

## Create A Full-Stack Web Application For An Online Course Platform.

Use Next.js with App Router and Bootstrap for the frontend UI, and Django REST Framework for the backend API.

The app should support the following features: Frontend (Next.js + Bootstrap): Use Next.js App Router with app/ directory structure Use Bootstrap 5 for responsive design and UI Pages:

Home, All Courses, Add Course, Edit Course, Course Details CRUD operations for Courses (title, description) Show list of Lectures for each Course Add/Edit/Delete Lectures (title, video URL) Use fetch to communicate with Django API at <http://localhost:8000/api/>

Use useState, useEffect, useParams and useRouter for dynamic behavior Structure components in src/components Backend (Django + Django REST Framework):

Django project and app named

### Courses Models:

Course: title (char), description (text)

Lecture: title (char), video\_url (url), course (foreign key)

### Use Django REST Framework to create API endpoints

#### Endpoints:

GET /api/courses/ – list all courses

POST /api/courses/ – create new course

GET /api/courses/<id>/ – retrieve course with lectures

PUT /api/courses/<id>/ – update course

DELETE /api/courses/<id>/ – delete course

GET/POST/PUT/DELETE /api/lectures/ – manage lectures

Enable CORS (all origins) for development Bonus:

Add loading state while fetching Add error handling for failed API requests Add LectureForm for adding lectures within course detail view<sup>\*\*\*</sup>

## Backend In Django :-

# Part 1: Backend – Django + Django REST Framework

## 1.1 Project Setup

```
django-admin startproject backend
cd backend
python manage.py startapp courses
```

## 1.2 Install Required Packages

```
pip install djangorestframework
pip install django-cors-headers
```

## Add to settings.py:-

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'corsheaders',
    'courses',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'corsheaders.middleware.CorsMiddleware',
]

CORS_ALLOWED_ORIGINS = [
    "http://localhost:3000", #  frontend origin
]
```

## 1.3 Models – courses/models.py

```
from django.db import models

class Course(models.Model):
    title = models.CharField(max_length=255)
    description = models.TextField()

    def __str__(self):
        return self.title

class Lecture(models.Model):
    course = models.ForeignKey(Course, related_name='lectures',
on_delete=models.CASCADE)
    title = models.CharField(max_length=255)
    video_url = models.URLField()

    def __str__(self):
        return self.title
```

python manage.py makemigrations

python manage.py migrate

## 1.4 Serializers – courses/serializers.py

```
from rest_framework import serializers
from .models import Course, Lecture

class LectureSerializer(serializers.ModelSerializer):
    class Meta:
        model = Lecture
        fields = '__all__'

class CourseSerializer(serializers.ModelSerializer):
    lectures = LectureSerializer(many=True, read_only=True)

    class Meta:
        model = Course
        fields = ['id', 'title', 'description', 'lectures']
```

## 1.6 URLs – courses/urls.py

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import CourseViewSet, LectureViewSet

router = DefaultRouter()
router.register(r'courses', CourseViewSet)
router.register(r'lectures', LectureViewSet)

urlpatterns = [
    path('', include(router.urls)),
]
```

## In backend/urls.py(Project Folder URL):

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('courses.urls')),
]
```

## 1.5 Views – courses/views.py

```
from rest_framework import viewsets
from .models import Course, Lecture
from .serializers import CourseSerializer, LectureSerializer

class CourseViewSet(viewsets.ModelViewSet):
    queryset = Course.objects.all()
    serializer_class = CourseSerializer

class LectureViewSet(viewsets.ModelViewSet):
    queryset = Lecture.objects.all()
    serializer_class = LectureSerializer
```

```
C:\Users\User>F:
F:\>cd Raj singh
F:\Raj Singh>cd backend
F:\Raj Singh\backend>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

→ 127.0.0.1:8000/api/courses/

Django REST framework

Api Root / Course List

## Course List

GET /api/courses/

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "id": 1,
    "title": "Python programming",
    "description": "Python is a programming Language which is used in to AI and Machine Learning",
    "lectures": [
      {
        "id": 1,
        "title": "Introduction to Python Variables",
        "video_url": "https://youtu.be/s2dP9fbfVTg?si=RpitS0G9snFlysgZ",
        "course": 1
      }
    ]
  }
],
```

## Frontend – Next.js + Bootstrap 5

### 2.1 Create Project

```
npx create-next-app@latest course-app
cd course-app
```

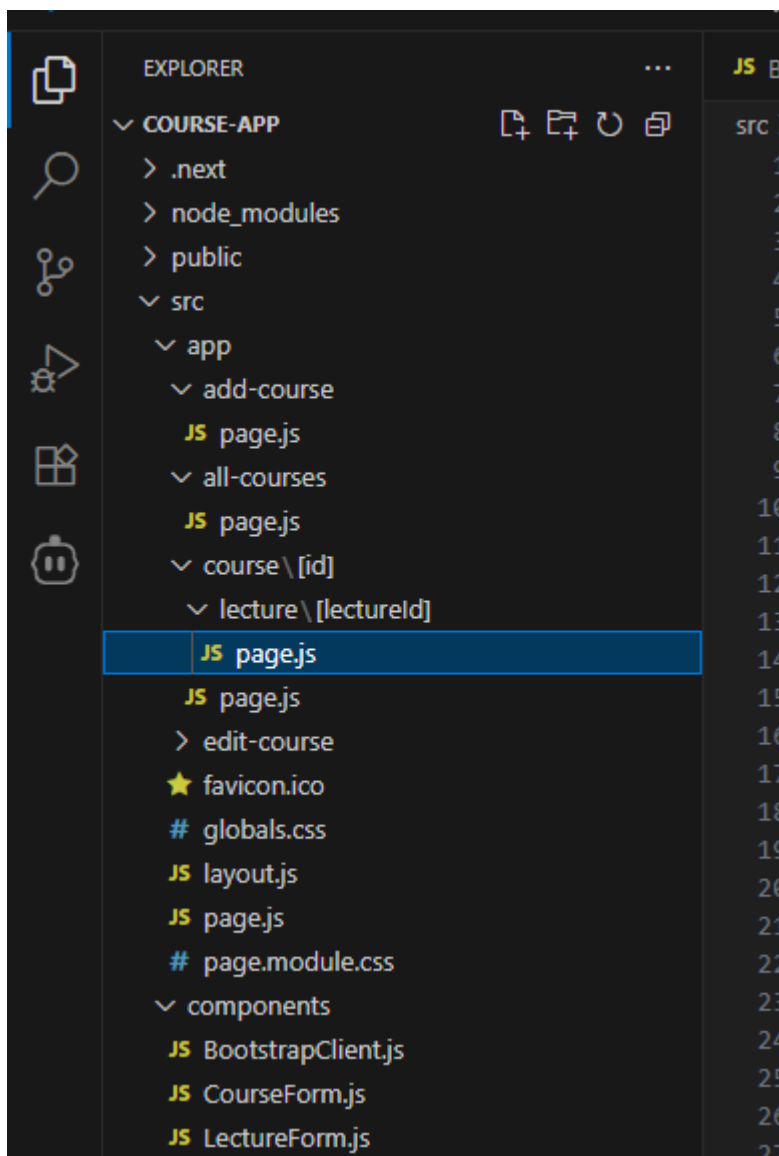
src/

├─ app/

| └─ page.js # Home

| └─ all-courses/page.js

- | └─ add-course/page.js
- | └─ edit-course/[id]/page.js
- | └─ course/[id]/page.js
- | └─ layout.js # With Bootstrap
- | └─ components/
  - | └─ CourseForm.js
  - | └─ LectureForm.js
  - | └─ CourseList.js
  - | └─ LectureList.js



## 2.2 Add Bootstrap

In `layout.js`:

```
import 'bootstrap/dist/css/bootstrap.min.css';
import BootstrapClient from '@components/BootstrapClient';

export const metadata = {
  title: 'Course Platform',
};

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body>
        <BootstrapClient />
        {children}
      </body>
    </html>
  );
}
```

## 2.3 Fetch Example – `all-courses/page.js`

```
'use client';
import { useEffect, useState } from 'react';
import Link from 'next/link';

export default function AllCourses() {
  const [courses, setCourses] = useState([]);
  const [loading, setLoading] = useState(false);

  const fetchCourses = async () => {
    const res = await fetch('http://localhost:8000/api/courses/');
    const data = await res.json();
    setCourses(data);
  };

  useEffect(() => {
    fetchCourses();
  }, []);

  const handleDelete = async (id) => {
    const confirmDelete = window.confirm('Are you sure you want to delete this course?');
    if (!confirmDelete) return;
  };
}
```

```

    setLoading(true);
    const res = await fetch(`http://localhost:8000/api/courses/${id}/`, {
      method: 'DELETE',
    });

    if (res.ok) {
      fetchCourses(); // Refresh list
    } else {
      alert('Failed to delete course');
    }
    setLoading(false);
  });

  return (
    <div className="container mt-4">
      <h2>All Courses</h2>
      <Link href="/add-course" className="btn btn-success mb-3">Add
Course</Link>
      <ul className="list-group">
        {courses.map(course => (
          <li key={course.id} className="list-group-item d-flex justify-
content-between align-items-center">
            <div>
              <Link href={` /course/${course.id}` } className="text-decoration-
none">{course.title}</Link>
            </div>
            <div>
              <Link href={` /edit-course/${course.id}` } className="btn btn-sm
btn-outline-secondary me-2">Edit</Link>
              <button className="btn btn-sm btn-danger" onClick={() =>
handleDelete(course.id)} disabled={loading}>
                {loading ? 'Deleting...' : 'Delete'}
              </button>
            </div>
          </li>
        ))}
      </ul>
    </div>
  );
}

```

## Create `CourseForm.js` component

`src/components/CourseForm.js`

```

'use client';
import { useState } from 'react';

export default function CourseForm({ initialData = {}, onSubmit }) {
  const [title, setTitle] = useState(initialData.title || '');
  const [description, setDescription] = useState(initialData.description ||
  '');
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState('');

  const handleSubmit = async (e) => {
    e.preventDefault();
    setLoading(true);
    setError('');
    try {
      await onSubmit({ title, description });
    } catch {
      setError('Error saving course');
    } finally {
      setLoading(false);
    }
  };

  return (
    <form onSubmit={handleSubmit}>
      {error} && <div className="alert alert-danger">{error}</div>
      <div className="mb-3">
        <label className="form-label">Title</label>
        <input className="form-control" value={title} onChange={e =>
setTitle(e.target.value)} required />
      </div>
      <div className="mb-3">
        <label className="form-label">Description</label>
        <textarea className="form-control" rows="4" value={description}
onChange={e => setDescription(e.target.value)} required />
      </div>
      <button className="btn btn-primary" disabled={loading}>
        {loading ? 'Saving...' : 'Save Course'}
      </button>
    </form>
  );
}

```

#### Add Course Page – /add-course/page.js

```

'use client';
import CourseForm from '@components/CourseForm';

```

```

import { useRouter } from 'next/navigation';

export default function AddCoursePage() {
  const router = useRouter();

  const handleAdd = async (data) => {
    const res = await fetch('http://localhost:8000/api/courses/', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(data),
    });

    if (!res.ok) throw new Error('Failed to add course');
    router.push('/all-courses');
  };

  return (
    <div className="container mt-4">
      <h2>Add New Course</h2>
      <CourseForm onSubmit={handleAdd} />
    </div>
  );
}

```

src/app/edit-course/[id]/page.js

```

'use client';
import { useEffect, useState } from 'react';
import { useRouter, useParams } from 'next/navigation';
import CourseForm from '@components/CourseForm';

export default function EditCoursePage() {
  const { id } = useParams();
  const router = useRouter();
  const [course, setCourse] = useState(null);

  useEffect(() => {
    fetch(`http://localhost:8000/api/courses/${id}/`)
      .then(res => res.json())
      .then(data => setCourse(data));
  }, [id]);

  const handleUpdate = async (data) => {
    const res = await fetch(`http://localhost:8000/api/courses/${id}/`, {
      method: 'PUT',
      headers: { 'Content-Type': 'application/json' },

```

```

    body: JSON.stringify(data),
  });

  if (!res.ok) throw new Error('Failed to update course');
  router.push('/all-courses');
};

return (
  <div className="container mt-4">
    <h2>Edit Course</h2>
    {course ? <CourseForm initialData={course} onSubmit={handleUpdate} /> :
  <p>Loading...</p>}
  </div>
);
}

```



src/components/LectureForm.js

```

'use client';
import { useState, useEffect } from 'react';

export default function LectureForm({ courseId, onSubmit, lectureToEdit,
onCancel }) {
  const [title, setTitle] = useState('');
  const [videoUrl, setVideoUrl] = useState('');

  useEffect(() => {
    if (lectureToEdit) {
      setTitle(lectureToEdit.title);
      setVideoUrl(lectureToEdit.video_url);
    } else {
      setTitle('');
      setVideoUrl('');
    }
  }, [lectureToEdit]);

  const handleSubmit = (e) => {
    e.preventDefault();
    const data = {
      title,
      video_url: videoUrl,
      course: courseId,
    };
    onSubmit(data, lectureToEdit?.id);
    setTitle('');
    setVideoUrl('');
  };

```

```

};

return (
  <form onSubmit={handleSubmit} className="mt-4">
    <h5>{lectureToEdit ? 'Edit Lecture' : 'Add Lecture'}</h5>
    <div className="mb-2">
      <input
        className="form-control"
        placeholder="Lecture Title"
        value={title}
        onChange={(e) => setTitle(e.target.value)}
        required
      />
    </div>
    <div className="mb-2">
      <input
        className="form-control"
        placeholder="Video URL"
        value={videoUrl}
        onChange={(e) => setVideoUrl(e.target.value)}
        required
      />
    </div>
    <button className="btn btn-primary me-2" type="submit">
      {lectureToEdit ? 'Update' : 'Add'}
    </button>
    {lectureToEdit && (
      <button className="btn btn-secondary" type="button"
onClick={onCancel}>
      Cancel
    </button>
    )}
  </form>
);
}

```

✓ src/app/course/[id]/page.js

```

'use client';
import { useEffect, useState } from 'react';
import { useParams } from 'next/navigation';
import LectureForm from '@components/LectureForm';

```

```

export default function CourseDetailPage() {
  const { id } = useParams();
  const [course, setCourse] = useState(null);
  const [lectureToEdit, setLectureToEdit] = useState(null);

  const fetchCourse = () => {
    fetch(`http://localhost:8000/api/courses/${id}/`)
      .then(res => res.json())
      .then(data => setCourse(data));
  };

  useEffect(() => {
    fetchCourse();
  }, [id]);

  const handleDeleteLecture = async (lectureId) => {
    await fetch(`http://localhost:8000/api/lectures/${lectureId}/`, {
      method: 'DELETE',
    });
    fetchCourse(); // refresh lectures
  };

  const handleSubmitLecture = async (data, lectureId) => {
    const url = lectureId
      ? `http://localhost:8000/api/lectures/${lectureId}/`
      : `http://localhost:8000/api/lectures/`;

    const method = lectureId ? 'PUT' : 'POST';

    await fetch(url, {
      method,
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(data),
    });

    setLectureToEdit(null);
    fetchCourse();
  };

  if (!course) return <div className="container mt-4">Loading...</div>;

  return (
    <div className="container mt-4">
      <h2>{course.title}</h2>
      <p>{course.description}</p>

      <LectureForm
        courseId={course.id}

```

```

    onSubmit={handleSubmitLecture}
    lectureToEdit={lectureToEdit}
    onCancel={() => setLectureToEdit(null)}
  />

  <h4 className="mt-4">Lectures</h4>
  <ul className="list-group">
    {course.lectures.map((lec) => (
      <li key={lec.id} className="list-group-item d-flex justify-content-
between align-items-center">
        <div>
          {lec.title} - <a href={lec.video_url} target="_blank"
rel="noopener noreferrer">Video</a>
        </div>
        <div>
          <button
            className="btn btn-sm btn-outline-secondary me-2"
            onClick={() => setLectureToEdit(lec)}
          >
            Edit
          </button>
          <button className="btn btn-sm btn-danger" onClick={() =>
handleDeleteLecture(lec.id)}>
            Delete
          </button>
        </div>
      </li>
    ))}
  </ul>

  <ul className="list-group">
    {course.lectures.map((lec) => (
      <li key={lec.id} className="list-group-item">
        <div className="d-flex justify-content-between align-items-center">
          <div>
            <strong>{lec.title}</strong>
          </div>
          <div>
            <button
              className="btn btn-sm btn-outline-secondary me-2"
              onClick={() => setLectureToEdit(lec)}
            >
              Edit
            </button>
            <button
              className="btn btn-sm btn-danger"
              onClick={() => handleDeleteLecture(lec.id)}
            >
              Delete
            </button>
          </div>
        </div>
      </li>
    ))}
  </ul>

```

```

        Delete
    </button>

    <div className="mt-2">
    <a
    href={` /course/${course.id}/lecture/${lec.id}`}
    className="btn btn-sm btn-primary"
    >
    Watch Video
    </a>
</div>

    </div>
</div>

    { /* 📌 Embedded video section */}
    <div className="mt-2">
        {lec.video_url.includes('youtube.com') ||
lec.video_url.includes('youtu.be')} ? (
        // 📌 YouTube embed
        <iframe
        width="100%"
        height="315"
        src={lec.video_url.replace('watch?v=', 'embed/')}
        title={lec.title}
        frameBorder="0"
        allow="accelerometer; autoplay; clipboard-write; encrypted-media;
gyroscope; picture-in-picture"
        allowFullScreen
        >></iframe>
        ) : (
        // 📌 MP4 or direct video
        <video width="100%" height="315" controls>
            <source src={lec.video_url} type="video/mp4" />
            Your browser does not support the video tag.
        </video>
        )}
    </div>
</li>
    )}
</ul>

    </div>
);
}

```



src/app/course/[id]/lecture/[lectureid]page.js:-

```
'use client';
import { useParams } from 'next/navigation';
import { useEffect, useState } from 'react';

export default function LectureVideoPage() {
  const { lectureId } = useParams();
  const [lecture, setLecture] = useState(null);

  useEffect(() => {
    fetch(`http://localhost:8000/api/lectures/${lectureId}/`)
      .then(res => res.json())
      .then(data => setLecture(data));
  }, [lectureId]);

  if (!lecture) return <div className="container mt-4">Loading...</div>;

  return (
    <div className="container mt-4">
      <h2>{lecture.title}</h2>
      <div className="mt-3">
        {lecture.video_url.includes('youtube.com')} ||
        lecture.video_url.includes('youtu.be') ? (
          <div className="ratio ratio-16x9">
            <iframe
              src={lecture.video_url.replace('watch?v=', 'embed/')}
              title={lecture.title}
              allowFullScreen
              width="100%"
              height="400"
            >></iframe>
          </div>
        ) : (
          <video width="100%" height="400" controls>
            <source src={lecture.video_url} type="video/mp4" />
            Your browser does not support the video tag.
          </video>
        )
      </div>
    </div>
  );
}
```

src/app/page.js:-

```
'use client';
import { useEffect } from 'react';
import { useRouter } from 'next/navigation';

export default function HomePage() {
  const router = useRouter();
  useEffect(() => {
    router.push('/all-courses');
  }, [router]);
  return null;
}
```

**src/components/BootstrapClient.js**

```
'use client';
import { useEffect } from 'react';

export default function BootstrapClient() {
  useEffect(() => {
    import('bootstrap/dist/js/bootstrap.bundle.min.js');
  }, []);

  return null;
}
```

**Output: -**

```
F:\Raj Singh\course-app>npm run dev
```

```
> course-app@0.1.0 dev
```

```
> next dev --turbo
```

```
▲ Next.js 15.3.4 (Turbopack)
```

```
- Local:      http://localhost:3000
```

```
- Network:    http://192.168.1.7:3000
```

```
👉 Starting...
```

```
Slow filesystem detected. If F:\Raj Singh\course-app\.next is a network drive or  
you have an antivirus enabled, consider excluding your project directory.
```

```
👉 Ready in 2.2s
```

```
o Compiling / ...
```

```
👉 Compiled / in 2.5s
```

```
GET / 200 in 3038ms
```

```
o Compiling /all-courses ...
```

```
GET /favicon.ico?favicon.45db1c09.ico 200 in 1487ms
```

```
👉 Compiled /all-courses in 2s
```

```
GET /all-courses 200 in 2110ms
```