**1. INTRODUCTION**

1.1 Project Overview

1.2 Purpose

**2. IDEATION PHASE**

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

**3. REQUIREMENT ANALYSIS**

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

**4. PROJECT DESIGN**

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

**5. PROJECT PLANNING & SCHEDULING**

5.1 Project Planning

**6. FUNCTIONAL AND PERFORMANCE TESTING**

6.1 Performance Testing

**7. RESULTS**

7.1 Output Screenshots

**8. ADVANTAGES & DISADVANTAGES**

**9. CONCLUSION**

**10. FUTURE SCOPE**

**11. APPENDIX**

Source Code(if any)

Dataset Link

GitHub & Project Demo LinkProject Report – Learn Hub

**1. INTRODUCTION**

**1.1 Project Overview**

**Learn Hub** is a full-stack MERN (MongoDB, Express.js, React.js, Node.js) based online learning platform designed for teachers to create courses, students to enroll and learn, and admins to manage the system. It supports video uploads, student progress tracking, course recommendations, and secure enrollment via mock payments.

**1.2 Purpose**

To build an intuitive, scalable platform that bridges the gap between educators and learners with interactive course content, tracking, and administrative control.

**2. IDEATION PHASE**

**2.1 Problem Statement**

With the rise of online education, many platforms lack personalization, performance tracking, or are costly. There is a need for an affordable and efficient system where teachers can easily create content, and students can access and learn with progress tracking.

**2.2 Empathy Map Canvas**

* **Think & Feel**: Learners seek quality, flexible learning with structure.
* **See**: Overwhelming platforms without clear navigation.
* **Hear**: Feedback about expensive or inaccessible courses.
* **Say & Do**: Prefer learning at their own pace.
* **Pain**: Disjointed learning experience, no progress tracking.
* **Gain**: Smooth, tracked, and engaging learning journey.

**2.3 Brainstorming**

* Modular course creation
* Student dashboards with visual progress
* Secure payment and enrollment
* Admin monitoring panel
* Recommendation engine

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

1. **Visitor** visits platform
2. **Registers/Login** as Student or Teacher
3. **Teacher** creates and uploads course
4. **Student** views, enrolls (with mock payment), and learns
5. **Progress is tracked** and reported
6. **Admin** monitors the platform

**3.2 Solution Requirement**

* User Authentication
* Role-based access
* Course creation (with video/image support)
* Payment integration
* Real-time progress tracking
* Admin dashboard

**3.3 Data Flow Diagram**

**Level 0 & Level 1** DFDs showing interaction between:

* Frontend ↔ Backend APIs ↔ Database
* Teacher ↔ Course Module ↔ Student ↔ Progress Tracker

**3.4 Technology Stack**

* **Frontend**: React.js, Tailwind CSS
* **Backend**: Node.js, Express.js
* **Database**: MongoDB
* **Storage**: Cloudinary (for videos/images)
* **Other Tools**: Postman (API Testing), GitHub, Render/Vercel

**4. PROJECT DESIGN**

**4.1 Problem-Solution Fit**

The platform directly addresses the need for a structured, personalized online learning system with clear role separation and learning analytics.

**4.2 Proposed Solution**

An online platform that allows:

* Teachers to create and manage content
* Students to learn and track their own progress
* Admins to view data and manage users

**4.3 Solution Architecture**

**Client–Server Architecture**

* React Frontend calls Express APIs
* APIs connect to MongoDB for data storage
* File uploads managed via Cloudinary
* Role-based routing and rendering

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

| **Week** | **Tasks** |
| --- | --- |
| 1 | Requirement gathering, wireframes |
| 2 | Backend setup (Node + MongoDB), API creation |
| 3 | Frontend: Login/Signup, Dashboards |
| 4 | Course creation and file upload integration |
| 5 | Enrollment + Payment integration |
| 6 | Progress tracking + Admin panel |
| 7 | Testing, Bug Fixing, UI Enhancements |

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

* All APIs tested using **Postman**
* Response times averaged < 400ms for CRUD operations
* Course load time optimized using lazy-loading of videos
* UAT completed with feedback from 5 sample users

**7. RESULTS**

**7.1 Output Screenshots**

* Login & Signup
* Teacher Dashboard
* Course Creation Page
* Student Dashboard with Enrolled Courses
* Video Section & Progress Tracker
* Admin Dashboard with User Summary

**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* Clear role separation (Student, Teacher, Admin)
* Easy course creation and content upload
* Progress tracking & recommendations
* Scalable and modular

**Disadvantages**

* No real payment gateway (mock used)
* Limited UI animations
* Basic design (no mobile app version yet)

**9. CONCLUSION**

**Learn Hub** successfully delivers a role-based online learning environment that supports both teaching and learning in a user-friendly way. It demonstrates the effective use of full-stack development for a practical, scalable solution.

**10. FUTURE SCOPE**

* Real Payment Gateway (Razorpay/Stripe)
* AI-based course recommendation engine
* Certificate generation
* Mobile App (React Native)
* Live class integration (Zoom API)

**11. APPENDIX**

* **Source Code**: Included in GitHub Repository
* **Dataset Link**: [If used, insert Google Drive/CSV Link]
* **GitHub Link** <https://github.com/botusreya/learn-hub>