

Module-1

MVC based Web Designing

Laboratory Component:

1. Installation of Python, Django and Visual Studio code editors can be demonstrated.

Python download and installation Link:

<https://www.python.org/downloads/>

Visual Studio Code download and installation link:

<https://code.visualstudio.com/>

Django installation:

Open a command prompt and type following command:

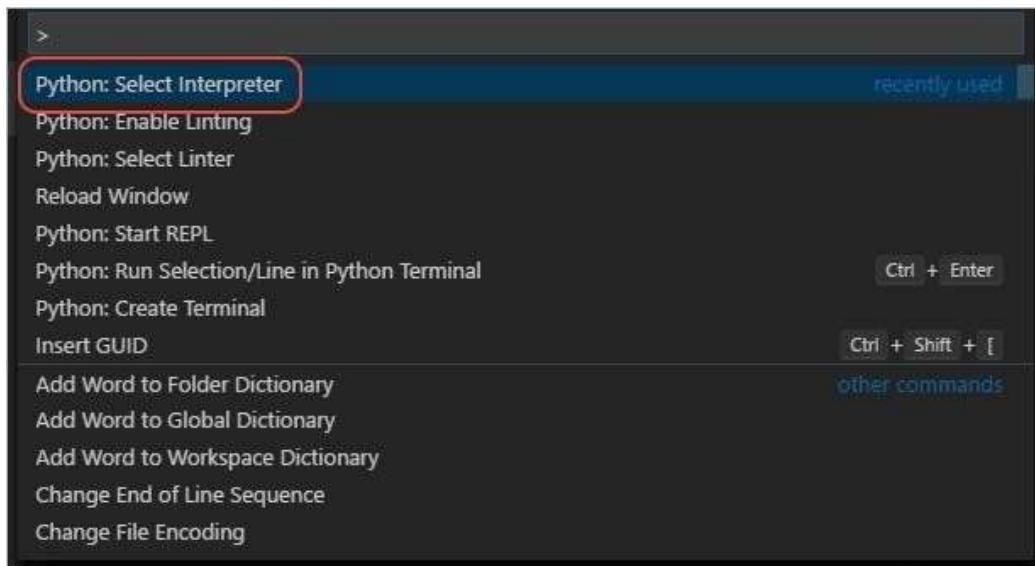
pip install django

2. Creation of virtual environment, Django project and App should be demonstrated Follow these steps

1. Install the Python extension.- Open VS Code IDE and click extensions there automatically u will be shown Python extension (Make sure you are connected to Internet)
2. On your file system, create a project folder
3. In that folder, use the following command (as appropriate to your computer) to create a virtual environment named env based on your current interpreter:

```
# Windows  
python -m venv env
```

4. Open the project folder in VS Code by running code .., or by running VS Code and using the **File > Open Folder** command.
5. In VS Code, open the Command Palette (**View > Command Palette** or (Ctrl+Shift+P)). Then select the **Python: Select Interpreter** command:



6. The command presents a list of available interpreters that VS Code can locate automatically (your list will vary; if you don't see the desired interpreter, see [Configuring Python environments](#)). From the list, select the virtual environment in your project folder that starts with ./env or .\env:
7. Create a New Terminal : In Menu Terminal -> New Terminal option

Creating project:

1. Create a django project -

Type following command in the terminal opened:

`django-admin startproject p .`

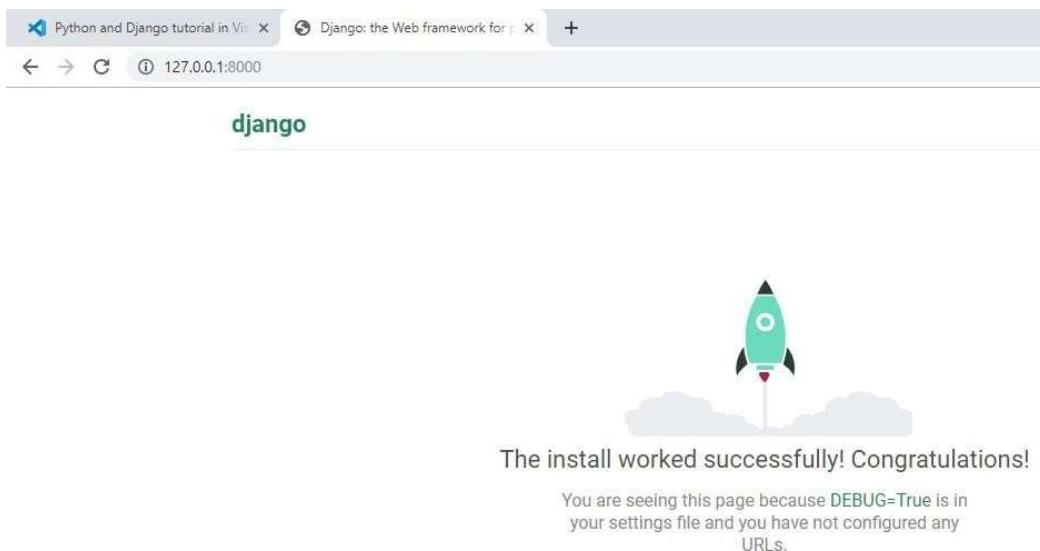
(dot following project name is important which refers to current directory)

This startproject command assumes (by use of . at the end) that the current folder is your project folder, and creates the following within it:

- `manage.py`: The Django command-line administrative utility for the project. You run administrative commands for the project using `python manage.py <command> [options]`.
 - A subfolder named `p` which contains the following files:
 - `__init__.py`: an empty file that tells Python that this folder is a Python package.
 - `wsgi.py`: an entry point for WSGI-compatible web servers to serve your project. You typically leave this file as-is as it provides the hooks for production web servers.
 - `settings.py`: contains settings for Django project, which you modify in the course of developing a web app.
 - `urls.py`: contains a table of contents for the Django project, which you also modify in the course of development.
2. To verify the Django project, make sure your virtual environment is activated, then start Django's development server using the command `python manage.py runserver`. The server runs on the default port 8000, and you see output like the following output in the terminal window:

Verify server by typing:

`python manage.py runserver`



When you run the server the first time, it creates a default SQLite database in the file `db.sqlite3`, which is intended for development purposes but can be used in production for low-volume web apps. Also, Django's built-in web server is intended *only* for local development purposes. When you deploy to a web host, however, Django uses the host's web server instead. The `wsgi.py` module in the Django project takes care of hooking into the production servers.

If you want to use a different port than the default 8000, specify the port number on the command line, such as `python manage.py runserver 5000`.

3. When you're done, close the browser window and stop the server in VS Code using `Ctrl+C` as indicated in the terminal output window.
4. In the VS Code Terminal with your virtual environment activated, run the administrative utility's `startapp` command in your project folder (where `manage.py` resides):

`python manage.py startapp lab1`

5. The command creates a folder called `lab1` that contains a number of code files and one subfolder. Of these, you frequently work with `views.py` (that contains the functions that define pages in your web app) and `models.py` (that contains classes defining your data objects). The `migrations` folder is used by Django's administrative utility to manage database versions. There are also the files `apps.py` (app configuration), `admin.py` (for creating an administrative interface), and `tests.py` (for unit tests).

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

In lab1 subfolder, make following changes to views.py:

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

# 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)
```

In project named first, make following changes to urls.py

```
from django.contrib import admin
from django.urls import path
from lab1.views import current_datetime
urlpatterns = [
```

```
    path('cdt/', current_datetime),
```

Output:



← → ⌛ 127.0.0.1:8000/cdt/

It is now 2024-02-03 18:42:01.631243.

VTU
Updates

4. 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.

In lab11 subfolder, make following changes to views.py:

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

```
# 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)
```

In project named first, make following changes to urls.py

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

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

```
from lab11.views import current_datetime,four_hours_ahead,four_hours_before
```

```
urlpatterns = [
```

```
    path('cdt/', current_datetime),
```

```
    path('fhrsa/',four_hours_ahead),
```

```
    path('fhrsb/',four_hours_before),
```

Output:

```
← → ⌂ ① 127.0.0.1:8000/fhrsa/
```

After 4hour(s), it will be 2024-02-03 22:43:50.544397.

```
← → ⌂ ① 127.0.0.1:8000/fhrsb/
```

Before 4 hour(s), it was 2024-02-03 14:44:10.994024.

Module-2

Django Templates and Models

Laboratory Component:

1. 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.http import HttpResponse
from django.shortcuts import render
from django.template import Context, Template

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

URLS.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist
urlpatterns = [
    path('admin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
    path('cts/<int:s>/<int:n>', create_table_of_squares),
    path('vc/<str:sentence>', vc),
    path('find_mode/<str:listofnum>', find_mode),
    path('template_test/', template_test),
    path('showlist/', showlist),
```

```
Template HTML file (inside ap2/templates subfolder)
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>
```

Output:

← → ⌂ ① 127.0.0.1:8000/showlist/

Unordered list of fruits

- Mango
- Apple
- Bananan
- Jackfruits

Ordered list of Students

1. Tony
2. Mony
3. Sony
4. Bob

2. 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 datetime import date
from django.http import HttpResponseRedirect
from django.shortcuts import render
from django.template import Context, Template
def home(request):
    return render(request,'home.html')

def aboutus(request):
    return render(request,'aboutus.html')

def contactus(request):
    return render(request,'contactus.html')
```

URLS.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus
urlpatterns = [
    path('admin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
```

```
path('cts/<int:s>/<int:n>', create_table_of_squares),
path('vc/<str:sentence>', vc),
path('find_mode/<str:listofnum>', find_mode),
path('template_test/', template_test),
path('showlist/', showlist),
path('list_of_subjects/', list_of_subjects),
path('aboutus/', aboutus),
path('home/', home),
path('contactus/', contactus),
```

Template files:

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; AIML, Developed by ABC, Inc.
        </footer>
    </body>
</html>
```

home.html

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

aboutus.html

```
{% extends 'layout.html' %}  
{% block title %}  
About Us  
{% endblock %}  
{% block content %}  
<h2>We are Django developers</h2>  
{% endblock %}
```

contactus.html

```
{% extends 'layout.html' %}  
{% block title %}  
Contact us  
{% endblock %}  
{% block content %}  
<h2>Our phone: 9900923050 <br>  
Address: Navule JNNCE</h2>  
{% endblock %}
```

Output:

← → ⌂ 127.0.0.1:8000/home/



[Home](#) | [About Us](#) | [Contact Us](#) |

This is the home page

© AIML, Developed by ABC, Inc.

3. 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

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()

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)
```

reg.html inside templates folder

```
<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>
```

views.py

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

from ap3.models import Course, Meeting, 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 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,"reg.html",{"students":students,
"courses":courses})
```

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus,getpos,stable
from ap3.views import insert_demo,update_demo,delete_demo,retrieve_demo
from ap3.views import reg
urlpatterns = [
    path('admin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
```

```
path('cts/<int:s>/<int:n>', create_table_of_squares),
path('vc/<str:sentence>', vc),
path('find_mode/<str:listofnum>', find_mode),
path('template_test/', template_test),
path('showlist/', showlist),
path('list_of_subjects/', list_of_subjects),
path('aboutus/', aboutus),
path('home/', home),
path('contactus/', contactus),
path('getpos/', getpos),
path('stable/', stable),
path('insert_demo/', insert_demo),
path('update_demo/', update_demo),
path('delete_demo/', delete_demo),
path('retreive_demo/', retreive_demo),
path('reg/', reg),
```



Database input:

Insert student and courses record in phpMyAdmin

☰ Browse Structure SQL Search Insert Export

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `ap3_student`
```

Profiling [Edit inline]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

			id	student_usn	student_name	student_sem				
<input type="checkbox"/>		Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	1	4JN22AI001	Sham	4
<input type="checkbox"/>		Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	2	4JN22AI002	Manish	4
<input type="checkbox"/>		Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	3	4JN22AI003	Suma	4

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

```
SELECT * FROM `ap3_course`
```

Profiling [Edit inline]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	course_code	course_name	course_credits
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	21AI42	DAA	4
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	21AI43	MPC	3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	21AI44	UHV	1

OUTPUT:

← → ⌂ 127.0.0.1:8000/reg/

Student Name Course Name

← → ⌂ 127.0.0.1:8000/reg/

Student enrolled successfully

BackEnd

Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `ap3_student_enrolment`
```

Profiling [Edit inline]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

				id	student_id	course_id
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	1	1
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	1	3
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	2	2
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	2	1
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	1	2

If you try again, you will get

← → ⌂ 127.0.0.1:8000/reg/

Student already enrolled

**VTU
Updates**

Module-3

Django Admin Interfaces and Model Forms

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

```
python manage.py createsuperuser
```

```
make following changes to admin.py
```

```
from django.contrib import admin

from ap3.models import Course, Student

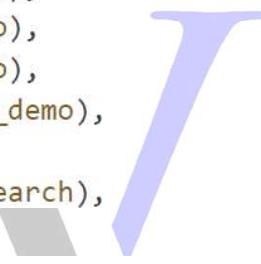
# Register your models here.
#admin.site.register(Student)
@admin.register(Student)
class StudentAdmin(admin.ModelAdmin):
    list_display = ('student_name','student_usn','student_sem')
    ordering=('student_name',)
    search_fields = ('student_name',)
admin.site.register(Course)
```

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus,getpos,stable
from ap3.views import insert_demo,update_demo,delete_demo,retrieve_demo
from ap3.views import reg,course_search
admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"

urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
```

```
path('cts/<int:s>/<int:n>', create_table_of_squares),
path('vc/<str:sentence>', vc),
path('find_mode/<str:listofnum>', find_mode),
path('template_test/', template_test),
path('showlist/', showlist),
path('list_of_subjects/', list_of_subjects),
path('aboutus/', aboutus),
path('home/', home),
path('contactus/', contactus),
path('getpos/', getpos),
path('stable/', stable),
path('insert_demo/', insert_demo),
path('update_demo/', update_demo),
path('delete_demo/', delete_demo),
path('retreive_demo/', retreive_demo),
path('reg/', reg),
path('course_search/', course_search),
```



Changes to 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+")"
```

Perform remigrations before running:

```
python manage.py makemigrations ap3
```

```
python manage.py migrate
```

Output:

The screenshot shows a browser window with the URL `127.0.0.1:8000/secretadmin/login/?next=/secretadmin/`. The page has a dark blue header bar with the text "My Site Header". Below it is a light gray content area containing two input fields labeled "Username:" and "Password:", each with a corresponding text input box. At the bottom right of this area is a blue "Log in" button.

The screenshot shows a browser window with the URL `127.0.0.1:8000/secretadmin/`. The page has a dark blue header bar with the text "My Site Header". Below it is a light gray content area with the message "WELCOME, AIMLJNN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)".

My Site Index

The screenshot shows a browser window with a dark blue header bar containing the text "AP3". Below it is a light gray content area with two sections: "Courses" and "Students". Each section has a "Add" button (with a green plus sign) and a "Change" button (with a yellow pencil icon). Below these sections is a dark blue footer bar with the text "AUTHENTICATION AND AUTHORIZATION". Underneath this footer are two more sections: "Groups" and "Users", each with its own "Add" and "Change" buttons.

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

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+")"
class Project(models.Model):
    student=models.ForeignKey(Student,on_delete=models.CASCADE)
    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=['student','ptopic','plangauges','pduration']
```

views.py

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

from ap3.models import Course, Meeting, ProjectReg, Student

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

add_project.html inside templates folder

```
<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 ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
```

```
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus,getpos,stable
from ap3.views import insert_demo,update_demo,delete_demo,retreive_demo
from ap3.views import reg,course_search,add_project

admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"
urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
    path('cts/<int:s>/<int:n>', create_table_of_squares),
    path('vc/<str:sentence>', vc),
    path('find_mode/<str:listofnum>', find_mode),
    path('template_test/', template_test),
    path('showlist/', showlist),
    path('list_of_subjects/', list_of_subjects),
    path('aboutus/', aboutus),
    path('home/', home),
    path('contactus/', contactus),
    path('getpos/', getpos),
    path('stable/', stable),
    path('insert_demo/', insert_demo),
    path('update_demo/', update_demo),
    path('delete_demo/', delete_demo),
    path('retreive_demo/', retreive_demo),
    path('reg/', reg),
    path('course_search/', course_search),
    path('add_project/', add_project),
```

Perform remigrations before running:

```
python manage.py makemigrations ap3
```

```
python manage.py migrate
```

Output:

← → ⌂ 127.0.0.1:8000/add_project/

Student: Sham(4JN22AI001) ▾

Ptopic: AI

Plangauges: Python,HTML

Pduration: 22

Submit

← → ⌂ 127.0.0.1:8000/add_project/

Record inserted successfully

BackEnd

Browse Structure SQL Search Insert Export

Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)

```
SELECT * FROM `ap3_project`
```

Profiling [Edit inline]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	id	ptopic	plangauges	pduration	student_id
<input type="checkbox"/>	1	IoT	Python,Arduino	20	1
<input type="checkbox"/>	2	AI	Python,HTML	22	1

Module-4

Generic Views and Django State Persistence

1. For student's 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.views import generic
class StudentListView(generic.ListView):
    model=Student
    template_name="student_list.html"

class StudentDetailView(generic.DetailView):
    model=Student
    template_name="student_detail.html"
```

student_list.html inside templates folder

```
<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>
  </body>
</html>
```

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

</body>
</html>
```

student_detail.html inside templates folder

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

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after, display_string
from ap2.views import create_table_of_squares, vc, find_mode
from ap2.views import template_test, showlist, list_of_subjects
from ap2.views import aboutus, home, contactus, getpos, stable
from ap3.views import insert_demo, update_demo, delete_demo, retreive_demo
from ap3.views import reg, course_search, add_project
from ap3.views import StudentListView, StudentDetailView

admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"
urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
```

```
re_path('check_number/(\d){1,2}/', check_number),
path('cts/<int:s>/<int:n>', create_table_of_squares),
path('vc/<str:sentence>', vc),
path('find_mode/<str:listofnum>', find_mode),
path('template_test/', template_test),
path('showlist/', showlist),
path('list_of_subjects/', list_of_subjects),
path('aboutus/', aboutus),
path('home/', home),
path('contactus/', contactus),
path('getpos/', getpos),
path('stable/', stable),
path('insert_demo/', insert_demo),
path('update_demo/', update_demo),
path('delete_demo/', delete_demo),
path('retreive_demo/', retreive_demo),
path('reg/', reg),
path('course_search/', course_search),
path('add_project/', add_project),
path('student_list/', StudentListView.as_view()),
path('student_detail/<int:pk>/', StudentDetailView.as_view()),
```

Output:

← → ⌂ ① 127.0.0.1:8000/student_list/

USN	Courses Enrolled
4JN22AI001	DAA MPC UHV
4JN22AI002	DAA MPC
4JN22AI003	

← → ⌂ ① 127.0.0.1:8000/student_detail/1/

Student Name: Sham

Student USN: 4JN22AI001

Student Sem: 4

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

views.py

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

from ap3.models import Course, Meeting, ProjectReg, Student

import csv
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_c
redits])
    return response
```

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus,getpos,stable
from ap3.views import insert_demo,update_demo,delete_demo,retrieve_demo
from ap3.views import reg,course_search,add_project
from ap3.views import StudentListView,StudentDetailView,construct_csv
from ap3.views import construct_csv_from_model

admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"
urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re path('check number/(\d){1,2}/',check number),
```

```
path('cts/<int:s>/<int:n>', create_table_of_squares),
path('vc/<str:sentence>', vc),
path('find_mode/<str:listofnum>', find_mode),
path('template_test/', template_test),
path('showlist/', showlist),
path('list_of_subjects/', list_of_subjects),
path('aboutus/', aboutus),
path('home/', home),
path('contactus/', contactus),
path('getpos/', getpos),
path('stable/', stable),
path('insert_demo/', insert_demo),
path('update_demo/', update_demo),
path('delete_demo/', delete_demo),
path('retreive_demo/', retreive_demo),
path('reg/', reg),
path('course_search/', course_search),
path('add_project/', add_project),
path('student_list/', StudentListView.as_view()),
path('student_detail/<int:pk>/', StudentDetailView.as_view()),
path('construct_csv/', construct_csv),
path('construct_csv_from_model/', construct_csv_from_model),
```

Output:

CSV file is generated and downloaded

VTU
Updates

- 3. Develop example Django app that performs PDF generation for any models created in previous laboratory component**

views.py

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

from ap3.models import Course, Meeting, ProjectReg, Student
from reportlab.pdfgen import canvas

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
```

Updates

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus,getpos,stable
```

```
from ap3.views import insert_demo,update_demo,delete_demo,retreive_demo
from ap3.views import reg,course_search,add_project
from ap3.views import StudentListView,StudentDetailView,construct_csv
from ap3.views import construct_csv_from_model
from ap3.views import construct_pdf_from_model
admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"
urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
    path('cts/<int:s>/<int:n>', create_table_of_squares),
    path('vc/<str:sentence>', vc),
    path('find_mode/<str:listofnum>', find_mode),
    path('template_test/', template_test),
    path('showlist/', showlist),
    path('list_of_subjects/', list_of_subjects),
    path('aboutus/', aboutus),
    path('home/', home),
    path('contactus/', contactus),
    path('getpos/', getpos),
    path('stable/', stable),
    path('insert_demo/', insert_demo),
    path('update_demo/', update_demo),
    path('delete_demo/', delete_demo),
    path('retreive_demo/', retreive_demo),
    path('reg/', reg),
    path('course_search/', course_search),
    path('add_project/', add_project),
    path('student_list/', StudentListView.as_view()),
    path('student_detail/<int:pk>/', StudentDetailView.as_view()),
    path('construct_csv/', construct_csv),
    path('construct_csv_from_model/', construct_csv_from_model),
    path('construct_pdf_from_model/', construct_pdf_from_model),
```

Output:

PDF file is generated and downloaded

Module-5

jQuery and AJAX Integration in Django

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

views.py

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

from ap3.models import Course, Meeting, ProjectReg, 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 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,"regaj.html",{"students":students,
"courses":courses})
```

IndoTec

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>
urls.py
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after,display_string
from ap2.views import create_table_of_squares,vc,find_mode
from ap2.views import template_test,showlist,list_of_subjects
from ap2.views import aboutus,home,contactus,getpos,stable
```

```
from ap3.views import insert_demo,update_demo,delete_demo,retreive_demo
from ap3.views import reg,course_search,add_project
from ap3.views import StudentListView,StudentDetailView,construct_csv
from ap3.views import construct_csv_from_model,construct_pdf
from ap3.views import construct_pdf_from_model
from ap3.views import expr,cbox,regaj,course_search_ajax
admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"
urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/',check_number),
    path('cts/<int:s>/<int:n>', create_table_of_squares),
    path('vc/<str:sentence>', vc),
    path('find_mode/<str:listofnum>', find_mode),
    path('template_test/', template_test),
    path('showlist/', showlist),
    path('list_of_subjects/', list_of_subjects),
    path('aboutus/', aboutus),
    path('home/', home),
    path('contactus/', contactus),
    path('getpos/', getpos),
    path('stable/', stable),
    path('insert_demo/', insert_demo),

    path('update_demo/', update_demo),
    path('delete_demo/', delete_demo),
    path('retreive_demo/', retreive_demo),
    path('reg/', reg),
    path('course_search/', course_search),
    path('add_project/', add_project),
    path('student_list/', StudentListView.as_view()),
    path('student_detail/<int:pk>/', StudentDetailView.as_view()),
    path('construct_csv/', construct_csv),
    path('construct_csv_from_model/', construct_csv_from_model),
    path('construct_pdf/', construct_pdf),
    path('construct_pdf_from_model/', construct_pdf_from_model),
    path('expr/', expr),
    path('cbox/', cbox),
    path('regaj/','regaj'),
```

Output:

← → ⌛ 127.0.0.1:8000/regaj/

Student Name

Course Name

Enroll

← → ⌛ 127.0.0.1:8000/regaj/

Student Name

Course Name

Student already enrolled

Enroll

← → ⌛ 127.0.0.1:8000/regaj/

Student Name

Course Name

Student enrolled successfully

Enroll

**VTU
Updates**

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

views.py

```
from django.http import HttpResponseRedirect

from django.shortcuts import render

from ap3.models import Course, Meeting, ProjectReg, 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 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_aj.html",{"courses":courses})
```

course_search_aj.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>
```

```
        }
    });
});
});
</script>
</html>
```

urls.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap1.views import check_number, current_date_time
from ap1.views import four_hours_after, four_hours_before
from ap1.views import n_hours_after, display_string
from ap2.views import create_table_of_squares, vc, find_mode
from ap2.views import template_test, showlist, list_of_subjects
from ap2.views import aboutus, home, contactus, getpos, stable
from ap3.views import insert_demo, update_demo, delete_demo, retreive_demo
from ap3.views import reg, course_search, add_project
from ap3.views import StudentListView, StudentDetailView, construct_csv
from ap3.views import construct_csv_from_model, construct_pdf
from ap3.views import construct_pdf_from_model
from ap3.views import expr, cbox, regaj, course_search_ajax, cookie_demo
admin.site.site_header="My Site Header"
admin.site.site_title="My Site Title"
admin.site.index_title="My Site Index"
urlpatterns = [
    path('secretadmin/', admin.site.urls),
    path('cdt/', current_date_time),
    path('fha/', four_hours_after),
    path('fhb/', four_hours_before),
    path('nha/<int:num>', n_hours_after),
    path('display_string/<slug:sentence>', display_string),
    re_path('check_number/(\d){1,2}/', check_number),
    path('cts/<int:s>/<int:n>', create_table_of_squares),
    path('vc/<str:sentence>', vc),
    path('find_mode/<str:listofnum>', find_mode),
    path('template_test/', template_test),
    path('showlist/', showlist),
    path('list_of_subjects/', list_of_subjects),
    path('aboutus/', aboutus),
    path('home/', home),
    path('contactus/', contactus),
    path('getpos/', getpos),
    path('stable/', stable),
    path('insert_demo/', insert_demo),
    path('update_demo/', update_demo),
    path('delete_demo/', delete_demo),
```

```
path('retreive_demo/', retreive_demo),
path('reg/', reg),
path('course_search/', course_search),
path('add_project/', add_project),
path('student_list/', StudentListView.as_view()),
path('student_detail/<int:pk>/', StudentDetailView.as_view()),
path('construct_csv/', construct_csv),
path('construct_csv_from_model/', construct_csv_from_model),
path('construct_pdf/', construct_pdf),
path('construct_pdf_from_model/', construct_pdf_from_model),
path('expr/', expr),
path('cbox/', cbox),
path('course_search_ajax/', course_search_ajax),
```

Output:

← → ⌂ 127.0.0.1:8000/course_search_ajax/

Courses **DAA** ▾ Search

← → ⌂ 127.0.0.1:8000/course_search_ajax/

Courses **DAA** ▾ Search

Student Name	Student USN	Sem
Sham	4JN22AI001	4
Manish	4JN22AI002	4