LEARNING PATH DASHBOARD FOR ENHANCING SKILLS

A

MINOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELORS OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 27

Akash Prajapati	**0187CS221027**
Adnan Baig	**0187CS221022**
Aakash Tiwari	**0187CS221001**
Ashish Mehra	**0187CS221061**

Under the guidance of

Dr. Komal Tahiliani

(Associate Professor)

Department of Computer Science & Engineering
Sagar Institute of Science & Technology (SISTec), Bhopal(M.P)

Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P)

May - 2025

Sagar Institute of Science & Technology (SISTec), Bhopal
Department of COMPUTER SCIENCE & ENGINEERING
Bhopal (M.P.)

CERTIFICATE

We hereby certify that the work presented in the B.Tech. Minor Project-II Report entitled **LEARNING PATH DASHBOARD FOR ENHANCING SKILLS SYSTEM,** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering, and submitted to the Department of Computer Science & Engineering, Sagar Institute of Science & Technology (SISTec), Bhopal (M.P.), is an authentic record of our own work carried out during the period from **Jan-2025 to May-2025** under the supervision of **Dr. Komal Tahiliani**.

The content presented in this project has not been submitted by us for the award of any other degree elsewhere.

Akash Prajapati	**Adnan Baig**	**Aakash Tiwari**	**Ashish Mehra**
0187CS221027	**0187CS221022**	**0187CS221001**	**0187CS221061**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Komal Tahiliani	**Prof. Nargish Gupta**	**Dr. Manish Billore**
Project Guide	**HOD**	**Principal**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. Manish Billore**, **Principal, SISTec** and **Dr. Swati Saxena, Vice Principal, SISTec** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Prof. Nargish Gupta, HOD, Department of Computer Science & Engineering** for his kindhearted support.

We extend our sincere and heartfelt thanks to our guide, **Dr. Komal Tahiliani**, for providing us with the right guidance and advice at crucial junctures and for showing us the right way**.**

I am thankful to the **Project Coordinator, Prof. Mayank Kurchaniya,** who devoted his precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue.

TABLE OF CONTENTS

ABSTRACT

In the rapidly evolving digital world, continuous skill enhancement is crucial for academic and professional growth. The "Learning Path Dashboard for Enhancing Skills System" is an intelligent, interactive platform designed to assist students and professionals in identifying, planning, and tracking personalized learning journeys. This system integrates user profiles, skill assessments, and career goals to recommend tailored learning paths using data-driven insights.

The dashboard offers a centralized interface that visualizes learning progress, suggests relevant courses, and tracks achievements through performance analytics and milestones. By leveraging concepts from machine learning and data visualization, it adapts over time based on user interaction and feedback. The system includes modules for skill gap analysis, goal setting, recommended content (e.g., courses, articles, projects), and progress monitoring.

The primary objective is to empower users to take control of their learning process by providing a structured, goal-oriented environment that aligns with industry trends and personal aspirations. This tool has potential applications in educational institutions, corporate training, and self-paced learning communities, ensuring that skills.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
UI	User Interface
UX	User Experience
CMS	Content Management System
API	Application Programming Interface
MVC	Model View Controller
DB	Database
JSON	JavaScript Object Notation
CRUD	Create, Read, Update, Delete
DFD	Data Flow Diagram
JS	Data Flow Diagram
CSS	Cascading Style Sheets

List of Figure

CHAPTER-1
INTRODUCTION

1.1 ABOUT PROJECT

The Learning Path Dashboard aims to bridge the gap between educational content and personalized skill development by offering a dynamic, resource-rich interface that guides learners through well-structured learning paths. A user-friendly platform that shows reading statistics (e.g., time spent on topics, skill progress). Real-time updates on a learner's journey visual indicators like completion percentage and skills acquired. Aims to assist students in staying on track with their educational goals and helping them make informed decisions about their coursework.

1.1.1 KEY FEATURES

1 Personalized Learning Paths

- Recommends courses based on user interests and goals
- Adapts to user progress and feedback.

2 Skill Gap Analysis

- Compares current skills with desired roles
- Highlights areas for improvement

3 Progress Tracking Dashboard

- Visual graphs showing learning progress.
- Tracks completed and pending tasks

4 Smart Content Recommendations

- Suggests relevant courses, articles, and videos.
- Integrates with popular platforms like YouTube and Coursera.

5 Admin and Content Control Panel

- Admins can add or update learning materials.
- Monitor user activity and engagement.

1.1.2 BENEFITS

1 Improved Skill Development

- Offers a structured, goal-based approach to acquiring new skills rather than random or scattered learning.
- Encourages learning based on user strengths and weaknesses, ensuring faster growth in relevant areas.

2 Time-Saving and Efficient Learning

- Saves time by curating quality resources, eliminating the need for users to search multiple platforms.
- Prioritizes essential topics to avoid overwhelming users with unnecessary information.

3 Personalized and Adaptive Experience

- Customizes the dashboard based on user profile, previous learning history, and career goals.
- Adjusts content and suggestions dynamically based on user interaction and performance.

4 Career-Oriented Learning

- Aligns skills development with job roles, certifications, and market demand.
- Suggests practical projects to build real-world experience and portfolio strength.
- Enhances employability by identifying and addressing critical skill gaps.

5 High Accessibility and Flexibility

- Provides a seamless learning experience across multiple devices including mobile, tablet, and desktop.
- Allows learning at any time and from anywhere, making it ideal for students and working professionals.
- Supports both self-paced and scheduled learning modes.

6 Increased Engagement Through Gamification

- Keeps users motivated with badges, levels, points, and leaderboards.
- Encourages consistent learning behavior through visual rewards and progress tracking.
- Makes the learning process interactive and less monotonous.

1.2 PROJECT OBJECTIVE

The objective of this project is to design an intelligent dashboard that guides users through personalized learning paths, identifies skill gaps, recommends relevant resources, and tracks progress. It aims to enhance skill development efficiently, support career growth, and promote self-paced, goal-oriented learning using data-driven strategies and user-centric design.

1.2.1 TECHNICAL APPROACH

1 Front-End Development

- Responsive dashboard interface for various screen sizes (desktop, tablet, mobile).
- Dynamic content rendering for personalized learning paths and progress charts.
- User-friendly navigation, interactive forms, and intuitive controls.

2 Back-End Development

- Handles user requests, processes logic, and communicates with the database.
- Manages modules like user authentication, goal tracking, and admin controls.

3 User Authentication and Profile Management

- Secure registration and login system using encryption.
- Role-based access control for users and administrators.

4 Learning Recommendation Engine

- Suggests personalized courses, articles, and videos from internal or external databases.
- Updates suggestions based on user feedback and progress.
- Uses logic-based rules by using Clerk.

5 Admin Dashboard

- Admins can add/update courses, skills, learning paths, and user permissions.
- Monitor user progress and system activity through analytics.

6 Future Enhancement Scope

- Add to refine recommendations based on behavior patterns.
- Predict user success rate or learning style.
- Reward systems, badge allocation, leaderboards for increased engagement.

CHAPTER-2
SOFTWARE & HARDWARE REQUIREMENTS

2.1 INTRODUCTION

In today's competitive and technology-driven world, structured skill development is crucial for academic and career success. The "Learning Path Dashboard for Enhancing Skills System" aims to provide users with a centralized platform to plan, track, and manage their learning goals. It offers features like personalized dashboards, skill gap analysis, goal setting, and progress monitoring. This system enhances user productivity by organizing learning tasks and providing visual insights, ultimately helping learners stay focused, consistent, and self-motivated.

2.2 SOFTWARE REQUIREMENTS

1 React.js

- A powerful JavaScript library used for building dynamic, responsive user interfaces.
- Allows component-based development, making the front-end modular, reusable, and easy to maintain.

2 Integrated Development Environment (IDE)

- VS Code: A lightweight and feature-rich coding environment.
- Provides debugging tools, extensions, and an intuitive interface for seamless development.

3 Next.js and Node.js

- Next.js is a React framework that enables server-side rendering (SSR) and static site generation (SSG).
- Nodejs is used facilitates the back-end of the application, handling APIs, server-side authentication, and database communication.

4 Hygraph

- A headless CMS (Content Management System) used for managing and delivering content via GraphQL APIs.
- Integrates easily with front-end frameworks like Next.js for content-driven experiences.

5 Clerk

- An authentication and user management platform.
- Provides pre-built components for login, sign-up, profile management, and secure session handling.

2.3 HARDWARE REQIUREMENTS

1 Computer with Internet

- A consistent internet connection is essential for accessing the dashboard, fetching content from Hygraph CMS, and authenticating users via Clerk.
- The platform supports browser access from desktops, laptops, and smartphones, all requiring an active internet connection.

2 Processor

- Intel i3 or above (or equivalent): Ensures smooth execution of tasks like real-time face detection and recognition.
- Higher-end processors (e.g., Intel i5/i7 or AMD Ryzen) are recommended for faster processing.

3 Ram

- 8 GB or more: Sufficient memory to handle resource-intensive applications and multitasking.
- 16 GB is ideal for enhanced performance.

4 Storage

- 256 GB SSD: Provides fast read/write speeds for storing the operating system, libraries, and application data.
- Additional external storage may be used for storing large datasets or images.

CHAPTER-3
PROBLEM DESCRIPTION

3.1 PROBLEM DESCRIPTION

In today’s fast-paced, technology-driven world, the demand for continuous learning and upskilling has increased significantly. Students, professionals, and self-learners are constantly trying to keep up with evolving technologies, industry requirements, and job market trends. Despite the availability of countless online learning platforms and resources, most individuals still face significant challenges in managing their learning journey effectively. The core issue lies not in the lack of resources but in the absence of structured guidance, centralized tracking, and personalized learning pathways.

Additionally, many learners find it difficult to balance their learning with other responsibilities due to poor time management and the absence of progress reminders. Current learning platforms rarely offer dashboards that combine visual progress indicators, goal setting, task management, and self-assessment tools in one place. The fragmented nature of most online systems forces users to rely on external tools or manual tracking methods, which are inefficient and time-consuming.

Another problem arises from the one-size-fits-all model followed by most learning portals. These platforms often fail to consider the user's background, interests, or current proficiency level while suggesting content. As a result, learners either get overwhelmed with advanced material or waste time revisiting basic concepts they already know.

To solve these issues, there is a need for a dedicated platform that provides a structured, user-centered learning path management system. The “Learning Path Dashboard for Enhancing Skills System” aims to bridge this gap by offering a digital workspace where users can define learning goals, view skill maps, plan study schedules, and monitor their progress effectively. By providing clarity, structure, and motivation, the dashboard empowers users to take control of their skill-building journey in an organized and goal-oriented way.

CHAPTER-4
LITERATURE SURVEY

4.1 INTRODUCTION

The literature survey examines existing research and systems related to online learning platforms, progress tracking dashboards, and self-regulated learning tools. It identifies gaps in current solutions, particularly the lack of structured learning paths and personalized progress management. This review supports the foundation of the proposed system and guides its design and functional improvements.

4.2 RELEVANT RESEARCH PAPERS AND CASE STUDIES

4.2.1 RELEVANT RESEARCH PAPERS

1 "A Personalized E-Learning System Based on User Learning Styles" (Lu et al., 2018)

- **Summary**: This paper discusses the development of a recommendation system that adapts content based on individual learning styles using the Felder-Silverman model. It emphasizes how personalized content delivery improves learner engagement and efficiency.
- **Relevance**: Although your current system does not include a recommendation engine, this paper highlights future potential for integrating learning styles into personalized learning paths to enhance user outcomes.

2 "Dashboard Design for Learner Engagement and Motivation" (Verbert et al., 2014)

- **Summary:** This research explores how learning analytics dashboards impact learner motivation and engagement. It presents design principles that focus on visual feedback, goal setting, and user control over their learning progress.
- **Relevance:** Directly informs your dashboard's UI/UX design, especially for visual progress tracking and goal management features to enhance user motivation.

3 "Designing Effective Learning Pathways in Online Environments" (De Jong et al., 2016)

- **Summary:** The study discusses how structured learning pathways improve knowledge retention and help students achieve long-term goals. It introduces strategies for sequencing content and defining learning objectives.
- **Relevance:** Supports your system's core idea of structured, goal-oriented learning paths, validating its importance in self-guided learning environments.

4.2.2 CASE STUDIES

1 "Khan Academy Learner Dashboard" (Khan Academy, 2019)

- **Summary:** Khan Academy's dashboard allows students to view progress across subjects, earn mastery points, and follow teacher-assigned learning paths. It also includes gamification elements.
- **Relevance**: Validates the use of personalized paths, skill tracking, and engagement tools—several of which your system includes or plans to include in future updates.

2 "edX Progress Tracker System" (Case Study by edX, 2020)

- **Summary:** edX introduced a structured progress tracking tool to show learners what percentage of a course they had completed. It supported features like assignment deadlines and grading overviews.
- **Relevance:** Reinforces the importance of visual and statistical progress data in motivating learners and helping them manage their pace effectively core aspects of your system.

3 "Coursera Learning Dashboard" (Case Study by Coursera, 2021)

- **Summary:** Coursera implemented a learner dashboard to help users track course progress, deadlines, and certificates. The dashboard includes visual indicators and reminders to boost learner engagement.
- **Relevance:** Demonstrates how progress visibility and deadline tracking improve learner commitment an idea your dashboard also incorporates to help users stay focused.

4 "LinkedIn Learning Skill Tracker" (LinkedIn Learning, 2021)

- **Summary:** LinkedIn Learning offers a dashboard that maps user progress to job-relevant skills and career paths. Users can track achievements and plan their learning goals.
- **Relevance**: Offers insight into how career-aligned learning dashboards help users set clearer goals, a feature that aligns with your vision of structured skill enhancement.

5 "SkillUp by Simplilearn" (Simplilearn, 2022)

- **Summary:** Simplilearn's SkillUp platform provides free courses with a learning dashboard that offers skill paths, completion badges, and goal tracking to encourage continuous learning.
- **Relevance**: Highlights how a dashboard with goal-based learning paths can enhance user motivation and accountability, aligning directly with your project's purpose.

CHAPTER-5
SOFTWARE REQUIREMENT SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS

1 User Registration and Authentication

- Users must be able to register and log in securely.
- Clerk authentication system will be used.
- Session management ensures only authorized access.

2 Role-Based Access Control

- Two main roles: Admin and Student/User.
- Admins manage learning paths, content, and users.
- Student can only view and track their own progress.

3 Learning Path Creation and Management (Admin)

- Admin can create/edit/delete learning paths.
- Each path includes stages/modules with content.
- Hygraph CMS is used to manage content dynamically.

4 Student Dashboard

- Shows enrolled learning paths with progress bars.
- Lists completed modules and upcoming tasks.
- Displays overall skill development statistics.

5 Content Management via Hygraph CMS

- Admins upload content (videos, articles, links, etc.)
- Easy updates without modifying source code.
- Supports multimedia for engaging learning content.

6 Responsive and Interactive UI

- Built using ReactJS and NextJS.
- Optimized for mobile, tablet, and desktop views.
- Fast loading with smooth navigation.

5.2 NON-FUNCTIONAL REQUIREMENTS

1 Performance

- The system should load pages quickly.
- Smooth navigation between modules and dashboard sections is essential.
- Must support concurrent users without noticeable lag.

2 Scalability

- The system should handle an increasing number of users and learning paths over time.
- Backend and CMS (Hygraph) should be scalable to accommodate content growth.

3 Security

- All user data must be securely stored and transmitted using HTTPS.
- Authentication and authorization are managed through Clerk, ensuring robust session control.
- Admin functionalities should be protected from unauthorized access.

4 Usability

- The interface should be intuitive, requiring minimal training for new users.
- Navigation should be consistent and user-friendly across all devices.
- Visual indicators like progress bars and charts should simplify understanding of progress.

5 Availability

- The system should be accessible 24/7 with minimal downtime.
- Backup mechanisms should be in place for data recovery.

6 Compatibility

- Should be compatible with all major browsers (Chrome, Firefox, Safari, Edge).
- Supports all modern operating systems (Windows, macOS, Linux).
- Fully responsive on mobile, tablet, and desktop screens.

7 Maintainability

- The codebase should be modular and well-documented for future updates.
- CMS integration (Hygraph) allows for content updates without altering the code.
- Easy debugging and logging should be supported for quick issue resolution.

CHAPTER-6
SOFTWARE DESIGN

6.1 DATA FLOW DIAGRAM

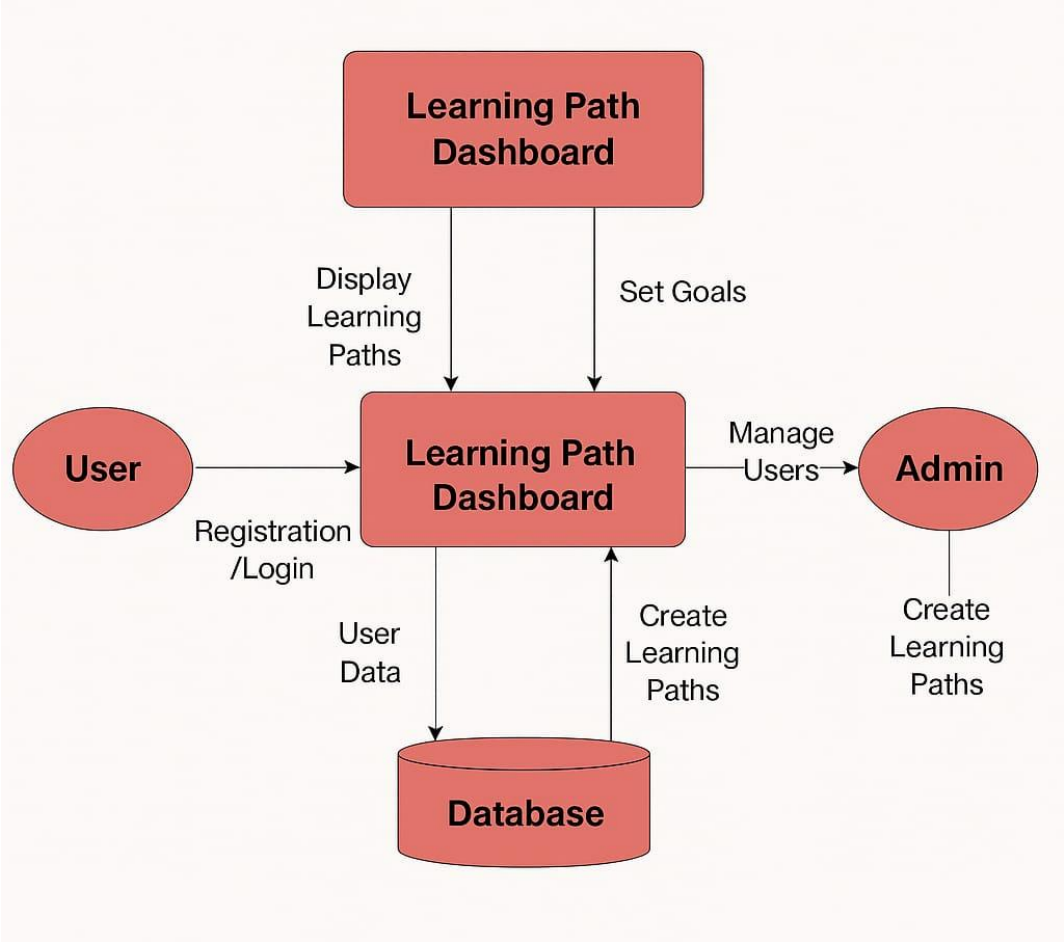

Figure 6.1: Data Flow Diagram

6.2 ER DIAGRAM

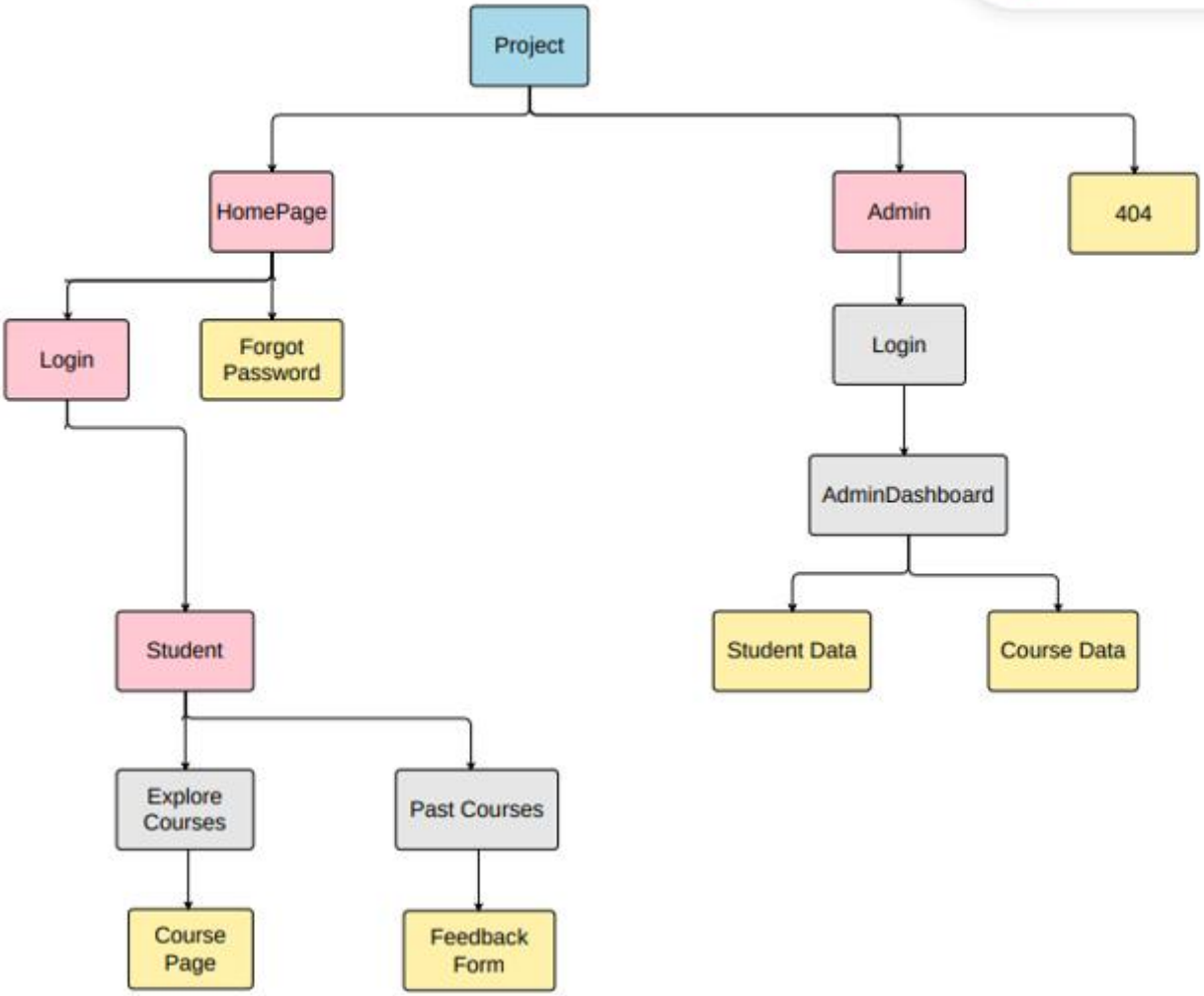

Figure 6.2: ER Diagram

CHAPTER-7
CODING PART

7.1 SOURCE CODE

```
##########################################HOMEPAGE#####################################
'use client';

import Link from 'next/link';
import { ReactTyped } from 'react-typed';
export default function HomePage() {
  return (
    <div className="relative min-h-screen w-
full overflow-hidden">
      <video
        className="absolute top-0 left-0 w-full
h-full object-cover z-0"
        autoPlay
        loop
        muted
        playsInline
      >
        <source src="/video.mp4"
type="video/mp4" />
      </video>

      <div className="relative z-10 flex flex-
col items-center justify-center min-h-screen
bg-black/50 text-center px-4 ">
        <h1 className="text-[60px] font-
extrabold text-gray-900"> Welcome to{' '}
          <ReactTyped
            strings={['Pathway', 'Your Mentor',
'Skill Builder']}
            typeSpeed={250}
            backSpeed={50}
```

```
  loop={true}
            className="text-blue-900 font-
  extrabold "
  />
  </h1>

        <h2 className="water-text text-xl text-
  center  text-gray-200 italic max-w-2xl">
          "Skill is not something you're born with
  — it's something you build.
          Every small step you take today is
  shaping the expert you'll become tomorrow."
        </h2>

        <Link href="/browse">
          <button className="mt-4 px-6 py-3
  bg-blue-700 text-white rounded-lg hover:bg-
  teal-700 transition">
            Start Exploring Courses
          </button>
        </Link>
      </div>
    </div>
   );
  }

import {
  ClerkProvider,
  SignInButton,
  SignedIn,
  SignedOut,
  UserButton
} from '@clerk/nextjs'
import { Geist, Geist_Mono } from "next/font/google";
import "./globals.css";
```

```
const geistSans = Geist({
  variable: "--font-geist-sans",
  subsets: ["latin"],
});

const geistMono = Geist_Mono({
  variable: "--font-geist-mono",
  subsets: ["latin"],
});

export const metadata = {
  title: "Create Next App",
  description: "Generated by create next app",
};

export default function RootLayout({ children }) {
  return (
    <ClerkProvider>
    <html lang="en">
      <body
        className={`${geistSans.variable} ${geistMono.variable} antialiased`}>{children}</body>
    </html>
    </ClerkProvider>
  )
}
```

##GLOBALA.CSS#####################################

```
@import "tailwindcss";

:root {

}

@theme inline {
```

```
 --color-background: var(--background);
  --color-foreground: var(--foreground);
  --font-sans: var(--font-geist-sans);
  --font-mono: var(--font-geist-mono);
}

@media (prefers-color-scheme: dark) {
  :root {

  }
}

body {
  background: var(--background);
  color: var(--foreground);
  font-family: Arial, Helvetica, sans-serif;
}

/* WATER FLOW ANIMATION TEXT */
/* .water-text {
  background: linear-gradient(
    120deg,
    #00c3ff,
    #1cff8e,
    #00c3ff,
    #1cffbb
  );
  background-size: 400% 400%;
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  animation: waterFlow 6s ease infinite;
}

@keyframes waterFlow {
  0% {
```

```
  background-position: 0% 50%;
 }
 50% {
  background-position: 100% 50%;
 }
 100% {
  background-position: 0% 50%;
 }
} */
```

##_WATCHCOURSE#####################################

```
"use client"
import React, { useEffect, useState } from 'react'
import ChapterNav from './_components/ChapterNav'
import FullVideoPlayer from './_components/FullVideoPlayer'
import { UserButton, useUser } from '@clerk/nextjs'
import { getCourseById } from '@/app/_services';
import { CompletedChapterContext } from '../../../../../app/_context/CompletedChapterContext'

function ViewCourse({params}) {
   const {user}=useUser();
   const [course,setCourse]=useState([]);
   const [userCourse,setUserCourse]=useState();
   const [activeChapter,setActiveChapter]=useState();
   const [completedChapter,setCompletedChapter]=useState();

   useEffect(()=>{
      user? getCourse():null;
   },[user])

   const getCourse=async()=>{
      await getCourseById(params?.courseId,
         user.primaryEmailAddress.emailAddress)
          .then(resp=>{
```

```
 console.log(resp?.userEnrollCourses[0]?.completedChapter);
  setCourse(resp.courseList);
              setUserCourse(resp.userEnrollCourses);
              setCompletedChapter(resp?.userEnrollCourses[0]?.completedChapter);
        })
    }
  return course?.name&& (
    <div className="">
        <CompletedChapterContext.Provider value={{completedChapter,setCompletedChapter}}>
        <div className='hidden fixed bg-white border-2 border-gray-200 md:block md:w-80 shadow-sm h-screen
z-50'>
       {course? <ChapterNav course={course}
            userCourse={userCourse}
            setActiveChapter={(chapter)=>setActiveChapter(chapter)}/>
            :null}

        </div>

        <div className='md:ml-80'>
            <div className='float-right p-5'>
            <UserButton/>
        </div>
          <FullVideoPlayer
          userCourse={userCourse}
          activeChapter={activeChapter}/>
        </div>

        </CompletedChapterContext.Provider>
    </div>
  )
}

export default ViewCourse
```

```
import { CheckCircle2 } from 'lucide-react';
import React, { useContext } from 'react'
import { CompletedChapterContext } from '../../../../../_context/CompletedChapterContext';
import { markChapterCompleted} from './../../../../../_services/index';

function FullVideoPlayer({userCourse, activeChapter }) {
  console.log(activeChapter);
  const {completedChapter,setCompletedChapter}=
  useContext(CompletedChapterContext)

  const isChapterCompleted=(chapterId)=>{
     return completedChapter.find(item=>item.chapterId==chapterId)
  }

  const markChapterCompleted=async()=>{
     if(!completedChapter?.length)
     {
       setCompletedChapter([]);
     }
     completedChapter?setCompletedChapter(
       [...completedChapter,
         {
           chapterId:activeChapter?.chapterNumber+""
         }]
     ):setCompletedChapter([ {
           chapterId:activeChapter?.chapterNumber+""
         }]);
         console.log(completedChapter);

     // await markChapterCompleted(userCourse?.id,activeChapter?.chapterNumber).then(resp=>{
     //   console.log(resp);
     //   })
  }

  return activeChapter&&(
```

```
 <div className='p-5'>
     <video width="1500" height="250"
     key={activeChapter?.video?.url}
       controls controlsList='nodownload'>
       <source src={activeChapter?.video?.url}
         type='video/mp4'
       />
     </video>
     <div className='p-5 border-2 border-gray-300 rounded-lg mt-5 flex justify-between items-center'>
       <h2 className='text-[20px] font-medium'>{activeChapter?.name}</h2>
       {!isChapterCompleted(activeChapter.chapterNumber)?
       <button className='bg-teal-500 text-white p-2 px-5 rounded-lg flex gap-2
       hover:bg-teal-700'
       onClick={()=>markChapterCompleted()}>
         <CheckCircle2/> <h2>Mark as Completed</h2>
       </button>:null}
       {/* {<button className=' text-teal-600
       border border-teal-600 p-2 px-5 rounded-lg flex gap-2
       hover:bg-teal-100'>
         <XCircle/> <h2>Mark InComplet</h2>
       </button>} */}
     </div>
   </div>
  )
}

export default FullVideoPlayer

import { CompletedChapterContext } from '@/app/_context/CompletedChapterContext';
import { CheckCircle2, PauseCircle, PlayCircle } from 'lucide-react'
import React, { useContext, useEffect, useState } from 'react'

function ChapterNav({ course, userCourse, setActiveChapter }) {
  const [activeIndex, setActiveIndex] = useState(0);
  const { completedChapter, setCompletedChapter } = useContext(CompletedChapterContext)
```

```
useEffect(() => {
   setActiveChapter(course?.chapter[0])

 }, [])

 const isChapterCompleted = (chapterId) => {
  return completedChapter.find(item => item.chapterId == chapterId)
 }
 return (
  <div>
   <div className='border border-gray-100 p-5'>
    <h2 className='font-medium text-[20px]'>{course.name}</h2>
    <h2 className='text-gray-500 text-[14px]'>By {course.author}</h2>
   </div>
   <div className='overflow-auto h-[600px]'>
    {course?.chapter?.map((chapter, index) => (
     <div key={index}
      onClick={() => {
       setActiveIndex(index);
       setActiveChapter(chapter)
      }}
      className={`flex gap-2 text-gray-500 md:text-[14px] text-[12px] px-5 p-4 cursor-pointer
      hover:bg-gray-100
      ${isChapterCompleted(chapter.chapterNumber) && activeIndex != index ? 'bg-teal-100 text-teal-600' :
null}
      ${activeIndex == index ? 'bg-teal-50 text-teal-600' : null}`}>
      {activeIndex == index
       ? <PauseCircle height={25} width={25} /> :
       isChapterCompleted(chapter.chapterNumber) ?
        <CheckCircle2 height={25} width={25} /> :
        <PlayCircle height={25} width={25} />}
      <h2 className='line-clamp-2'>{chapter.name}</h2>
     </div>
    ))}
```

```
  </div>
    </div>
   )
  }

  export default ChapterNav
```

##DASHBOARD##

```
import React from 'react'
import SideBarNav from './../_components/SideBarNav'
import Header from '../_components/Header'
function homeLayout({children}) {
 return (
  <div>
    <div className='h-full w-64 flex-col fixed inset-y-0 z-50
    ' >
       <SideBarNav/>
    </div>

    <Header/>
    <div className='ml-64 p-5'>
    {children}
    </div>

  </div>
 )
}

export default homeLayout
```

```
// "use client"
// import React, { useEffect, useState } from 'react'
// import { GetUserCourseList } from '../../../_services'
// import { useUser } from '@clerk/nextjs'
// import CategoryItem from '../../_components/CategoryItem';

// function Dashboard() {
//     const {user} = useUser();
//     const [userCourseList,setUserCourseList] = useState([]);

//     useEffect(()=>{
//         user?getUserCourse():null;
//     },[user])
//     const getUserCourse = async()=>{
//         await GetUserCourseList(user.primaryEmailAddress.emailAddress)
//         .then(resp=>{
//             console.log(resp?.userEnrollCourses)
//             if(resp)
//                 setUserCourseList(resp?.userEnrollCourses)
//         })
//     }
//   return (
//     <div>
//         {/* {!userCourseList?   <> */}
//         <h2 className='text-[20px] font-medium'>My Enrolled Courses:</h2>
//       <div className='grid grid-cols-1 sm:grid-cols-2
//         md:grid-cols-3 lg:grid-cols-4 mt-5 gap-5'>
//             {/* {userCourseList&&userCourseList.map((course)=>(
//                 <div>
//                     <CategoryItem course={course?.courseLists[0]}/>
//                 </div>
//             ))} 
//         </div>
//         </>:
//         <div className='flex justify-center items-center
```

```
 text-[20px] mt-20 text-gray-500'>
//            <h2>You don't have any course enrolled</h2>
//         </div>}
//     </div>
//   )
// } */}
//   {userCourseList && userCourseList.length > 0 ? (
//          userCourseList.map((course, index) => {
//            const courseData = course?.courseLists?.[0]
//            return courseData ? (
//              <div key={courseData?.id || index}>
//                <CategoryItem course={courseData} />
//              </div>
//            ) : null
//          })
//        ) : (
//          <p className='text-gray-500'>You haven't enrolled in any courses yet.</p>
//        )}
//      </div>
//    </div>
//   )
// }

// export default Dashboard
"use client"
import React, { useEffect, useState } from 'react'
import { GetUserCourseList } from '../../../_services'
import { useUser } from '@clerk/nextjs'
import CategoryItem from '../../_components/CategoryItem'

function Dashboard() {
 const { user } = useUser()
 const [userCourseList, setUserCourseList] = useState([])

 useEffect(() => {
```

```
if (user) {
    getUserCourse()
  }
 }, [user])

 const getUserCourse = async () => {
  try {
   const resp = await GetUserCourseList(user.primaryEmailAddress.emailAddress)
   console.log("API Response:", resp?.userEnrollCourses)
   if (resp?.userEnrollCourses) {
    setUserCourseList(resp.userEnrollCourses)
   }
  } catch (error) {
   console.error("Failed to fetch user courses:", error)
  }
 }

 return (
  <div>
   <h2 className='text-[20px] font-medium'>My Enrolled Courses:</h2>
   <div className='grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 mt-5 gap-5'>
    {userCourseList && userCourseList.length > 0 ? (
     userCourseList.map((course, index) => {
      const courseData = course?.courseLists?.[0]
      return courseData ? (
       <div key={courseData?.id || index}>
        <CategoryItem course={courseData} />
       </div>
      ) : null
     })
    ) : (
     <p className='text-gray-500'>You haven't enrolled in any courses yet.</p>
    )}
   </div>
  </div>
```

```
  )
}

export default Dashboard
```

##COURSE_PREVIEW###

```
"use client"
import { getCourseById } from '../../../../_services'
import React, { useEffect, useState } from 'react'
import VideoPlayer from './_components/VideoPlayer';
import CourseDetails from
'./_components/CourseDetails';
import OptionSection from
'./_components/OptionSection';
import EnrollmentSection from
'./_components/EnrollmentSection';
import { useUser } from '@clerk/nextjs';

function CoursePreview({params}) {
  const [courseDetail,setCourseDetails]=useState([]);
  const [userCourse,setUserCourse]=useState([]);

  const {user}=useUser();
  useEffect(()=>{
    params.courseId?getCourse(params.courseId):null;
  },[user])

  const getCourse=()=>{
    getCourseById(params.courseId,user?.primaryEmail
Address?.emailAddress).then(resp=>{
      console.log(resp);
      setCourseDetails(resp.courseList);
      setUserCourse(resp.userEnrollCourses[0]);
      setCompletedChapter(resp?.userEnrollCourses[0]?.c
```

```
ompletedChapter);
    })
  }
  return courseDetail?.name&&(
    <div className="">
      <div className='grid grid-cols-1 md:grid-cols-3
gap-5'>
        <div className='col-span-2'>
          {courseDetail?.chapter[0]?<VideoPlayer
          videoUrl={courseDetail?.chapter[0]?.video.url}/>
:null}

          <CourseDetails courseDetail={courseDetail}/>

        </div>
        <div className=' mt-5 md:mt-0'>
          <OptionSection/>
          <EnrollmentSection courseDetail={courseDetail}
          userCourse={userCourse}/>
        </div>
      </div>
    </div>
  )
}

export default CoursePreview
```

##BROWSE###

```
"use client"
import React, { useEffect, useState } from 'react'
import CategoryFilter from '../browse/_components/CategoryFilter'
import { getCourseList } from './../../../_services/index'
import CourseList from './_components/CourseList';
function browse() {

  const[courses,setCourses]=useState([]);
  const[coursesOrg,setCoursesOrg]=useState([]);
  useEffect(()=>{
    getCourses()
  },[])
  const getCourses=()=>{
    getCourseList().then(resp=>{
      console.log(resp)
      setCourses(resp.courseLists);
      setCoursesOrg(resp.courseLists)
    })
  }
  const filterCourse=(category)=>{
    if(category=='all')
    {
      setCourses(coursesOrg);
      return;
    }

    const filteredList=coursesOrg.filter(course=>{
      return course.tag.includes(category);
    })
    setCourses(filteredList);
  }
  return (
    <div>
      <CategoryFilter selectedCategory={(category)=>filterCourse(category)}/>
```

```
    {courses? <CourseList courses={courses}/>:null}
      </div>
    )
  }

  export default browse
```

```
import Image from 'next/image'
import React from 'react'
import { Book, Youtube, YoutubeIcon} from 'lucide-react'
import Link from 'next/link'
import CategoryItem from './../../../_components/CategoryItem'

function CourseList({courses}) {
  return (

    <div className='mt-5 grid grid-cols-1
    sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-5'>
     {courses.map((course,index)=>(
       <Link href={'/course-preview/'+course.id} key={index}>
      <CategoryItem course={course}/>
       </Link>
     ))}
   </div>
  )
}

export default CourseList
```

##CATEGORYFILTER#####################################

```
"use client"
import React, { useState } from 'react'

function CategoryFilter({selectedCategory}) {
   const [activeIndex,setActiveIndex]=useState(0)
   const filterOptions=[
      {
         id:1,
         name:'All',
         value:'all',
      },
      {
         id:2,
         name:'React Js',
         value:'reactjs',
      },
      {
         id:3,
         name:'Next Js',
         value:'nextjs',
      },
      {
         id:4,
         name:'Tailwind Css',
         value:'tailwindcss',
      },
      {
         id:5,
         name:'Firebase',
         value:'firebase'
      }
   ]
 return (
```

```
    <div className='flex gap-5'>
        {filterOptions.map((item,index)=>(

    <button key={index}
        onClick={()=>
            {setActiveIndex(index);
                selectedCategory(item.value)
            }}
        className={`border-2 border-gray-200 p-2 px-4 text-sm rounded-md
        hover:border-teal-600 font-semibold
        hover:bg-gray-100
         ${activeIndex==index?'border-teal-600 bg-teal-50 text-teal-600':null}`}>
            <h2>{item.name}</h2>
        </button>

       ))}
      </div>
   )
  }

  export default CategoryFilter

"use client"
import React, { useState } from 'react'
import Image from 'next/image'
import { Layout, Mail, Search, Shield } from 'lucide-react'
import { useRouter } from 'next/navigation'

function SideBarNav() {
    const router = useRouter();
    const menuList = [
```

```
 {
      id: 1,
      name: 'Browse',

     icon: Search,
      path: '/browse'
   },
   {
      id: 2,
      name: 'Dashboard',
      icon: Layout,
      path: '/dashboard'
   },
   {
      id: 3,
      name: 'Upgrade',
      icon: Shield,
      path: '/upgrade'
   },
   {
      id: 4,
      name: 'Newsletter',
      icon: Mail,
      path: '/newsletter'
   },
]

const [activeIndex, setActiveIndex] = useState(0);

return (
   <div className='h-full bg-white border-b border-gray-200 flex flex-col overflow-y-auto shadow-md'>
      <div className='p-5 border-0 z-50'>
         <Image src='/logo.jpg' alt='logo' width={170} height={100} />
      </div>
      <div className='flex flex-col'>
```

```
            {menuList.map((item, index) => (

        <div

  key={item.id}
                    className={`flex gap-2 items-center p-4 px-6 text-gray-500
                    hover:border-emerald-400 font-semibold

 hover:bg-gray-100 cursor-pointer
                    ${activeIndex === index ? 'border-teal-600 bg-teal-50 text-teal-600' : ''}`}
                    onClick={() => {
                       setActiveIndex(index);
                       router.push(item.path); // <--- navigation added
                    }}
                  >
                    <item.icon />
                    <h2>{item.name}</h2>
                  </div>
              ))}
          </div>
      </div>
   )
}

export default SideBarNav

import { Search } from 'lucide-react'
import React from 'react'

function SearchBar(onSearch) {
  return (
    <div className='flex gap-3 text-[14px]
    iteam-center border-1 border-gray-200 p-2 rounded-md bg-gray-50 text-gray-500'>
```

```
  <Search height={17}/>
      <input type="text"
      placeholder='Search Course'
      className='bg-transparent outline-none'/>
   </div>
  )
}

export default SearchBar
```

```
"use client"
import React, { useEffect } from 'react'
import SearchBar from './../_components/SearchBar'
import { UserButton, useUser } from '@clerk/nextjs'
import { useRouter } from 'next/navigation';

function Header() {
   const {user}=useUser();
   const router = useRouter();
   useEffect(()=>{
      console.log(user)
   },[user])

  return (
   <div className='ml-64 p-6 border-0 border-gray-50 shadow-sm flex item-center justify-between'>
     <SearchBar/>

     {!user?
     <button onClick={()=>router.push('/sign-in')} className='border-2 border-teal-200
     bg-teal-100 text-teal-700 pl-4 pr-4 text-[18px] rounded-2xl font-semibold
     hover:bg-gray-200 hover:border-gray-200'>Login</button>
     :
     <UserButton/>}
   </div>
  )
```

```
}

export default Header



"use client"
import React from 'react';
import Image from 'next/image';
import { Book, YoutubeIcon } from 'lucide-react';

function CategoryItem({ course }) {
  return (
    <div>
    <div className='border-2 border-gray-200 rounded-lg p-2 shadow-md cursor-pointer hover:border-teal-200
bg-emerald-100'>
      {course?.banner?.url && (
        <Image
          src={course.banner.url}
          alt={course.name}
          width={1000}
          height={500}
          className='rounded-lg'
        />
      )}
      <div className='mt-2'>
        <h2 className='text-[17px] md:text-[15px] font-medium'>{course.name}</h2>
        <h2 className='text-gray-400 text-[12px]'>{course.author}</h2>

        {course.totalChapters && (
          <div className='flex items-center gap-2 mt-2'>
            <Book className='h-6 w-6 text-teal-600 rounded-full bg-teal-100 p-1' />
            <h2 className='text-[12px] text-gray-400'>{course.totalChapters} Chapters</h2>
          </div>
        )}
```

```
    {course.youtubeUrl && (
        <div className='flex items-center gap-2 mt-2'>
          <YoutubeIcon className='h-6 w-6 text-red-600 rounded-full bg-red-100 p-1' />
          <h2 className='text-[12px] text-gray-400'>Available on YouTube</h2>
        </div>
       )}

       <h2 className='mt-2 text-gray-700 text-[13px]'>{course.free ? 'Free' : 'Paid'}</h2>
     </div>
    </div>
    </div>
  );
}

export default CategoryItem;
```

##SERVICE_PAGE#####################################

```
import request, { gql } from "graphql-request"
const MASTER_URL="https://ap-south-
1.cdn.hygraph.com/content/"+process.env.NEXT_PUBLIC_HYGRAPH_KEY+"/master"

export const getCourseList=async()=>{
    const query=gql`
    query courseList {
  courseLists {
    name
    banner {
      url

    }
    free
    id
    author
```

```
  totalChapters
   tag
  }
}
  `
  const result=await request(MASTER_URL,query);
  return result;
}
export const getCourseById=async(id,userEmail)=>{
  const query=gql`
  query course {
 courseList(where: {id: "`+id+`"}) {
  chapter {
   ... on Chapter {
    id
    name
    chapterNumber
    video {
     url
    }

    youtubeUrl

   }
  }
  description
  name
  id
  free
  author
  totalChapters


 }
```

```
  userEnrollCourses(where: {courseId: "`+id+`",
    userEmail: "`+userEmail+`"}) {
    courseId
    userEmail
    completedChapter {
      ... on CompletedChapter {
        chapterId
      }
    }
      id
  }
}
    `
    const result=await request(MASTER_URL,query);
    return result;
}

export const EnrollCourse=async(courseId,userEmail)=>{
  const mutationQuery=gql`
  mutation EnrollCourse {
   createUserEnrollCourse(data: {userEmail: "`+userEmail+`",
    courseId: "`+courseId+`"}) {
    id
   }
}
  `
  const result=await request(MASTER_URL,mutationQuery);
    return result;
}

export const PublishCourse=async(id)=>{
  const mutationQuery=gql`
```

```
 mutation EnrollCourse {
  publishUserEnrollCourse(where: {id: "`+id+`"}) {
    id
  }
}
  `
  const result=await request(MASTER_URL,mutationQuery);
    return result;
}

export const markChapterCompleted=async(recordId,chapterNumber)=>{
  const mutationQuery=gql`
  mutation MarkChapterComplete {
  updateUserEnrollCourse(
    where: {id: "`+recordId+`"}
    data: {completedChapter: {create:
    {CompletedChapter: {data: {chapterId: "`+chapterNumber+`"}}}}}
  ) {
    id
  }
  publishManyUserEnrollCoursesConnection(to: PUBLISHED) {
    edges {
      node {
        id
      }
    }
  }
}
  `
  const result=await request(MASTER_URL,mutationQuery);
    return result;
}
export const GetUserCourseList=async(userEmail)=>{
```

```
  const query=gql`
  query UserCourseList {
  userEnrollCourses(where: {userEmail: "`+userEmail+`"}) {
    courseLists {
      banner {
        url
      }
      description
      id
      name
      free
      sourceCode
      tag
      totalChapters
      author
    }
  }
}`

const result=await request(MASTER_URL,query);
    return result;
}
```

CHAPTER-8
RESULT AND OUTPUT SCREENS

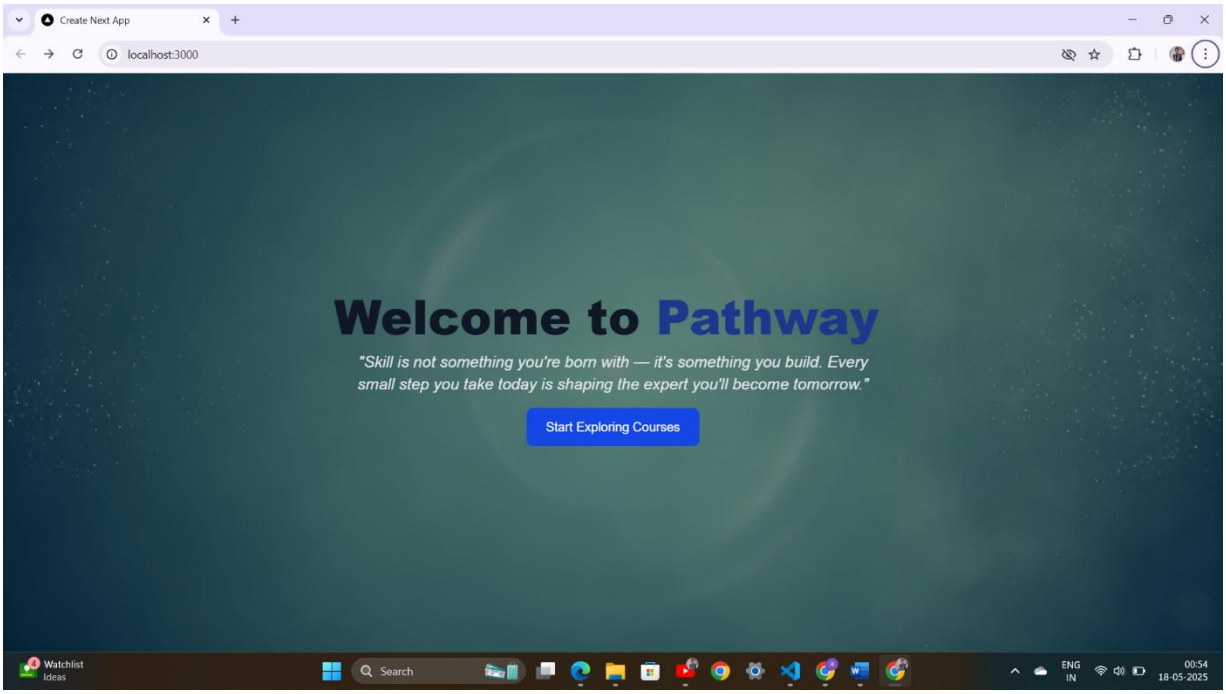

Figure 8.1: Home Page

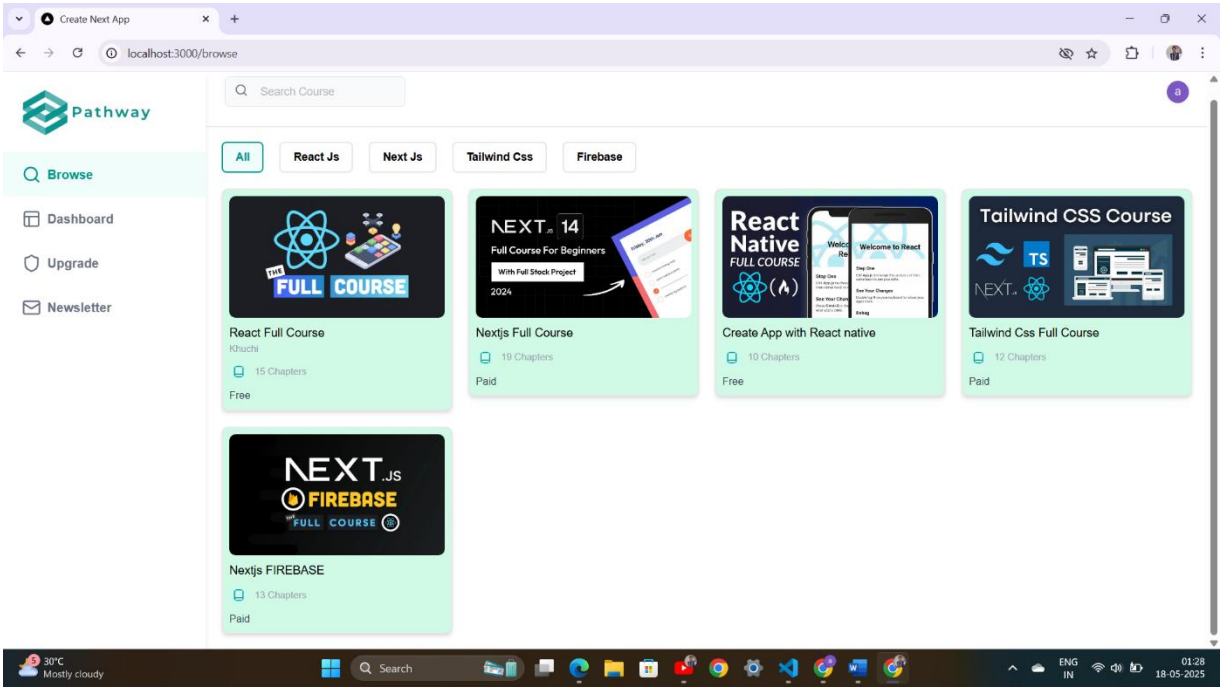

Figure 8.2: Browse Page or Course Section

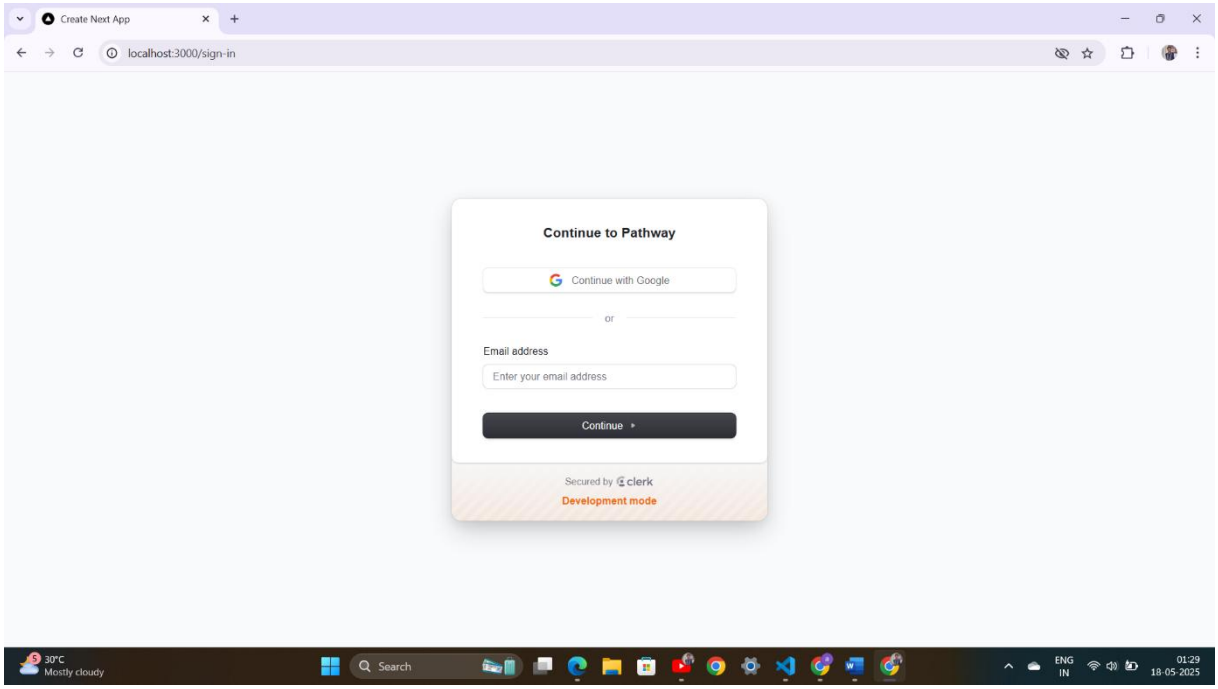

Figure 8.3: login page

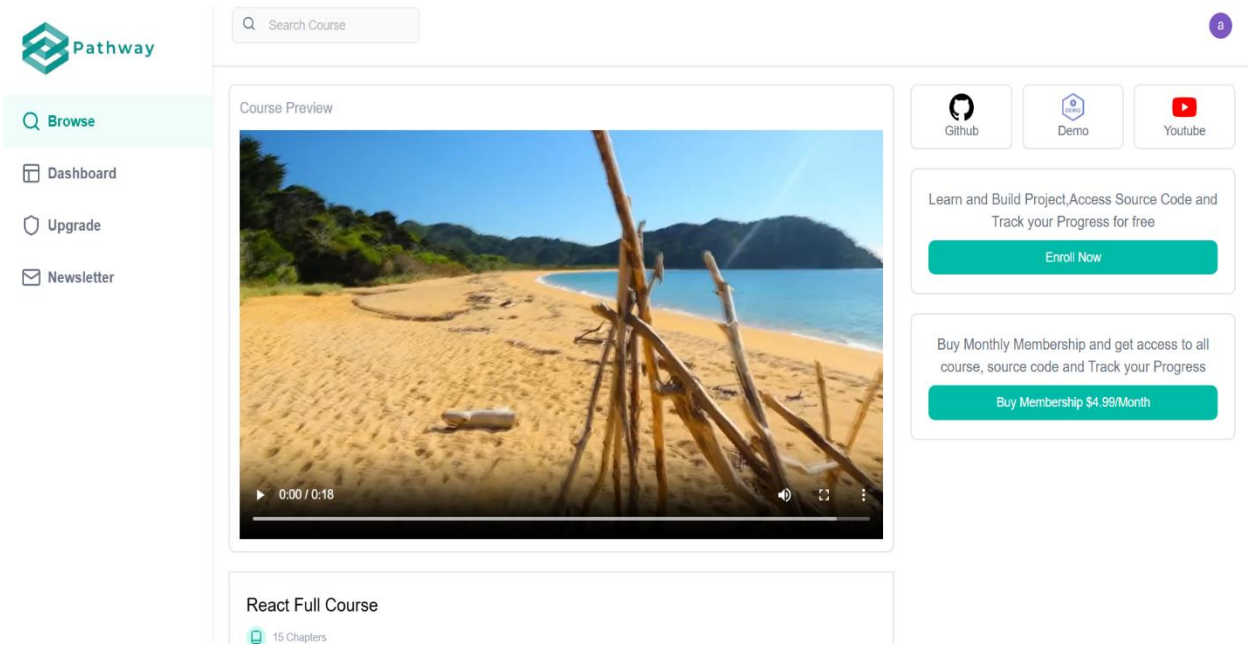

Figure 8.4: For Course Enrollment

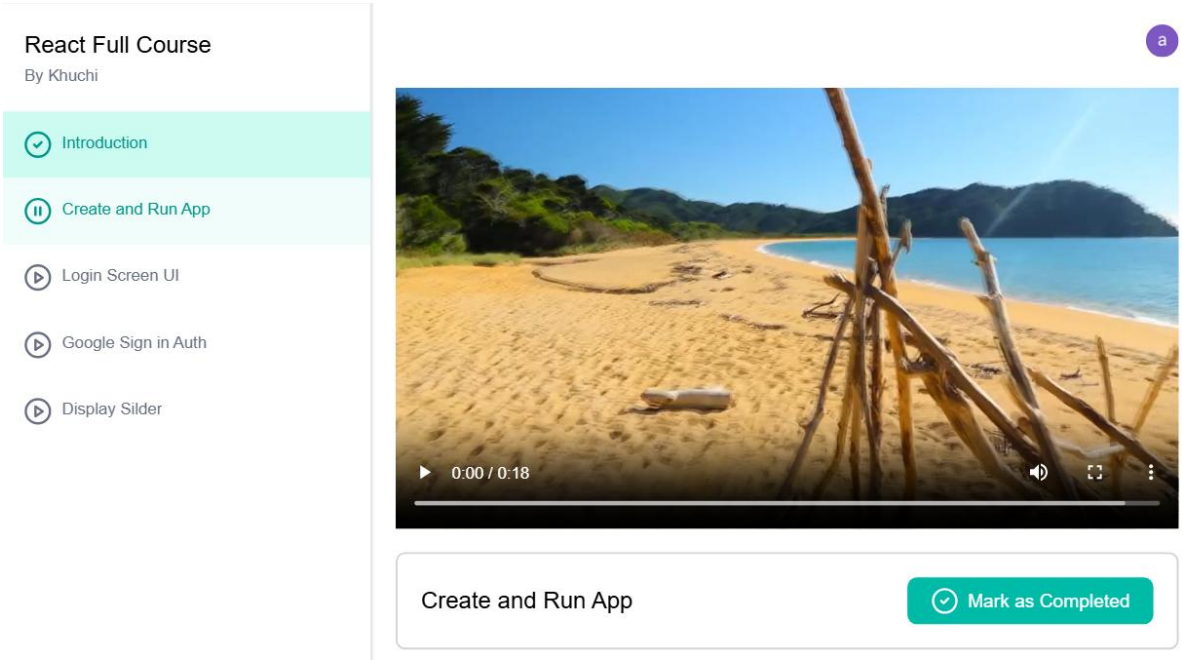

Figure 8.5: Course Chapter

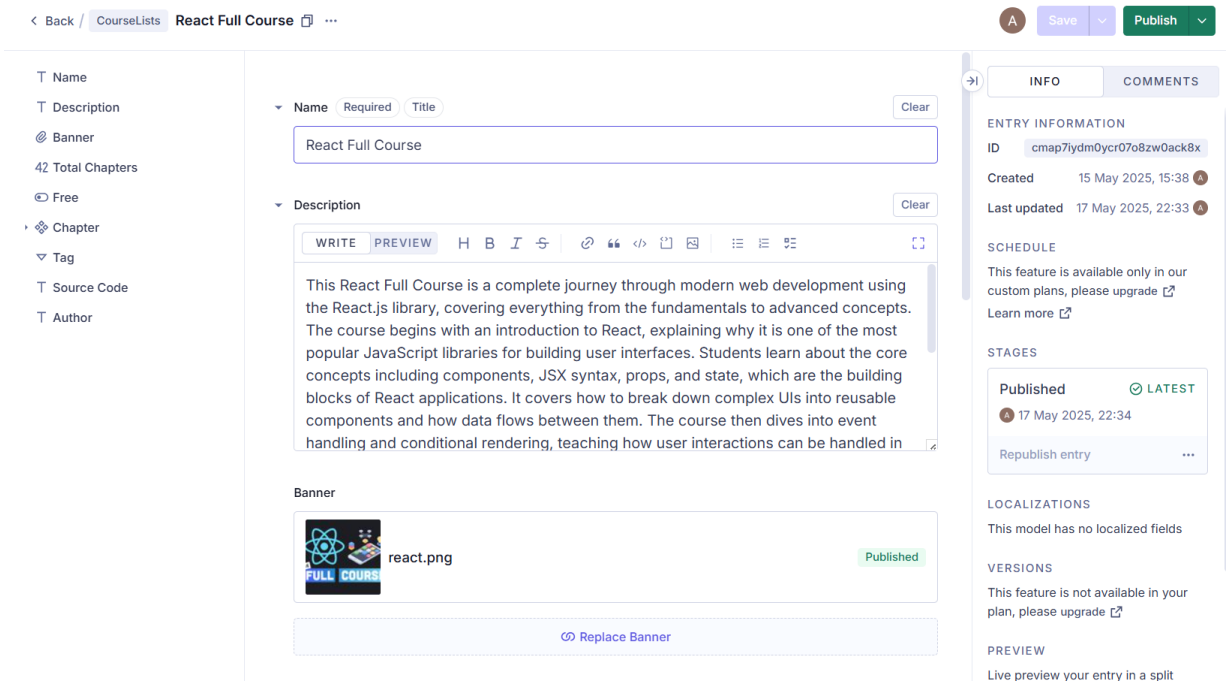

Figure 8.6: Adding New Courses

CHAPTER-9
CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

The Learning Path Dashboard for Enhancing Skills System offers a modern, structured, and user-friendly solution to the growing need for personalized skill development platforms. By utilizing technologies like React.js, Next.js, Node.js, Hygraph, and Clerk, the system delivers seamless navigation, secure authentication, and dynamic content management. It empowers learners to monitor their progress and stay on track through interactive dashboards and organized modules. The modular and scalable nature of the project ensures that it can evolve with technological trends and user expectations. The system has the potential to significantly impact self-paced learning and upskilling efforts, especially for students and professionals. This project not only demonstrates technical competence but also addresses a real-world need in a meaningful way. With a strong foundation in place, it can be extended with intelligent features and broader integrations to reach a wider audience and enhance user engagement in the future.

9.2 FUTURE WORK:

While the Learning Path Dashboard has demonstrated solid functionality, several avenues can be explored to enhance its effectiveness and scalability:

1 Personalized Recommendation System

- A machine learning-based feature can recommend courses and learning paths tailored to individual user goals, history, and performance, helping them stay focused and accelerate their skill acquisition effectively.

2 Real-Time Collaboration Tools

- Adding features like shared learning paths, live chat, and peer progress tracking can boost engagement and simulate a classroom environment, fostering interaction and motivation among learners. Enable group learning features like discussion rooms, shared learning goals, and real-time progress sharing.

3 Gamification and Achievement Badges

- Introducing rewards such as badges, points, or streaks can make learning more enjoyable, motivating users to maintain consistency and complete their learning goals with higher interest.

4 Mobile Application Development

- Creating a dedicated mobile app would enhance accessibility, enabling users to continue learning anytime and anywhere, especially for those with limited access to computers or fixed locations. develop to improve accessibility and allow users to learn on the go. Ensure real-time syncing with the web app for continuity.

5 Admin and Mentor Panel

- Building an interface for mentors to assign paths, track learner performance, and provide feedback will encourage guided learning and structured progress, especially in institutional or group settings. Add features for mentors or instructors to track group progress, assign tasks, and provide feedback.

6 Offline Learning Capabilities

- Implementing offline access to selected resources with sync functionality ensures learners in low-connectivity areas can continue progressing without interruptions and have a smooth experience. Allow users to download materials and track learning offline, with automatic sync once online.

7 AI-Powered Progress Forecasting

- Using AI to predict user progress and provide timely tips or interventions can help learners avoid stagnation, identify weak areas early, and stay on track to meet goals. Offer users insights on their performance and suggest ways to improve, boosting retention and motivation.

8 Data Privacy and Compliance

- Enhancing the system with secure data practices, encryption, and clear user consent mechanisms ensures compliance with regulations like GDPR and builds user trust in handling personal data. Implement data encryption, secure storage, and transparent user consent mechanisms.

REFERENCES

REFERENCES WEBSITES

- **React.js** Official Documentation (https://reactjs.org/docs/getting-started.html)
- **Next.js** Official Documentation (https://nextjs.org/docs)
- **Node.js** Official Documentation (https://nodejs.org/en/docs)
- **Hygraph** (Headless CMS) (https://hygraph.com)
- **Clerk** Authentication Platform (https://clerk.dev)
- **GeeksforGeeks** (Programming Concepts) (https://www.geeksforgeeks.org)
- **FreeCodeCamp** (Learning Resources & Tutorials) (https://www.freecodecamp.org)

REFERENCES BOOKS

- **William Horton**

  Title: Computer Vision: E-Learning by Design

  Publisher: Wiley

  Description: Covers instructional design principles and best practices in e-learning systems.

- **Alex Banks & Eve Porcello**

  Title: Learning React

  Publisher: O'Reilly Media

  Description: A practical guide for building applications using React.js. which used in project.

- **Kirill Konshin**

  Title: Next.js Quick Start Guide

  Publisher: Packt Publishing

  Description: Provides insights into building and production-ready apps with Next.js.

PROJECT SUMMARY

About Project

Title of the project	Learning Path Dashboard for Enhancing Skills System
Semester	6th
Members	4
Team Leader	Akash Prajapati
Describe role of every member in the project	**Aakash Tiwari : (UI/UX Designer)** Conducted usability testing and gathered user feedback. Ensured the design was intuitive and aligned with user needs. **Akash Prajapati : (Full-stack Developer)** Oversees the project, assigns tasks, and ensures deadlines and Designed and implemented the user interface using React.js ,Next.js and backend using Hygraph. **Adnan Baig : (Back-End Developer)** Integrated Hygraph CMS for content management and Clerk for authentication. Managed database interactions and API security. **Ashish Mehra : (Documentation & Deployment)** Prepared project documentation including reports, manuals, and presentations.
What is the motivation for selecting this project?	The motivation behind developing the Learning Path Dashboard for Enhancing Skills System stems from the growing need for personalized and structured skill development in today's fast-evolving job market. Many learners struggle to find clear guidance on what skills to acquire and how to track their progress effectively. This project aims to bridge that gap by providing a centralized platform that helps users plan, monitor, and enhance their learning journey efficiently. Additionally, integrating modern web technologies and user-friendly design makes skill-building accessible and engaging.
Project Type (Desktop Application, Web Application, Mobile App, Web)	Web Application

Tools & Technologies

Programming language used	JavaScript (React.js, Next.js, Node.js)
IDE used (with version)	VS Code(1.100.2),ect.
Front End Technologies (with version, wherever Applicable)	React.js, Next.js, Clerk, Flaticon, Hygraph Client SDK
Back End Technologies (with version, wherever applicable)	Node.js, Next.js API Routes, Hygraph (formerly GraphCMS), Clerk
For Real time	Next.js, SSE

Software Design& Coding

Is prototype of the software developed?	Yes
SDLC model followed (Waterfall, Agile, Spiral etc.)	Agile Model
Why above SDLC model is followed?	The Agile model was chosen because it supports flexible and incremental development, which is ideal for a dynamic project like the Learning Path Dashboard. Agile allows the team to quickly adapt to changing requirements, incorporate user feedback continuously, and deliver functional modules in short iterations. This improves collaboration, reduces risks, and ensures the project evolves based on real user needs. Additionally, Agile promotes frequent testing and refinement, which enhances product quality and user satisfaction.
Justify that the SDLC model mentioned above is followed in the project.	The Learning Path Dashboard for Enhancing Skills System follows the Agile SDLC model, which is well-suited for modern web application development. Agile allows for iterative development, enabling continuous feedback and improvements throughout the project lifecycle. This approach was beneficial as it allowed the team to rapidly prototype features like user authentication (Clerk), content management (Hygraph), and front-end components (ReactJS), then refine them based on testing and user feedback.

Software Design approach followed (Functional or Object Oriented)	Object Oriented
Name the diagrams developed (according to the Design approach followed)	Data Flow diagram, ER diagram
In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	The project's design incorporates key Object-Oriented Programming (OOP) principles to enhance code modularity and maintainability. Encapsulation is used by grouping data and related functions within React components, hiding internal states from other parts of the application. Abstraction simplifies complex processes like authentication and API communication, providing clean interfaces for users. Where applicable, inheritance allows sharing of common features between components, and polymorphism enables components to adapt behavior based on input, making the system flexible and reusable.
No. of Tiers (example 3-tier)	3-tier
Total no. of front end pages	7 to 8
Front end validations applied (Yes / No)	Yes
Session management done (in case of web applications)	Yes
Is application browser compatible (in case of web applications)	Yes
Exception handling done (Yes / No)	No

Commenting done in code (Yes / No)	Yes
Naming convention followed (Yes / No)	Yes
What difficulties faced during deployment of project?	Difficulties during deployment often include configuration issues and API routing , Hygraph.
Total no. of Use-cases	2
Give titles of Use-cases	ER Diagram, Data Flow Diagram

Project Requirements

MVC architecture followed (Yes / No)	Yes
If yes, write the name of MVC architecture followed (MVC-1, MVC-2)	MVC-2 (React/Next.js , API routes/controllers , Hygraph/Backend APIs)
Design Pattern used (Yes / No)	Yes
If yes, write the name of Design Pattern used	ReactJS, Next.js, Clear, Hygraph CMS
Interface type (CLI / GUI)	GUI
No. of Actors	2
Name of Actors	Admin, Student
Total no. of Functional Requirements	6
List few important non-Functional Requirements	Performance, Scalability, Usability, Maintainability, Security.

Testing

Which testing is performed? (Manual or Automation)	Manual
Is Beta testing done for this project?	No

Write project narrative covering above mentioned points

The "Learning Path Dashboard for Enhancing Skills System" is designed to provide learners with a structured and personalized environment to improve their skills over time. With the rise of self-paced learning and digital education platforms, learners often face difficulty in organizing their journey, tracking progress, and staying motivated. This system aims to address these challenges by offering a visually engaging and goal-driven dashboard.

The project includes user registration and authentication powered by Clerk, ensuring secure and role-based access. Admins can create and manage learning paths using Hygraph CMS, which allows for dynamic content management without altering the codebase. Students can enroll in available learning paths and track their module-wise progress in real-time through a well-structured dashboard developed using ReactJS, Next.js, and Flaticon for visual enhancement.

Each learning path is made up of stages or modules which can include text, links, videos, or tasks. The dashboard visually represents progress through percentage bars and analytics charts, helping students see how far they've come and what remains. Features like goal setting and reminders are built in to enhance user engagement and promote consistent learning behavior.

Functionally, the system includes modules for content delivery, progress tracking, skill management, and dashboard analytics. Notifications and reminders help keep users active, while reports provide feedback on completed skills and learning trends. On the non-functional side, the system ensures performance, security, usability, scalability, and compatibility across devices and browsers. It is also designed with future expansion in mind, including the potential addition of a recommendation system for suggesting personalized content paths.

Akash Prajapati	0187CS221027
Adnan Baig	0187CS221022
Aakash Tiwari	0187CS221001
Ashish Mehra	0187CS221061

Guide Signature

(Dr. Komal Tahiliani)

APPENDIX-1
GLOSSARY OF TERMS

A	
Authentication	The process of verifying a user's identity before granting access to the system, typically using tools like Clerk.
C	
Clerk	A user authentication and session management platform integrated to secure login/signup functionalities in web apps.
CMS	A platform (like Hygraph) used to manage, create, and organize digital content dynamically without hard-coding.
D	
Dashboard	An interface that provides users with visual feedback on their learning progress and personalized skill tracking.
F	
Frontend	The part of the application users interact with directly, built using technologies like React.js and Next.js.
H	
Hygraph	A headless CMS that allows developers to manage content using GraphQL APIs and integrate it into web platforms dynamically.
L	
Learning Path	A structured sequence of modules or activities designed to guide learners through specific skill development journeys.