

## Module-1

### 1. Develop a Django app that displays current date and time in server

#### views.py

```
from django.http import HttpResponse
from django.shortcuts import render
import datetime

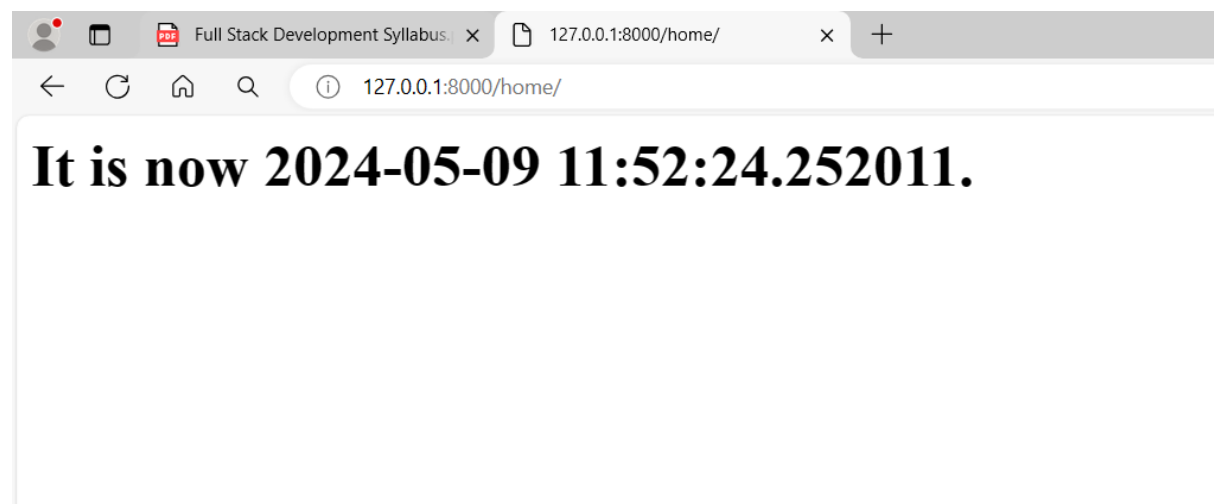
def current_date_time(request):
    dt = datetime.datetime.now()
    resp = "<h1>It is now %s.</h1>" % (dt)
    return HttpResponse(resp)
```

#### urls.py

```
from django.contrib import admin
from django.urls import path, re_path

from ap1.views import current_date_time

urlpatterns = [
    path('admin/', admin.site.urls),
    path('home/', current_date_time),
]
```



## 2. Develop a Django app that displays date and time four hours ahead and four hours before as an offset of current date and time in server.

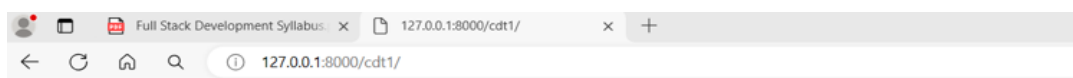
views.py

```
from django.http import HttpResponse
from django.shortcuts import render
# Create your views here.
import datetime
def current_datetime(request):
    now = datetime.datetime.now()
    html = "<html><body><h1>It is now %s.</h1></body></html>" % now
    return HttpResponse(html)
def four_hours_ahead(request):
    dt = datetime.datetime.now() + datetime.timedelta(hours=4)
    html = "<html><body><h1>After 4hour(s), it will be %s.</h1>"% (dt,)
    return HttpResponse(html)
def four_hours_before(request):
    dt = datetime.datetime.now() + datetime.timedelta(hours=-4)
    html = "<html><body><h1>Before 4 hour(s), it was %s.</h1>"% (dt,)
    return HttpResponse(html)
```

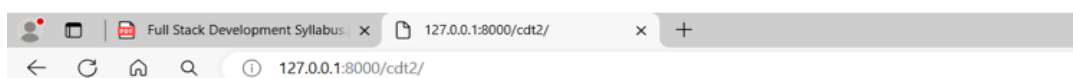
urls.py

```
from django.contrib import admin
from django.urls import path, re_path

from labprg11.views import current_datetime, four_hours_ahead,
four_hours_before
urlpatterns = [
    path('admin/', admin.site.urls),
    path('cdt/',current_datetime),
    path('cdt1/',four_hours_ahead),
    path('cdt2/',four_hours_before),
]
```



**After 4hour(s), it will be 2024-05-09 15:59:00.515271.**



**Before 4 hour(s), it was 2024-05-09 07:59:40.678952.**

## Module-2

### 3. Develop a simple Django app that displays an unordered list of fruits and ordered list of selected students for an event

#### Views.py

```
from datetime import date
from django.shortcuts import render
from django.http import HttpResponse
from django.template import Context, Template

# Create your views here.
def showlist(request):
    fruits=["Mango", "Apple", "Bananan", "Jackfruits"]
    student_names=["Tony", "Mony", "Sony", "Bob"]
    return
render(request, 'showlist.html', {"fruits":fruits, "student_names":student_names})
```

#### Template – showlist.html

```
<html>
  <style type="text/css">
    #i1 {background-color: lightgreen;color:brown;display:table}
    #i2 {background-color: black;color:yellow}
  </style>
  <body>
    <h1 id="i1">unordered list of fruits</h1>
    <ul>
      {% for fruit in fruits %}
      <li>{{ fruit }}</li>
      {% endfor %}
    </ul>
    <h1 id="i2">ordered list of students</h1>
    <ol>
      {% for student in student_names %}
      <li>{{ student }}</li>
      {% endfor %}
    </ol>
  </body>
</html>
```

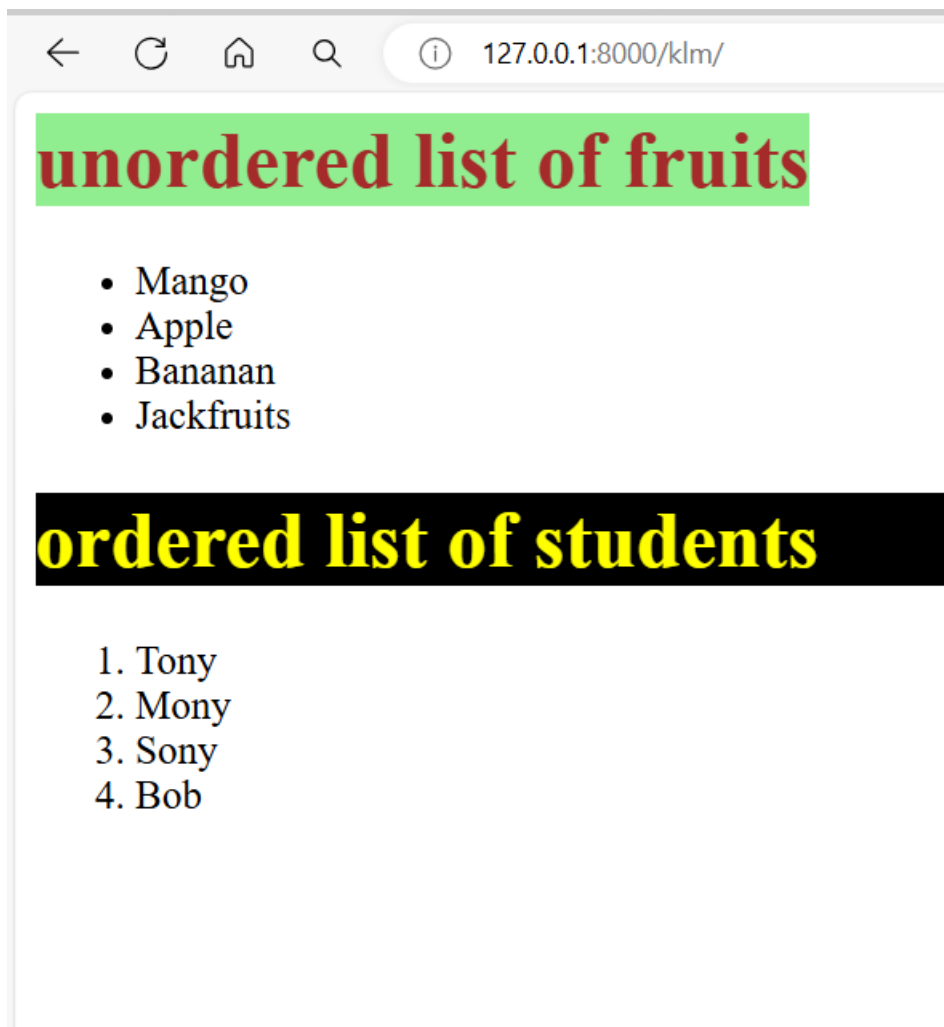
#### Urls.py

```
from django.contrib import admin
from django.urls import path, re_path

from labprg1.views import showlist

urlpatterns = [
    path('admin/', admin.site.urls),
    path('klm/', showlist),
]
```

## Output



4. **Develop a layout.html with a suitable header (containing navigation menu) and footer with copyright and developer information. Inherit this layout.html and create 3 additional pages: contact us, About Us and Home page of any website.**

## Views.py

```
from django.shortcuts import render

# Create your views here.
def home(request):
    return render(request, 'home.html')
def aboutus(request):
    return render(request, 'aboutus.html')
def contactus(request):
    return render(request, 'contactus.html')
```

## Templates – 1. Home.html

```
{% extends 'layout.html' %}
{% block title %}
Home
{% endblock %}
{% block content %}
<h2>This is the home page</h2>
{% endblock %}
```

## 2. aboutus.html

```
{% extends 'layout.html' %}
{% block title %}
About Us
{% endblock %}
{% block content %}
<h2>SJBIT ASSISTANT PROFESSOR</h2>
{% endblock %}
```

## 3. contactus.html

```
{% extends 'layout.html' %}
{% block title %}
Contact us
```

```
{% endblock %}
{% block content %}
<h2>Out phone: 7760965811 <br>
Address: KENGERI SJBIT</h2>
{% endblock %}
```

## layout.html

```
<html>
  <title>{% block title %} {% endblock %} </title>
  <style type="text/css">
    nav {background-color: lightblue;padding:10px}
  </style>
  <body>
    <nav>
      <a href="/home/">Home</a>|
      <a href="/aboutus/">About Us</a>|
      <a href="/contactus/">Contact Us</a>|
    </nav>
    <section>
      {% block content %}{% endblock %}
    </section>
    <footer>
      <hr>
      &copy; SJBIT, Developed by Ranjitha, Inc.
    </footer>
  </body>
</html>
```

## Urls.py

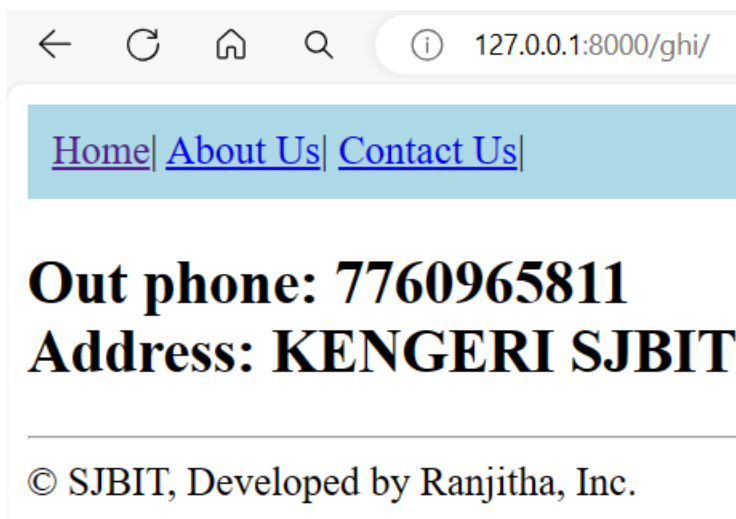
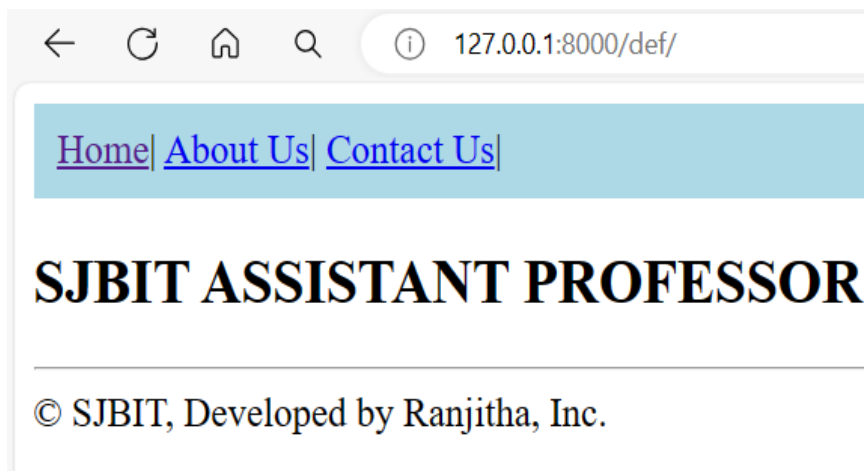
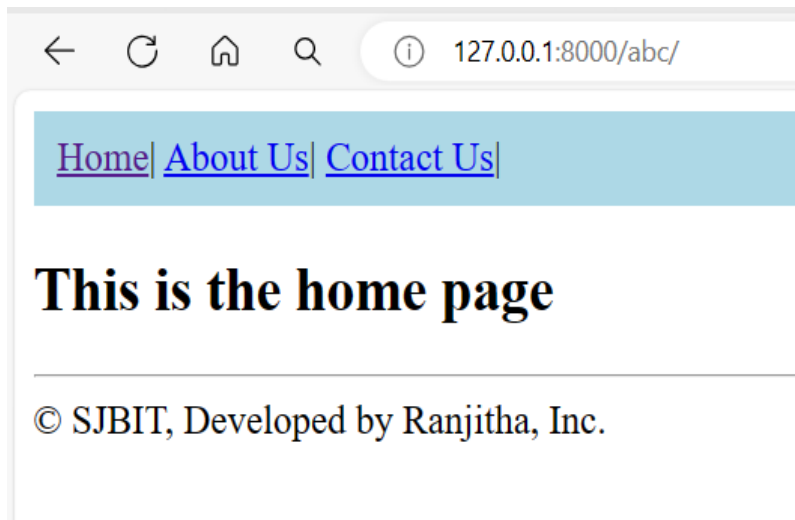
```
from django.contrib import admin
from django.urls import path, re_path

from labprg21.views import aboutus, contactus, home

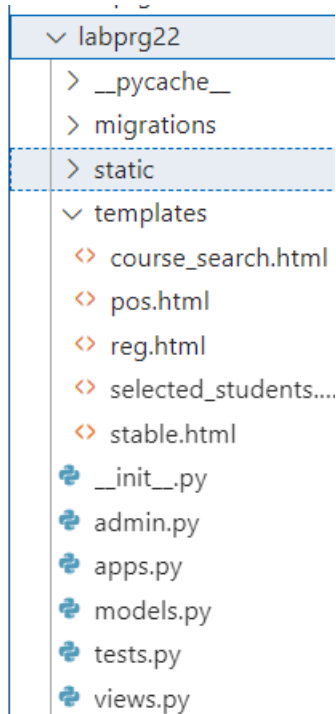
urlpatterns = [
    path('admin/', admin.site.urls),
    path('abc/',home),
    path('def/',aboutus),
    path('ghi/',contactus),

]
```

## Output



5. Develop a Django app that performs student registration to a course. It should also display list of students registered for any selected course. Create students and course as models with enrolment as ManyToMany field.



### Views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from labprg22.models import Course, Student
def reg(request):
    if request.method == "POST":
        sid=request.POST.get("sname")
        cid=request.POST.get("cname")
        student=Student.objects.get(id=sid)
        course=Course.objects.get(id=cid)
        res=student.enrolment.filter(id=cid)
        if res:
            return HttpResponseRedirect("<h1>Student already enrolled</h1>")
        student.enrolment.add(course)
        return HttpResponseRedirect("<h1>Student enrolled successfully</h1>")
    else:
        students=Student.objects.all()
        courses=Course.objects.all()
        return
render(request,"reg.html",{"students":students,"courses":courses})
def course_search(request):
    if request.method=="POST":
        cid=request.POST.get("cname")
```



```

        s=Student.objects.all()
        student_list=list()
        for student in s:
            if student.enrolment.filter(id=cid):
                student_list.append(student)
        if len(student_list)==0:
            return HttpResponse("<h1>No Students enrolled</h1>")
        return
    render(request,"selected_students.html",{"student_list":student_list})

    else:
        courses=Course.objects.all()
        return render(request,"course_search.html",{"courses":courses})

```

models.py

```

from django.db import models

# Create your models here.
class Course(models.Model):
    course_code=models.CharField(max_length=10)
    course_name=models.CharField(max_length=30)
    course_credits=models.IntegerField()

class Student(models.Model):
    student_usn=models.CharField(max_length=12)
    student_name=models.CharField(max_length=30)
    student_sem=models.IntegerField()
    enrolment=models.ManyToManyField(Course)

```

templates - course\_search.html, pos.html, reg.html, selected\_students.html

```

course_search.html
<html>
    <body>
        <form method="POST" action="">
            Courses
            {% csrf_token %}
            <select name="cname">
                {%for course in courses %}
                <option value="{{course.id}}">{{course.course_name}}</option>
                {% endfor %}
            </select>
            <input type="submit" value="Search">
        </form>
    </body>
</html>

```

reg.html

```
<html>
  <body>
    <form method="post" action="">
      {% csrf_token %}
      Student Name
      <select name="sname">
        {%for student in students %}
        <option value="{{student.id}}">{{student.student_name}}</option>
        {% endfor %}
      </select><br>
      Course Name
      <select name="cname">
        {%for course in courses %}
        <option value="{{course.id}}">{{course.course_name}}</option>
        {% endfor %}
      </select><br>
      <input type="submit" value="Enroll">
    </form>
  </body>
</html>
```

Selected\_students.html

```
<html>
  <body>
    <table border>
      <tr>
        <th>Student Name</th>
        <th>Student USN</th>
        <th>Sem</th>
      </tr>
      {% for student in student_list %}
      <tr>
        <td>{{student.student_name}}</td>
        <td>{{student.student_usn}}</td>
        <td>{{student.student_sem}}</td>
      </tr>
      {% endfor %}
    </table>
  </body>
</html>
```

```

Urls.py
from django.contrib import admin
from django.urls import path, re_path
from labprg22.views import reg, course_search
urlpatterns = [
    path('admin/', admin.site.urls),
    path('reg/', reg),
    path('course_search/', course_search)
]

```

#### admin.py

```

from django.contrib import admin
from labprg22.models import Student, Course
# Register your models here.
admin.site.register(Student)
admin.site.register(Course)

```

**In apps.py file check whether it is configure**

```

from django.apps import AppConfig

```

```

class Labprg22Config(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'labprg22'

```

**In setting file specify the filename**

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'crudoperation',
    'theoryexample1',
    'labprg22'
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'college',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',

```

```

        'PORT': '3306',
    }
}

```

In WAMP server

Localhost/phpMyAdmin

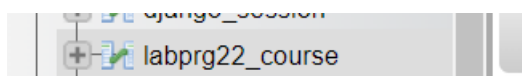
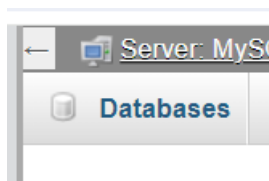
create a database - 'college'

Insert a row and fill the value to it.

Python manage.py makemigrations labprg22

Python manage.py migrate

Python manage.py runserver



Server: MySQL:3306 » Database: college » Table: labprg22\_course

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#)

✓ Showing rows 0 - 3 (4 total, Query took 0.0363 seconds.)

`SELECT * FROM `labprg22_course``


☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows:  Filter rows:  Sort by key:

Extra options

				id	course_code	course_name	course_credits
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	1	C001	MATHS	4
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	2	C002	JAVA	3
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	3	C003	PYTHON	3
<input type="checkbox"/>	<a href="#">Edit</a>	<a href="#">Copy</a>	<a href="#">Delete</a>	4	C004	FSD	4

Column	Type	Function	Null	Value
id	bigint	<input type="text"/>	<input type="text"/>	
course_code	varchar(10)	<input type="text"/>	<input type="text"/>	
course_name	varchar(30)	<input type="text"/>	<input type="text"/>	
course_credits	int	<input type="text"/>	<input type="text"/>	

 labprg22\_student

Server: MySQL:3306 » Database: college » Table: labprg22\_student

✓ Showing rows 0 - 2 (3 total, Query took 0.0070 seconds.)

`SELECT * FROM `labprg22_student``

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows:  Filter rows:  Sort by key:

Extra options

				id	student_usn	student_name	student_sem
<input type="checkbox"/>				1	U001	aba	3
<input type="checkbox"/>				2	U002	nam	3
<input type="checkbox"/>				3	U003	Ranjitha	6

☐ Check all With selected: Edit Copy Delete Export

python manage.py runserver

127.0.0.1:8000/reg/

Student Name

Course Name



127.0.0.1:8000/course\_search/

Student Name	Student USN	Sem
nam	U002	3

## Module-3

6. For student and course models created in Lab experiment for Module2, register admin interfaces, perform migrations and illustrate data entry through admin forms

### Views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from program3.models import Course, Student
def reg(request):
    if request.method == "POST":
        sid=request.POST.get("sname")
        cid=request.POST.get("cname")
        student=Student.objects.get(id=sid)
        course=Course.objects.get(id=cid)
        res=student.enrolment.filter(id=cid)
        if res:
            return HttpResponseRedirect("<h1>Student already enrolled</h1>")
        student.enrolment.add(course)
        return HttpResponseRedirect("<h1>Student enrolled successfully</h1>")
    else:
        students=Student.objects.all()
        courses=Course.objects.all()
        return
render(request,"reg.html",{"students":students,"courses":courses})
def course_search(request):
    if request.method=="POST":
        cid=request.POST.get("cname")
        s=Student.objects.all()
        student_list=list()
        for student in s:
            if student.enrolment.filter(id=cid):
                student_list.append(student)
        if len(student_list)==0:
            return HttpResponseRedirect("<h1>No Students enrolled</h1>")
        return
render(request,"selected_students.html",{"student_list":student_list})

    else:
        courses=Course.objects.all()
        return render(request,"course_search.html",{"courses":courses})
```

### admin.py

```
from django.contrib import admin
```

```

from program3.models import Course, Student

# Register your models here.
#admin.site.register(Student)
@admin.register(Student)
class StudentAdmin(admin.ModelAdmin):
    list_display = ('student_name', 'student_usn', 'student_sem')
    ordering= ('student_name',)
    search_fields = ('student_name',)
admin.site.register(Course)

```

### models.py

```

from django.db import models
from django.forms import ModelForm

# Create your models here.
class Meeting(models.Model):
    meeting_code=models.CharField(max_length=100)
    meeting_dt=models.DateField(auto_now_add=True)
    meeting_subject=models.CharField(max_length=100)
    meeting_np=models.IntegerField()

class Course(models.Model):
    course_code=models.CharField(max_length=40)
    course_name=models.CharField(max_length=100)
    course_credits=models.IntegerField(blank=True, null=True)
    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+"("+self.student_usn+")"

```

### templates – reg.html, selected\_students.html, course\_search.html

#### reg.html

```

<html>
    <body>
        <form method="post" action="">
            {% csrf_token %}

```



```

Student Name
<select name="sname">
{%for student in students %}
<option value="{{student.id}}">{{student.student_name}}</option>
{% endfor %}
</select><br>
Course Name
<select name="cname">
{%for course in courses %}
<option value="{{course.id}}">{{course.course_name}}</option>
{% endfor %}
</select><br>
<input type="submit" value="Enroll">
</form>
</body>
</html>

```

#### selected\_students.html

```

<html>
<body>
<table border>
<tr>
<th>Student Name</th>
<th>Student USN</th>
<th>Sem</th>
</tr>
{% for student in student_list %}
<tr>
<td>{{student.student_name}}</td>
<td>{{student.student_usn}}</td>
<td>{{student.student_sem}}</td>
</tr>
{% endfor %}
</table>
</body>
</html>

```

#### course\_search.html

```

<html>
<body>
<form method="POST" action="">
Courses
{% csrf_token %}
<select name="cname">
{%for course in courses %}
<option value="{{course.id}}">{{course.course_name}}</option>

```

```

        {% endfor %}
    </select>
    <input type="submit" value="Search">
</form>
</body>
</html>

```

### Urls.py

```

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

from program3.views import course_search, reg
urlpatterns = [
    path('admin/', admin.site.urls),
    path('reg/', reg),
    path('coursesearch/', course_search)

```

### setting.py

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'crudoperation',
    'theoryexample1',
    'program2',
    'labprg22',
    'program3'
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'program3/templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

```

```
]
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'studenttable',  
        'USER': 'root',  
        'PASSWORD': '',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```


**python manage.py createsuperuser**




**Perform remigrations before running: python manage.py makemigrations program3**








**python manage.py migrate**

**Run a WAMP Server**

**Create a new database - studenttable**

 studenttable

 program3\_course  
 program3\_meeting  
 program3\_student

 Browse  Structure  SQL  Search  Insert  Export  Import







✓ Showing rows 0 - 1 (2 total, Query took 0.0233 seconds.)

`SELECT * FROM `program3_student``

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 | Filter rows:  Sort by key:

Extra options

				id	student_usn	student_name	student_sem
<input type="checkbox"/>				1	JB001	RANJITHA J	6
<input type="checkbox"/>				2	JB002	REKHA	6

Server: MySQL:3306 » Database: studenttable » Table: program3\_course

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#)

✓ Showing rows 0 - 1 (2 total, Query took 0.0026 seconds.)

```
SELECT * FROM `program3_course`
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:  Sort by key: N

Extra options

				id	course_code	course_name	course_credits
<input type="checkbox"/>				1	c001	Full Stack Development	4
<input type="checkbox"/>				2	c002	Cloud Computing	3

☐ Check all With selected: Edit Copy Delete Export

Server: MySQL:3306 » Database: studenttable » Table: program3\_student\_enrolment

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#)

✓ Showing rows 0 - 2 (3 total, Query took 0.0051 seconds.)

```
SELECT * FROM `program3_student_enrolment`
```

☐ Profiling [\[ Edit inline \]](#) [\[ Edit \]](#) [\[ Explain SQL \]](#) [\[ Create PHP code \]](#) [\[ Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:  Sort by key: N

Extra options

				id	student_id	course_id
<input type="checkbox"/>				1	1	1
<input type="checkbox"/>				2	1	2
<input type="checkbox"/>				3	2	1

☐ Check all With selected: Edit Copy Delete Export

Run the terminal – python manage.py runserver

## Site administration

## AUTHENTICATION AND AUTHORIZATION

[Groups](#) [+ Add](#) [Change](#)[Users](#) [+ Add](#) [Change](#)

## LABPRG22

[Courses](#) [+ Add](#) [Change](#)[Students](#) [+ Add](#) [Change](#)

## PROGRAM3

[Courses](#) [+ Add](#) [Change](#)[Students](#) [+ Add](#) [Change](#)

## Recent actions

## My actions

None available

127.0.0.1:8000/coursesearch/

Student Name	Student USN	Sem
RANJITHA J	JB001	6
REKHA	JB002	6

127.0.0.1:8000/reg/

Student Name

Course Name

127.0.0.1:8000/reg/

# Student already enrolled

**7. Develop a Model form for student that contains his topic chosen for project, languages used and duration with a model called project.**

**Views.py**

```
from django.http import HttpResponseRedirect
from django.shortcuts import render

from program7.models import ProjectReg

def add_project(request):
    if request.method=="POST":
        form=ProjectReg(request.POST)
        if form.is_valid():
            form.save()
            return HttpResponseRedirect("<h1>Record inserted successfully</h1>")
        else:
            return HttpResponseRedirect("<h1>Record not inserted</h1>")
    else:
        form=ProjectReg()
        return render(request,"add_project.html",{ "form":form})
```

**models.py**

```
from django.db import models
from django.forms import ModelForm

# Create your models here.

class Project(models.Model):
    ptopic=models.CharField(max_length=200)
    plangauges=models.CharField(max_length=200)
    pduration=models.IntegerField()

class ProjectReg(ModelForm):
    required_css_class="required"
    class Meta:
        model=Project
        fields=['ptopic','plangauges','pduration']
```

**templates - add\_project.html**

**add\_project.html**

```
<html>
    <form method="post" action="">
        {% csrf_token %}
        <table>
```

```

        {{ form.as_table }}
    <tr>
        <td>
            <input type="submit" value="Submit">
        </td>
    </tr>
</table>
</form>
</html>

```

## Urls.py

```

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

from program31.views import add_project
urlpatterns = [
    path('admin/', admin.site.urls),
    path('add_project/', add_project)
]

```

## setting.py

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'crudoperation',
    'theoryexample1',
    'program2',
    'labprg22',
    'program3',
    'program31'
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'program31/templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
            ]
        }
    }
]

```

```

        'django.contrib.messages.context_processors.messages',
    ],
},
},
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'dataforms',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

```

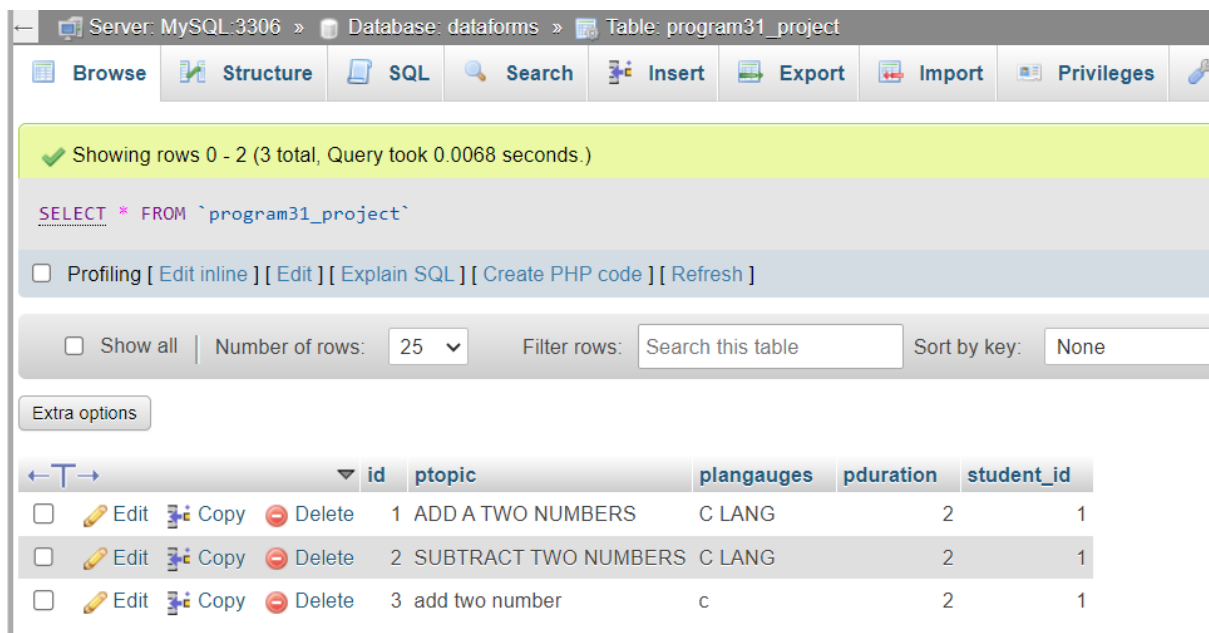
Perform remigrations before running:

`python manage.py makemigrations ap3`

`python manage.py migrate`

In Wamp server

Create a database – dataforms



Server: MySQL:3306 » Database: dataforms » Table: program31\_project

Showing rows 0 - 2 (3 total, Query took 0.0068 seconds.)

`SELECT * FROM `program31_project``

☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

			id	ptopic	plangauges	pduration	student_id	
<input type="checkbox"/>	Edit	Copy	Delete	1	ADD A TWO NUMBERS	C LANG	2	1
<input type="checkbox"/>	Edit	Copy	Delete	2	SUBTRACT TWO NUMBERS	C LANG	2	1
<input type="checkbox"/>	Edit	Copy	Delete	3	add two number	c	2	1

Run – `python manage.py runserver`



# Record inserted successfully

## Module-4

8. 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.

### Views.py

```
from django.shortcuts import render
from django.views import generic
from prg81.models import Student
class StudentListView(generic.ListView):
    model=Student
    template_name="student_list.html"
class StudentDetailView(generic.DetailView):
    model=Student
    template_name="student_detail.html"
```

In templates - student\_list, student\_detail

### student\_list.html

```
<html>
  <body>
    {% if student_list %}
      <table border>
        <tr>
          <th>USN</th>

          </tr>
          {% for student in student_list %}
            <tr>
              <td><a href="/student_detail/{{student.pk}}">{{
student.student_usn }}</a></td>

            </tr>
          {% endfor %}
        </table>
      {% else %}
        <h1>No Students Enrolled</h1>
      {% endif %}

    </body>
  </html>
```

### Student\_detail.html

```
<h1>Student Name: {{ student.student_name }}</h1>
```

<h1>Student USN: {{ student.student\_usn }}</h1>

<h1>Student Sem: {{ student.student\_sem }}</h1>

## Models.py

```
from django.db import models
class Student(models.Model):
    student_usn=models.CharField(max_length=12)
    student_name=models.CharField(max_length=30)
    student_sem=models.IntegerField()
```

## Urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from prg81.views import StudentDetailView, StudentListView
urlpatterns = [
    path('admin/', admin.site.urls),
    path('student_list/', StudentListView.as_view()),
    path('student_detail/<int:pk>/', StudentDetailView.as_view()),
```

## setting.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'crudoperation',
    'theoryexample1',
    'program2',
    'labprg22',
    'program3',
    'program31',
    'program41'
]
```

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS':
[os.path.join(BASE_DIR, 'crudoperation/templates'),os.path.join(BASE_DIR, 'progr
am41/templates')],
```

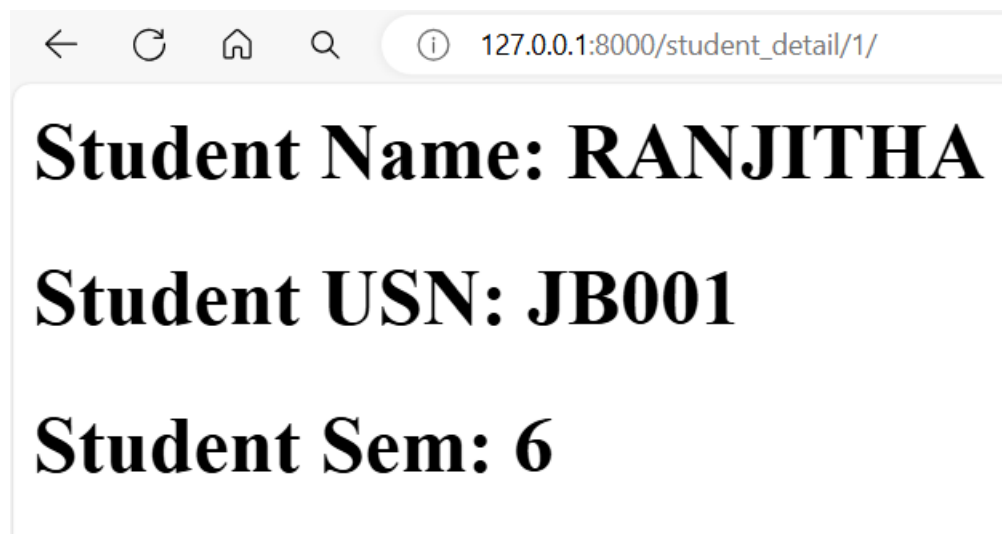
```
'APP_DIRS': True,
'OPTIONS': {
    'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'college',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

`python manage.py makemigrations`

`python manage.py migrate`

Run - `python manage.py runserver`



127.0.0.1:8000/student_list/	
USN	Courses Enrolled
{{ student.student_usn }}	
{{ student.student_usn }}	

**Student Name: RANJITHA**

**Student USN: JB001**

**Student Sem: 6**

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

#### Views.py

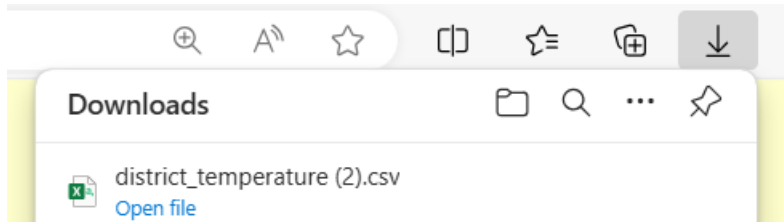
```
from django.http import HttpResponse
from django.shortcuts import render
import csv
def construct_csv(request):
    districts=["Shimoga","Bhadravathi","Bangalore","Dharwad","Raichur"]
    temperatures=[38,36,34,35,40]
    response=HttpResponse(content_type="text/csv")
    response['Content-Disposition'] = 'attachment;
filename="district_temperature.csv"'
    writer=csv.writer(response)
    writer.writerow(["Districts","Temperatures"])
    for d,t in zip(districts,temperatures):
        writer.writerow([d,t])
    return response
```

#### urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from program42.views import construct_csv
urlpatterns = [
    path('admin/', admin.site.urls),
```

```
path('construct_csv/', construct_csv),
]
```

Run- `python manage.py runserver`



	A	B	C	D
1	Districts	Temperatures		
2	Shimoga	38		
3	Bhadravath	36		
4	Bangalore	34		
5	Dharwad	35		
6	Raichur	40		
7				
8				

CSV file is generated and downloaded

## PDF generation

For reportlab.pdfgen they need to install `pip install reportlab` then later `pip freeze`

Views.py

```
from django.http import HttpResponse
from django.shortcuts import render
from reportlab.pdfgen import canvas
from labprg22.models import Course

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, re_path
from program43.views import construct_pdf_from_model
urlpatterns = [
    path('admin/', admin.site.urls),
    path('construct_pdf_from_model/', construct_pdf_from_model),
]
```



courses\_data (1).pdf

[Open file](#)

Course Name	Course Code	Credits
MATHS	C001	4
JAVA	C002	3
PYTHON	C003	3
FSD	C004	4
Cloud computing	C005	4

PDF file is generated and downloaded

## Module-5

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

#### Views.py

```
from django.shortcuts import render

# Create your views here.
from django.http import HttpResponse
from Module51.models import Course, Student
def regaj(request):
    if request.method == "POST":
        sid=request.POST.get("sname")
        cid=request.POST.get("cname")
        student=Student.objects.get(id=sid)
        course=Course.objects.get(id=cid)
        res=student.enrolment.filter(id=cid)
        if res:
            return HttpResponse("<h1>Student already enrolled</h1>")
        student.enrolment.add(course)
        return HttpResponse("<h1>Student enrolled successfully</h1>")

    else:
        students=Student.objects.all()
        courses=Course.objects.all()

        return render(request,"regaj.html",{ "students":students,
        "courses":courses})
```

#### Models.py

```
from django.db import models
class Course(models.Model):
    course_code=models.CharField(max_length=40)
    course_name=models.CharField(max_length=100)
    course_credits=models.IntegerField(blank=True, null=True)
    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+"("+self.student_usn+")"
```



## template file

### regaj.html

```
{% load static %}
<html>
  <body>
    <form method="post" action="">
      {% csrf_token %}
      Student Name
      <select name="sname" id="sname">
        {%for student in students %}
        <option value="{{student.id}}">{{student.student_name}}</option>
        {% endfor %}
      </select><br>
      Course Name
      <select name="cname" id="cname">
        {%for course in courses %}
        <option value="{{course.id}}">{{course.course_name}}</option>
        {% endfor %}

      </select><br>
      <span id="ans"></span>
      <input type="button" value="Enroll" id="ebtn">
    </form>
    <script src="{% static 'jquery.min.js' %}"></script>
    <script>
      $(document).ready(function(){
        $("#ebtn").click(function(){
          var sname = $("#sname").val();
          var cname = $("#cname").val();
          $.ajax({
            type: "POST",
            url: "/regaj/",
            data: {sname: sname, cname: cname,
              csrfmiddlewaretoken:"{{ csrf_token }}"
            },
            success: function(response){
              $("#ans").html(response)
            }
          });
        });
      });
    </script>
  </body>
</html>
```

## Static folder – jquery3.7.1.min.js

### Setting.py

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIR = [os.path.join(BASE_DIR, 'Module52/static')]
```

### Urls.py

```
from django.urls import include, path
from django.urls import path,reverse_lazy
from django.views.generic import CreateView

from Module51.views import regaj
from Module52.views import course_search_ajax

urlpatterns = [
    path('admin/', admin.site.urls),
    path('regaj/',regaj),
    path('course_search_ajax/', course_search_ajax),
]
```

### Setting.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'Module52',
    'Module51'
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'Module51/template')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
```

```

        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'stud',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

```

```

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

```

```


STATIC_URL = 'static/'
STATICFILES_DIR = [os.path.join(BASE_DIR, 'Module52/static')]

```

**python manage.py makemigrations**

**python manage.py migrate**

**python manage.py runserver**



← ↻ 🏠 🔍 ⓘ 127.0.0.1:8000/regaj/

Student Name

Course Name

# Student already enrolled

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

### Views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from Module51.models import Course, Student
def course_search_ajax(request):
    if request.method=="POST":
        cid=request.POST.get("cname")
        s=Student.objects.all()
        student_list=list()
        for student in s:
            if student.enrolment.filter(id=cid):
                student_list.append(student)
        if len(student_list)==0:
            return HttpResponse("<h1>No Students enrolled</h1>")
        return
    render(request, "selected_students.html", {"student_list":student_list})

    else:
        courses=Course.objects.all()
        return render(request, "course_search_ajax.html", {"courses":courses})
```

### 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(blank=True, null=True)
    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+"("+self.student_usn+")"
```

template – course\_search\_ajax.html, selected\_students.html

course\_search\_ajax.html

```

{% load static %}
<html>
  <body>
    <form method="POST" action="">
      Courses
      {% csrf_token %}
      <select name="cname" id="cname">
        {%for course in courses %}
          <option value="{{course.id}}">{{course.course_name}}</option>
        {% endfor %}
      </select>
      <input type="button" value="Search" id="serbtn">
      <span id="result"></span>
    </form>
  </body>
  <script src="{% static 'jquery.min.js' %}"></script>
  <script>
    $(document).ready(function(){
      $("#serbtn").click(function(){
        var cname = $("#cname").val();
        $.ajax({
          url: "/course_search_ajax/",
          type: "POST",
          data: {cname:cname,csrfmiddlewaretoken:"{{csrf_token }}"},
          success: function(response){
            $("#result").html(response);
          }
        });
      });
    });
  </script>
</html>

```

## Selected\_students.html

```

<html>
  <body>
    <table border>
      <tr>
        <th>Student Name</th>
        <th>Student USN</th>
        <th>Sem</th>
      </tr>
      {% for student in student_list %}
      <tr>
        <td>{{student.student_name}}</td>
        <td>{{student.student_usn}}</td>
        <td>{{student.student_sem}}</td>
      </tr>
      </table>

```

```

        </tr>
    {% endfor %}
</table>
</body>
</html>

```

## Urls.py

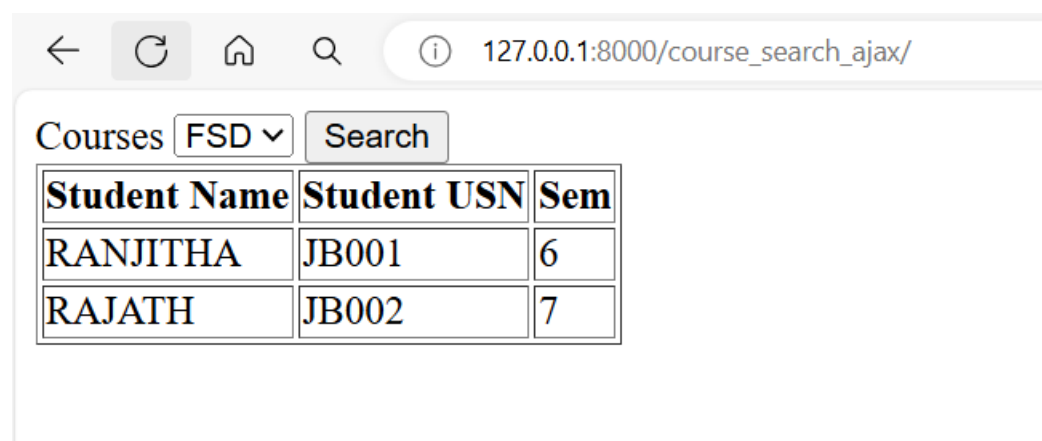
```

from django.urls import include, path
from django.urls import path,reverse_lazy
from django.views.generic import CreateView

from Module51.views import regaj
from Module52.views import course_search_ajax

urlpatterns = [
    path('admin/', admin.site.urls),
    path('regaj/',regaj),
    path('course_search_ajax/', course_search_ajax),
]

```



Courses

Student Name	Student USN	Sem
RANJITHA	JB001	6
RAJATH	JB002	7

**Prepared By,**  
**Ranjitha J**  
**Assistant Professor**  
**Dept, of ISE**  
**SJB Institute of Technology**