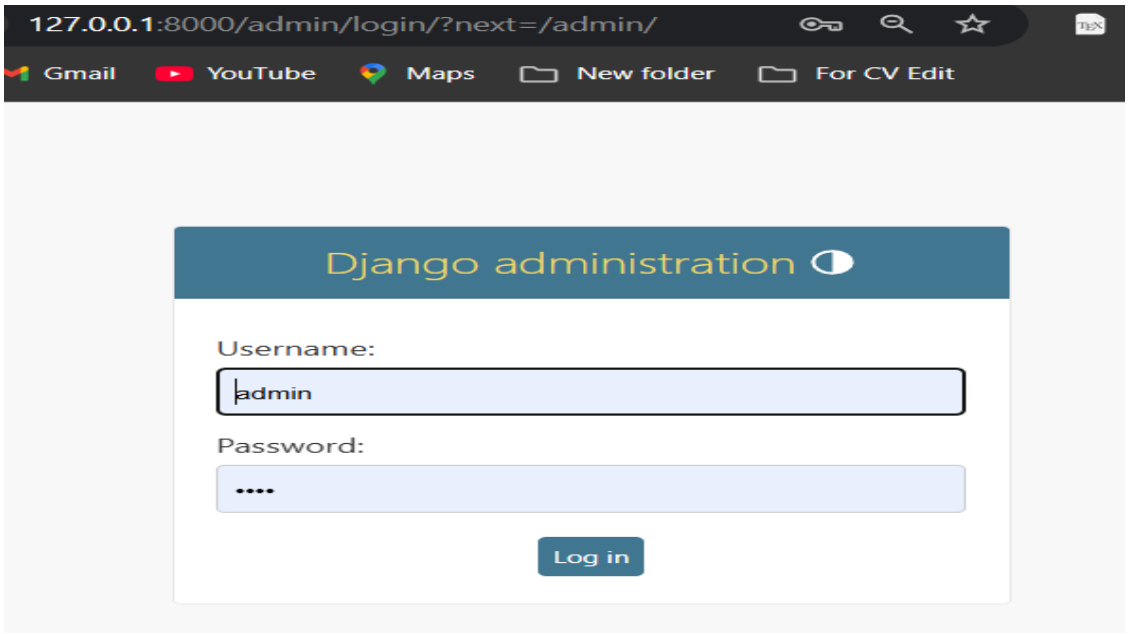
Admin Panel of Django

Django's **admin interface** is a powerful built-in tool that automatically generates a management panel from your models. It allows trusted users to easily manage site content. However, it is mainly intended for **internal use** within an organization, **not** for building the public front end of a website. It allows developers and administrators to **create, read, update, and delete (CRUD)** records without writing SQL commands or custom views.

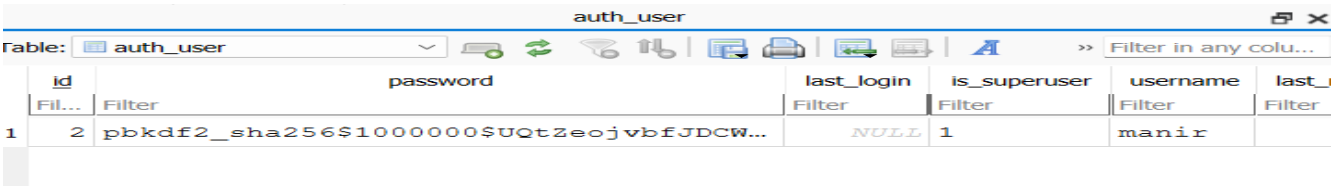
Let's work with admin panel.



Before accessing the admin panel, let's create a superuser to log in and manage the Django admin interface. Write following command:

```
PS D:\DjangoRoom\studentListToNbAddStudentAdminPanel\studentList> python manage.py createsuperuser
Username (leave blank to use 'lenovo'): manir
Email address: manir@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

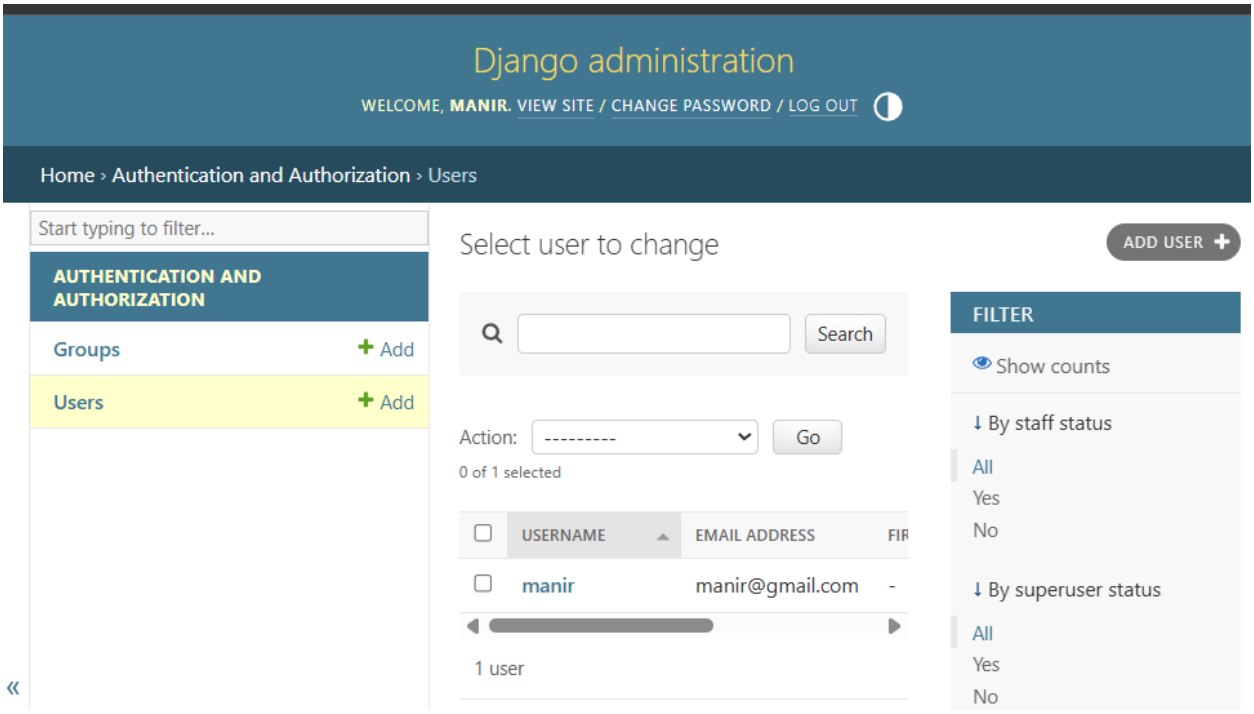
A record is inserted to the table "auth_user" open DB browser and browse your database.



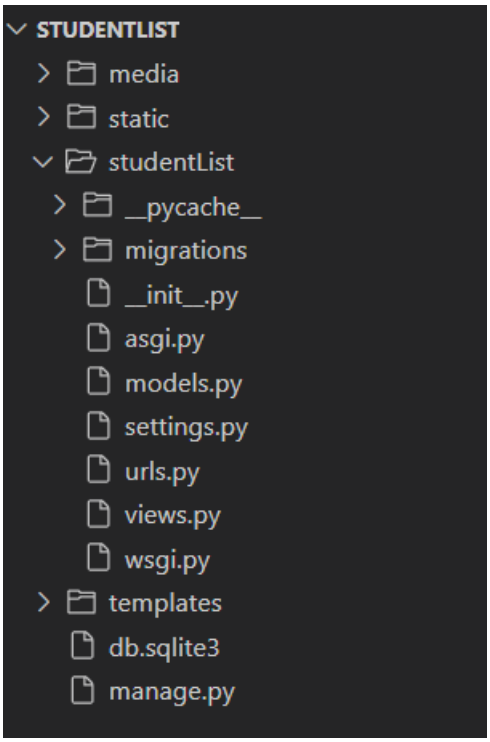
Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

Knowledge is Power
DIU
Estd. 7th April, 1995

Open Django's admin panel, and you will see that one user has already been created.



Your project folder structure:

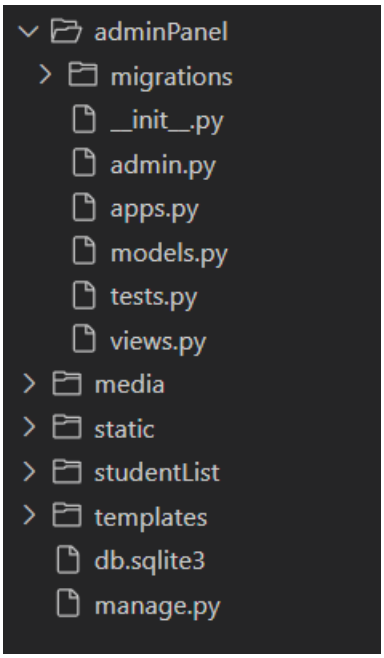


Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

Now let's create a new 'app' set name as 'adminPanel' follow given command:

```
python manage.py startapp adminPanel
```

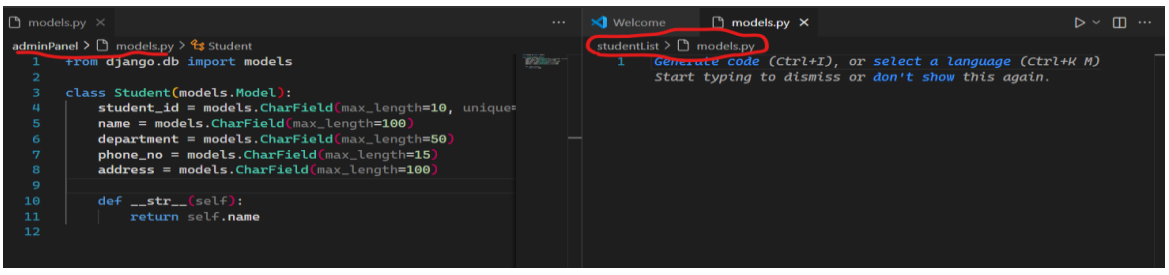
Your folder structure will be like:



You can register the models located in **adminPanel/models.py**.
We have already worked on a model named **Student** — now let's remove the table code from **studentList/models.py** and write the **Student** model code inside **adminPanel/models.py** instead.

This change allows us to access and manage the **Student** table directly from the **Django Admin Panel** and add data through it.

We already have an **add_student.html** page for adding data manually, but from now on, we'll use the **Admin Panel**.
If you want, you can still keep the HTML page — just **remove the Student model class code** from the **studentList** app.



Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

To keep the previous work unchanged, make a small modification in **studentList/views.py**:

Change:

```
from .models import Student
```

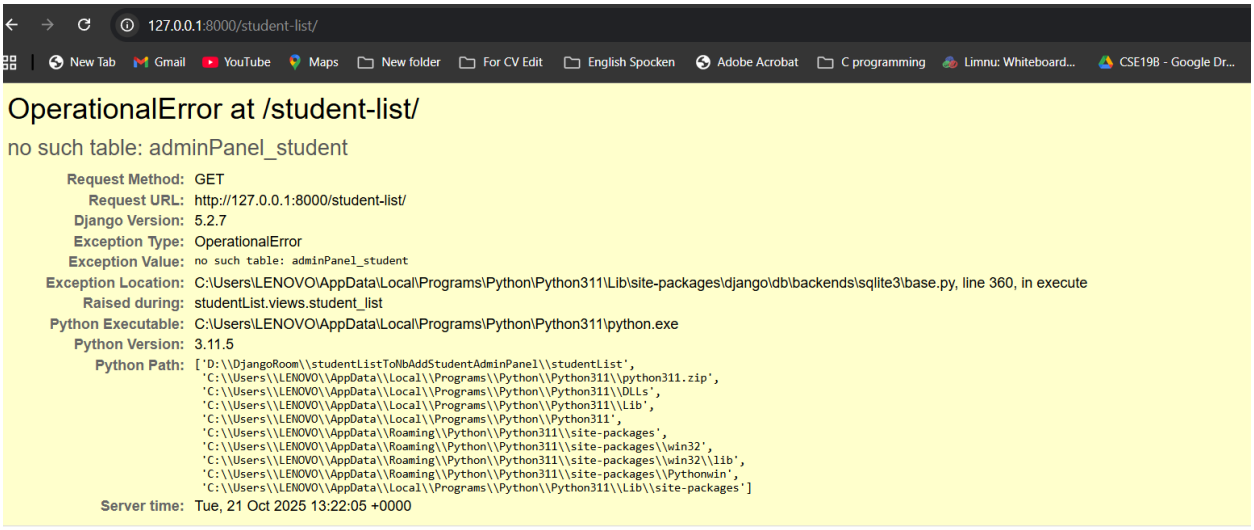
To:

```
from adminPanel.models import Student
```

Next, add the **adminPanel** app to the **INSTALLED_APPS** list in **studentList/settings.py**:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'studentList',
    'adminPanel',
]
```

Now the project runs smoothly, but an error appears when opening **studentlist.html**.



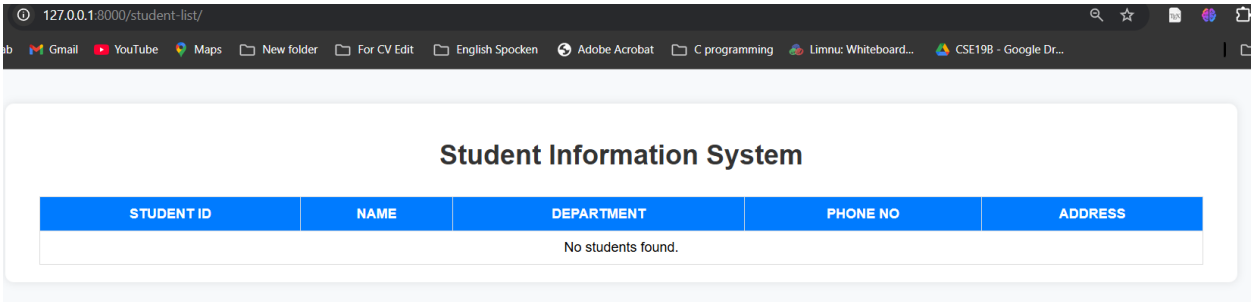
This happens because the previous **Student** model was removed, and the new **Student** model in **adminPanel/models.py** has not been migrated yet.

To fix this issue, run the following commands to create and apply migrations:

Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

```
PS D:\DjangoRoom\studentListToNbAddStudentAdminPanel\studentList> python manage.py makemigrations
Migrations for 'adminPanel':
  adminPanel\migrations\0001_initial.py
    + Create model Student
Migrations for 'studentList':
  studentList\migrations\0002_delete_student.py
    - Delete model Student
PS D:\DjangoRoom\studentListToNbAddStudentAdminPanel\studentList> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, adminPanel, auth, contenttypes, sessions, studentList
Running migrations:
  Applying adminPanel.0001_initial... OK
  Applying studentList.0002_delete_student... OK
```

Now open the **studentlist** page. You will notice that the table is empty because we are using the new **Student** table, and no records have been added yet.

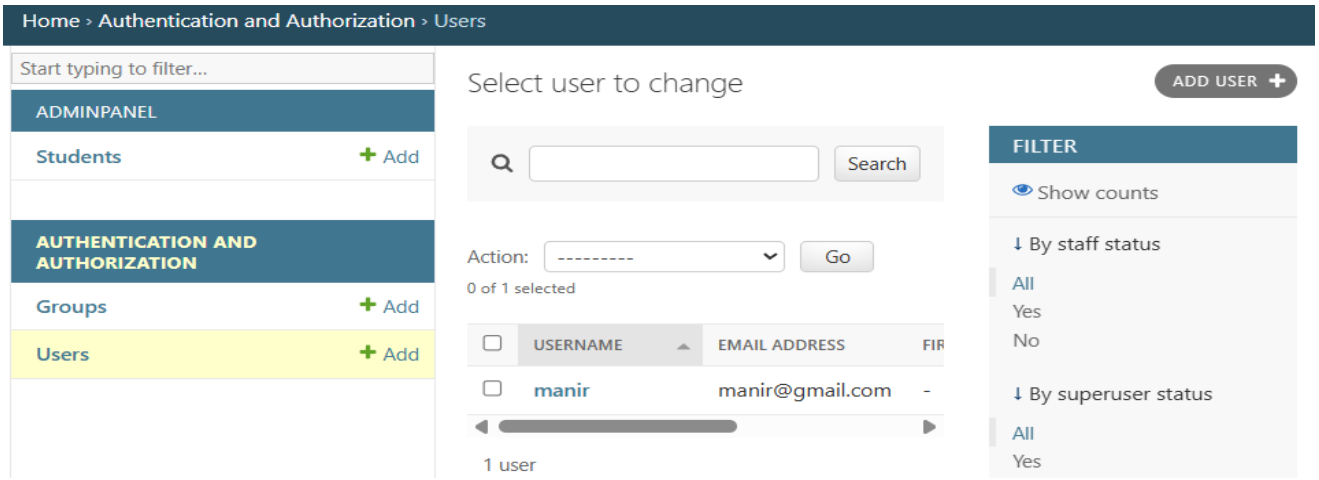


Now our task is to register the Student table from adminPanel/models.py in adminPanel/admin.py.

Write the following code inside **adminPanel/admin.py**:

```python
from django.contrib import admin
# Register your models here.
from .models import Student   # import your model
# Register the model so it appears in admin panel
admin.site.register(Student)
```

Open Django admin panel you will se a table is created in the app ADMINPANEL:

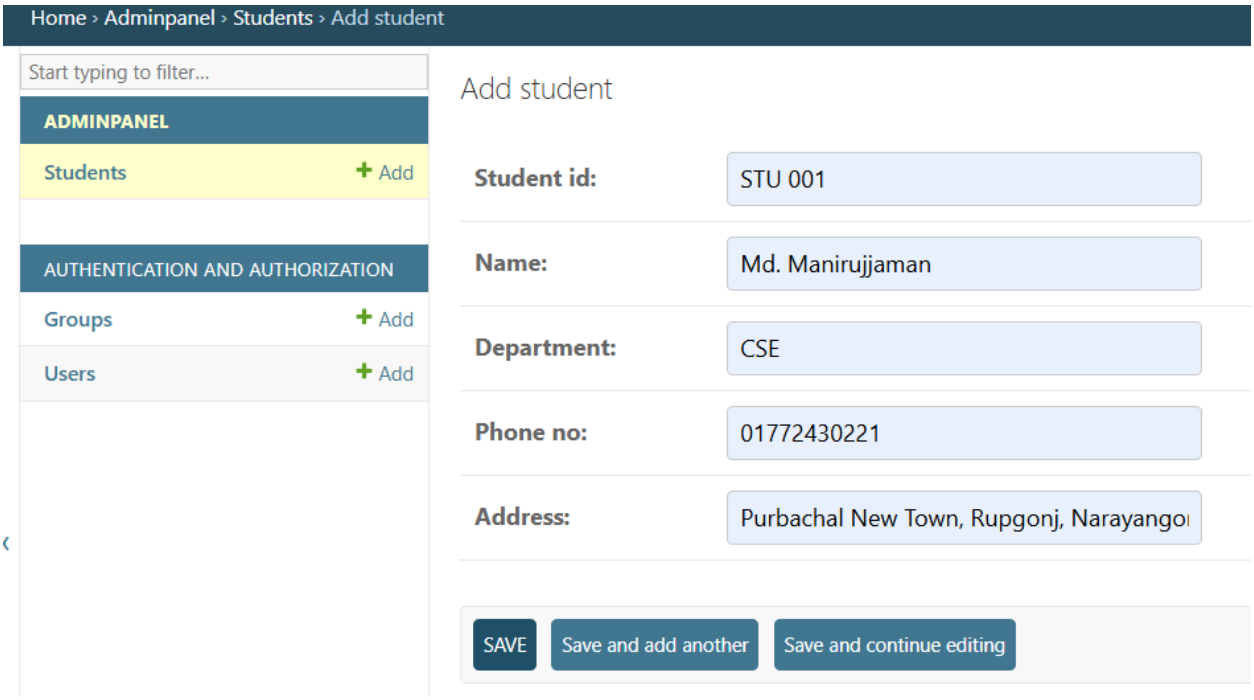


Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

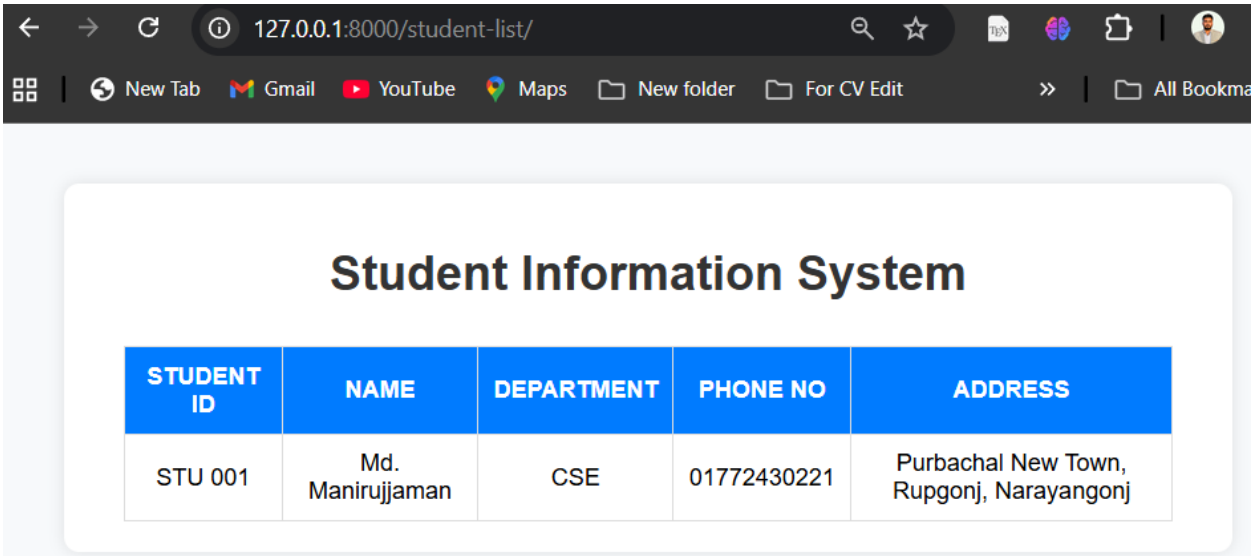
Knowledge is Power
DIU
Estd. 7th April, 1995

If you want to add student just click on +add 

and you will get following form now insert data from here and click save.

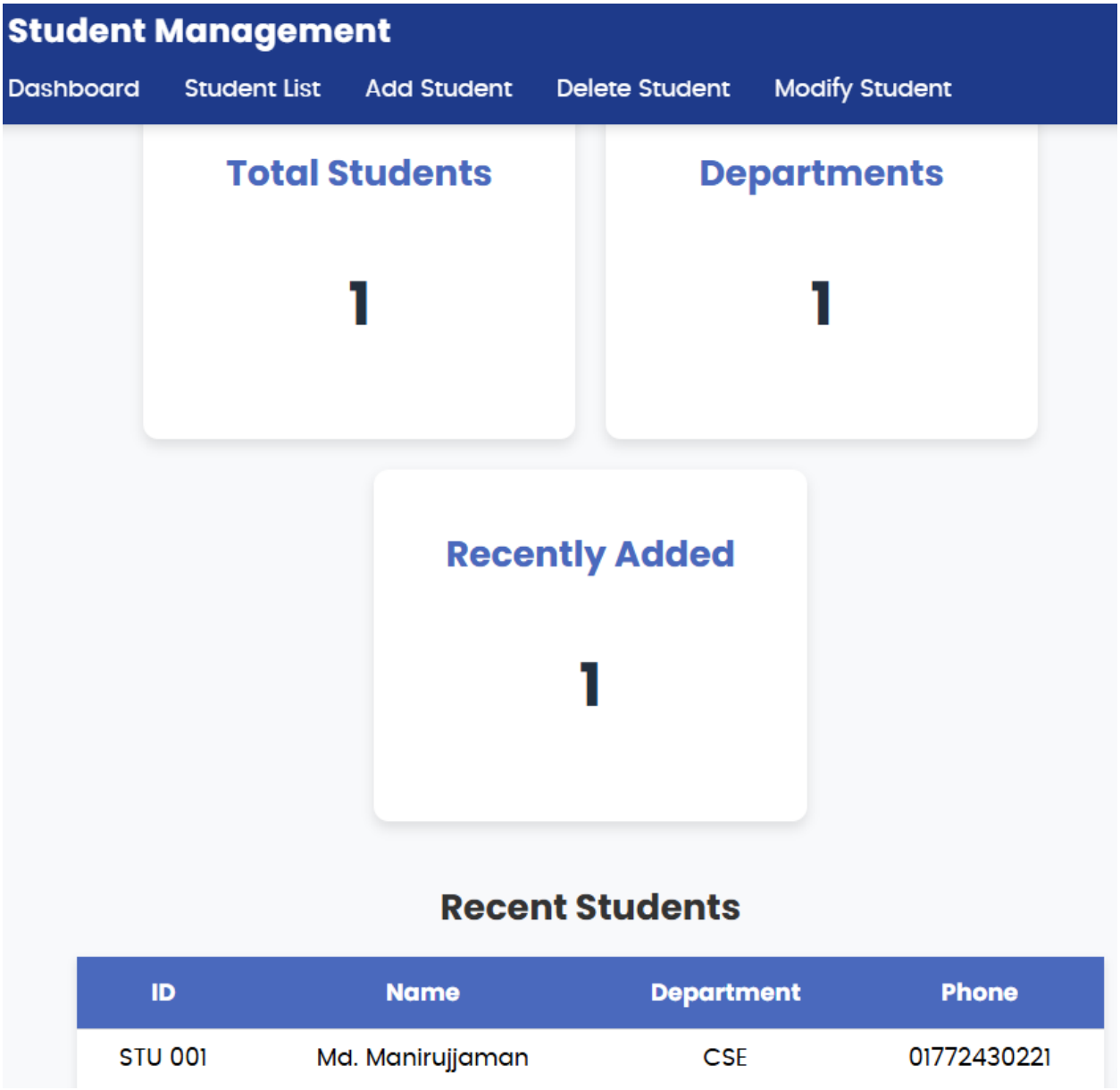


Open studentlist url you will se that data is inserted.



Open your daseboard also.

Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com



Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

Explain why we created a new **adminPanel** app instead of using our main **studentList** app.

To answer this question first of all we need to know why we need apps.

In Django, a project is the overall web application, it contains all the settings, configurations, and apps needed to run your site.An app, on the other hand, is a specific module or component inside that project.

Why we need to create an app in a Django project?

1. Modularity (Separation of features)
   Each app focuses on a single purpose — like user management, blog, or payment system.
   Example:
   - accounts app → handles login, signup, profile
   - blog app → handles posts, comments
   - shop app → handles products, orders

   This makes your project organized and easy to maintain.

2. Reusability
   You can easily reuse an app in another project.
   Example: A "contact form" app you built for one project can be copied and plugged into another Django project with little modification.

3. Team Collaboration
   When multiple developers work together, each person can work on a separate app without affecting others.

4. Scalability
   As your project grows, you can add or remove apps without changing the whole project structure.

(just for checking is new rendering works)Let's render the dashboard.html page from adminPanel app.

Need make some changes. Copy dashboard function from studentList/views.py and paste it to

adminPanel/views.py

adminPanel/views.py will be like:

```
from django.shortcuts import render
from .models import Student

def dashboard(request):
    # Fetch all students
```

Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

```
    students = Student.objects.all()

    # Total count
    total_students = students.count()

    # Unique department count
    departments = students.values_list('department', flat=True).distinct()
    department_count = len(departments)

    # Get last 5 recently added students
    recent_students = students.order_by('-id')[:5]

    context = {
        'total_students': total_students,
        'department_count': department_count,
        'recent_students': recent_students,
    }
    return render(request, 'dashboard.html', context)
```

add **urls.py** in adminPanel app. Set the path of dashboard.html page. Ooh remove the path first from adminPanel/urls.py then add the path to urls.py.

change studentList/urls.py

```
from django.contrib import admin
from django.urls import path,include
from . import views

urlpatterns = [
    path('admin/', admin.site.urls),
    #path('',views.dashboard, name='dashboard'),
    path('student-list/', views.student_list, name='student_list'),
    path('add-student/', views.add_student, name='add_student'),
    path('', include('adminPanel.urls')),

]
```

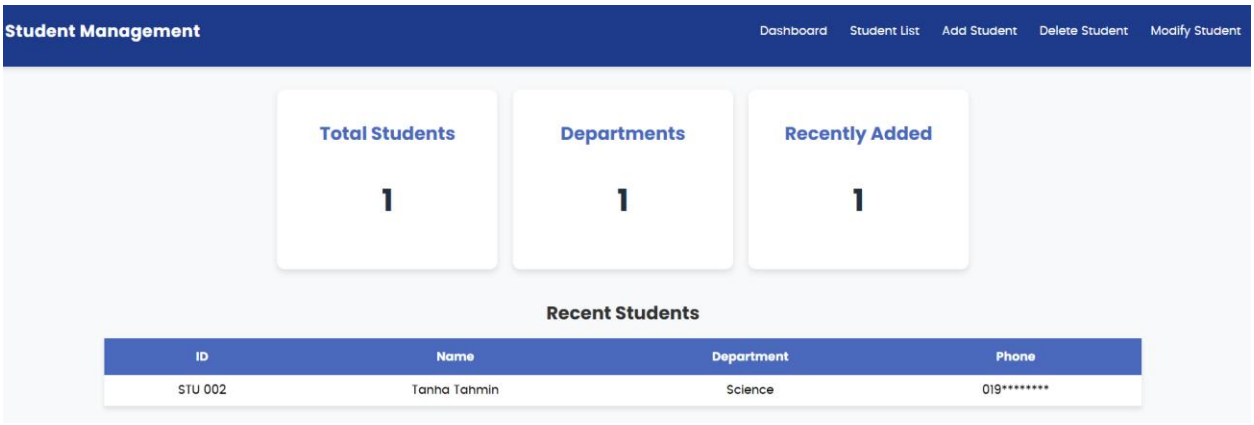
Update adminPanel/urls.py

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

Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com

```
urlpatterns = [
    path('', views.dashboard, name='dashboard'),
]
```

Now run the server you will get dashboard.page at home but it is rendered from adminPanel app.



Md. Manirujjaman
Lecturer, CSE, DIU
manirujjaman25803690@gmail.com