



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:

Srihari S

Reg No : 21SECD45

Introduction:

- A Django-based web application managing student-teacher relationships
- Implements a one-to-many relationship where each student has one teacher but teachers can have multiple students
- Features both admin interface and user-facing views
- Uses Django's Class-Based Views for displaying data

Project Structure:

school_management/

```
|— school_management/  # Project directory
| |— __init__.py
| |— settings.py      # Project settings
| |— urls.py          # Main URL configuration
| |— wsgi.py
|— school/             # App directory
| |— __init__.py
| |— admin.py         # Admin interface configuration
| |— models.py        # Database models
| |— views.py         # View logic
| |— urls.py          # App URL configuration
|— templates/         # Project-wide templates
| |— base.html        # Base template
|— manage.py          # Django management script
```

Implementation Steps:

Step 1: Environment Setup

- Created virtual environment for project isolation
- Installed Django using pip
- Created new Django project and app

Step 2: Database Models

- Created Teacher model with fields:
 - first_name
 - last_name
 - email (unique)
 - subject
 - hire_date
- Created Student model with fields:
 - first_name
 - last_name
 - email (unique)
 - roll_number
 - date_of_birth
 - teacher (ForeignKey relationship)

Step 3: Admin Interface

- Registered models with Django admin
- Customized admin display with:
 - List displays
 - Search fields
 - List filters

Step 4: Views

- Implemented Class-Based Views:
 - TeacherListView
 - TeacherDetailView
 - StudentListView
 - StudentDetailView
- Created corresponding URL patterns

Step 5: Templates

- Created base template with:
 - Navigation menu
 - Basic styling
 - Content block
- Created specific templates:
 - Home page
 - Teacher list and detail views
 - Student list and detail views

Problem-Solving Process:

Initial Setup Issues:

- Problem: 'django-admin' command not recognized
- Solution: Used python -m django instead
- Reason: PATH variables not set correctly

Template Error Resolution:

- Problem: TemplateDoesNotExist error
- Solution:
 1. Created proper template directory structure
 2. Added template directories to settings.py
 3. Created base.html template
- Reason: Django couldn't find the template files

URL Configuration:

- Problem: Initial URL import error
- Solution:
 1. Properly structured URL patterns
 2. Used include() for app URLs
 3. Added homepage view and URL

Features Implemented:

Admin Interface:

- Custom list displays
- Search functionality
- Filtering options
- Easy data management

User Interface:

- Homepage with navigation
- Teacher list and detail views
- Student list and detail views
- Responsive design with basic styling

Data Relationships:

- One-to-many relationship between teachers and students
- Easy navigation between related records
- Proper data integrity maintenance

Technologies Used:

- Django 5.1.6
- Python 3.12.5
- SQLite database (Django default)
- HTML/CSS for templates
- Django Class-Based Views
- Django Admin interface

Future Enhancements Possible:

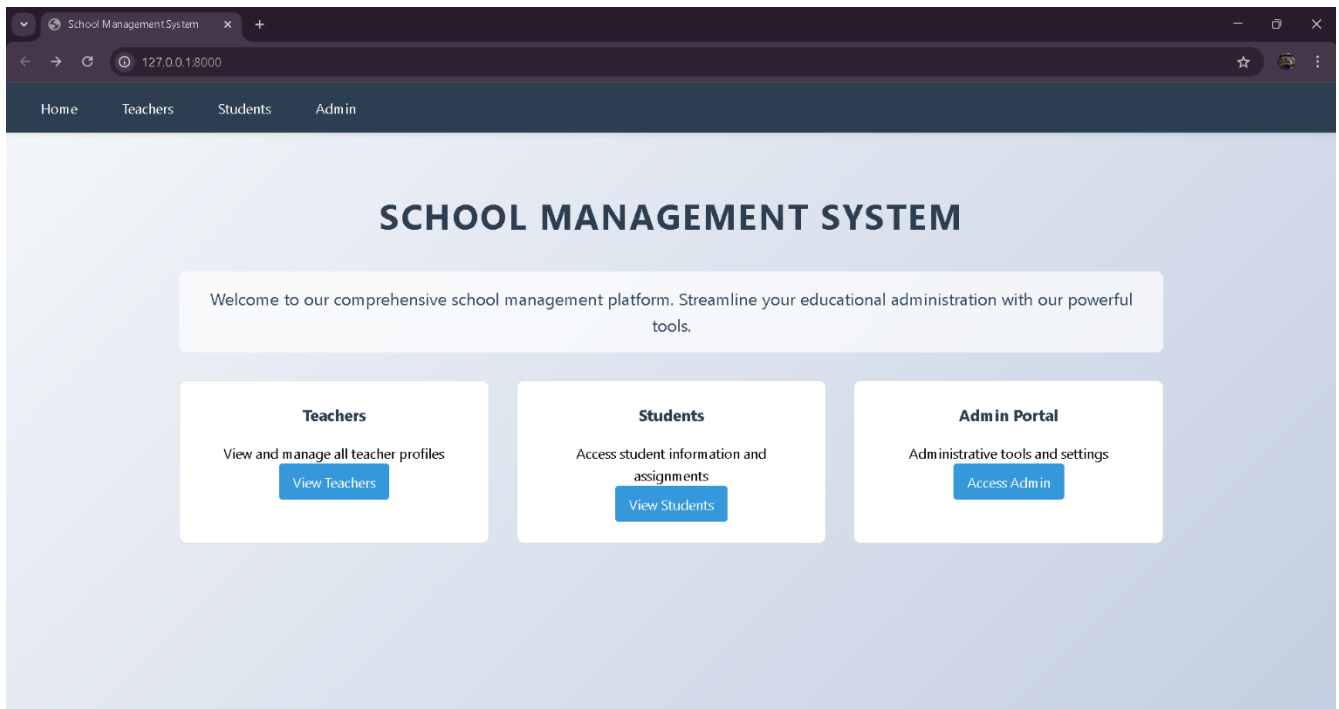
- User authentication and permissions
- Student/teacher dashboards
- Grade management system
- Attendance tracking
- Parent portal
- File upload for assignments
- Email notifications
- API endpoints for data access

Learning Outcomes:

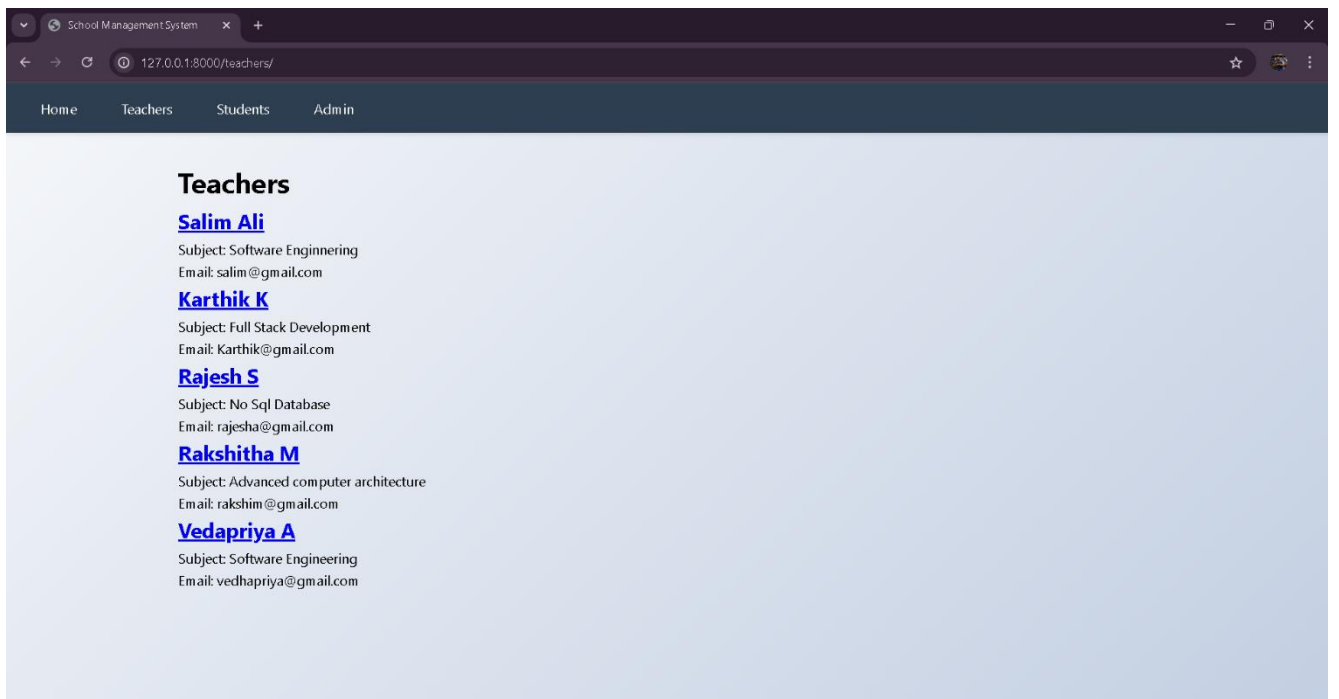
- Django project structure organization
- Model relationships implementation
- Class-Based Views usage
- Template inheritance
- Admin interface customization
- URL routing and configuration
- Error handling and debugging

This project demonstrates the basic concepts of Django web development while providing a practical solution for managing student-teacher relationships in an educational setting.

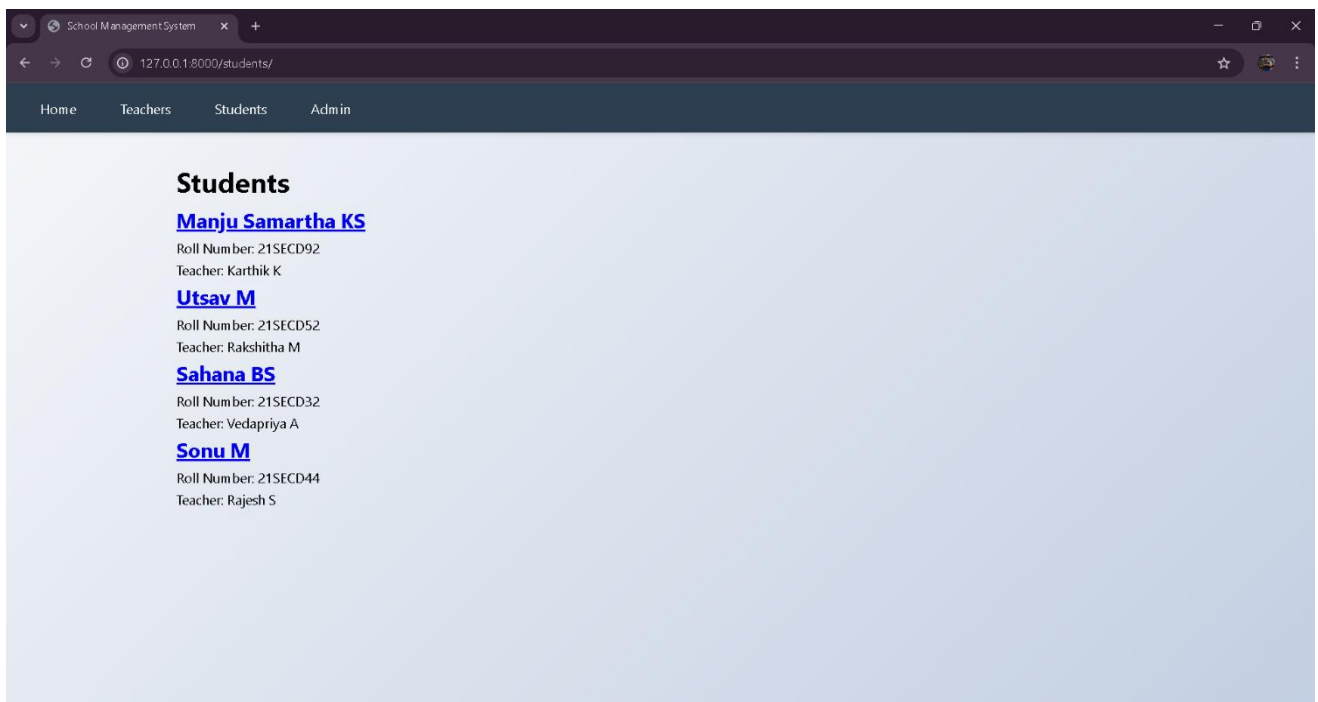
Output:



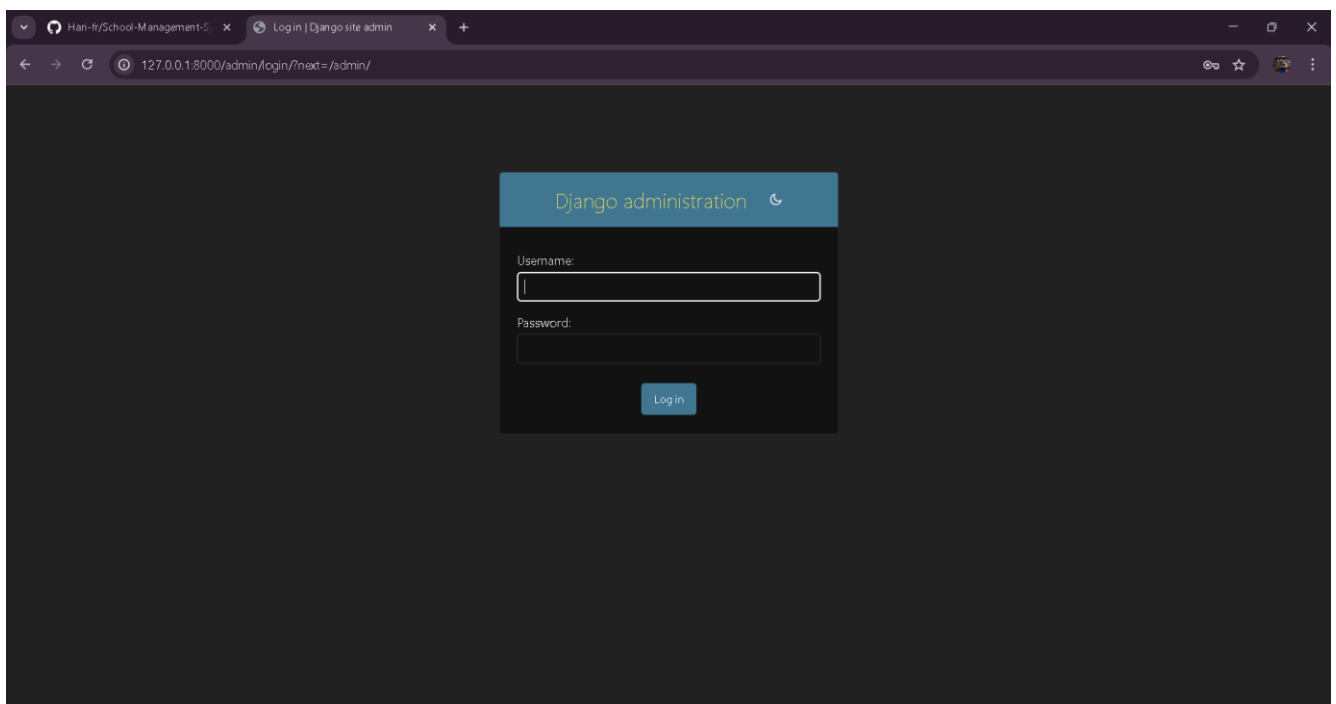
homepage



Teacher list and details page



Students list and details page



Admin login page

Browser tabs: Hari-It/School-Management-5, Add teacher | Django site admin

Address bar: 127.0.0.1:8000/admin/school/teacher/add/

Django administration

WELCOME, SRIHARI | [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > School > Teachers > Add teacher

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

SCHOOL

- Students + Add
- Teachers + Add

Add teacher

First name:

Last name:

Email:

Subject:

Hire date: Today |

Note: You are 5.5 hours ahead of server time.

Add teacher page

Browser tabs: Hari-It/School-Management-5, Add student | Django site admin

Address bar: 127.0.0.1:8000/admin/school/student/add/

Django administration

WELCOME, SRIHARI | [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > School > Students > Add student

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

SCHOOL

- Students + Add
- Teachers + Add

Add student

First name:

Last name:

Email:

Roll number:

Date of birth: Today |

Note: You are 5.5 hours ahead of server time.

Teacher:

Add student page