StudyNotion - EdTech Platform

**🔗 Live Demo:** **<https://studynotion-1-if95.onrender.com/>**  
**🔗 GitHub Repository:** <https://github.com/Naitik4897/StudyNotion>

StudyNotion is a fully functional ed-tech platform that enables users to create, consume, and rate educational content. The platform is built using the MERN stack (MongoDB, ExpressJS, ReactJS, NodeJS) and provides a seamless and interactive learning experience for students and a platform for instructors to sh owcase their expertise.

# 🚀 Features

## For Students

* User Registration & Authentication: Secure signup/login with email verification
* Course Catalog: Browse and search through available courses
* Course Enrollment: Purchase and enroll in courses using Razorpay integration
* Video Streaming: Watch course videos with progress tracking
* Course Progress: Track learning progress and completion status
* User Profile: Manage personal information and view enrolled courses
* Rating & Reviews: Rate and review courses after completion
* Wishlist: Save courses for later purchase

## For Instructors

* Instructor Dashboard: Comprehensive dashboard to manage courses
* Course Creation: Create courses with videos, descriptions, and pricing
* Content Management: Upload and organize course materials
* Student Analytics: View student enrollment and progress statistics
* Earnings Overview: Track revenue and course performance
* Course Editing: Edit and update existing courses

## System Features

* Responsive Design: Works seamlessly on desktop and mobile devices
* Cloud Storage: Media files stored securely on Cloudinary
* Payment Integration: Secure payment processing with Razorpay
* Email Notifications: Automated email notifications for various actions
* Search & Filter: Advanced search and filtering capabilities
* Media Management: Efficient handling of images and videos

# 🛠️ Tech Stack

## Frontend

* React.js
* Redux Toolkit
* Tailwind CSS
* React Router
* Axios
* React Hook Form
* React Player
* React Hot Toast
* React Icons

## Backend

* Node.js
* Express.js
* MongoDB
* Mongoose
* JWT
* Bcrypt
* Multer
* Cloudinary
* Razorpay
* Nodemailer
* Express Fileupload

# 📁 Project Structure

A well-organized folder structure separating frontend, backend, static assets, and documentation for better maintainability.

# 🚀 Getting Started

## Prerequisites

* Node.js (v14 or higher)
* MongoDB (local or cloud instance)
* Cloudinary account
* Razorpay account

## Installation

Clone the repository and install dependencies for both frontend and backend.

## Environment Configuration

Create .env files in the root and server directories with respective environment variables.

## Start the application

* Run both frontend and backend concurrently using `npm run dev`
* Or run separately using `npm start` for frontend and `npm run dev` for backend

## Access the application

* Frontend: http://localhost:3000
* Backend: http://localhost:4000

# 🔧 API Endpoints

## Authentication

* POST /api/v1/auth/signup - User registration
* POST /api/v1/auth/login - User login
* POST /api/v1/auth/sendotp - Send OTP for verification
* POST /api/v1/auth/changepassword - Change password

## Courses

* GET /api/v1/course/getAllCourses - Get all courses
* POST /api/v1/course/createCourse - Create new course
* GET /api/v1/course/getCourseDetails/:id - Get course details
* PUT /api/v1/course/editCourse - Edit course
* DELETE /api/v1/course/deleteCourse - Delete course

## Payment

* POST /api/v1/payment/capturePayment - Capture payment
* POST /api/v1/payment/verifyPayment - Verify payment
* POST /api/v1/payment/sendPaymentSuccessEmail - Send payment confirmation

## Profile

* GET /api/v1/profile/getUserDetails - Get user profile
* PUT /api/v1/profile/updateProfile - Update profile
* PUT /api/v1/profile/updateDisplayPicture - Update profile picture
* DELETE /api/v1/profile/deleteProfile - Delete account

# 🌟 Key Features Implementation

## Authentication System

* JWT-based authentication with refresh tokens
* Email verification using OTP
* Password reset functionality
* Role-based access control (Student/Instructor)

## Payment Integration

* Razorpay payment gateway integration
* Secure payment processing
* Payment verification and confirmation
* Automated email receipts

## Course Management

* Rich course creation interface
* Video upload and streaming
* Course progress tracking
* Rating and review system

## User Experience

* Responsive design for all devices
* Fast loading with optimized images
* Interactive UI with smooth animations
* Real-time notifications

# 🤝 Contributing

Fork the repository, create a feature branch, commit changes, push to GitHub, and open a Pull Request.

# 👥 Authors

Naitik - Initial work - Naitik4897

# 🙏 Acknowledgments

* React.js community for excellent documentation
* MongoDB for robust database solutions
* Cloudinary for seamless media management
* Razorpay for secure payment processing
* All contributors who helped improve this project

# 📞 Support

For support, email nrmaisuriya3@gmail.com or create an issue in this repository.

# 🔮 Future Enhancements

* Mobile app development
* Advanced analytics dashboard
* AI-powered course recommendations
* Live streaming capabilities
* Multi-language support
* Offline course downloads
* Integration with learning management systems
* Advanced search with filters
* Discussion forums for courses
* Certificate generation system