# Full Stack Development with MERN Project Documentation

# 1. Introduction

Project Title: Learn Hub : Your Center for Skill Enhancement

Team Members:

• Nandyala Pranathi – Full Stack Developer (Frontend, Backend, Database Design, Testing)

# 2. Project Overview

## Purpose:

Learn Hub is a MERN stack-based online learning platform designed to provide affordable and structured digital education. The project aims to create a scalable web application where users can register, enroll in courses, track their progress, and download certificates upon completion.

## Features:

• User Registration and Login (JWT Authentication)

• Course Creation and Management

• Course Enrollment System

• Video-Based Learning Module

• Progress Tracking

• Certificate Generation using PDFKit

• Responsive User Interface

# 3. Architecture

## Frontend:

The frontend is developed using React.js. It follows a component-based architecture where UI components are reusable and state management is handled using React hooks. Routing is managed using React Router.

## Backend:

The backend is built using Node.js and Express.js. RESTful APIs handle authentication, course management, enrollment, and certificate generation. Middleware is used for authentication and error handling.

## Database:

MongoDB is used as the NoSQL database. Collections include Users, Courses, and Enrollments. Mongoose is used for schema modeling and data validation.

# 4. Setup Instructions

## Prerequisites:

• Node.js (v16 or above)

• MongoDB

• Git

## Installation:

1. Clone the repository: git clone <repository-url>

2. Navigate to project directory.

3. Install backend dependencies: cd server && npm install

4. Install frontend dependencies: cd client && npm install

5. Configure environment variables in .env file.

6. Start backend and frontend servers.

# 5. Folder Structure

## Client:

client/  
 ├── src/  
 │ ├── components/  
 │ ├── pages/  
 │ ├── App.js  
 │ └── index.js

## Server:

server/  
 ├── controllers/  
 ├── models/  
 ├── routes/  
 ├── middleware/  
 └── server.js

# 6. Running the Application

Frontend: npm start (inside client directory)

Backend: npm start (inside server directory)

# 7. API Documentation

POST /api/auth/register – Register new user

POST /api/auth/login – User login

GET /api/courses – Get all courses

POST /api/enroll/:id – Enroll in course

GET /api/certificate/:id – Download certificate

# 8. Authentication

Authentication is handled using JWT (JSON Web Token). After successful login, a token is generated and stored on the client side. Protected routes verify the token before granting access.

# 9. User Interface

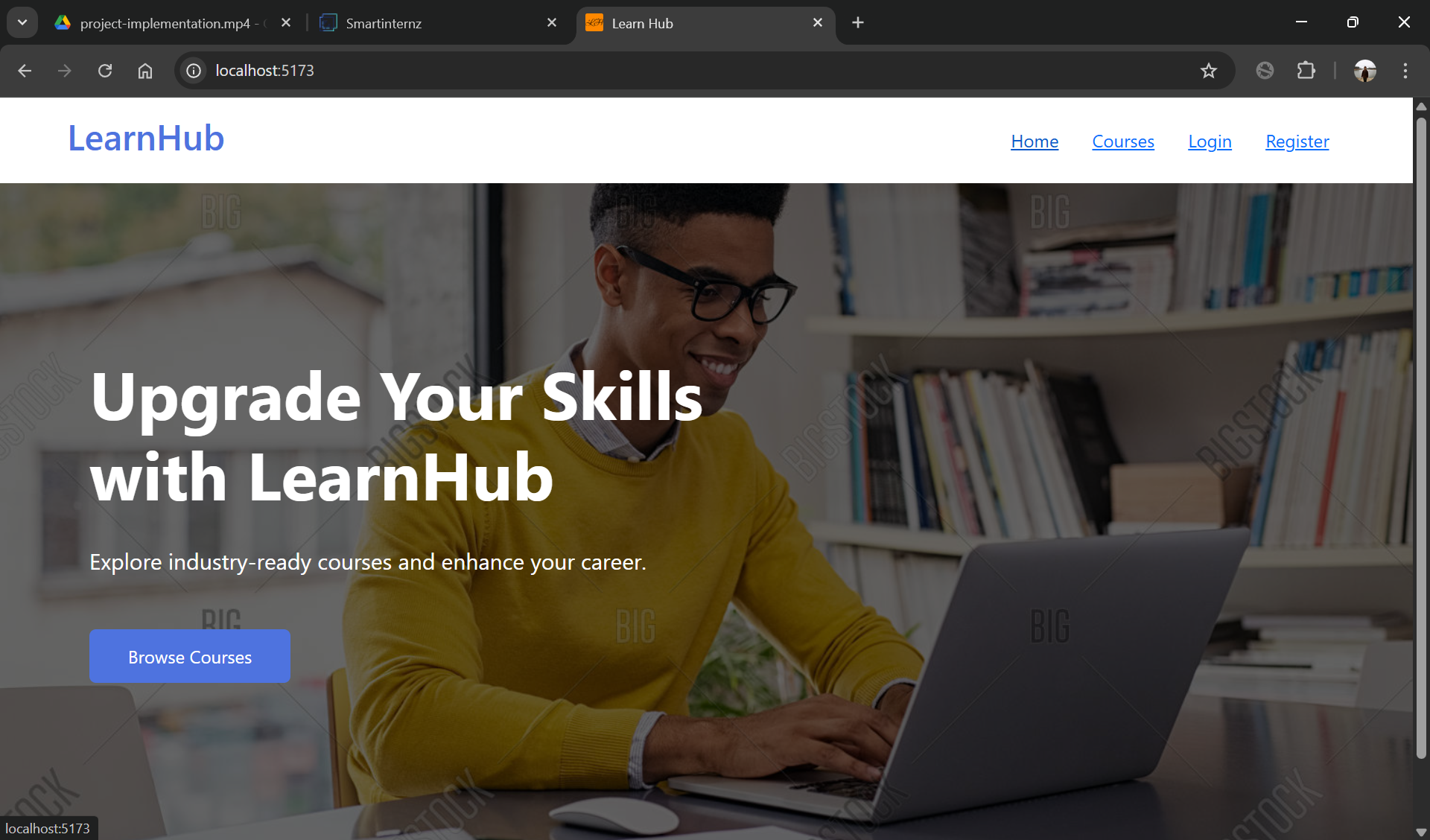


Fig 1: Home Page

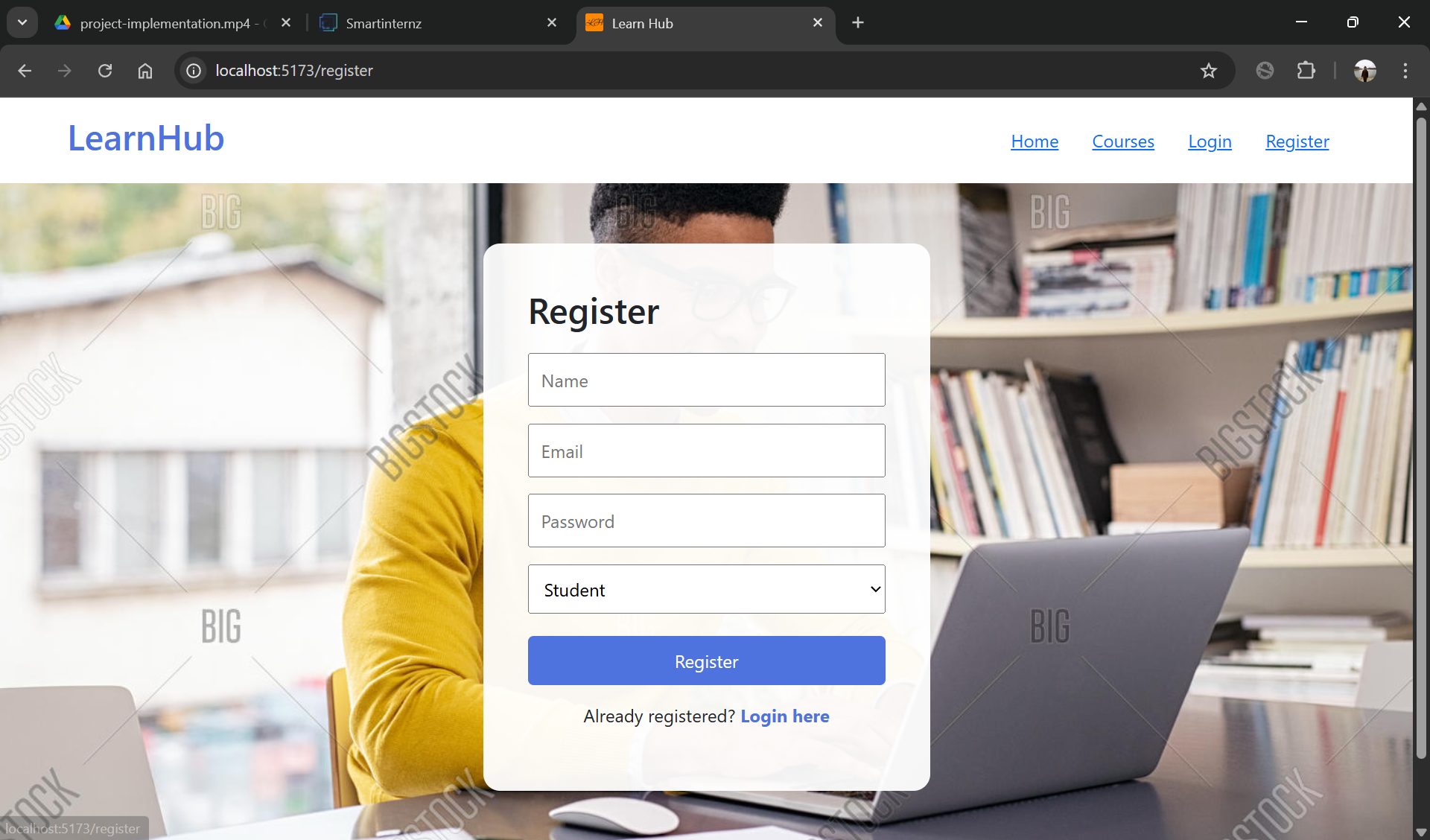


Fig 2: Registration page

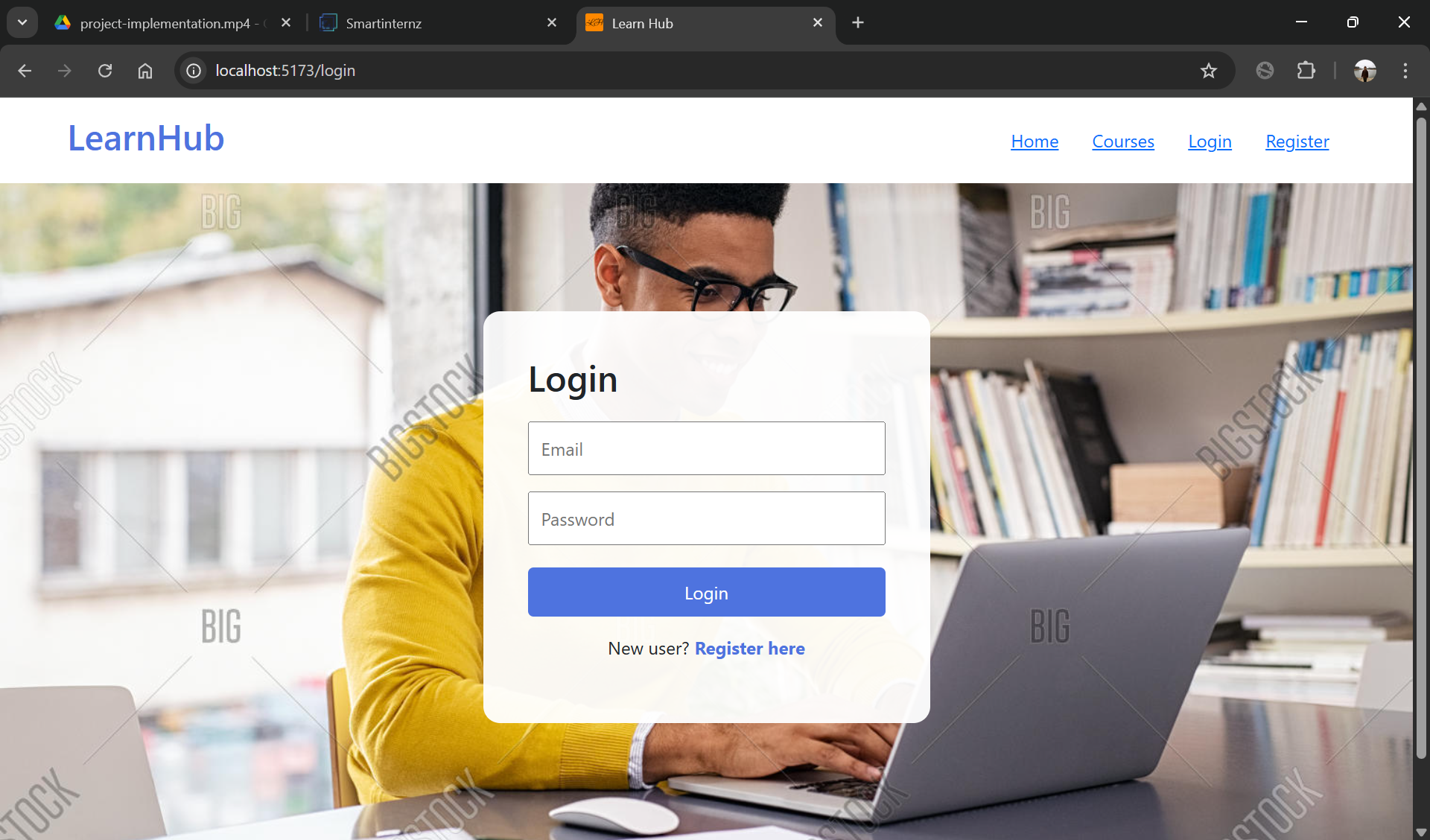


Fig 3: Login page

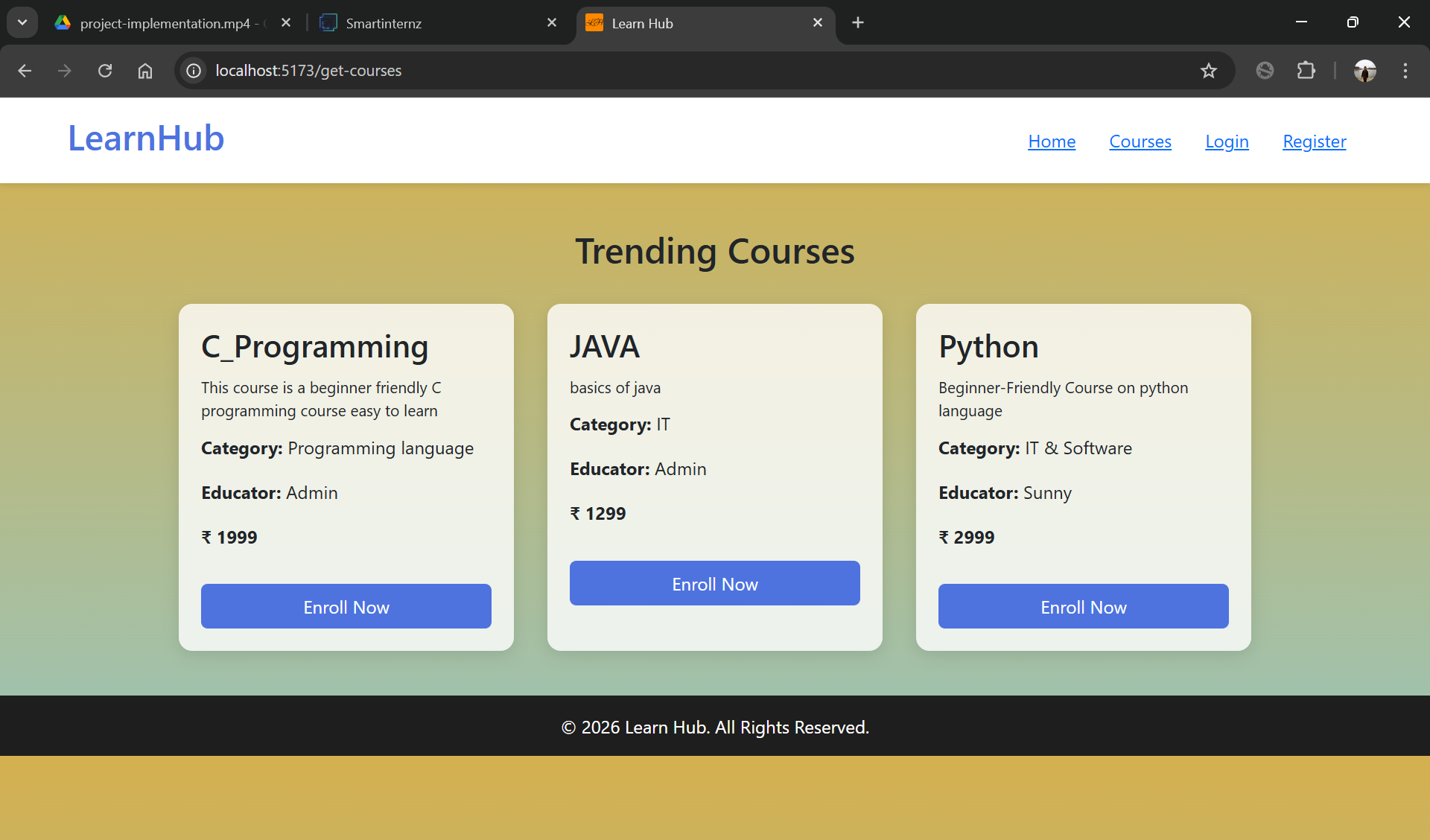


Fig 4: Course Page

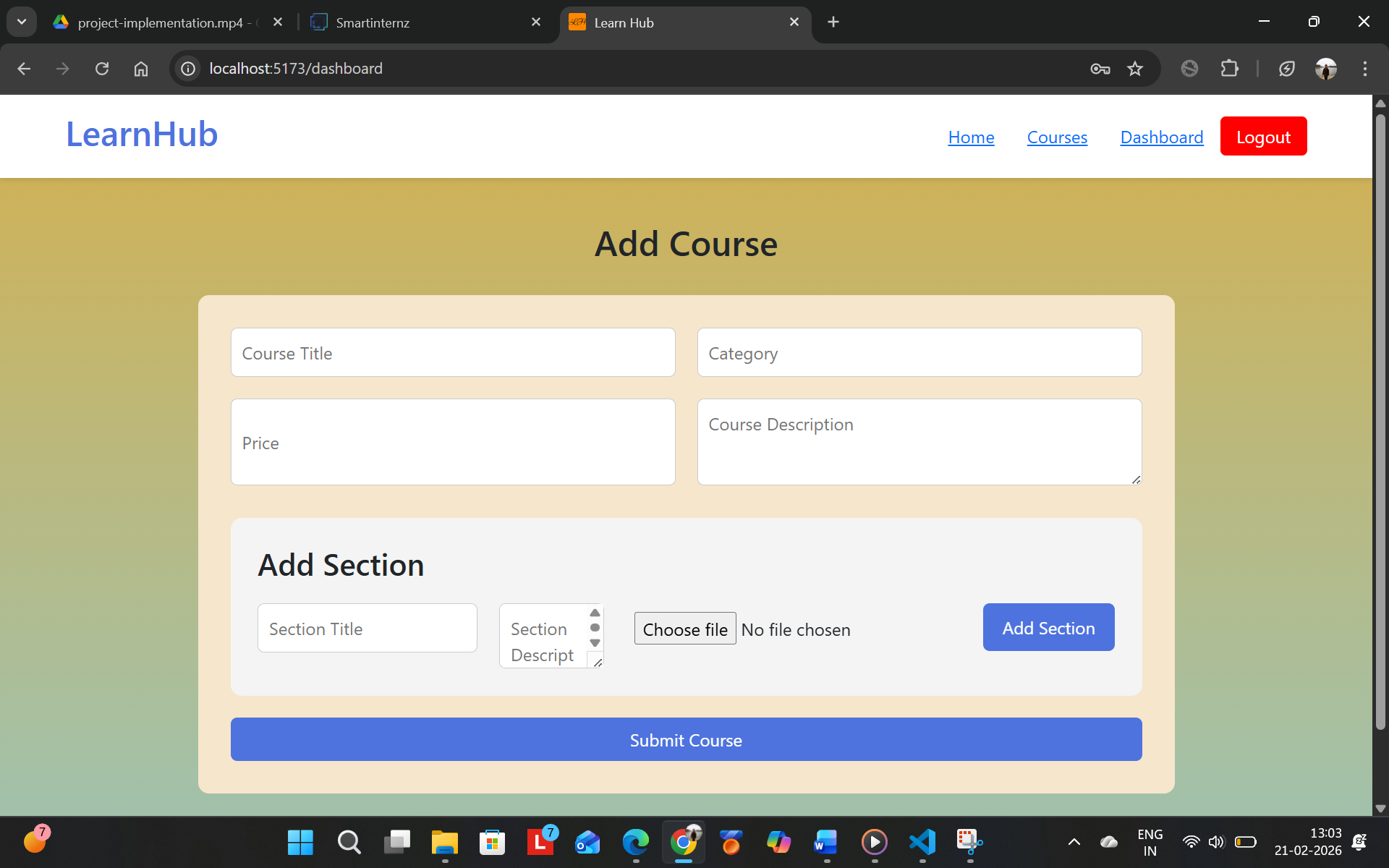


Fig 5: Add Course Page

# 10. Testing

Manual testing was conducted for registration, login, enrollment, progress tracking, and certificate generation. All test cases were validated successfully.

# 11. Screenshots or Demo

GitHub Repository: [Pranathi-collab/Learn-Hub](https://github.com/Pranathi-collab/Learn-Hub)

Project Demo Video: https://drive.google.com/file/d/1yTByqq4RpRp6UUXlGWEZMctMLB7fDe02/view?usp=sharing

# 12. Known Issues

• Application requires stable internet connection.

• No live interaction or chat functionality implemented.

# 13. Future Enhancements

• Mobile application support.

• AI-based course recommendations.

• Live classes and discussion forums.

• Cloud deployment and scalability improvements.