MYSORE UNIVERSITY SCHOOL OF ENGINEERING

Manasagangotri campus, Mysuru-570006

(Approved by AICTE, New Delhi)

UNIVERSITY OF MYSORE

REPORT FOR FULL STACK DEVELOPMENT

ON DJANGO

"STUDENT_MANAGEMENT_SYSTEM"

Submitted By

GUNARI NISITHA SRI

21SECD06

7TH SEMESTER

DEPARTMENT OF COMPUTER SCIENCE AND

Under Faculty Incharge

1) Dr. M.S. Govinde Gowda, Director
2) MR. Karthik M.N
   Asst. Professor
   Dept. Of CS&D
   MUSE

DJANGO PROJECT-ASSESSENT:

USN: 21SECD06

QUESTION:

Develop a Django-based Student Management System with the following features:

- The system should allow the admin to **add, update, and** delete student records using Django Admin.
- Each student should have fields like **name, roll number, course, email**, and **date of birth.**
- Use Django's generic **ListView** to display all students and **DetailView** to show a particular student's details.
- Implement URL configuration that allows navigation between student list and detail views using **reverse_lazy().**

Step 1: Set Up Virtual Environment and Install Django using the command prompt:

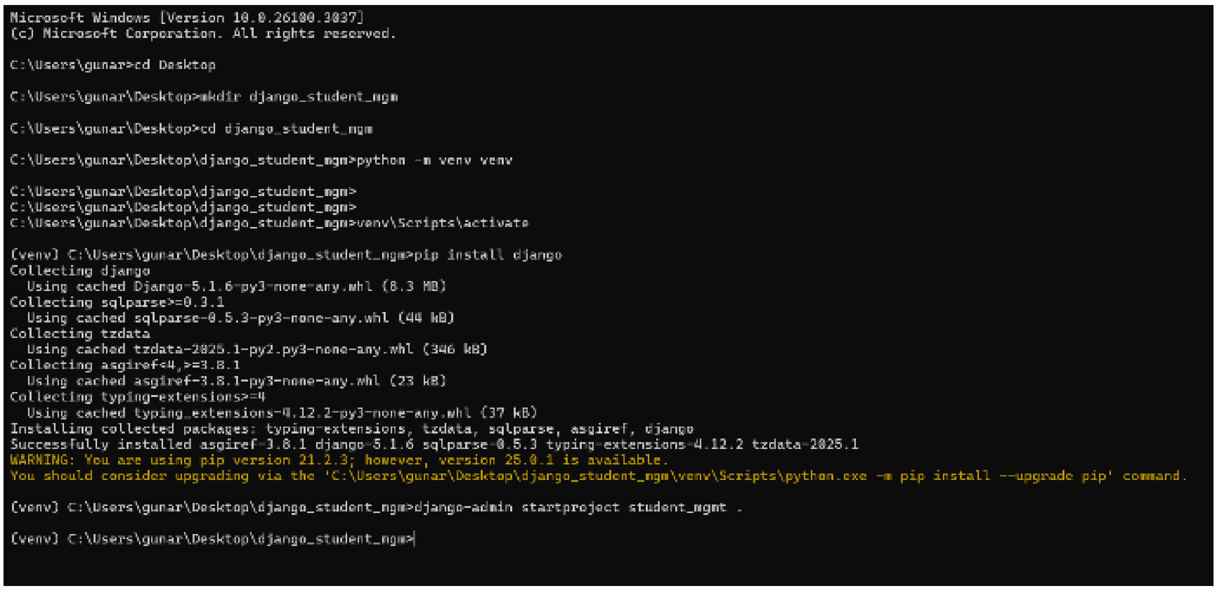
- Install Python.
- Create a Virtual Environment using COMMAND PROMPT:

Inside the command prompt, give the following codes:

```
 >>cd Desktop  # navigate to desktop
 >>mkdir django_student_mgm  #Create the project folder
    cd django_student_mgm
>>python -m venv venv    # Create a virtual environment
>>venv\Scripts\activate   # Activate the virtual environment
>> pip install Django   # Install Django
```

Refer the following image where in the command prompt we have navigated to the desktop, made a directory named django_student_mgm, then we navigate to the project and create a virtual environment and later activated using scripts.

After the whole process, we reinstall django inside the virtual environment.



```
Microsoft Windows [Version 10.0.26100.3037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\gunar>cd Desktop

C:\Users\gunar\Desktop>mkdir django_student_mgm

C:\Users\gunar\Desktop>cd django_student_mgm

C:\Users\gunar\Desktop\django_student_mgm>python -m venv venv

C:\Users\gunar\Desktop\django_student_mgm>
C:\Users\gunar\Desktop\django_student_mgm>
C:\Users\gunar\Desktop\django_student_mgm>venv\Scripts\activate

(venv) C:\Users\gunar\Desktop\django_student_mgm>pip install django
Collecting django
  Using cached Django-5.1.6-py3-none-any.whl (8.3 MB)
Collecting sqlparse>=0.3.1
  Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Collecting tzdata
  Using cached tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Collecting asgiref<4,>=3.8.1
  Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Collecting typing-extensions>=4
  Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.1.6 sqlparse-0.5.3 typing-extensions-4.12.2 tzdata-2025.1
WARNING: You are using pip version 21.2.3; however, version 25.0.1 is available.
You should consider upgrading via the 'C:\Users\gunar\Desktop\django_student_mgm\venv\Scripts\python.exe -m pip install --upgrade pip' command.

(venv) C:\Users\gunar\Desktop\django_student_mgm>django-admin startproject student_mgmt .

(venv) C:\Users\gunar\Desktop\django_student_mgm>
```

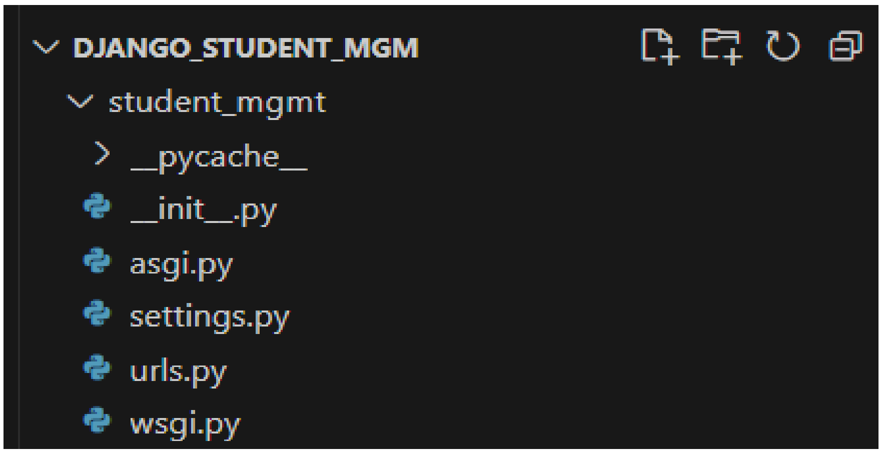
Step 2: Create a Django Project

Create a Django project inside django_student_mgm:

```
>>django-admin startproject student_mgmt .
```

This creates a student_mgmt folder containing:

- manage.py (Django's main command-line utility)
- student_mgmt/ (project configuration)

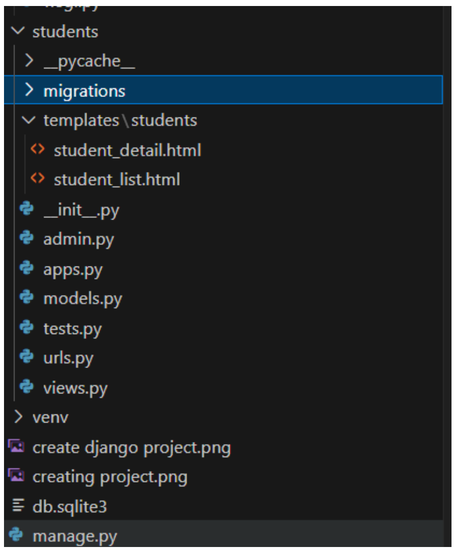


Step 3: Create a Django App

Create a Django app called students to handle student records.

1. Run:

```
>> python manage.py startapp students
```



```
students
  __pycache__
  migrations
  templates\students
    student_detail.html
    student_list.html
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
venv
create django project.png
creating project.png
db.sqlite3
manage.py
```

2. In student_mgmt/settings.py, **register the app**:

```
INSTALLED_APPS = [
  'django.contrib.admin',
  'django.contrib.auth',
  'django.contrib.contenttypes',
  'django.contrib.sessions',
  'django.contrib.messages',
  'django.contrib.staticfiles',
  'students’, # Add this   ]
```

Step 4: Define the Student Model

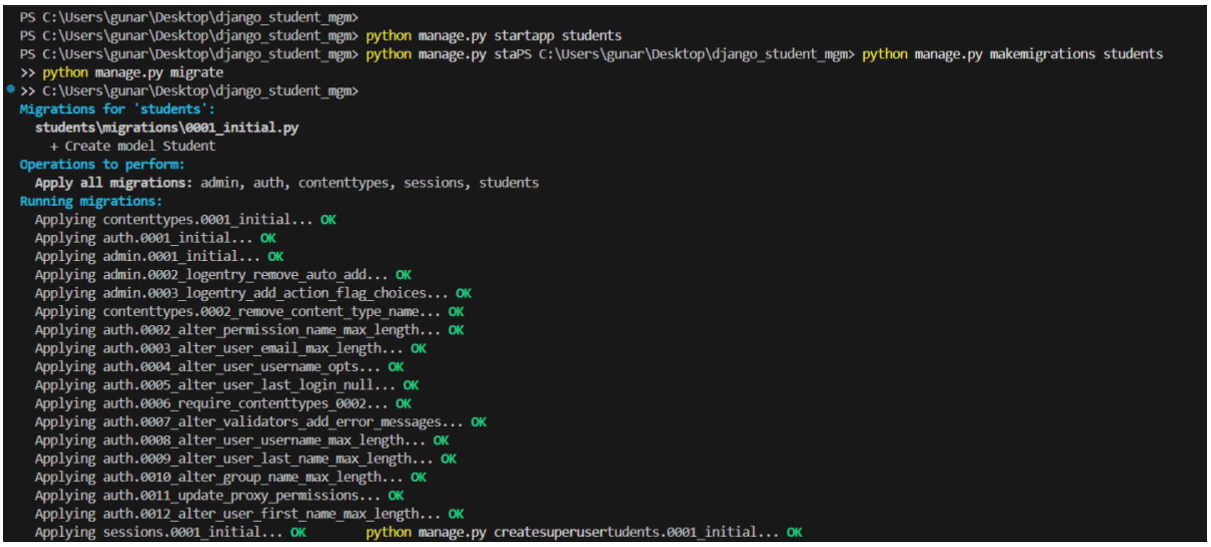
Open students/models.py and add:

```
from django.db import models
class Student(models.Model):
  name = models.CharField(max_length=100)
  roll_number = models.CharField(max_length=20, unique=True)
  course = models.CharField(max_length=100)
  email = models.EmailField(unique=True)
  date_of_birth = models.DateField()
  def __str__(self):
    return self.name
```

Step 5: Migrate the Database

```
python manage.py makemigrations students

python manage.py migrate
```



```
PS C:\Users\gunar\Desktop\django_student_mgm>
PS C:\Users\gunar\Desktop\django_student_mgm> python manage.py startapp students
PS C:\Users\gunar\Desktop\django_student_mgm> python manage.py staPS C:\Users\gunar\Desktop\django_student_mgm> python manage.py makemigrations students
>> python manage.py migrate
>> C:\Users\gunar\Desktop\django_student_mgm>
Migrations for 'students':
  students\migrations\0001_initial.py
    + Create model Student
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions, students
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK        python manage.py createsuperusertudents.0001_initial... OK
```

Step 6: Register the Model in Django Admin

Open students/admin.py and add:

```
from django.contrib import admin

from .models import Student


@admin.register(Student)

class StudentAdmin(admin.ModelAdmin):

    list_display = ('name', 'roll_number', 'course', 'email', 'date_of_birth')

    search_fields = ('name', 'roll_number', 'course')
```

Run the server: Create a superuser

```
python manage.py createsuperuser
```

Enter:

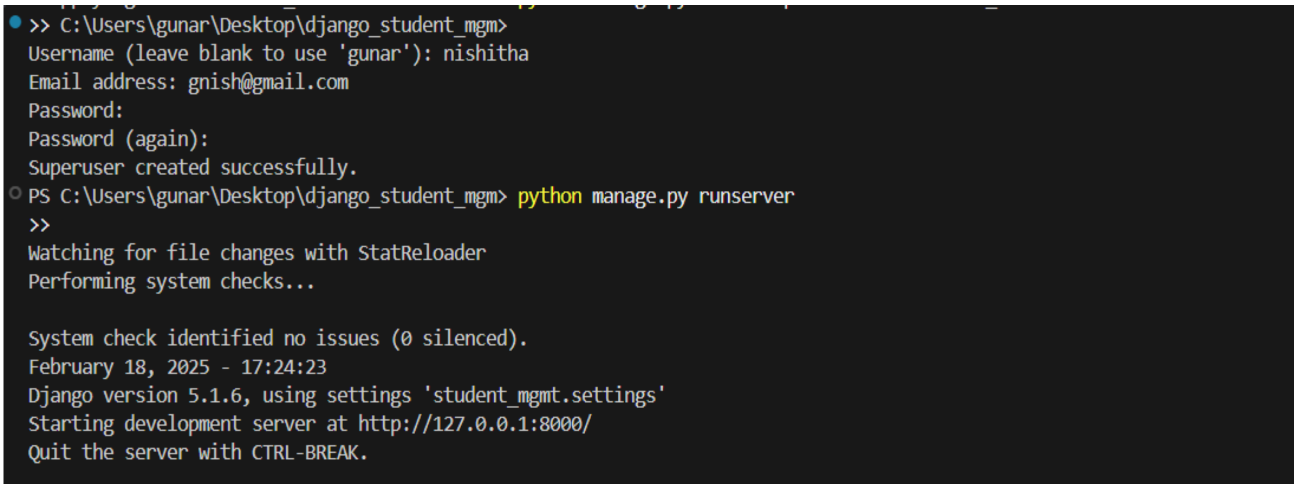
***Username** ***Password** ***Email**

Start the server:

```
python manage.py runserver
```

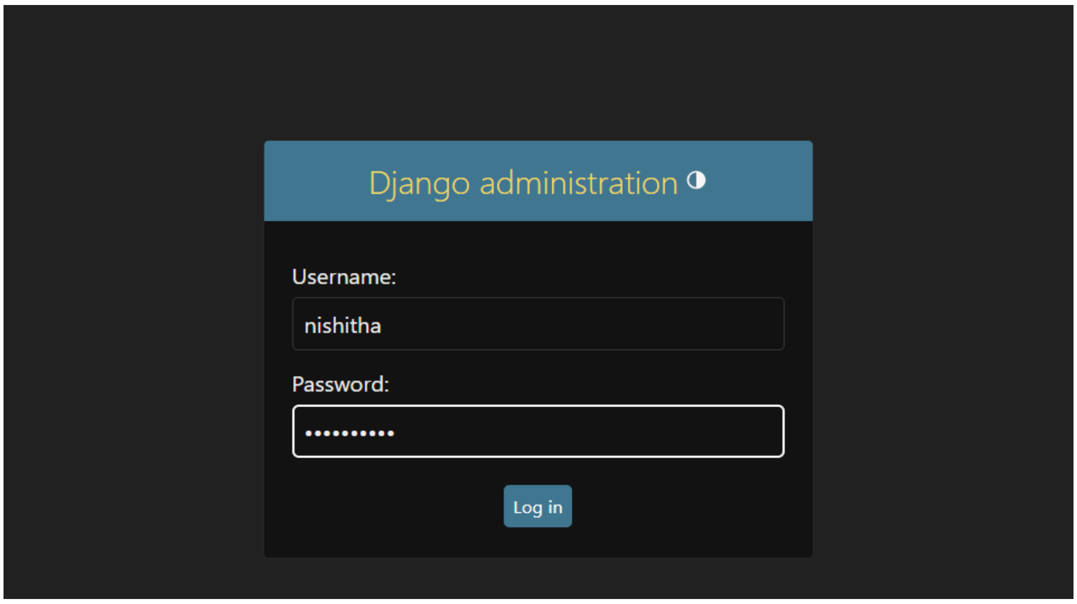
Go to: http://127.0.0.1:8000/admin/

Log in and add students manually.



```
>> C:\Users\gunar\Desktop\django_student_mgm>
Username (leave blank to use 'gunar'): nishitha
Email address: gnish@gmail.com
Password:
Password (again):
Superuser created successfully.
PS C:\Users\gunar\Desktop\django_student_mgm> python manage.py runserver
>>
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
February 18, 2025 - 17:24:23
Django version 5.1.6, using settings 'student_mgmt.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



Step 7: Create Views

Open students/views.py and add the following code:

```
from django.views.generic import ListView, DetailView
from django.urls import reverse_lazy
from .models import Student

class StudentListView(ListView):
    model = Student
    template_name = 'students/student_list.html'
    context_object_name = 'students'

class StudentDetailView(DetailView):
    model = Student
    template_name = 'students/student_detail.html'
    context_object_name = 'student'
```

Step 8: Set Up URLs

Create a **urls.py** file inside students/:

```
from django.urls import path
from .views import StudentListView, StudentDetailView

urlpatterns = [
    path('', StudentListView.as_view(), name='student-list'),
    path('<int:pk>/', StudentDetailView.as_view(), name='student-detail'),
]
```

Now, link it to the project **urls.py** (student_mgmt/urls.py):

```
from django.contrib import admin

from django.urls import path, include


urlpatterns = [

    path('admin/', admin.site.urls),

    path('students/', include('students.urls')),

]
```

Step 9: Create HTML Templates

- **Student List (student_list.html)**

Create a templates/students/ student_list.html
Add the following codes:

- **Student Detail (student_detail.html)**

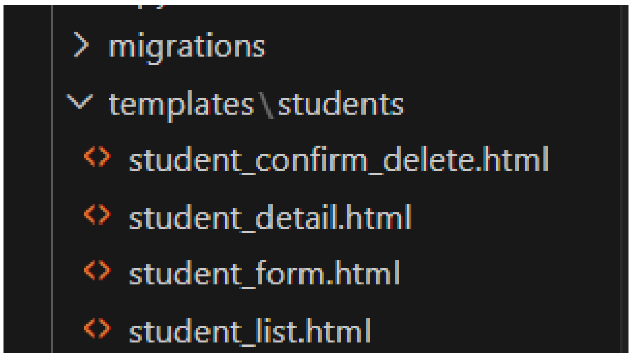
Create a templates/students/ student_detail.html
Add the following codes:

- **Student Form (student_form.html)**

Create a templates/students/ student_form.html
Add the following codes:

- **Student confirm Delete (student_ confirm_delete.html)**

Create a templates/students/ student_ confirm_delete.html
Add the following codes:



STUDENT_LIST:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Student Management</title>
   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="bg-light">

   <div class="container mt-4">
      <h1 class="text-center text-primary">Student Management System</h1>

      <!-- Display Success Messages -->
      {% if messages %}
         <div class="alert alert-success">
            {% for message in messages %}
               {{ message }}
            {% endfor %}
         </div>
      {% endif %}
      <!-- Student Form -->
      <div class="card shadow p-3">
         <h2 class="text-secondary">Add a New Student</h2>
         <form method="post">
            {% csrf_token %}
            <div class="mb-3">
                {{ form.as_p }}
            </div>
            <button type="submit" class="btn btn-success">Add Student</button>
        </form>
    </div>
    <hr>
<!-- Student List -->
    <div class="card shadow p-3 mt-4">
        <h2 class="text-secondary">Student List</h2>
        <table class="table table-striped">
            <thead class="table-dark">
                <tr>
                    <th>Name</th>
                    <th>Course</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                {% for student in students %}
                    <tr>
                        <td>{{ student.name }}</td>
                        <td>{{ student.course }}</td>
                        <td>
```

```
<a href="{% url 'student-detail' student.id %}" class="btn btn-primary btn-
sm">View</a>

<a href="{% url 'student-edit' student.id %}" class="btn btn-warning btn-
sm">Edit</a>

<a href="{% url 'student-delete' student.id %}" class="btn btn-danger btn-
sm">Delete</a>

                    </td>

                  </tr>

                {% endfor %}

              </tbody>

            </table>

        </div>

    </div>

</body>

</html>
```

STUDENT_DETAIL.HTML:

```
<!DOCTYPE html>

<html>

<head>

    <title>{{ student.name }}</title>

</head>

<body>

    <h1>{{ student.name }}</h1>

    <p>Roll Number: {{ student.roll_number }}</p>

    <p>Course: {{ student.course }}</p>
    <p>Email: {{ student.email }}</p>

    <p>Date of Birth: {{ student.date_of_birth }}</p>

    <a href="{% url 'student-list' %}">Back to List</a>

</body>

</html>
```

STUDENT_FORM.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

    <title>Edit Student</title>

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.mi
n.css">

</head>

<body class="bg-light">

    <div class="container mt-4">

        <h1 class="text-center text-primary">Edit Student</h1>

        <!-- Display Success Messages -->

        {% if messages %}

            <div class="alert alert-success">

                {% for message in messages %}

                    {{ message }}

                {% endfor %}

            </div>

        {% endif %}
```

```
      <div class="card shadow p-4">
          <form method="post">
             {% csrf_token %}
             {{ form.as_p }}
             <button type="submit" class="btn btn-warning">Update
Student</button>
          </form>
          <a href="{% url 'student-list' %}" class="btn btn-secondary mt-
3">Back to List</a>
       </div>
   </div>
</body>
</html>
```

STUDENT_CONFIRM_DELETE :

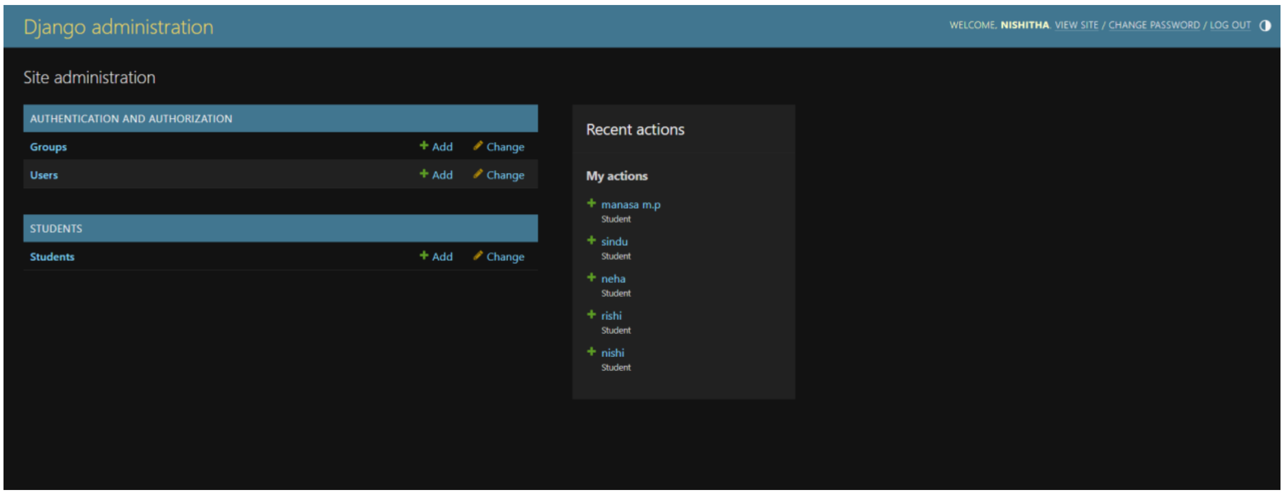
```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Delete Student</title>
   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.cs
s">
</head>
<body class="bg-light">
   <div class="container mt-4">
      <h1 class="text-center text-danger">Delete Student</h1>
      <div class="card shadow p-4">
         <p class="text-center">Are you sure you want to delete <strong>{{
student.name }}</strong>?</p>
         <form method="post" class="text-center">
            {% csrf_token %}
            <button type="submit" class="btn btn-danger">Yes, Delete</button>
            <a href="{% url 'student-list' %}" class="btn btn-
secondary">Cancel</a>
         </form>
      </div>
   </div>
</body>
</html>
```

Step 10: Run and Test

Restart the server

```
python manage.py runserver
```

- Visit: http://127.0.0.1:8000/students/
- Click on any student name to view details


Django administration

WELCOME, **NISHITHA**. VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

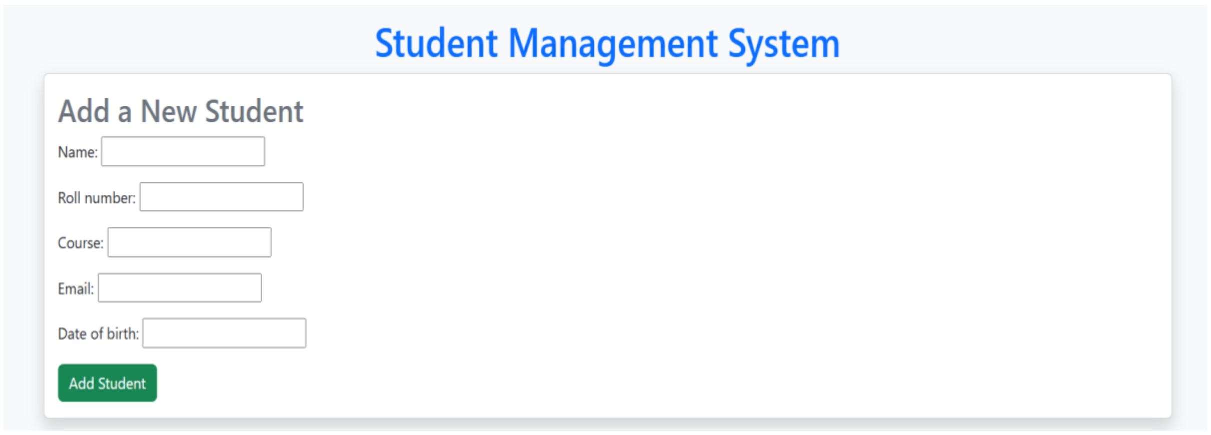
STUDENTS		
Students	+ Add	Change

Recent actions

My actions

- manasa m.p — Student
- sindu — Student
- neha — Student
- rishi — Student
- nishi — Student

Open: http://127.0.0.1:8000/students/


Student Management System

Add a New Student

Name:

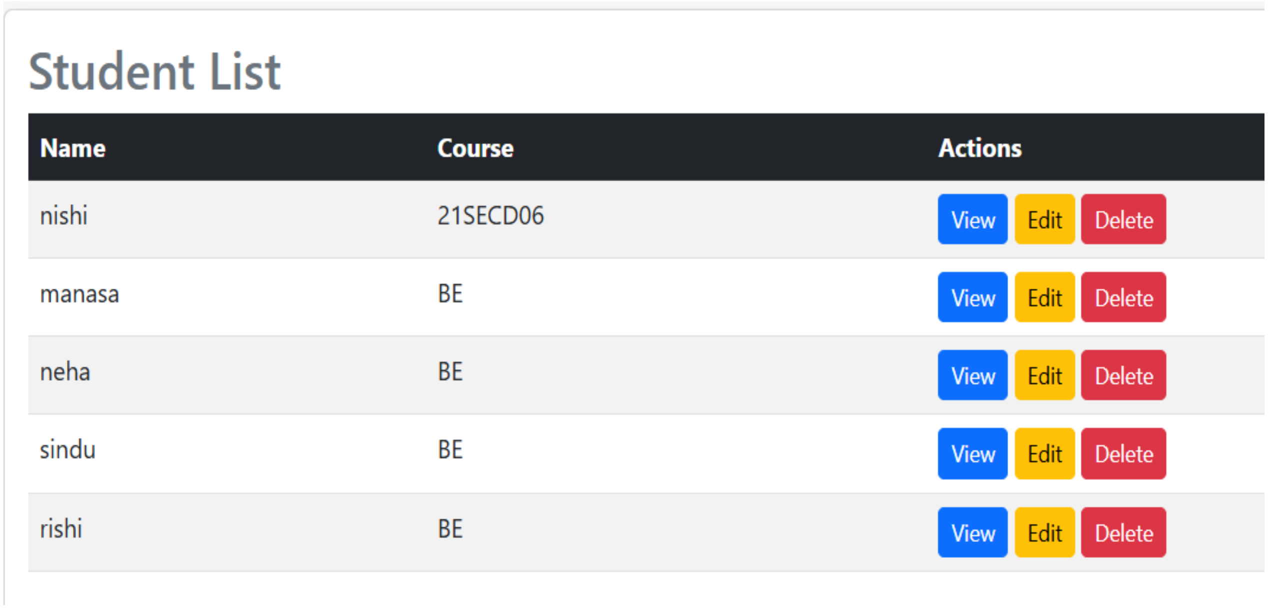
Roll number:

Course:

Email:

Date of birth:

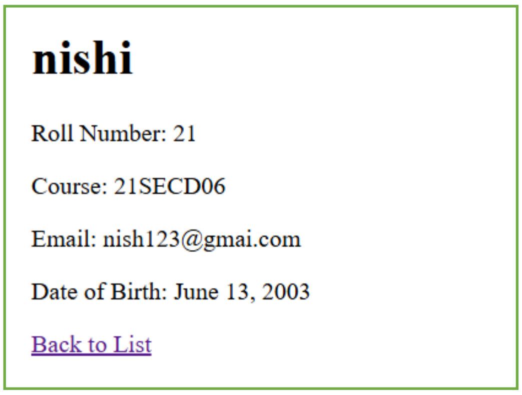
This is the Student Management System part of webpage is used to add student by add Name, Roll Number, Course, Email, Date of Birth of New Students



Student List

Name	Course	Actions
nishi	21SECD06	View Edit Delete
manasa	BE	View Edit Delete
neha	BE	View Edit Delete
sindu	BE	View Edit Delete
rishi	BE	View Edit Delete

This List displays the total number of students, clicking on their respective buttons, will open the respective students details.



nishi

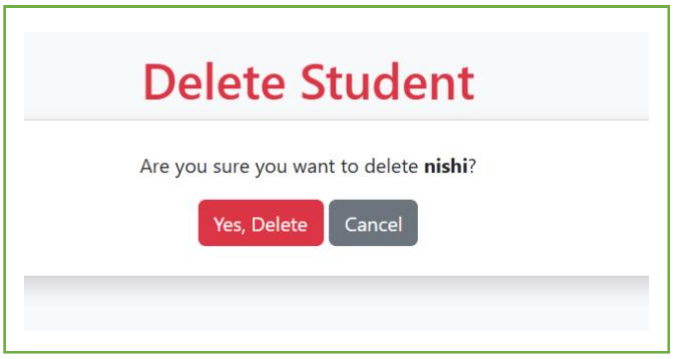
Roll Number: 21

Course: 21SECD06

Email: nish123@gmai.com

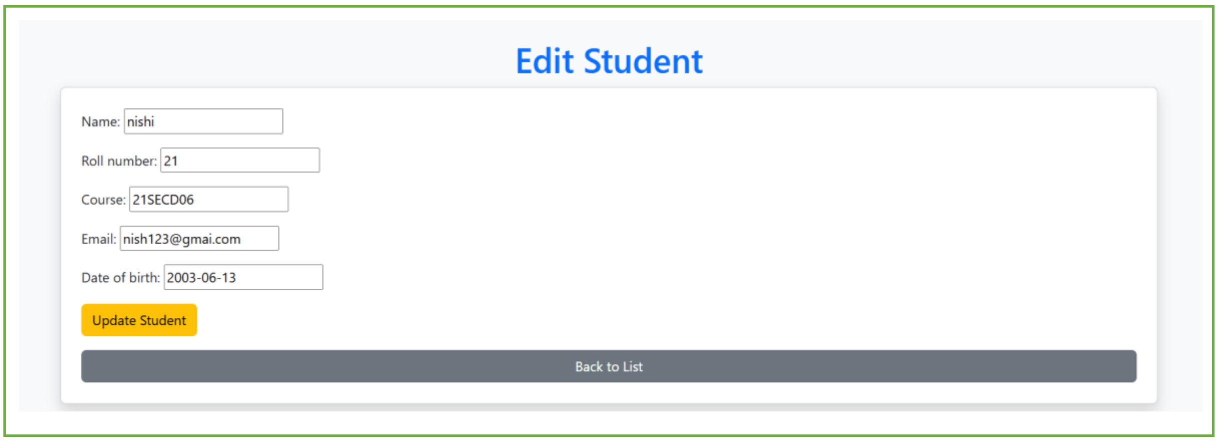
Date of Birth: June 13, 2003

Back to List



Delete Student

Are you sure you want to delete **nishi**?



Edit Student

Name: nishi

Roll number: 21

Course: 21SECD06

Email: nish123@gmai.com

Date of birth: 2003-06-13

This webpage on clicking particular student details will open their updates or allow us to delete or add or update respective student details.

****