DAY 1: Implementation of CRUD functionality of a model '*student*'.

Steps

1. Create a new project
2. Create the models
3. Create a form
4. Add templates folder
5. Create the views

1. Create new project

Create a new django project (preferably in a virtual environment) using django-admin startproject weblabproject
proceed to create an app *students* using python manage.py startapp day1.
Register the app in your *settings.py* file.

2. Create the models

We are going to have a model, *Student* model. Register the app in your *settings.py* file. Open your *students* app's *models.py* file. This is where we will create our models, using the code below:

```
class Student(models.Model):
    name = models.CharField(max_length= 50)
    present_address = models. CharField(max_length= 150)
    registration_date = models.DateTimeField(auto_now_add = True)
    reg_no = models.CharField(max_length=100, unique=True)



    def __str__(self):
        return f'{self.name} reg {self.reg_no}'
```

Navigate to *admin.py* and register your model as follows:

```
from students.models import Student, Stream
# Register your model here.
admin.site.register(Student)
```

3.Create a form

In your app's folder, create a new file and name it *forms.py*.
Add the following code to your file:

```
from django import forms
from .models import Student


class CreateStudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = "__all__"
```

4.Create a folder for our templates

To make it easier to access our templates, we shall create a single folder named *templates* that will contain all of our templates.
Navigate to your *settings.py* , locate your *Templates* array and replace it with the following:

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

Navigate to the root folder of your project and create a folder named *templates*

5. Create Views

Navigate to your *views.py* file and we'll start writing code.
We are going to use function-based views for this.

First, we create a view to create a student instance (object). This will also be the view for our landing page. To do this, we need to import the form we created in *forms.py*.
Include this import *from .forms import CreateStudentForm*
Here now goes the view:

```
def CreateStudent(request):
    if request.method == "POST":
        form = CreateStudentForm(request.POST)
        if form.is_valid():
            form.save()
            form = CreateStudentForm()
    else:
        form = CreateStudentForm()

    students = Student.objects.all().order_by("-id")
    context = {'form':form, 'students':students}
    return render(request, index.html', context)
```

We want to display our students starting with the last added, hence *order_by("-id")* in
students = Student.objects.all().order_by("-id")

Create a file *index.html* in the *templates* we created in the previous step.
This template will show a form for creating new students and also list the existing students in a table.
For each student instance or record, you will be able to view or delete it.
I have added bootstrap to it to make it look better.
Your *index.html* should be like this:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
        integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
        integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
        crossorigin="anonymous"></script>
    <title> Home</title>
</head>

<body>
  <div>
      <form method="post">
          {% csrf_token %}
          {{form.as_p}}
          <button type="submit">Submit</button>
      </form>
  </div>
        <!-- list of students  -->
        <div>
            <h4 class="text-center mb-3 mt-3">Students List</h4>
            <div class="table-responsive-sm">
                <table class="table table-hover">
                    <thead class="thead-light">
                        <tr>
                            <th scope="col">Name</th>
                            <th scope="col">Reg No.</th>
                            <th scope="col"> Present Address</th>
                            <th scope="col">Registration Date</th>
                            <th scope="col">View</th>
                            <th scope="col">Delete</th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for student in students %}
                        <tr>
                            <td class="text-capitalize">{{student.name}}</td>
```

```
                    <td>{{student.reg_no}}</td>
                    <td class="text-capitalize">{{student.present_address}}</td>
                    <td>{{student.registration_date}}</td>
                    <td>
                        <a role="button" class="btn btn-sm btn-outline-success"
                            href="{% url 'std_detail' reg=student.reg_no %}">View</a>
                    </td>
                    <td>
                        <a role="button" class="btn btn-sm btn-outline-danger" href="{% url
'delete' id=student.id %}">Delete</a>
                        <!-- <button class="btn btn-sm btn-outline-
danger">Delete</button></td> -->
                    </tr>
                    {% empty %}
                    <span class="text-center">No students to show</span>
                    {% endfor %}
                </tbody>
            </table>
        </div>
    </div>
</body>
```

As mentioned above, we want to view a specific student. To achieve this, we need to create another view. We are going to fetch a student from the database by using their *reg_no* as used in our *Student* model.

This same view will be used to update a student instance, hence passing the student instance will pre-populate the form.

```
def GetStudent(request, **kwargs):
    reg = kwargs.get('reg')
    student = get_object_or_404(Student, reg_no=reg)

    if request.method == "POST":
        form = CreateStudentForm(request.POST, instance=student)
        if form.is_valid():
            form.save()
    else:
        form = CreateStudentForm(instance=student)

    context = {'student':student, 'form': form}
    return render(request, 'std_detail.html', context )
```

Our *std_detail.html* will look like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>

    <title>Student Detail | {{student.reg_no}}</title>
  </head>
  <body>
    <div class="big-box container-fluid d-flex justify-content-center align-items-center">
      <div class="container mt-4 mb-5">
        <h4 class="text-center text-capitalize">{{student.name}}'s Profile</h4>
        <form class="mt-3" method="post">
          {% csrf_token %}
            <div class="form-group">
              <div class="row">
                <div class="col">
                  <label>Name:</label>
                    {{form.name}}
                </div>
              </div>
            </div>
            <div class="form-group">
              <div class="row">
                <div class="col">
                  <label>Reg No.</label>
                    {{form.reg_no}}
                </div>

              </div>
            </div>
            <div class="form-group">
              <label>Description:</label>
                {{form.description}}
            </div>
            <div class="form-group">
              <div class="row">
                <div class="col d-flex justify-content-center">
```

```
                    <button class="btn btn-warning btn-block text-white w-
50">Update</button>
                  </div>
                  <div class="col d-flex justify-content-center">
                    <a href="{% url 'delete' reg=student.reg_no %}" role="button" class="btn
btn-danger btn-block text-white w-50">Delete</a>
                  </div>
                </div>
              </div>
            </form>
          </div>
        </div>
        <div class="container d-flex justify-content-center">
          <a href="{% url 'create_student' %}" role="button" class="btn btn-outline-primary btn-
block w-25">All Students</a>
        </div>
      </body>
    </html>
```

To delete a student, we will also use their *reg_no* to fetch them, then proceed to delete them.
Here is the view:

```
def DeleteStudent(request, **kwargs):
reg = kwargs.get('reg')
student = Student.objects.get(reg_no=reg)
student.delete()
return redirect(CreateStudent)
```

We now need to define our urls so that we can view and access what we have done.
Create a *urls.py* file in your app and include the following:

```
from django.urls import path
from .views import CreateStudent, GetStudent, DeleteStudent


urlpatterns = [
    path('', CreateStudent, name='create_student'),
    path('std_detail/<str:reg>/', GetStudent, name='std_detail'),
    path('delete/<str:reg>', DeleteStudent, name='delete'),
]
```

Head to your project's *url.py* and add the following:

```
from django.urls import path,include
urlpatterns = [
    path('', include("day1.urls")),
    path('admin/', admin.site.urls),
]
```