

Django Project

E-Classroom

Tools: -

- 1) Virtual Studio
- 2) Xampp and Command Prompt.

Code:

- 1) Python
- 2) HTML
- 3) MySQL

Steps to create E-classroom Project: -

Step 1: - Install Virtual Studio.

Step 2: - Create a folder in C-drive.

Step 3: - Go to terminal and use command Prompt.

Step 4: - Install once virtual environment if you are doing project for the first time.

Syntax: -

```
pip install virtualenv
```

Step 5: - Create virtual environment.

Syntax:

```
Virtualenv environmentname
```

In my Project i.e.

```
Virtualenv env
```

Step 6: - Activate Virtual Environment.

Syntax: -

```
env\Scripts\activate
```

Deactivate Virtual Environment

Syntax: -

```
env\Scripts\deactivate
```

Step 7: - Install Django in virtual Environment.

Syntax: -

```
pip install Django
```

Step 8: - Create Django Project

Syntax: -

```
django-admin startproject projectname
```

In my Project, i.e.

```
django-admin startproject learningProject
```

Step 9: - Create Application.

Syntax: -

```
python manage.py startapp appname
```

In my Project, i.e.

```
python manage.py startapp learningapp
```

Step 10: - Running Server.

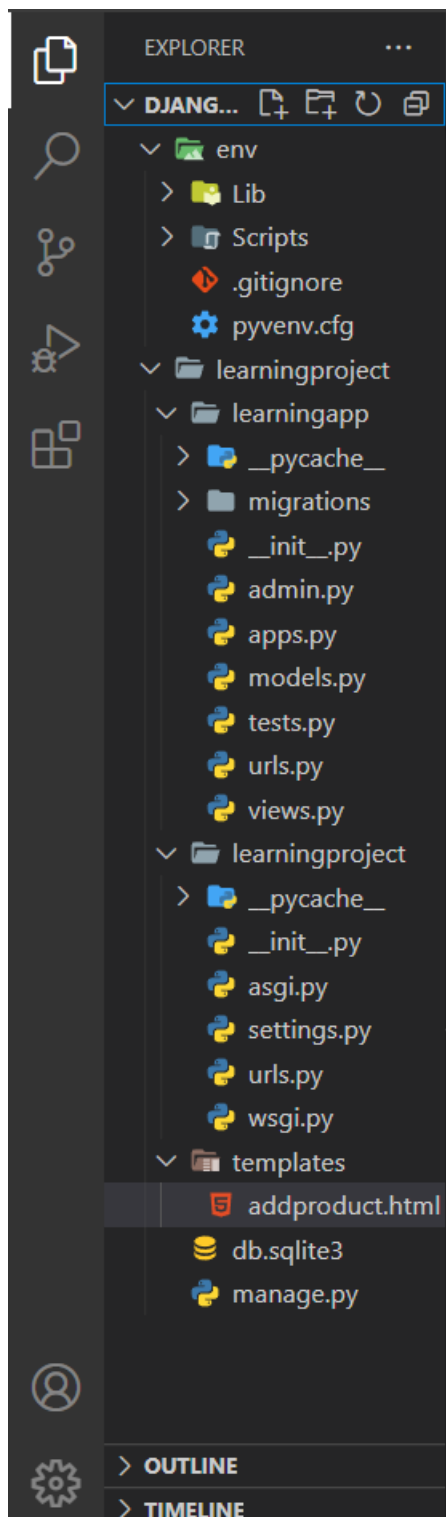
Syntax: -

```
python manage.py runserver
```

To stop server.

```
Ctrl + C
```

- **File Content:** -



Step 11: -

- 1) Create urls.py file in application folder.
- 2) Register urls.py file from application folder into urls.py file in project folder. So that Django server recognised all urls define in urls.py file in application folder.

```
17 from django.contrib import admin
18 from django.urls import path,include
19 from learningapp import views
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('',include('learningapp.urls')),
24 ]
25
```

Step 12: -

- 1) Create templates folder in project folder.
- 2) Go to settings.py in project folder to make some changes.
- 3) Write import os on line no.14

```
13 from pathlib import Path
14 import os
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
```

- 4) Write application name in INSTALLED_APP on line no.41.

```
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     'learningapp'
42 ]
43
```

- 5) Add one code on line no.59.

`os.path.join(BASE_DIR,'templates')`

BASE_DIR: - path which is create in C-drive

```
56 TEMPLATES = [
57     {
58         'BACKEND': 'django.template.backends.django.DjangoTemplates',
59         'DIRS': [os.path.join(BASE_DIR,'templates')],
60         'APP_DIRS': True,
61         'OPTIONS': {
62             'context_processors': [
63                 'django.template.context_processors.debug',
64                 'django.template.context_processors.request',
65                 'django.contrib.auth.context_processors.auth',
66                 'django.contrib.messages.context_processors.messages',
67             ],
68         },
69     ],
70 ]
```

Step 13: -

- 1) Create a database in DBMS

Create database learn1234.

- 2) Make changes in setting.py

Change the default sqlite3 to mysql

```
learningproject > learningproject > settings.py > ...
75 # Database
76 # https://docs.djangoproject.com/en/4.2/ref/settings/#databases
77
78 # DATABASES = {
79 #     'default': {
80 #         'ENGINE': 'django.db.backends.sqlite3',
81 #         'NAME': BASE_DIR / 'db.sqlite3',
82 #     }
83 # }
84
85 DATABASES = {
86     'default': {
87         'ENGINE': 'django.db.backends.mysql',
88         'NAME': 'learn1234',
89         'HOST': 'localhost',
90         'USER': 'root',
91         'PASSWORD': '',
92     }
93 }
```

- 3) Getting an error then do install mysqlclient.

- Stop the server.
- pip install mysqlclient

Step 14: -

- 1) from models.py import models

from Django.db import models

- 2) creating table/model in Django.

```
learningproject > learningapp > models.py > LearningCourses
1 from django.db import models
2
3 # Create your models here.
4
5 class LearningCourses(models.Model):
6     name=models.CharField(max_length=50)
7     cat1=models.CharField(max_length=50)
8     cat2=models.CharField(max_length=50)
9     job_type=models.CharField(max_length=50)
10    price=models.IntegerField()
11    time=models.CharField(max_length=50)
12    status=models.CharField(max_length=50)
```

- 3) After Creating table/model in Django make migrations.

python manage.py makemigrations

4) Run migration.

```
python manage.py migrate
```

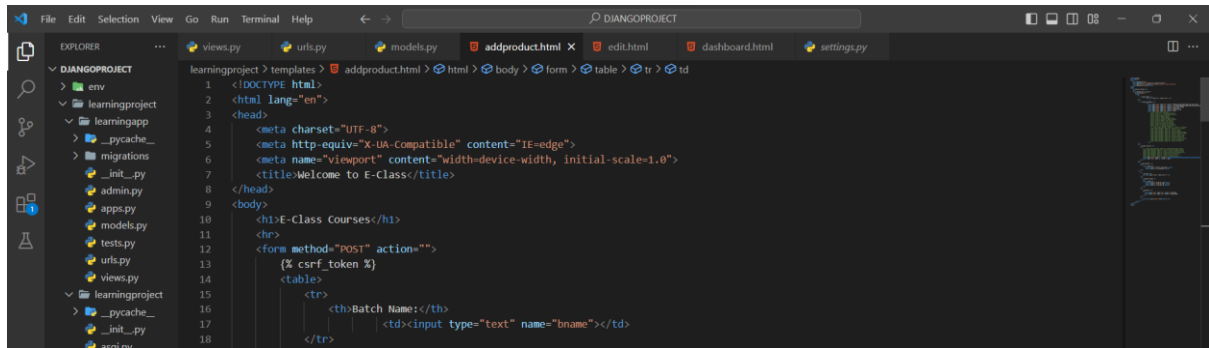
Note: - It will migrate all the pending application.

Step 15: -

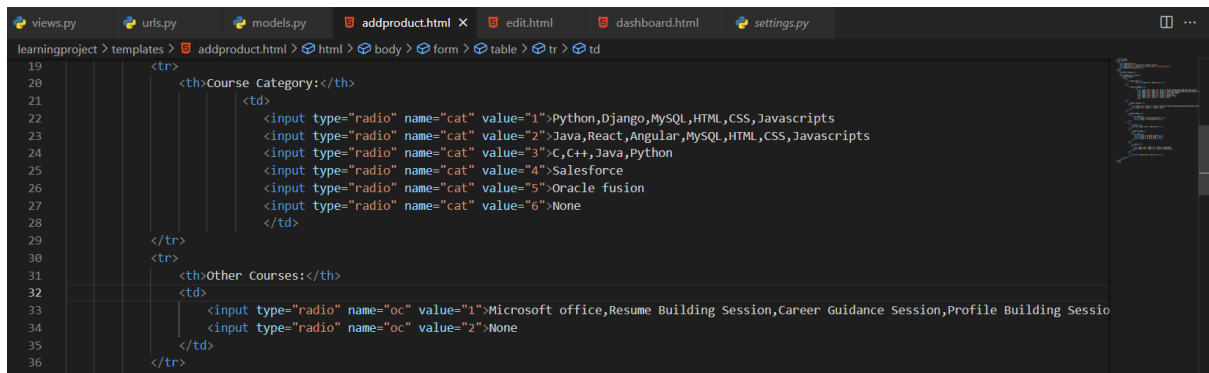
- 1) Create addproduct.html, dashboard.html and edit.html inside templates folder.

- addproduct.html: -

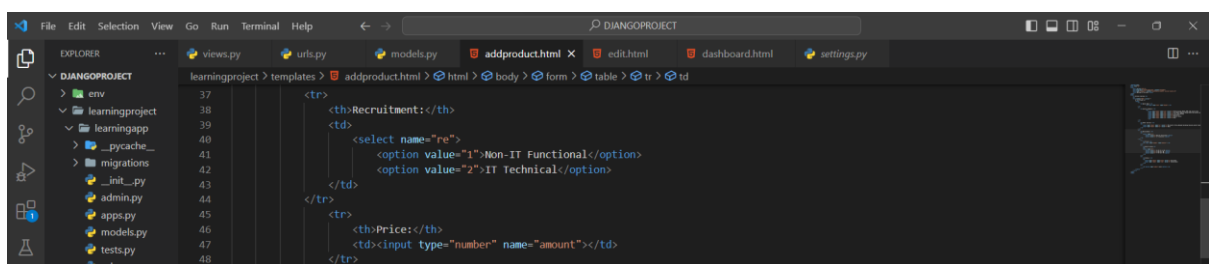
- I. Views.py
- II. Urls.py



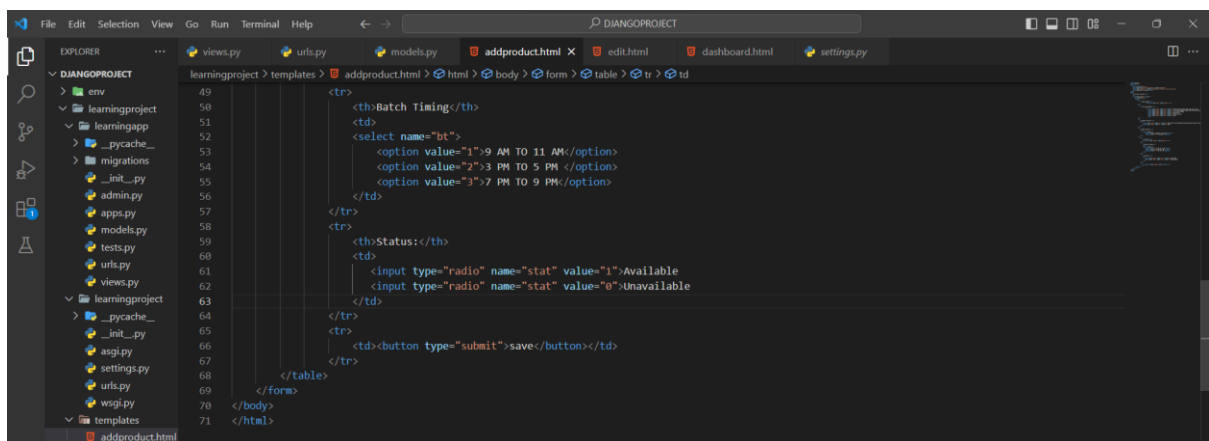
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Welcome to E-Class</title>
8 </head>
9 <body>
10 <h1>E-Class Courses</h1>
11 <hr>
12 <form method="POST" action="">
13 <div>
14 <table>
15 <tr>
16 <th>Batch Name:</th>
17 <td><input type="text" name="bname"></td>
18 </tr>
```



```
19 <tr>
20 <th>Course Category:</th>
21 <td>
22 <input type="radio" name="cat" value="1">Python,Django,MySQL,HTML,CSS,Javascripts
23 <input type="radio" name="cat" value="2">Java,React,Angular,MySQL,HTML,CSS,Javascripts
24 <input type="radio" name="cat" value="3">C++,Java,Python
25 <input type="radio" name="cat" value="4">Salesforce
26 <input type="radio" name="cat" value="5">Oracle fusion
27 <input type="radio" name="cat" value="6">None
28 </td>
29 </tr>
30 <tr>
31 <th>Other Courses:</th>
32 <td>
33 <input type="radio" name="oc" value="1">Microsoft office,Resume Building Session,Career Guidance Session,Profile Building Session
34 <input type="radio" name="oc" value="2">None
35 </td>
36 </tr>
```



```
37 <tr>
38 <th>Recruitment:</th>
39 <td>
40 <select name="re">
41 <option value="1">Non-IT Functional</option>
42 <option value="2">IT Technical</option>
43 </select>
44 </td>
45 </tr>
46 <tr>
47 <th>Price:</th>
48 <td><input type="number" name="amount"></td>
49 </tr>
```



```
49 <tr>
50 <th>Batch Timing:</th>
51 <td>
52 <select name="bt">
53 <option value="1">9 AM TO 11 AM</option>
54 <option value="2">3 PM TO 5 PM</option>
55 <option value="3">7 PM TO 9 PM</option>
56 </select>
57 </td>
58 </tr>
59 <tr>
60 <th>Status:</th>
61 <td>
62 <input type="radio" name="stat" value="1">Available
63 <input type="radio" name="stat" value="0">Unavailable
64 </td>
65 </tr>
66 <tr>
67 <td><button type="submit">save</button></td>
68 </tr>
69 </table>
70 </form>
71 </body>
72 </html>
```

views.py for addproduct: -

```
33 def addproduct(request):
34
35     if request.method=="POST":
36         Batch_Name=request.POST['bname']
37         Course_Category=request.POST['cat']
38         Other_Course=request.POST['oc']
39         Recruitment=request.POST['re']
40         Price=request.POST['amount']
41         Batch_Timing=request.POST['bt']
42         Status=request.POST['stat']
43
44         p=LearningCourses.objects.create(name=Batch_Name,cat1=Course_Category,cat2=Other_Course,job_type=Recruitment,price=Price,time=Batch_Timing,s
45         p.save()
46         return redirect('/dash')
47
48     else:
49         return render(request,'addproduct.html')
50
```

urls.py for addproduct: -

```
path('addproduct',views.addproduct),
```


- dashboard.html: -

In dashboard also added edit, delete, filter, sort and multifilter.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>welcome to E-class Dashboard</title>
8 </head>
9 <body>
10     <h1>Coordinator Dashboard</h1>
11     <p>
12         <a href="/addproduct">Add Batch</a>
13     </p>
14     <table border="1" align="center">
15     <tr>
16         <th>Sr.No</th>
17         <th>Batch Name</th>
18         <th>Course Category</th>
19         <th>Extra Course Category</th>
20         <th>Recruitment</th>
21         <th>Prices</th>
22         <th>Batch Time</th>
23         <th>Status</th>
24         <th>Edit</th>
25         <th>Delete</th>
26     </tr>
27     {% for x in data %}

```

```

28 <tr>
29     <td>{{x.id}}</td>
30     <td>{{x.name}}</td>
31     {% if x.cat1 == '1' %}
32     <td>Python,Django,MySQL,HTML,CSS,Javascript</td>
33     {% elif x.cat1 == '2' %}
34     <td>Java,React,Angular,MySQL,HTML,CSS,Javascript</td>
35     {% elif x.cat1 == '3' %}
36     <td>C,C++,Java,Python</td>
37     {% elif x.cat1 == '4' %}
38     <td>Salesforce</td>
39     {% elif x.cat1 == '5' %}
40     <td>Oracle fusion</td>
41     {% else %}
42     <td>None</td>
43     {% endif %}
44
45     {% if x.cat2 == '1' %}
46     <td>Microsoft office,Resume Building Session,Career Guidance Session,Profile Building Session,Interview Session,Linux Session,Github Ses
47     {% else %}
48     <td>None</td>
49     {% endif %}
50
51     {% if x.job_type == '1' %}
52     <td>Non-IT Functional</td>
53     {% else %}
54     <td>IT Technical</td>
55     {% endif %}

```

```

56 <td>{{x.price}}</td>
57     {% if x.time == '1' %}
58     <td>9 AM TO 11 AM</td>
59     {% elif x.time == '2' %}
60     <td>3 PM TO 5 PM</td>
61     {% else %}
62     <td>7 PM TO 9 PM</td>
63     {% endif %}
64     {% if x.status == '1' %}
65     <td>Available</td>
66     {% else %}
67     <td>Unavailable</td>
68     {% endif %}
69
70     <td><a href="/edit/{{x.id}}">Edit</a></td>
71     <td><a href="/delete/{{x.id}}">Delete</a></td>
72 </tr>
73 {% endfor %}
74

```

views.py for dashboard: -

```
def dashboard(request):  
    da=LearningCourses.objects.all()  
    content={}  
    content['data']=da  
    return render(request,'dashboard.html',content)
```

urls.py for dashboard: -

```
path('dash',views.dashboard),
```

I. Edit: -

First create edit.html in the template folder copy and past addproduct html code inside edit.html file.

The screenshot shows a VS Code editor with a Django project named 'DIANGOPROJECT'. The Explorer sidebar on the left displays the project structure, including files like 'views.py', 'urls.py', 'models.py', 'addproduct.html', 'edit.html', 'dashboard.html', and 'settings.py'. The main editor area shows the content of 'edit.html', which is an HTML template for editing courses. The template includes a Django form for 'Edit Courses' and a table for listing courses. The table has columns for 'Batch Name' and 'Name'. The form includes a text input for the name, a 'Batch Name' label, and a 'Save' button. The table is currently empty.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Welcome to E-Class</title>
8 </head>
9 <body>
10     <h1>Edit Courses Section</h1>
11     <hr>
12     {% for x in data %}
13     <form method="post" action="">
14         {% csrf_token %}
15         <table>
16             <tr>
17                 <th>Batch Name:</th>
18                 <td><input type="text" name="bname" value="{{x.name}}"></td>
19             </tr>

```

The screenshot shows a VS Code editor window with a project named 'DIANGOPROJECT'. The Explorer sidebar on the left displays the project structure, including folders like 'env', 'learningproject', 'learningapp', and 'templates', and files like 'urls.py', 'models.py', 'addproduct.html', 'edit.html', 'dashboard.html', and 'settings.py'. The 'edit.html' file is currently open in the editor, showing a Django template for a form. The template includes a form with a 'category' field and a 'cat' field, both with radio button options. The 'cat' field has six options: 'Python', 'Django', 'MySQL', 'HTML', 'CSS', and 'JavaScripts'. The 'category' field has three options: 'Python', 'Django', and 'MySQL'. The form is rendered using Django's {% csrf_token %} and {{ form.as_table }} template tags.

```
File Edit Selection View Go Run Terminal Help DIANGOPROJECT
EXPLORER
  DIANGOPROJECT
    env
    learningproject
      learningapp
        __pycache__
        migrations
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
        views.py
      learningproject
        __pycache__
        __init__.py
        asgi.py
        settings.py
        urls.py
        wsgi.py
      templates
        addproduct.html
        dashboard.html
        edit.html
        manage.py
    views.py
    urls.py
    models.py
    addproduct.html
    edit.html
    dashboard.html
    settings.py

learningproject > templates > edit.html > html > body > form > table > tr > td
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113

<tr>
  <th>Recruitment:</th>
  <td>
    <select name="re">
      {% if x.job_type == '1' %}
      <option value="1">Non-IT Functional</option>
      {% else %}
      <option value="2">IT Technical</option>
      {% endif %}
      <option value="1">Non-IT Functional</option>
      <option value="2">IT Technical</option>
    </select>
  </td>
</tr>
<tr>
  <th>Price:</th>
  <td><input type="number" name="amount" value="{{x.price}}"></td>
</tr>
<tr>
  <th>Batch Timing</th>
  <td>
    <select name="bt">
      {% if x.time == '1' %}
      <option value="1">9 AM TO 11 AM</option>
      {% elif x.time == '2' %}
      <option value="2">3 PM TO 5 PM</option>
      {% else %}
      <option value="3">7 PM TO 9 PM</option>
      {% endif %}
      <option value="1">9 AM TO 11 AM</option>
      <option value="2">3 PM TO 5 PM</option>
      <option value="3">7 PM TO 9 PM</option>
    </select>
  </td>
</tr>
```

```
File Edit Selection View Go Run Terminal Help DIANGOPROJECT
EXPLORER
  DIANGOPROJECT
    env
    learningproject
      learningapp
        __pycache__
        migrations
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
        views.py
      learningproject
        __pycache__
        __init__.py
        asgi.py
        settings.py
        urls.py
        wsgi.py
      templates
        addproduct.html
        dashboard.html
        edit.html
        manage.py
    views.py
    urls.py
    models.py
    addproduct.html
    edit.html
    dashboard.html
    settings.py

learningproject > templates > edit.html > html > body > form > table > tr > td
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133

<tr>
  <th>Status:</th>
  <td>
    {% if x.status == '1' %}
    <input type="radio" name="stat" value="1" checked>Available
    <input type="radio" name="stat" value="0">Unavailable
    {% else %}
    <input type="radio" name="stat" value="1">Available
    <input type="radio" name="stat" value="0" checked>Unavailable
    {% endif %}
  </td>
</tr>
<tr>
  <td><button type="submit">save</button></td>
</tr>
</table>
</form>
{% endfor %}
</body>
</html>
```

views.py for edit: -

Syntax: -

p=LearningCourses.objects.filter(id=rid)

p.update(name=Batch_Name,cat1=Course_Category,cat2=Other_Course,job_type=Recruitment,price=Price,time=Batch_Timing,status=Status)

```
def edit(request,rid):
    if request.method=="POST":
        Batch_Name=request.POST['bname']
        Course_Category=request.POST['cat']
        Other_Course=request.POST['oc']
        Recruitment=request.POST['re']
        Price=request.POST['amount']
        Batch_Timing=request.POST['bt']
        Status=request.POST['stat']
        p=LearningCourses.objects.filter(id=rid)
        p.update(name=Batch_Name,cat1=Course_Category,cat2=Other_Course,job_type=Recruitment,price=Price,time=Batch_Timing,status=Status)
        return redirect('/dash')
    else:
        p=LearningCourses.objects.filter(id=rid)
        content={}
        content['data']=p
        return render(request,'edit.html',content)
```

urls.py for edit: -

```
learningproject > learningapp > urls.py / ...
1  from django.urls import path,include
2  from learningapp import views
3
4  urlpatterns = [
5      path('edit/<rid>',views.edit),
```

II. Delete: -

Do not create any html file for delete.

views.py for delete: -

Syntax: -

```
p=LearningCourses.objects.filter(id=rid)
```

```
p.delete()
```

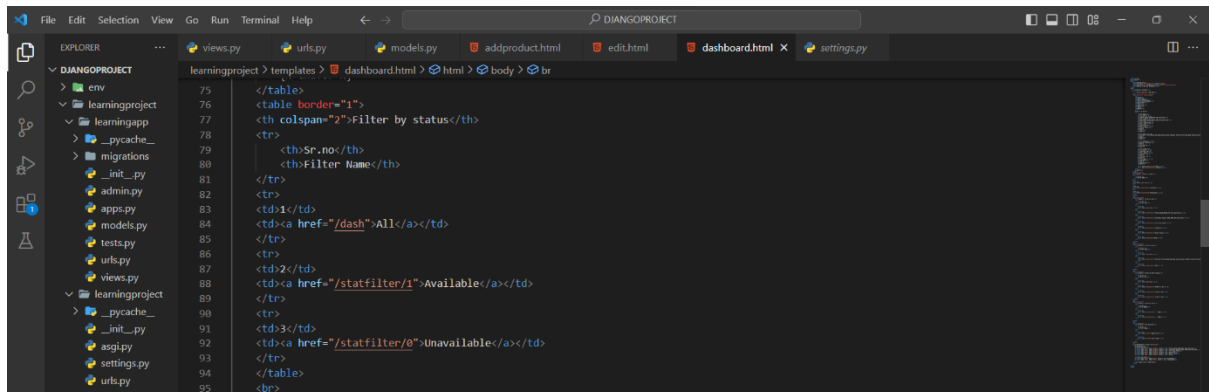
```
def delete(request,rid):  
    p=LearningCourses.objects.filter(id=rid)  
    p.delete()  
    return redirect('/dash')
```

urls.py for delete: -

```
path('delete/<rid>',views.delete),
```

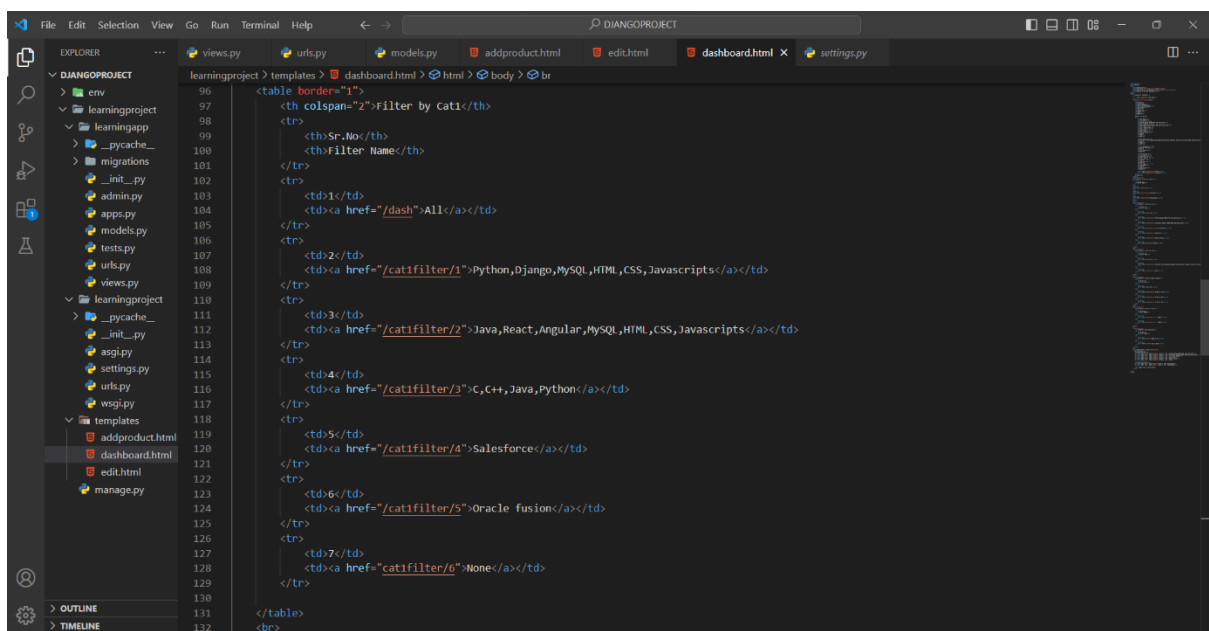
III. Filters for dashboard: -

Do not create html file for filter. Assign it to dashboard.html.



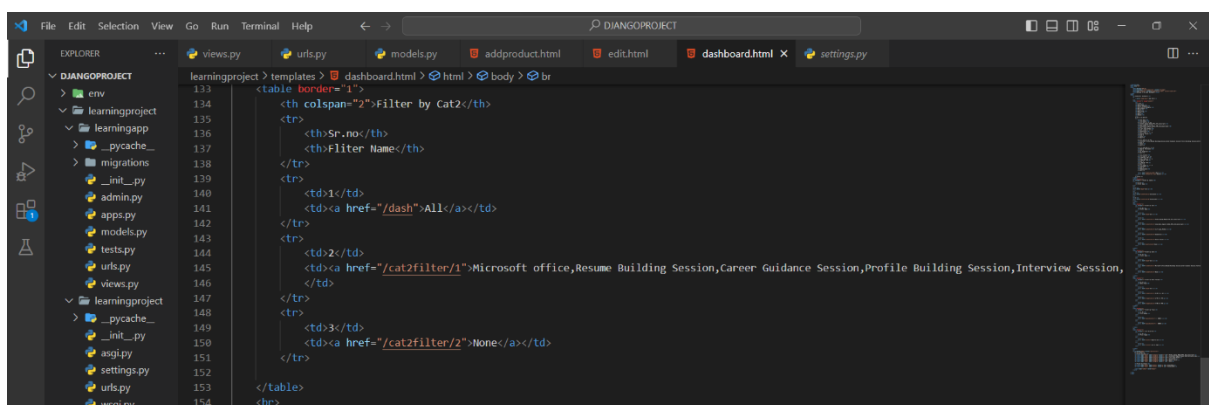
This screenshot shows the first implementation of a filter in the `dashboard.html` file. The file explorer on the left shows the project structure. The main editor displays the `dashboard.html` file with the following HTML code:

```
75 </table>
76 <table border="1">
77 <th colspan="2">Filter by status</th>
78 <tr>
79 <th>Sr.no</th>
80 <th>Filter Name</th>
81 </tr>
82 <tr>
83 <td>1</td>
84 <td><a href="/dash">All</a></td>
85 </tr>
86 <tr>
87 <td>2</td>
88 <td><a href="/statfilter/1">Available</a></td>
89 </tr>
90 <tr>
91 <td>3</td>
92 <td><a href="/statfilter/0">Unavailable</a></td>
93 </tr>
94 </table>
95 <br>
```



This screenshot shows the second implementation of a filter in the `dashboard.html` file. The file explorer on the left shows the project structure. The main editor displays the `dashboard.html` file with the following HTML code:

```
96 <table border="1">
97 <th colspan="2">Filter by cat1</th>
98 <tr>
99 <th>Sr.No</th>
100 <th>Filter Name</th>
101 </tr>
102 <tr>
103 <td>1</td>
104 <td><a href="/dash">All</a></td>
105 </tr>
106 <tr>
107 <td>2</td>
108 <td><a href="/catfilter/1">Python,Django,MySQL,HTML,CSS,Javascript</a></td>
109 </tr>
110 <tr>
111 <td>3</td>
112 <td><a href="/catfilter/2">Java,React,Angular,MySQL,HTML,CSS,Javascript</a></td>
113 </tr>
114 <tr>
115 <td>4</td>
116 <td><a href="/catfilter/3">C,C++,Java,Python</a></td>
117 </tr>
118 <tr>
119 <td>5</td>
120 <td><a href="/catfilter/4">Salesforce</a></td>
121 </tr>
122 <tr>
123 <td>6</td>
124 <td><a href="/catfilter/5">Oracle fusion</a></td>
125 </tr>
126 <tr>
127 <td>7</td>
128 <td><a href="/catfilter/6">None</a></td>
129 </tr>
130 </table>
131 <br>
132
```



This screenshot shows the third implementation of a filter in the `dashboard.html` file. The file explorer on the left shows the project structure. The main editor displays the `dashboard.html` file with the following HTML code:

```
133 <table border="1">
134 <th colspan="2">Filter by cat2</th>
135 <tr>
136 <th>Sr.no</th>
137 <th>Filter Name</th>
138 </tr>
139 <tr>
140 <td>1</td>
141 <td><a href="/dash">All</a></td>
142 </tr>
143 <tr>
144 <td>2</td>
145 <td><a href="/cat2filter/1">Microsoft office,Resume Building Session,Career Guidance Session,Profile Building Session,Interview Session,
146 </td>
147 </tr>
148 <tr>
149 <td>3</td>
150 <td><a href="/cat2filter/2">None</a></td>
151 </tr>
152 </table>
153 <br>
154
```

```

155 <table border="1">
156 <th colspan="2">Filter by Batch Timings</th>
157 <tr>
158 <th>Sr.no</th>
159 <th>Batch Time</th>
160 </tr>
161 <tr>
162 <td>1</td>
163 <td><a href="/dash">All</a></td>
164 </tr>
165 <tr>
166 <td>2</td>
167 <td><a href="/timefilter/1">9 AM TO 11 AM</a></td>
168 </tr>
169 <tr>
170 <td>3</td>
171 <td><a href="/timefilter/2">3 PM TO 5 PM</a></td>
172 </tr>
173 <tr>
174 <td>4</td>
175 <td><a href="/timefilter/3">7 PM TO 9 PM</a></td>
176 </tr>
177 </table>
178 <br>

```

```

179 <table border="1">
180 <th colspan="2">Filter by Price</th>
181 <tr>
182 <th>Sr.No</th>
183 <th>Price Range</th>
184 </tr>
185 <tr>
186 <td>1</td>
187 <td><a href="/pricefilter/1"> > 30000 </a></td>
188 </tr>
189 <tr>
190 <td>2</td>
191 <td><a href="/pricefilter/2"> < 30000</a></td>
192 </tr>
193 </table>
194 <br>
195

```

views.py for Filter: -

```

57 # filter
58
59 def statfilter(request,sv):
60     p=LearningCourses.objects.filter(status=sv)
61     content={}
62     content['data']=p
63     return render(request,'dashboard.html',content)
64
65 def cat1filter(request,cv1):
66     p=LearningCourses.objects.filter(cat1=cv1)
67     content={}
68     content['data']=p
69     return render(request,'dashboard.html',content)
70
71 def cat2filter(request,cv2):
72     p=LearningCourses.objects.filter(cat2=cv2)
73     content={}
74     content['data']=p
75     return render(request,'dashboard.html',content)
76
77 def timefilter(request,tf):
78     p=LearningCourses.objects.filter(time=tf)
79     content={}
80     content['data']=p
81     return render(request,'dashboard.html',content)
82
83 def pricefilter(request,pf):
84     if pf=='1' :
85         p=LearningCourses.objects.filter(price__gt=30000)
86     else:
87         p=LearningCourses.objects.filter(price__lt=30000)
88     content={}
89     content['data']=p
90     return render(request,'dashboard.html',content)
91

```

url.py for Filter: -

```

path('statfilter/<sv>',views.statfilter),
path('cat1filter/<cv1>',views.cat1filter),
path('cat2filter/<cv2>',views.cat2filter),
path('timefilter/<tf>',views.timefilter),
path('pricefilter/<pf>',views.pricefilter),

```


IV. Sort for Dashboard: -

Do not create html file for filter. Assign it to dashboard.html

```
<table border="1">
  <th colspan="2">sort by price</th>
  <tr>
    <th>Sr.No</th>
    <th>Price range</th>
  </tr>
  <tr>
    <td>1</td>
    <td><a href="/sort/1">High to Low</a></td>
  </tr>
  <tr>
    <td>2</td>
    <td><a href="/sort/0">Low to High</a></td>
  </tr>
</table>
<br>
```

views.py for Sort: -

Syntax: -

p=LearningCourses.objects.order_by('price/-price')

where,

price is ascending order.

-price is descending order.

```
def sortfilter(request,s):
    if s=='0':
        p=LearningCourses.objects.order_by('price')
    else:
        p=LearningCourses.objects.order_by('-price')
    content={}
    content['data']=p
    return render(request,'dashboard.html',content)
```

urls.py for sort: -

```
path('sort/<s>',views.sortfilter),
```

V. Multifilter for dashboard: -

Do not create html file for filter. Assign it to dashboard.html

```
213 <form method="POST" action="/multifilter">
214     {% csrf_token %}
215     <h2>Filter by Cat1</h2>
216     <p><input type="radio" name="category" value="1" id="">Python,Django,MySQL,HTML,CSS,Javascripts</p>
217     <p><input type="radio" name="category" value="2" id="">Java,React,Angular,MySQL,HTML,CSS,Javascripts</p>
218     <p><input type="radio" name="category" value="3" id="">C,C++,Java,Python</p>
219     <p><input type="radio" name="category" value="4" id="">Salesforce</p>
220     <p><input type="radio" name="category" value="5" id="">Oracle fusion</p>
221     <p><input type="radio" name="category" value="6" id="">None</p>
222
223     <h2>Filter by Status</h2>
224     <p><input type="radio" name="status" value="1" id="">Available</p>
225     <p><input type="radio" name="status" value="0" id="">Unavailable</p>
226
227     <input type="submit" value="Apply">
228 </form>
229
230 </body>
231 </html>
```

Views.py for multifilter: -

In views, import Q from django.db.models

Syntax: -

from django.db.models import Q

q1 = Q(condition 1)

q2 = Q(condition 2)

queryset = Modelname.objects.filter(q1 & q2)

```
def multifilter(request):
    if request.method=="POST":
        sv=request.POST['status']
        cv1=request.POST['category']
        q1=Q(cat1=cv1)
        q2=Q(status=sv)
        p=LearningCourses.objects.filter(q1 & q2)
        content={}
        content['data']=p
        return render(request,'dashboard.html',content)
```

urls.py for multifilter: -

```
path('multifilter',views.multifilter),
]
```

