# Module-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):

    name = models.CharField(max_length=100)

    description = models.TextField()

    def str (self):

        return self.name

class Student(models.Model):

    name = models.CharField(max_length=100)

    email = models.EmailField()

    courses = models.ManyToManyField(Course, related_name='students')

    def str (self):

        return self.name
```

## migrate and enter data in both tables using admin interface and register students with some courses:

## views.py:

```
class StudentListView(ListView):

    model = Student

    template_name = 'student_list.html'  # Specify your template name

    context_object_name = 'students'


class StudentDetailView(DetailView):

    model = Student

    template_name = 'student_detail.html'  # Specify your template name

    context_object_name = 'student'
```

## templates:

## student_list.html

```
<!DOCTYPE html>

<html>

<head>

    <title>Student List</title>

</head>

<body>

    <h1>Student List from Student Model</h1>

    <ul>

        {% for student in students %}

            <li>

                <a href="{% url 'student_detail' student.pk %}">{{ student.name }}</a>

            </li>
        {% endfor %}
    </ul>
</body>
</html>
```

# student_detail.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Detail</title>
</head>
<body>
    <h1>========================={{ student.name
}}=============================</h1>
    <p>Email: {{ student.email }}</p>

    <h2>================Registered Course====================</h2>
    <ul>
        {% for course in student.courses.all %}
            <li>{{ course.name }}</li><br>
            <b> Description : {{ course.description }}</b>

        {% endfor %}
    </ul>
    <a href="{% url 'student_list' %}">Back to Student List</a>
</body>
</html>
```

## urls.py:

```
urlpatterns =[
    path('admin/', admin.site.urls),
    path('student_list/', views.StudentListView.as_view(), name='student_list'),
    path('student/<int:pk>/',
views.StudentDetailView.as_view(),name='student_detail'),
    ]
```

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

## models.py

## same as previous

## views.py

```
import csv
from django.http import HttpResponse


from django.shortcuts import render
from Module4_genericViews.models import Course, Student
from reportlab.pdfgen import canvas
def generate_csv_response(request):
    queryset = Student.objects.all() # Get data from your models
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = f'attachment; filename="student_data.csv"'
    writer = csv.writer(response)
    # Write header row based on model fields using List comprehension
    writer.writerow([field.name for field in queryset.model._meta.fields])
    # Write data rows
    for obj in queryset:
        writer.writerow([getattr(obj, field.name) for field in
queryset.model._meta.fields])
    return response




def generate_pdf_response(request):
    queryset = Student.objects.all()
    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = 'attachment;
filename="student_pdfdata.pdf"'

    # Create PDF document
    pdf = canvas.Canvas(response)
    y = 800 # Initial y position for writing text
    pdf.setFont("Helvetica-Bold", 12)
    pdf.drawString(100, y, "Student Data")

    # Write data
    pdf.setFont("Helvetica", 10)
    y -= 30 # Move down for data rows

    for obj in queryset:
        data = f"Name: {obj.name}, Email: {obj.email}" # Customize based on your model fields

        if y < 50:  # If we're near the bottom of the page, start a new page
            pdf.showPage()
            y = 800
            pdf.setFont("Helvetica", 10)

        pdf.drawString(100, y, data)
        y -= 15  # Move down for next row

    pdf.showPage()
    pdf.save()
    return response
```

## urls.py

```
urlpatterns =[
      path('admin/', admin.site.urls),
      path('generatecsv/', views.generate_csv_response),
      path('generatepdf/', views.generate_pdf_response),
    ]
```

# Module-5

## 1. Develop a registration page for student enrolment as done in Module 2 but without page refresh using AJAX.

### Step-1 Create models……………………………….

### models.py

```
from django.db import models

class Course(models.Model):

    name = models.CharField(max_length=100)

    description = models.TextField()

    def __str__(self):

        return self.name


class Student(models.Model):

    name = models.CharField(max_length=100)

    email = models.EmailField()

    course = models.ForeignKey(Course, on_delete=models.CASCADE)

    def __str__(self):

        return self.name
```

**Note: Give migration commands and enter data in course table**

## Step-2 Create form in model.py itself

```
from django import forms
class StudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = ['name', 'email', 'course']
```

## views.py

```
from django.shortcuts import render
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from .models import Student, Course, StudentForm

def register_student(request):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return JsonResponse({'success': True})
        else:
```

```python
            return JsonResponse({'success': False, 'errors': form.errors})

    else:

        courses = Course.objects.all()

        form= StudentForm()

        return render(request, 'student_registration.html', {'courses':
courses,'form': form})


def student_list(request):

    students = Student.objects.all()

    return render(request, 'studentlist.html', {'students': students})
```

## student_registration.html

```html
<!-- student_registration.html -->

<!DOCTYPE html>

<html>

<head>

    <title>Student Registration</title>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js">

</script>

</head>

<body>

    <h1>Register Student</h1>
    <form id="studentForm">
        {% csrf_token %}
       {{form.as_p}}
        <button type="submit">Register</button>
    </form>
<div id="message"></div>
<!-- JQuery Code  -->
       <script>
        $(document).ready(function(){
            $('#studentForm').on('submit', function(event){
                event.preventDefault();
                $.ajax({
                    url: '{% url "student_registration" %}',
                    method: 'POST',
                    data: $(this).serialize(),
                    success: function(response){
                        if(response.success){
                            $('#message').html('<p>Student registered
successfully! <a href="{% url "studentlist" %}">View students</a></p>');
                            $('#studentForm')[0].reset();
                        } else {
                            let errors = '<ul>';
                            for (let field in response.errors) {
                                response.errors[field].forEach(function(error) {
                                    errors += '<li>' + error + '</li>';
                                });
                            }
                            errors += '</ul>';
                            $('#message').html(errors);
                        }
                    },
                    error: function(xhr, status, error) {
                        console.error("AJAX request failed:", error);
                        $('#message').html('<p>An error occurred while processing your request.</p>');
                    }
                });
            });
        });
    </script>
</body>
</html>
```

## studentlist.html

```
<!--student_list.html -->
<!DOCTYPE html>
```

```
<html>
<head>
    <title>Student List</title>
</head>
<body>
    <h1>Registered Students</h1>
    <ul>
        {% for student in students %}
            <li>{{ student.name }} - {{ student.email }} - {{ student.course.name
}}</li>
        {% endfor %}
    </ul>
    <a href="{% url 'student_registration' %}">Register a new student</a>
</body>
</html>
```

urls.py

```
  path('registerajax/', register_student, name='student_registration'),
    path('studentsajax/', student_list, name='studentlist'),
```

## 2. Develop a search application in Django using AJAX that displays courses enrolled by a student being searched

### models.py

```
from django.db import models
```

```python
class Course(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()

    def __str__(self):
        return self.name


class Student(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField()
    courses = models.ManyToManyField(Course)

    def __str__(self):
        return self.name
# forms.py
from django import forms
class StudentSearchForm(forms.Form):
    student_name = forms.CharField(label='Student Name', max_length=100)
```

## views.py

```python
from django.shortcuts import render
```

```python
# Create your views here.
from django.shortcuts import render
from django.http import JsonResponse
from .models import Student, StudentSearchForm


def search_form(request):
    form = StudentSearchForm()
    return render(request, 'search_form.html', {'form': form})


def search_courses(request):
    if request.method == 'GET' and 'student_name' in request.GET:
        student_name = request.GET['student_name']
        student = Student.objects.filter(name__icontains=student_name).first()
        if student:
            courses = student.courses.all()
            course_list = [{'name': course.name, 'description': course.description} for course in courses]
            return JsonResponse({'success': True, 'courses': course_list})
        else:
            return JsonResponse({'success': False, 'error': 'Student not found.'})
    return JsonResponse({'success': False, 'error': 'Invalid request.'})
```

## search_form.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Search for Courses by Student Name</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body>
    <h1>Search for Courses by Student Name</h1>
    <form id="searchForm" method="GET">
        {{form.as_p}}
        <button type="submit">Search</button>
    </form>
    <div id="courseList"></div>
    <script>
        $(document).ready(function () {
            $('#searchForm').on('submit', function(event) {
                event.preventDefault();
                var formData = $(this).serialize();
                $.ajax({
                    url: '{% url "search_courses" %}',
                    type: 'GET',
                    data: formData,
                    success: function (response) {
                        if (response.success) {
                            var courses = response.courses;
                            var courseList = '<ul>';
                            for (var i = 0; i < courses.length; i++) {
                                courseList += '<li>' + courses[i].name + ': ' + courses[i].description + '</li>';
                            }
                            courseList += '</ul>';
                            $('#courseList').html(courseList);
                        } else {
                            $('#courseList').html('<p>' + response.error + '</p>');
                        }
                    },
                    error: function (xhr, errmsg, err) {
                        console.log(xhr.status + ': ' + xhr.responseText);
                        $('#courseList').html('<p>An error occurred while processing your request.</p>');
                    }
                });
            });
        });
```

```
    </script>

</body>

</html>
```

## urls.py

```
 path('', search_form, name='search_form'),

    path('search-courses/', search_courses, name='search_courses'),]
```