLs

*studentproject/urls.py*

---

```

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('testapp.urls')),
]

```

---

```
from django.urls import path
from . import views

urlpatterns = [
    path('students/', views.student_view, name='student-view'),
    path('api/students/', views.student_json_view, name='student-json'),
]
```

#### 4. HTML Template (std.html)

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
    <title>Student Info</title>
    <link rel="stylesheet" href="{% static 'css/std.css' %}">
</head>
<body>
    <h1>Student Information</h1>
    {% if student_list %}
        {% for student in student_list %}
            <h2>{{ student.name }}</h2>
            <ul>
                <li>Roll No: {{ student.rollno }}</li>
                <li>DOB: {{ student.dob }}</li>
                <li>Marks: {{ student.marks }}</li>
                <li>Email: {{ student.email }}</li>
                <li>Phone: {{ student.phonenumber }}</li>
                <li>Address: {{ student.address }}</li>
            </ul>
            <hr>
        {% endfor %}
    {% else %}
        <p>No student data found.</p>
    {% endif %}
</body>
</html>
```

## 5. ORM Commands in Tabular Format

Command	Description	Example
.all()	Fetch all records	Student.objects.all()
.filter()	Apply condition	Student.objects.filter(marks__lt=35)
.exclude()	Exclude matching	Student.objects.exclude(name='Ali')
.order_by()	Sort records	Student.objects.order_by('-marks')
.values()	Return dict-like data	Student.objects.values()
.get()	Return single object	Student.objects.get(rollno=101)
.count()	Total entries	Student.objects.count()
.first() / .last()	First / Last entry	Student.objects.first()

## 6. FAQs

Question	Answer
What is ORM in Django?	ORM (Object Relational Mapper) lets you interact with DB using Python code instead of SQL.
What is makemigrations?	It creates migration files to prepare DB structure from models.
What is migrate?	It applies migration files to the actual database.
How do you show data in admin panel?	Register your model using admin.site.register().
How to return JSON data?	Use JsonResponse() with values() or values_list().

## 7. Tricky Concept

*ORM works like a translator between Python code and SQL queries.*

Example: Student.objects.filter(marks\_\_lt=35) → SQL: SELECT \* FROM student WHERE marks < 35;