



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 startproject course_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
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Courses</title>
7      {% load static %}
8      <link rel="stylesheet" type="text/css" href="{% static 'enrollments/styles.css' %}">
9      <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&family=Montserrat:wght@700&family=Roboto:wght@400&display=swap"
10 </head>
11 <body>
12 <div class="container">
13     <h2>Available Courses</h2>
14     {% for course in courses %}
15         <div class="course-card">
16             <h3>{{ course.name }}</h3>
17             <p>{{ course.description }}</p>
18         </div>
19     {% endfor %}
20     <a href="{% url 'enroll' %}" class="button">Enroll in Courses</a>
21 </div>
22 </body>
23 </html>
```

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

```
enrollments > templates > <> enroll.html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Enroll</title>
7      {% load static %}
8      <link rel="stylesheet" type="text/css" href="{% static 'enrollments/styles.css' %}">
9      <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&family=Montserrat:wght@700&family=Roboto:wght@400&display=swap"
10 </head>
11 <body>
12 <div class="container">
13     <h2>Enroll in a Course</h2>
14     <form method="post">
15         {% csrf_token %}
16         {{ form.as_p }}
17         <button type="submit">Enroll</button>
18     </form>
19 </div>
20 </body>
21 </html>
```

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

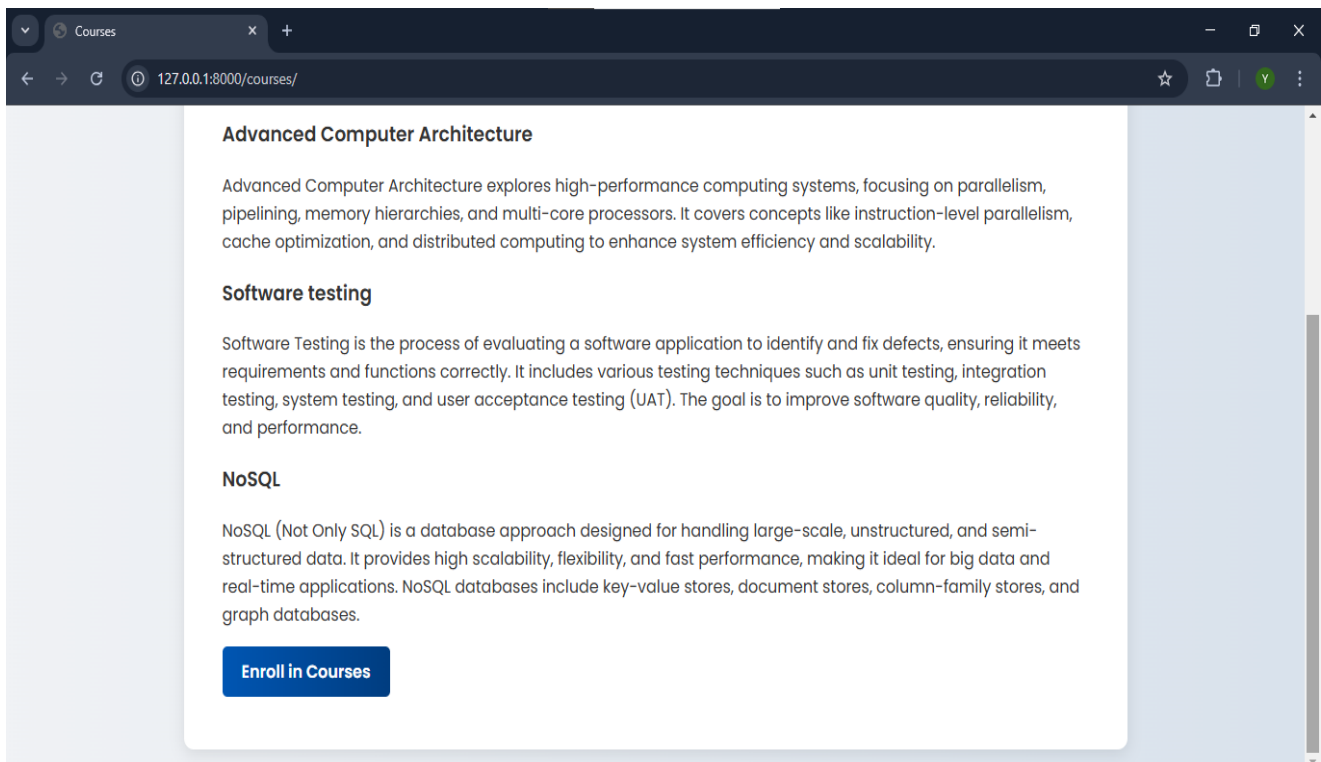
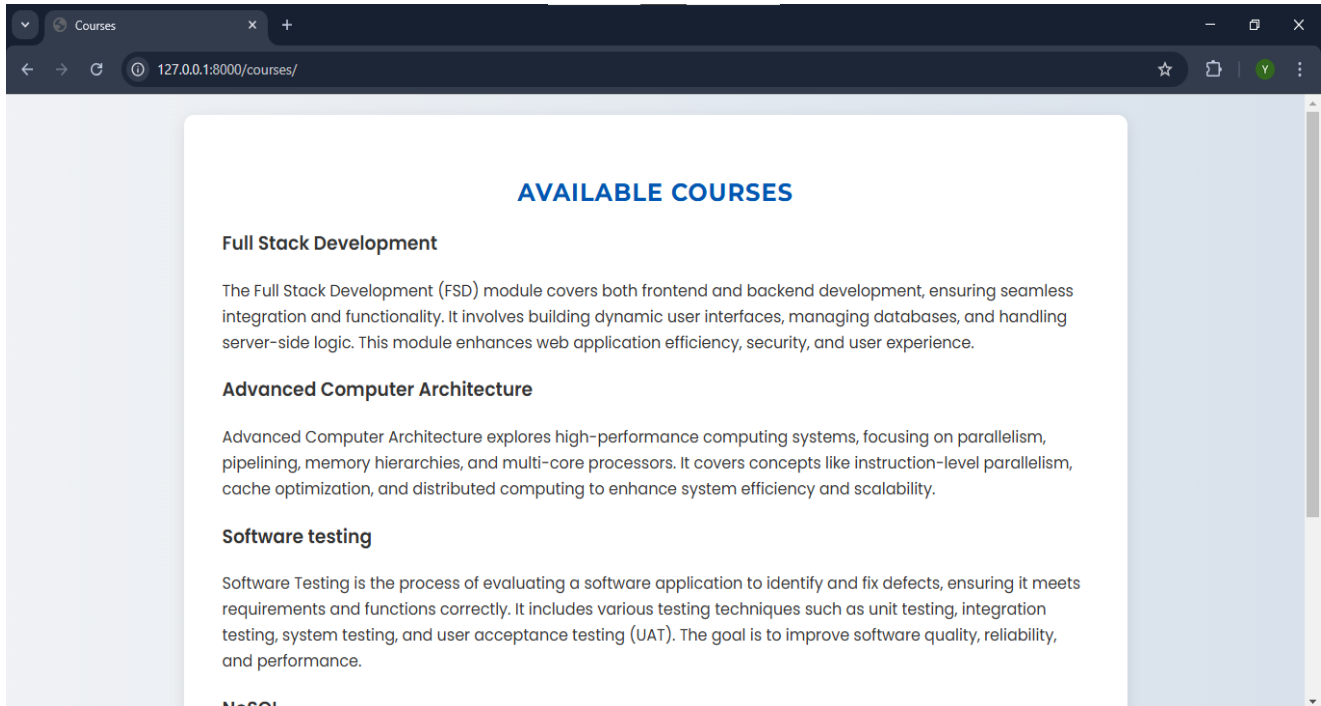
```
enrollments > templates > enrollments > success.html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Success</title>
7      {% load static %}
8      <link rel="stylesheet" type="text/css"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

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

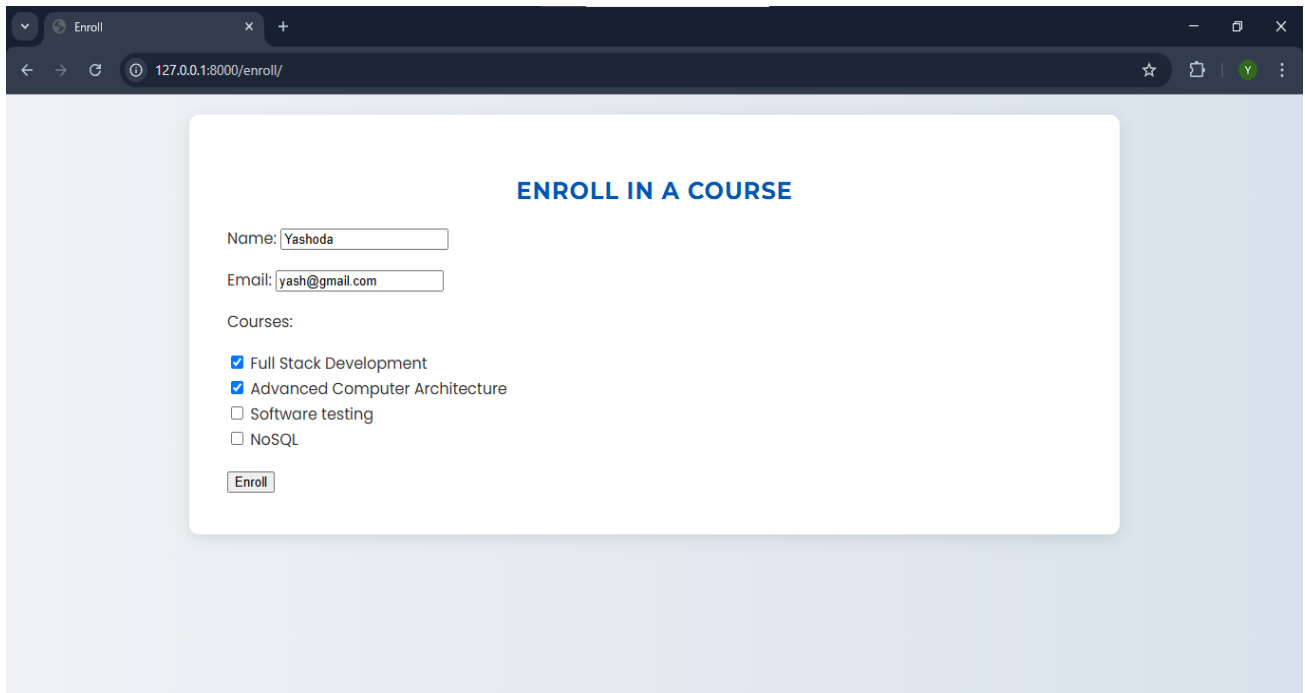
```
enrollments > templates > enrolled_students.html
11 <body>
12     <div class="container">
13         <h2>Enrolled Students</h2>
14         <table>
15             <thead>
16                 <tr>
17                     <th>Name</th>
18                     <th>Email</th>
19                     <th>Enrolled Courses</th>
20                 </tr>
21             </thead>
22             <tbody>
23                 {% for student in students %}
24                 <tr>
25                     <td>{{ student.name }}</td>
26                     <td>{{ student.email }}</td>
27                     <td>
28                         {% for course in student.courses.all %}
29                             {{ course.name }}{% if not forloop.last %}, {% endif %}
30                         {% empty %}
31                             No courses enrolled.
32                         {% endfor %}
33                     </td>
34                 </tr>
35                 {% empty %}
36                 <tr><td colspan="3">No students enrolled yet.</td></tr>
37                 {% endfor %}
38             </tbody>
39         </table>
```


SCREENSHOTS

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



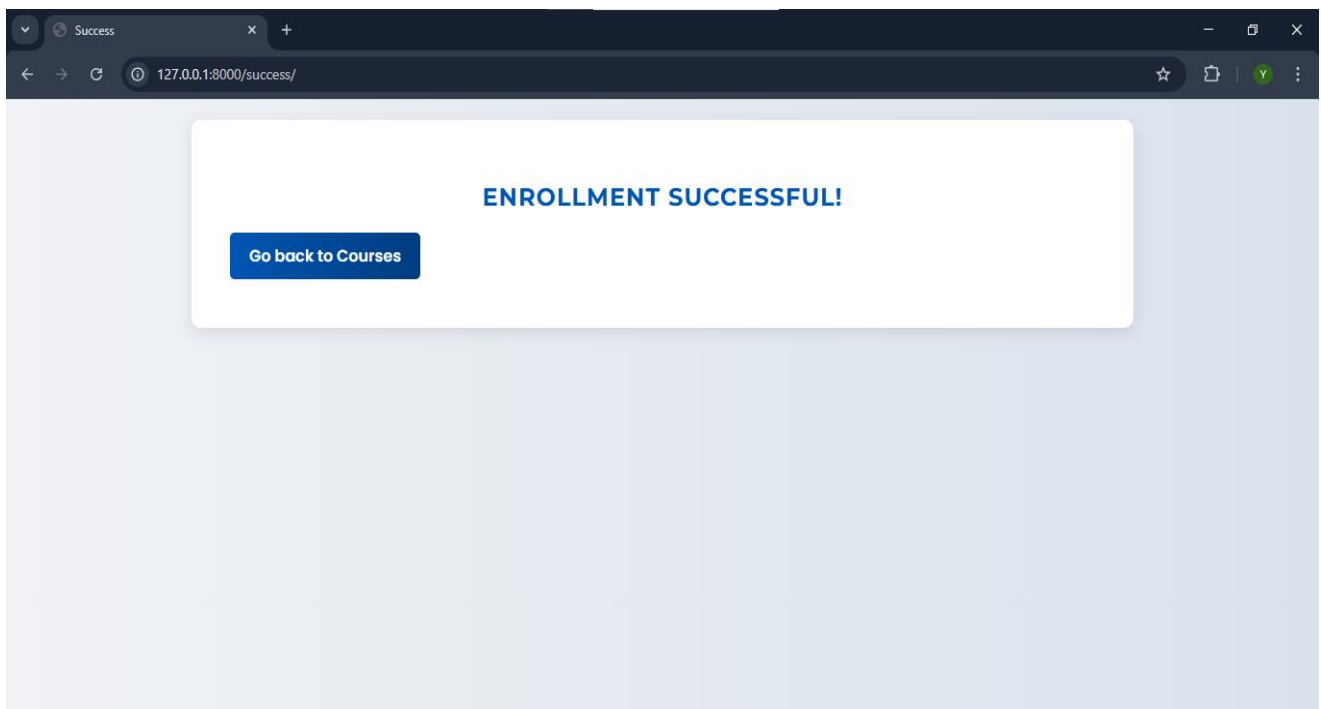
Course Enrollment Form Screenshot → Screenshot of enroll.html, which contains the form for students to enroll in courses



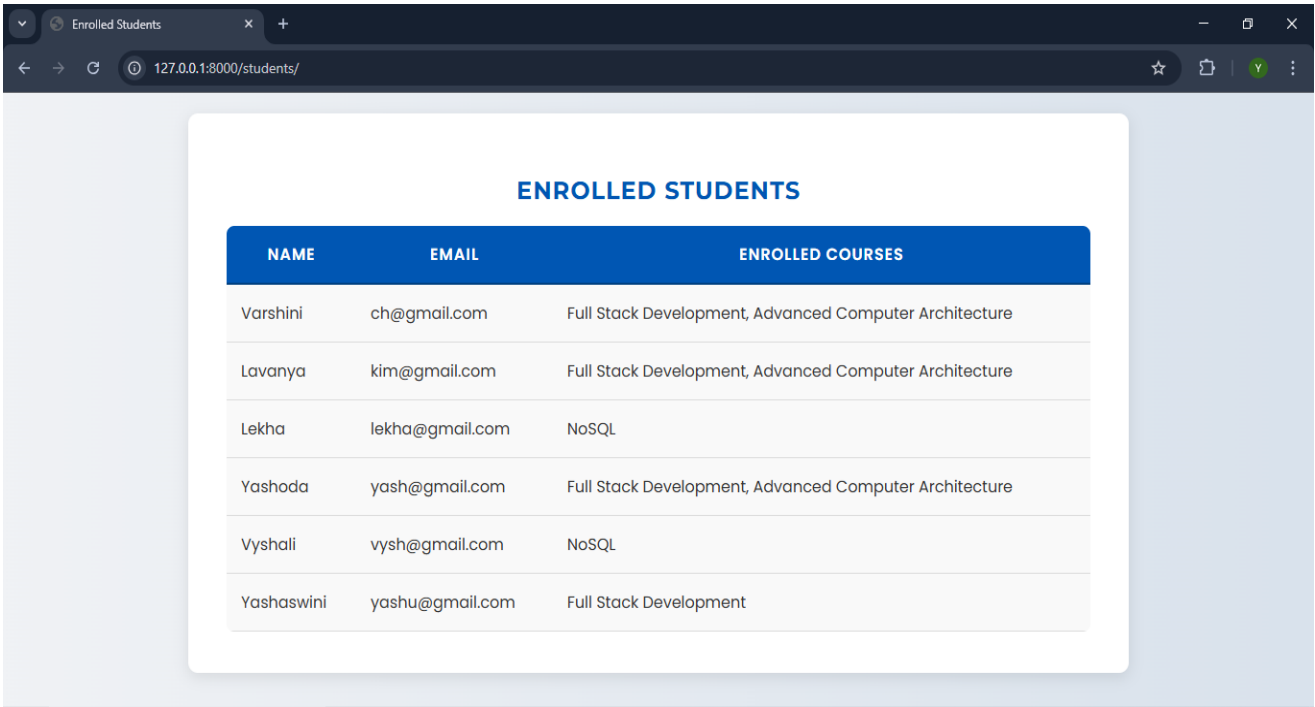
A screenshot of a web browser showing the 'ENROLL IN A COURSE' form. The browser's address bar displays '127.0.0.1:8000/enroll/'. The form is centered on a light blue background and contains the following elements:

- ENROLL IN A COURSE** (Section Header)
- Name:**
- Email:**
- Courses:**
 - ☒ Full Stack Development
 - ☒ Advanced Computer Architecture
 - ☐ Software testing
 - ☐ NoSQL
- Enroll** (Submit Button)

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

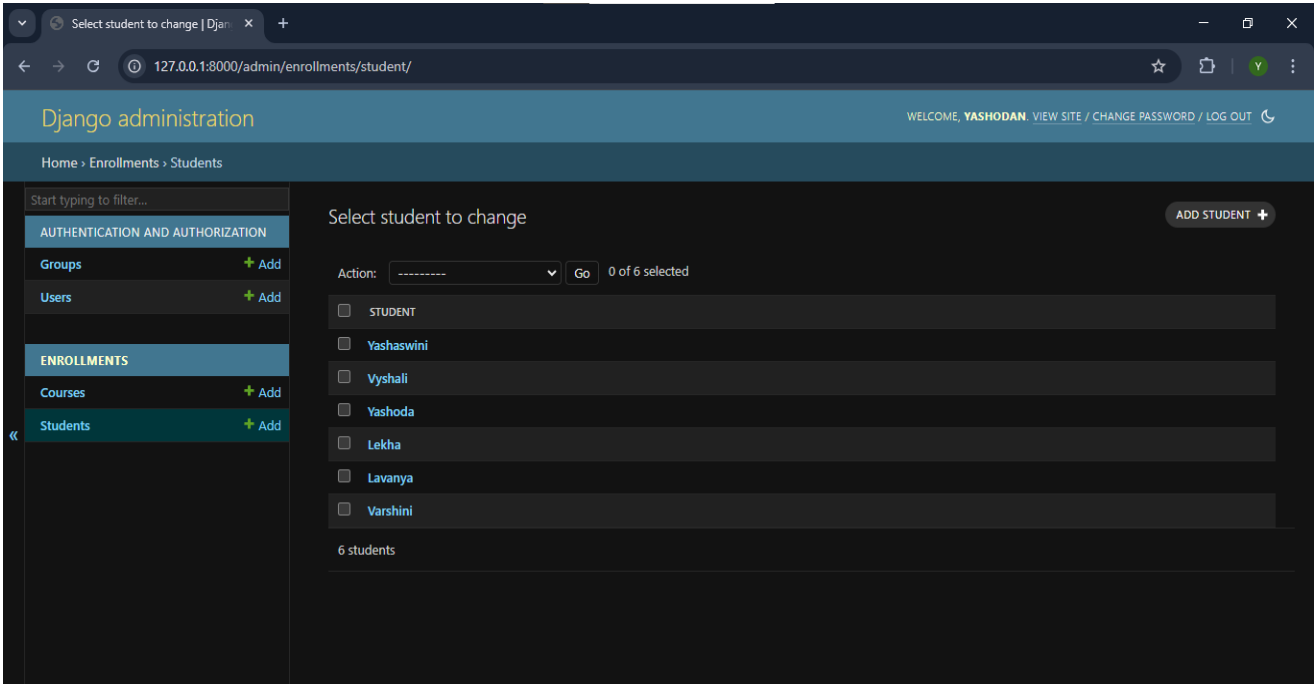


Enrolled Students Page Screenshot → Screenshot of enrolled_students.html, which displays the list of students along with their enrolled courses



NAME	EMAIL	ENROLLED COURSES
Varshini	ch@gmail.com	Full Stack Development, Advanced Computer Architecture
Lavanya	kim@gmail.com	Full Stack Development, Advanced Computer Architecture
Lekha	lekha@gmail.com	NoSQL
Yashoda	yash@gmail.com	Full Stack Development, Advanced Computer Architecture
Vyshali	vysh@gmail.com	NoSQL
Yashaswini	yashu@gmail.com	Full Stack Development

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



Django administration

WELCOME, YASHODAN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Enrollments > Students

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

ENROLLMENTS

- Courses + Add
- Students + Add

Select student to change

ADD STUDENT +

Action: ----- Go 0 of 6 selected

- ☐ STUDENT
- ☐ Yashaswini
- ☐ Vyshali
- ☐ Yashoda
- ☐ Lekha
- ☐ Lavanya
- ☐ Varshini

6 students

Implement `reverse_lazy()` to redirect students to Confirmation page

```
enrollments > views.py > enroll_student
3 from django.urls import reverse_lazy
4 from .models import Course, Student
5 from .forms import EnrollmentForm
6
7 class CourseListView(ListView):
8     model = Course
9     template_name = 'enrollments/course_list.html'
10    context_object_name = 'courses'
11
12 class StudentListView(ListView):
13     model = Student
14     template_name = 'enrolled_students.html' # This needs to be created
15     context_object_name = 'students'
16
17 def enroll_student(request):
18     if request.method == "POST":
19         form = EnrollmentForm(request.POST)
20         if form.is_valid():
21             student = form.save(commit=False)
22             student.save() # ✓ Save student first
23             form.save_m2m() # ✓ Save the many-to-many relationship
24             return redirect(reverse_lazy("enrollment_success")) # ✓ Redirect after success
25         else:
26             form = EnrollmentForm()
27
28     return render(request, "enroll.html", {"form": form})
29
30 def enrollment_success(request):
31     return render(request, 'enrollments/success.html')
32
```