

# Student-Teacher Relationship Management System

This report details the step-by-step implementation of a Django-based Student-Teacher Relationship Management System. The system allows managing students and teachers with a one-to-many relationship and uses Django's generic views for displaying data.

## 1. Environment Setup

First, install Django and create a virtual environment:

- Install Django: ``pip install django``
- Check installation: ``django-admin --version``
- Create a project: ``django-admin startproject student_teacher_mgmt``
- Navigate to the project folder: ``cd student_teacher_mgmt``
- Run the server: ``python manage.py runserver``

## 2. Creating the Management App

Inside the Django project, create an app for managing student-teacher relationships:

- ``python manage.py startapp management``
- Add 'management' to ``INSTALLED_APPS`` in ``settings.py``.

## 3. Defining Models

Define models for ``Teacher`` and ``Student`` in ``management/models.py``:

```
```python
from django.db import models

class Teacher(models.Model):
    name = models.CharField(max_length=100)
    subject = models.CharField(max_length=100)

class Student(models.Model):
    name = models.CharField(max_length=100)
    age = models.IntegerField()
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
...

```

Run migrations:

- `python manage.py makemigrations`
- `python manage.py migrate`

#### 4. Register Models in Django Admin

Register models in `management/admin.py`:

```
```python
from django.contrib import admin
from .models import Teacher, Student

admin.site.register(Teacher)
admin.site.register(Student)
```
```

Now, you can manage data from `http://127.0.0.1:8000/admin/`.

#### 5. Creating Views

Use Django's generic views in `management/views.py`:

```
```python
from django.views.generic import ListView, DetailView
from .models import Teacher, Student

class TeacherListView(ListView):
    model = Teacher
    template_name = 'management/teacher_list.html'

class TeacherDetailView(DetailView):
    model = Teacher
    template_name = 'management/teacher_detail.html'

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

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

## 6. Defining URLs

Define paths in `management/urls.py`:

```
```python
from django.urls import path
from .views import TeacherListView, TeacherDetailView, StudentListView, StudentDetailView

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

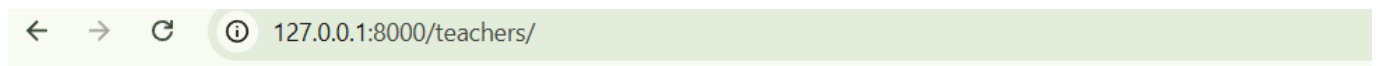
## 7. Creating Templates

Create HTML files in `management/templates/management/`:

- `teacher\_list.html`
- `teacher\_detail.html`
- `student\_list.html`
- `student\_detail.html`

Place screenshots of your pages here.

## 8. Screenshots



### Teachers

- [ram](#)

# Students

- [Zoya](#)
- [Mark](#)

Site administration

| AUTHENTICATION AND AUTHORIZATION |       |          |
|----------------------------------|-------|----------|
| Groups                           | + Add | ✎ Change |
| Users                            | + Add | ✎ Change |
| MANAGEMENT                       |       |          |
| Students                         | + Add | ✎ Change |
| Teachers                         | + Add | ✎ Change |

Recent actions

My actions

- + Mark Student
- ✎ Zoya Student
- + satish Student
- + ram Teacher

Django administration ⓘ

Username:

Password:

Log in

The screenshot shows a VS Code editor with the file explorer on the left, displaying the project structure for 'student\_teacher\_mgmt'. The 'manage.py' file is selected in the explorer and open in the editor. The code in 'manage.py' is as follows:

```
1 #!/usr/bin/env python
2 """Django's command-line utility for administrative tasks."""
3 import os
4 import sys
5
6
7 def main():
8     """Run administrative tasks."""
9     os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'student_teacher_mgmt.settings')
10    try:
11        from django.core.management import execute_from_command_line
12    except ImportError as exc:
13        raise ImportError(
14            "Couldn't import Django. Are you sure it's installed and "
15            "available on your PYTHONPATH environment variable? Did you "
16            "forget to activate a virtual environment?"
17        ) from exc
18    execute_from_command_line(sys.argv)
19
20
21 if __name__ == '__main__':
22     main()
23
```

Below the editor, the terminal window shows the following output:

```
[21/Feb/2025 22:12:59] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 304 0
[21/Feb/2025 22:12:59] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 304 0
Not Found: /
[21/Feb/2025 22:13:22] "GET / HTTP/1.1" 404 3257
[21/Feb/2025 22:13:25] "GET /admin/ HTTP/1.1" 200 8768
[21/Feb/2025 22:13:29] "POST /admin/logout/ HTTP/1.1" 200 3555
[21/Feb/2025 22:13:32] "GET /admin/ HTTP/1.1" 302 0
[21/Feb/2025 22:13:32] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4160
[21/Feb/2025 22:13:32] "GET /static/admin/css/login.css HTTP/1.1" 304 0
```

## Conclusion

The Student-Teacher Relationship Management System is now fully implemented using Django's generic views, models, templates, and admin management. You can now add teachers, assign students to them, and display the details using ListView and DetailView.