

## LAB COMPONENT SOLUTION

### Module-1: MVC based Web Designing

#### Laboratory Component:

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

1. Python download and installation Link:  
<https://www.python.org/downloads/>
2. Visual Studio Code download and installation link: <https://code.visualstudio.com/>
3. 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

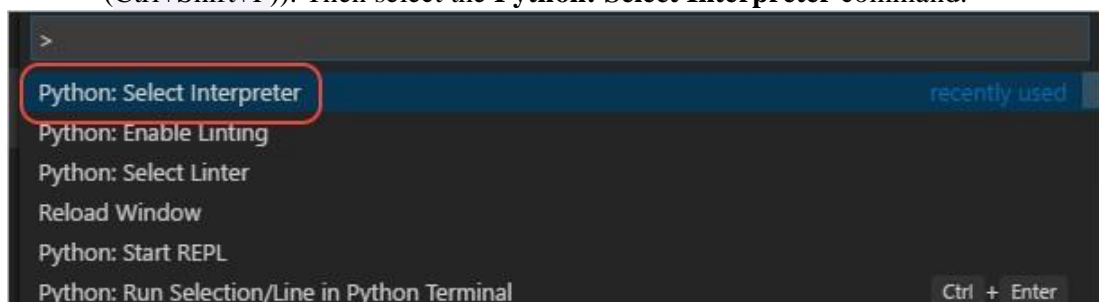
Follow these steps

- a. 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)
- b. On your file system, create a project folder
- i. 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
```

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



- iv. 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`:
- v. Create a New Terminal : In Menu Terminal -> New Terminal option

### Creating project:

- i. Create a django project -

Type following command in the terminal opened:

`django-admin startproject p .`

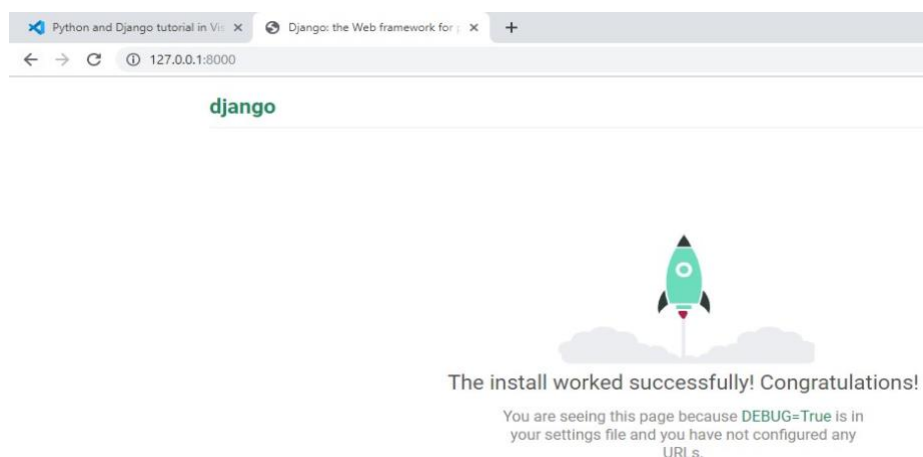
(dot following project name is important which refers to current directory)

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.
- i. 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`.
- 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.
- 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
```

- 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:**



### 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 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)

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 lab1.views import current_datetime, four_hours_ahead, four_hours_before

urlpatterns = [

    path('cdt/', current_datetime), path('fhrs_a/', four_hours_ahead),
    path('fhrs_b/', four_hours_before),

]

```

**Output:**

← → ↻ ⓘ 127.0.0.1:8000/fhrs\_a/

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

← → ↻ ⓘ 127.0.0.1:8000/fhrs\_b/

**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","Bananan","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('fhhb/', 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:

← → ↻ ⓘ 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 HttpResponse 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
import template_test,showlist,list_of_subjects from ap2.views import
aboutus,home,contactus

urlpatterns = [
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>Out 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,retreive_demo from ap3.views
import reg
urlpatterns = [
    path('admin/', admin.site.urls),
    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**



← → ↻ ⓘ 127.0.0.1:8000/course\_search/

Courses  Search

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:

Extra options

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

Browse
 Structure
 SQL
 Search
 Insert
 Export

✓ 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:

Extra options

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

Output:

Student Name

Course Name

## Student enrolled successfully

BackEnd

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

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

☐ Print

☐ Show all | Number of rows:  Filter rows:

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

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

# Student already enrolled





### 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,retreive_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('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):
```

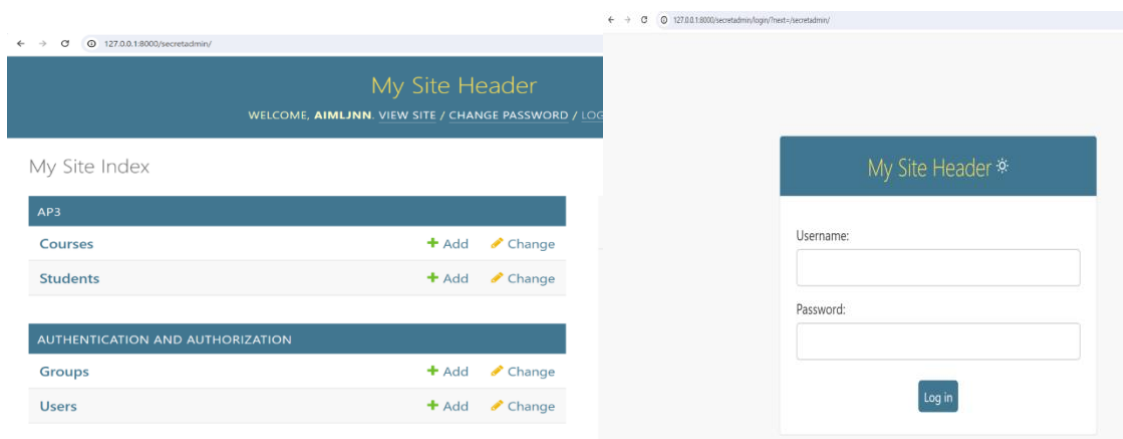
```
        return self.student_name+"("+self.student_usn+")"
```

### Perform remigrations before running: python

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

```
migrate
```

**Output:**



**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 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})

```

### **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, retrieve_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('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('retrieve_demo/', retrieve_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

← → ↻ 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:

Extra options

id

ptopic

plangauges

pduration

student\_id

☐

Edit

Copy

Delete

1 IoT Python,Arduino 20 1

☐

Edit

Copy

Delete

2 AI Python,HTML 22 1

## Module 4: Generic views and Django State Persistence

1. For students enrollment 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>
    {% 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, retrieve_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:

← → ↻ ⓘ 127.0.0.1:8000/student\_list/

USN	Courses Enrolled
<a href="#">4JN22AI001</a>	DAA MPC UHV
<a href="#">4JN22AI002</a>	DAA MPC
<a href="#">4JN22AI003</a>	

← → ↻ ⓘ 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(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 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
```

```
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('retrieval_demo/', retrieval_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),

```

]

### Output:

CSV file is generated and downloaded

### 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_credits])
    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('stable/', stable), path('insert_demo/',
    insert_demo), path('update_demo/',
    update_demo), path('delete_demo/',
    delete_demo), path('retrieve_demo/',
    retrieve_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

## Develop example Django app that performs PDF 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
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

```

### **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
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

SUBJECT: FULLSTACK DEVELOPMENT (21CS62)

## LAB COMPONENT SOLUTIONS

*Develop a Django app to produce following web page*

Save water



### pos.html inside templates folder

```
{% load static %}
<html>
  <link rel="stylesheet" href="{% static 'pos.css' %}">
  <body>
    <p id="p1">Save Water</p>
    
    
  </body>
</html>
```

### pos.css inside static folder

```
#p1 {color:blue;font-size:20pt;font-weight:bold}
#i2 {position:absolute;top:400px;left:250px;opacity:0.3}
```

### views.py

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



```

from django.template import Context, Template
def getpos(request):
    return render(request, 'pos.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
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),

```

]

### Output:

*Develop a Django app to produce following web page*

**stable.html in templates folder**

```
{% load static %}
<html>
  <link rel="stylesheet" href="{% static 'stable.css' %}">
  <body>
    <table border>
      <tr>
        <th>Name</th>
        <th>USN</th>
        <th>Marks</th>
      </tr>
      <tr>
        <td>Mani</td>
        <td>4JN22AI019</td>
        <td>80</td>
      </tr>
```

```

        <tr>
            <td>Gani</td>
            <td>4JN22AI020</td>
            <td>70</td>
        </tr>
        <tr>
            <td>Pani</td>
            <td>4JN22AI021</td>
            <td>82</td>
        </tr>
    </table>
</body>
</html>

```

### **stable.css in static folder**

```

td,th {padding:35px }
table { background-image:url(tp.png);background-repeat:no-repeat; background-
position: center;}

```

### **views.py**

```

from datetime import date
from django.http import HttpResponse from
django.shortcuts import render
from django.template import Context, Template def
getpos(request):
    return render(request,'pos.html') def
stable(request):
    return render(request,'stable.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 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),

```

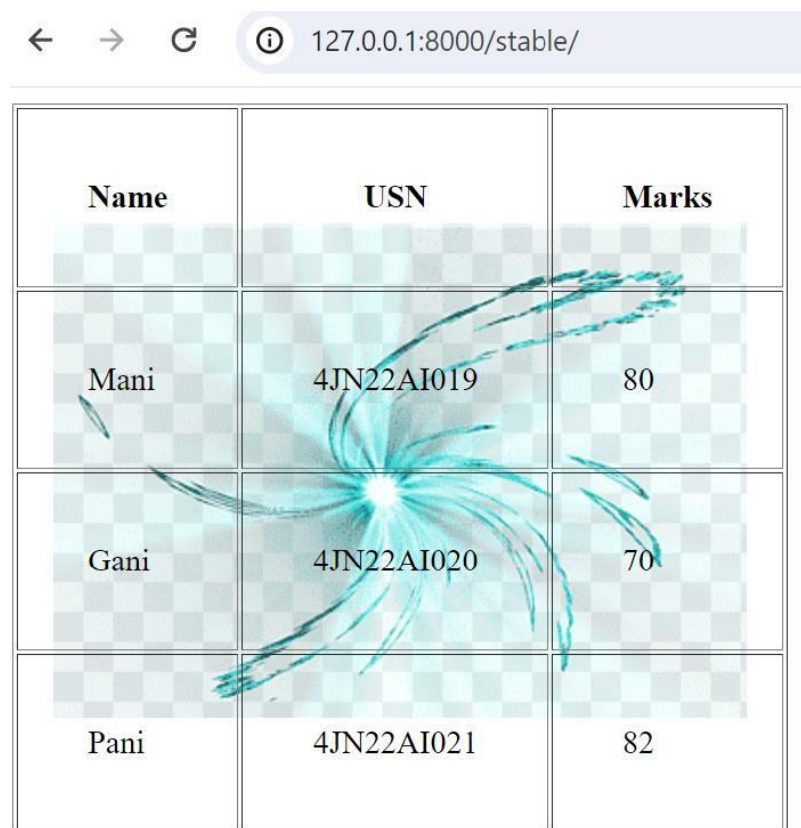
```

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),

```

]

### Output:



Name	USN	Marks
Mani	4JN22AI019	80
Gani	4JN22AI020	70
Pani	4JN22AI021	82

**Develop a Django app that demonstrates all CRUD operations for an onlinemeeting database**

### models.py

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

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()

```

### **views.py**

```

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

```

```

from ap3.models import Course, Meeting, Student # Create

```

your views here.

```

def insert_demo(request):
    m=Meeting(meeting_code="m002",meeting_dt="2024-04-
10",meeting_subject="WTW",meeting_np=50)
    m.save() m=Meeting(meeting_code="m003",meeting_dt="2024-
04-
10",meeting_subject="Parent Meet",meeting_np=50) m.save()
    m=Meeting(meeting_code="m004",meeting_dt="2024-04-
10",meeting_subject="Course Attainment",meeting_np=50) m.save()
    m=Meeting(meeting_code="m005",meeting_dt="2024-04-
10",meeting_subject="Infrastructure",meeting_np=50) m.save()
    return HttpResponse("<h1>Record inserted successfully</h1>")

```

```

def update_demo(request):
    m=Meeting.objects.get(meeting_code="m002")
    m.meeting_dt="2024-04-11"
    m.meeting_np=100
    m.save()
    return HttpResponse("<h1>Record updated successfully</h1>")

```

```

def delete_demo(request):
    m=Meeting.objects.get(meeting_code="m005") m.delete()
    return HttpResponse("<h1>Record deleted successfully</h1>")

```

```

from django.db.models import Q def
retreive_demo(request):
    m=Meeting.objects.filter(Q(meeting_subject__contains = "Meet") & Q(meeting_np__lte =
50))
    result=""
    for meeting in m:

```

```

        result+="

>%s,%s,%s,%d</p>"%(meeting.meeting_code, meeting.meeting_subject,
meeting.meeting_dt,meeting.meeting_np)
    return HttpResponse(result)


```

### 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 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),
]

```

### Output:

← → ↻ ⓘ 127.0.0.1:8000/insert\_demo/

# Record inserted successfully

### Backend:

✓ Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.)

```
SELECT * FROM `ap3_meeting`
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain](#) ]

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

Extra options

<div>← T →</div>					id	meeting_code	meeting_dt	meeting_subject	meeting_np
<input type="checkbox"/>	 Edit	 Copy	 Delete		1	m001	2024-04-10	Lab syllabus	50
<input type="checkbox"/>	 Edit	 Copy	 Delete		2	m002	2024-04-11	Mentor Meet	100
<input type="checkbox"/>	 Edit	 Copy	 Delete		3	m003	2024-04-10	Parent Meet	50
<input type="checkbox"/>	 Edit	 Copy	 Delete		4	m004	2024-04-10	Course Attainment	50
<input type="checkbox"/>	 Edit	 Copy	 Delete		9	m005	2024-04-10	Infrastructure	50

← → ↻ ⓘ 127.0.0.1:8000/update\_demo/

## Record updated successfully

### BackEnd

✓ Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

```
SELECT * FROM `ap3_meeting`
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain](#) ]

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

Extra options

<div><div><div>←</div><div>T</div><div>→</div></div><div></div></div>					id	meeting_code	meeting_dt	meeting_subject	meeting_np
<div><div><div></div></div><div><div><div></div></div><div><div></div></div></div><div><div>Edit</div></div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Copy</div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Delete</div></div>	1	m001	2024-04-10	Lab syllabus	50		
<div><div><div></div></div><div><div><div></div></div><div><div></div></div></div><div><div>Edit</div></div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Copy</div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Delete</div></div>	2	m002	2024-04-11	Mentor Meet	100		
<div><div><div></div></div><div><div><div></div></div><div><div></div></div></div><div><div>Edit</div></div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Copy</div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Delete</div></div>	3	m003	2024-04-10	Parent Meet	50		
<div><div><div></div></div><div><div><div></div></div><div><div></div></div></div><div><div>Edit</div></div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Copy</div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Delete</div></div>	4	m004	2024-04-10	Course Attainment	50		
<div><div><div></div></div><div><div><div></div></div><div><div></div></div></div><div><div>Edit</div></div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Copy</div></div>	<div><div><div></div></div><div><div></div></div></div> <div><div>Delete</div></div>	9	m005	2024-04-10	Infrastructure	50		

← → ↻ ⓘ 127.0.0.1:8000/delete\_demo/

## Record deleted successfully

## BackEnd

Browse Structure SQL Search Insert Export Import P

✓ Showing rows 0 - 3 (4 total, Query took 0.0005 seconds.)

```
SELECT * FROM `ap3_meeting`
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain](#) ]

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

Extra options

					id	meeting_code	meeting_dt	meeting_subject	meeting_np
<input type="checkbox"/>	Edit	Copy	Delete		1	m001	2024-04-10	Lab syllabus	50
<input type="checkbox"/>	Edit	Copy	Delete		2	m002	2024-04-11	Mentor Meet	100
<input type="checkbox"/>	Edit	Copy	Delete		3	m003	2024-04-10	Parent Meet	50
<input type="checkbox"/>	Edit	Copy	Delete		4	m004	2024-04-10	Course Attainment	50

127.0.0.1:8000/retrieve\_demo/

m003,Parent Meet,2024-04-10,50



## LAB COMPONENT SOLUTIONS

**Module-1: Additional Programs on Django Views and URLs** *Develop a Django app that displays tables of squares of pairs of numbers input in the URL.*

### **Views.py**

```
from datetime import date
from django.http import HttpResponse
from django.shortcuts import render
from django.template import Context, Template
def create_table_of_squares(request,s,n):
    result=""
    for i in range(1,n+1): result+="

" +str(s)+"*" +str(i)+"=" +str((s*i))+"</p>"
    return HttpResponse(result)


```

### **URLS.py**

```
from django.contrib import admin
from django.urls import path, re_path
from ap2.views import create_table_of_squares
urlpatterns = [
    path('admin/', admin.site.urls), path('cts/<int:s>/<int:n>',
    create_table_of_squares),

]
```

### Output

← → 🔍 127.0.0.1:8000/cts/3/6

## Table of squares

3\*1=3

3\*2=6

3\*3=9

3\*4=12

***Develop a Django app that displays number of vowels and consonants and also list of vowels and consonants for any input sentence specified in the URL.***

#### **Views.py**

```
def vc(request,sentence): vow_cnt=0
    cons_cnt=0 vow_dict=dict()
    cons_dict=dict()
    for letter in sentence: if
        letter.isalpha():
            if letter in "aeiouAEIOU": vow_cnt=vow_cnt+1
                vow_dict[letter]=vow_dict.get(letter,0)+1 else:
                    cons_cnt=cons_cnt+1 cons_dict[letter]=cons_dict.get(letter,0)+1

    result="<h1>%d Vowels and %d Consonants</h1>" % (vow_cnt,cons_cnt) result+="<h2>Vowel
    Counter</h2>"
    for key,value in vow_dict.items():
        result+="<p>%s:%d</p>"%(key,value)
    result+="<h2>Consonant Counter</h2>" for
    key,value in cons_dict.items():
        result+="<p>%s:%d</p>"%(key,value) return

    HttpResponse(result)
```

#### **URLS.py**

```
from django.contrib import admin
from django.urls import path, re_path
from ap2.views import create_table_of_squares,vc urlpatterns = [

    path('cts/<int:s>/<int:n>',create_table_of_squares), path('vc/<str:sentence>', vc),

]
```

Output:

## 6 vowels and 8 consonants

### Vowel counter

a: 5

e: 1

### Consonant counter

b: 1

n: 3

d: 1

p: 2

l: 1

*Develop a Django app that finds the mode of a given set of numbers specified in the URL*

### Views.py

```
def find_mode(request, listofnum):
    arr=listofnum.split(",") num_count=dict()
    for num in arr:
        num_count[num]=num_count.get(num,0)+1
    num_count=sorted(num_count.items(),key=lambda item:item[1])
    num_count.reverse()
    result="<p><span style=color:red>%s</span> appears <span style=background-
color:yellow>%s</span> times"% (num_count[0][0],num_count[0][1])
    return HttpResponse(result)
```

### URLS.py

```
from django.contrib import admin
from django.urls import path, re_path
from ap2.views import create_table_of_squares,vc,find_mode urlpatterns = [
    path('admin/', admin.site.urls), path('cts/<int:s>/<int:n>',
    create_table_of_squares), path('vc/<str:sentence>', vc),
    path('find_mode/<str:listofnum>', find_mode),

]
```

Output:

2 occurred 3 times

## Module-2: Django Templates and Models

### Template example program:

#### Views.py

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

```
def template_test(request):
    t=Template("""
        <html>
        <body>
        {% if attending %}
            <h1>Welcome {{ participant.name|upper }},{{ participant.dept }}
to FDP {{ fdp_name }}
            on {{ fdpdate|date:"F j,Y" }} </h1>
            {% if atd_per > 80 %}
                <h2> Very Good </h2>
            {% elif atd_per > 60 %}
                <h2> Good </h2>
            {% else %}
                <h2> Not satisfactory </h2>
            {% endif %}
            Your phone no is {{ participant.pno }}
        <h1>List of Topics</h1>
        <ul>
        {% for topic in topics %}
            <li>{{ forloop.revcounter0 }}: {{ topic }}</li>
        {% endfor %}
        {% else %}
            <h1>Thank you </h1>
        {% endif %}

        </body>
        </html>
    """)
```

```

        """)
    fdp_name="Programming with Julia"
    atd_per=66
    participant={"name":"Chetan","pno":"9900923050","dept":"AIML"}
    topics=["Models","Views","Templates","AJAX","NonHTML"]
    c=Context({"fdp_name":
fdp_name,"topics":topics,"participant":participant,"attending":True,"atd_per":
atd_per,"fdpdate":date(2024,4,9)})
    return HttpResponse(t.render(c))

```

## 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

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),

]

```

## Output:

# Welcome CHETAN,AIML to FDP Programming with Julia on April 9,2024

**Good**

Your phone no is 9900923050

## List of Topics

- 4: Models
- 3: Views
- 2: Templates
- 1: AJAX
- 0: NonHTML

**Develop a Django app that displays list of subject codes and subject names of any semester in tabular format. Even rows should have a light green background color and subject names should be in all caps**

### **Views.py**

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

def list_of_subjects(request):
    s1={"scode":"21CS51","sname":"cn"}
    s2={"scode":"21CS52","sname":"ATc"}
    s3={"scode":"21CS53","sname":"DbMS"}
    s4={"scode":"21AI54","sname":"PAI"}
    l=list()
    l=[s1,s2,s3,s4]
    return render(request,'list_of_subjects.html',{'l':l})
```

### **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
```

```
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),

]
```

### Template file: list\_of\_subjects.html

```
<html>
<body>
    <table border>
        <tr>
            <th>Subject Code</th>
            <th>Subject Name</th>
        </tr>
        {% for subject in l %}
        {% if forloop.counter|divisibleby:"2" %}
        <tr>
            <td style="background-color: lightgreen;">{{ subject.scode }}</td>
            <td style="background-color: lightgreen;">{{ subject.sname|upper
        }}</td>
        </tr>
        {% else %}
        <tr>
            <td>{{ subject.scode }}</td>
            <td>{{ subject.sname|upper }}</td>
        </tr>
        {% endif %}
        {% endfor %}
    </table>
</body>
</html>
```

Output:



Subject Code	Subject Name
21CS51	CN
21CS52	ATC
21CS53	DBMS
21AI54	PAI



















