FSD - Lab Component– 4

1. For students enrolment developed in Module 2, create a generic class view which displays list of students and detailview that displays student details for any selected student in the list

models.py

```
from django.db import models

# Create your models here.
class Course(models.Model):
    course_code = models.CharField(max_length=40)
    course_name = models.CharField(max_length=100)
    course_credits = models.IntegerField()

    def __str__(self):
        return self.course_name

class Student(models.Model):
    student_usn = models.CharField(max_length=20)
    student_name = models.CharField(max_length=100)
    student_sem = models.IntegerField()
    enrolment = models.ManyToManyField(Course)

    def __str__(self):
        return self.student_name
```

admin.py

```
from django.contrib import admin

from ap1.models import Course, Student

# Register your models here.
admin.site.register(Course)
admin.site.register(Student)
```

urls.py

```
from django.contrib import admin
from django.urls import path
from django.views.generic import ListView, DetailView
from ap1.models import Student

student_list_info = {
    "model": Student,
    "context_object_name": "student_list",  # Ensure this matches the template variable
    "template_name": "student_list.html"
}

student_detail_info = {
    "model": Student,
    "context_object_name": "student",  # Ensure this matches the template variable
    "template_name": "student_detail.html"
}

urlpatterns = [
    path('admin/', admin.site.urls),
    path('student_list/', ListView.as_view(**student_list_info), name='student_list'),
    path('student_detail/<int:pk>/', DetailView.as_view(**student_detail_info),
name='student_detail'),
]
```

In templates folder create the following two html files

student_list.html

```
<html>
<body>
{% if student_list %}
    <table border>
        <tr>
            <th>USN</th>
            <th>Courses Enrolled</th>
        </tr>
        {% for student in student_list %}
            <tr>
                <td><a href="/student_detail/{{student.pk}}">
                    {{student.student_usn }}</a></td>
                <td>
                    {% for course in student.enrolment.all %}
                        <span>{{ course.course_name }}</span>
                    {% endfor %}
                </td>
                </tr>
            {% endfor %}
        </table>
    {% else %}
        <h1>No Students Enrolled</h1>
    {% endif %}
```

student_detail.html

```
    <h1>Student Name: {{ student.student_name }}</h1>
    <h1>Student USN: {{ student.student_usn }}</h1>
    <h1>Student Sem: {{ student.student_sem }}</h1>
```

Output:

1. After Creating Files:

- models.py: Define Course and Student models.
- urls.py: Set up URL patterns for student_list and student_detail.
- student_list.html: Create template for listing students.
- student_detail.html: Create template for student details.

2. Add ap1 to INSTALLED_APPS:

- Edit settings.py and add 'ap1' to the INSTALLED_APPS list.

3. Run Migration Commands:

- Run python manage.py makemigrations ap1
- Run python manage.py migrate

4. Create Superuser:

- Run python manage.py createsuperuser

5. Run Development Server:

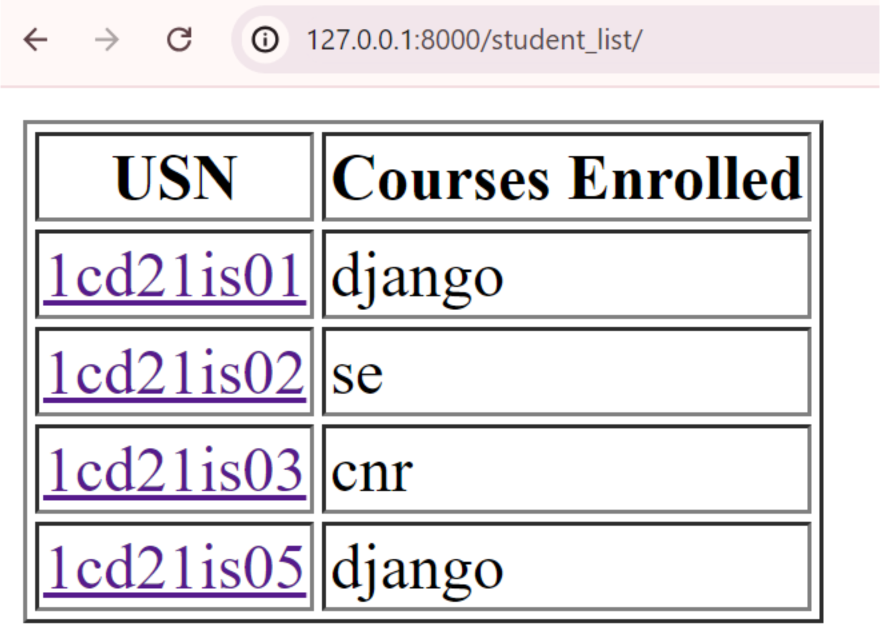
- Run python manage.py runserver

6. Enter Data through Admin Interface:

- Log in to the admin interface at http://127.0.0.1:8000/admin/
- Add Course and Student entries.

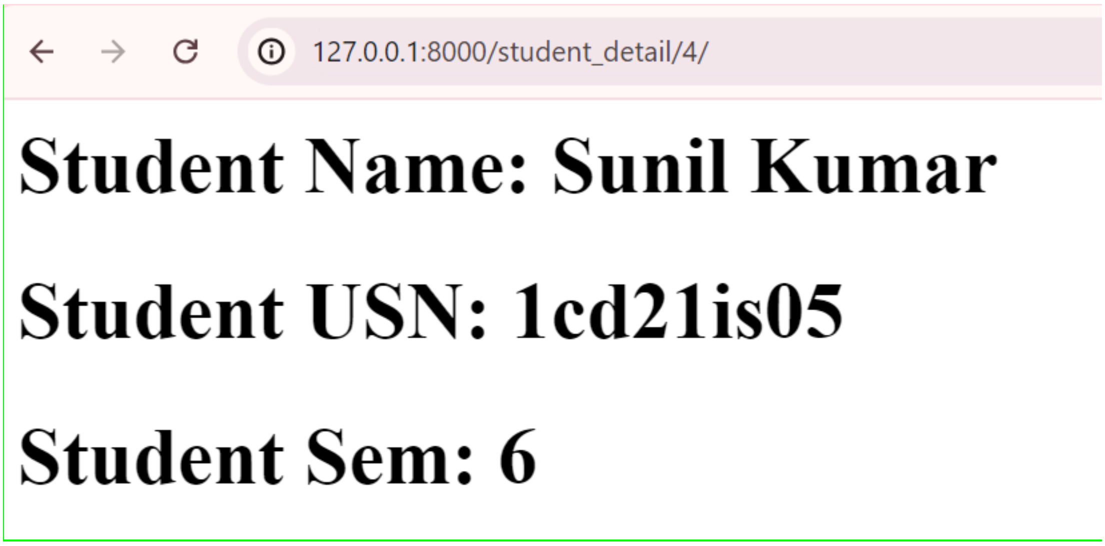
7. Access URLs:

- Visit http://127.0.0.1:8000/student_list/ to see the student list.
- Click on a student’s USN to view their details at http://127.0.0.1:8000/student_detail/<int:pk>/.



127.0.0.1:8000/student_list/

USN	Courses Enrolled
1cd21is01	django
1cd21is02	se
1cd21is03	cnr
1cd21is05	django



127.0.0.1:8000/student_detail/4/

Student Name: Sunil Kumar

Student USN: 1cd21is05

Student Sem: 6

2. Develop example Django app that performs CSV and PDF generation for any models created in previous laboratory component.

models.py

```
from django.db import models

class Course(models.Model):
    course_name = models.CharField(max_length=100)
    course_code = models.CharField(max_length=10)
    course_credits = models.IntegerField()

    def __str__(self):
        return self.course_name
```

admin.py

```
from django.contrib import admin

from ap1.models import Course

# Register your models here.
admin.site.register(Course)
```

views.py

```
from django.http import HttpResponse
from .models import Course
import csv
from reportlab.pdfgen import canvas

def construct_csv_from_model(request):
    courses = Course.objects.all()
    response = HttpResponse(content_type="text/csv")
    response['Content-Disposition'] = 'attachment; filename="courses_data.csv"'
    writer = csv.writer(response)
    writer.writerow(["Course Name", "Course Code", "Credits"])
    for course in courses:
        writer.writerow([course.course_name, course.course_code, course.course_credits])
    return response

def construct_pdf_from_model(request):
    courses = Course.objects.all()
    response = HttpResponse(content_type="application/pdf")
    response['Content-Disposition'] = 'attachment; filename="courses_data.pdf"'
    c = canvas.Canvas(response)
    c.drawString(70, 720, "Course Name")
    c.drawString(170, 720, "Course Code")
    c.drawString(270, 720, "Credits")
    y = 660
    for course in courses:
        c.drawString(70, y, course.course_name)
        c.drawString(170, y, course.course_code)
        c.drawString(270, y, str(course.course_credits))
        y = y - 60
    c.showPage()
    c.save()
    return response
```

urls.py

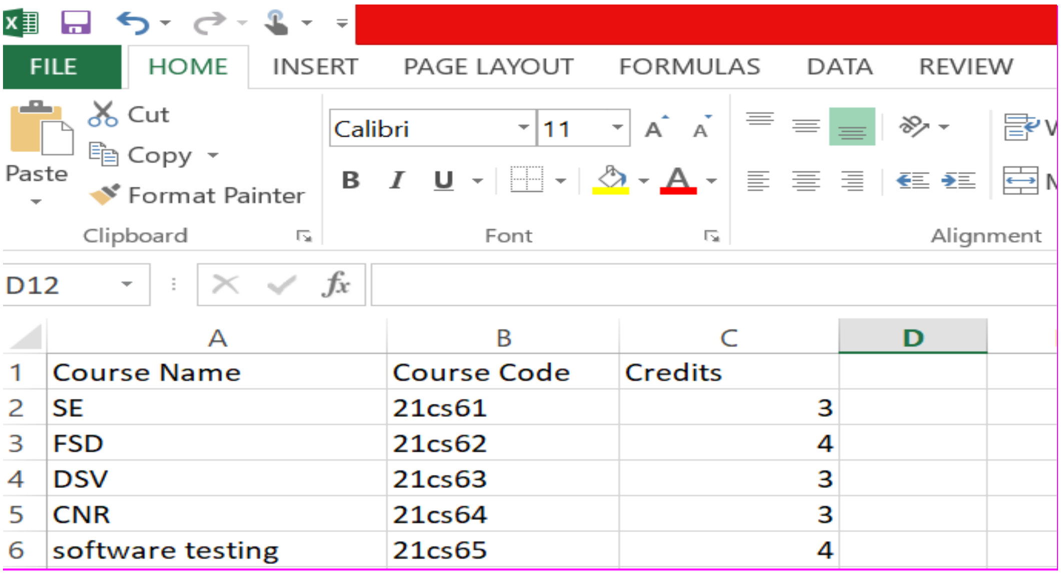
```
from django.contrib import admin
from django.urls import path
from ap1.views import construct_csv_from_model, construct_pdf_from_model

urlpatterns = [
    path('admin/', admin.site.urls),
    path('courses/csv/', construct_csv_from_model, name='course_csv'),
    path('courses/pdf/', construct_pdf_from_model, name='course_pdf'),
]
```

Output :

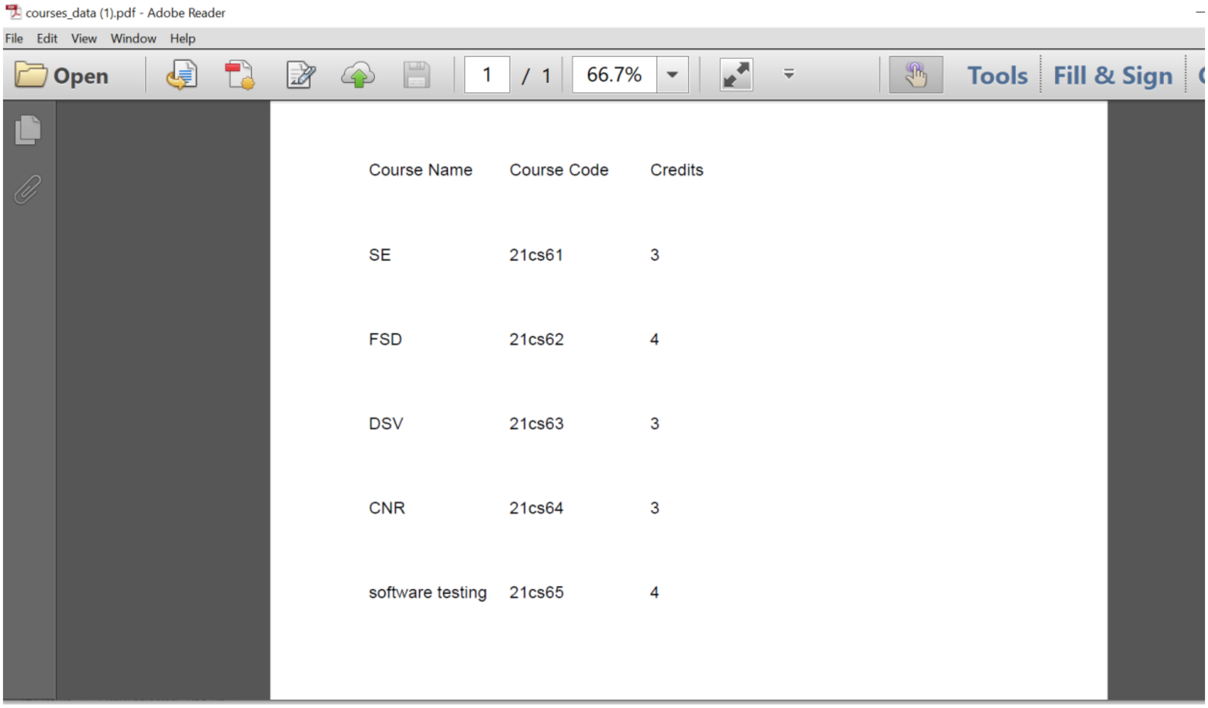
1. Create folder , activate virtual environment
2. Create Django project and application
3. Create models, views, urls and register model in admin
4. Run Migrations
5. Create super user
6. Login to Admin Interface and Enter course data in Admin Interface
7. Generate and download CSV and PDF files by accessing urls

Downloaded CSV file :



	A	B	C	D
1	Course Name	Course Code	Credits	
2	SE	21cs61	3	
3	FSD	21cs62	4	
4	DSV	21cs63	3	
5	CNR	21cs64	3	
6	software testing	21cs65	4	

Downloaded PDF file:


courses_data (1).pdf - Adobe Reader
File Edit View Window Help
Open
1 / 1
66.7%
Tools
Fill & Sign

Course Name	Course Code	Credits
SE	21cs61	3
FSD	21cs62	4
DSV	21cs63	3
CNR	21cs64	3
software testing	21cs65	4