# SmartInternz: Role-Based Learning Platform

Full Stack MERN Learning Platform

Introduction

**Project Title**: Role Based-Learning Platform

**Team Members:**

* **Yalamati Jeswanth Kumar Raju:** Frontend UI design with React, routing, state management, API integration, role-based dashboards. Backend API development, JWT authentication, role-based routing, database schema design,

📄 Project Overview

* SmartInternz is a full-stack learning management system designed to simulate a real-world internship platform. It features three user roles—Students, Teachers, and Admins—each with role-specific capabilities like course creation, enrollment, and management.

## 🔧 Technologies Used

* **Frontend**: React.js, React Router, Bootstrap
* **Backend**: Node.js, Express.js
* **Database**: MongoDB
* **Authentication**: JWT (JSON Web Tokens)
* **Tools**: Axios, dotenv, Mongoose

## 📅 Key Features

### Students

* Register/Login
* View available courses
* Enroll in courses
* View enrolled courses

### Teachers

* Register/Login
* Create new courses
* View own created courses

### Admins

* Register/Login
* View all users
* Delete users or courses

## 📁 Folder Structure

SmartInternz/  
├── client/ # React frontend  
│ ├── src/  
│ │ ├── pages/ # All pages per role (Login, Register, Dashboards, etc.)  
│ │ ├── App.js # Routing & Role protection  
│ │ └── api.js # Axios instance with token interceptor  
│  
├── server/ # Express backend  
│ ├── controllers/  
│ ├── middleware/  
│ ├── models/  
│ ├── routes/  
│ └── server.js

## 📁 Environment Setup

### Backend

**Create .env in /server**

PORT=5000  
MONGO\_URI=your\_mongodb\_uri  
JWT\_SECRET=your\_jwt\_secret

**Run backend:**

cd server  
npm install  
npm start

### **Frontend**

cd client  
npm install  
npm start

## 🔘 API Routes Summary

### Users

* POST /api/users/register → Register a user
* POST /api/users/login → Login and receive JWT

### Courses

* GET /api/courses → View all courses
* POST /api/courses/create → Teacher creates a course
* POST /api/courses/enroll → Student enrolls in a course

### Admin

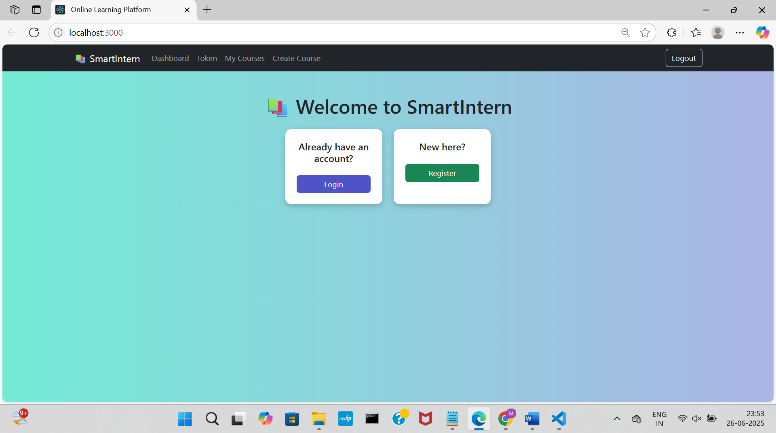
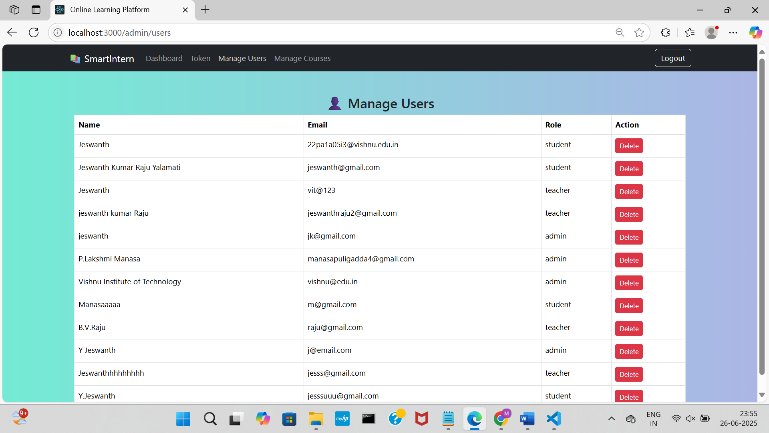
* GET /api/admin/users → View all users
* DELETE /api/admin/user/:id → Delete user
* DELETE /api/admin/course/:id → Delete course

## 🎯 Authentication Flow

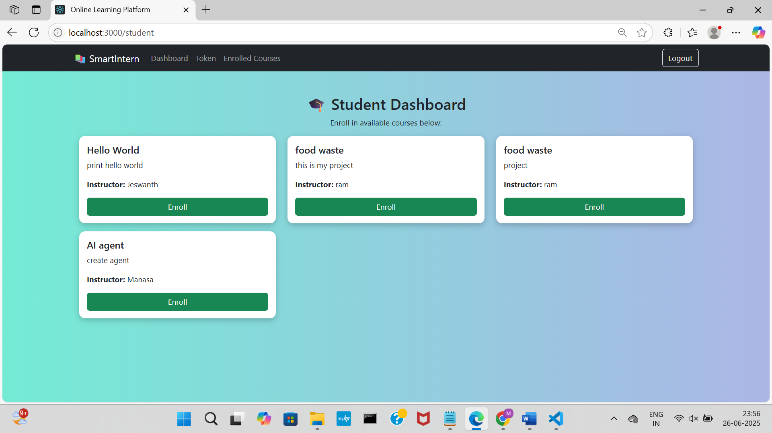
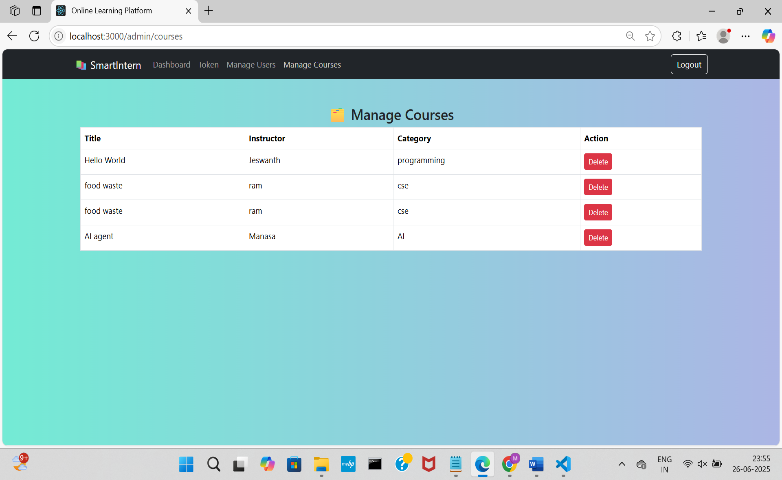
* On login/register, user receives a JWT and profile.
* Stored in localStorage for session persistence.
* Axios attaches token to each request.
* Protected routes are checked in React before rendering.

## ✨ Future Enhancements

* Student course completion tracking
* Teacher dashboards with stats
* Admin analytics (charts, user stats)
* Email verification and notifications

Login and Admin Dashboard



Teacher and Student Dashboard