

# STUDENT MANAGEMENT SYSTEM

## 1.set up Django

- Virtual environment
- Create virtual env → `Python -m venv myenv`
- Activate virtual environment --> `Myenv\Scripts\activate`
- Install Django
- Installation --> `Pip install Django`
- Start project → `Django-admin startproject student_management`
- Change/go to project folder → `cd student_management`
- Start app → `python manage.py startapp student`

## 2. database

- ```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'student_management_db',  
        'USER': 'your_mysql_username',  
        'PASSWORD': 'your_mysql_password',  
        'HOST': 'localhost',  
        'PORT': '3306',  
    }  
}
```
- Install mysql → `pip install mysql`
- Connect app to settings
- Install app  
    ‘student’,

## 3.Models

`from django.db import models`

`class Student(models.Model):`

`roll_number = models.CharField(max_length=10)`

`name = models.CharField(max_length=100)`

`age = models.PositiveIntegerField()`

`degree = models.CharField(max_length=50)`

```
passed_out_year = models.CharField(max_length=4)
emailid = models.EmailField(unique=True)
mobile = models.CharField(max_length=10)
```

```
def __str__(self):
    return self.name
```

#### 4. create forms

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

```
class StudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = '__all__'
```

#### 5.views

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Student
from .forms import StudentForm
```

# Create a student record

```
def add_student(request):
    if request.method == 'POST':
        form = StudentForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('student_list')
```

```

else:

    form = StudentForm()

    return render(request, 'student/student_form.html', {'form': form})


# List all students
def student_list(request):

    students = Student.objects.all()

    return render(request, 'student/student_list.html', {'students': students})


# Edit a student record
def edit_student(request, pk):

    student = get_object_or_404(Student, pk=pk)

    if request.method == 'POST':

        form = StudentForm(request.POST, instance=student)

        if form.is_valid():

            form.save()

            return redirect('student_list')

    else:

        form = StudentForm(instance=student)

    return render(request, 'student/student_form.html', {'form': form})


# Delete a student record
def delete_student(request, pk):

    student = get_object_or_404(Student, pk=pk)

    student.delete()

    return redirect('student_list')

```

## 6.urls

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

```
from . import views
```

```
urlpatterns = [  
    path('create/', views.add_student, name='add_student'),  
    path('read/', views.student_list, name='student_list'),  
    path('edit/<int:pk>/', views.edit_student, name='edit_student'),  
    path('delete/<int:pk>/', views.delete_student, name='delete_student'),  
]
```

## 7.setting up templates

- Student/student\_list.html

```
<h1>Student List</h1>
```

```
<a href="{% url 'add_student' %}">Add New Student</a>
```

```
<table>
```

```
<tr>
```

```
<th>Name</th>
```

```
<th>Age</th>
```

```
<th>Degree</th>
```

```
<th>Email</th>
```

```
<th>Actions</th>
```

```
</tr>
```

```
{% for student in students %}
```

```
<tr>
```

```
<td>{{ student.name }}</td>
```

```
<td>{{ student.age }}</td>
```

```
<td>{{ student.degree }}</td>
```

```

<td>{{ student.emailid }}</td>
<td>
    <a href="{% url 'edit_student' student.pk %}">Edit</a> |
    <a href="{% url 'delete_student' student.pk %}">Delete</a>
</td>
</tr>
{% endfor %}
</table>

```

- Templates/student/Form.html
 

```

<h1>{% if form.instance.pk %}Edit{% else %}Create{% endif %}
Student</h1>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Submit</button>
</form>

```

## 8. Register the model to admin

```

from django.contrib import admin
from .models import Student

```

```

admin.site.register(Student)

```

## 9.run migrations

- Python manage.py makemigrations
- Python manage.py migrate

## 10. run server

- Python manage.py runserver