MYSORE UNIVERSITY SCHOOL OF ENGINEERING

Manasagangotri campus, Mysuru-570006

(Approved by AICTE, New Delhi)

UNIVERSITY OF MYSORE

Full Stack Development(21CD71) Assessment Report On:

"ONLINE COURSE ENROLLMENT SYSTEM "

Under the guidance :
Mr. Karthik M N
Assistant Professor,
Department of Computer Science & Design,
MUSE.

Submitted by:
YASHODA N
Reg No : 21SECD58

ONLINE COURSE ENROLLMENT SYSTEM - DJANGO PROJECT REPORT

INTRODUCTION

The **Online Course Enrollment System** is a Django-based web application that allows students to enroll in multiple courses seamlessly. The system provides a user friendly interface for course registration and enables viewing all enrolled students. Enrollment data is securely stored and managed via the **Django Admin Panel**. After successful enrollment , students are redirected to a confirmation page for a smooth experience.

KEY FEATURES:

✓ User-friendly course enrollment form

✓ Admin panel for managing students and courses

✓ Confirmation page after successful enrollment

✓ ListView to display available courses and enrolled students

TECHNOLOGIES USED

The project is developed using the following technologies:

- **Backend:** Django (Python Web Framework)
- **Frontend:** HTML, CSS
- **Database:** SQLite3
- **Development Environment:** VS Code
- **Version Control:** GitHub

PROJECT SETUP

Installation Steps:

Install Django:

pip install django

Create Django Project & App:

django-admin startprojectcourse_enrollment

cd course_enrollment

python manage.py startapp enrollments

Apply Migrations & Run Server:

python manage.py migrate

python manage.py runserver

FOLDER STRUCTURE:

course_enrollment/

├── enrollments/

│ ├── migrations/

│ ├── static/

│ │ ├── enrollments/style.css

│ ├── templates/

│ │ ├── enrollments/course_list.html

│ │ ├── enrollments/enrollment_form.html

│ │ ├── enrollments/success.html

│ ├── models.py

│ ├── views.py

│ ├── urls.py

├── course_enrollment/

├── db.sqlite3

├── manage.py

MODELS (DATABASE STRUCTURE)

The **Course** and **Student** models are used to store student and course data in the database with a **many-to-many relationship** for enrollment.

models.py file:

```
from django.db import models

class Course(models.Model):

name = models.CharField(max_length=200)

description = models.TextField()

def __str__(self):

return self.name

class Student(models.Model):

name = models.CharField(max_length=100)

email = models.EmailField(unique=True)

courses = models.ManyToManyField(Course, related_name="students")

def __str__(self):

return self.name
```

FORMS & VIEWS

The **Enrollment Form** collects student details and allows them to enroll in courses.

forms.py file:

```
from django import forms

from .models import Student

class EnrollmentForm(forms.ModelForm):

class Meta:
    model = Student
    fields = ['name', 'email', ‘courses’]
```

views.py file:

```
from django.shortcuts import render, redirect
from django.views.generic import ListView
from django.urls import reverse_lazy
from .models import Course, Student
from .forms import EnrollmentForm
class CourseListView(ListView):
    model = Course
    template_name = 'enrollments/course_list.html'
    context_object_name = 'courses'
class StudentListView(ListView):
    model = Student
    template_name = 'enrolled_students.html'
    context_object_name = 'students'
def enroll_student(request):
    if request.method == "POST":
        form = EnrollmentForm(request.POST)
      if form.is_valid():
            student = form.save(commit=False)
            student.save()
            form.save_m2m()
            return redirect(reverse_lazy("enrollment_success"))
     else:
```

```
        form = EnrollmentForm()
    return render(request, "enroll.html", {"form": form})
def enrollment_success(request):
    return render(request, 'enrollments/success.html')
```

Users are redirected to a success page after submitting the enrollment form.

URL CONFIGURATION

Project-level (course_enrollment/urls.py) and app-level (enrollments/urls.py) URL patterns are defined as follows:

```
# Project URLs (course_enrollment/urls.py)
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
path('admin/', admin.site.urls),
path('', include('enrollments.urls')),
]

# App URLs (enrollments/urls.py)
from django.urls import path
from .views import CourseListView, StudentListView, enroll_student, enrollment_success
urlpatterns = [
path('courses/', CourseListView.as_view(), name='course_list'),
path('students/', StudentListView.as_view(), name='enrolled_students'),
path('enroll/', enroll_student, name='enroll'),
path('success/', enrollment_success, name='enrollment_success'),
]
```

TEMPLATES

Templates (HTML Files):

- **course_list.html** → Landing page

enrollments > templates > enrollments > course_list.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Courses</title>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'enrollments/styles.css' %}">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&family=Montserrat:wght@700&family=Roboto:wght@400&display=swap"
</head>
<body>
<div class="container">
    <h2>Available Courses</h2>
    {% for course in courses %}
        <div class="course-card">
            <h3>{{ course.name }}</h3>
            <p>{{ course.description }}</p>
        </div>
    {% endfor %}
    <a href="{% url 'enroll' %}" class="button">Enroll in Courses</a>
</div>
</body>
</html>
```

- **enroll.html** → Course enrollment form

enrollments > templates > enroll.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Enroll</title>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'enrollments/styles.css' %}">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&family=Montserrat:wght@700&family=Roboto:wght@400&display=swap"
</head>
<body>
<div class="container">
    <h2>Enroll in a Course</h2>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Enroll</button>
    </form>
</div>
</body>
</html>
```

- **success.html** → Enrollment Successfully Completed message

enrollments > templates > enrollments > success.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Success</title>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'enrollments/styles.css' %}">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&family=Montserrat:wght@700&family=Roboto:wght@400&display=swap"
</head>
<body>
<div class="container">
    <h2>Enrollment Successful!</h2>
    <a href="{% url 'course_list' %}" class="button">Go back to Courses</a>
</div>
</body>
</html>
```

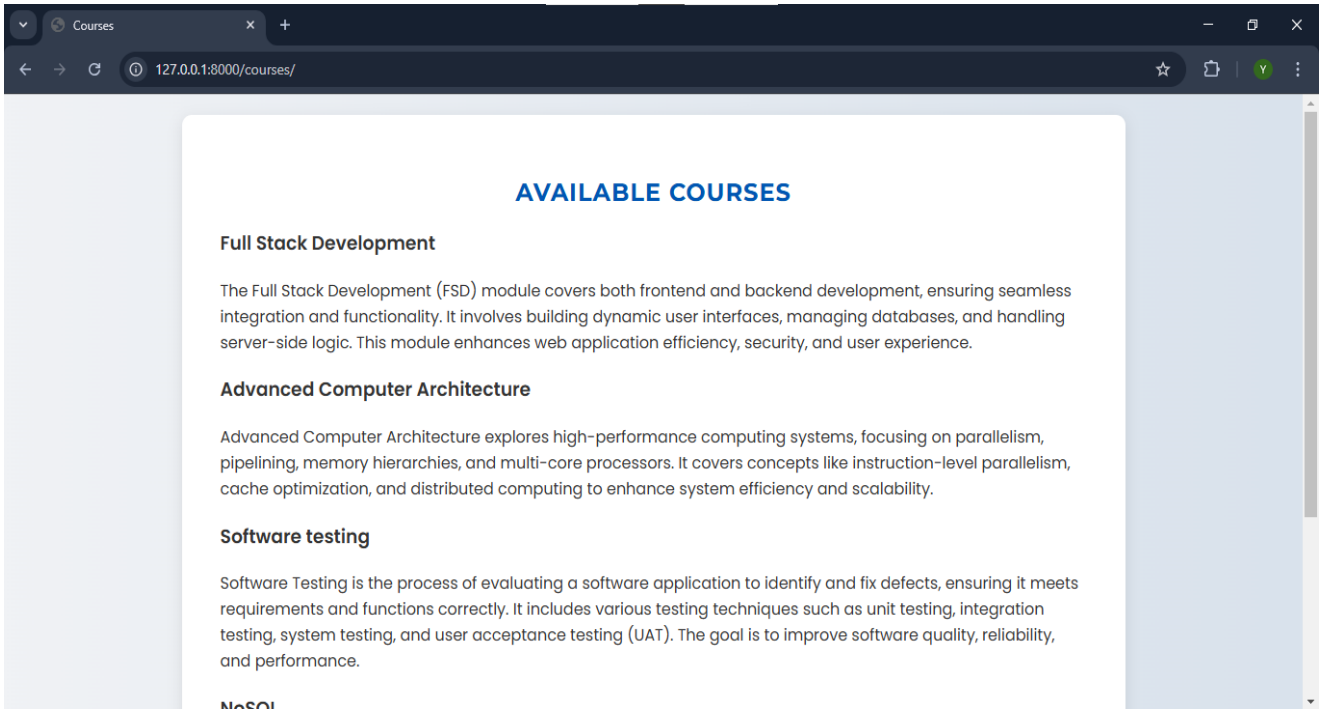
- **enrolled_students.html** → Displays the list of enrolled students and their courses

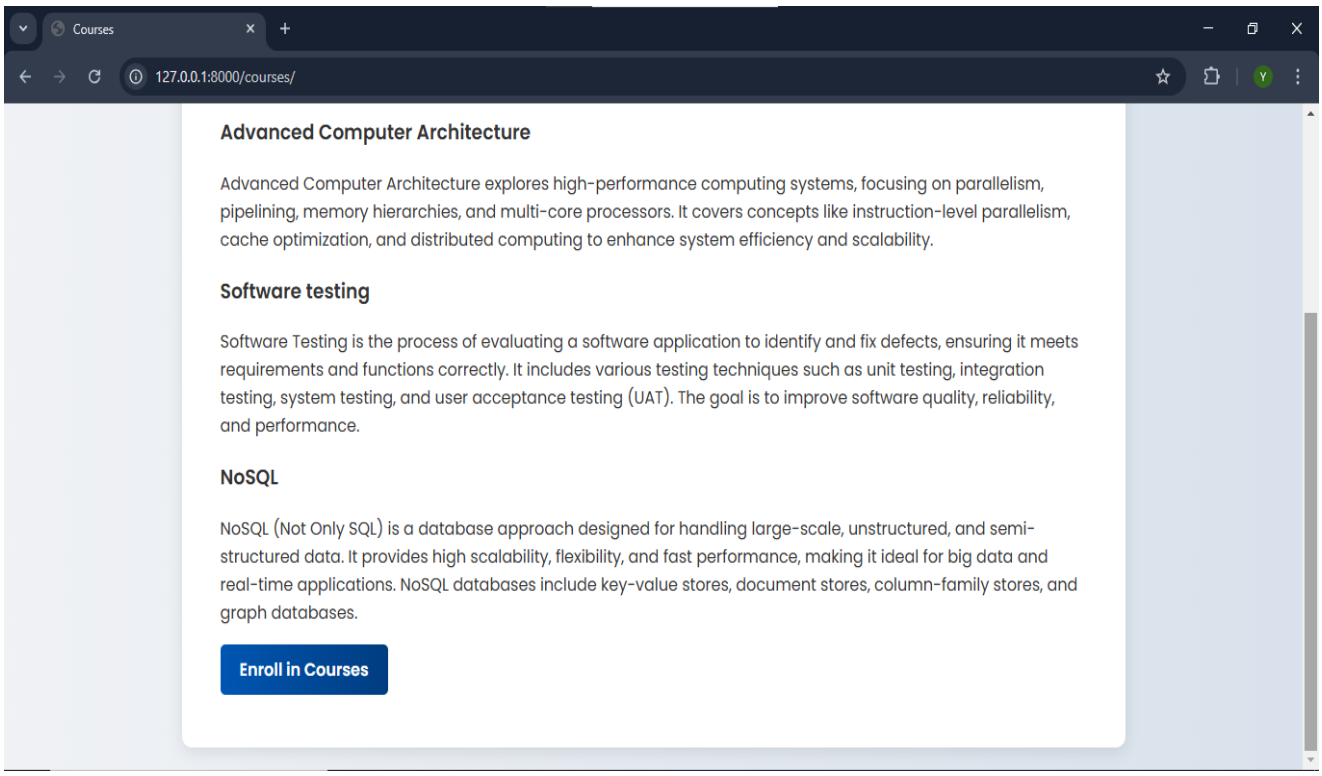
enrollments > templates > enrolled_students.html

```
<body>
    <div class="container">
        <h2>Enrolled Students</h2>
        <table>
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Email</th>
                    <th>Enrolled Courses</th>
                </tr>
            </thead>
            <tbody>
                {% for student in students %}
                <tr>
                    <td>{{ student.name }}</td>
                    <td>{{ student.email }}</td>
                    <td>
                        {% for course in student.courses.all %}
                            {{ course.name }}{% if not forloop.last %}, {% endif %}
                        {% empty %}
                            No courses enrolled.
                        {% endfor %}
                    </td>
                </tr>
                {% empty %}
                <tr><td colspan="3">No students enrolled yet.</td></tr>
                {% endfor %}
            </tbody>
        </table>
```

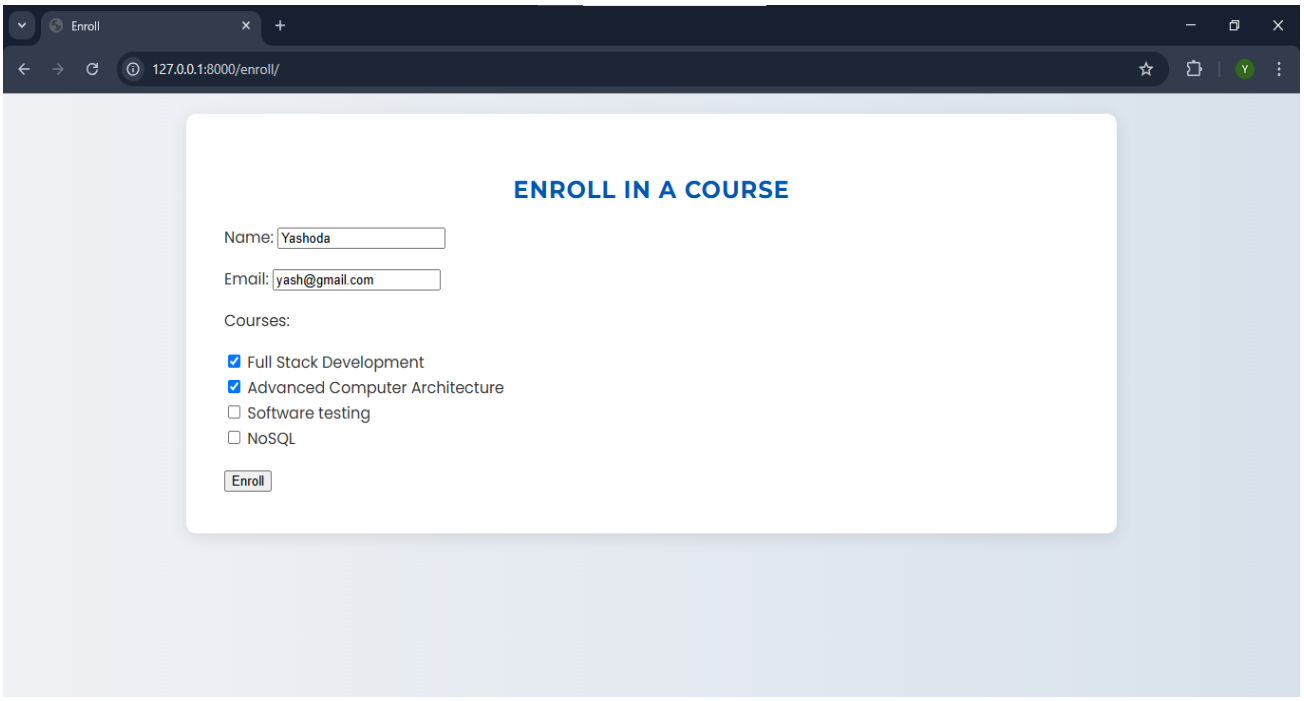
SCREENSHOTS

Home Page Screenshot → Screenshot of **course_list.html**, which serves as the landing page displaying all available courses

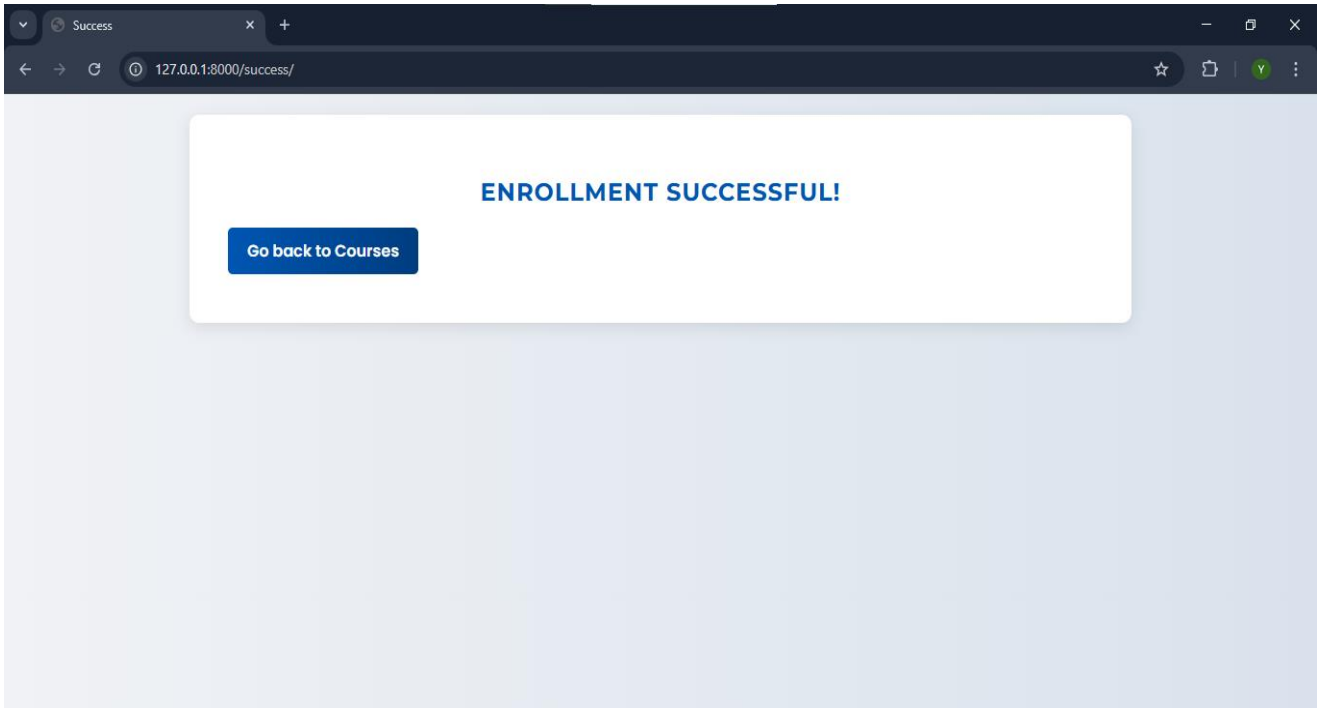




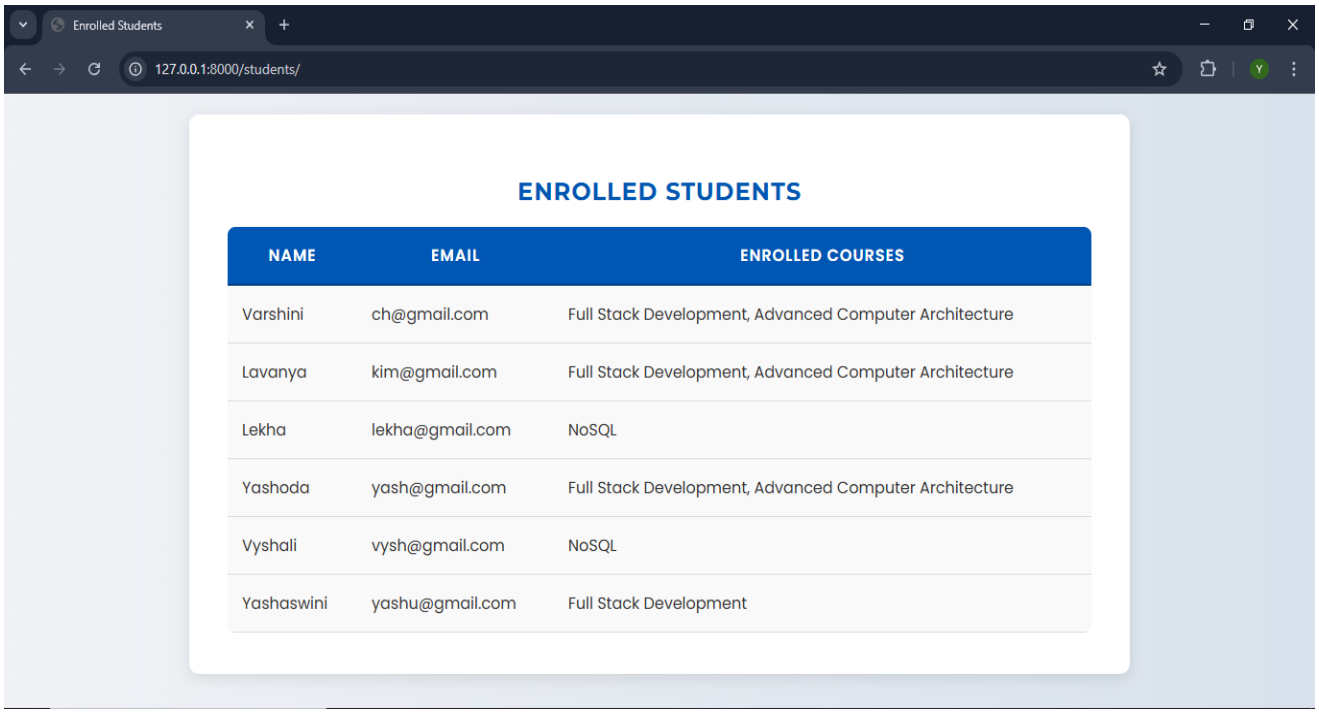
Course Enrollment Form Screenshot → Screenshot of enroll.html, which contains the form for students to enroll in courses



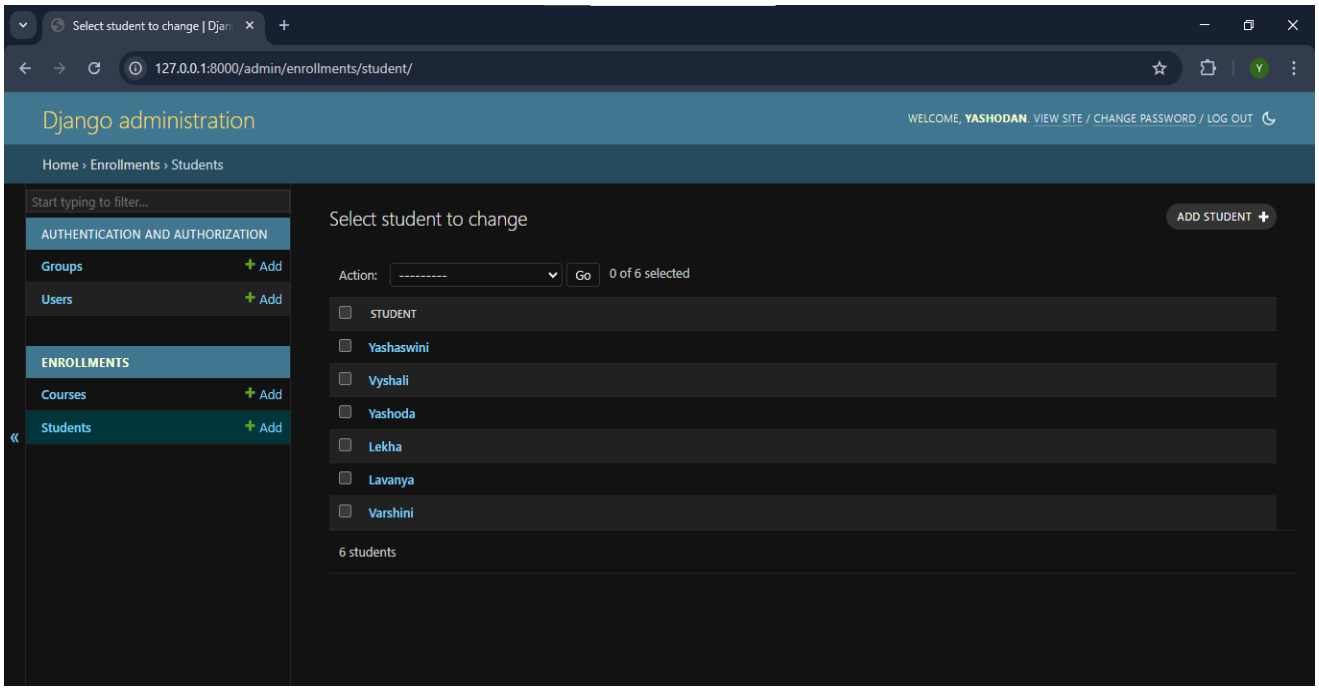
Success Page ScreenShot → Screenshot of success.html, which confirms successful course enrollment



Enrolled Students Page Screenshot → Screenshot of enrolled_students.html, which displays the list of students along with their enrolled courses



Admin Panel Screenshot → Screenshot of Django Admin Panel, showing the management of students and courses



Implement reverse_lazy() to redirect students to Confirmation page

enrollments > views.py > enroll_student

```python
from django.urls import reverse_lazy
from .models import Course, Student
from .forms import EnrollmentForm

class CourseListView(ListView):
    model = Course
    template_name = 'enrollments/course_list.html'
    context_object_name = 'courses'

class StudentListView(ListView):
    model = Student
    template_name = 'enrolled_students.html'  # This needs to be created
    context_object_name = 'students'

def enroll_student(request):
    if request.method == "POST":
        form = EnrollmentForm(request.POST)
        if form.is_valid():
            student = form.save(commit=False)
            student.save()  # ✅ Save student first
            form.save_m2m()  # ✅ Save the many-to-many relationship
            return redirect(reverse_lazy("enrollment_success"))  # ✅ Redirect after success
    else:
        form = EnrollmentForm()

    return render(request, "enroll.html", {"form": form})

def enrollment_success(request):
    return render(request, 'enrollments/success.html')

```