

# **ONLINE LEARNING PLATFORM USING MERN STACK**

**SMART INTERNZ (NAAN MUDHALVAN)**

## **PROJECT REPORT**

Submitted by

<b>BARATH S</b>	<b>211121104008</b>
<b>RAVIVARMA V</b>	<b>211121104307</b>
<b>JAYARATHINAM JL</b>	<b>211121104022</b>
<b>VELUSAMY M</b>	<b>211121104057</b>
<b>BALAJI D</b>	<b>211121104303</b>

**STUDENTS OF  
COMPUTER SCIENCE & ENGINEERING DEPARTMENT**



**MADHA ENGINEERING COLLEGE  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
KUNDRATHUR, CHENNAI-600069.**

# Table of Contents

<b>S.No.</b>	<b>Topic</b>	<b>Page Number</b>
<b>I</b>	<b>Table of Contents</b>	<b>-</b>
<b>II</b>	<b>List of Figures</b>	<b>-</b>
<b>III</b>	<b>List of Tables</b>	<b>-</b>
<b>IV</b>	<b>Abbreviations</b>	<b>-</b>
<b>V</b>	<b>Abstract</b>	<b>-</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
	1.1 The Evolution of E-Learning Platforms	<b>1</b>
	1.2 The MERN Stack: A Framework for Modern Web Development	<b>1</b>
	1.3 Key Considerations in Enterprise E-Learning Platform Development	<b>2</b>
	1.4 Objectives of the Report	<b>2</b>
<b>2</b>	<b>Description</b>	<b>3</b>
	2.1 Architecture	<b>2</b>
	2.2 Design Principles	<b>2</b>
	2.3 Key Functionalities	<b>2</b>
	2.4 Integration & Scalability	<b>4</b>
<b>3</b>	<b>SRS Document</b>	<b>5</b>
<b>4</b>	<b>Data Flow Diagrams</b>	<b>31</b>
<b>5</b>	<b>Data Dictionary</b>	<b>35</b>

<b>6</b>	<b>Use Case Diagram</b>	<b>37</b>
<b>7</b>	<b>ER Diagram</b>	<b>41</b>
<b>8</b>	<b>Class Diagram</b>	<b>44</b>
<b>9</b>	<b>Activity Diagram</b>	<b>47</b>
<b>10</b>	<b>State Chart Diagram</b>	<b>49</b>
<b>11</b>	<b>Collaborative Diagram</b>	<b>51</b>
<b>12</b>	<b>Execution Screenshots</b>	<b>55</b>
<b>13</b>	<b>Technologies Used</b>	<b>60</b>
<b>14</b>	<b>Description about Front-end and Back-end</b>	<b>61</b>
<b>15</b>	<b>Testing</b>	<b>63</b>
<b>16</b>	<b>Concluding Remarks and Limitations</b>	<b>73</b>
<b>17</b>	<b>Future Work</b>	<b>74</b>
	<b>References</b>	<b>75</b>

## **List of Figures**

<b>Figure 1</b>	<b>Level 0 Data Flow Diagram</b>	<b>31</b>
<b>Figure 2</b>	<b>Level 1 Data Flow Diagram</b>	<b>32</b>
<b>Figure 3</b>	<b>Level 2 Data Flow Diagram</b>	<b>33</b>
<b>Figure 4</b>	<b>Use Case Diagram</b>	<b>37</b>
<b>Figure 5</b>	<b>ER Diagram</b>	<b>41</b>
<b>Figure 6</b>	<b>Class Diagram</b>	<b>44</b>
<b>Figure 7</b>	<b>Activity Diagram</b>	<b>47</b>
<b>Figure 8</b>	<b>State Chart Diagram</b>	<b>49</b>
<b>Figure 9</b>	<b>Collaborative Diagram for Admin</b>	<b>51</b>
<b>Figure 10</b>	<b>Collaborative Diagram For User</b>	<b>53</b>
<b>Figure 11</b>	<b>Home Page</b>	<b>55</b>
<b>Figure 12</b>	<b>Sign Up Page</b>	<b>55</b>
<b>Figure 13</b>	<b>Login Page</b>	<b>56</b>
<b>Figure 14</b>	<b>User Dashboard</b>	<b>56</b>
<b>Figure 15</b>	<b>User Shown to Admin</b>	<b>57</b>
<b>Figure 16</b>	<b>All Courses Shown to Admin</b>	<b>57</b>
<b>Figure 17</b>	<b>Courses Overview</b>	<b>58</b>
<b>Figure 18</b>	<b>All Courses Available</b>	<b>58</b>
<b>Figure 19</b>	<b>User Enrolled Courses</b>	<b>59</b>
<b>Figure 20</b>	<b>Path Coverage Testing for Sign Up</b>	<b>67</b>
<b>Figure 21</b>	<b>Path Coverage Testing for Login Frontend</b>	<b>68</b>

<b>Figure 22</b>	<b>Path Coverage Testing for Add User Frontend</b>	<b>69</b>
<b>Figure 23</b>	<b>Path Coverage Testing for Login Backend</b>	<b>70</b>
<b>Figure 24</b>	<b>Path Coverage Testing for Add User Backend</b>	<b>71</b>
<b>Figure 25</b>	<b>Path Coverage Testing for Add Courses</b>	<b>72</b>

## List of Tables

<b>Table No.</b>	<b>Description</b>	<b>Page Number</b>
<b>Table 1</b>	<b>Definitions, Acronyms, Abbreviations</b>	<b>6</b>
<b>Table 2</b>	<b>Data Dictionary of Users Table</b>	<b>35</b>
<b>Table 3</b>	<b>Data Dictionary of Courses Table</b>	<b>35</b>
<b>Table 4</b>	<b>Data Dictionary of Enrollments Table</b>	<b>36</b>
<b>Table 5</b>	<b>Unit Testing of SignUp/SignIn</b>	<b>63</b>
<b>Table 6</b>	<b>Unit Testing of Adding Course</b>	<b>64</b>
<b>Table 7</b>	<b>Unit Testing of Adding Users</b>	<b>65</b>
<b>Table 8</b>	<b>Functional Point Analysis on SignUp/SignIn</b>	<b>66</b>
<b>Table 9</b>	<b>Functional Point Analysis on Adding Course</b>	<b>66</b>
<b>Table 10</b>	<b>Functional Point Analysis on Add Users</b>	<b>66</b>

## Abbreviations

HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	JavaScript

## ABSTRACT

Skills are essential for cracking high-paying jobs, starting a company, and more. However, many students encounter challenges in finding suitable resources to acquire the skills demanded by modern industries. Our initiative, the “Enterprise E-Learning Platform,” is designed to alleviate this struggle by providing students with convenient access to a diverse range of courses tailored to their needs. We offer an extensive selection of courses at competitive prices, ensuring affordability without compromising quality. The project features a login system with two options: users can either log in using their email or continue with Google. Upon logging in, users can enroll in the courses they require, based on reviews and ratings for each course. Once a course is chosen, users can provide personal information and make payments to enroll. Users can effortlessly provide necessary personal information and complete the enrollment process by choosing from a variety of secure payment methods, including UPI, Net Banking, Credit Card, Debit Card, and an automated email system facilitating smooth transactions. Upon successful completion of the payment process, users receive a prompt confirmation email, granting them immediate access to their chosen course and enabling them to commence their learning journey without delay. Our platform also offers transparent tracking of user engagement, displaying valuable insights such as the number of courses enrolled and the total revenue generated. All user data, including enrolled courses, payment details, and course ratings and reviews, are stored in the database and displayed on the dashboard. Leveraging the power of MERN Technology (MongoDB, ExpressJS, ReactJS, NodeJS), our platform stands as a testament to innovation in online education, empowering learners worldwide to unlock their full potential and thrive in today's competitive landscape.

### **Technologies to be used:**

**Front-end :** HTML, CSS, JavaScript, ReactJS

**Back-end :** NodeJS, ExpressJS

**DataBase :** MongoDB

# 1.INTRODUCTION

In an era where digital education has become an integral part of the learning landscape, our platform aims to provide a holistic and interactive environment for individuals seeking knowledge and those looking to share their expertise. The platform will feature a range of functionalities, ensuring a smooth user experience and fostering an engaging and collaborative learning community. This platform is designed to cater to the needs of both learners and instructors, offering a seamless and enriching educational experience. The primary goal of this project is to create a robust and feature-rich e-learning solution that addresses the distinct requirements of students, instructors, and administrators alike.

By leveraging the capabilities of the MERN stack and adopting a comprehensive approach to platform development, we aim to deliver a robust, intuitive, and scalable solution capable of meeting the diverse learning needs of modern enterprises

.

## 1.1. The Evolution of E-Learning Platforms

The evolution of e-learning platforms has been shaped by advancements in technology, pedagogical theories, and the changing needs of learners and organizations. From early Learning Management Systems (LMS) to modern, interactive learning platforms, the landscape of online education continues to evolve rapidly. Enterprise E-Learning Platforms represent the next phase in this evolution, catering to the unique requirements of businesses and large-scale learning environment

## 1.2 The MERN Stack: A Framework for Modern Web Development

The MERN stack – MongoDB, Express.js, React.js, and Node.js – has gained prominence as a leading framework for building modern web applications. Each component of the stack brings its own strengths to the development process, enabling developers to create dynamic, efficient, and scalable solutions. MongoDB provides a flexible and scalable database solution, while Express.js simplifies the process of building robust web servers. React.js facilitates the creation of interactive user interfaces, and Node.js allows for the development of server-side



applications using JavaScript. Together, these technologies form a powerful toolkit for building enterprise-grade e-learning platforms.

### **1.3. Key Considerations in Enterprise E-Learning Platform Development**

Developing an enterprise e-learning platform entails addressing a myriad of challenges and considerations unique to the corporate environment. Scalability, security, integration with existing systems, and compliance with regulatory requirements are among the key factors that must be carefully navigated. Additionally, considerations such as user experience design, accessibility features, and analytics capabilities play a crucial role in optimizing learning outcomes and fostering user engagement.

### **1.4. Objectives of the Report**

This report aims to explore the architecture, design principles, and implementation strategies involved in the development of an enterprise e-learning platform using the MERN stack. By examining the technical aspects of platform development as well as the broader implications for organizational learning.

## **2.DESRIPTION**

The e-learning platform developed as part of this project is designed to provide a holistic and interactive environment for individuals seeking knowledge and expertise sharing within enterprises. Leveraging the capabilities of the MERN stack, the platform offers a robust, intuitive, and scalable solution capable of meeting the diverse learning needs of modern organizations.

### **2.1 Architecture**

The architecture of the e-learning platform is based on the MERN stack, comprising MongoDB, Express.js, React.js, and Node.js. MongoDB serves as the database solution, providing flexibility and scalability for storing and managing data related to users, courses, modules, and learning materials. Express.js facilitates the development of robust web servers, handling routing, middleware, and HTTP requests. React.js powers the frontend of the platform, enabling the creation of dynamic and interactive user interfaces. Node.js serves as the runtime environment for server-side applications, allowing for the implementation of business logic and integration with external systems.

### **2.2 Design Principles**

The design of the e-learning platform follows key principles aimed at enhancing user experience, accessibility, and engagement. The user interface is designed to be intuitive and user-friendly, with clear navigation and consistent design elements. Accessibility features, such as keyboard navigation and screen reader support, are implemented to ensure inclusivity for users with diverse needs. The platform adopts responsive design principles, ensuring compatibility across various devices and screen sizes.

### **2.3 Key Functionalities**

The e-learning platform offers a range of functionalities to facilitate learning and collaboration within enterprises:

- **User Authentication and Authorization:** Users can register, login, and manage their accounts securely. Role-based access control ensures appropriate access levels for learners, instructors, and administrators.
- **Course Management:** Instructors can create and manage courses, including adding modules, assignments, quizzes, and learning materials. Learners can enroll in courses and track their progress.
- **Interactive Learning Environment:** The platform provides interactive features such as discussion forums, live chat, and virtual classrooms to foster collaboration and engagement among learners and instructors.
- **Analytics and Reporting:** Analytics capabilities are integrated to track user activity, monitor learning progress, and generate reports for administrators and instructors. Insights derived from analytics data help optimize learning outcomes and identify areas for improvement.

## **2.4 Integration and Scalability**

The e-learning platform is designed to integrate seamlessly with existing enterprise systems, such as HR management systems and learning management systems (LMS). APIs and webhooks are utilized for data exchange and integration. The platform is built with scalability in mind, allowing for the addition of new features, courses, and users as the organization grows

## **3.SRS (SOFTWARE REQUIREMENTS SPECIFICATION) DOCUMENT**

### **3.1 Introduction:**

In an era where digital education has become an integral part of the learning landscape, our platform aims to provide a holistic and interactive environment for individuals seeking knowledge and those looking to share their expertise. The platform will feature a range of functionalities, ensuring a smooth user experience and fostering an engaging and collaborative learning community. This platform is designed to cater to the needs of both learners and instructors, offering a seamless and enriching educational experience. The primary goal of this project is to create a robust and feature-rich e-learning solution that addresses the distinct requirements of students, instructors, and administrators alike.

#### **3.1.1 Purpose:**

The purpose of the proposed online learning platform is to create a dynamic and user-centric environment that facilitates effective teaching and learning experiences. This platform aims to bridge the gap between learners and instructors, providing a comprehensive suite of features for education seekers, content creators, and administrators. Key objectives include:

- Enhanced Learning Experience
- Empower Instructors
- Administrative Efficiency
- Community Building:
- Scalability

#### **3.1.2 Scope:**

This document pertains to the development of an innovative Online eLearning Platform. This is designed to revolutionize the education sector by providing a comprehensive and interactive online learning environment. The platform caters to both learners and instructors, offering a wide range of features that enhance the teaching and learning experience.

This allows learners to register and log in, providing a personalized profile management system for updating personal information, profile pictures, and preferences. The platform

facilitates course exploration with a user-friendly catalog, search, filter, and sorting options. Learners can enroll and unenroll from courses, utilizing a Wishlist or bookmarking feature to save courses of interest. Learning progress and tracking features include tools for monitoring course progress, bookmarking within lectures or modules, and receiving personalized learning recommendations. The platform also incorporates badges, achievements, or certificates to acknowledge and motivate learners upon course completion.

Communication and collaboration are integral aspects of an eLearning platform, featuring discussion forums, direct messaging, and Q&A functionalities for learner-to-learner and learner-to-instructor interaction. Instructors benefit from profile creation and management tools, course creation, content upload, pricing, scheduling, and analytics. Content creation and organization tools enable instructors to create lectures, quizzes, assignments, and upload multimedia content. Administering the platform is simplified with an admin dashboard for managing settings, user accounts, and course content. User management features include role assignment, permissions, and account moderation.

The software is expected to be developed within a timeframe of Four months, with an estimated cost of Rs.10 lakhs. This scope outlines the functionalities and features that contribute to the eLearning platform, ensuring it becomes a dynamic and scalable online learning platform catering to the diverse needs of both learners and instructors.

### **3.1.3 Definitions, Acronyms, and Abbreviations.**

LMS	Learning Management System
SCORM	Sharable Content Object Reference Model
API	Application Programming Interface
LRS	Learning Record Store
ILT	Instructor-Led Training

VILT	Virtual Instructor-Led Training
eLearning	Electronic Learning
LXP	Learning Experience Platform
CMI	Content Management Interoperability
CMS	Content Management System

Table - 1 Definitions, Acronyms, Abbreviations

### 3.1.4 Reference:

The references for the above software are as follows: -

- i. [www.google.co.in](http://www.google.co.in)
- ii. [www.wikipedia.com](http://www.wikipedia.com)
- iii. <https://www.learnupon.com/blog/what-is-elearning/>
- iv. Enterprise E-Learning Success Factors: An Analysis of Practitioners' Perspective
- v. <https://www.techtarget.com/whatis/definition/Web-based-training-e-learning>

### 3.1.5 Overview:

Section 1.0 discusses the purpose and scope of the software.

Section 2.0 describes the overall functionalities and constraints of the software and user characteristics.

Section 3.0 details all the requirements needed to design the software.

Section 4.0 describes the system software features.

Section 5.0 detailed information on Non-Functional Requirements of Enterprise eLearning Platform.

## **3.2 The Overall Description**

### **3.2.1 Product Perspective**

- In today's digital age, mobile compatibility is essential. These platforms are designed to be accessible from various devices, including smartphones and tablets, allowing learners to access content anytime, anywhere.
- An enterprise eLearning platform is a software system that helps students to learn new skills and teachers can teach.
- It provides features for creating and uploading courses, tracking progress, all while scaling to accommodate large user bases and complex training needs.
- This software primarily focuses on upskill or reskill for professional development.
- Often deliver self-paced, asynchronous learning with content like videos, simulations, and quizzes. They can also facilitate collaborative learning through forums and discussions.
- May involve training professionals who create and curate content, but generally, there's no single "teacher" figure guiding individual learners.
- Aim to improve student performance and achieve organizational goals by building specific skills and knowledge relevant to their work.
- It provides a seamless experience with low latency experience using MERN technologies.

### **3.2.2 Product Functions**

The major functions that Enterprise eLearning Platform performs are described as follows: -

#### **For Students/Learner:**

##### **1. User Authentication and Profiles:**

- User registration and login functionality.
- User profile management with options to update personal information, profile picture, and preferences.
- Password reset and account settings.

## **2. Course Exploration and Enrollment:**

- Course catalog with search, filters, and sorting options.
- Course details with descriptions, instructor information, ratings, and reviews.
- Course enrollment functionality.
- Wishlist or bookmarking feature for saving courses of interest.

## **3. Learning Progress and Tracking:**

- Course progress tracking, including completed lectures, quizzes, and assignments.
- Bookmarking or saving progress within lectures or modules.
- Personalized learning paths or recommendations based on learner's interests and progress.
- Badges, achievements, or certificates for completed courses.

## **4. Communication and Collaboration:**

- Discussion forums or community spaces for learners to interact, ask questions, and engage with peers and instructors.
- Direct messaging or chat functionality for learner-to-learner or learner-to-instructor communication.
- Q&A features to submit questions and receive answers from instructors or fellow learners.

## **For Instructors/Course Creators:**

### **1. Instructor Profiles and Course Management:**

- Instructor profile creation and management with bio, expertise, and social links.
- Course creation and management with options to upload course content, set pricing, and schedule.
- Course editing and updating functionality.



- Course analytics and insights on learner engagement and course performance.

## **2. Content Creation and Organization:**

- Content creation tools for creating lectures, quizzes, assignments, and other learning materials.
- Content organization and structuring within courses or modules.
- Multimedia support for uploading videos, documents, and other supplementary materials.

## **3. Communication and Interaction:**

- Communication channels for instructors to engage with learners, provide announcements, and address queries.
- Instructor-led discussions or Q&A sessions within courses.
- Feedback and rating system for instructors to receive learner reviews and ratings.

## **For Admin/Platform Owner:**

### **1. Admin Dashboard and User Management:**

- Admin dashboard for managing platform settings, user accounts, and course content.
- User management features for user role assignment, permissions, and account moderation.
- User analytics and reporting.

### **2. Course Content Management:**

- Course content moderation, review, and approval workflows.
- Ability to edit or remove inappropriate or flagged content.
- Quality control measures for ensuring course content meets platform guidelines.

### **3. Platform Analytics and Reporting:**

- Platform analytics and reporting features to track usage, learner engagement, revenue, and other key metrics.
- Reporting and analytics for user-generated content, including feedback and ratings.

## **2.3 User Characteristics**

There are different kinds of users that will be interacting with the system. The intended user of the software are as follows: -

### **Students/Learners:**

For students, the Enterprise eLearning Platform caters to a diverse audience with varying demographics, tech savviness, learning preferences, and motivation levels. The platform accommodates users of different ages, educational backgrounds, and interests, acknowledging that individuals may have distinct learning styles and preferences. Whether users are visual, auditory, or kinesthetic learners, the platform aims to provide a comprehensive and engaging learning experience. Furthermore, it recognizes that motivation levels and commitment to learning may differ among users.

### **Instructors/Course Creators:**

Instructors and course creators on the platform exhibit a spectrum of characteristics. They may possess varying levels of teaching experience, ranging from seasoned educators to industry professionals new to the realm of teaching. Additionally, the platform acknowledges differences in content expertise, as instructors contribute courses based on their unique areas of knowledge. The technological proficiency of instructors varies, with some being adept at using eLearning tools and technology, while others may be in the process of developing these skills. Effective communication with learners is also recognized as a key skill set that varies among instructors.

### **Admin/Platform Owner:**

Administrators and platform owners demonstrate specific characteristics essential for the effective management of the eLearning platform. Strong technical proficiency is crucial, encompassing a deep understanding of platform management, analytics, and reporting tools. Administrative skills play a pivotal role in efficiently handling user accounts, content moderation, and platform settings. The ability to think analytically is emphasized, as administrators interpret analytics to make informed decisions for platform improvement. Quality control measures are also a focus, ensuring that course content aligns with established guidelines and meets high standards of quality and relevance.

### **3.2.4 Constraints**

The major constraints that the project has are as follows: -

#### **Technical Constraints:**

- **Scalability:** The platform should be designed to handle a potentially large number of concurrent users without sacrificing performance.
- **Accessibility:** Compliance with accessibility standards to ensure the platform is usable by individuals with disabilities.
- **Integration:** Compatibility with existing learning management systems (LMS), if applicable, to facilitate data exchange and interoperability.
- **Data Privacy:** Adherence to data privacy regulations such as GDPR or CCPA to protect user data and ensure compliance.
- **Localization:** Support for multiple languages and cultural considerations to cater to a diverse user base.
- **Browser Compatibility:** Ensuring the platform functions correctly across different web browsers and versions to provide a consistent user experience.

#### **Time Constraints:**

- **Regulatory Compliance:** Ensuring that the platform meets all relevant legal and regulatory requirements related to e-learning platforms.
- **Testing:** Comprehensive testing within the project timeline to identify and resolve any issues or bugs before deployment.
- **Content Development:** Creation and upload of course content within the specified timeframe to ensure the platform launches with sufficient educational materials.
- **Training:** Provision of training sessions for administrators, instructors, and support staff within the project schedule to ensure smooth operation post-launch.

#### **Budget Constraints:**

- **Resource Allocation:** Optimizing resource allocation, including personnel, infrastructure, and tools, to stay within the specified budget.

- **Third-Party Costs:** Consideration of any additional costs associated with third-party services or integrations required for the platform.
- **Contingency Planning:** Setting aside a contingency budget for unexpected expenses or scope changes that may arise during the project.
- **Cost-Effective Solutions:** Prioritization of features and functionalities based on their impact and cost-effectiveness to maximize the value within the budget constraints.
- **Cost of Maintenance:** Estimating ongoing maintenance costs beyond the initial development phase to ensure long-term sustainability within the budget.

### **3.2.5 Assumptions and Dependencies**

- The requirements stated in the SRS could be affected by the following factors:
- It is furthermore assumed that the scope of the package will increase considerably in the future.
- The successful implementation of the Enterprise eLearning Platform outlined in this project is based on certain assumptions and dependencies. It is crucial to acknowledge these factors to ensure a realistic understanding of the project's scope and potential challenges.

#### **Assumptions:**

- **Technological Advancements:** The platform assumes a continued advancement in technology and compatibility with emerging web and mobile technologies. It relies on the availability of stable and secure internet connections for users.
- **User Accessibility:** It is assumed that users have access to compatible devices (computers, tablets, smartphones) and possess the basic technical skills required for interacting with the e-learning platform.
- **Content Creation:** The successful functioning of the platform assumes that instructors and course creators have the necessary tools and skills to create high-quality educational content, including multimedia elements.
- **User Engagement:** The effectiveness of the platform relies on active user participation, including both learners and instructors engaging in discussions, collaboration, and providing feedback.

- Scalability: The platform assumes a scalable infrastructure to accommodate a growing user base, increasing volumes of courses, and expanding functionalities as the platform evolves.
- The successful implementation of the Enterprise eLearning Platform outlined in this project is based on certain assumptions and dependencies. It is crucial to acknowledge these factors to ensure a realistic understanding of the project's scope and potential challenges.

### **Dependencies:**

- Third-Party Integrations: The successful implementation of the e-learning platform may depend on integrations with external services or tools, such as payment gateways for transaction processing, video hosting platforms, or authentication services.
- Regulatory Compliance: The platform's functionality assumes adherence to relevant educational and data protection regulations. Any changes in these regulations may impact the platform's features and data handling procedures.
- Continuous Development: The platform's success relies on ongoing development and updates to meet evolving user needs, address security concerns, and incorporate technological advancements.
- User Adoption: The platform's success is dependent on user adoption and acceptance. Strategies for user onboarding and engagement will play a critical role in the platform's overall success.
- Collaboration with Instructors: The platform assumes active collaboration with instructors for creating, updating, and managing course content. The success of the platform is tied to the commitment and involvement of qualified and motivated instructors.
- Community Building: The effectiveness of community features, such as discussion forums, depends on the willingness of users to actively participate and contribute positively to the learning community.

## **3.3. External Interface Requirements**

### **3.3.1 User Interface Requirements**

The interface provided to the user should be a very user-friendly one and it should provide an optional interactive help for each of the services listed. The interface provided is a menu driven one and the following screens will be provided: -

#### **Learner Interface:**

##### **Course Catalog:**

- Clear and visually appealing presentation of available courses.
- Sorting options based on relevance, popularity, or category.

##### **Enrollment Process:**

- Intuitive enrollment process with clear instructions and a progress tracker.
- Confirmation messages upon successful enrollment.

##### **Learning Dashboard:**

- Personalized dashboard displaying enrolled courses, progress, and recommendations.
- Visual indicators for course completion and upcoming deadlines.

#### **Profile Management Interfaces:**

##### **User Profile:**

- Easy-to-navigate profile pages with options for learners to update personal information.
- Customizable profile pictures and preferences.

##### **Settings and Preferences:**

- User-friendly account settings with options for notification preferences.
- Password reset functionality with clear instructions.

**Admin Dashboard:****System Overview:**

- Comprehensive overview of user activity, course statistics, and platform health.
- Quick access to critical settings and reports.

**User Management:**

- Efficient tools for managing user accounts, roles, and permissions.
- Account moderation features for addressing rule violations.

### **3.3.2 Hardware Interface Requirements**

**Device Compatibility:**

- Ensure compatibility with a range of devices, including desktops, laptops, tablets, and smartphones.
- Responsive design for optimal user experience across different screen sizes.

**Resource Optimization:**

- Efficient use of device resources to prevent excessive resource consumption.
- Optimal loading times for multimedia content.

### **3.3.3 Software Interface Requirements**

To perform various functions, this software needs to interact with various other software's. So, there are certain software interface requirements that need to be fulfilled which are listed as follows: -

**Operating System Compatibility:**

- Compatibility with major operating systems such as Windows, macOS, and Linux.
- Consistent functionality across different operating systems.

**MERN Stack Integration:**

- Seamless integration with MongoDB, Express.js, React, and Node.js components.
- Ensuring the platform leverages the strengths of the MERN stack for performance and scalability.

**3.3.4 Communication Interface Requirements**

The machine needs to communicate with the main branch for each session for various functions such as login verification, account access etc. so the following are the various communication interface requirements that are needed to be fulfilled in order to run the software successfully:

**Learner Interaction:****Discussion Forums:**

- Robust discussion forums with threaded discussions for organized interaction.
- Features for attaching multimedia files or links within discussions.

**Direct Messaging:**

- User-friendly direct messaging features for private learner-to-learner communication.
- Notification system for new messages.

**Instructor Communication:****Announcements:**

- Instructor ability to post announcements visible to all enrolled learners.
- Timely notifications for learners about important announcements.



**Q&A Sessions:**

- Platform support for scheduled Q&A sessions initiated by instructors.
- Learner ability to submit questions in advance or during live sessions.

### **3.4. System Features**

**For Students/Learners:****Intuitive User Interface:**

- Clean and intuitive user interface design for easy navigation and usability.
- Responsive design to ensure seamless user experience across devices.

**Comprehensive Course Catalog:**

- User-friendly course catalog with advanced search, filter, and sorting options.
- Detailed course descriptions, instructor profiles, and ratings/reviews to aid decision-making.

**Personalized Learning Experience:**

- User profiles with customizable preferences and settings for a personalized learning journey.
- Progress tracking features to monitor course completion, achievements, and learning milestones.

**Interactive Learning Tools:**

- Interactive quizzes, assessments, and assignments embedded within course modules.
- Multimedia support for video lectures, presentations, and downloadable resources.

**For Instructors/Course Creators:****Efficient Course Management:**

- Easy-to-use course creation and management tools with drag-and-drop functionality.

- Content upload and editing features for lectures, quizzes, assignments, and supplementary materials.

#### **Real-time Course Analytics:**

- Analytics dashboard to track learner engagement, progress, and performance on a per-course basis.
- Insights into course popularity, completion rates, and areas for improvement.

#### **Engagement and Communication:**

- Discussion forums and chat functionalities for real-time interaction between instructors and learners.
- Announcement features for sharing updates, reminders, and important information with course participants.

#### **For Admin/Platform Owner:**

##### **Robust Admin Dashboard:**

- Centralized admin dashboard for platform management, user administration, and content moderation.
- Role-based access control with customizable permissions for admins and moderators.

##### **Content Moderation and Quality Control:**

- Tools for content moderation, including flagging, reporting, and removal of inappropriate content.
- Quality assurance checks to ensure course content meets platform guidelines and standards.

##### **Platform Customization:**

- Customizable branding options to align the platform with the organization's branding guidelines.
- Configurable settings for platform features, pricing, and user policies.

**Security and Data Privacy:**

- Robust security measures to protect user data, including encryption, secure authentication, and data access controls.
- Compliance with data privacy regulations such as GDPR to safeguard user privacy rights.

**Infrastructure and Technical Features:****Scalability and Performance:**

- Scalable architecture leveraging cloud infrastructure to handle varying levels of user traffic.
- Performance optimization techniques to ensure fast loading times and responsive user interactions.

**Reliable Backend Infrastructure:**

- Utilization of Node.js for backend development to ensure a reliable and efficient server-side infrastructure.
- Integration with MongoDB for flexible and scalable database management.

**Interactive Frontend Development:**

- Frontend development using React.js for building dynamic and interactive user interfaces.
- Component-based architecture for modular and maintainable frontend code.

**API Integration and Third-party Services:**

- Integration with third-party services such as payment gateways, analytics platforms, and communication tools via RESTful APIs.
- Seamless integration with external services for extended functionality and enhanced user experience.

## **3.5. Other Nonfunctional Requirements**

### **3.5.1 Performance Requirements:**

#### **3.5.1.2 Capacity:**

##### **Concurrent Users:**

- The platform must support a minimum of 10,000 concurrent users during peak usage times.
- Scalability considerations to handle up to 20,000 concurrent users to accommodate potential future growth.

##### **Course Enrollment:**

- Simultaneous enrollment capacity of at least 1,000 users across various courses.
- Efficient handling of peak enrollment periods without degradation of system performance.

##### **Data Throughput:**

- Ensure a minimum data throughput rate of 1 Gbps to support seamless content delivery.
- Adequate bandwidth for multimedia content streaming without buffering delays.

#### **3.5.1.3 Scalability:**

##### **User Base Growth:**

- The system should scale horizontally to handle a 50% growth in the user base within the next year.
- Vertical scaling options should be considered to accommodate increased system demands.

**Course Content Scaling:**

- Scalability to accommodate an expanding library of courses, modules, and multimedia content.
- Load balancing strategies for even distribution of user requests across servers.

**3.5.1.4 Dynamic Requirements:****User Interactions:**

- User interactions, such as course navigation and content access, should have a response time of less than 1 second.
- Dynamic loading of content to provide a seamless and uninterrupted learning experience.

**Latency Reduction:**

- Achieve a low-latency experience by optimizing database queries and leveraging caching mechanisms.
- Implementing content delivery networks (CDNs) to minimize latency for global users.

**3.5.1.5 Quality:****Multimedia Support:**

- Support high-quality multimedia content, including videos, interactive simulations, and 3D models.
- Optimization for various multimedia formats and resolutions to ensure compatibility.

**Consistent Performance:**

- Implement regular performance monitoring and optimization processes to maintain consistent system performance.
- Conduct load testing to identify and address performance bottlenecks proactively.

**Content Loading Times:**

- Multimedia content, including videos and images, should load within 3 seconds on average.
- Implement progressive loading techniques for multimedia content to enhance user experience.

**3.5.2 Software System Attributes:****3.5.2.1 Reliability:****Fault Tolerance:**

- Implement fault-tolerant mechanisms to handle unexpected system failures without causing service disruptions.
- Automatic system failover in case of server or component failures.

**Error Handling:**

- Comprehensive error handling to gracefully manage and recover from system errors.
- User-friendly error messages that guide users in case of unexpected events.

**Monitoring and Logging:**

- Real-time monitoring of system health and performance metrics.
- Extensive logging for debugging and post-incident analysis.

**Rollback Mechanism:**

- Implement a rollback mechanism for updates to revert to the previous stable version in case of issues.
- Automated rollback procedures triggered by critical failures during updates.

### **3.5.2.2 Availability:**

#### **Cross-Device Compatibility:**

- Ensure the platform's accessibility from various devices, including desktops, laptops, tablets, and smartphones.
- Responsive design and adaptive layouts to accommodate different screen sizes and resolutions.

#### **Server Uptime:**

- Maintain a minimum server uptime of 99.9%.
- Scheduled maintenance windows communicated to users in advance to minimize disruption.

#### **Content Delivery:**

- Implement content delivery networks (CDNs) to enhance content availability and reduce latency.
- Distributed server architecture to ensure content is served from the nearest server.

### **3.5.2.3 Security:**

#### **User Data Encryption:**

- Use strong encryption algorithms to secure user data during transmission and storage.
- Implementation of HTTPS protocol for secure data transmission.

#### **Authentication and Authorization:**

- Implement secure authentication mechanisms, such as multi-factor authentication, to protect user accounts.
- Role-based access control (RBAC) to ensure proper authorization for different user roles.

**Security Audits:**

- Regular security audits and vulnerability assessments conducted by independent security experts.
- Immediate patching of identified vulnerabilities with a documented process for response.

**Data Privacy Compliance:**

- Adherence to data privacy regulations (e.g., GDPR, CCPA) to protect user privacy rights.
- Transparent privacy policies communicated to users.

**3.5.2.4 Maintainability:****Modular Codebase:**

- Maintain a modular and well-organized codebase for ease of updates and future enhancements.
- Implement version control to track changes and facilitate collaborative development.

**Documentation:**

- Comprehensive documentation for developers, administrators, and end-users.
- User-friendly guides for routine tasks, troubleshooting, and system configuration.

**Automated Testing:**

- Implement automated testing suites for regression testing and ensuring code changes do not introduce new issues.
- Continuous integration and continuous deployment (CI/CD) pipelines for automated testing and deployment.

**Scalability Planning:**

- Documented scalability plans to accommodate future growth and changes in user demand.



- Scalability testing procedures to validate the platform's ability to handle increased loads.

### **3.5.3 Business Rules**

**Let's break down the provided business model and evaluate its feasibility in real-life scenarios:**

#### **Revenue Model:**

**Course Fee:** Let each course is priced at 100 rupees. And for every 100 members there will be an increase of 5% in course price.

#### **Cost Structure:**

##### **Cost of Goods Sold (COGS):**

- The primary cost associated with each course sold is the Goods and Services Tax (GST) imposed by the government. The GST rate is 18%.

#### **Revenue Allocation:**

**Teacher Revenue:** Teachers receive 40% of the course fee.

**Admin Revenue:** Administrators receive 60% of the course fee.

#### **Expenses Allocation:**

- **Marketing:** 10% of the total revenue is allocated to marketing expenses.
- **Payment Gateway:** 1% of the total revenue is allocated to payment gateway charges.
- **Others:** 9% of the total revenue is allocated to other miscellaneous expenses.

**Profit Calculation:**

- **Profit Before Tax:** The profit before tax is calculated as the difference between total revenue and total expenses. We get 40%.
- **Tax:** A tax rate of 8% is applied to the profit before tax.
- **Profit After Tax:** The profit after tax is calculated as the profit before tax minus the tax amount. We get 32%.

**Evaluation:****Revenue Model:**

- The course fee of 100 rupees seems reasonable for an e-learning platform.

**Cost Structure:**

- The inclusion of GST as a cost of goods sold is accurate and aligns with real-world taxation.

**Revenue Allocation:**

- Allocating 40% to teachers and 60% to admins can vary depending on the platform's business model and agreements with instructors. It's essential to ensure that this split is fair and incentivizes teachers to contribute quality content.

**Expenses Allocation:**

- The allocation of revenue to marketing, payment gateway charges, and other expenses seems reasonable and accounts for essential operational costs.

**Profit Calculation:**

- The profit before tax, tax rate, and profit after tax calculations appear accurate.

**Example:** Course Price 100 + GST (18%). Total Price 118.

Course Instructor will get 40% of total and Admin will get 60% of total.

Now Marketing, Payment Gateway and others will cost 20% from Admin.

Profit before Tax: 40% for admin. Now Tax is 6%. So, Profit after Tax is 36% (Finally).

### **3.6. Other Requirements**

#### **Comprehensive Testing Requirements:**

##### **Regression Testing:**

- Conduct regression testing to ensure that new features or modifications do not adversely affect existing functionalities.
- Automated regression test suites to streamline testing processes.

##### **Performance Testing:**

- Performance testing to evaluate the platform's responsiveness, scalability, and stability under various load conditions.
- Load testing, stress testing, and endurance testing to identify performance bottlenecks and optimize system resources.

##### **Security Testing:**

- Security testing to assess the platform's resilience against potential threats and vulnerabilities.
- Penetration testing to identify and address security weaknesses in the system.

#### **Documentation Requirements:**

##### **End-User Documentation:**

- User manuals or guides for learners, instructors, and administrators to navigate the platform effectively.
- Video tutorials or interactive guides for onboarding and training purposes.

**Developer Documentation:**

- Technical documentation for developers, including API references, coding standards, and architecture diagrams.
- Documentation for setting up development environments and contributing to the platform's codebase.

**Appendix A: Glossary:****Comprehensive Term Definitions:**

- A detailed glossary containing definitions for all technical terms, acronyms, and industry-specific terminology used throughout the document.
- Definitions provided in a clear and concise manner to facilitate understanding for all stakeholders.

**Appendix S: Analysis Models:****Flowcharts:**

- Visual representations of system processes, user workflows, and data flows within the platform.
- Flowcharts to illustrate the sequence of steps involved in course enrollment, content creation, and user interactions.

**UML Diagrams:**

- Unified Modeling Language (UML) diagrams, such as use case diagrams, class diagrams, and sequence diagrams, to depict system architecture and interactions.
- Use case diagrams to describe user interactions with the platform, including learners, instructors, and administrators.

**Wireframes:**

- Wireframes for the platform's user interface design, depicting layout, navigation, and content placement.
- Interactive prototypes or mockups to gather feedback and validate design concepts before implementation.

## 4.DATA FLOW DIAGRAM

### 4.1 Level 0:

Level - 0

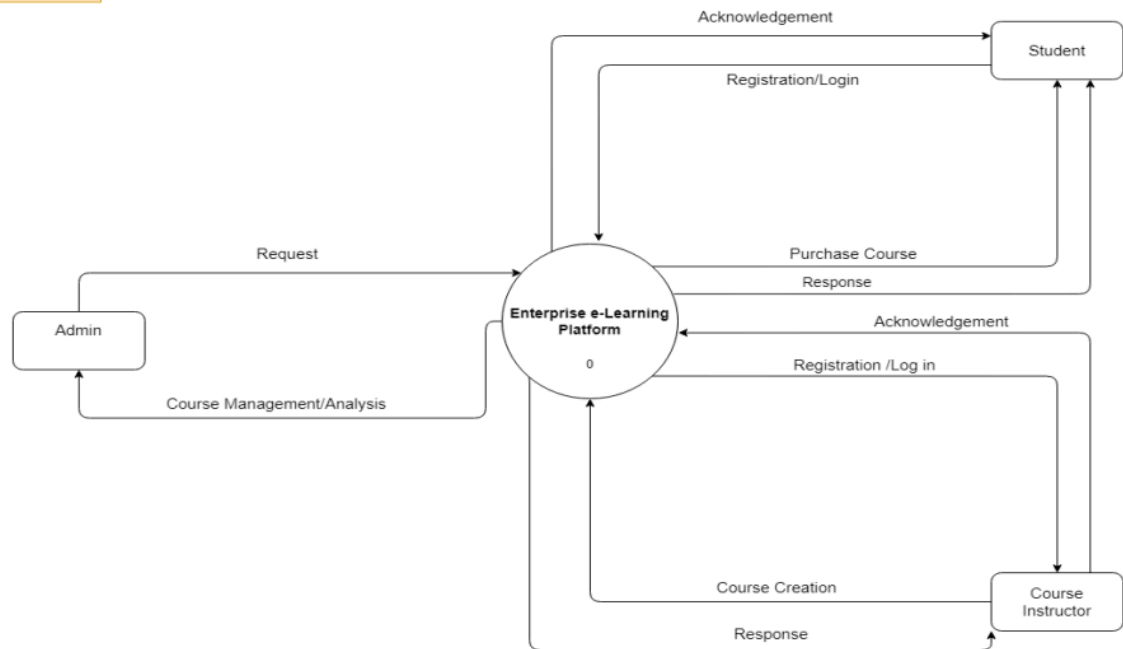


Figure - 1 Level 0 Data Flow Diagram

A level 0 data flow diagram (DFD) for a course management system. It shows the high-level processes involved in the system and how data flows between them. Here's a more detailed breakdown of the diagram:

- **Registration/Login:** This process allows students to register for an account or log in to an existing account.
- **Request:** Once logged in, a student can request to purchase a course.
- **Response:** The system responds to the student's request by either acknowledging it or providing a reason why the purchase cannot be made.
- **Course Creation:** Instructors can create courses and add content to them.
- **Course Management/Analysis:** This process allows administrators to manage and analyze courses. This might include tasks like enrolling students, tracking progress, and generating reports.
- **Acknowledgement:** The system sends an acknowledgement to the enterprise e-learning platform once a course is created or purchased.

The data flow diagram does not show the specific details of how each process works, but it provides a good overview of the system's functionality. Data flow diagrams are a type of flowchart that can be used to visualize the flow of data in a system. They are often used in the early stages of system design to help identify the system's components and how they interact with each other.

## 4.2 Level 1:

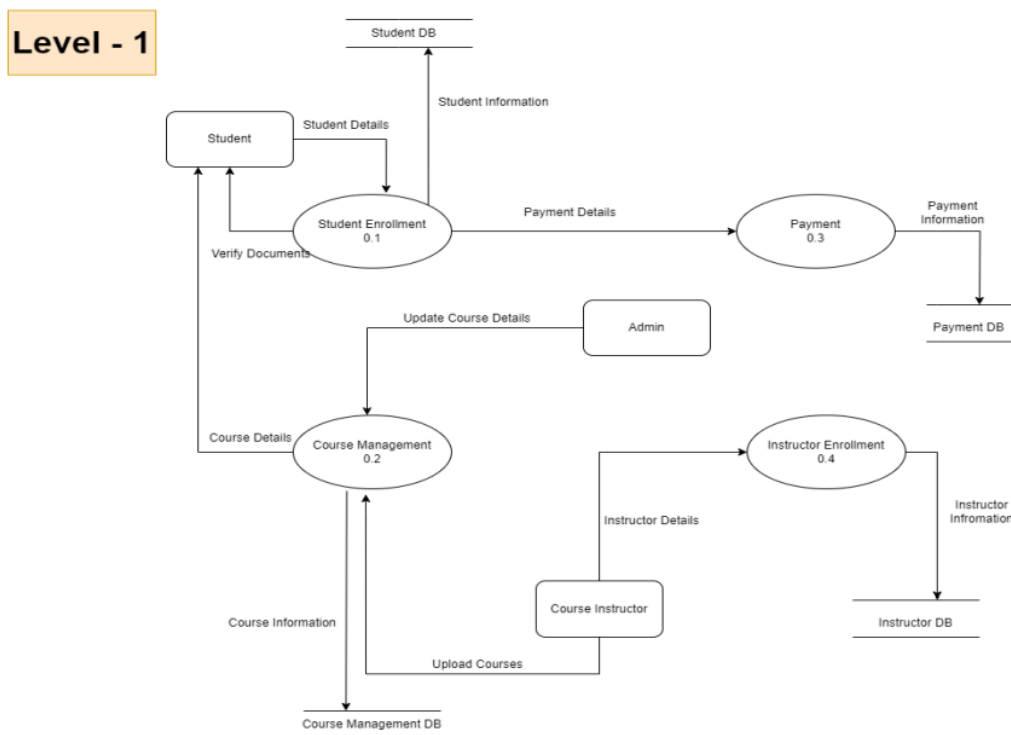


Figure - 2 Level 1 Data Flow Diagram

- A level 1 DFD, which is more detailed than a level 0 DFD. Here's a breakdown of the information in the level 1 DFD:
- Student: The student interacts with the system by providing their details and uploading required documents.
- Verify Documents: The system verifies the uploaded documents.
- Student Enrollment: After document verification, the student gets enrolled in the course. This process updates the Student DB and Payment DB.
- Admin: The admin can update course details and manage the system.
- Course Management: This process updates the Course Management DB.

- **Instructor:** The instructor interacts with the system by providing their details and enrolling courses.
- **Course Instructor:** This process updates the Instructor DB and Course Management DB.
- **Upload Courses:** The instructor uploads courses to the system. This process updates the Course Management DB.

### 4.3 Level 2 :

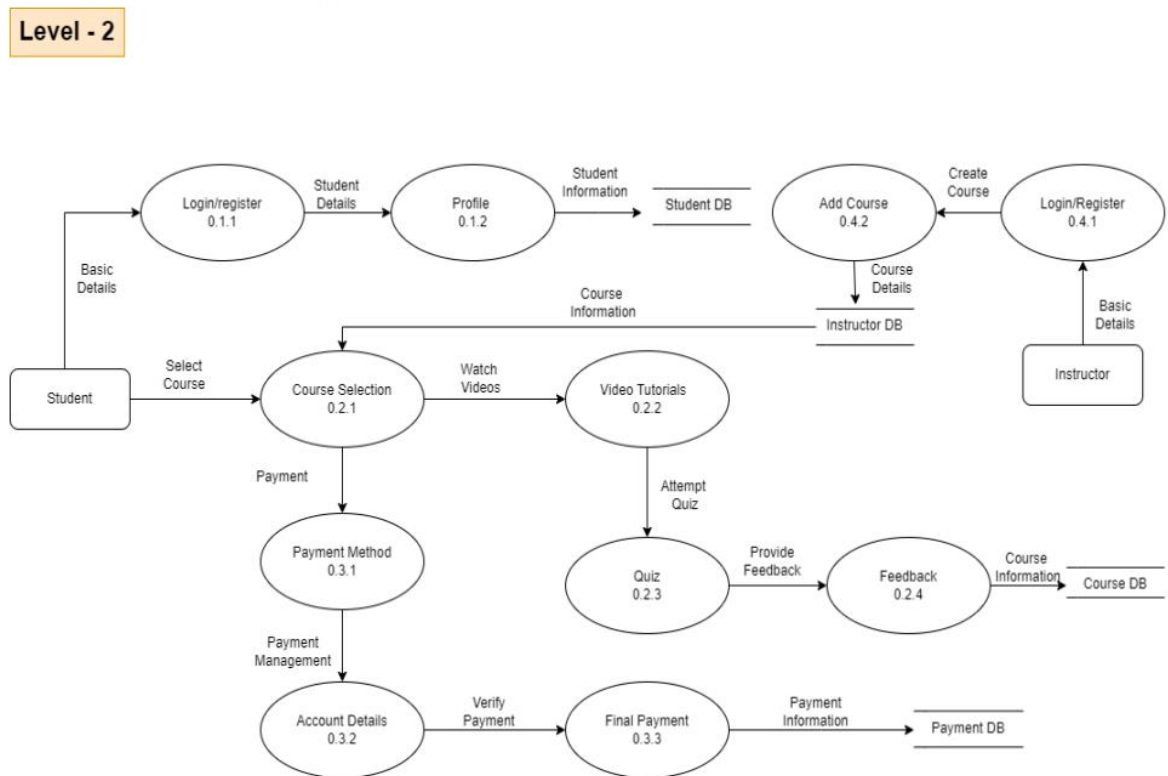


Figure - 3 Level 2 Data Flow Diagram

- **Login/Register:** The student interacts with the system by providing their login credentials or registering for a new account.
- **Student DB:** The system retrieves or stores student information in the Student DB.
- **Student Profile:** The student updates their profile information. This information is stored back in the Student DB.
- **Create Course:** The student searches for courses and adds them to their cart. This data is stored in the Student DB.



- **Course DB:** The system retrieves course information from the Course DB and displays it to the student.
- **Select Course:** The student selects a course from their cart.
- **Course Selection:** The system retrieves the selected course details from the Course DB and displays them to the student.
- **Watch Videos:** The system retrieves video lectures for the selected course and streams them to the student.
- **Payment:** The student selects their preferred payment method and enters their payment information.
- **Payment Method:** The system validates the student's chosen payment method.
- **Payment Information:** The system validates the student's credit card or bank account information.
- **Payment Gateway:** The system sends the encrypted payment information to the payment gateway for processing.
- **Payment DB:** The system stores the payment information and status in the Payment DB.
- **Attempt Quiz:** The student attempts the quiz for the selected course.
- **Course DB:** The system retrieves the quiz questions and stores the student's quiz attempt in the Course DB.
- **Provide Feedback:** The student provides feedback about the course. This feedback is stored in the Course DB.

## 5.DATA DICTIONARY

### Users Table

This table stores information about the users or students of the e-learning platform.

Field Name	Data Type	Description
user_id	Integer	Unique identifier for the user
username	String	User's username for login purposes
password	String	Encrypted password for login
email	String	User's email address
first_name	String	User's first name
last_name	String	User's last name
date_joined	Date	Date when the user joined the platform
is_instructor	Boolean	Flag indicating if user is an instructor

Table -2 Data Dictionary of Users table

### Courses Table

This table stores information about the courses available on the platform.

Field Name	Data Type	Description
course_id	Integer	Unique identifier for the course
title	String	Title of the course
description	Text	Description or overview of the course
instructor_id	Integer	User ID of the instructor teaching the course
price	Decimal	Price of the course (if applicable)
start_date	Date	Start date of the course
end_date	Date	End date of the course
is_published	Boolean	Flag indicating if course is published and visible

Table - 3 Data Dictionary of Courses Table

## Enrollments Table

This table tracks the enrollments of users in courses.

Field Name	Data Type	Description
enrollment_id	Integer	Unique identifier for the enrollment
user_id	Integer	User ID of the student enrolled in the course
course_id	Integer	Course ID of the course the user is enrolled in
enrollment_date	Date	Date when the user enrolled in the course
completion_status	String	Status of completion (e.g., In Progress, Completed)
grade	Decimal	Grade achieved by the user in the course

Table - 4 Data Dictionary of Enrollments Table

This is a basic structure and can be expanded based on the specific needs of the e-learning platform. Additional tables may include sections for lessons, quizzes, assignments, discussions, and more, depending on the platform's features. Each table would then have its own set of fields to represent the relevant data.

## 6. USE CASE DIAGRAM

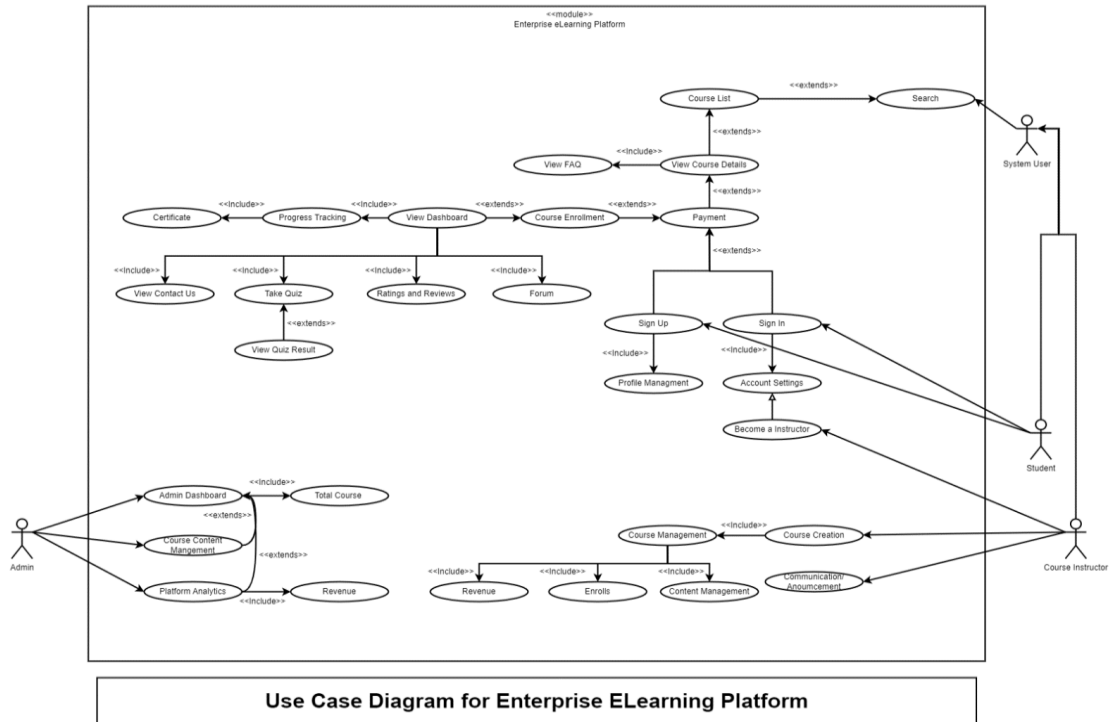


Figure 4. Use Case Diagram

### 6.1 Actors:

- Students:** These are the individuals who take courses on the platform. They can view course details, enroll in courses, track their progress, view certificates, take quizzes, view ratings and reviews, participate in forums, view frequently asked questions (FAQs), sign up for the platform, sign in, view quiz results, manage their profiles, and update their account settings.
- Course instructors:** These are the individuals who create and manage courses on the platform. They can view course content, manage course content, create courses, manage enrollments, manage content, send announcements, and view platform analytics.
- Admins:** These are the individuals who manage the platform. They can view revenue, manage enrollments, manage content, manage communication, and view course instructors.

## 6.2 Use Cases:

- **Course Management:** This use case group includes all the functions related to managing courses, such as creating, editing, and deleting courses.
- **Course Creation:** This use case allows the admin to create new courses.
- **Total Course:** This use case shows the total number of courses available on the platform.
- **Content Management:** This use case allows the admin to manage the content of the courses, such as uploading and editing learning materials.
- **Communication:** This use case group includes all the functions related to communication between users, such as sending announcements and messages.
- **Announcement:** This use case allows the admin to send announcements to all users or specific groups of users.
- **Enrolment:** This use case allows users to enroll in courses.
- **Payment:** This use case allows users to pay for courses (if applicable).
- **Certificate:** This use case allows users to view and download their certificates of completion for courses.
- **Progress Tracking:** This use case allows users to track their progress in courses.
- **View Dashboard:** This use case allows users to view their dashboard, which shows them their enrolled courses, progress, and other relevant information.
- **Search:** This use case allows users to search for courses by keyword.
- **Course List:** This use case shows a list of all the courses available on the platform.
- **View Course Details:** This use case allows users to view the details of a specific course, such as the description, learning objectives, and instructor.
- **Take Quiz:** This use case allows users to take quizzes and exams.
- **View Quiz Results:** This use case allows users to view their results for quizzes and exams. (This use case is not explicitly shown in the image, but it is implied by the existence of the "Take Quiz" use case.)
- **Ratings and Reviews:** This use case allows users to rate and review courses.
- **Forum:** This use case allows users to participate in discussion forums related to the courses.
- **Profile Management:** This use case allows users to manage their profiles, such as updating their contact information and preferences.

- **Account Settings:** This use case allows users to manage their account settings, such as changing their password.
- **Become an instructor:** This use case allows users to sign up to become an instructor on the platform.
- **Sign Up:** This use case allows users to create new accounts on the platform.
- **Sign In:** This use case allows users to sign in to their existing accounts on the platform.
- **View Contact Us:** This use case shows the contact information for the platform.

### 6.3 Relationships:

- **Includes:** This relationship indicates that one use case includes the functionality of another use case. For example, the "View Course Details" use case includes the "View FAD" and "View Course Details" use cases.
- **Extends:** This relationship indicates that one use case extends the functionality of another use case. For example, the "Search" use case extends the "Course List" use case by adding the ability to search for courses.

The use case diagram for the enterprise eLearning platform serves as a valuable tool in conveying the system's functionality to various stakeholders involved in its development and utilization. At its core, the diagram offers a visual representation of the system's interactions with external entities, highlighting key functionalities and user roles. By identifying and illustrating distinct use cases, such as user registration, course enrolment, content creation, and assessment, the diagram helps stakeholders comprehend the system's overall structure and purpose.

This high-level overview facilitates effective communication between developers, project managers, and end-users, fostering a shared understanding of the platform's requirements. Stakeholders can gain insights into the user journey, system capabilities, and the interplay between different components, ensuring that the development process aligns with the intended goals and user expectations. Additionally, the use case diagram provides a foundation for iterative refinement and validation of requirements throughout the development lifecycle,

allowing for agile adjustments based on feedback and evolving business needs. Overall, this visual representation is a crucial artifact that streamlines collaboration and guides the development team toward building a robust and user-friendly enterprise eLearning platform.

## 7.ER DIAGRAM

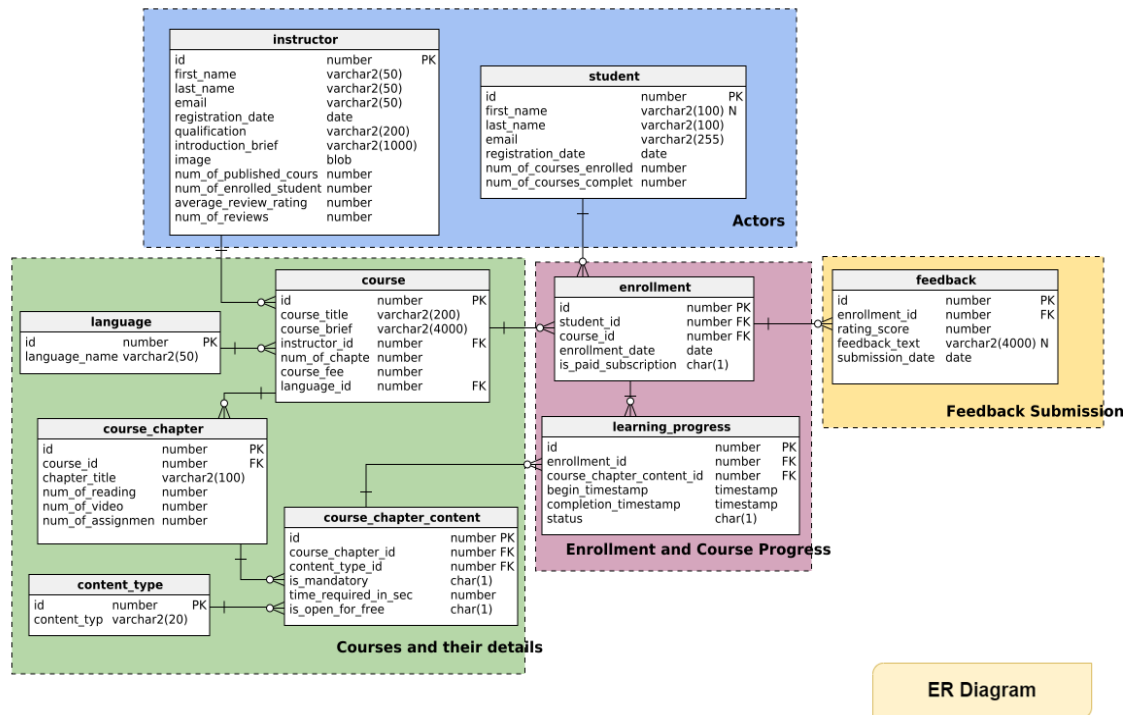


Figure - 5 ER Diagram

### 7.1 ER Diagram

An ER diagram for an enterprise e-learning platform would likely include several tables representing different aspects of the learning system. Here's a breakdown of some possible tables and their attributes:

#### Users:

- UserID (Primary Key): Unique identifier for each user
- Username: Login name for the user
- Password: Securely stored password for user authentication
- Name: Full name of the user
- Email: User's email address
- Role: User's role in the system (e.g., Learner, Instructor, Administrator)

#### Courses:



- CourseID (Primary Key): Unique identifier for each course
- Title: Name of the course
- Description: A brief overview of the course content
- Category: Subject area or category the course belongs to (e.g., IT, Management)
- Difficulty: Level of difficulty (e.g., Beginner, Intermediate, Advanced)
- InstructorID (Foreign Key): References the UserID of the instructor who created the course

#### **Modules:**

- ModuleID (Primary Key): Unique identifier for each module within a course
- CourseID (Foreign Key): References the CourseID of the course the module belongs to
- Title: Name of the module
- Description: Brief overview of the module content
- Order: Sequence in which the module appears within the course

#### **Content:**

- ContentID (Primary Key): Unique identifier for each piece of content
- ModuleID (Foreign Key): References the ModuleID of the module the content belongs to
- Type: Type of content (e.g., Video, Document, Quiz)
- Title: Title of the content piece
- Data: The actual content data (e.g., video file, text document, quiz questions)

#### **Enrollments:**

- EnrollmentID (Primary Key): Unique identifier for each user enrollment
- UserID (Foreign Key): References the UserID of the learner enrolled
- CourseID (Foreign Key): References the CourseID of the course the user is enrolled in
- Status: Enrollment status (e.g., In Progress, Completed)
- CompletionDate: Date the user completed the course (if applicable)

**Progress:**

- ProgressID (Primary Key): Unique identifier for each user's progress record
- EnrollmentID (Foreign Key): References the EnrollmentID of the specific enrollment
- ModuleID (Foreign Key): References the ModuleID of the module the progress refers to
- Status: Completion status for the module (e.g., Not Started, In Progress, Completed)
- Score: Score achieved on a quiz or assessment within the module (if applicable)

**Additional Tables:**

- Depending on the platform's features, there might be additional tables for things like:
- Assessments: To store details about quizzes and exams.
- Certificates: To track and record course completion certificates.

**Relationships:**

- The ER diagram would show relationships between these tables using lines. For example:
- A User can enroll in many Courses (one-to-many).
- A Course can have many Modules (one-to-many).
- A Module can have many Content pieces (one-to-many).
- A User can have many Enrollments (one-to-many).
- An Enrollment belongs to one User and one Course (many-to-many relationship represented by a separate Enrollment table).

## 8.CLASS DIAGRAM

A Class is a blueprint for an object. Objects and classes go hand in hand. We can't talk about one without talking about the other. And the entire point of Object-Oriented Design is not about objects, it's about classes, because we use classes to create objects. So, a class describes what an object will be, but it isn't the object itself.

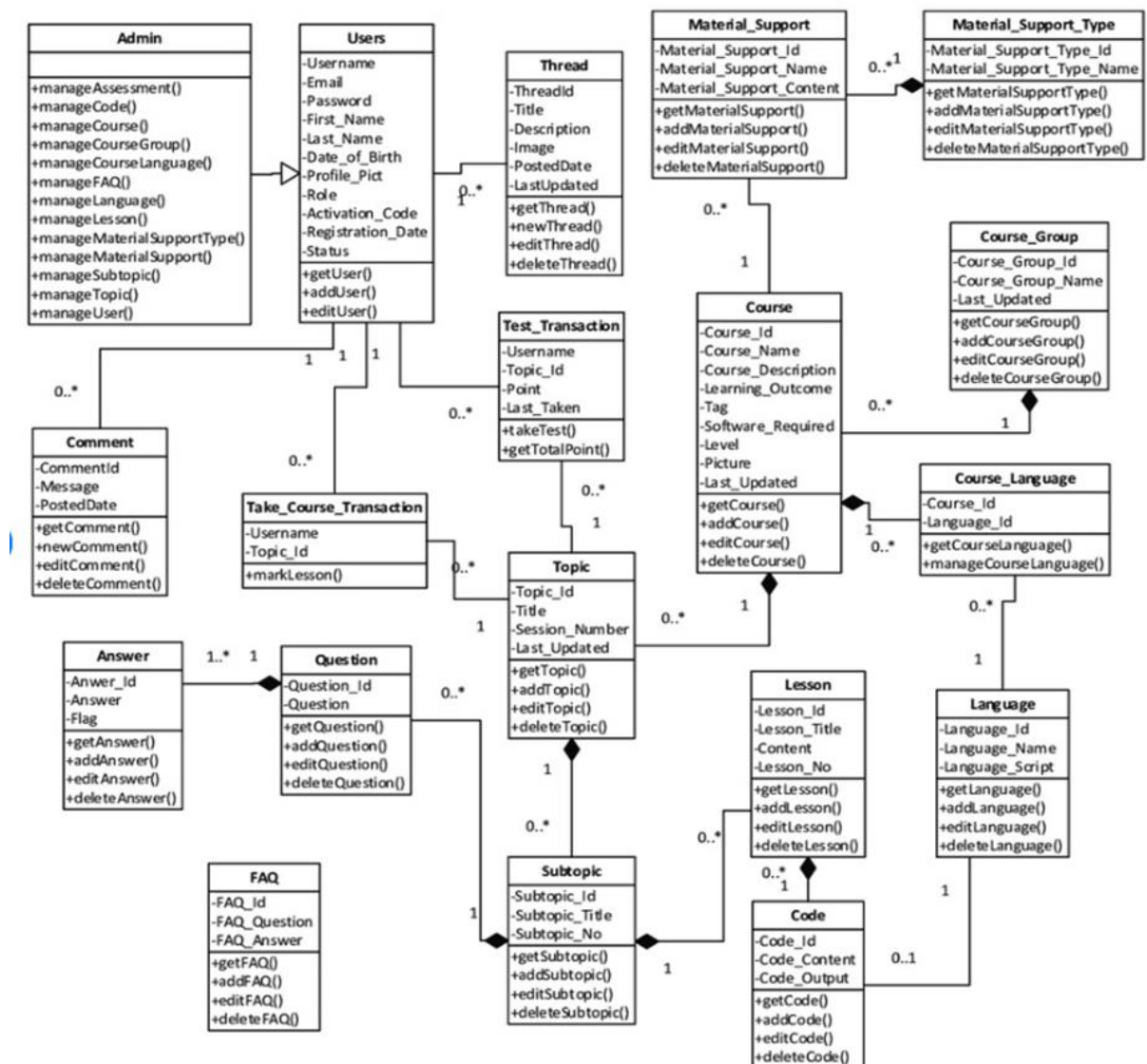


Figure - 6 Class Diagram

### Users:

**Admin:** This class likely represents administrators of the platform with permissions to manage all aspects of the system, including users, courses, materials, and assessments.

**Users:** This class represents the general users of the platform, including students and instructors. Users can have different roles with varying privileges within the system.

## **Content Management:**

**Course:** This class represents a learning course offered on the platform. It likely has attributes like course name, description, learning outcomes, and a pointer to the language the course is offered in.

**Topic:** This class represents a topic within a course. A course likely consists of multiple topics.

**Subtopic:** This class represents a subtopic within a topic, further dividing the course content into smaller manageable units.

**Lesson:** This class represents a lesson within a topic. A topic likely consists of multiple lessons which may include text, videos, or other educational content.

**FAQ (Frequently Asked Questions):** This class represents a collection of frequently asked questions and answers related to a course, topic, or subtopic.

## **Learning Activities:**

**Test\_Transaction:** This class likely represents a record of a user taking a test or assessment within a course. It likely has attributes capturing the username, course, and the test score achieved.

**Code:** This class might represent code snippets or programming exercises included as part of a course or lesson.

## **Material Support:**

**Material\_Support\_Type:** This class represents the different categories of material support offered by the platform, such as videos, documents, or cheat sheets.

## **Communication:**

**Comment:** This class represents a comment posted by a user on a forum, discussion board, or a course topic.

**Thread:** This class likely represents a discussion thread within a forum or course topic where users can post comments and replies.

**Message:** This class might represent a messaging system within the platform allowing users to communicate with each other.

## **User Management:**

**Course\_Group:** This class likely represents a group of users enrolled in a particular course together. It might be used to manage group assignments or discussions.

## **Additional Functionalities:**

The class diagram also suggests functionalities like managing user accounts, adding new courses, editing course content, taking tests, and adding comments to discussions.

## **Relationships:**

The diagram depicts various relationships between the classes. For instance, a course can have many topics, and a topic can consist of multiple lessons. Similarly, a user can enroll in multiple courses, and a course can have many enrolled users.

This is a high-level overview of the class diagram for the e-learning platform. The specific implementation details would depend on the specific platform's design and functionalities.

## 9. ACTIVITY DIAGRAM

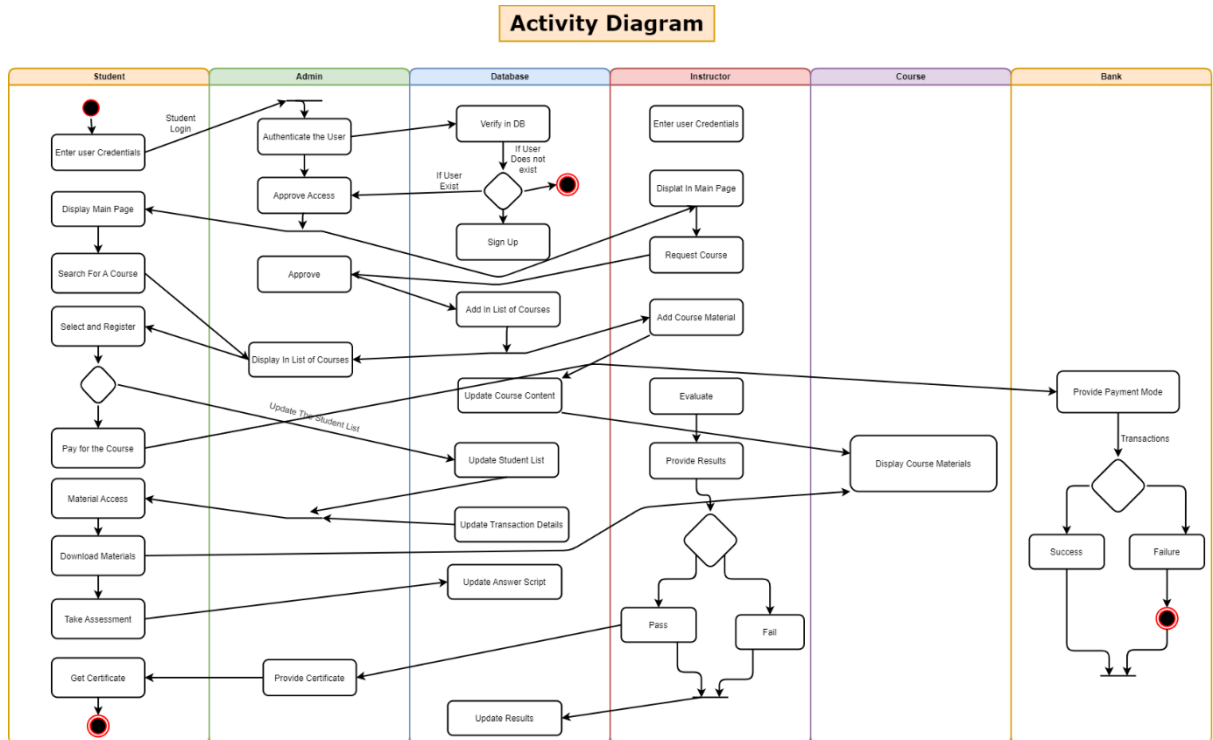


Figure - 7 Activity Diagram of Enterprise e-Learning Platform

## Elements:

- Rectangles: Represent activities or actions performed within the system.
- Rounded rectangles: Represent concurrent activities that can happen simultaneously.
- Diamonds: Represent decision points where the flow of the process can branch based on a condition.
- Arrows: Represent the flow of the process between activities, decisions, and other elements.
- Swimlanes (optional): Can be used to group activities based on the role or component that performs them (not visible in this case).

### Possible Interpretations:

- **Activities:** The rectangles might represent steps involved in a process, such as "Enter User Credentials," "Authenticate User," "Display Main Page," etc.
- **Decisions:** The diamonds could indicate branching based on conditions, such as "Login Successful?" or "Valid Request?"

- Flows: The arrows show the sequence of steps and how the process progresses based on decisions.

### **User Management:**

- The process starts with users (likely instructors and students) creating accounts on the LMS platform (10-00).
- Authentication is likely depicted by "Enter User Credentials -> Authenticate the User -> Approve Access". If successful, this leads to displaying the main page of the LMS.

### **Learner Activities:**

- Once logged in, learners can search for courses using the "Search For A Course" function.
- After enrolling in a course, the diagram shows learners accessing course materials through "Display Course Materials -> Material Access -> Download Materials".

### **Instructor Activities:**

- While the instructor activities aren't extensively detailed, the initial step (10-00) likely encompasses instructors creating their accounts as well.
- The possibility of instructors adding course materials is indirectly implied by the learners' ability to access them.

## 10.STATE CHART DIAGRAM

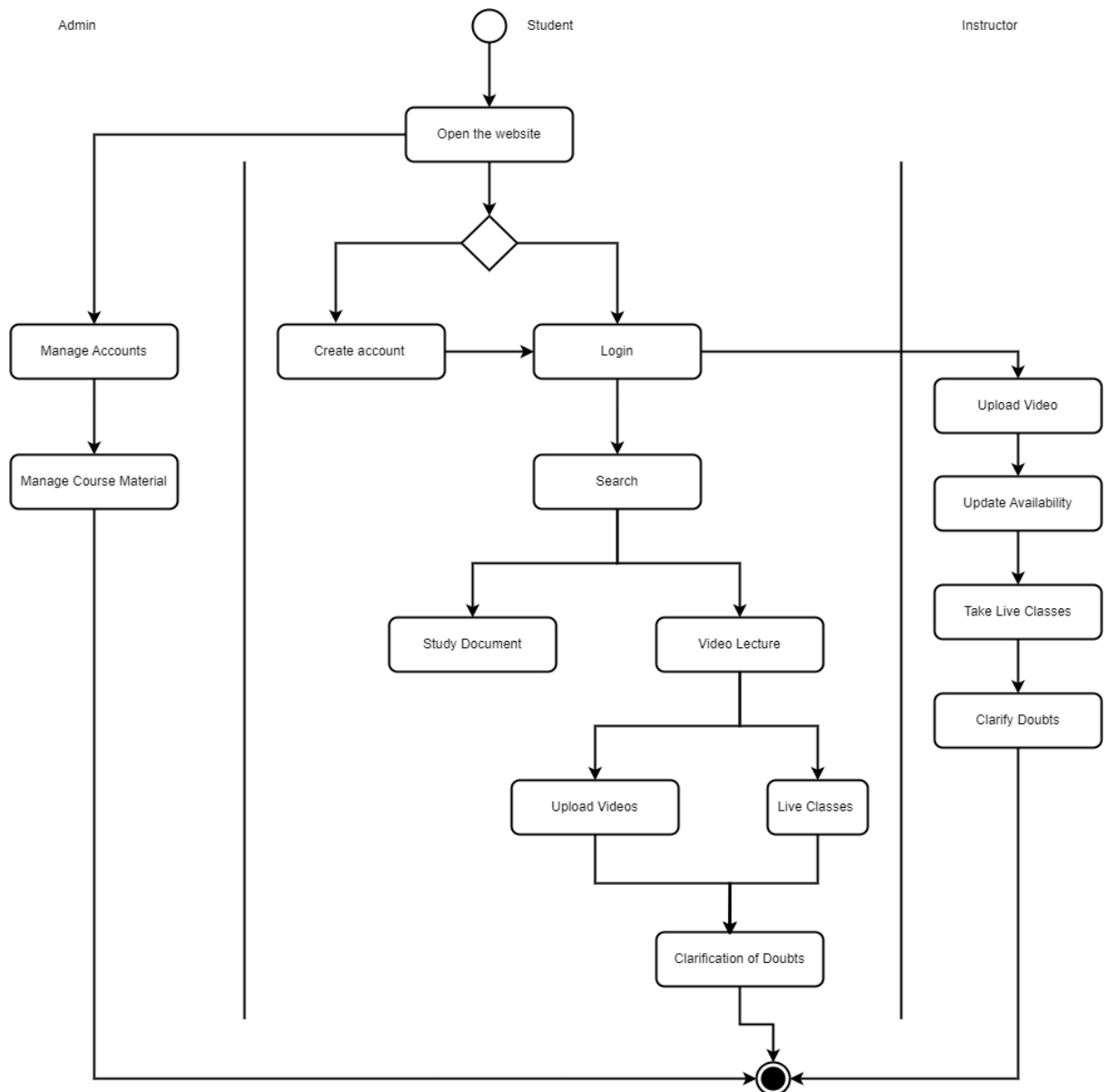


Figure - 8 State Chart Diagram of Enterprise e-Learning Platform

### Instructor

- Create an account on the LMS platform
- Manage accounts (possibly create student accounts or assign roles)
- Upload videos
- Upload course materials (documents)
- Manage course material (update availability, etc.)
- Clarify doubts (presumably answer student questions)
- Conduct live classes



## Student

- Create an account on the LMS platform
- Login to the LMS platform
- Search for courses
- Take live classes
- Study documents (presumably course materials uploaded by instructors)
- Watch video lectures (presumably uploaded by instructors)
- Clarify doubts (presumably ask questions of instructors)

## User Workflow:

- **Account Creation:** Learners typically create an account to enroll in courses on the LMS platform.
- **Course Registration:** Learners may browse a course catalog to find relevant courses. Enrollment options may vary - students might self-enroll in open courses, or instructors or administrators may assign them to specific courses.
- **Accessing Course Materials:** Once enrolled, learners can access the course content according to the structure defined by the instructor. This includes: Watching video lectures, Reading documents and presentations, Completing assignments and quizzes, Participating in discussion forums
- **Taking Assessments:** Learners can take quizzes, exams, and other assessments created by the instructor. The LMS may provide immediate feedback on results or allow instructors to review and provide personalized feedback later.
- **Tracking Progress:** Learners can typically view their progress through the course, including completed modules, grades on assessments, and overall completion percentage.

## 11.COLLABORATIVE DIAGRAM

### 11.1 Collaborative Diagram For Admin

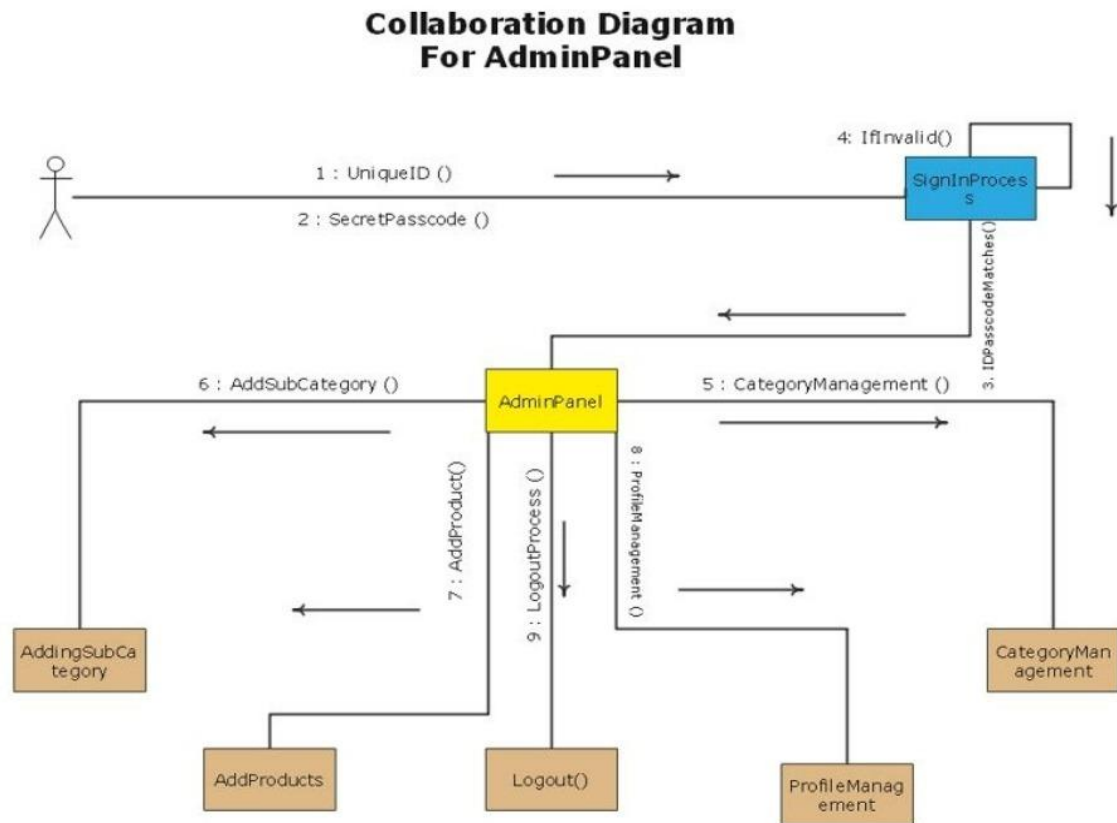


Figure -9 Collaborative Diagram for Admin of Enterprise e-Learning Platform

- **AdminPanel:** An AdminPanel is the graphical user interface (GUI) that allows administrators to manage a website or application. It typically provides features for creating and editing user accounts, managing content, and configuring settings.
- **UniqueID :** This refers to a unique identifier, which is a code that is used to identify a specific user or piece of data. In the context of the diagram, it likely refers to the username or ID that the admin uses to sign in to the AdminPanel.
- **SecretPasscode:** This refers to a secret passcode, which is a confidential code that is used to verify the identity of the admin. This is typically used in conjunction with the UniqueID to sign in to the AdminPanel.
- **SigninProcess:** This refers to the process of signing in to the AdminPanel. It involves the admin entering their UniqueID and SecretPasscode, and the system verifying that this information is correct.

- **IDPasscodeMatches** : This term checks whether the entered UniqueID and SecretPasscode match the information stored in the system. If they do match, the admin is granted access to the AdminPanel.
- **Category Management**: This refers to the functionality that allows admins to manage categories. This might include adding, editing, or deleting categories.
- **AddSubCategory**: This refers to the function that allows admins to add subcategories to existing categories. Subcategories are used to further organize content within a category.
- **AddingSubCategory**: This part of the diagram shows the action of adding a subcategory. It likely involves the admin selecting the parent category and then entering the name of the new subcategory.
- **AddProducts**: This refers to the functionality that allows admins to add products to the system. This might involve entering product details such as the name, description, price, and images.
- **AddProduct**: This part of the diagram shows the action of adding a product. It likely involves the admin entering the details of the new product.
- **Logout Process**: This refers to the process of logging out of the AdminPanel. This typically involves ending the admin's session and returning them to the login screen.
- **Logout()**: This refers to the function that initiates the logout process.
- **ProfileManagement**: This refers to the functionality that allows admins to manage their profiles. This might include editing their contact information or changing their password.
- **ProfileManagement()**: This refers to the function that allows admins to access their profile management options.

## 11.2 Collaborative Diagram For User

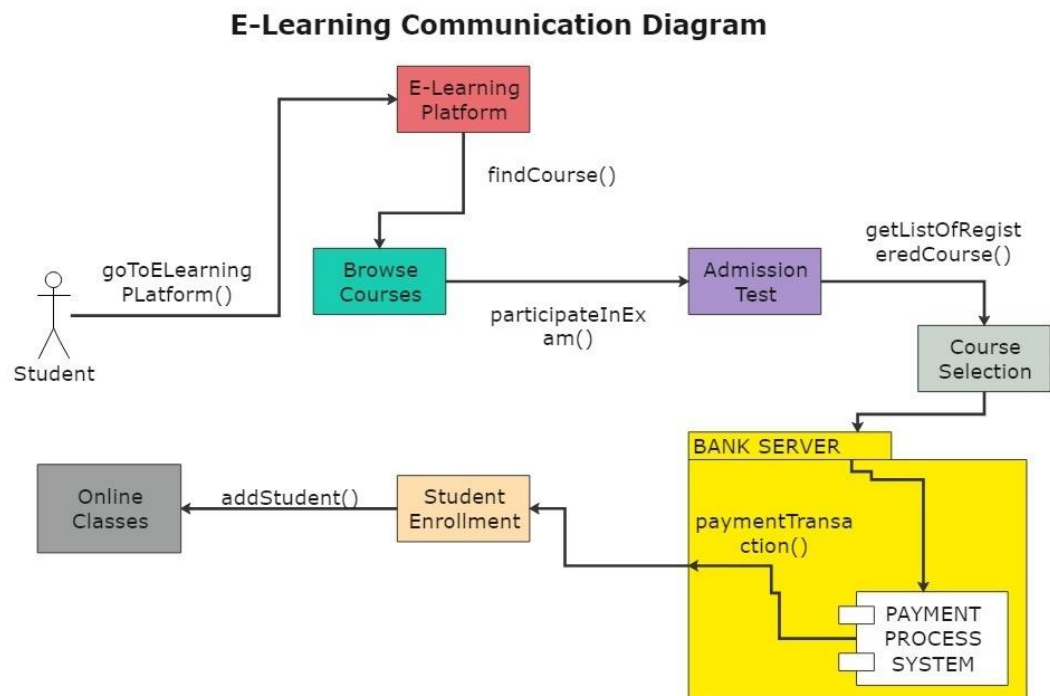


Figure - 10 Collaborative Diagram For User of Enterprise e-Learning Platform

- **Student:** This represents the users who enroll in courses and learn from the elearning platform.
- **Course Selection:** This represents the process where students browse and select courses they want to enroll in.
- **Browse Courses:** This functionality allows students to browse the available courses offered by the platform.
- **getListOfRegisteredCourse():** This function retrieves a list of courses that the student is registered for.
- **findCourse():** This function allows students to search for specific courses based on keywords or criteria.
- **Admission Test:** This could represent an entrance test that students need to take before enrolling in certain courses.
- **Online Classes:** This represents the online learning environment where students attend classes, lectures, or access course materials.

- **participateInExam():** This function allows students to participate in online exams or assessments.
- **Course:** This represents the learning modules or programs offered by the platform.
- **Enrollment:** This represents the process where students register for a course and gain access to the learning materials.
- **addStudent():** This function adds a new student to the system.
- **Student Enrollment:** This part of the diagram shows the student enrolling in a course.
- **Payment Process System:** This refers to the external system that handles student fee payments.
- **paymentTransaction():** This function initiates a payment transaction with the payment gateway.
- **Bank Server:** This represents the bank's server that authorizes and processes the student's payment.

## 12.EXECUTION SCREENSHOTS

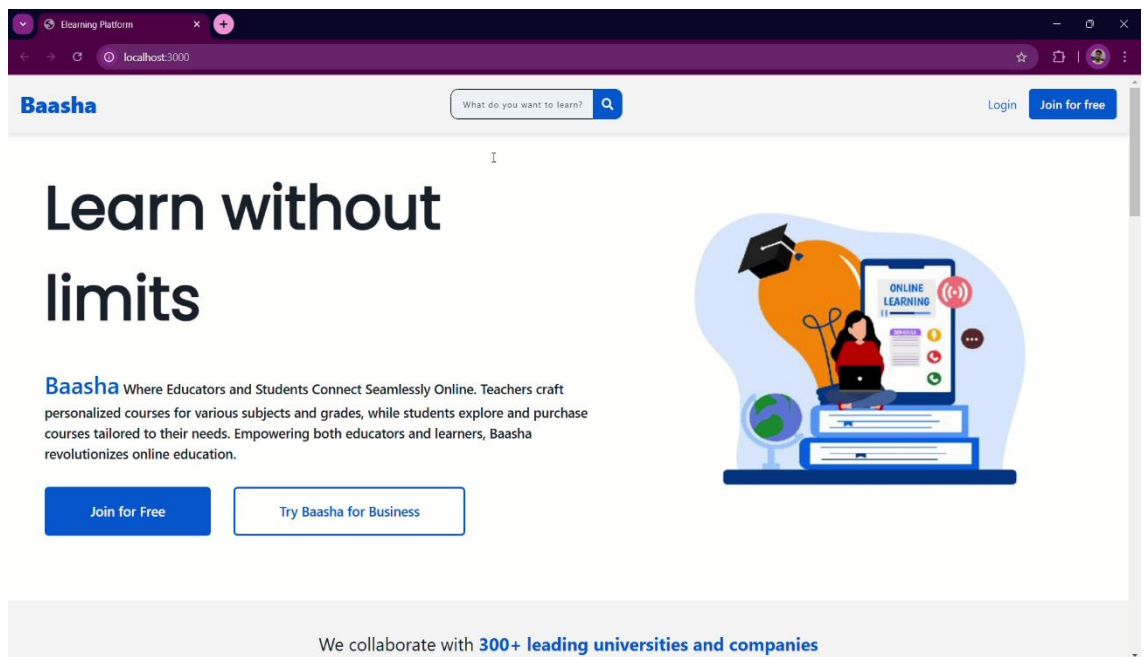


Figure - 11 Home Page in Enterprise e-Learning Platform

The screenshot shows the Baasha SignUp Page in a web browser. The browser's address bar displays 'localhost:3000/signup'. The page features a purple header with the Baasha logo, a search bar with the placeholder text 'What do you want to learn?', and links for 'Login' and 'Join for free'. The main content area has a heading 'Sign up and start learning' and a form with the following fields: 'Name' (with the value 'asd'), 'Email' (with the value 'asd@gmail.com'), 'Password' (with a masked value '.....'), and 'Confirm Password' (with a masked value '.....'). Below the form is a checkbox labeled 'Send me special offers, personalized recommendations, and learning tips.' and a 'Sign Up' button. At the bottom of the page, a footer states 'You already have Account with us? Login'.

Figure - 12 SignUp Page in Enterprise e-Learning Platform

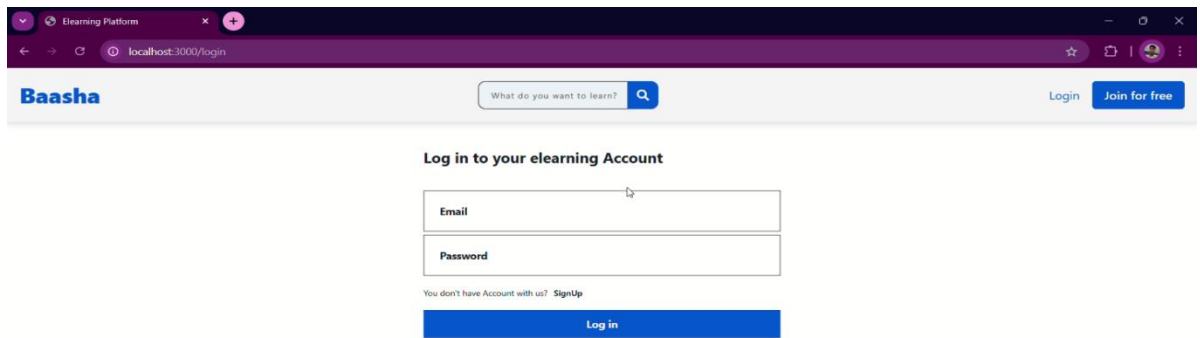


Figure - 13 Login Page in Enterprise e-Learning Platform

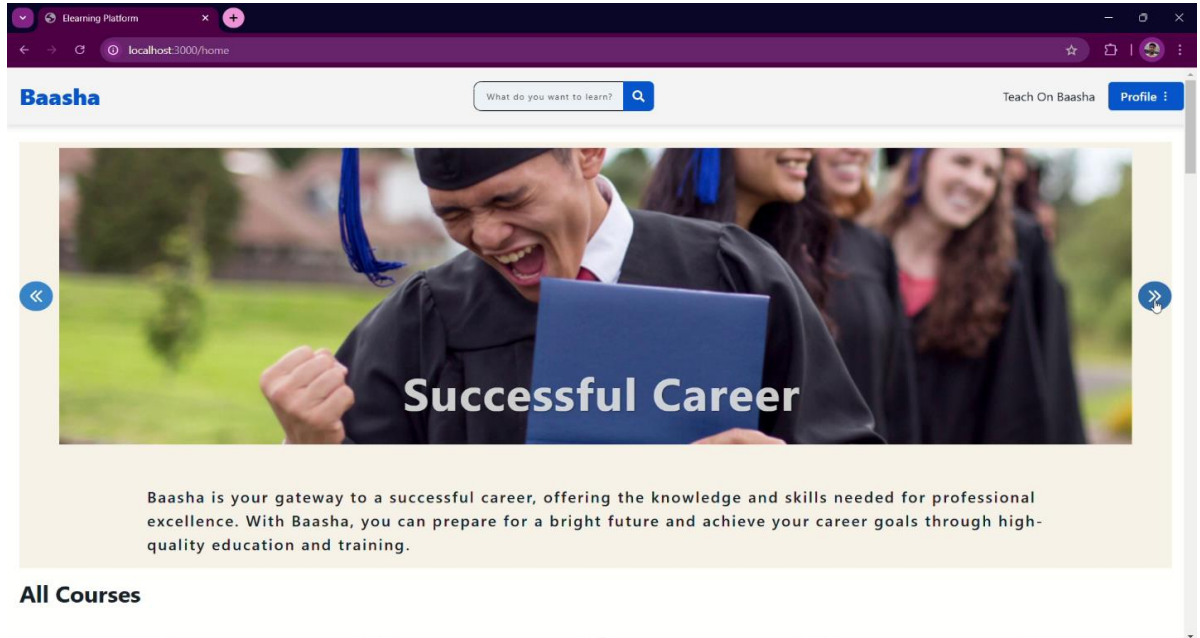


Figure - 14 User Dashboard in Enterprise e-Learning Platform

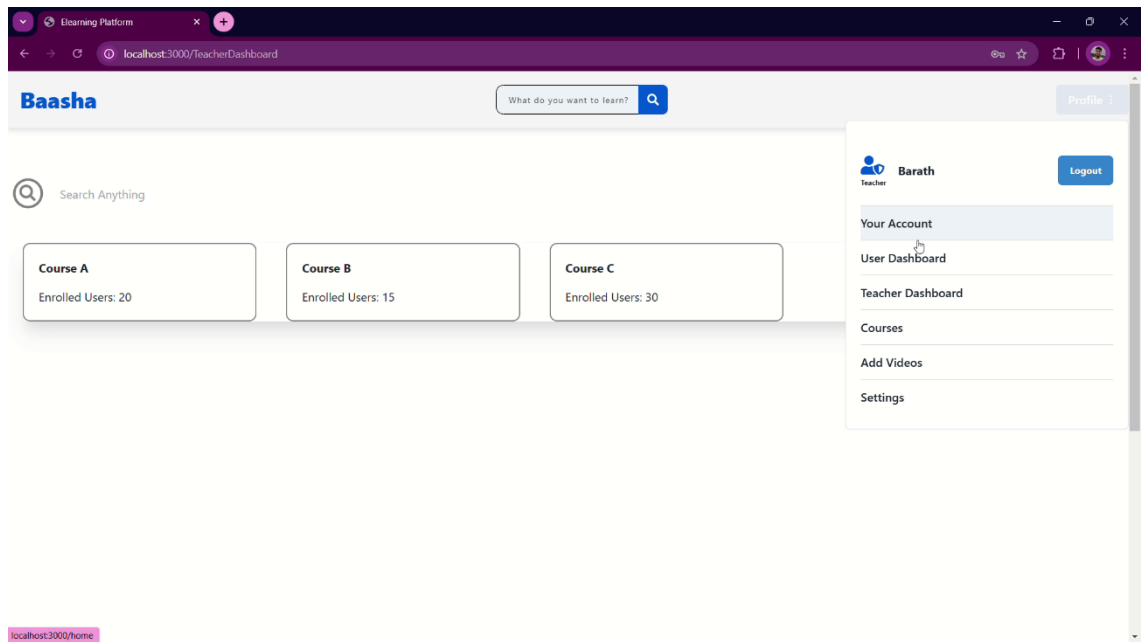


Figure - 15 Users Show to Admin in Enterprise e-Learning Platform

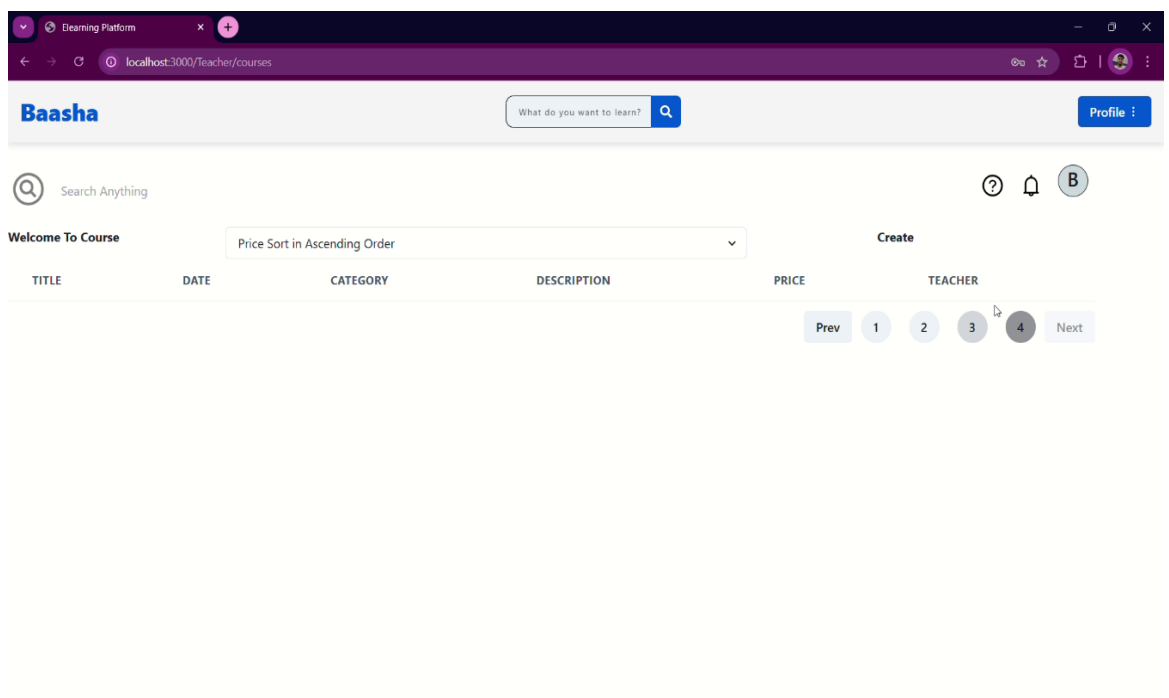


Figure - 16 All Course Shown to Admin in Enterprise e-Learning Platform



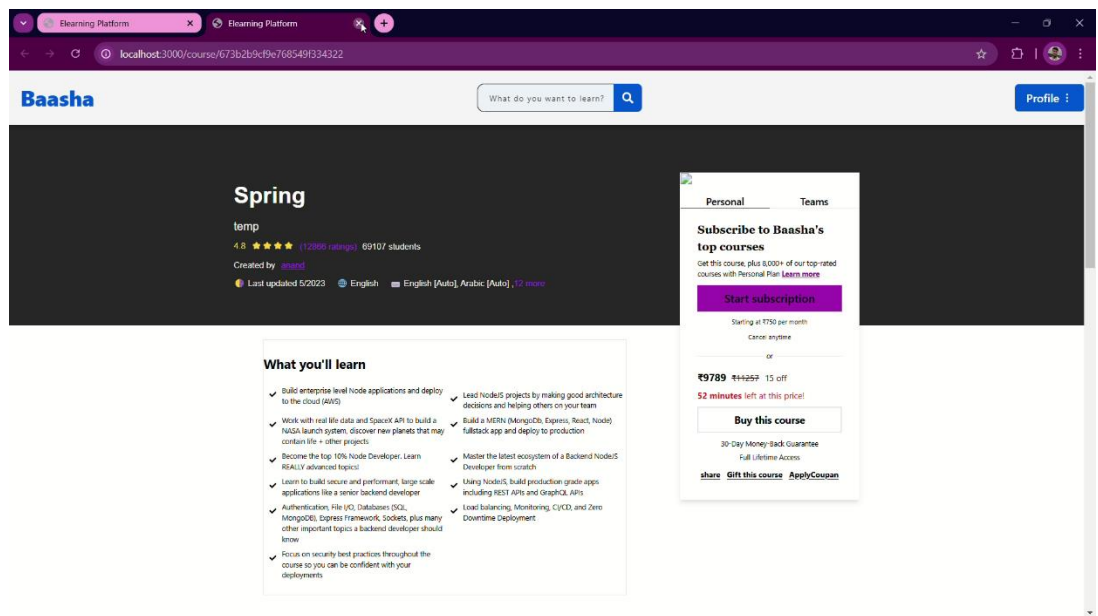


Figure - 17 Course OverView in Enterprise e-Learning Platform

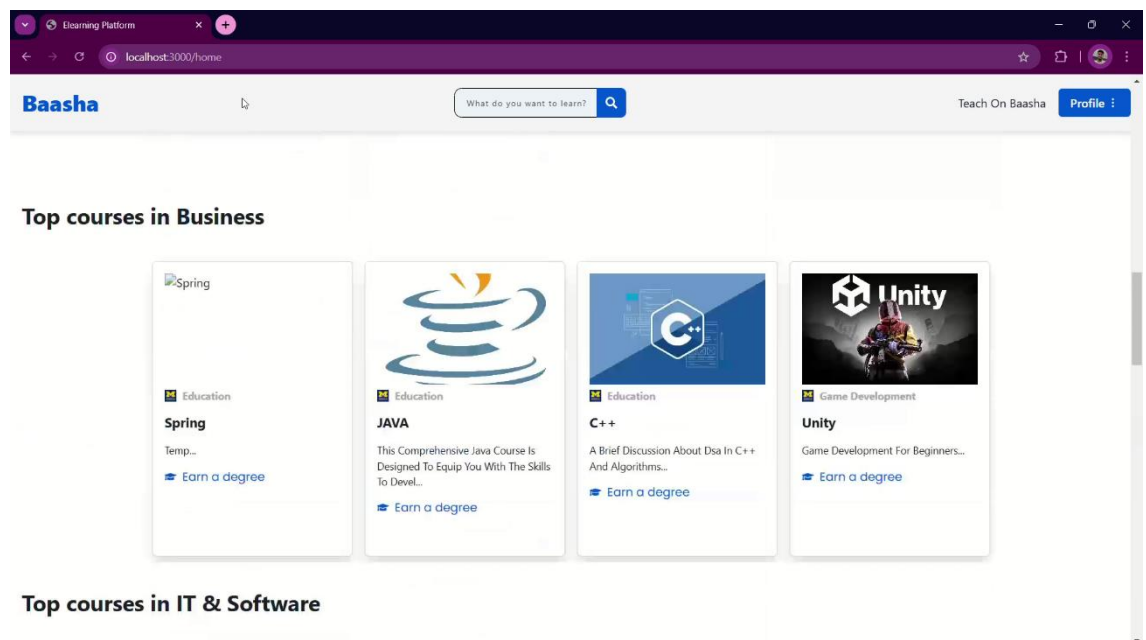


Figure - 18 All Courses Available in Enterprise e-Learning Platform

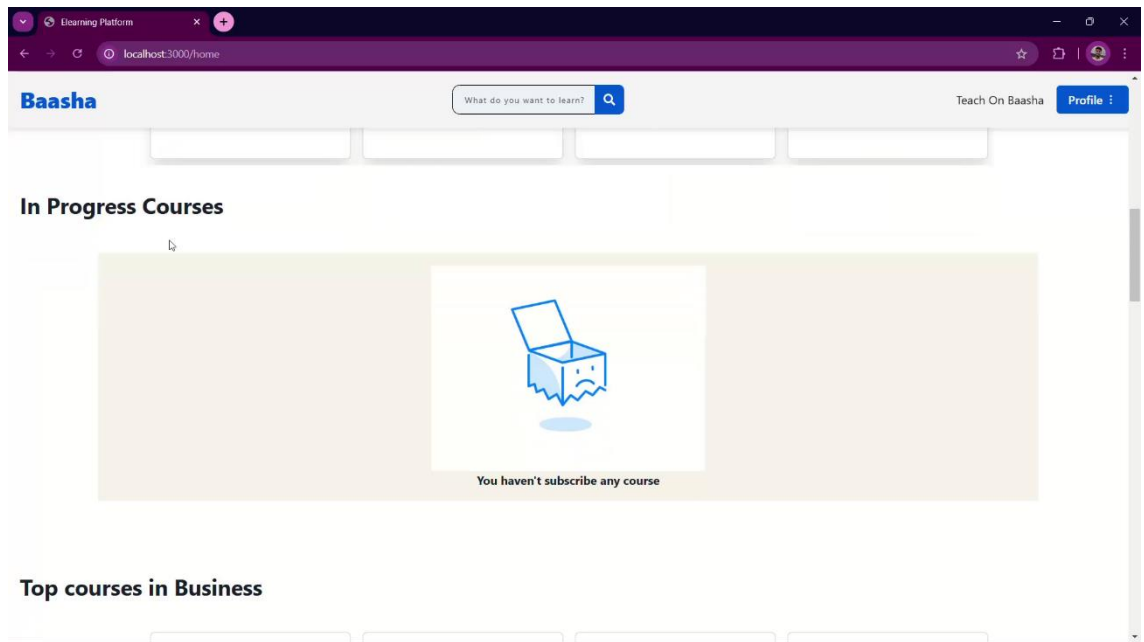


Figure - 19 User Enroll Courses in Enterprise e-Learning Platform

## **13. TECHNOLOGIES USED**

**Front-end :** HTML, CSS, JavaScript, ReactJS

**Back-end :** NodeJS, ExpressJS

**DataBase :** MongoDB

## **14.DESCRPTION ABOUT FRONT-END AND BACK-END**

### **Frontend Development:**

#### **What is it?**

Frontend development is all about building the parts of a website or web application that users interact with directly. It's like the front door of a house – it's what people see and use.

#### **What it involves:**

- Design: Creating the layout, colors, and styles that make the website look good.
- Interactivity: Adding buttons, forms, and other elements that users can click on or interact with.
- Responsiveness: Making sure the website works well on different devices like phones, tablets, and computers.
- Performance: Optimizing the website so it loads quickly and runs smoothly.

#### **Tools and Technologies:**

- HTML: For structuring the content of the website.
- CSS: For styling the appearance of the website.
- JavaScript: For adding interactivity and dynamic behavior to the website.

### **Backend Development:**

#### **What is it?**

Backend development is like the engine of a car – it's what powers everything behind the scenes. It handles the data and logic that make a website or web application work.

#### **What it involves:**

- Server-Side Logic: Writing code that runs on the server and handles requests from the frontend.
- Database Management: Storing and retrieving data from a database.
- Security: Protecting the website from hackers and other security threats.

- Performance Optimization: Making sure the website can handle a lot of traffic without slowing down.

### **Tools and Technologies:**

- Node.js: A runtime environment that allows you to run JavaScript code on the server.
- Express.js: A web application framework for Node.js, used to build web servers and handle HTTP requests.
- Databases: Such as MongoDB, MySQL, or PostgreSQL, for storing and managing data.
- Security Libraries: Like bcrypt for encrypting passwords and Helmet for securing HTTP headers.

## 15.TESTING

Unit

Testing

UNIT TESTING				
SIGN UP/SIGN IN				
Frontend				
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC-FE-SU-001	Render signup form without crashing	Signup form renders successfully	Signup form rendered without errors	Pass
TC-FE-SU-002	Check if all input fields are rendered correctly	All input fields (Name, Email, Password, Confirm Password) are visible	All input fields rendered correctly	Pass
TC-FE-SU-003	Check if input fields are empty initially	All input fields are empty initially	All input fields are empty initially	Pass
TC-FE-SU-004	Test input field functionality (typing, focus, blur)	Input fields respond correctly to user actions	Input fields function as expected	Pass
TC-FE-SU-005	Test show/hide password functionality	Password field toggles visibility correctly	Password field visibility toggles correctly	Pass
TC-FE-SU-006	Test checkbox functionality for promotion	Checkbox state toggles correctly	Checkbox state toggles correctly	Pass
TC-FE-SU-007	Validate signup form submission with all fields filled	Form submits successfully with all fields filled	Form submits successfully with all fields filled	Pass
TC-FE-SU-008	Validate signup form submission with password mismatch	Form displays error message for password mismatch	Form displays error message for password mismatch	Pass
TC-FE-SU-009	Validate signup form submission with password length < 8	Form displays error message for short password	Form displays error message for short password	Pass
TC-FE-SU-010	Validate navigation to login page after successful signup	User navigates to login page after successful signup	User navigates to login page after successful signup	Pass
Backend				
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC-BE-SU-001	Test authentication middleware	Verify if the user is authenticated	User is authenticated	Pass
TC-BE-SU-002	Verify if token is present in request headers	Token is present in request headers	Token is present in request headers	Pass
TC-BE-SU-003	Verify if token is blacklisted	Token is not blacklisted	Token is not blacklisted	Pass
TC-BE-SU-004	Verify if token is decoded properly	Token is decoded correctly	Token is decoded correctly	Pass
TC-BE-SU-005	Verify if decoded token contains necessary data	Token contains username, userID, and role	Token contains username, userID, and role	Pass

Table - 5 Unit Testing on Signup/SignIn

Add Courses				
Frontend				
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC-FE-001	Form renders without crashing	Form renders successfully	Form rendered without errors	Pass
TC-FE-002	All form fields are rendered	All form fields are visible	All form fields are visible	Pass
TC-FE-003	Input fields are empty initially	All input fields are empty initially	All input fields are empty	Pass
TC-FE-004	Input fields update correctly when typing	Input fields update with typed values	Input fields update correctly	Pass
TC-FE-005	Submit button triggers handleSubmit function	handleSubmit function is triggered	handleSubmit triggered	Pass
TC-FE-006	addProduct function is called on submit	addProduct function is called	addProduct function called	Pass
TC-FE-007	navigate function is called after submission	navigate function is called	navigate function called	Pass
TC-FE-008	Alert is shown after successful submission	Alert is displayed	Alert displayed	Pass
TC-FE-009	handleChange function updates state correctly	State is updated correctly	State updated correctly	Pass
TC-FE-010	Correct components rendered based on auth	Components rendered based on auth status	Components rendered correctly	Pass
Backend				
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC-BE-001	/all route returns status code 200	Status code 200 returned	Status code 200 returned	Pass
TC-BE-002	/ route returns status code 200	Status code 200 returned	Status code 200 returned	Pass
TC-BE-003	/ route returns correct number of courses	Correct number of courses returned	Correct number of courses returned	Pass
TC-BE-004	/courseID route returns status code 200	Status code 200 returned	Status code 200 returned	Pass
TC-BE-005	/courseID route returns correct course	Correct course returned	Correct course returned	Pass
TC-BE-006	/add route returns status code 201	Status code 201 returned	Status code 201 returned	Pass
TC-BE-007	/update/:courseID route returns status code 202	Status code 202 returned	Status code 202 returned	Pass
TC-BE-008	/delete/:courseID route returns status code 200	Status code 200 returned	Status code 200 returned	Pass
TC-BE-009	Unauthorized users cannot access protected routes	Access denied for unauthorized users	Access denied for unauthorized users	Pass
TC-BE-010	Routes handle errors appropriately	Status code 400 returned for errors	Status code 400 returned for errors	Pass

Table - 6 Unit Testing on Adding Courses

Add Users				
Frontend				
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC-BE-UR-001	Test if the endpoint to get all users returns data only for admin role	Endpoint returns data only if the authenticated user is an admin	Endpoint returns data only if the authenticated user is an admin	Pass
TC-BE-UR-002	Test if the user registration endpoint creates a new user	New user is created successfully	New user is created successfully	Pass
TC-BE-UR-003	Test if the login endpoint authenticates users with correct credentials	Users with correct credentials are authenticated successfully	Users with correct credentials are authenticated successfully	Pass
TC-BE-UR-004	Test if the update endpoint updates user information correctly	User information is updated successfully	User information is updated successfully	Pass
TC-BE-UR-005	Test if the delete endpoint deletes users only if the requester is an admin	Users are deleted only if the requester is an admin	Users are deleted only if the requester is an admin	Pass
TC-BE-UR-006	Test if the logout endpoint adds tokens to the blacklist	Tokens are added to the blacklist upon logout	Tokens are added to the blacklist upon logout	Pass
TC-BE-UR-007	Test if the endpoint to get user courses returns courses for a specific user	Endpoint returns courses for a specific user	Endpoint returns courses for a specific user	Pass
TC-BE-UR-008	Test if the endpoint to add a course to a user updates the user's course list	Course is added to the user's course list successfully	Course is added to the user's course list successfully	Pass
TC-BE-UR-009	Test if the endpoint to upgrade a user's role to teacher works correctly	User's role is upgraded to teacher successfully	User's role is upgraded to teacher successfully	Pass
Backend				
Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
TC-BE-UM-001	Test if user model is created properly	User model is created with the specified fields	User model created with the specified fields	Pass
TC-BE-UM-002	Test if user model schema has the correct field types	User model schema has correct field types	User model schema has correct field types	Pass
TC-BE-UM-003	Test if user model schema has the correct default role	User model schema has default role as 'user'	User model schema has default role as 'user'	Pass
TC-BE-UM-004	Test if user model schema version key is set to false	User model schema version key is set to false	User model schema version key is set to false	Pass

Table - 7 Unit Testing on Adding Users



## Functional Point Analysis

FUNCTIONAL POINT TESTING					
SIGN UP/SIGN IN					
Measurement Parameter	Count	Weighing Factor	Total Count (Simple)	Total Count (Average)	Total Count (Complex)
Number of external inputs (EI)	7	$7 * 4 = 28$	3	4	6
Number of external outputs (EO)	1	$1 * 4 = 4$	4	5	7
Number of external inquiries (EQ)	0	$0 * 4 = 0$	3	4	6
Number of internal files (ILF)	2	$2 * 15 = 30$	7	10	15
Number of external interfaces (EIF)	6	$6 * 10 = 60$	5	7	10
Total: $28 + 4 + 0 + 30 + 60 = 122$					

Table - 8 Functional Point Analysis on Sign Up/Sign In

ADD COURSES					
Measurement Parameter	Count	Weighing Factor	(Simple)	(Average)	(Complex)
Number of external inputs (EI)	5	$5 * 6 = 30$	3	4	6
Number of external outputs (EO)	1	$1 * 5 = 5$	4	5	7
Number of external inquiries (EQ)	1	$1 * 4 = 4$	3	4	6
Number of internal files (ILF)	1	$1 * 15 = 15$	7	10	15
Number of external interfaces (EIF)	3	$3 * 5 = 15$	5	7	10
Total: $30 + 5 + 4 + 15 + 15 = 69$					

Table - 9 Functional Point Analysis on Add Courses

ADD USERS					
Measurement Parameter	Count	Weighing Factor	Total Count (Simple)	Total Count (Average)	Total Count (Complex)
Number of external inputs (EI)	7	$7 * 3 = 21$	3	4	6
Number of external outputs (EO)	1	$1 * 7 = 7$	4	5	7
Number of external inquiries (EQ)	3	$3 * 4 = 12$	3	4	6
Number of internal files (ILF)	2	$2 * 7 = 14$	7	10	15
Number of external interfaces (EIF)	0	$0 * 7 = 0$	5	7	10
Total: $21 + 7 + 12 + 14 + 0 = 54$					

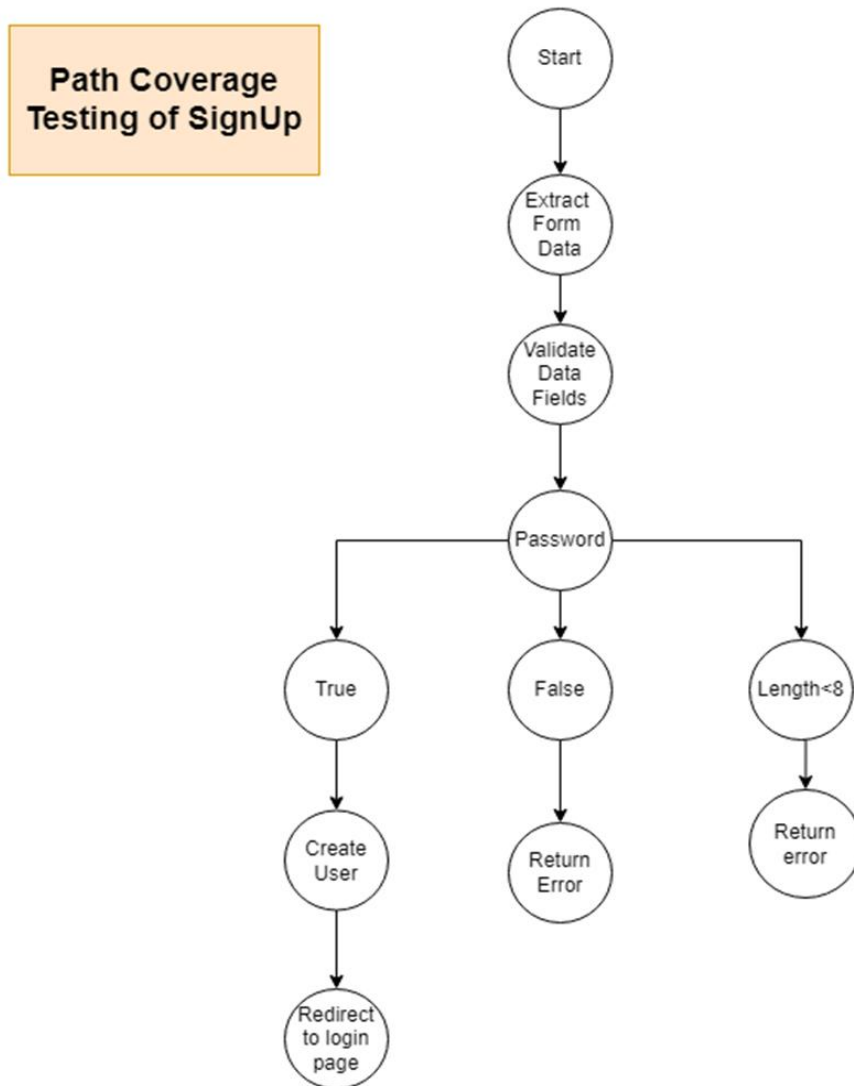
Table - 10 Functional Point Analysis on Add Users

## Path Coverage Testing

Path coverage testing is a structural testing technique used in software engineering to ensure that all possible execution paths through a program are tested at least once. The goal is to evaluate the correctness and robustness of the software by examining various paths through its source code.

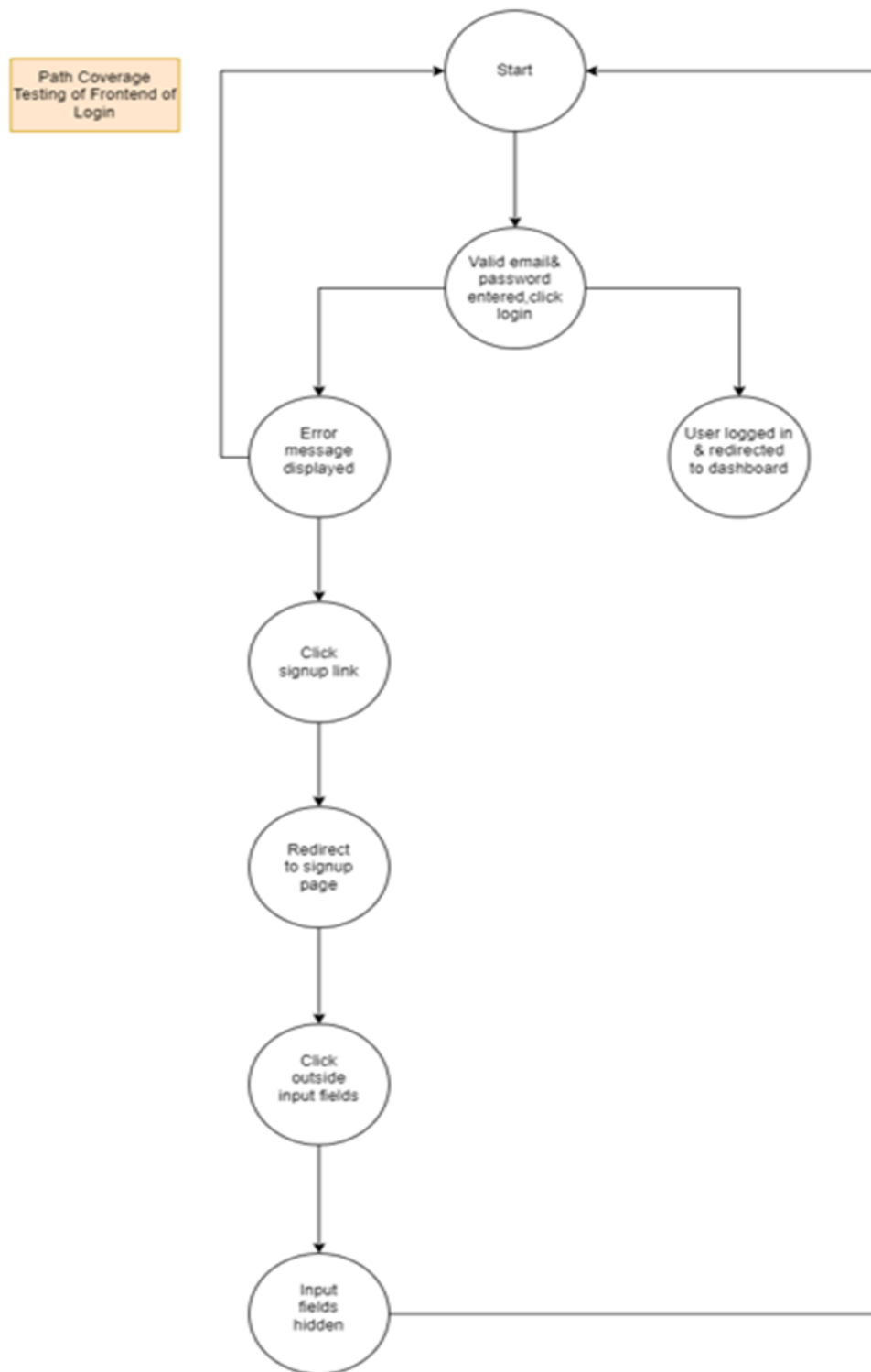
**Cyclomatic Complexity:** Cyclomatic complexity is a software metric used to measure the complexity of a program. It quantifies the number of linearly independent paths through a program's source code.

$$V(G) = E - V + 2$$



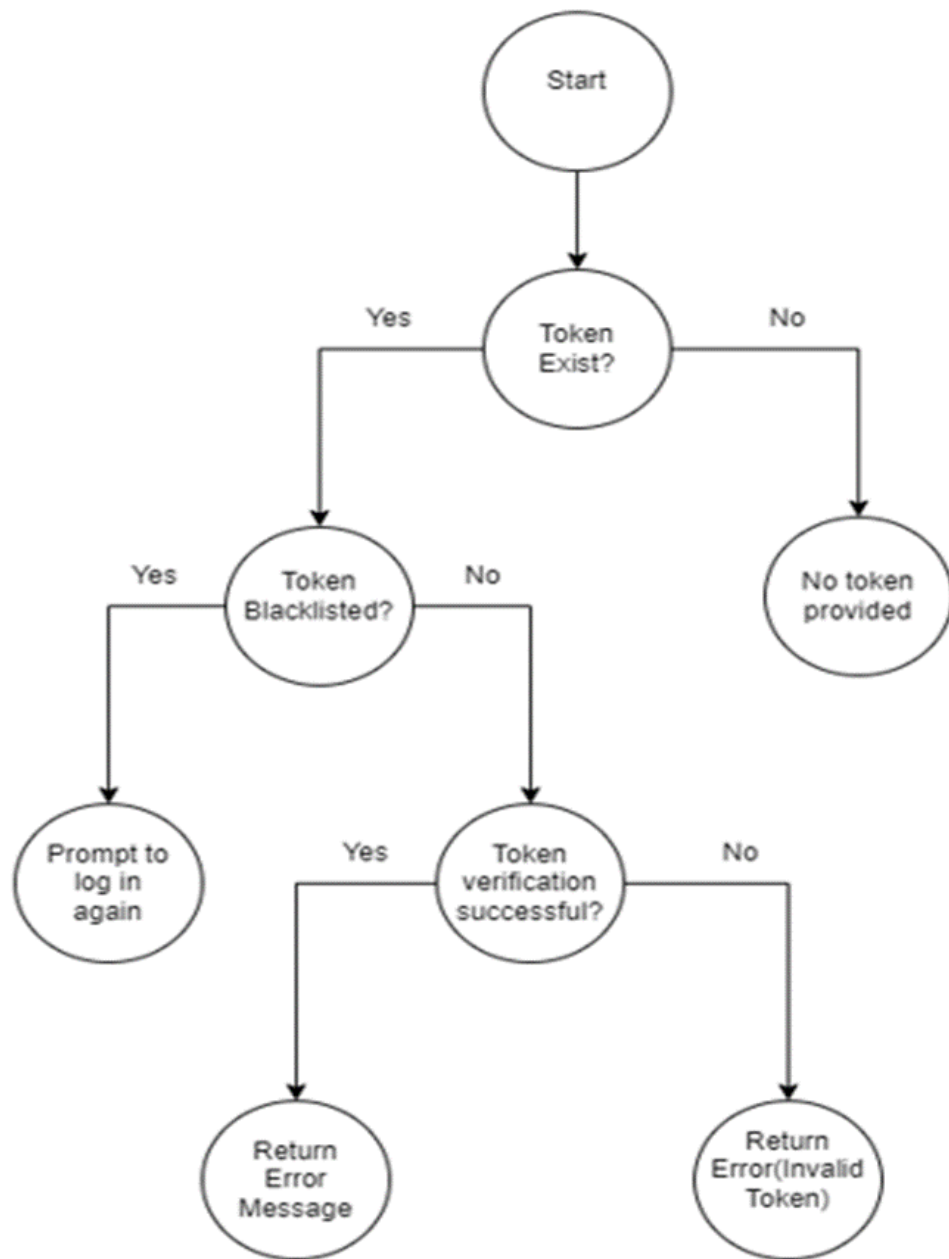
$$V(G) = 10 - 11 + 2 = 1$$

Figure - 20 Path Coverage Testing of Signup



$$V(G) = 9 - 8 + 2 = 3$$

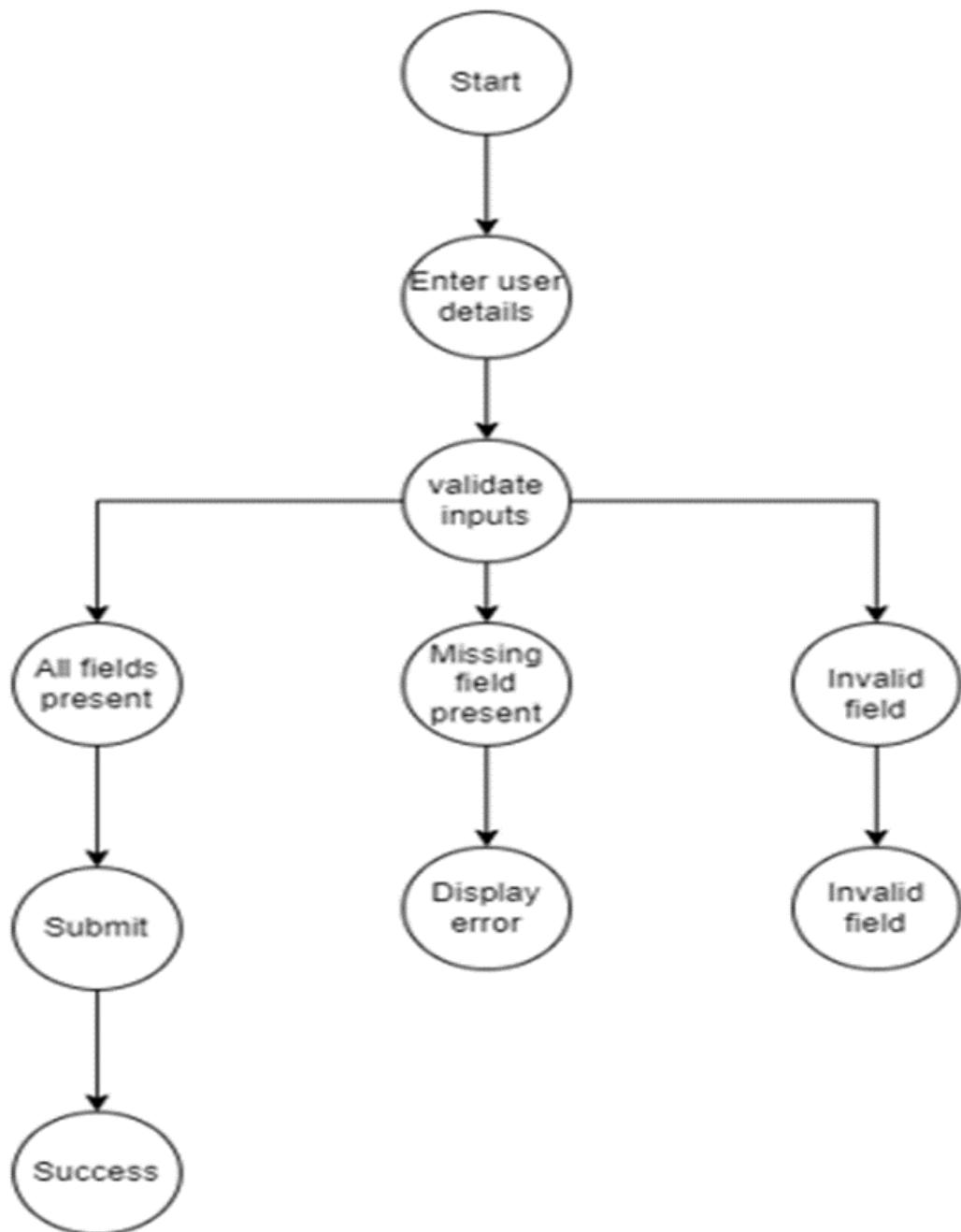
Figure - 21 Path Coverage Testing of Backend of Login



$$V(G) = 7 - 8 + 2 = 1$$

Figure - 22 Path Coverage Testing Frontend of Login

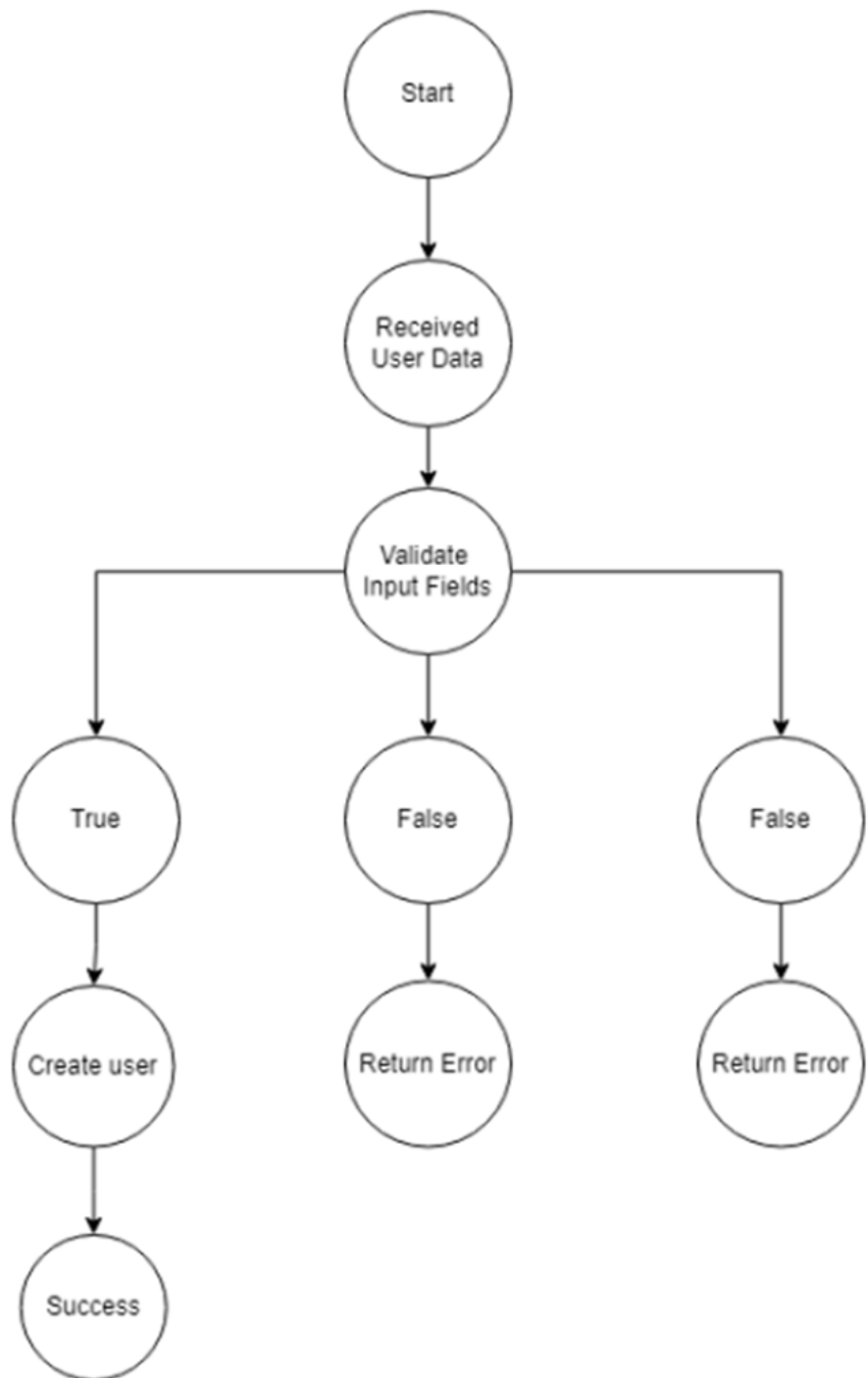
## Path Coverage Testing of Frontend of AddUser



$$V(G) = 9 - 10 + 2 = 1$$

Figure - 23 Path Coverage of Frontend of AddUser

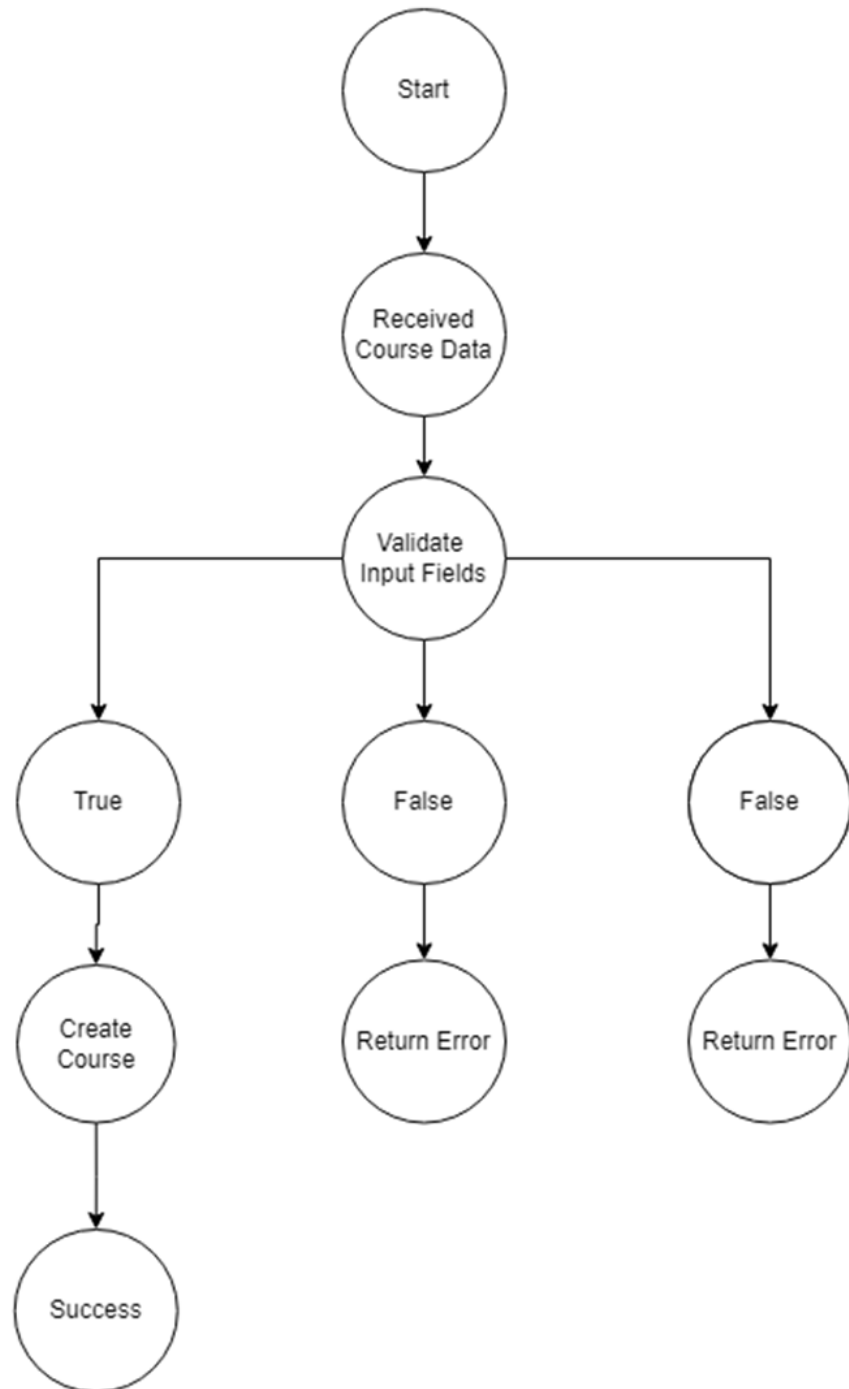
Path Coverage Testing  
of Backend of Add  
User



$$V(G) = 9 - 10 + 2 = 1$$

Figure - 24 Path Coverage Testing of Backend of Add User

Path Coverage Testing  
of Add Course



$$V(G) = 9 - 10 + 2 = 1$$

Figure - 25 Path Coverage Testing of Add Course

## **16.CONCLUSION**

In conclusion, the development of the Enterprise E-Learning Platform represents a significant milestone in addressing the challenges students face in accessing quality educational resources tailored to modern industry demands. By leveraging the power of the MERN stack - MongoDB, ExpressJS, ReactJS, and NodeJS - we have created a robust, user-friendly platform that empowers learners worldwide to unlock their full potential.

Our platform's seamless user experience, transparent enrollment process, and comprehensive tracking capabilities set it apart as an innovative solution in online education. The ability for users to conveniently browse, enroll in, and access courses, coupled with the secure payment system and prompt confirmation emails, ensures a smooth and efficient learning journey.

Moreover, the transparent tracking of user engagement provides valuable insights for administrators, enabling them to make data-driven decisions to enhance platform performance and user experience. The platform's architecture and design principles prioritize scalability, security, and usability, ensuring it can adapt to the evolving needs of both learners and organizations.



## **FUTURE WORK**

**Personalized Learning Experience:** Implementing machine learning algorithms to analyze user behaviour and preferences, allowing for personalized course recommendations and adaptive learning paths. Incorporating features such as progress tracking, quizzes, and assessments to tailor the learning experience to individual needs.

**Accessibility Features:** Implementing accessibility features such as screen reader compatibility, keyboard navigation, and alternative text for multimedia content to ensure inclusivity and compliance with accessibility standards.

## REFERENCES

- i. [www.google.co.in](http://www.google.co.in)
- ii. [www.wikipedia.com](http://www.wikipedia.com)
- iii. <https://www.learnupon.com/blog/what-is-elearning/>
- iv. Enterprise E-Learning Success Factors: An Analysis of Practitioners' Perspective
- v. <https://www.techtarget.com/whatis/definition/Web-based-training-e-learning>