

LearnHub: Your Center for Skill Enhancement

1. Introduction

LearnHub is a modern Online Learning Platform (OLP) designed to facilitate and enhance the process of skill acquisition through an intuitive and interactive interface. This platform enables students, professionals, and educators to connect and share knowledge via self-paced, structured learning modules.

Team Members:

1. Student1(Team head): Kurmala Theerdha Sree(22F01A4957).
 2. Student2: Vasupalli Gowthami(22F01A4961).
 3. Student3: Shaik Shafiya(22F01A4950).
 4. Student4: Vamani Jahnavi Mahalakshmi(22F01A4959).
-

2. Project Overview

LearnHub aims to provide a comprehensive digital environment where users can:

- Register and browse courses
- Enroll in free or paid learning modules
- Track learning progress and receive certifications
- Interact with peers and instructors via forums and webinars
- Access content from any device

Roles supported:

- **Student** (Learner): Browse, enroll, complete courses
 - **Teacher** (Instructor): Create, manage courses
 - **Admin**: Oversee platform operations and user activity
-

3. Technical Architecture

The LearnHub application is built on a full-stack architecture with the following components:

- **Frontend**: React.js with Bootstrap and Material UI
- **Backend**: Node.js + Express.js
- **Database**: MongoDB (NoSQL)
- **API Communication**: RESTful APIs using Axios

Flow: Client (React) → Axios → Express.js Backend → MongoDB

4. Setup Information

Pre-requisites:

- **Node.js & npm:** [Download](#)
 - **Vite:** npm create vite@latest
 - **Express.js:** npm install express
 - **MongoDB:** [Download](#)
 - **React.js:** Installed via Vite scaffolding
 - **Mongoose (ODM):** npm install mongoose
 - **Bootstrap & Material UI:** npm install bootstrap @mui/material
-

5. Folder Structure

```
containment-zone/  
├── frontend/  
│   ├── public/  
│   ├── src/  
│   │   ├── components/  
│   │   ├── pages/  
│   │   ├── App.jsx  
│   │   └── main.jsx  
│   └── package.json  
└── backend/  
    ├── models/  
    ├── routes/  
    ├── controllers/  
    ├── config/  
    ├── server.js  
    └── package.json
```

6. Running the Application

Clone the repository

cd containment-zone

Frontend Setup

cd frontend

npm install

Backend Setup

cd ../backend

npm install

Start Server

npm start

Application will run at: <http://localhost:5172>

7. API Documentation

Authentication APIs:

- POST /api/auth/register – Register user
- POST /api/auth/login – Login user

Courses APIs:

- GET /api/courses – Fetch all courses
 - POST /api/courses – Add new course
 - GET /api/courses/:id – Fetch specific course
 - PUT /api/courses/:id – Update course
 - DELETE /api/courses/:id – Delete course
-

8. Authentication

JWT (JSON Web Tokens) are used to authenticate and authorize users. Upon login, a token is issued and attached to subsequent API calls.

9. User Interface

- Dashboard (role-based UI)
- Course browsing and filtering
- Enrollments and learning progress tracking
- Certificate download
- Admin and teacher panels

Technologies used: React.js + Material UI + Bootstrap

10. Testing

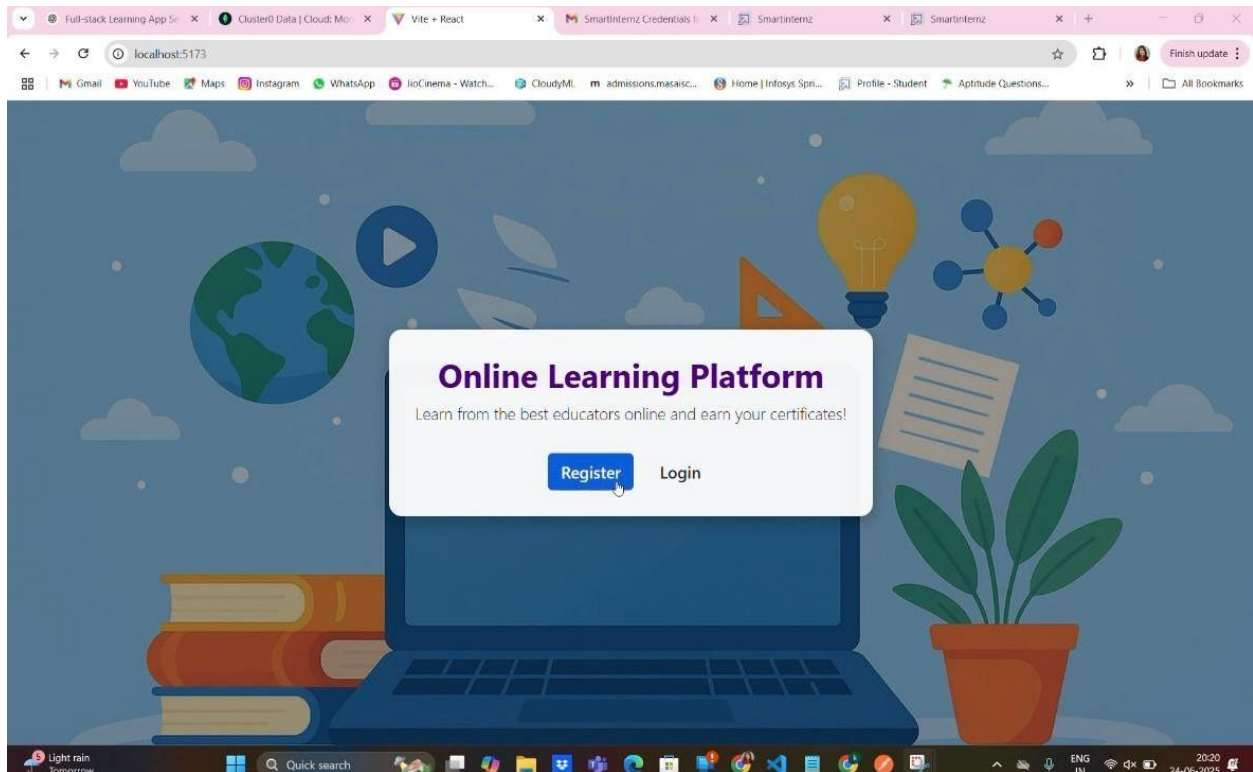
Manual testing steps:

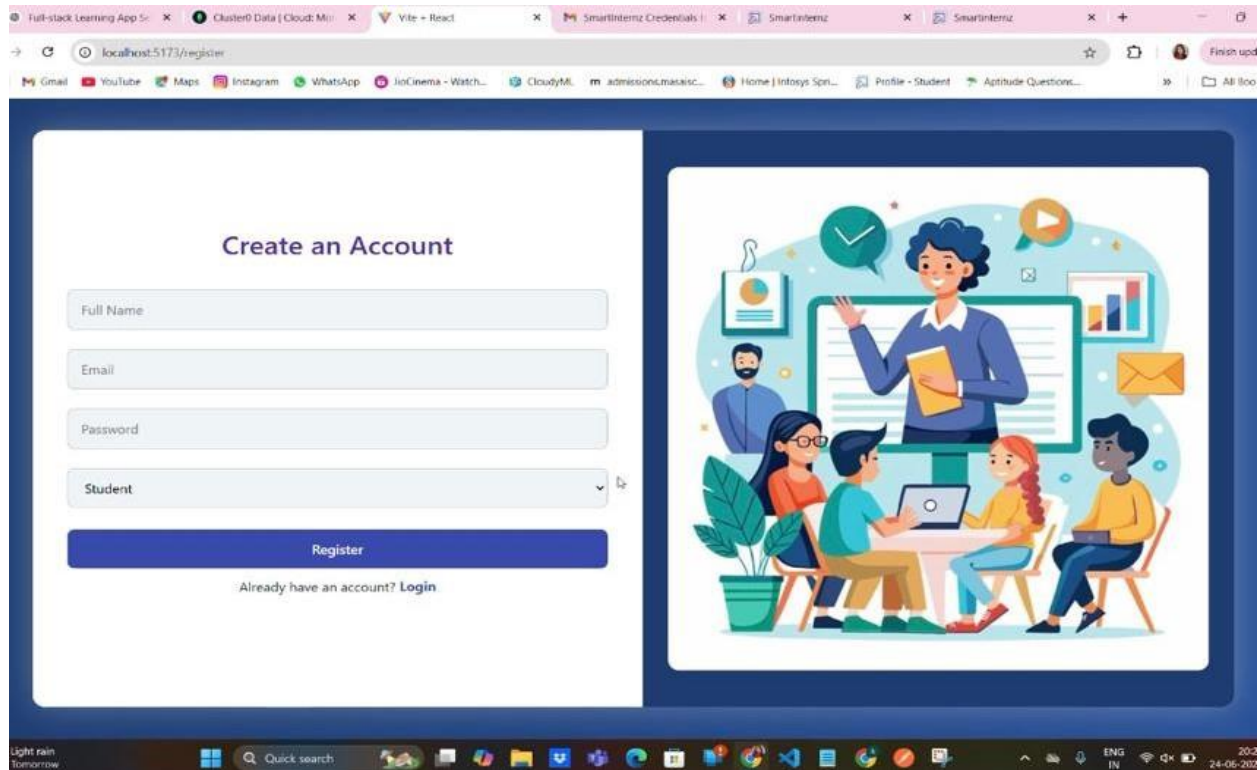
- Register/login flow
- Course enrollment and completion
- Payment flow (mocked)

Automated testing: Can be integrated using Jest for unit and integration tests.

11. Screenshots / Demo

Screenshots to be added from UI views.





12. Known Issues

- Real-time notifications are not integrated
 - Payment gateway is mocked
 - No offline functionality
-

13. Future Enhancements

- Integrate real-time notifications and messaging
- AI-based course recommendation engine
- Video conferencing via WebRTC or integration with Zoom/Meet
- Mobile app version (React Native)
- Enhanced analytics for instructors and admin