



# **MYSORE UNIVERSITY SCHOOL OF ENGINEERING**

Manasagangotri campus, Mysuru-570006  
(Approved by AICTE, New Delhi)



## **UNIVERSITY OF MYSORE**

**Full Stack Development(21CD71) Assessment Report On:**

**“COURSE ENROLLMENT SYSTEM”**

**Under the guidance :**

**Mr. Karthik M N**  
Assistant Professor,  
Department of Computer  
Science & Design,  
MUSE.

**Submitted by:**

**Vyshali M**  
Reg No : 21SECD56

# **COURSE ENROLLMENT SYSTEM – DJANGO PROJECT REPORT**

## **INTRODUCTION**

The Course Enrollment System is a web-based application built using Django. The application enables administrators to add courses via the Django Admin panel. Registered users log in to create/update their profiles and then enroll in the available courses. Once enrolled, the system displays a success page, and the enrolled courses appear on the user's profile in a read-only format (course enrollment modifications can only be performed by an administrator). The website features a professional design with a radiant black-blue theme and smooth transitions.

## **KEY FEATURES:**

- ✓ User authentication with login and logout functionality
- ✓ Profile creation with fields: current year of study, LinkedIn, GitHub, phone number, address, and guardian phone number
- ✓ Course enrollment form available only after profile creation
- ✓ Enrollment success page after course registration
- ✓ Courses page displaying only course names and the names of enrolled students
- ✓ Professional UI with a radiant black-blue theme and smooth transitions
- ✓ Use of `reverse_lazy()` to handle redirections post submission

## **TECHNOLOGIES USED**

The project is developed using the following technologies:

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

## **PROJECT SETUP & FOLDER STRUCTURE**

### **Installation Steps:**

#### **1. Install Django:**

```
pip install Django
```

## 2. Create Django Project & App:

```
django-admin startproject projectbyvyshali
cd projectbyvyshali
python manage.py startapp enrollment
```

## 3. Apply Migrations & Run Server:

```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

## FOLDER STRUCTURE:

projectbyvyshali/

```
|— enrollment/
|   |— migrations/
|   |— static/
|   |   |— enrollment/
|   |   |   |— style.css
|   |— templates/
|   |   |— enrollment/
|   |   |   |— base.html
|   |   |   |— home.html
|   |   |   |— login.html
|   |   |   |— profile_form.html
|   |   |   |— profile.html
|   |   |   |— profile_update.html
|   |   |   |— enroll.html
|   |   |   |— success.html
|   |   |   |— courses.html
|   |— models.py
|   |— forms.py
|   |— views.py
|   |— urls.py
|— projectbyvyshali/
|— db.sqlite3
|— manage.py
```

## MODELS (DATABASE STRUCTURE)

The project utilizes three primary models defined in **models.py**:

- **Course Model:**

Stores course details.

```
class Course(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField(blank=True)
    def __str__(self):
        return self.name
```

- **Student Model:**

Records student information along with a many-to-many relationship to courses.

```
class Student(models.Model):
    user = models.OneToOneField(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE, null=True, blank=True)
    name = models.CharField(max_length=100)
    email = models.EmailField(unique=True)
    courses = models.ManyToManyField(Course, related_name='students', blank=True)
    def __str__(self):
        return self.name
```

- **Profile Model:**

Captures additional user details required during profile creation.

```
class Profile(models.Model):
    user = models.OneToOneField(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE)
    current_year_of_study = models.CharField(max_length=20)
    linkedin = models.URLField(blank=True, null=True)
    github = models.URLField(blank=True, null=True)
    phone_number = models.CharField(max_length=20, blank=True)
    address = models.TextField(blank=True)
    guardian_phone_number = models.CharField(max_length=20, blank=True)
    def __str__(self):
        return f'{self.user.username}'s Profile"
```

## FORMS & VIEWS

- **Forms (forms.py):**

Defines forms for profile creation and course enrollment.

```
from django import forms
```

```
from .models import Profile, Student, Course
```

```
class ProfileForm(forms.ModelForm):
```

```
    class Meta:
```

```
        model = Profile
```

```
        fields = ['current_year_of_study', 'linkedin', 'github', 'phone_number', 'address',  
'guardian_phone_number']
```

```
class EnrollmentForm(forms.ModelForm):
```

```
    courses = forms.ModelMultipleChoiceField(  
        queryset=Course.objects.all(),
```

```
        widget=forms.CheckboxSelectMultiple,
```

```
        help_text="Select the courses you want to enroll in."  
    )
```

```
    class Meta:
```

```
        model = Student
```

```
        fields = ['name', 'email', 'courses']
```

- **Views (views.py):**

Handles the user flows for home, login, profile (creation, display, update), enrollment, and course listing.

```
from django.shortcuts import render, redirect
```

```
from django.contrib.auth.decorators import login_required
```

```
from django.views.generic import CreateView, TemplateView, UpdateView, ListView
```

```
from django.urls import reverse_lazy
```

```
from django.contrib.auth.mixins import LoginRequiredMixin
```

```
from django.contrib.auth.views import LoginView, LogoutView
```

```
from .models import Profile, Student, Course
```

```
from .forms import ProfileForm, EnrollmentForm
```

```
class HomePageView(TemplateView):
```

```
    template_name = "enrollment/home.html"
```

```
class CustomLoginView(LoginView):
```

```
    template_name = "enrollment/login.html"
```

```

@login_required
def profile_view(request):
    try:
        profile = request.user.profile
    except Profile.DoesNotExist:
        return redirect('profile_create')
    student = request.user.student if hasattr(request.user, 'student') else None
    return render(request, "enrollment/profile.html", {'profile': profile, 'student': student})

```

```

@login_required
def profile_create_view(request):
    if hasattr(request.user, 'profile'):
        return redirect('profile')
    if request.method == "POST":
        form = ProfileForm(request.POST)
        if form.is_valid():
            profile = form.save(commit=False)
            profile.user = request.user
            profile.save()
            return redirect('profile')
    else:
        form = ProfileForm()
    return render(request, "enrollment/profile_form.html", {'form': form})

```

```

class ProfileUpdateView(LoginRequiredMixin, UpdateView):
    model = Profile
    form_class = ProfileForm
    template_name = "enrollment/profile_update.html"
    success_url = reverse_lazy('profile')
    def get_object(self):
        return self.request.user.profile

```

```

class EnrollmentCreateView(LoginRequiredMixin, CreateView):
    model = Student
    form_class = EnrollmentForm
    template_name = "enrollment/enroll.html"
    success_url = reverse_lazy('enrollment_success')
    def dispatch(self, request, *args, **kwargs):
        if not hasattr(request.user, 'profile'):
            return redirect('profile_create')
        if hasattr(request.user, 'student'):
            return redirect('profile')
        return super().dispatch(request, *args, **kwargs)

```

```

def form_valid(self, form):
    form.instance.user = self.request.user
    return super().form_valid(form)

class EnrollmentSuccessView(TemplateView):
    template_name = "enrollment/success.html"

class CourseListView(ListView):
    model = Course
    template_name = "enrollment/courses.html"
    context_object_name = "courses"

```

**Note: reverse\_lazy() is used in EnrollmentCreateView and ProfileUpdateView to redirect users after successful submissions.**

## **URL CONFIGURATION**

- **Project-level URLs (projectbyvyshali/urls.py):**

```

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

```

```

urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('enrollment.urls')),
]

```

- **App-level URLs (enrollment/urls.py):**

```

from django.urls import path
from django.contrib.auth.views import LogoutView
from .views import (
    HomePageView, CustomLoginView, profile_create_view, profile_view,
    ProfileUpdateView, EnrollmentCreateView, EnrollmentSuccessView, CourseListView
)
urlpatterns = [
    path("", HomePageView.as_view(), name='home'),
    path('login/', CustomLoginView.as_view(), name='login'),
    path('logout/', LogoutView.as_view(next_page='home'), name='logout'),
    path('profile/create/', profile_create_view, name='profile_create'),
    path('profile/', profile_view, name='profile'),
    path('profile/update/', ProfileUpdateView.as_view(), name='profile_update'),
    path('enroll/', EnrollmentCreateView.as_view(), name='enroll'),
    path('enroll/success/', EnrollmentSuccessView.as_view(), name='enrollment_success'),
    path('courses/', CourseListView.as_view(), name='courses'),
]

```

## TEMPLATES

The following templates are used:

- **base.html** – Contains the common header (with navigation links for Home, Profile, Enroll, Courses, Login/Logout) and footer.

```
{% load static %}
<link rel="stylesheet" type="text/css" href="{% static 'enrollment/style.css' %}">

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>ProjectbyVyshali - {% block title %}{% endblock %}</title>
  <link rel="stylesheet" type="text/css" href="{% static 'enrollment/style.css' %}">
</head>
<body>
  <header>
    <h1>Online Course Enrollment</h1>
    <nav>
      <a href="{% url 'home' %}">Home</a> |
      {% if user.is_authenticated %}
      <a href="{% url 'profile' %}">Profile</a> |
      <form action="{% url 'logout' %}" method="post" style="display:inline;">
        {% csrf_token %}
        <button type="submit" style="background:none;border:none;color:#fff;cursor:pointer;">Logout</button>
      </form>
      {% else %}
      <a href="{% url 'login' %}">Login</a>
      {% endif %} |
      <a href="{% url 'courses' %}">Courses</a>
    </nav>
  </header>
  <div class="container">
    {% block content %}{% endblock %}
  </div>
  <footer>
    <p>&copy; 2025 ProjectbyVyshali</p>
  </footer>
</body>
</html>
```

- **home.html** – Landing page with a welcome message.

```
{% extends 'enrollment/base.html' %}
{% block title %}Home{% endblock %}
{% block content %}
  <h2>Welcome to the Online Course Enrollment System</h2>
  <p>Please login to continue.</p>
{% endblock %}
```

- **login.html** – User login form.

```
{% extends 'enrollment/base.html' %}
{% block title %}Login{% endblock %}
{% block content %}
  <h2>Login</h2>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Login</button>
  </form>
{% endblock %}
```

- **profile\_form.html** – Profile creation form for new users.

```
{% extends 'enrollment/base.html' %}
{% block title %}Create Profile{% endblock %}
{% block content %}
  <h2>Create Your Profile</h2>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Save Profile</button>
  </form>
{% endblock %}
```



- **profile.html** – Displays profile details along with enrolled courses; includes an “Edit Profile” button.

```
{% extends 'enrollment/base.html' %}
{% block title %}Profile{% endblock %}
{% block content %}
<h2>Your Profile</h2>
<p><strong>Email:</strong> {{ request.user.email }}</p>
<p><strong>Current Year of Study:</strong> {{ profile.current_year_of_study }}</p>
<p><strong>LinkedIn:</strong>
  {% if profile.linkedin %}
    <a href="{{ profile.linkedin }}" target="_blank">{{ profile.linkedin }}</a>
  {% else %}
    N/A
  {% endif %}
</p>
<p><strong>GitHub:</strong>
  {% if profile.github %}
    <a href="{{ profile.github }}" target="_blank">{{ profile.github }}</a>
  {% else %}
    N/A
  {% endif %}
</p>
<p><strong>Phone Number:</strong> {{ profile.phone_number|default:"N/A" }}</p>
<p><strong>Address:</strong> {{ profile.address|default:"N/A" }}</p>
<p><strong>Guardian Phone Number:</strong> {{ profile.guardian_phone_number|default:"N/A" }}</p>
<hr>
<h3>Enrolled Courses</h3>
{% if student %}
  <ul>
    {% for course in student.courses.all %}
      <li>{{ course.name }}</li>
    {% empty %}
      <li>No courses enrolled.</li>
    {% endfor %}
  </ul>
{% else %}
  <p>You have not enrolled in any courses yet.</p>
  <a href="{% url 'enroll' %}" class="btn">Enroll Now</a>
{% endif %}
<p><a href="{% url 'profile_update' %}" class="btn">Edit Profile</a></p>
{% endblock %}
```

- **profile\_update.html** – Allows users to update their profile information.

```
{% extends 'enrollment/base.html' %}
{% block title %}Enroll in Courses{% endblock %}
{% block content %}
<h2>Enroll in Courses</h2>
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit">Submit Enrollment</button>
</form>
{% endblock %}
```

- **enroll.html** – Course enrollment form where users select courses via checkboxes.

```
{% extends 'enrollment/base.html' %}
{% block title %}Enroll in Courses{% endblock %}
{% block content %}
<h2>Enroll in Courses</h2>
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit">Submit Enrollment</button>
</form>
{% endblock %}
```

- **success.html** – Confirmation page displayed after successful enrollment.

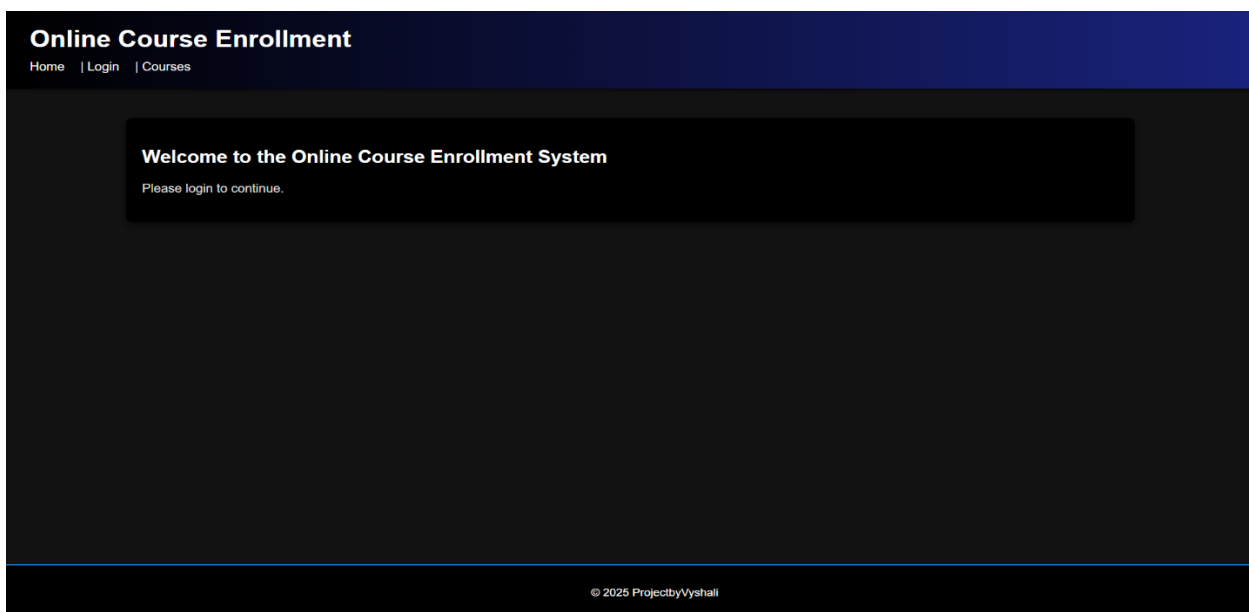
```
{% extends 'enrollment/base.html' %}
{% block title %}Enrollment Successful{% endblock %}
{% block content %}
    <h2>Enrollment Successful</h2>
    <p>Thank you for enrolling in your selected courses.</p>
    <a href="{% url 'profile' %}">Go to Profile</a>
{% endblock %}
```

- **courses.html** – Lists available courses along with the names of enrolled students.

```
{% extends 'enrollment/base.html' %}
{% block title %}Available Courses{% endblock %}
{% block content %}
    <h2>Available Courses</h2>
    <ul>
        {% for course in courses %}
            <li>
                <h3>{{ course.name }}</h3>
                <ul>
                    {% for student in course.students.all %}
                        <li>{{ student.name }}</li>
                    {% empty %}
                        <li>No students enrolled yet.</li>
                    {% endfor %}
                </ul>
            </li>
        {% endfor %}
    </ul>
{% endblock %}
```

## Screenshots:

### 1.home page



2. Login page

Online Course Enrollment

[Home](#) | [Login](#) | [Courses](#)

Login

Username:

Password:

Login

© 2025 ProjectbyVyshall

3.Profile Creation page

Online Course Enrollment

[Home](#) | [Profile](#) | [Logout](#) | [Courses](#)

Create Your Profile

Current year of study:

Linkedin:

Github:

Phone number:

Address:

Guardian phone number:

Save Profile

© 2025 ProjectbyVyshall

## 4.Profile display page

### Online Course Enrollment

Home | Profile | Logout | Courses

#### Your Profile

Email:

Current Year of Study: 4TH

LinkedIn: <http://example.com>

GitHub: <http://example.com>

Phone Number: 12346789

Address: BOGADI, MYSORE.

Guardian Phone Number: 8457966252

---

#### Enrolled Courses

You have not enrolled in any courses yet.

Enroll Now

Edit Profile

© 2025 ProjectbyVyshali

## 5.Enrollment form and Success page:

### Online Course Enrollment

Home | Profile | Logout | Courses

#### Enroll in Courses

Name:

Email:

Courses:

☒ Software Testing

☐ NoSQL

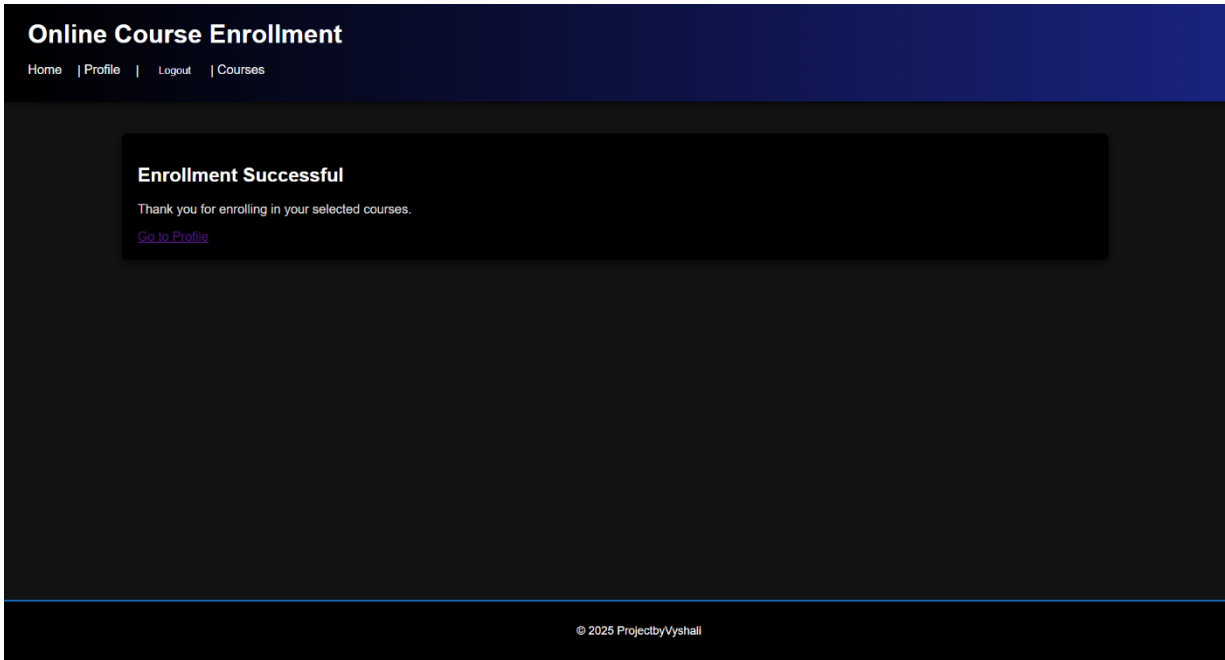
☐ Advanced Computer Architecture

☐ Full Stack Development

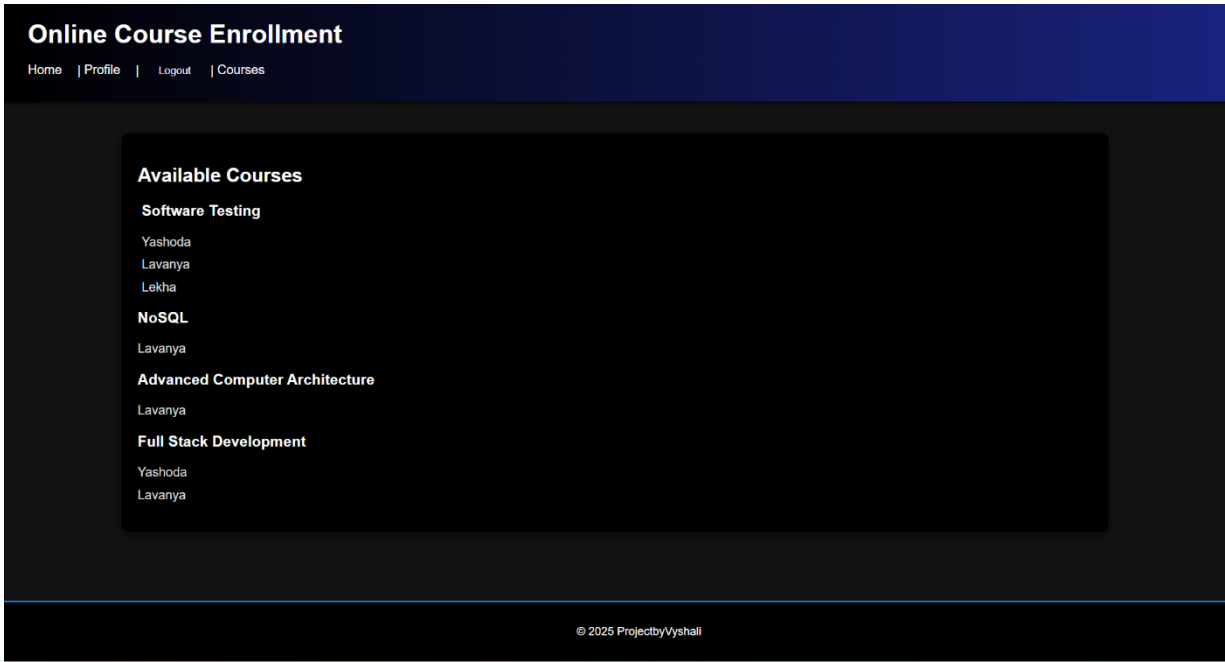
Select the courses you want to enroll in.

Submit Enrollment

© 2025 ProjectbyVyshali




6.Course page displaying course names and enrolled students



## 7.Django admin panel


Django administration

WELCOME, **VYSHAL.M** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#) 

Site administration


AUTHENTICATION AND AUTHORIZATION

Groups

+ Add 

Change


Users

+ Add 

Change


ENROLLMENT

Courses

+ Add 


Change

Profiles

+ Add 

Change

Students

+ Add 

Change

Recent actions

My actions

+ Karthik

User

+ Yashaswini

User

+ Varshini

User

+ Lekha

User

✖ Yashoda

Student

✖ Lavanya

Student

 Lavanya's Profile

Profile

✖ Machine Learning

Course

+ Full Stack Development

Course

+ Advanced Computer Architecture

Course

