

Course List Page (course_list.html) - Main CRUD Dashboard

This page will display all courses and allow users to:

- ✓ **Create** a new course
 - ✓ **Edit** an existing course
 - ✓ **Delete** a course
 - ✓ **Manage lessons** (add, update, delete lessons for each course)
-

1. Create base.html

- **Location:** templates/courses/base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>{% block title %}EduLearn{% endblock %}</title>

  <!-- Bootstrap Local Integration -->
  {% load static %}
  <link rel="stylesheet" href="{% static 'courses/bootstrap/css/bootstrap.min.css' %}">
  <script src="{% static 'courses/bootstrap/js/bootstrap.bundle.min.js' %}" defer></script>

  <style>
    body {
      background-color: #f8f9fa;
    }
  </style>
</head>
<body>
  <!-- Navbar -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <div class="container">
      <a class="navbar-brand" href="{% url 'course_list' %}">EduLearn</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">

        </div>
    </div>
  </nav>

  <!-- Main Content -->
```

```

<div class="container mt-4">
    {% if messages %}
        {% for message in messages %}
            <div class="alert alert-{{ message.tags }}">{{ message }}</div>
        {% endfor %}
    {% endif %}

    {% block content %}{% endblock %}
</div>

<!-- Footer -->
<footer class="text-center mt-4 p-3 bg-light">
    <p>&copy; 2025 EduLearn. All rights reserved.</p>
</footer>
</body>
</html>

```

course_list.html - The Course Dashboard

- **Location:** templates/courses/course_list.html
- **Functionality:**
 - Displays all courses.
 - Allows adding, updating, and deleting courses.
 - Allows managing lessons for each course.

```

{% extends "courses/base.html" %}
{% block title %}Course List{% endblock %}

{% block content %}
<div class="container mt-4">
    <h2 class="text-center">Course Management</h2>

    <!-- Success Messages -->
    {% if messages %}
        {% for message in messages %}
            <div class="alert alert-success">{{ message }}</div>
        {% endfor %}
    {% endif %}

    <!-- Button to Create a New Course -->
    <a href="{% url 'course_create' %}" class="btn btn-success mb-3">Create New Course</a>

    <table class="table table-striped">

```

```

<thead>
  <tr>
    <th>Title</th>
    <th>Description</th>
    <th>Duration (hrs)</th>
    <th>Actions</th>
    <th>Manage Lessons</th>
  </tr>
</thead>
<tbody>
  {% for course in courses %}
    <tr>
      <td>{{ course.title }}</td>
      <td>{{ course.description|truncatewords:10 }}</td>
      <td>{{ course.duration }}</td>
      <td>
        <a href="{% url 'course_update' course.id %}" class="btn btn-warning btn-sm">Edit</a>
        <form action="{% url 'course_delete' course.id %}" method="post" style="display:inline;">
          {% csrf_token %}
          <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Are you
sure you want to delete this course?')">Delete</button>
        </form>
      </td>
      <td>
        <a href="{% url 'lesson_create' %}" class="btn btn-info btn-sm">Add Lesson</a>
        <a href="{% url 'course_detail' course.id %}" class="btn btn-secondary btn-sm">View
Lessons</a>
      </td>
    </tr>
  {% empty %}
    <tr>
      <td colspan="5" class="text-center">No courses available.</td>
    </tr>
  {% endfor %}
</tbody>
</table>
</div>
{% endblock %}

```

Lesson Management (From Course Details Page)

- **Page:** course_detail.html
- **Functionality:**
 - Lists all lessons in a selected course.

- Allows adding, updating, and deleting lessons.

```
{% extends "courses/base.html" %}
{% block title %}Course Details{% endblock %}

{% block content %}
<div class="container mt-4">
  <h2>{{ course.title }}</h2>
  <p>{{ course.description }}</p>
  <h4>Lessons</h4>

  <a href="{% url 'lesson_create' %}" class="btn btn-success mb-3">Add New Lesson</a>

  <ul class="list-group">
    {% for lesson in lessons %}
      <li class="list-group-item d-flex justify-content-between align-items-center">
        {{ lesson.title }}
        <div>
          <a href="{% url 'lesson_update' lesson.id %}" class="btn btn-warning btn-sm">Edit</a>
          <form action="{% url 'lesson_delete' lesson.id %}" method="post" style="display:inline;">
            {% csrf_token %}
            <button type="submit" class="btn btn-danger btn-sm" onclick="return confirm('Are you
sure you want to delete this lesson?')">Delete</button>
          </form>
        </div>
      </li>
    {% empty %}
      <li class="list-group-item text-center">No lessons available for this course.</li>
    {% endfor %}
  </ul>
</div>
{% endblock %}
```

2: Create Forms for Course and Lesson

- Inside the courses/ app, create a new file forms.py:

```
from django import forms
from .models import Course, Lesson

class CourseForm(forms.ModelForm):
    class Meta:
        model = Course
        fields = ['title', 'description', 'duration', 'thumbnail']
        widgets = {
```

```
        'title': forms.TextInput(attrs={'class': 'form-control'}),
        'description': forms.Textarea(attrs={'class': 'form-control'}),
        'duration': forms.NumberInput(attrs={'class': 'form-control'}),
    }
}
```

```
class LessonForm(forms.ModelForm):
    class Meta:
        model = Lesson
        fields = ['course', 'title', 'content']
        widgets = {
            'course': forms.Select(attrs={'class': 'form-control'}),
            'title': forms.TextInput(attrs={'class': 'form-control'}),
            'content': forms.Textarea(attrs={'class': 'form-control'}),
        }
}
```

Step 3: Implement CRUD Views for Course

- Open views.py inside the courses app and add CRUD views for Course:

```
from django.shortcuts import render, get_object_or_404, redirect
from django.contrib.auth.decorators import login_required
from .models import Course, Lesson
from .forms import CourseForm, LessonForm
from django.contrib import messages

# Course List
def course_list(request):
    courses = Course.objects.all()
    return render(request, 'courses/course_list.html', {'courses': courses})

# Course Detail
def course_detail(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    lessons = course.lessons.all()
    return render(request, 'courses/course_detail.html', {'course': course, 'lessons': lessons})

# Create Course
def course_create(request):
    if request.method == "POST":
        form = CourseForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            messages.success(request, "Course created successfully!")
            return redirect('course_list')
```

```
else:
    form = CourseForm()
return render(request, 'courses/course_form.html', {'form': form})
```

Update Course

```
def course_update(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    if request.method == "POST":
        form = CourseForm(request.POST, request.FILES, instance=course)
        if form.is_valid():
            form.save()
            messages.success(request, "Course updated successfully!")
            return redirect('course_list')
    else:
        form = CourseForm(instance=course)
    return render(request, 'courses/course_form.html', {'form': form})
```

Delete Course

```
def course_delete(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    if request.method == "POST":
        course.delete()
        messages.success(request, "Course deleted successfully!")
        return redirect('course_list')
    return render(request, 'courses/course_confirm_delete.html', {'course': course})
```

Step 4: Implement CRUD Views for Lesson

- Add CRUD views for Lesson inside views.py:

```
# Create Lesson
def lesson_create(request):
    if request.method == "POST":
        form = LessonForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, "Lesson created successfully!")
            return redirect('course_list')
    else:
        form = LessonForm()
    return render(request, 'courses/lesson_form.html', {'form': form})
```

Update Lesson

```
def lesson_update(request, lesson_id):
    lesson = get_object_or_404(Lesson, id=lesson_id)
```

```

if request.method == "POST":
    form = LessonForm(request.POST, instance=lesson)
    if form.is_valid():
        form.save()
        messages.success(request, "Lesson updated successfully!")
        return redirect('course_list')
    else:
        form = LessonForm(instance=lesson)
return render(request, 'courses/lesson_form.html', {'form': form})

```

Delete Lesson

```

def lesson_delete(request, lesson_id):
    lesson = get_object_or_404(Lesson, id=lesson_id)
    if request.method == "POST":
        lesson.delete()
        messages.success(request, "Lesson deleted successfully!")
        return redirect('course_list')
    return render(request, 'courses/lesson_confirm_delete.html', {'lesson': lesson})

```

Step 5: Define URL Patterns

- Open `urls.py` inside `courses/` and add CRUD URLs:

```

from django.urls import path
from .views import *

urlpatterns = [
    path('', course_list, name='course_list'),
    path('<int:course_id>/', course_detail, name='course_detail'),

    path('create/', course_create, name='course_create'),
    path('<int:course_id>/update/', course_update, name='course_update'),
    path('<int:course_id>/delete/', course_delete, name='course_delete'),

    path('lesson/create/', lesson_create, name='lesson_create'),
    path('lesson/<int:lesson_id>/update/', lesson_update, name='lesson_update'),
    path('lesson/<int:lesson_id>/delete/', lesson_delete, name='lesson_delete'),
]

```