



MYSORE UNIVERSITY SCHOOL OF ENGINEERING



Manasagangotri campus, Mysuru-
570006 (Approved by AICTE, New
Delhi)

UNIVERSITY OF MYSORE

Full Stack Development(21CD71)

Assessment Report On:

“STUDENT MANAGEMENT SYSTEM ”

Under the guidance :

Mr. Karthik M N
Assistant Professor,
Department of Computer Science
& Design,
MUSE.

Submitted by:

CHANDANA B S
Reg No : 21SECD04

Introduction:

This Django-based Student Management System includes:

- A Student model with fields for name, roll number, course, email, and date of birth.
- An admin interface to manage student records.
- Class-based views (ListView and DetailView) for displaying student lists and details.
- URL configuration for navigation using `reverse_lazy()`.

Project overview:

```
student_management/  
| — manage.py  
| — db.sqlite3  
| — student_management/  
|   | — __init__.py  
|   | — settings.py  
|   | — urls.py  
|   | — asgi.py  
|   | — wsgi.py  
| — students/  
|   | — __init__.py  
|   | — admin.py  
|   | — apps.py  
|   | — models.py  
|   | — tests.py  
|   | — views.py  
|   | — urls.py  
|   | — templates/
```

```
| | | — students/  
| | | | — student_list.html  
| | | | — student_detail.html  
| | | | — student_form.html  
| | | | — student_confirm_delete.html  
| — venv/ (Virtual Environment)
```

Detailed steps Implementation:

Step 1: Install Django and Create a Virtual Environment

Create a virtual environment

```
python -m venv venv
```

Activate the virtual environment

On Windows:

```
venv\Scripts\activate
```

On macOS/Linux:

```
source venv/bin/activate
```

Install Django

```
pip install Django
```

Step 2: Create a Django Project

Run the following command to create a Django project:

```
django-admin startproject student_management  
cd student_management
```

Step 3: Create a Django App

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

Step 4: Configure settings.py:

Open *student_management/settings.py* and add '*students*' to *INSTALLED_APPS*

Step 5: Create the Student Model:

Run migrations to apply the model:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

Step 6: Register the Model in Django Admin:

In *students/admin.py*:

Step 7: Create Views for Student Management:

In *students/views.py*

Step 8: Configure URLs:

Create *students/urls.py*

Link the *students* app to the project's main *urls.py* in *student_management_system/urls.py*

Step 9: Create HTML Templates:

Inside *students/templates/students/* create files naming:

1. *student_list.html*
2. *student_form.html*
4. *student_detail.html*
5. *student_confirm_delete.html*

Step 10: Create a Superuser for Admin Panel:

```
python manage.py createsuperuser
```

Step 11: Run the Django Development Server

```
python manage.py runserver
```

Conclusion

You have successfully created a **Django-based Student Management System** with:

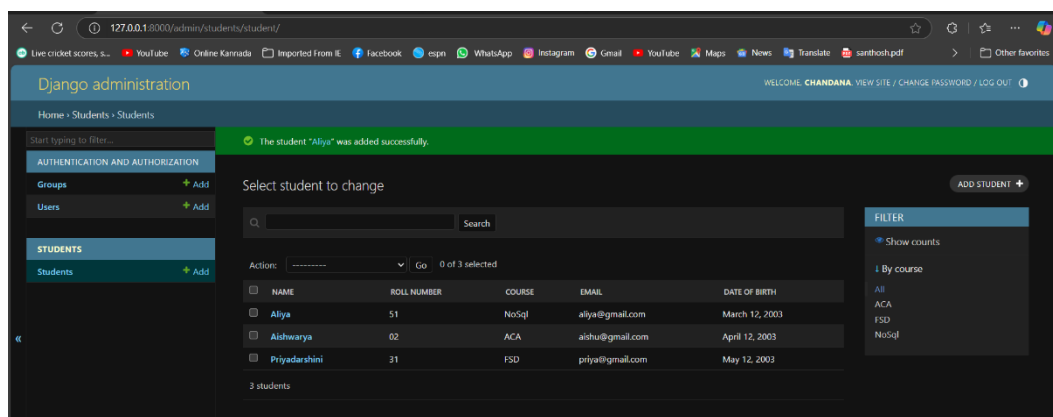
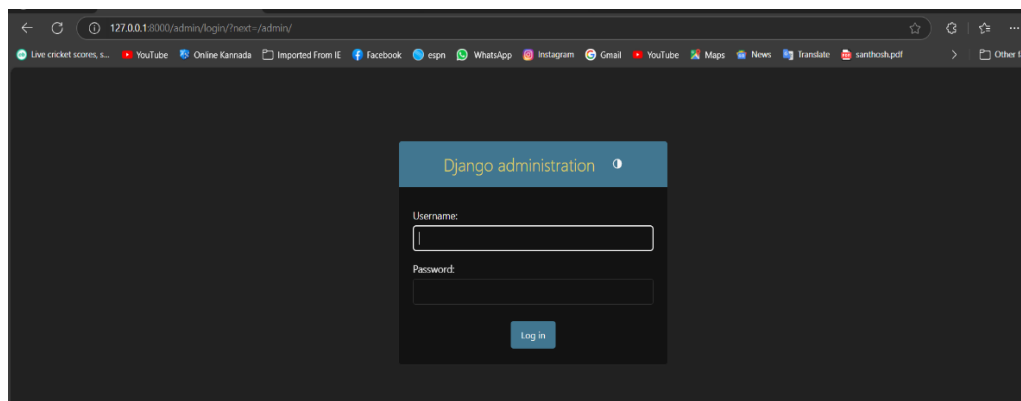
Django Admin to manage students

List and Detail views using Django's generic ListView & DetailView

Navigation with reverse_lazy()

Fully functional database & templates

Output:



127.0.0.1:8000/students/

Student List

[Priyadarshini](#) - FSD

[Aishwarya](#) - ACA

[Aliya](#) - NoSql

Student List

Name	Roll Number	Course	Email	Actions
Priyadarshini	31	FSD	priya@gmail.com	View Edit Delete
Aishwarya	02	ACA	aishu@gmail.com	View Edit Delete
Aliya	51	NoSql	aliya@gmail.com	View Edit Delete

127.0.0.1:8000/students/1/

Priyadarshini

Roll Number: 31

Course: FSD

Email: priya@gmail.com

Date of Birth: May 12, 2003

[Edit](#) [Delete](#) [Back to List](#)