LearnHub – Your Center for Skill Enhancement

(MERN)

Project Details

Project Title:
LearnHub – Your Center for Skill Enhancement

Team Member

Name: Perabattula Ramesh
SmartInternz ID: SBAP0047438

Academic Information

College: Bonam Venkata Chalamayya Engineering College, Odalarevu
Branch: Electronics and Communication Engineering
Semester: 7th
Hall Ticket Number: 22221A04D0

Project Overview:

LearnHub – Your Center for Skill Enhancement

LearnHub is a comprehensive full-stack **Online Learning Platform (OLP)** built to make quality education more **accessible**, **engaging**, and **adaptable** for learners across all backgrounds. Whether you're a student seeking new skills, a teacher aiming to share knowledge, or an administrator managing learning infrastructure, LearnHub provides a unified and intuitive digital environment to support all aspects of modern education.

The platform empowers students to browse and enroll in a diverse range of courses, learn at their own pace, track their progress, and earn industry-recognized certificates. It also enables educators to effortlessly create and manage course content.

Technically, LearnHub is developed using a robust **client-server architecture**. The **frontend** is built with **React.js**, offering a dynamic and responsive user interface enhanced with libraries like **Material UI** and **Bootstrap**. The **backend** leverages **Node.js** with the **Express.js** framework, while **MongoDB** serves as the primary NoSQL database, ensuring scalability, performance, and seamless data management.

The platform supports **three distinct user roles**:

- **Students:** Enroll in and complete courses, join live sessions, and receive certifications.
- **Teachers:** Create, update, and manage course content and monitor student engagement.
- **Admins:** Oversee the platform’s operations, handle user management, and ensure policy enforcement.

✅ **Key Features:**

- Secure authentication and role-based access control
- Course creation, enrollment, and real-time progress tracking
- Interactive tools like discussion forums, comments, and live webinars
- Digital certification upon course completion
- Mobile-friendly and desktop-responsive design
- Payment integration for premium course access
- Scalable backend for handling high traffic and data volumes

LearnHub delivers a **scalable**, **modular**, and **feature-rich** solution tailored to the evolving demands of online education. It not only supports academic institutions and private educators but also aligns with the growing trend of self-paced and remote learning in today’s digital era.

Architecture:

Frontend Architecture – LearnHub

Technology Stack:

The frontend of LearnHub is developed using a modern and modular tech stack to ensure performance, scalability, and a smooth user experience:

- React.js 18 – For building dynamic and responsive user interfaces
- Vite – Fast build tool and development server
- React Router v6 – For seamless client-side routing
- Tailwind CSS – Utility-first CSS framework for efficient styling
- Lucide React – Lightweight icon library for consistent UI elements
- Context API – For managing global application state (authentication, courses, etc.)

Architecture Pattern:
The frontend follows a component-based and context-driven architecture. This promotes clean separation of concerns, reusability, and ease of testing.

```
Component Hierarchy
App.jsx
├── AuthProvider (Context API)
└── Router
    └── AppContent
        ├── Navbar
        └── Routes
            └── PrivateRoute
                └── Page Components
```

- App.jsx is the entry point of the application.
- AuthProvider wraps the application and provides authentication and user data via React Context.
- Router handles the application’s navigation structure.
- PrivateRoute restricts access to authenticated pages based on roles.

State Management – AuthContext

The AuthContext.jsx file manages global application state and provides all necessary methods to interact with backend APIs. It covers user authentication, course handling, and enrollment logic.

```
AuthContext.jsx
├── User State
|   ├── user: User | null
|   ├── loading: boolean
|   ├── courses: Course[]
|   ├── enrollments: Enrollment[]
|   └── certificates: Certificate[]
|
├── Authentication Methods
|   ├── login(email, password)
|   ├── register(userData)
|   ├── logout()
|   └── updateProfile(userData)
|
├── Course Methods
|   ├── getCourses()
|   ├── createCourse(courseData)
|   ├── updateCourse(courseId, courseData)
|   ├── deleteCourse(courseId)
|   └── getTeacherCourses()
|
└── Enrollment Methods
    ├── enrollInCourse(courseId)
    ├── getUserEnrollments()
    ├── markCourseCompleted(courseId)
    └── isEnrolledInCourse(courseId)
```

Frontend Highlights

- Single Source of Truth: All user/course/enrollment states are centralized in AuthContext
- Reusable Components: Navbar, Layouts, Forms, and Cards follow modular structure
- Secure Routing: Authenticated and role-based access using PrivateRoute
- Responsive Design: Tailwind CSS ensures mobile and desktop compatibility
- Seamless Integration: Axios used for RESTful API calls with backend services

This architecture supports a maintainable, scalable, and interactive user experience across different devices and user roles.

Backend Architecture – LearnHub
Technology Stack:
The backend of LearnHub is built using modern, scalable technologies to ensure fast, secure, and maintainable development:

Node.js – Runtime environment for executing JavaScript server-side
Express.js – Minimal and flexible web application framework
MongoDB – NoSQL database for scalable data storage
Mongoose – ODM (Object Data Modeling) library for MongoDB
JWT (JSON Web Token) – Stateless user authentication
bcryptjs – Password hashing for secure credential storage
multer – Middleware for handling file uploads (images, videos, documents)
cors – Enables cross-origin requests for frontend-backend communication

Architecture Pattern

The backend follows a layered architecture pattern with clearly separated concerns for middleware, routing, database operations, and error handling. This improves modularity, scalability, and maintainability.

📁 Server Structure

```
server.js
├── Middleware Layer
│   ├── rateLimit()       → Request throttling
│   ├── cors()            → CORS configuration
│   ├── express.json()    → JSON body parser
│   └── express.urlencoded() → Form data parser
│
├── Database Connection
│   ├── Mongoose connection to MongoDB
│   └── Error handling and auto-reconnection logic
│
├── Route Registration
│   ├── /api/auth          → Authentication (login/register)
│   ├── /api/users         → User management (roles, profiles)
│   ├── /api/courses       → Course creation, editing, listing
│   ├── /api/enrollments   → Enrollment logic and tracking
│   ├── /api/certificates  → Certification generation and records
│   ├── /api/dashboard     → User and admin dashboard data
│   ├── /api/upload        → File uploads (using Multer & Cloudinary)
│   ├── /api/search        → Course/content search features
│
└── Error Handling
    ├── Global error handler middleware
    └── 404 Not Found route handler
```

This backend architecture ensures:

- Secure and stateless authentication using JWT
- Modular API design for easy extension and maintenance
- Robust error handling for improved reliability
- Scalable database interactions through Mongoose models
- Efficient media/file management via Multer

Together with the frontend, this server-side structure powers LearnHub's dynamic e-learning ecosystem and ensures a seamless experience for all users.

Let me know if you want to include database schema diagrams or API endpoint documentation next.

LearnHub – Project Setup Instructions

LearnHub – Quick Setup Guide

Prerequisites

Install the following before starting:

- Node.js (v16+) → `node --version`
- MongoDB (Local) **or** MongoDB Atlas (Cloud)
- Git → `git --version`
- VS Code (Recommended)

Installation Steps

Clone the Repository

bash

```
git clone <repository-url>
cd learnhub-project
```

Backend Setup

bash

```
cd backend
npm install
```

Create `.env` file and add

```
MONGODB_URI=your-mongo-uri
PORT=5000
JWT_SECRET=your-secret
FRONTEND_URL=http://localhost:5173
CLOUDINARY_CLOUD_NAME=...
CLOUDINARY_API_KEY=...
CLOUDINARY_API_SECRET=...
```

Start server:

```
bash
npm run dev
```

Frontend Setup

```
bash
        cd frontend
        npm install
        npm run dev
```

✓ Verification

- Backend: http://localhost:5000/api/health → `{ status: "OK" }`
- Frontend: http://localhost:5173
- MongoDB: Check terminal for "✓ Connected to MongoDB"

Common Issues

Problem	Fix
Port in use	Change `PORT` in `.env`
CORS errors	Ensure `FRONTEND_URL` is set correctly
MongoDB not connecting	Check URI or IP whitelist in MongoDB Atlas
JWT errors	Set `JWT_SECRET` in `.env` and restart server

Build for Production

```
bash
        cd frontend
        npm run build

        cd backend
        npm start
You're all set!
LearnHub now ready on:is
```

- **Backend → http://localhost:5000**
- **Frontend → http://localhost:5173**

LearnHub– Running Guide

Prerequisites

Ensure the following are completed before running the app:

- Node.js is installed

- MongoDB is running (local or Atlas)
- All dependencies are installed
- Environment variables are correctly configured in .env

⍰ Start the Application

Step 1: Start Backend

bash

```
cd backend
npm run dev
```

Verify:

- Console should show ✅ `Connected to MongoDB`
- Also: 🚀 `Server running on port 5000`

Step 2: Start Frontend

bash

```
cd frontend
npm run dev
```

Verify:

- App should open at: http://localhost:5173

Step 3: Access the App

Open your browser and visit:
http://localhost:5173
Register or log in to begin using LearnHub.

Useful Commands

📁 Backend

bash

```
cd backend
npm run dev        # Start dev server
npm start          # Start production server
npm run seed       # Seed database (optional)
```

🌐 Frontend

- bash

```
cd frontend
npm run dev        # Start dev server
npm run build      # Build for production
```

❌ Stop Servers

Press Ctrl + C in the terminal running

LearnHub API – Quick Reference

Base URL: http://localhost:5000/api
Authentication: Bearer Token (JWT)
Roles: `user` (student), `teacher`, `admin`

Authentication Endpoints

- `POST /auth/register` – Register user
- `POST /auth/login` – Login and get JWT
- `POST /auth/logout` – Logout (client-side token removal)
- `POST /auth/refresh` – Refresh JWT token

User Management

- GET /users/profile – Get current user profile
- PUT /users/profile – Update user profile
- GET /users – Admin: Get all users (with filters/pagination)

Course Management

- `GET /courses` – List all courses (filters supported)
- `GET /courses/:id` – Get course by ID
- `POST /courses` – Create course (teacher/admin)
- `PUT /courses/:id` – Update course
- `DELETE /courses/:id` – Delete course
- `GET /courses/teacher/my-courses` – Teacher's course list

Enrollments

- POST /enrollments – Enroll in a course
- GET /enrollments/user – Get enrolled courses
- PUT /enrollments/:id/complete – Mark as completed
- PUT /enrollments/:id/progress – Update progress

Certificates

- GET /certificates/user – List user's certificates
- POST /certificates/generate – Generate certificate

- GET /certificates/verify/:code – Verify certificate

Dashboards

- `GET /dashboard/user` – User stats (learning progress)
- `GET /dashboard/teacher` – Teacher stats (revenue, courses)
- `GET /dashboard/admin` – Platform stats (users, growth)

Search

- `GET /search/courses` – Search courses with filters

Required Environment Variables

- `MONGODB_URI`, `JWT_SECRET`, `JWT_EXPIRE`
- `FRONTEND_URL`, `CLOUDINARY_*`

LEARNHUB AUTHENTICATION SYSTEM

OVERVIEW

LearnHub uses a secure, token-based authentication system built with **JWT (JSON Web Tokens)**.
No sessions or cookies are used — authentication is fully stateless and handled via tokens.
All protected requests must include a valid JWT in the **Authorization** header.

AUTHENTICATION FLOW SUMMARY

USER REGISTRATION

- User fills and submits the registration form
- Backend validates all input fields
- Password is hashed using **bcrypt** (12 salt rounds)
- A **JWT token** is generated on successful registration
- Token is returned and stored in **localStorage** as `'learnhub_token'`

USER LOGIN

- User submits email and password
- Backend checks if email exists and password matches (using bcrypt)
- On success, a JWT token is generated and returned
- Token is stored in **localStorage** as `'learnhub_token'`

TOKEN VERIFICATION

- Frontend sends token in each protected request using the `Authorization` header
- Backend middleware validates the token’s **signature and expiration**
- If valid, request proceeds and user data is attached to `req.user`

JWT TOKEN STRUCTURE & CONFIGURATION

```
{

  "id": "user_id_here",

  "iat": <issued_at_timestamp>,

  "exp": <expiration_timestamp>

}
```

ROLE-BASED AUTHORIZATION

USER ROLES:

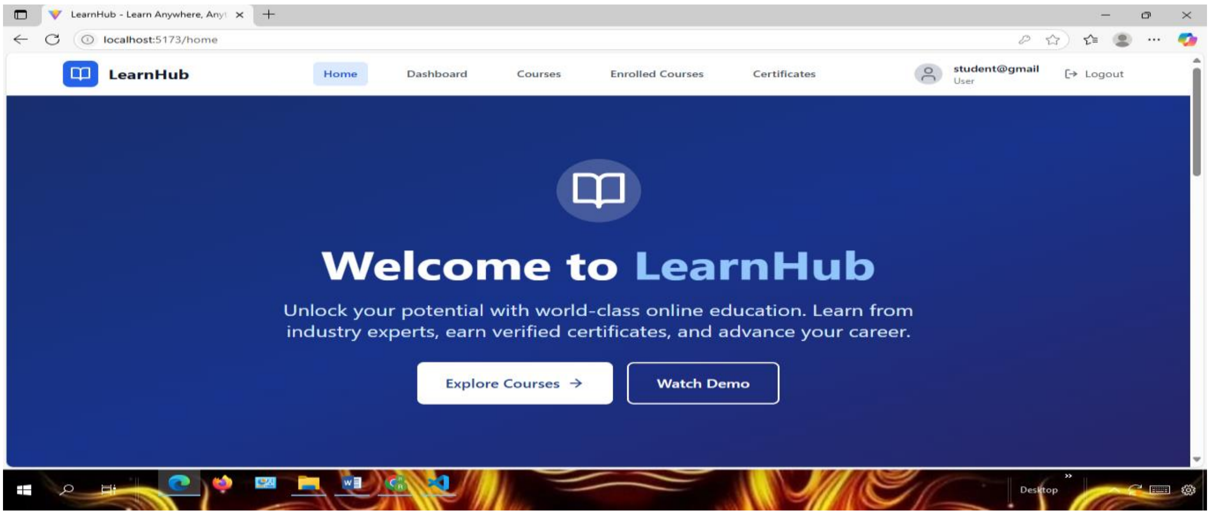
- `user` (Student): Can enroll in courses, access lessons, view certificates
- `teacher`: Can create/edit/delete courses, manage enrolled students
- `admin`: Full access to all data and user management

MIDDLEWARE FUNCTIONS:

- `auth`: Verifies JWT and attaches user info to request
- `authorize(...roles)`: Ensures the user has at least one allowed role

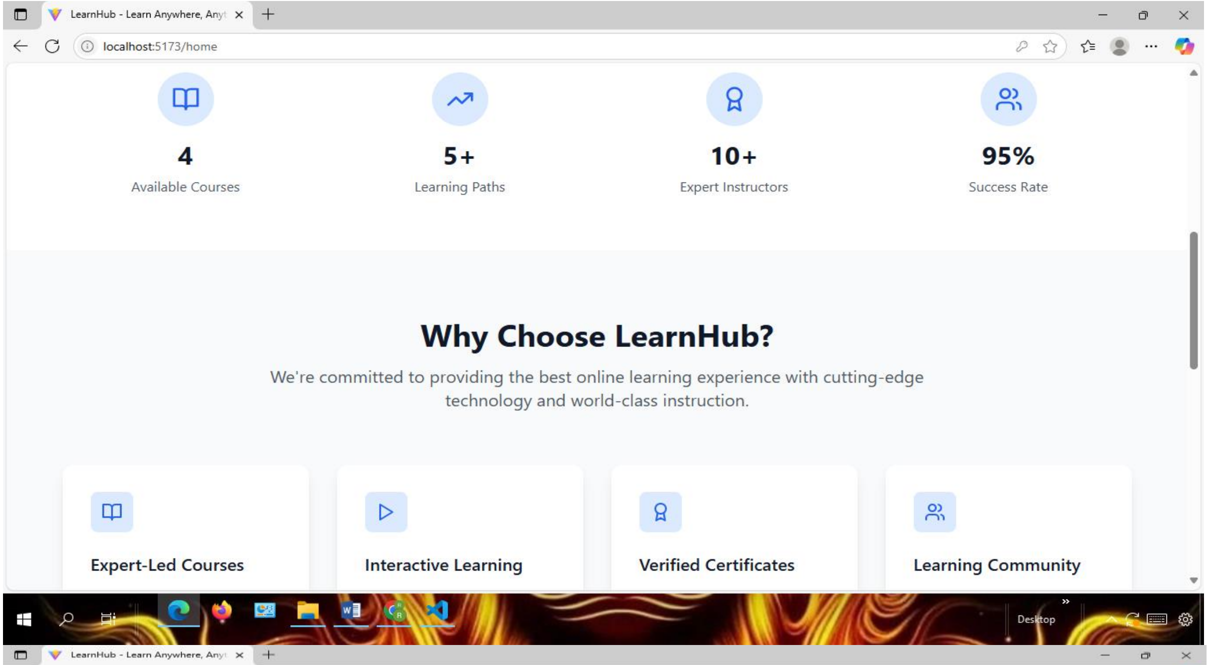
USER INTERFACE

Home Page Overview:



Welcome to LearnHub

Unlock your potential with world-class online education. Learn from industry experts, earn verified certificates, and advance your career.



4	5+	10+	95%
Available Courses	Learning Paths	Expert Instructors	Success Rate

Why Choose LearnHub?

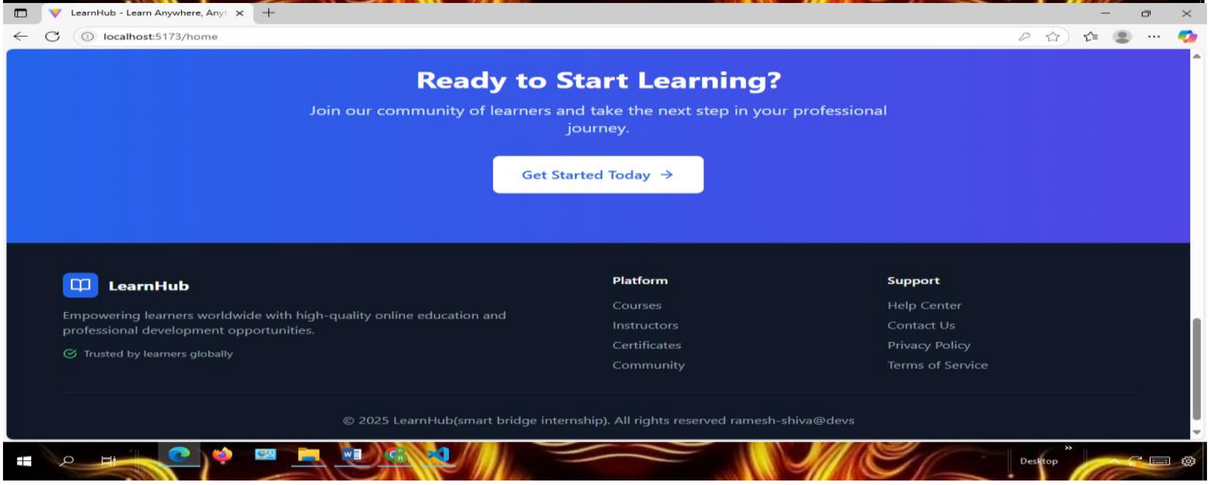
We're committed to providing the best online learning experience with cutting-edge technology and world-class instruction.

Expert-Led Courses

Interactive Learning

Verified Certificates

Learning Community



Ready to Start Learning?

Join our community of learners and take the next step in your professional journey.

LearnHub

Empowering learners worldwide with high-quality online education and professional development opportunities.

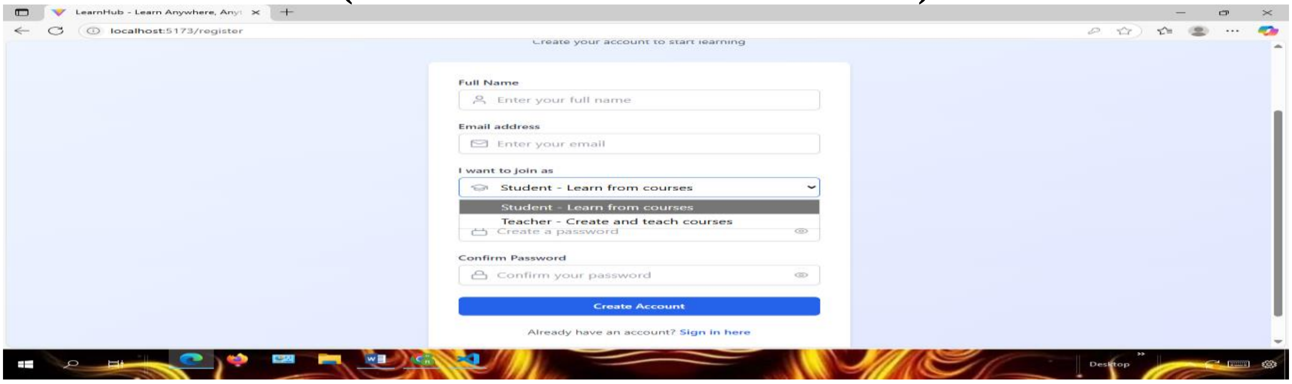
Trusted by learners globally

Platform
Courses
Instructors
Certificates
Community

Support
Help Center
Contact Us
Privacy Policy
Terms of Service

© 2025 LearnHub(smart bridge internship). All rights reserved ramesh-shiva@devs

REGISTER PAGE : (common for Students and Teachers)


LearnHub - Learn Anywhere, Any
localhost:5173/register
Create Account
Already have an account? Sign in here
Desktop

Create your account to start learning

Full Name: Enter your full name

Email address: Enter your email

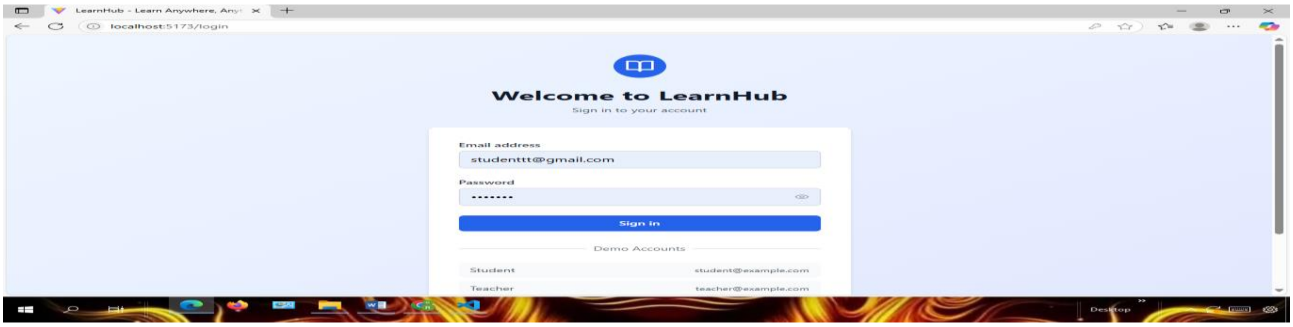
I want to join as: Student - Learn from courses

- Student - Learn from courses
- Teacher - Create and teach courses

Create a password

Confirm Password: Confirm your password

LOGIN PAGE:



Welcome to LearnHub

Sign in to your account

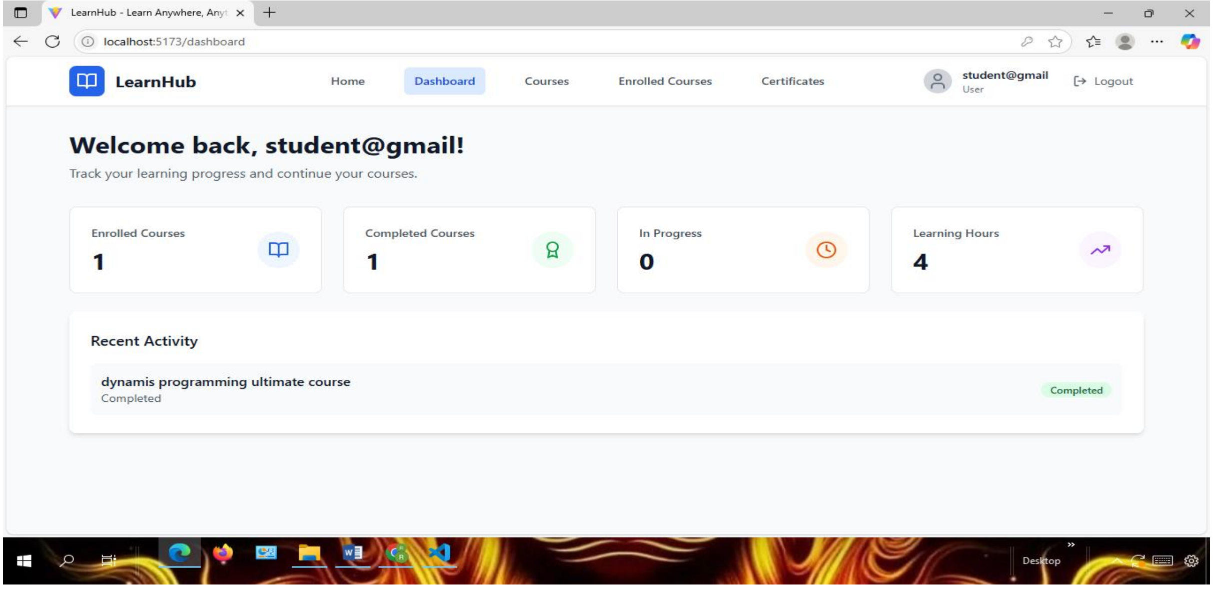
Email address: studenttt@gmail.com

Password: ••••••••

Demo Accounts

Student	student@example.com
Teacher	teacher@example.com

Student Dashboard Overview:



Welcome back, student@gmail!

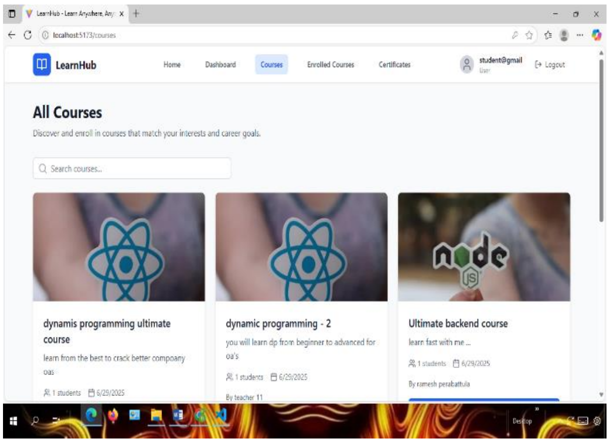
Track your learning progress and continue your courses.

Enrolled Courses	Completed Courses	In Progress	Learning Hours
1	1	0	4

Recent Activity

dynamis programming ultimate course
Completed — Completed

Courses Tab Overview:



localhost:5173/courses

LearnHub | Home | Dashboard | Courses | Enrolled Courses | Certificates | student@gmail User | Logout

All Courses

Discover and enroll in courses that match your interests and career goals.

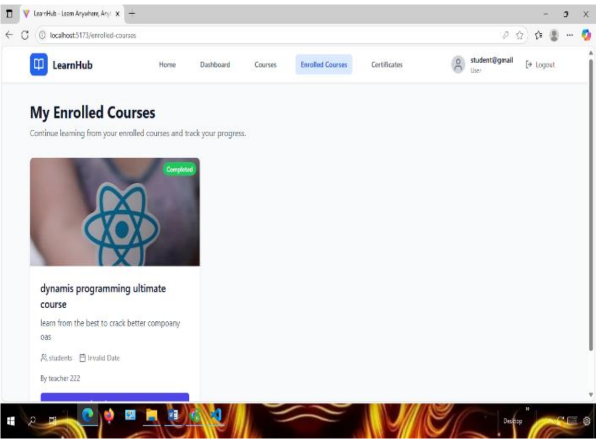
Search courses...

dynamis programming ultimate course
learn from the best to crack better compoany oas
1 students 6/29/2025

dynamic programming - 2
you will learn dp from beginner to advanced for oa's
1 students 6/29/2025
By teacher 11

Ultimate backend course
learn fast with me ...
1 students 6/29/2025
By ramesh perabattula

Enrolled courses Tab Overview:



localhost:5173/enrolled-courses

LearnHub | Home | Dashboard | Courses | Enrolled Courses | Certificates | student@gmail User | Logout

My Enrolled Courses

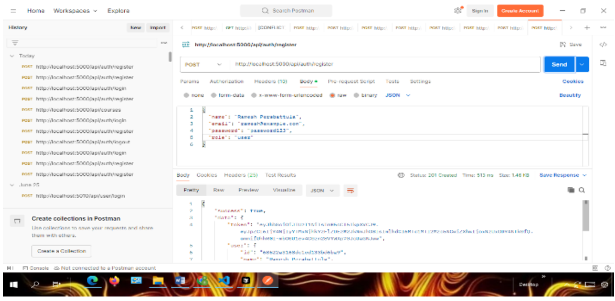
Continue learning from your enrolled courses and track your progress.

Completed

dynamis programming ultimate course
learn from the best to crack better compoany oas
students Invalid Date
By teacher 222

API TESTING — POSTMAN TOOL

Register & Login API Test (http://localhost:5000/api/auth/register or /login):



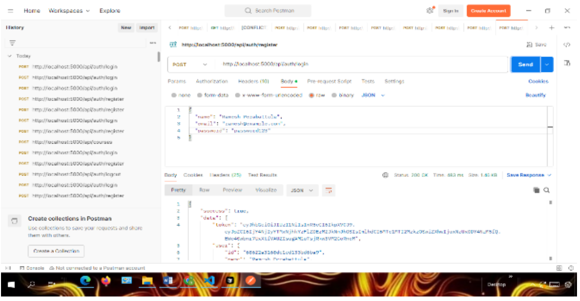
POST http://localhost:5000/api/auth/register

Body: raw JSON

```
{
"name": "Ramesh Perabattula",
"email": "ramesh@example.com",
"password": "password123",
"role": "user"
}
```

Status: 201 Created Time: 513 ms Size: 1.46 KB

```
{
    "success": true,
    "data": {
        "token": "[illegible]",
        "user": {
            "id": "68622e3168dc1cd133bd6ba9",
            "name": "Ramesh Perabattula",
```



POST http://localhost:5000/api/auth/login

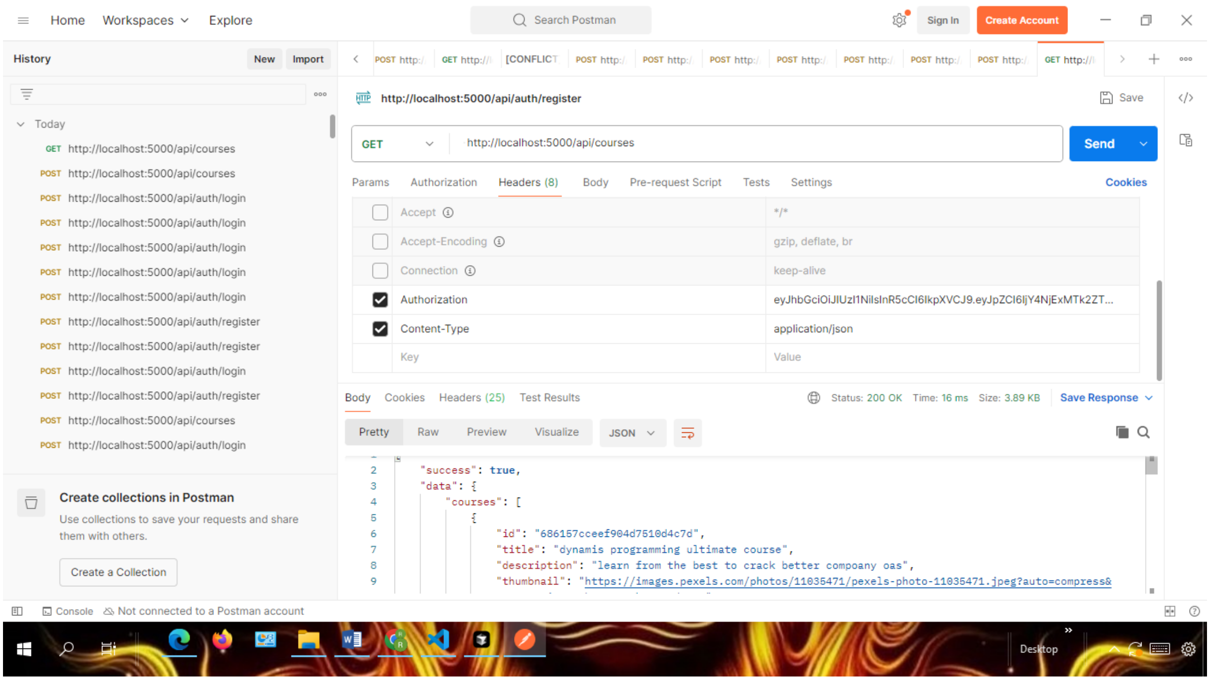
Body: raw JSON

```
{
"name": "Ramesh Perabattula",
"email": "ramesh@example.com",
"password": "password123"
}
```

Status: 200 OK Time: 483 ms Size: 1.40 KB

```
{
    "success": true,
    "data": {
        "token": "[illegible]",
        "user": {
            "id": "68622e3168dc1cd133bd6ba9",
            "name": "Ramesh Perabattula",
```

Admin getAllCourses & getAllUsers API Test(http://localhost:5000/api/)



Home
Workspaces
Explore
Search Postman
Sign In
Create Account
History
New
Import
Create collections in Postman
Use collections to save your requests and share them with others.
Create a Collection
Console
Not connected to a Postman account

History – Today:

- GET http://localhost:5000/api/courses
- POST http://localhost:5000/api/courses
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/register
- POST http://localhost:5000/api/auth/register
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/register
- POST http://localhost:5000/api/courses
- POST http://localhost:5000/api/auth/login

http://localhost:5000/api/auth/register

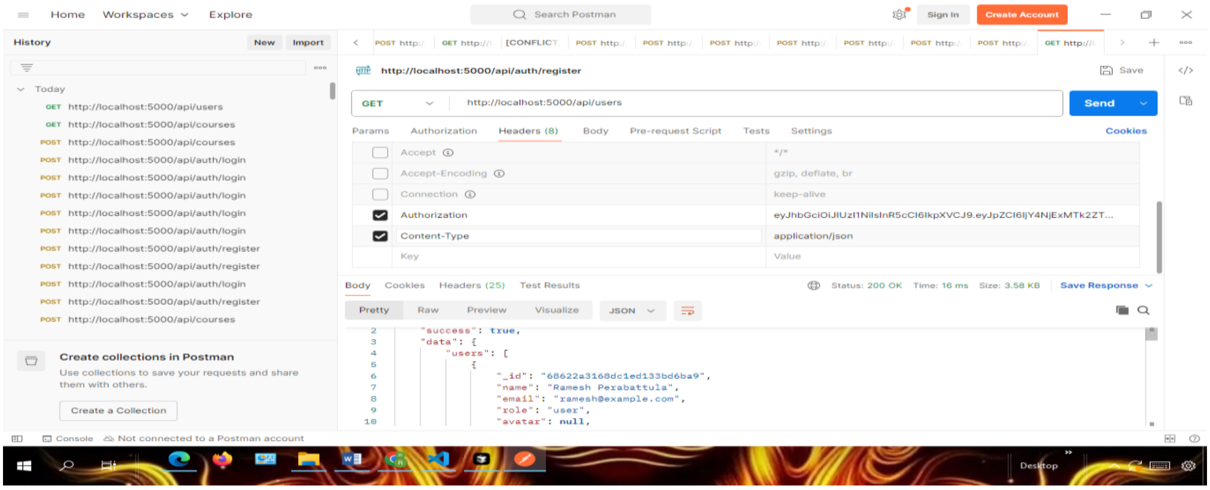
GET http://localhost:5000/api/courses

Headers (8)

	Key	Value
□	Accept	*/*
□	Accept-Encoding	gzip, deflate, br
□	Connection	keep-alive
☑	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4NjExMTk2ZT...
☑	Content-Type	application/json
	Key	Value

Status: 200 OK Time: 16 ms Size: 3.89 KB

```
    "success": true,
    "data": {
        "courses": [
            {
                "id": "686157cceef904d7510d4c7d",
                "title": "dynamis programming ultimate course",
                "description": "learn from the best to crack better compoany oas",
                "thumbnail": "https://images.pexels.com/photos/11035471/pexels-photo-11035471.jpeg?auto=compress&
```



- GET http://localhost:5000/api/users
- GET http://localhost:5000/api/courses
- POST http://localhost:5000/api/courses
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/register
- POST http://localhost:5000/api/auth/register
- POST http://localhost:5000/api/auth/login
- POST http://localhost:5000/api/auth/register
- POST http://localhost:5000/api/courses

Create collections in Postman

Use collections to save your requests and share them with others.

http://localhost:5000/api/auth/register

GET http://localhost:5000/api/users

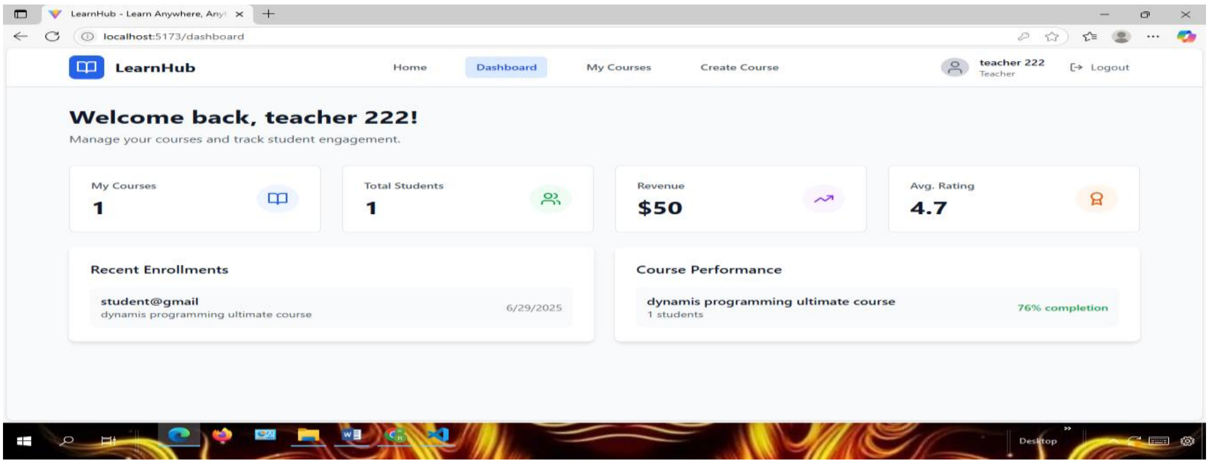
□	Accept	*/*
□	Accept-Encoding	gzip, deflate, br
□	Connection	keep-alive
☑	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY4NjExMTk2ZT...
☑	Content-Type	application/json
	Key	Value

Status: 200 OK Time: 16 ms Size: 3.58 KB

```
    "success": true,
    "data": {
        "users": [
            {
                "_id": "68622a3168dc1ed133bd6ba9",
                "name": "Ramesh Perabattula",
                "email": "ramesh@example.com",
                "role": "user",
                "avatar": null,
```

SCREENSHOTS / DEMO SECTION

Teacher Dashboard Overview:



Welcome back, teacher 222!

Manage your courses and track student engagement.

My Courses	Total Students	Revenue	Avg. Rating
1	1	$50	4.7

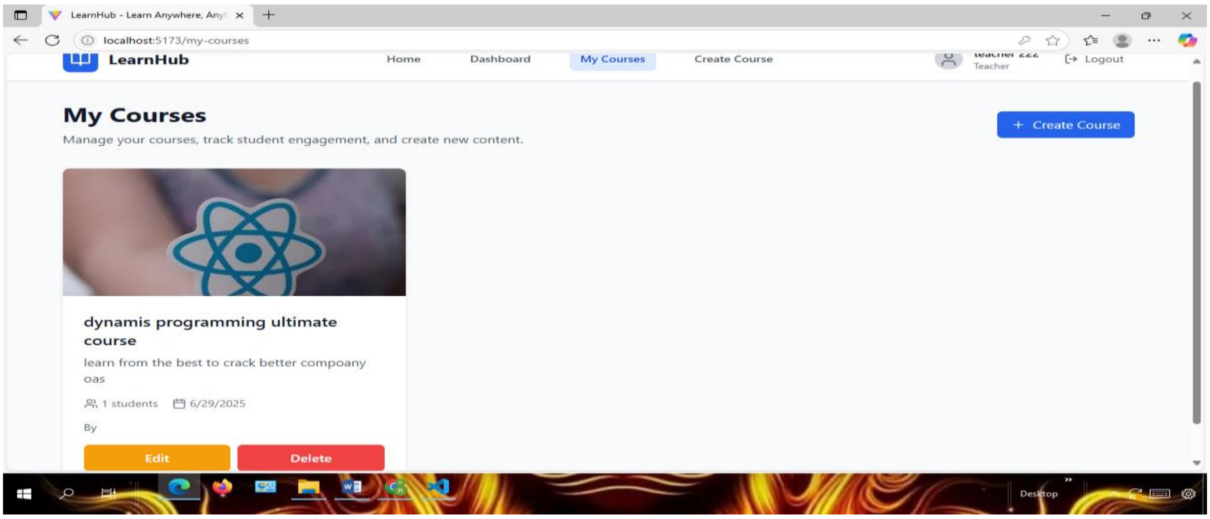
Recent Enrollments

student@gmail — dynamis programming ultimate course — 6/29/2025

Course Performance

dynamis programming ultimate course — 1 students — 76% completion

Teacher My Couses Tab Overview:



My Courses

Manage your courses, track student engagement, and create new content.

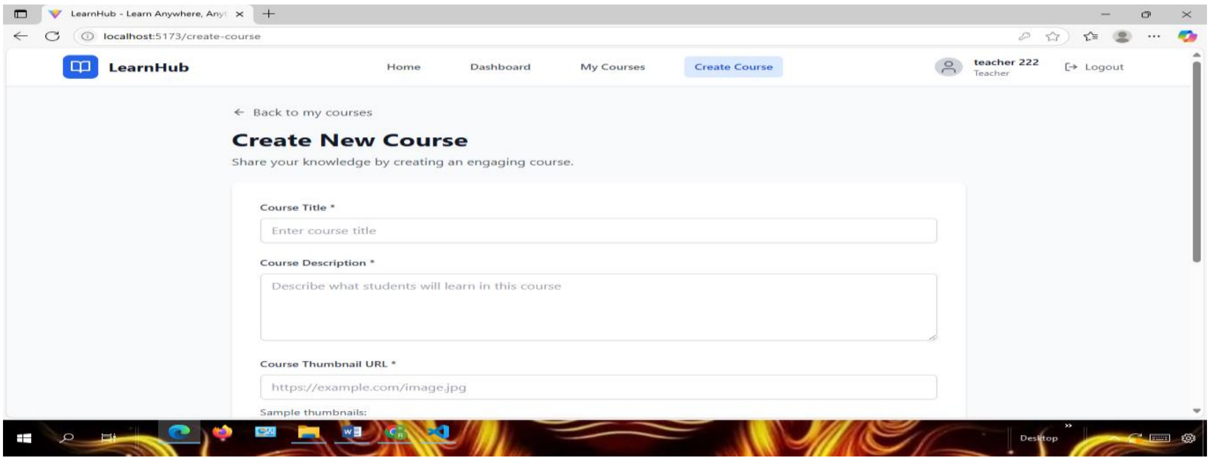
dynamis programming ultimate course

learn from the best to crack better compoany oas

1 students 6/29/2025

By

Create Course Tab Overview:



← Back to my courses

Create New Course

Share your knowledge by creating an engaging course.

Course Title *

Enter course title

Course Description *

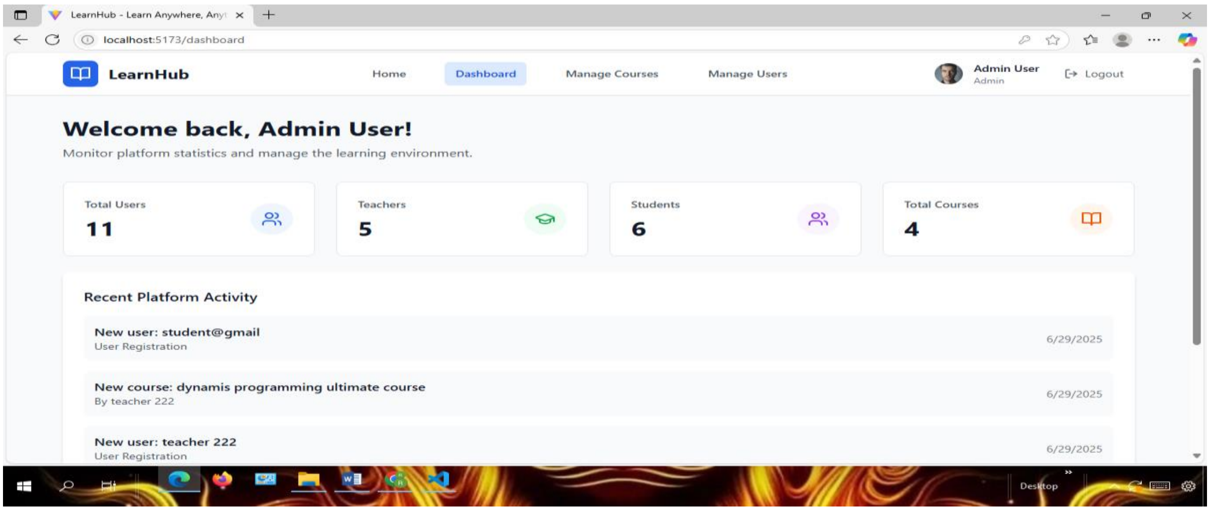
Describe what students will learn in this course

Course Thumbnail URL *

https://example.com/image.jpg

Sample thumbnails:

Admin Dashboard Overview:



Welcome back, Admin User!

Monitor platform statistics and manage the learning environment.

Total Users	Teachers	Students	Total Courses
11	5	6	4

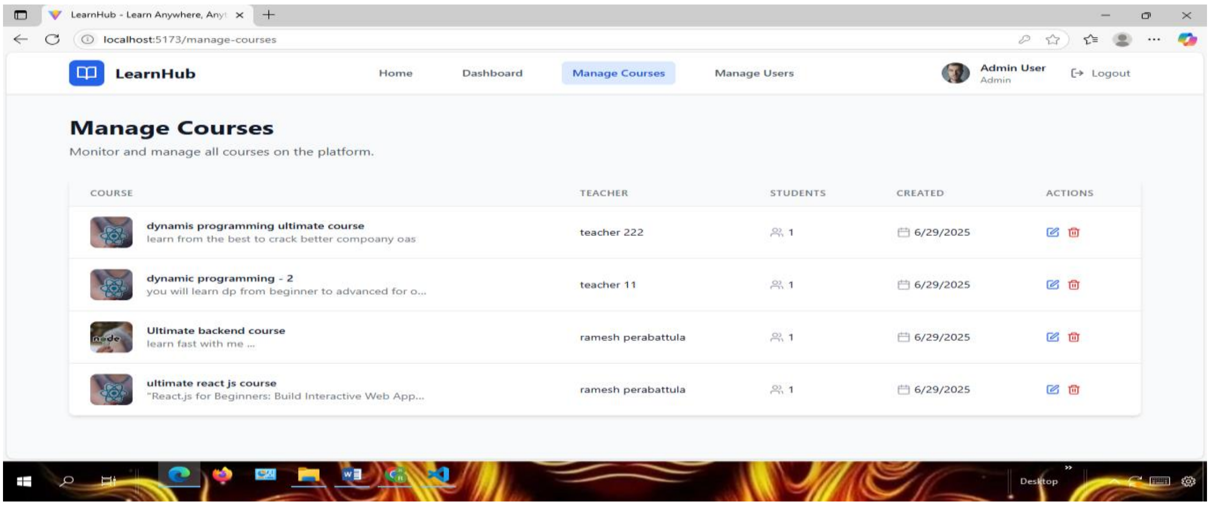
Recent Platform Activity

New user: student@gmail
User Registration — 6/29/2025

New course: dynamis programming ultimate course
By teacher 222 — 6/29/2025

New user: teacher 222
User Registration — 6/29/2025

Manage Courses Tab Overview (Admin can perform Update and Delete operations):

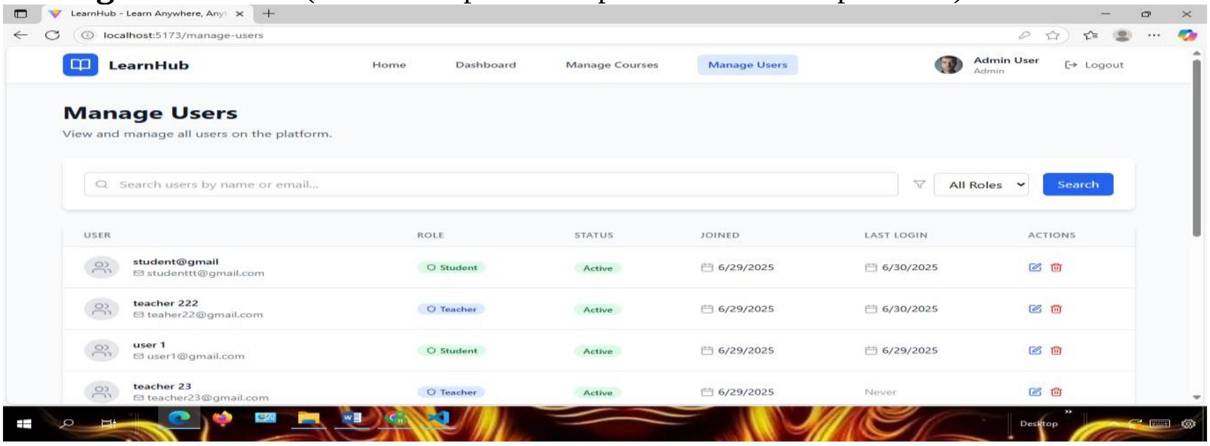


Manage Courses

Monitor and manage all courses on the platform.

COURSE	TEACHER	STUDENTS	CREATED	ACTIONS
dynamis programming ultimate course learn from the best to crack better compoany oas	teacher 222	1	6/29/2025	✎ 🗑
dynamic programming - 2 you will learn dp from beginner to advanced for o...	teacher 11	1	6/29/2025	✎ 🗑
Ultimate backend course learn fast with me ...	ramesh perabattula	1	6/29/2025	✎ 🗑
ultimate react js course "React.js for Beginners: Build Interactive Web App...	ramesh perabattula	1	6/29/2025	✎ 🗑

Manage Users Tab(Admin can perform Update and delete Operations)



Manage Users

View and manage all users on the platform.

USER	ROLE	STATUS	JOINED	LAST LOGIN	ACTIONS
student@gmail studenttt@gmail.com	Student	Active	6/29/2025	6/30/2025	
teacher 222 teaher22@gmail.com	Teacher	Active	6/29/2025	6/30/2025	
user 1 user1@gmail.com	Student	Active	6/29/2025	6/29/2025	
teacher 23 teacher23@gmail.com	Teacher	Active	6/29/2025	Never	

LearnHub – Known Issues

Frontend Issues

1. Token Expiration

JWT expires after 7 days; no auto-refresh.
Workaround: Re-login manually.

2. File Upload Limit

Max 10MB upload size; large videos fail.
Workaround: Compress files or use external links.

3. Mobile Responsiveness

Some pages not mobile-friendly.
Workaround: Use desktop for full features.

Backend Issues

4. Rate Limiting

100 requests/15 mins may block active users.
Workaround: Add caching or increase limits.

5. DB Connection Drops

MongoDB may timeout when idle.
Workaround: Restart backend if needed.

6. File Storage Dependency

Fails if Cloudinary is down.
Workaround: Ensure config or add fallback.

7. Slow Data Loading

No pagination; large datasets lag.
Workaround: Add pagination or lazy loading.

8. Unoptimized Images
Big images slow down pages.
Workaround: Compress before upload.

FUTURE ENHANCEMENTS

LearnHub – Future Enhancements

1. Authentication & Security

- Add OAuth, SSO, and MFA
- Granular role-based access
- Session timeout & audit logging
- Strong password rules

2. Course Management

- Multimedia content editor
- Course templates & version control
- Quizzes, prerequisites, and progress tracking
- Support for live webinars

3. User Experience

- AI-based course recommendations
- Personalized dashboards & adaptive learning
- Gamification and social features

4. Mobile Support

- Native iOS/Android apps
- Offline access & push notifications
- Progressive Web App (PWA) support

5. Analytics & Reporting

- Dashboards for all roles
- Custom reports and exports
- Predictive analytics for at-risk learners

6. Live Learning

- Virtual classroom with chat, polls, and recording
- Integration with Zoom, Teams, etc.