

# **EDUSPHERE FUSION**

Foreign Study Management System with E-Learning

*Mini Project Report*

*Submitted by*

**VISHAL C VISWAM**

**Reg. No.: AJC22MCA-2099**

*In Partial Fulfillment for the Award of the Degree of*

**MASTER OF COMPUTER APPLICATIONS**

**(MCA TWO YEAR)**

[Accredited by NBA]

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,  
Accredited by NAAC. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2023-2024**

# **DEPARTMENT OF COMPUTER APPLICATIONS**

**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**



## **CERTIFICATE**

This is to certify that the Project report, “**EDUSPHERE FUSION**” is the bona fide work of **VISHAL C VISWAM (Regno: AJC22MCA-2099)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

**Ms. Navyamol K T**

**Ms. Meera Rose Mathew**

**Internal Guide**

**Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

## **DECLARATION**

I hereby declare that the project report “**EDUSPHERE FUSION**” is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

**Date:**

**VISHAL C VISWAM**

**KANJIRAPPALLY**

**Reg: AJC22MCA-2099**

## ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Navyamol K T** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

VISHAL C VISWAM

## **ABSTRACT**

EduSphere Fusion is an innovative online platform designed to redefine the landscape of global education. It uniquely combines a Foreign Study Management System with an e-learning platform, offering a comprehensive solution to educational needs worldwide. The platform is structured around three key modules: Admin, User, and College, each tailored to cater to the specific requirements of its users.

The Admin module is focused on managing user and college accounts, overseeing public posts, and implementing various monetization strategies. It also provides essential support, ensuring the smooth operation of the platform. The User module offers functionalities such as registration, login, profile management, and access to a vast array of online courses from top international colleges. Users can also engage with educational content, follow colleges for the latest updates, participate in public discussions, and track their educational progress. The College module allows educational institutions to manage their profiles, upload educational content, and interact with potential students.

EduSphere Fusion stands out for its integration of advanced technologies like Python Django for backend development, SQLite3 for database management, and a combination of HTML, CSS, JavaScript, and Bootstrap for the frontend, ensuring a seamless and user-friendly experience.

Looking ahead, EduSphere Fusion's future scope includes the integration of AI and ML for personalized recommendation systems, chatbot assistance, and expansion into areas like IELTS coaching. The platform also envisions the development of an interactive learning community, where users can share resources, engage in live video conferencing, and foster a collaborative learning environment.

EduSphere Fusion is not just a platform; it's a gateway to global educational opportunities, bridging educational divides and offering seamless access to top-tier educational resources. It marks a significant step forward in making learning more accessible, interactive, and efficient, positioning itself as a leader in the digital education space.

# CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	5
2.1	INTRODUCTION	6
2.2	EXISTING SYSTEM	6
2.2.1	NATURAL SYSTEM STUDIED	6
2.2.2	DESIGNED SYSTEM STUDIED	7
2.3	DRAWBACKS OF EXISTING SYSTEM	8
2.4	PROPOSED SYSTEM	9
2.5	ADVANTAGES OF PROPOSED SYSTEM	10
3	REQUIREMENT ANALYSIS	11
3.1	FEASIBILITY STUDY	12
3.1.1	ECONOMICAL FEASIBILITY	12
3.1.2	TECHNICAL FEASIBILITY	13
3.1.3	BEHAVIORAL FEASIBILITY	15
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	16
3.2	SYSTEM SPECIFICATION	18
3.2.1	HARDWARE SPECIFICATION	18
3.2.2	SOFTWARE SPECIFICATION	18
3.3	SOFTWARE DESCRIPTION	18
3.3.1	PYTHON DJANGO	18
3.3.2	SQLITE3	19
4	SYSTEM DESIGN	21
4.1	INTRODUCTION	22
4.2	UML DIAGRAM	22

<b>4.2.1</b>	<b>USE CASE DIAGRAM</b>	<b>23</b>
<b>4.2.2</b>	<b>SEQUENCE DIAGRAM</b>	<b>25</b>
<b>4.2.3</b>	<b>STATE CHART DIAGRAM</b>	<b>27</b>
<b>4.2.4</b>	<b>ACTIVITY DIAGRAM</b>	<b>29</b>
<b>4.2.5</b>	<b>CLASS DIAGRAM</b>	<b>31</b>
<b>4.2.6</b>	<b>OBJECT DIAGRAM</b>	<b>33</b>
<b>4.2.7</b>	<b>COMPONENT DIAGRAM</b>	<b>35</b>
<b>4.2.8</b>	<b>DEPLOYMENT DIAGRAM</b>	<b>36</b>
<b>4.3</b>	<b>USER INTERFACE DESIGN USING FIGMA</b>	<b>37</b>
<b>4.4</b>	<b>DATABASE DESIGN</b>	<b>40</b>
<b>5</b>	<b>SYSTEM TESTING</b>	<b>56</b>
<b>5.1</b>	<b>INTRODUCTION</b>	<b>57</b>
<b>5.2</b>	<b>TEST PLAN</b>	<b>57</b>
<b>5.2.1</b>	<b>UNIT TESTING</b>	<b>58</b>
<b>5.2.2</b>	<b>INTEGRATION TESTING</b>	<b>58</b>
<b>5.2.3</b>	<b>VALIDATION TESTING</b>	<b>59</b>
<b>5.2.4</b>	<b>USER ACCEPTANCE TESTING</b>	<b>59</b>
<b>5.2.5</b>	<b>AUTOMATION TESTING</b>	<b>60</b>
<b>5.2.6</b>	<b>SELENIUM TESTING</b>	<b>61</b>
<b>6</b>	<b>IMPLEMENTATION</b>	<b>78</b>
<b>6.1</b>	<b>INTRODUCTION</b>	<b>79</b>
<b>6.2</b>	<b>IMPLEMENTATION PROCEDURE</b>	<b>80</b>
<b>6.2.1</b>	<b>USER TRAINING</b>	<b>80</b>
<b>6.2.2</b>	<b>TRAINING ON APPLICATION SOFTWARE</b>	<b>80</b>
<b>6.2.3</b>	<b>SYSTEM MAINTENANCE</b>	<b>80</b>
<b>7</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>81</b>
<b>7.1</b>	<b>CONCLUSION</b>	<b>82</b>
<b>7.2</b>	<b>FUTURE SCOPE</b>	<b>83</b>
<b>8</b>	<b>BIBLIOGRAPHY</b>	<b>84</b>

<b>9</b>	<b>APPENDIX</b>	<b>86</b>
<b>9.1</b>	<b>SAMPLE CODE</b>	<b>87</b>
<b>9.2</b>	<b>SCREEN SHOTS</b>	<b>96</b>



## LIST OF ABBREVIATION

<b>UML</b>	Unified Modelling Language
<b>IDE</b>	Integrated Development Environment
<b>UML</b>	Unified Modelling Language
<b>ORM</b>	Object-Relational Mapping
<b>MVT</b>	Model-View-Template
<b>RDBMS</b>	Relational Database Management System
<b>1NF</b>	First Normal Form
<b>2NF</b>	Second Normal Form
<b>3NF</b>	Third Normal Form
<b>IDE</b>	Integrated Development Environment
<b>HTML</b>	HyperText Markup Language
<b>JS</b>	JavaScript
<b>CSS</b>	Cascading Style Sheets
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>JSON</b>	JavaScript Object Notation
<b>API</b>	Application Programming Interface
<b>UI</b>	User Interface
<b>HTTP</b>	Hypertext Transfer Protocol
<b>URL</b>	Uniform Resource Locator
<b>PK</b>	Primary Key
<b>FK</b>	Foreign Key
<b>SQL</b>	Structured Query Language
<b>CRUD</b>	Create, Read, Update, Delete

# **CHAPTER 1**

## **INTRODUCTION**

## 1.1 PROJECT OVERVIEW

EduSphere Fusion represents a groundbreaking initiative in the realm of education technology, serving as a comprehensive Foreign Study Management System combined with an e-learning platform. This innovative project is designed to revolutionize the way education is accessed, managed, and delivered globally, catering to the diverse needs of students, and colleges. The importance of EduSphere Fusion lies in its ability to integrate global educational resources, streamline administrative processes, and enhance the learning experience through a single, user-friendly platform.

At its technical core, EduSphere Fusion is developed using Python's Django framework, chosen for its robustness and scalability, which is essential for managing complex, data-driven applications. The database management is handled efficiently by SQLite3, known for its lightweight and reliable performance. The front-end design is crafted with a combination of HTML, CSS, JavaScript, and Bootstrap, ensuring a responsive and engaging interface that enhances user interaction across various devices.

EduSphere Fusion stands as a testament to the transformative power of technology in education. By leveraging cutting-edge digital solutions, it promises to bridge educational divides, offering seamless access to top-tier educational resources and institutions. This platform is not just a tool but a gateway to global educational opportunities, marking a significant step forward in making learning more accessible, interactive, and efficient.

## 1.2 PROJECT SPECIFICATION

EduSphere Fusion is an integrated Foreign Study Management System and e-learning platform designed to provide comprehensive educational services to administrators, students, and colleges. The system is built using Python's Django framework for backend, SQLite3 for database management, and HTML, CSS, JavaScript, and Bootstrap for frontend development.

### Admin Module

- User and College Management: Manage user accounts and college profiles, including creation, modification, and deletion.
- Monitoring Public Posts: Oversee and regulate the content shared on the platform to ensure quality and relevance.

- **Monetization Strategies:** Implement and manage various revenue-generating models and strategies.
- **Support Provision:** Provide assistance and support to users and colleges, addressing technical issues and general inquiries.

**User Module:**

- **Account Operations:**
  - **Registration and Login:** Enable users to create accounts and access the platform.
  - **Password Management:** Features for resetting or retrieving forgotten passwords.
- **Profile Management:** Allow users to view, create, and edit their personal profiles.
- **Educational Engagement:**
  - **Follow Colleges:** Users can follow specific colleges to get updates and news.
  - **Latest News and Updates:** Receive information about educational institutions and offerings.
  - **Public Domain Interaction:** Post queries, participate in discussions, and receive notifications.
- **E-Learning and Development:**
  - **Course Enrollment:** Sign up for online courses offered by various colleges.
  - **Certification and Progress Tracking:** Earn certifications and track progress in enrolled courses.
  - **Save and Track Posts:** Bookmark important educational posts and track course progress.
  - **Payment Processing:** Make payments for course enrollments and other services.

**College Module**

- **College Administration:**
  - **Registration and Login Management:** Manage college account creation and access.
  - **Profile Management:** Update and maintain college profiles.
  - **Public Post Uploads:** Share educational content and updates on the public feed.

- Department Management: Add, modify, and manage academic departments and instructors.
- Course Management:
  - Course Creation and Updates: Develop and update academic courses and their content.
  - Material Provision: Offer various learning materials, including texts, videos, and quizzes.
  - Examination and Certification: Conduct exams and issue certifications for courses.

## **CHAPTER 2**

### **SYSTEM STUDY**

## **2.1 INTRODUCTION**

Data collection and analysis, problem-solving, and system change recommendations are all steps in the process of system analysis. During this problem-solving process, there must be considerable communication between the system users and the system developers. A system analysis or research should be the first step in any system development process. The system analyst acts as an interrogator and examines the operation of the current system in great detail. The system's input is acknowledged, and the system is viewed as a whole. The many processes might be connected to the organizations' outcomes. System analysis involves comprehending the problem, identifying the significant and crucial variables, analyzing and synthesizing the numerous components, and choosing the best or, at the very least, most acceptable course of action.

Preliminary research is the process of gathering and analyzing data in order to use it for upcoming system investigations. Initial research requires strong collaboration between system users and developers since it involves problem-solving. It carries out several feasibility studies. These studies offer a rough idea of the system activities, which can be utilized to choose the methods to employ for effective system research and analysis.

## **2.2 EXISTING SYSTEM**

Existing educational management and e-learning platforms face several key challenges. Firstly, many suffer from poor integration, leading to fragmented user experiences as various functionalities operate separately. Secondly, scalability is a major issue, with systems struggling to adapt to growing user bases and increased data demands. Thirdly, outdated technology is common, limiting system capabilities and posing security risks. Customization and flexibility are also limited, preventing educational institutions from tailoring platforms to their specific needs. Finally, user interfaces often lack modernization, making navigation and full utilization of the platform's features challenging for users. These challenges underscore the need for more advanced and adaptable educational technology solutions.

### **2.2.1 NATURAL SYSTEM STUDIED**

In creating EduSphere Fusion, it was really important for us to understand how schools and colleges usually work. This normal way of doing things in education involves how teachers teach, how students learn, and how schools and colleges handle their day-to-day work.

We looked closely at how teachers teach their classes and how students learn from them. This helped us figure out how EduSphere Fusion can make learning and teaching better using technology. We also paid attention to how students and teachers talk to each other and work together, because this is a big part of learning.

Another important thing we looked at was how schools and colleges do their everyday tasks. Sometimes these tasks can take a lot of time and be a bit complicated. We wanted EduSphere Fusion to make these jobs easier and quicker for schools and colleges.

We also thought about how teachers give students their lessons and how students stay interested in what they're learning. In the past, they mostly used books, but now they also use videos and online stuff. We wanted to make sure EduSphere Fusion could help with this too.

By understanding all these things, EduSphere Fusion was designed to fit nicely into schools and colleges, making things better and easier for everyone. We wanted to help teachers teach better, make learning more fun and interesting for students, and help schools and colleges work more smoothly. This way, EduSphere Fusion can really help change the way we learn and teach.

## 2.2.2 DESIGNED SYSTEM STUDIED

The "Designed System Studied" for EduSphere Fusion refers to the carefully planned and structured system that we envisioned to improve and enhance the educational process. This system, as conceived, is a result of meticulous planning, taking into account the insights gained from studying the existing natural educational system.

In designing EduSphere Fusion, we focused on creating a system that not only addresses the gaps and challenges identified in traditional educational settings but also leverages the power of technology to bring about a transformation in learning and administrative efficiency.

**Key aspects of the Designed System include:**

**Enhanced Learning Experience:** We envisioned a system where the learning experience is enriched with interactive and engaging digital content, making education more accessible and enjoyable for students. This includes integrating multimedia resources, interactive modules, and online courses that cater to diverse learning styles.



**Streamlined Administrative Processes:** Recognizing the challenges in traditional administrative tasks, EduSphere Fusion is designed to automate and simplify these processes. This includes digital enrollment systems, automated course management tools, and efficient data handling to reduce administrative burdens on staff.

**Improved Communication and Collaboration:** A core feature of the designed system is to facilitate better communication between students and educators. We incorporated tools for real-time collaboration, discussion forums, and feedback mechanisms to foster a more interactive and responsive educational environment.

**Data-Driven Insights:** The system is designed to provide actionable insights into student performance and learning patterns. By analyzing data, educators can tailor their teaching methods to better meet the needs of their students, leading to improved educational outcomes.

**Scalability and Flexibility:** Keeping in mind the diverse needs of different educational institutions, the system is designed to be scalable, accommodating varying numbers of users and data volumes, and flexible enough to adapt to different educational settings.

## 2.3 DRAWBACKS OF EXISTING SYSTEM

- **Limited Integration:** Different functionalities like course management and student services often lack seamless integration.
- **Scalability Issues:** Difficulty in accommodating a growing number of users or handling increased data.
- **Outdated Technology:** Many systems rely on older technologies, which can restrict capabilities and lead to security vulnerabilities.
- **Limited Customization:** Reduced flexibility to tailor the system to specific institutional or user needs.
- **Inferior User Interface:** User interfaces can be outdated, non-intuitive, and less engaging.
- **Limited Data Analytics:** Lack of sophisticated tools for analyzing student performance and educational outcomes.
- **Inadequate User Support:** Often, there is limited assistance available for users encountering technical difficulties or needing guidance.
- **Inefficient Administrative Processes:** Many systems have cumbersome and time-consuming administrative procedures.

## 2.4 PROPOSED SYSTEM

The proposed system, EduSphere Fusion, is designed as a comprehensive platform that brings together global educational resources into a single, accessible space. This system is tailored not just for traditional student-teacher relationships, but for anyone who is eager to learn and explore educational opportunities offered by top colleges around the world.

EduSphere Fusion serves as a bridge connecting learners with a wide array of educational institutions globally. This platform stands out by offering a diverse range of learning opportunities, making it possible for users to access courses, materials, and knowledge from various international universities and colleges. Whether it's for professional development, personal enrichment, or academic pursuits, EduSphere Fusion caters to a wide spectrum of learning needs and interests.

One of the key features of EduSphere Fusion is the ability to obtain certifications from foreign universities. This aspect of the platform opens doors for learners to gain recognized qualifications from prestigious institutions, enhancing their professional and academic credentials.

Additionally, EduSphere Fusion is designed to keep users informed about various activities and opportunities offered by foreign colleges. This includes updates on new courses, seminars, workshops, and other educational events, allowing users to stay connected with the latest trends and developments in global education.

The platform is built with a focus on user-friendliness and accessibility, ensuring that learners from different backgrounds can easily navigate and utilize its resources. The use of advanced technology in EduSphere Fusion guarantees a smooth and efficient learning experience, accommodating a diverse and growing user base.

In essence, EduSphere Fusion is more than just an educational platform; it's a gateway to global education, offering unparalleled access to world-class learning resources. It embodies the spirit of bringing together a wealth of knowledge and opportunities under one roof, making it an invaluable tool for learners worldwide.

## 2.5 ADVANTAGES OF PROPOSED SYSTEM

- **Worldwide Educational Network:** Connects learners with top global universities and colleges.
- **Accredited Certifications:** Enables users to obtain recognized qualifications from international institutions.
- **Diverse Learning Options:** Offers a broad selection of courses catering to various interests and fields.
- **Real-Time Educational Updates:** Provides timely information on new courses and global educational events.
- **Ease of Use:** Designed for straightforward and intuitive user navigation.
- **Advanced Technology Integration:** Employs cutting-edge technology for an enhanced online learning experience.

## **CHAPTER 3**

### **REQUIREMENT ANALYSIS**

### 3.1 FEASIBILITY STUDY

A feasibility study is conducted to determine if the project will, upon completion, fulfil the objectives of the organization in relation to the labor, effort, and time invested in it. A feasibility study enables the developer to predict the project's usefulness and potential future. A system proposal's workability, which includes the influence on the organization, capacity to satisfy user demands, and efficient use of resources, is the basis for a feasibility study. As a result, a feasibility analysis is frequently performed before a new application is approved for development. The paper outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility assessment, including its technical, economic, and operational viabilities. The following are its features: -

#### 3.1.1 ECONOMICAL FEASIBILITY

**Revenue Streams:** The project has multiple revenue streams, including advertisements, fees from paid courses and subscription fees for personalized recommendations.

**Cost Estimations:** Costs will include development and maintenance expenses, marketing, and operational costs.

**Market Analysis:** The increasing trend in foreign education seekers and the need for reliable platforms make the project economically viable. With a growing demand for global education, revenue potential is promising.

##### **Economic Feasibility Questions:**

- Does the revenue model include multiple streams such as advertisements, paid courses, subscriptions, and IELTS coaching fees?

YES

- Have the development costs, operational expenses, and marketing/promotion costs been thoroughly assessed and considered in the economic feasibility analysis?

YES

- Is there a reasonable expectation of profitability based on projected revenue and cost estimates?

YES

### 3.1.2 TECHNICAL FEASIBILITY

The project is technically feasible due to the use of established technologies. It utilizes Python Django for the backend and HTML with CSS and Bootstrap for the frontend. These technologies are robust, widely supported, and suitable for building a scalable web application. Security Measures: The following security measures will be implemented:

- **Data Encryption:** User data, including personal information and payment details, will be encrypted using industry-standard encryption algorithms to protect against data breaches.
- **User Authentication:** Secure user authentication methods will be employed to ensure that only authorized users can access their accounts.
- **Data Access Control:** Access to sensitive user data will be strictly controlled, and role-based access control will be implemented for admin roles.
- **Regular Security Audits:** Regular security audits will be conducted to identify and address vulnerabilities.

**API Integration:** The project may integrate with external APIs to enhance functionality, such as payment gateways for handling transactions, geolocation services for college location information.

**Performance:** The system's performance will be monitored and optimized to ensure fast response times, even with a large user base.

#### **Recommendation Algorithm:**

- **User Profiling:** Users will provide information about their study interests, academic performance, and preferences.
- **Machine Learning:** Machine learning models may be employed to enhance recommendation accuracy as the system collects more data.

#### **Technical Feasibility Questions:**

- Is the chosen technology stack (Python Django for the backend, HTML/CSS/Bootstrap for the frontend) readily available and suitable for building the GlobalEd Fusion platform?

YES

- Are there existing libraries or APIs that can be used to implement essential features, such as payment processing and product browsing?

YES

- Do We Have the Required Expertise?

YES

- Are there team members skilled in user interface (UI) and user experience (UX) design to create an appealing and user-friendly interface?

YES

- Can the chosen technology stack handle the anticipated number of users accessing the system simultaneously?

YES

- Are there strategies in place to ensure the performance and responsiveness as the user base grows?

YES

- Are there well-defined APIs or integration points available for seamless data exchange between the system and backend?

YES

- Can we implement strong security measures to protect user data and payment information?

YES

- Are there encryption protocols and authentication mechanisms that can be employed to ensure data privacy?

YES

### **3.1.3 BEHAVIORAL FEASIBILITY**

EduSphere Fusion, a comprehensive and innovative online education platform, aims to unite global educational resources. Based on a thorough assessment, it is concluded that the project is behaviorally feasible. This conclusion is drawn from its alignment with current educational trends, user needs, and technological adaptability.

#### **User Acceptance and Engagement:**

EduSphere Fusion's intuitive and user-friendly design is highly conducive to positive user engagement. Its interface is crafted considering a diverse user base, making it accessible to students, educators, and administrative staff with varying levels of tech-savviness. The platform's features, such as course enrollment, content access, and administrative functionalities, are tailored to meet the common and specialized needs of its users. This level of user-centric design is likely to foster a high rate of acceptance and sustained engagement.

#### **Cultural and Educational Versatility:**

The platform's design takes into account the diverse cultural and educational landscapes it aims to serve. Its adaptability to different languages, educational norms, and teaching methodologies enhances its feasibility across various global regions. EduSphere Fusion's capability to customize and cater to a wide range of educational contexts underpins its behavioral feasibility on a global scale.

#### **Training and Support Systems:**

Understanding the importance of smooth transition and adoption, EduSphere Fusion is complemented by comprehensive training and support systems. These resources are planned to assist users in navigating the platform efficiently, addressing a critical aspect of behavioral feasibility - ease of adoption.

#### **Stakeholder Engagement:**

The project has garnered positive responses from key stakeholders, including educational institutions and industry experts. Their preliminary support and interest indicate a readiness to integrate and promote such a platform within their networks. This stakeholder buy-in is a strong indicator of the project's behavioral feasibility, as it underlines the perceived value and utility of EduSphere Fusion in the educational sector.



**Effective Change Management:**

The implementation strategy for EduSphere Fusion includes well-structured change management processes. Anticipating the natural resistance to transitioning from traditional methods to a digital platform, these processes focus on clear communication, addressing potential concerns, and highlighting the benefits of the system. This approach is expected to facilitate a smoother transition and greater acceptance among users.

**3.1.4 FEASIBILITY STUDY QUESTIONNAIRE****1. What inspired the development of EduSphere Fusion and what problems does it aim to solve?**

EduSphere Fusion was inspired by the increasing demand for global education opportunities and the need to address issues related to scammers and intermediaries in the foreign education space. The project aims to provide a reliable platform for students seeking foreign education, offering accurate information about colleges, courses, and admission procedures while protecting them from fraudulent activities.

**2. What features and services do foreign colleges provide through the platform?**

Foreign colleges on the EduSphere Fusion platform can offer both paid and free online certification courses. They can also post timely updates and feeds to inform their followers about important information, events, and news related to the college.

**3. Can you explain the revenue model including how the platform generates income from advertisements and paid courses?**

Advertisements: Advertisers pay to display targeted ads to the user base.

Paid Courses: Colleges charge fees for their online certification courses, and a portion of these fees goes to EduSphere Fusion.

**4. How does EduSphere Fusion plan to address competition in the market and stay relevant in the long term?**

The platform aims to differentiate itself by providing verified and trustworthy information, a strong user community, and a range of features to meet the needs of students and colleges.

**5. Who is responsible for verifying tutor and college details and what are the criteria for verification?**

The platform's admin is responsible for verifying tutor and college details. Verification criteria include checking the credentials and ensuring colleges are accredited and confirming the authenticity of the information provided.

**6. What measures are in place to protect user data and privacy?**

To protect user data and privacy, EduSphere Fusion employs industry-standard security practices. This includes data encryption, secure authentication, regular security audits, and compliance with data protection regulations.

**7. How does the platform facilitate personalized recommendations through paid subscriptions?**

To access personalized college recommendations, users can subscribe to premium services for a fee. Subscribers provide information about their study interests and academic performance, allowing the platform to offer tailored suggestions.

**8. What plans are in place for scaling and expanding the platform in the future?**

The project plans to expand by adding more features, increasing the number of colleges and courses, expanding IELTS coaching options, and enhancing user engagement through community-building activities. There are also plans to explore partnerships with educational institutions to broaden the platform's offerings.

**9. How does the platform ensure the accuracy and reliability of information about foreign colleges?**

EduSphere Fusion relies on a combination of user-generated content and verified data. Admin verification ensures that colleges listed on the platform meet certain standards and criteria. User feedback and reporting mechanisms also help in maintaining accuracy and authenticity.

**10. How does the public domain for questions and answers work and how does it benefit users?**

The public domain allows users to ask questions or queries related to foreign education. Any user can respond, creating a collaborative environment where students can get quick and diverse responses to their doubts and concerns

### **3.1 SYSTEM SPECIFICATION**

#### **3.2.1 HARDWARE SPECIFICATION**

Processor	-	Intel I5 12 <sup>th</sup> Gen Processor
RAM	-	16 Gb DDR4 RAM
Hard disk	-	512Gb

#### **3.2.2 SOFTWARE SPECIFICATION**

Front End	-	HTML, CSS, Bootstrap
Back End	-	Python Djnago
Database	-	Sqlite3
Client on PC	-	Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, Python, CSS, Bootstrap, Razorpay, Allauth

### **3.3 SOFTWARE DESCRIPTION**

#### **3.3.1 PYTHON DJANGO**

Python Django is a high-level, open-source web framework written in Python, one of the world's most popular programming languages. Django is designed to encourage rapid development and clean, pragmatic design. It takes care of much of the hassle of web development, allowing developers to focus on writing their app without needing to reinvent the wheel.

Key aspects of Django include:

- **MTV Architecture:** Django follows the Model-Template-View (MTV) architectural pattern, which is similar to the Model-View-Controller (MVC) framework. This structure separates the database (Model), user interface (Template), and business logic (View), leading to organized and efficient code development.
- **Batteries-Included Philosophy:** Django comes with a multitude of built-in features for common web development tasks. This includes tools for user authentication, content administration, site maps, RSS feeds, and many more, essentially offering a comprehensive package for web developers.
- **ORM (Object-Relational Mapper):** Django's ORM allows developers to interact with a database using Python code instead of SQL. This makes database operations easier and more intuitive, especially for developers who are more familiar with Python than with database query languages.
- **Security Emphasis:** Django is renowned for its robust security features. It helps developers avoid common security mistakes by providing a framework that has been engineered to "do the right things" in terms of security, like protecting against SQL injection, cross-site scripting, cross-site request forgery, and clickjacking.
- **Scalability and Versatility:** Django is used for all sizes of projects, from small-scale applications to large-scale, complex sites. It is also versatile enough to be used in various types of projects, including content management systems, social networks, and scientific computing platforms.
- **Strong Community and Documentation:** Django has a large and active community, which contributes to a wealth of documentation, third-party packages, plugins, and resources. This community support makes Django a reliable choice for developers.

### 3.3.2 SQLITE3

SQLite3 is a widely-used database engine renowned for its reliability, simplicity, and lightweight nature. It's a self-contained, serverless, zero-configuration, and transactional SQL database engine, making it a popular choice for applications of all sizes, especially in embedded systems and small to medium-sized web applications.

Key characteristics of SQLite3 include:

- **Self-Contained System:** SQLite3 is self-contained, meaning it requires minimal support from the operating system or external libraries. This makes it highly portable and easy to set up and use.
- **Serverless Architecture:** Unlike traditional databases, SQLite3 doesn't operate on a client-server model. The SQLite3 library accesses its storage files directly. This serverless architecture simplifies the configuration process and eliminates the need for a dedicated database server.
- **Zero Configuration:** With no server to set up and no settings to configure, SQLite3 is essentially a plug-and-play database. It does not require any installation or setup procedures, which is a huge advantage in terms of ease of use and deployment.
- **Transactional:** SQLite3 is transactional, meaning that all changes within a single transaction in SQLite3 are atomic, consistent, isolated, and durable (ACID), even in cases of system crashes or power failures. This ensures data integrity and reliability.
- **Cross-Platform:** It runs on all major operating systems, and its database files are compatible across different platforms. This cross-platform capability makes it highly adaptable and suitable for a wide range of applications.
- **Lightweight:** One of the most significant features of SQLite3 is its lightweight nature. It has a relatively small footprint, both in terms of memory usage and disk space requirements, making it ideal for use in devices with limited resources, such as smartphones or IoT devices.
- **Wide Usage:** SQLite3 is used extensively in various applications, from web browsers to mobile apps and operating systems. It's the database of choice for applications that require an embedded database and is known for its reliability and straightforward implementation.

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

System design in software and web development is akin to creating a master blueprint for a building. It involves laying out a detailed architecture for the software, defining how each component interacts with others, and ensuring the overall system achieves its intended functionality. This step is crucial in determining how well the software meets both functional requirements (like specific tasks it needs to perform) and non-functional ones (such as security and performance).

A well-planned system design streamlines the development process, saving time and resources, especially in large-scale projects where multiple teams are involved. It ensures that different parts of the software integrate seamlessly. Furthermore, designing with security in mind is essential in today's digital landscape, as it helps safeguard against potential vulnerabilities from the outset.

In terms of maintenance, a clear and well-documented system design simplifies updates and troubleshooting, making the software more sustainable and adaptable over time. In summary, effective system design is key to creating software that is not only functional but also secure, efficient, and capable of evolving with technological advancements and user needs.

## 4.2 UML DIAGRAM

Unified Modeling Language (UML) stands as a foundational tool in software engineering, renowned for its role in visually representing complex systems and processes. It provides a standardized set of graphical notations that facilitate the clear depiction of various aspects of a system's structure and behavior. Originating from the collaboration of industry experts, UML has gained widespread acceptance and adoption in both academia and industry. It serves as a powerful communication tool, enabling stakeholders, including developers, designers, and clients, to attain a shared understanding of system architecture, design, and functionality. UML diagrams act as a lingua franca, transcending language barriers and ensuring a consistent means of conveying intricate software concepts, ultimately enhancing the efficiency and effectiveness of the software development process.

### Types of UML diagrams

- Class diagram
- Object diagram

- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

### 4.2.1 USE CASE DIAGRAM

Use Case Diagrams, a cornerstone in software engineering, serve as a visual representation of the interactions between a system and its external entities. At their core, they provide a structured means of identifying and defining the various functionalities a system offers and how these functionalities are accessed by different actors or entities. Actors, representing users, systems, or external entities, are depicted along with the specific use cases they engage with. Associations between actors and use cases elucidate the nature of these interactions, clarifying the roles and responsibilities of each entity within the system. This detailed visual representation not only enhances communication among stakeholders but also provides a clear blueprint for system functionality, laying the foundation for the subsequent stages of the software development process. Overall, Use Case Diagrams play a pivotal role in aligning development efforts with user expectations, ensuring that the resulting software system fulfills its intended purpose effectively and efficiently.

- **Actor Definition:** Clearly define and label all actors involved in the system. Actors represent external entities interacting with the system.
- **Use Case Naming:** Use descriptive names for use cases to accurately convey the functionality they represent.
- **Association Lines:** Use solid lines to represent associations between actors and use cases. This signifies the interaction between entities.
- **System Boundary:** Draw a box around the system to indicate its scope and boundaries. This defines what is inside the system and what is outside.
- **Include and Extend Relationships:** Use "include" relationships to represent common functionalities shared among multiple use cases. Use "extend" relationships to show optional or extended functionalities.





Figure 4.2.1.1 USE CASE DIAGRAM

### 4.2.2 SEQUENCE DIAGRAM

Sequence Diagrams stand as dynamic models in software engineering, portraying the chronological flow of interactions between various objects or components within a system. They spotlight the order in which messages are exchanged, revealing the behavior of the system over time. Actors and objects are represented along a vertical axis, with arrows indicating the sequence of messages and their direction. Lifelines, extending vertically from actors or objects, illustrate their existence over the duration of the interaction. These diagrams serve as a vital tool for visualizing system behavior and understanding the temporal aspects of a software process. Through Sequence Diagrams, stakeholders gain valuable insights into how different elements collaborate to achieve specific functionalities, facilitating more effective communication among development teams and stakeholders alike. This detailed representation not only aids in detecting potential bottlenecks or inefficiencies but also provides a foundation for refining system performance in the later stages of software development.

- **Vertical Ordering:** Represent actors and objects along a vertical axis, indicating the order of interactions from top to bottom.
- **Lifelines:** Extend vertical lines from actors or objects to denote their existence and participation in the interaction.
- **Activation Bars:** Use horizontal bars along lifelines to show the period during which an object is active and processing a message.
- **Messages and Arrows:** Use arrows to indicate the flow of messages between objects, specifying the direction of communication.
- **Self-Invocation:** Use a looped arrow to represent self-invocation, when an object sends a message to itself.
- **Return Messages:** Indicate return messages with a dashed arrow, showing the response from the recipient.
- **Focus on Interaction:** Sequence Diagrams focus on the chronological order of interactions, avoiding implementation details.
- **Concise Notation:** Use clear and concise notation to represent messages and interactions, avoiding unnecessary complexity.
- **Consider System Boundaries:** Clearly define the boundaries of the system to indicate what is included in the interaction.

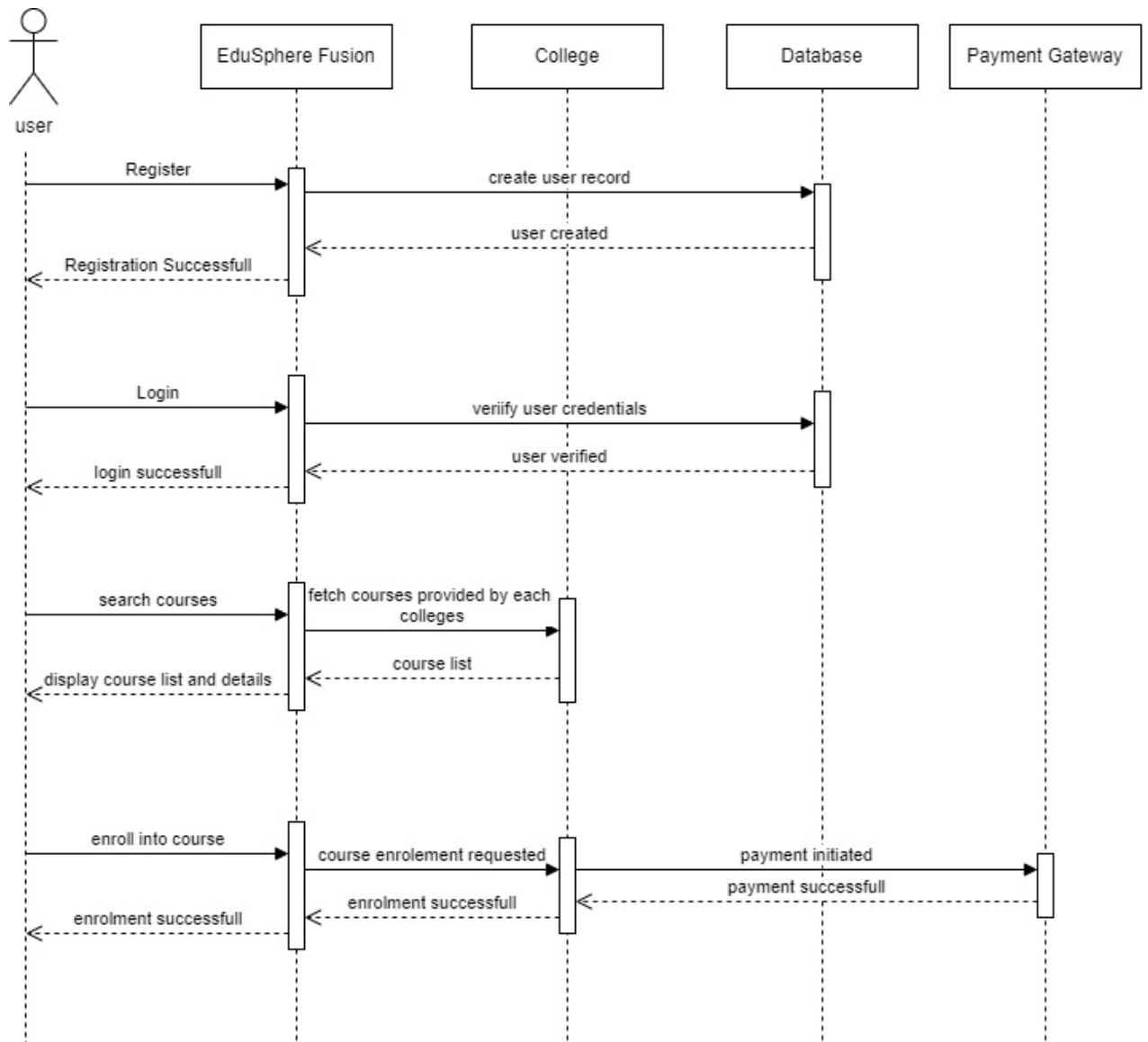


Figure 4.2.2.1 SEQUENCE DIAGRAM

### 4.2.3 STATE CHART DIAGRAM

A State Chart Diagram, a fundamental component of UML, provides a visual representation of an object's lifecycle states and the transitions between them. It depicts the dynamic behavior of an entity in response to events, showcasing how it transitions from one state to another. Each state represents a distinct phase in the object's existence, while transitions illustrate the conditions triggering state changes. Initial and final states mark the commencement and termination of the object's lifecycle. Orthogonal regions allow for concurrent states, capturing multiple aspects of the object's behavior simultaneously. Hierarchical states enable the representation of complex behaviors in a structured manner. Entry and exit actions depict activities occurring upon entering or leaving a state. Moreover, guard conditions ensure that transitions occur only under specified circumstances. State Chart Diagrams play a crucial role in understanding and designing the dynamic behavior of systems, aiding in the development of robust and responsive software applications.

Key notations for State Chart Diagrams:

- **Initial State:** Represented by a filled circle, it signifies the starting point of the object's lifecycle.
- **State:** Depicted by rounded rectangles, states represent distinct phases in an object's existence.
- **Transition Arrow:** Arrows denote transitions between states, indicating the conditions triggering a change.
- **Event:** Events, triggers for state changes, are labeled on transition arrows.
- **Guard Condition:** Shown in square brackets, guard conditions specify criteria for a transition to occur.
- **Final State:** Represented by a circle within a larger circle, it indicates the end of the object's lifecycle.
- **Concurrent State:** Represented by parallel lines within a state, it signifies concurrent behaviors.
- **Hierarchy:** States can be nested within other states to represent complex behavior.
- **Entry and Exit Actions:** Actions occurring upon entering or leaving a state are labeled within the state.
- **Transition Labels:** Labels on transition arrows may indicate actions or operations that accompany the transition.

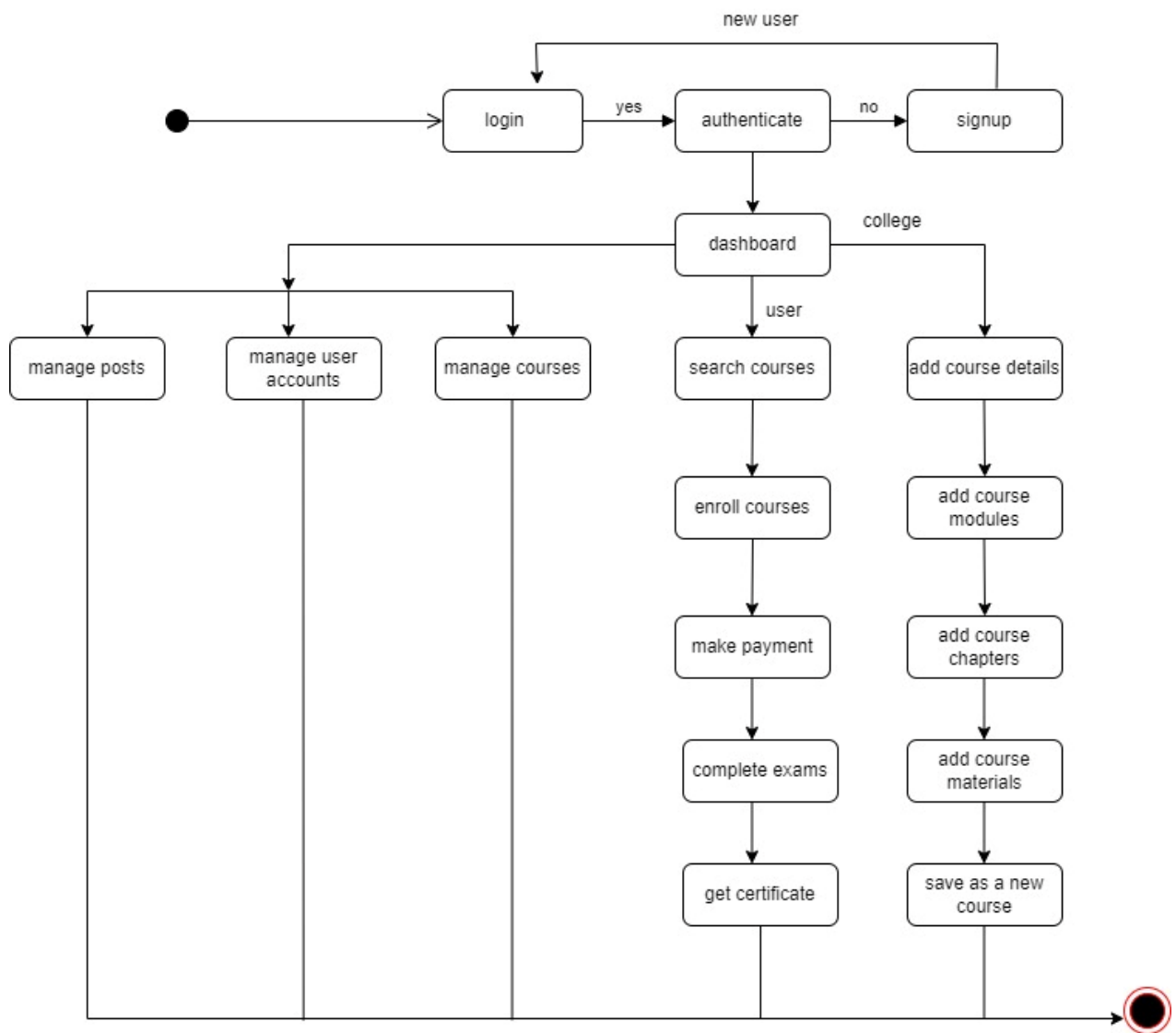


Figure 4.2.3.1 STATE CHART DIAGRAM

## 4.2.4 ACTIVITY DIAGRAM

An Activity Diagram is a visual representation within UML that illustrates the flow of activities and actions in a system or process. It employs various symbols to depict tasks, decision points, concurrency, and control flows. Rectangles signify activities or tasks, while diamonds represent decision points, allowing for conditional branching. Arrows indicate the flow of control from one activity to another. Forks and joins denote concurrency, where multiple activities can occur simultaneously or in parallel. Swimlane segregate activities based on the responsible entity, facilitating clarity in complex processes. Initial and final nodes mark the commencement and completion points of the activity. Decision nodes use guards to determine the path taken based on conditions. Synchronization bars enable the coordination of parallel activities. Control flows direct the sequence of actions, while object flows depict the flow of objects between activities. Activity Diagrams serve as invaluable tools for understanding, modeling, and analyzing complex workflows in systems and processes. They offer a structured visual representation that aids in effective communication and system development.

Key notations for Activity Diagrams:

- **Initial Node:** Represented by a solid circle, it signifies the starting point of the activity.
- **Activity:** Shown as a rounded rectangle, it represents a task or action within the process.
- **Decision Node:** Depicted as a diamond shape, it indicates a point where the process flow can diverge based on a condition.
- **Merge Node:** Represented by a hollow diamond, it signifies a point where multiple flows converge.
- **Fork Node:** Shown as a horizontal bar, it denotes the start of concurrent activities.
- **Join Node:** Depicted as a vertical bar, it marks the point where parallel flows rejoin.
- **Final Node:** Represented by a solid circle with a border, it indicates the end of the activity.
- **Control Flow:** Arrows connecting activities, showing the sequence of actions.
- **Object Flow:** Lines with arrows representing the flow of objects between activities.
- **Swimlane:** A visual container that groups activities based on the responsible entity or system component.
- **Partition:** A horizontal or vertical area within a swimlane, further organizing activities.

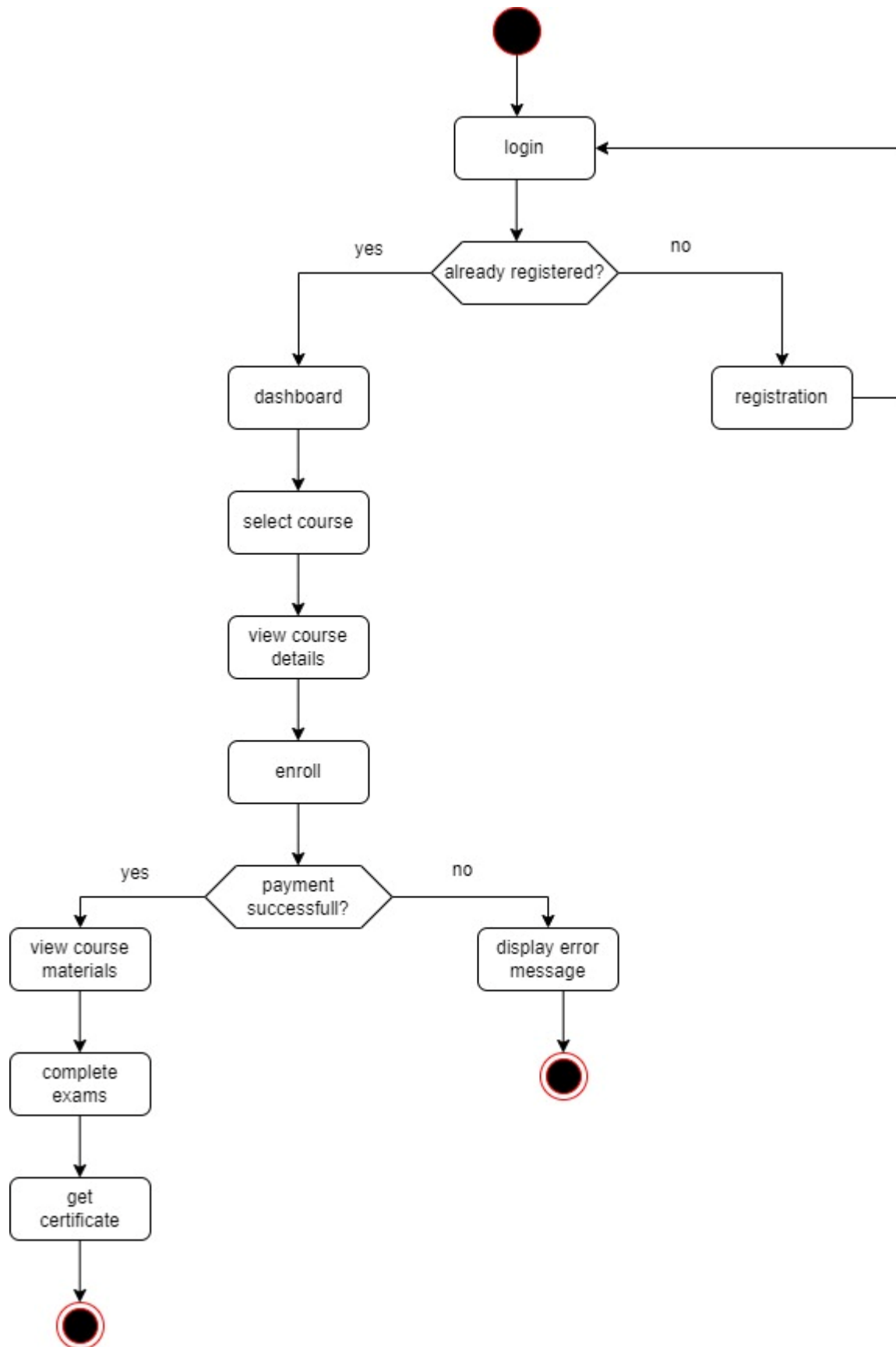


Figure 4.2.4.1 ACTIVITY DIAGRAM

### 4.2.5 CLASS DIAGRAM

A Class Diagram, a fundamental tool in UML, visually represents the structure of a system by illustrating classes, their attributes, methods, and relationships. Classes, depicted as rectangles, encapsulate data and behavior within a system. Associations between classes indicate relationships, showcasing how they interact. Multiplicity notations specify the cardinality of associations. Inheritance is denoted by an arrow indicating the subclass inheriting from a super-class. Aggregation and composition illustrate whole-part relationships between classes. Interfaces, depicted as a circle, outline the contract of behavior a class must implement. Stereotypes provide additional information about a class's role or purpose. Dependencies highlight the reliance of one class on another. Association classes facilitate additional information about associations. Packages group related classes together, aiding in system organization. Class Diagrams play a pivotal role in system design, aiding in conceptualizing and planning software architectures. They serve as a blueprint for the development process, ensuring a clear and structured approach to building robust software systems.

Key notations for Class Diagrams:

- **Class:** Represented as a rectangle, it contains the class name, attributes, and methods.
- **Attributes:** Displayed as a list within the class, they describe the properties or characteristics of the class.
- **Methods:** Also listed within the class, they define the behaviors or operations of the class.
- **Associations:** Lines connecting classes, indicating relationships and connections between them.
- **Multiplicity Notation:** Indicates the number of instances one class relates to another.
- **Inheritance:** Shown as an arrow, it signifies that one class inherits properties and behaviors from another.
- **Interfaces:** Represented by a dashed circle, they define a contract of behavior that implementing classes must follow.
- **Stereotypes:** Additional labels or annotations applied to classes to provide more information about their role or purpose.
- **Dependencies:** Shown as a dashed line with an arrow, they indicate that one class relies on another in some way.
- **Association Classes:** Represented as a class connected to an association, they provide additional information about the relationship.



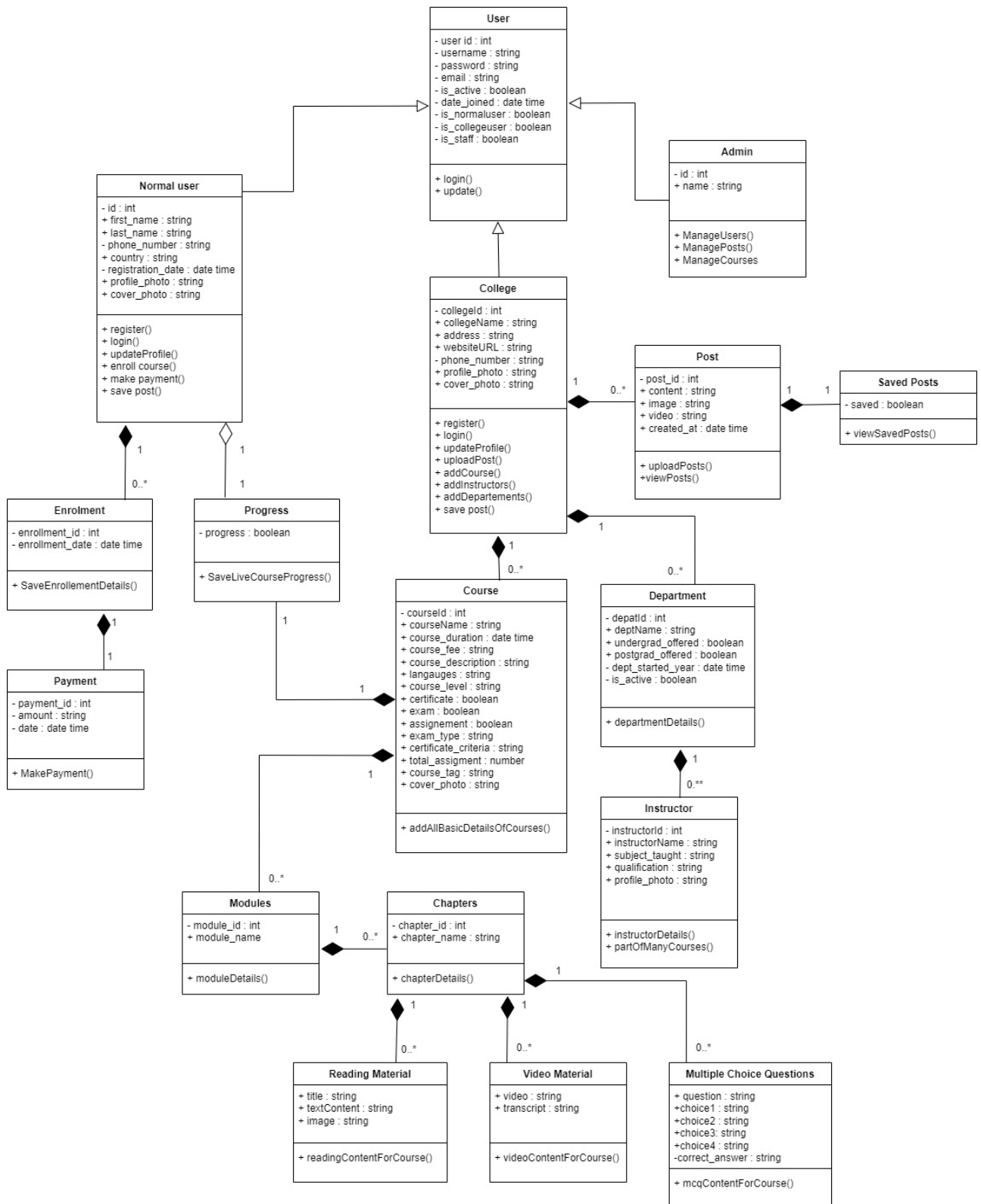


Figure 4.2.5.1 CLASS DIAGRAM

## 4.2.6 OBJECT DIAGRAM

An Object Diagram in UML provides a snapshot of a system at a specific point in time, displaying the instances of classes and their relationships. Objects, represented as rectangles, showcase the state and behavior of specific instances. Links between objects depict associations, highlighting how they interact. Multiplicity notations indicate the number of instances involved in associations. The object's state is displayed through attributes and their corresponding values. Object Diagrams offer a detailed view of runtime interactions, aiding in system understanding and testing. They focus on real-world instances, providing a tangible representation of class relationships. While similar to Class Diagrams, Object Diagrams emphasize concrete instances rather than class definitions. They serve as valuable tools for validating system design and verifying that classes and associations work as intended in practice. Object Diagrams play a crucial role in system validation, ensuring that the system's components and their interactions align with the intended design and requirements.

Key notations for Object Diagrams:

- **Object:** Represented as a rectangle, it contains the object's name and attributes with their values.
- **Links:** Lines connecting objects, indicating associations or relationships between them.
- **Multiplicity Notation:** Indicates the number of instances involved in associations.
- **Attributes with Values:** Displayed within the object, they represent the state of the object at a specific point in time.
- **Role Names:** Labels applied to associations, providing additional information about the nature of the relationship.
- **Object Name:** Represents the name of the specific instance.
- **Association End:** Indicates the end of an association, often with a role name and multiplicity.
- **Dependency Arrow:** Indicates a dependency relationship, where one object relies on another.
- **Composition Diamond:** Represents a stronger form of ownership, where one object encapsulates another.
- **Aggregation Diamond:** Signifies a whole-part relationship between objects.

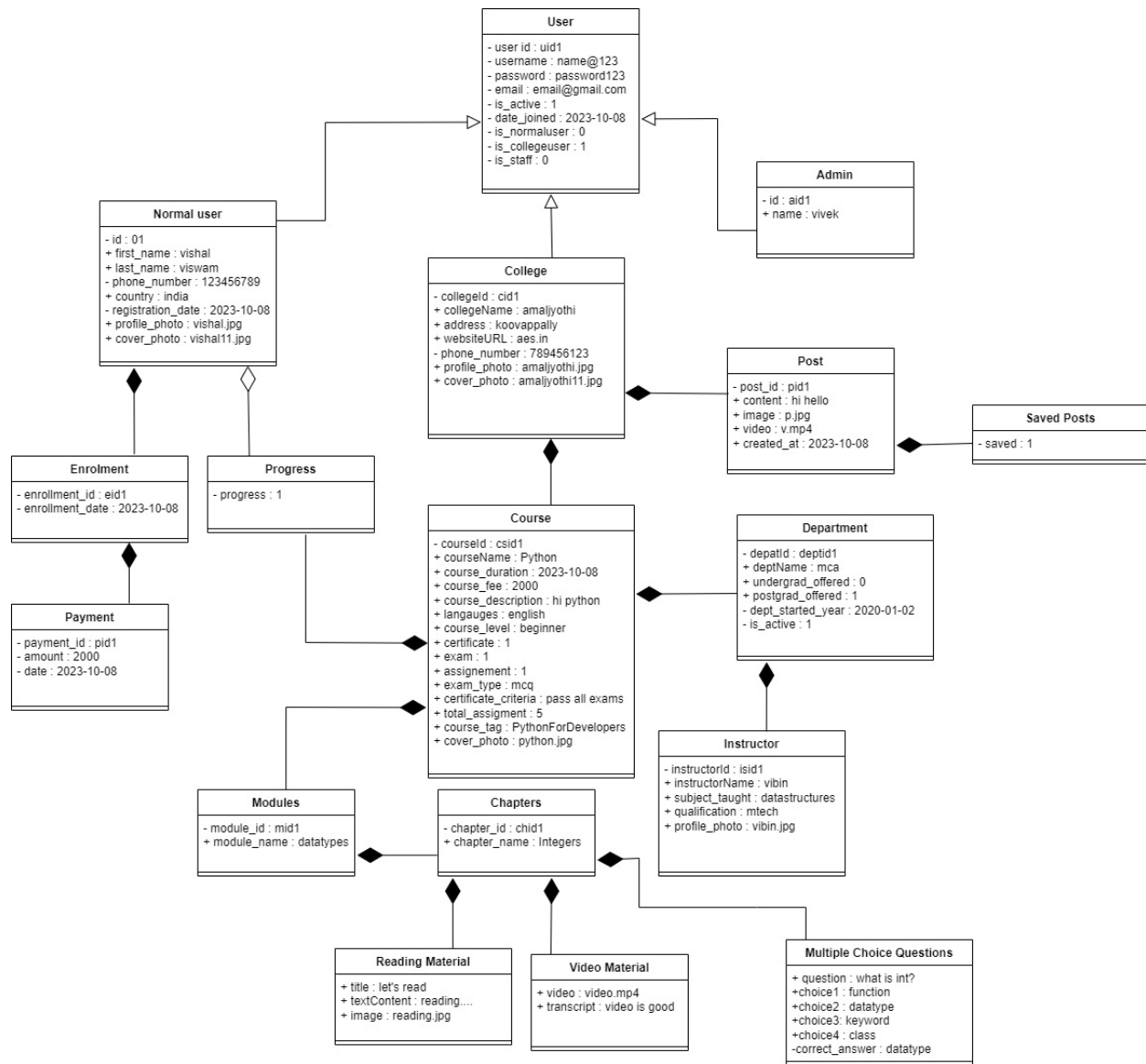


Figure 4.2.6.1 OBJECT DIAGRAM

## 4.2.7 COMPONENT DIAGRAM

A Component Diagram, a vital aspect of UML, offers a visual representation of a system's architecture by showcasing the high-level components and their connections. Components, depicted as rectangles, encapsulate modules, classes, or even entire systems. Dependencies between components are displayed through arrows, signifying the reliance of one component on another. Interfaces, represented by a small circle, outline the services a component offers or requires. Connectors link interfaces to denote the required or provided services. Ports, depicted as small squares, serve as connection points between a component and its interfaces. Stereotypes provide additional information about the role or purpose of a component. Deployment nodes indicate the physical location or environment in which components are deployed. Component Diagrams are instrumental in system design, aiding in the organization and visualization of system architecture. They emphasize the modular structure, facilitating ease of development, maintenance, and scalability of complex software systems. Overall, Component Diagrams play a pivotal role in planning and orchestrating the architecture of sophisticated software applications.

Key notations for Component Diagrams:

- **Component:** Represented as a rectangle, it encapsulates a module, class, or system.
- **Dependency Arrow:** Indicates that one component relies on or uses another.
- **Interface:** Depicted as a small circle, it outlines the services a component offers or requires.
- **Provided and Required Interfaces:** Connectors link provided interfaces to required interfaces.
- **Port:** Shown as a small square, it serves as a connection point between a component and its interfaces.
- **Stereotypes:** Additional labels or annotations applied to components to provide more information about their role or purpose.
- **Assembly Connector:** Represents the physical connection between two components.
- **Artifact:** A physical piece of information that is used or produced by a software development process.
- **Deployment Node:** Indicates the physical location or environment in which components are deployed.
- **Manifestation Arrow:** Indicates the implementation of an interface by a component

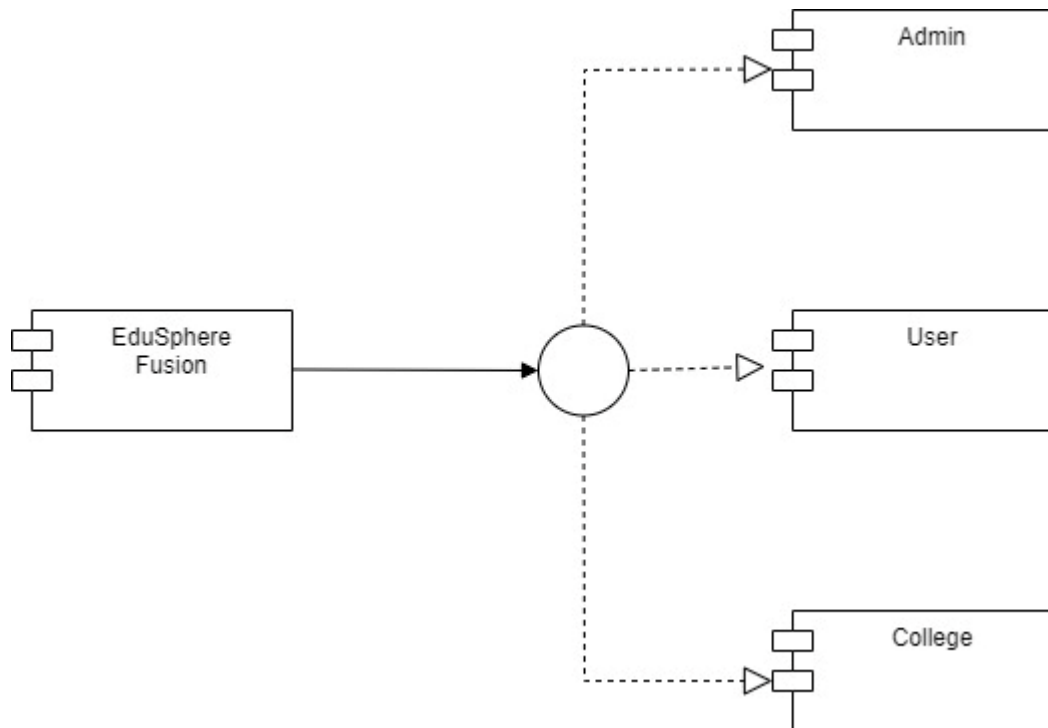


Figure 4.2.7.1 COMPONENT DIAGRAM

## 4.2.8 DEPLOYMENT DIAGRAM

A Deployment Diagram, a crucial facet of UML, provides a visual representation of the physical architecture of a system, showcasing the hardware nodes and software components. Nodes, representing hardware entities like servers or devices, are depicted as rectangles. Artifacts, denoted by rectangles with a folded corner, represent software components or files deployed on nodes. Associations between nodes and artifacts indicate the deployment of software on specific hardware. Dependencies illustrate the reliance of one node on another. Communication paths, shown as dashed lines, represent network connections between nodes. Stereotypes provide additional information about the role or purpose of nodes and artifacts. Deployment Diagrams are instrumental in system planning, aiding in the visualization and organization of hardware and software components. They emphasize the allocation of software modules to specific hardware nodes, ensuring efficient utilization of resources. Overall, Deployment Diagrams play a pivotal role in orchestrating the physical infrastructure of complex software applications.

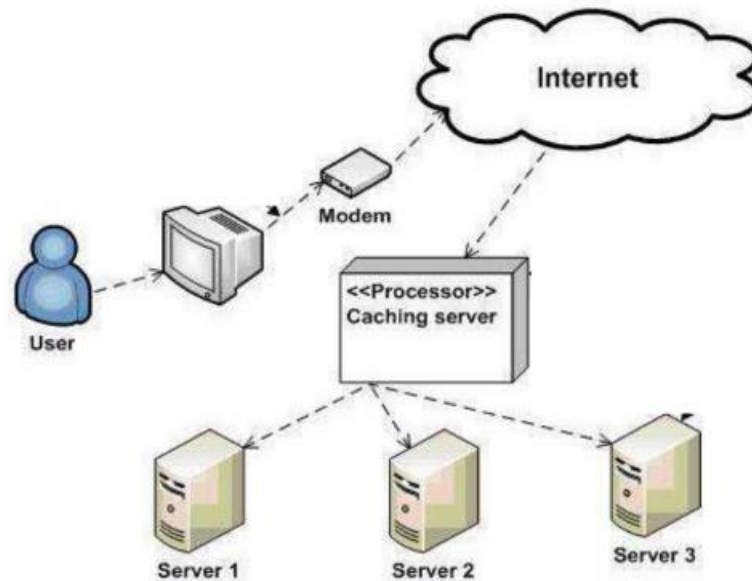


Figure 4.2.8.1 DEPLOYMENT DIAGRAM

## 4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: Registration Page

The registration page features a blue header with the 'EduSphere Fusion' logo. Below the header, the text 'Create an account' is displayed. The form contains six input fields: Email, Username, Full Name, Phone, Password, and Confirm Password. Each field is accompanied by a corresponding icon (envelope, person, person, phone, padlock, and padlock). Below the input fields, there is a 'Continue with Google?' button with the Google logo and a 'Register' button. The right side of the page shows a smiling man with a backpack holding a green folder, standing in front of a building.

Figure 4.3.1 REGISTRATION PAGE

**Form Name: Login Page**

Figure 4.3.2 LOGIN PAGE

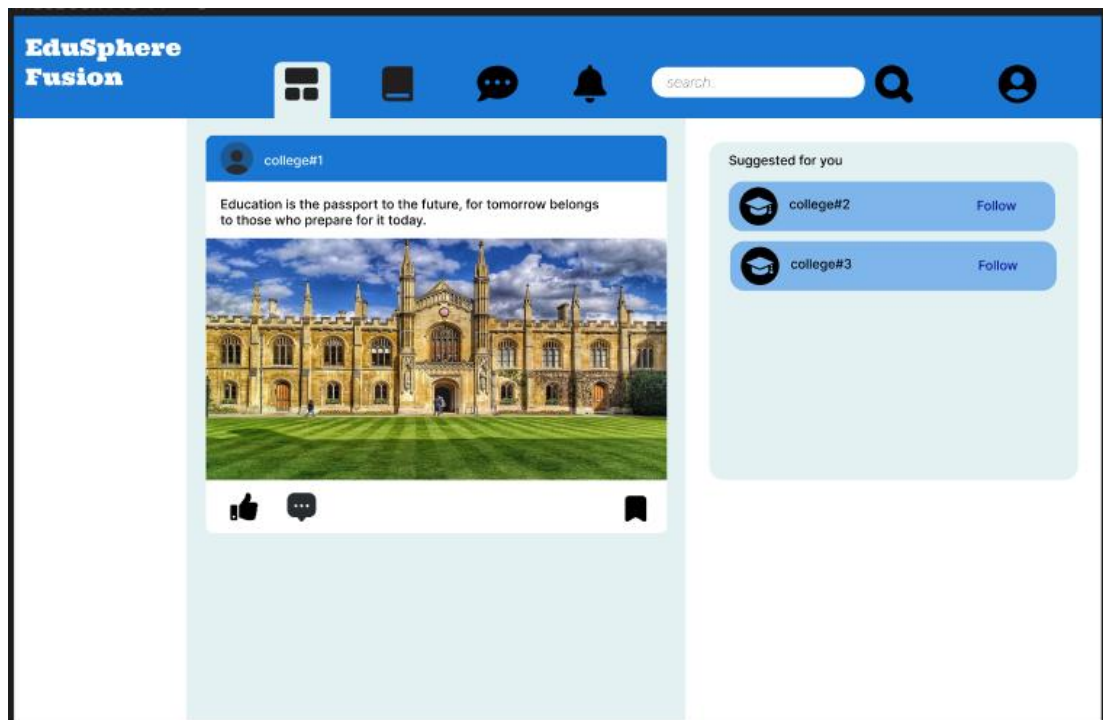
**Form Name: User Dashboard**

Figure 4.3.3 USER HOME PAGE



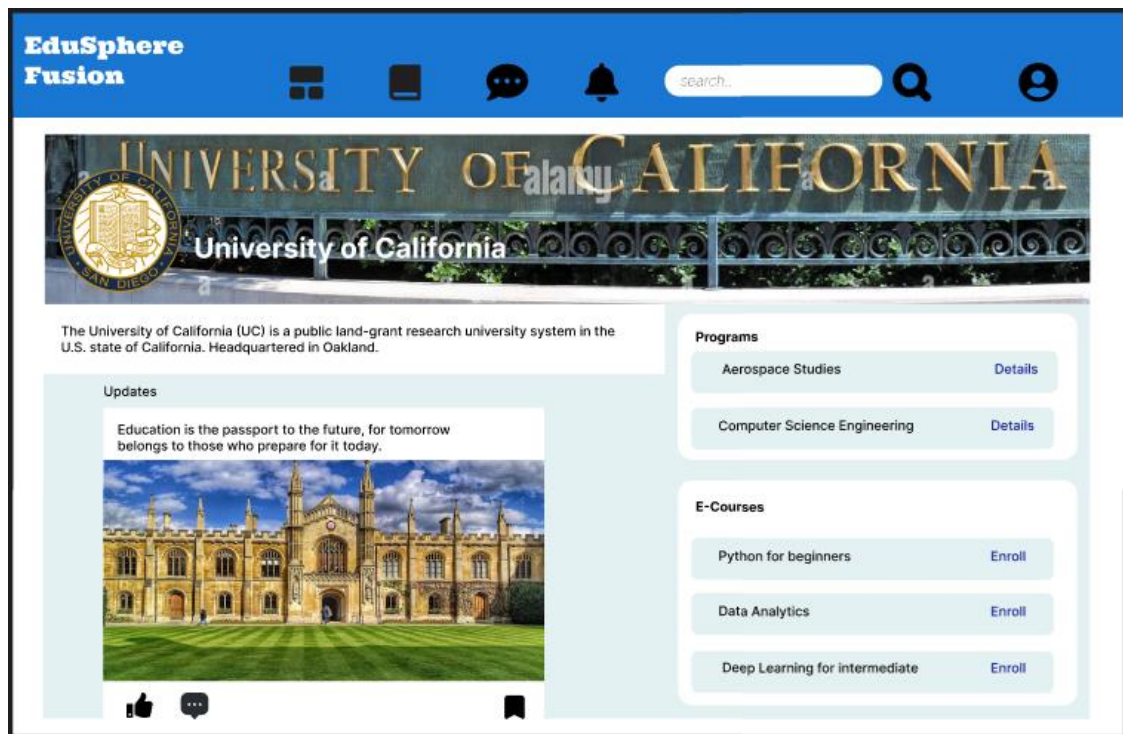
**Form Name: College Profile Page**

Figure 4.3.4 COLLEGE PROFILE PAGE

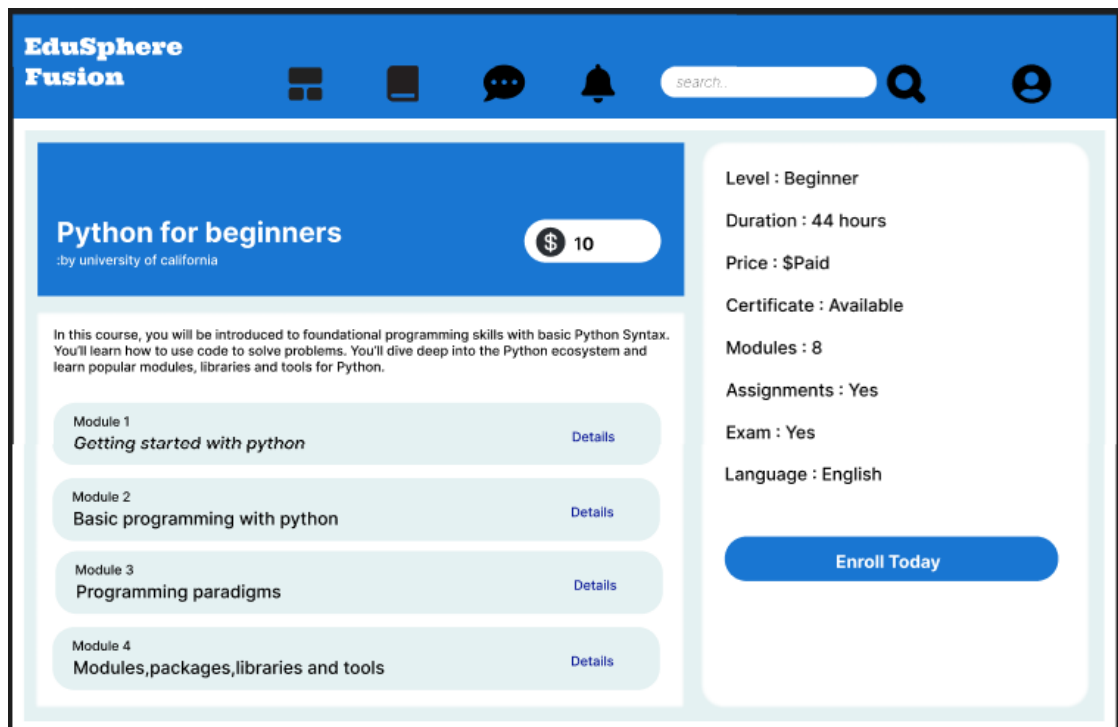
**Form Name: Course/Course Details Page**

Figure 4.3.5 COURSE PAGE



## **4.4 DATABASE DESIGN**

Database Design is a critical component in the realm of information management and software development. It involves the thoughtful and systematic organization of data to ensure efficient storage, retrieval, and manipulation. A well-designed database serves as the backbone of applications, enabling them to handle large volumes of information with speed and accuracy. This process encompasses defining the structure, relationships, and constraints of data entities, optimizing for performance and scalability. Effective database design is pivotal in minimizing redundancy, ensuring data integrity, and providing a foundation for robust data analytics. It involves a deep understanding of business requirements and user needs, translating them into a coherent and logical data model. The goal of a sound database design is to create a reliable, scalable, and maintainable system that supports the organization's objectives and facilitates seamless information flow.

### **4.4.1 RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)**

A Relational Database Management System (RDBMS) is a pivotal technology in the data management realm, essential for storing and retrieving data in a structured and efficient manner. At its heart, an RDBMS organizes data into tables, making it easy to establish relationships between different data sets. This structure not only facilitates data management but also enhances the integrity and consistency of the data.

One of the standout features of an RDBMS is its use of SQL (Structured Query Language), which allows for a wide range of data operations, from basic retrieval to complex queries. SQL's versatility makes it a powerful tool for data analysis and manipulation.

RDBMSs are also characterized by their adherence to ACID (Atomicity, Consistency, Isolation, Durability) properties, ensuring that data transactions are processed reliably. This robustness is crucial in maintaining data integrity, especially in environments where data is constantly being updated.

Scalability is another key attribute of RDBMSs. They are designed to accommodate growing data needs, scaling up to handle larger datasets and more users as required. This flexibility makes RDBMSs suitable for a wide range of applications, from small businesses to large enterprises.

In summary, RDBMSs play a crucial role in modern data management. Their ability to efficiently store, retrieve, and manage data, combined with the power of SQL for data manipulation and analysis, makes them indispensable in today's data-driven world.

#### 4.4.2 NORMALIZATION

Normalization in the context of database design is a systematic approach to organizing data in a database to reduce redundancy and improve data integrity. The primary goal of normalization is to divide large tables into smaller, interrelated tables and define relationships between them to minimize the duplication of information. This process enhances the efficiency and consistency of the database, making it easier to manage and update.

The concept of normalization is based on a set of principles or "normal forms," each addressing potential issues in table structures that can lead to anomalies and inefficiencies in data handling. Here are the various normal forms commonly used in normalization:

- **First Normal Form (1NF):** This is the most basic level of normalization. A table is in 1NF if it only contains atomic (indivisible) values and each row in the table is unique. Essentially, 1NF requires that a database table has a primary key and that its columns contain only individual, undivided values.
- **Second Normal Form (2NF):** A table is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key. This means that there is no partial dependency of any column on the primary key. 2NF eliminates redundancy by ensuring that each piece of data is stored in only one place.
- **Third Normal Form (3NF):** A table is in 3NF if it is in 2NF and all its attributes are not only fully functionally dependent on the primary key but also non-transitively dependent. In simpler terms, no non-key attribute should depend on another non-key attribute. This level further eliminates redundancy and ensures that each piece of information is stored only once.
- **Boyce-Codd Normal Form (BCNF):** A refinement of 3NF, a table is in BCNF if, for every one of its non-trivial functional dependencies,  $X \rightarrow Y$ ,  $X$  is a superkey – a set of attributes that uniquely identifies a row. BCNF addresses certain types of anomalies not covered by 3NF.

- **Fourth Normal Form (4NF):** This level addresses multi-valued dependencies. A table is in 4NF if it is in BCNF and has no multi-valued dependencies, which are dependencies where one key determines multiple values of two other independent columns.
- **Fifth Normal Form (5NF):** Also known as "Projection-Join Normal Form" (PJNF), a table is in 5NF if it is in 4NF and every join dependency in the table is a consequence of its candidate keys. This form deals with cases where information can be reconstructed from smaller pieces of data.

### 4.4.3 SANITIZATION

Data sanitization refers to the process of deliberately and securely removing or destroying data stored on a memory device to ensure that it cannot be retrieved or reconstructed. This practice is crucial, especially when disposing of or repurposing storage devices, as it protects sensitive information from unauthorized access and potential misuse.

Sanitization is particularly important in a world where data breaches and information theft are prevalent. Businesses, organizations, and individuals often handle sensitive data, including personal information, financial records, and confidential corporate data. When these data-bearing devices reach the end of their life or are scheduled for reuse, simply deleting files or formatting the storage medium is insufficient. This is because standard deletion methods typically remove only the pointers to the data and not the actual data, leaving it vulnerable to recovery through various software tools.

There are several methods of data sanitization, each suitable for different types of storage media and levels of data sensitivity:

- **Physical Destruction:** This is a straightforward approach where the physical device is destroyed using shredders, incineration, or other destructive methods. Physical destruction is often used for highly sensitive data, ensuring that the storage medium is completely unreadable and irreparable.
- **Degaussing:** This method involves using a high-powered magnet (degausser) to disrupt the magnetic field of the storage medium, effectively destroying the data stored on magnetic media like hard drives and tapes. However, degaussing is not effective on solid-state drives (SSDs).

- **Overwriting:** Also known as data wiping, this method involves writing new data over the existing data. This can be done multiple times to ensure the original data is completely unrecoverable. Overwriting is a common sanitization method as it allows the storage medium to be reused.
- **Cryptographic Erasure:** In this method, data is encrypted, and then the encryption key is destroyed, making the data inaccessible. This is an efficient method for devices that were encrypted from the start.

#### 4.4.4 INDEXING

Indexing in the context of databases is a critical technique used to optimize the performance of a database by minimizing the amount of disk access required when a database is queried. Much like an index in a book that allows you to quickly find specific information, an index in a database provides a fast pathway to access the data that users are looking for.

When a database does not use indexing, retrieving information can be a slow and inefficient process, especially for large databases. This is because the database management system needs to scan through each row of a table (a process known as a full table scan) to find the relevant data. This can be time-consuming and resource-intensive.

By implementing indexes, databases create an internal structure that allows the system to quickly locate the data without scanning every row. An index is typically created on one or more columns (fields) in a database table. When a query is performed using these columns, the database uses the index to rapidly narrow down the rows that need to be examined, significantly reducing the search time.

There are several types of indexes in databases, including:

- **Single-Column Indexes:** Created on a single column in a table. They are effective when queries are commonly performed on that particular column.
- **Composite Indexes:** Involve multiple columns. They are useful when queries frequently use several columns.
- **Unique Indexes:** Ensure that the index key contains only unique values. This is important for maintaining data integrity.

- **Full-Text Indexes:** Designed for efficient searching of text within columns. They are used in queries that perform complex searches on strings.

While indexing significantly improves query performance, it's not without its trade-offs. Creating and maintaining indexes requires additional storage space. Also, indexes can slightly slow down data insertion, update, and deletion operations, because the indexes themselves need to be updated whenever the data changes.

Furthermore, the choice of which indexes to create requires careful consideration. Over-indexing (creating too many indexes) or creating the wrong type of index can lead to inefficiency. Therefore, database administrators need to balance between the optimal performance during queries and the overhead cost of maintaining the indexes.

## 4.5 TABLE DESIGN

### 1. Table name: Tbl\_users\_login

Primary key: id

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the user
username	VARCHAR(150)	Unique	The user's login username
email	VARCHAR(254)	Unique	The user's email address
password	VARCHAR(128)	Not Null	The user's hashed password
is_active	BOOLEAN	Not Null	Whether the user's account is active
is_normaluser	BOOLEAN	Not Null	Whether the user is a normal user (not a college user)

is_collegeuser	BOOLEAN	Not Null	Whether the user is associated with a college
date_joined	DATETIME	Not Null	The date and time when the account was created

## 2. Table name: Tbl\_NormalUser

Primary key: id

Foreign key: user\_id references Tbl\_users\_login

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for NormalUser
user_id	INT	Foreign Key	References id from Tbl_users_login
first_name	VARCHAR(30)	Not Null	First name of the user
last_name	VARCHAR(30)	Not Null	Last name of the user
phone_number	VARCHAR(15)	Not Null	User's phone number
country	VARCHAR(100)	Not Null	Country of the user
gender	VARCHAR(10)	Not Null	Gender of the user
profile_photo	VARCHAR(255)	Not Null	Path to the user's profile photo
cover_photo	VARCHAR(255)	Not Null	Path to the user's cover photo
about_me	TEXT	Not Null	User's biography
registration_date	DATETIME	Not Null	Date and time the user registered

**3. Table name: Tbl\_CollegeUser**

Primary key: id

Foreign key: user\_id references Tbl\_users\_login

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for CollegeUser
user_id	INT	Foreign Key	References id from Tbl_users_login
college_name	VARCHAR(255)	Not Null	Name of the college
address	TEXT	Not Null	Physical address of the college
website	VARCHAR(2048)	Not Null	Website URL of the college
contact_email	VARCHAR(254)	Not Null	Contact email of the college
contact_phone_number	VARCHAR(15)	Not Null	Contact phone number of the college
college_type	VARCHAR(15)	Not Null	Type of the college
profile_photo	VARCHAR(255)	Not Null	Path to the college's profile photo
cover_photo	VARCHAR(255)	Not Null	Path to the college's cover photo

**4. Table name: Tbl\_Department**

Primary key: id

Foreign key: college\_id references Tbl\_CollegeUser

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the department
college_id	INT	Foreign Key	References id from Tbl_CollegeUser
name	VARCHAR(255)	Unique	Name of the department
undergrad_offered	BOOLEAN	Not Null	Whether undergraduate programs are offered
postgrad_offered	BOOLEAN	Not Null	Whether postgraduate programs are offered
head_of_department	VARCHAR(255)	Not Null	Name of the head of the department
department_start_year	DATE	Not Null	The year when the department was started
is_active	BOOLEAN	Not Null	Whether the department is currently active



**5. Table name: Tbl\_Instructor**

Primary key: id

Foreign key: college\_id references Tbl\_CollegeUser, department\_id references  
Tbl\_Departement

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the instructor
college_id	INT	Foreign Key	References id from Tbl_CollegeUser
department_id	INT	Foreign Key	References id from Tbl_Department
instructor_name	VARCHAR(255)	Not Null	Name of the instructor
subject_taught	VARCHAR(255)	Not Null	Subject taught by the instructor, nullable
qualification	VARCHAR(255)	Not Null	Qualification of the instructor, nullable
profile_photo	VARCHAR(255)	Not Null	Path to the instructor's profile photo, nullable

**6. Table name: Tbl\_Course**

Primary key: course\_id

Foreign key: college\_id references Tbl\_CollegeUser, department\_id references  
Tbl\_Departement

Field name	Datatype	Key Constraints	Description
course_id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the course
college_id	INT	Foreign Key	References id from Tbl_CollegeUser
department_id	INT	Foreign Key	References id from Tbl_Department
course_name	VARCHAR(255)	Not Null	Name of the course
course_duration	INT	Not Null	Duration of the course in days
course_fee	DECIMAL(10, 2)	Not Null	Fee for the course
course_description	TEXT	Not Null	Description of the course
languages	VARCHAR(255)	Not Null	Languages in which the course is offered
course_level	VARCHAR(20)	Not Null	Level of the course (beginner, intermediate, expert)
certificate_available	BOOLEAN	Not Null	Whether a certificate is available for the course
exam	BOOLEAN	Not Null	Whether the course has an exam
assignment	BOOLEAN	Not Null	Whether the course has assignments
exam_types	VARCHAR(255)	Not Null	Types of exams in the course, if any
certificate_criteria	TEXT	Not Null	Criteria for obtaining a certificate, if available
total_assignments	INT	Not Null	Total number of assignments in the course
course_tags	VARCHAR(255)	Not Null	Tags associated with the course

cover_photo	VARCHAR(255)	Not Null	Path to the course's cover photo
-------------	--------------	----------	----------------------------------

### 7. Table name: Tbl\_Course\_Instructor

Foreign key: course\_id references Tbl\_Course, instructor\_id references Tbl\_Instructor

Field name	Datatype	Key Constraints	Description
course_id	INT	Foreign Key	References course_id from Tbl_Course
instructor_id	INT	Foreign Key	References id from Tbl_Instructor

### 8. Table name: Tbl\_Module

Primary key: module\_id

Foreign key: course\_id references Tbl\_Course

Field name	Datatype	Key Constraints	Description
module_id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the module
course_id	INT	Foreign Key	References course_id from Tbl_Course
module_name	VARCHAR (255)	Not Null	Name of the module

**9. Table name: Tbl\_Chapter**

Primary key: chapter\_id

Foreign key: module\_id references Tbl\_Module

Field name	Datatype	Key Constraints	Description
chapter_id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the chapter
module_id	INT	Foreign Key	References module_id from Tbl_Module
chapter_name	VARCHAR(255)	Not Null	Name of the chapter

**10. Table name: Tbl\_ReadingMaterial**

Primary key: id

Foreign key: chapter\_id references Tbl\_Chapter

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the reading material
chapter_id	INT	Foreign Key	References chapter_id from Tbl_Chapter
title	VARCHAR(255)	Not Null	Title of the reading material
text_content	TEXT	Not Null	Text content of the reading material
images	VARCHAR(255)	Not Null	Path to the images associated with the material

**11. Table name: Tbl\_VideoMaterial**

Primary key: id

Foreign key: chapter\_id references Tbl\_Chapter

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the video material
chapter_id	INT	Foreign Key	References chapter_id from Tbl_Chapter
video	VARCHAR(255)	Not Null	Path to the video file
transcript	TEXT	Not Null	Transcript of the video

**12. Table name: Tbl\_MultipleChoiceQuestion**

Primary key: id

Foreign key: chapter\_id references Tbl\_Chapter

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the multiple choice question
chapter_id	INT	Foreign Key	References chapter_id from Tbl_Chapter
question_text	TEXT	Not Null	Text of the multiple-choice question

choice_1	VARCHAR(200)	Not Null	Text of the first choice
choice_2	VARCHAR(200)	Not Null	Text of the second choice
choice_3	VARCHAR(200)	Not Null	Text of the third choice
choice_4	VARCHAR(200)	Not Null	Text of the fourth choice
correct_answer	VARCHAR(200)	Not Null	Text of the correct answer

### 13. Table name: Tbl\_Post

Primary key: post\_id

Foreign key: college\_user\_id references Tbl\_CollegeUser

Field name	Datatype	Key Constraints	Description
post_id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the post
college_user_id	INT	Foreign Key	References id from Tbl_CollegeUser
content	TEXT	Not Null	Text content of the post
image	VARCHAR(255)	Not Null	Path to the image associated with the post
video	VARCHAR(255)	Not Null	Path to the video associated with the post
created_at	DATETIME	Not Null	Date and time when the post was created

**14. Table name: Tbl\_Post\_Likes**

Foreign key: post\_id references Tbl\_Post, user\_id references Tbl\_user\_login

Field name	Datatype	Key Constraints	Description
post_id	INT	Foreign Key	References post_id from Tbl_Post
user_id	INT	Tbl_user_login	References id from Tbl_user_login

**15. Table name: Tbl\_SavedPost**

Foreign key: post\_id references Tbl\_Post, user\_id references Tbl\_user\_login

Field name	Datatype	Key Constraints	Description
post_id	INT	Foreign Key	References post_id from Tbl_Post
user_id	INT	Foreign Key	References id from Tbl_user_login

**16. Table name: Tbl\_Progress**

Primary key: id

Foreign key: chapter\_id references Tbl\_Chapter, user\_id references Tbl\_user\_login

Field name	Datatype	Key Constraints	Description
id	INT AUTO_INCREMENT	Primary Key	Unique identifier for the progress record
user_id	INT	Foreign Key	References id from Tbl_user_login

---

chapter_id	INT	Foreign Key	References chapter_id from Tbl_Chapter
progress	BOOLEAN	Not Null	Indicates if the chapter is completed



## **CHAPTER 5**

### **SYSTEM TESTING**

## 5.1 INTRODUCTION

System Testing is a crucial phase in the software development life cycle, where the entire system is evaluated against specified requirements and functionalities. It is a comprehensive and structured approach to validate that the software meets its intended objectives. This phase involves testing the integrated system as a whole to ensure that all components work together seamlessly. System Testing verifies the system's compliance with both functional and non-functional requirements, including performance, security, and usability. It is conducted in an environment that closely simulates the production environment, providing a real-world scenario for testing. The primary goal of System Testing is to identify and rectify any discrepancies or defects before the software is deployed to end-users. Through rigorous testing processes and thorough documentation, System Testing helps in delivering a reliable and high-quality software product.

Testing is the systematic process of running a program to uncover potential errors or flaws. An effective test case possesses a high likelihood of revealing previously unnoticed issues. A test is considered successful when it reveals a previously unidentified error. If a test functions as intended and aligns with its objectives, it can detect flaws in the software. The test demonstrates that the computer program is operating in accordance with its intended functionality and performing optimally. There are three primary approaches to assessing a computer program: evaluating for accuracy, assessing implementation efficiency, and analyzing computational complexity.

## 5.2 TEST PLAN

A test plan is a thorough document that delineates the strategy, scope, objectives, resources, schedule, and expected outcomes for a specific testing endeavor. It functions as a guiding framework for carrying out testing activities, guaranteeing that every facet of the testing process is methodically organized and executed. Additionally, the test plan establishes the roles and responsibilities of team members, outlines the required testing environment, and sets forth the criteria for the successful completion of testing activities. This document plays a pivotal role in ensuring that the testing phase is conducted in a structured and effective manner, ultimately contributing to the overall success of the project.

The levels of testing include:

- Integration Testing
- Unit testing
- Validation Testing or System Testing

- Output Testing or User Acceptance Testing
- Automation Testing
- Widget Testing

### **5.2.1 UNIT TESTING**

Unit Testing is not only a meticulous examination of discrete units or components within a software system but also an indispensable quality assurance measure. This phase serves as a crucial foundation for the entire software testing process, where the focus lies on isolating and scrutinizing individual units of code. The objective remains unwavering: to verify that each unit performs its designated function accurately, yielding precise outputs for predefined inputs.

Moreover, Unit Testing operates independently, detached from other components, and any external dependencies are either emulated or replaced by "mock" objects, ensuring controlled evaluation. This meticulous process establishes a robust foundation for the software, confirming that each unit functions reliably and adheres meticulously to its predefined behavior.

The significance of Unit Testing cannot be overstated, as it acts as a vanguard against potential discrepancies or errors early in the development cycle. This proactive approach not only fortifies the integrity and reliability of the software but also lays the groundwork for subsequent testing phases, thereby fostering a robust and dependable software solution. This meticulous process ensures that each unit functions reliably and adheres precisely to its defined behavior. By subjecting individual code units to rigorous scrutiny, any discrepancies or errors are identified and rectified early in the development cycle, bolstering the overall integrity and reliability of the software.

### **5.2.2 INTEGRATION TESTING**

Integration Testing stands as a pivotal phase in the software testing process, dedicated to scrutinizing the interactions and interfaces among diverse modules or components within a software system. Its primary objective is to ascertain that individual units of code seamlessly converge to create a unified and functional system. In stark contrast to unit testing, which assesses individual units in isolation, integration testing delves into the interplay between these units, with a keen eye for any disparities, communication glitches, or integration hurdles. By subjecting the integrated components to rigorous testing, development teams aim to affirm that these elements function

cohesively, addressing any potential issues before deployment. This systematic evaluation is instrumental in ensuring that the software operates as an integrated whole, free from any unforeseen conflicts or errors that may arise from the convergence of individual modules.

### 5.2.3 VALIDATION TESTING OR SYSTEM TESTING

Validation Testing places the end-users at the forefront of evaluation, ensuring that the software aligns precisely with their anticipated needs and expectations. This phase stands distinct from other testing methodologies, as its primary objective is to authenticate that the software, in its final form, serves its intended purpose seamlessly within the real-world scenarios it was designed for. As a culmination of the testing process, Validation Testing carries the responsibility of confirming that the software not only meets the defined technical specifications but also delivers genuine value to its users. It does so by scrutinizing the software against the backdrop of actual usage, thereby fortifying its readiness for deployment. Moreover, in Validation Testing, user stories and acceptance criteria form the cornerstone of assessment. Stakeholders' expectations are meticulously validated, ensuring that every specified requirement is met. Additionally, beta testing, a common practice in this phase, involves a select group of end-users testing the software in a live environment, providing invaluable feedback that can inform potential refinements.

### 5.2.4 OUTPUT TESTING OR USER ACCEPTANCE TESTING

Output Testing, also known as Results Validation, is a critical phase in the software testing process. Its primary focus is to verify the correctness and accuracy of the output generated by a software application. The goal is to ensure that the system produces the expected results for a given set of inputs and conditions.

Key aspects of Output Testing include:

- **Comparison with Expected Results:** This phase involves comparing the actual output of the software with the expected or predefined results.
- **Test Case Design:** Test cases are designed to cover various scenarios and conditions to thoroughly evaluate the accuracy of the output.
- **Validation Criteria:** The criteria for validating the output are typically defined during the requirements and design phase of the software development process.

- **Regression Testing:** Output Testing often includes regression testing to ensure that changes or updates to the software do not affect the correctness of the output.
- **Data Integrity:** It verifies that data is processed and displayed correctly, without any corruption or loss.
- **Precision and Completeness:** Output Testing assesses not only the precision of the results but also their completeness in addressing the requirements.
- **Error Handling:** It evaluates how the system handles errors or exceptions and ensures that appropriate error messages are displayed.

### 5.2.5 AUTOMATION TESTING

Automation Testing stands as a cornerstone in the software testing process, harnessing the power of automated tools and scripts to meticulously execute test cases. In stark contrast to manual testing, which hinges on human intervention, automation testing brings forth a streamlined approach, employing software to conduct repetitive, intricate, and time-consuming tests. This methodology not only heightens operational efficiency but also significantly diminishes the likelihood of human error, ensuring precise and reliable results. Moreover, it empowers thorough testing across a diverse array of scenarios and configurations, from browser compatibility to load and performance assessments.

By automating the testing process, organizations can realize a myriad of benefits. It enables the seamless execution of regression tests, providing confidence that existing functionalities remain intact after each round of enhancements or modifications. Furthermore, automation facilitates the concurrent execution of multiple tests, thereby expediting the overall testing cycle. This approach is particularly invaluable in environments characterized by rapid development and frequent software updates, such as Agile and DevOps setups.

### 5.2.6 SELENIUM TESTING

Selenium is a popular open-source framework used for automating web browsers, playing a key role in the field of software testing. Primarily, Selenium is used for automating web applications for testing purposes, but it can also automate web-based administrative tasks. The versatility of Selenium stems from its ability to mimic real-user interactions with web browsers, making it an invaluable tool for testing web applications under real-world conditions.

The core advantage of Selenium lies in its ability to support multiple programming languages like Java, C#, Python, and Ruby, allowing testers to write test scripts in the language they are most comfortable with. Additionally, Selenium supports various operating systems (Windows, Mac, Linux) and browsers (Chrome, Firefox, IE, and Safari), ensuring comprehensive test coverage across different environments.

Selenium comes in several flavors, with Selenium WebDriver and Selenium Grid being the most prominent. WebDriver directly communicates with the web browser and uses its native compatibility to automate. On the other hand, Selenium Grid allows for running tests in parallel across different machines and browsers, significantly speeding up the testing process.

Using Selenium for automated testing not only reduces the manual effort involved in repetitive testing but also enhances the accuracy of the tests by eliminating human error. This leads to more reliable, efficient, and faster testing cycles, which is crucial in agile and DevOps environments where quick product releases are essential.

## TEST CASE 1

### Code

```
from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions as EC

import unittest

import time # Import the time module


class Logintest(unittest.TestCase):

    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.implicitly_wait(10)

        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):

        self.driver.quit()

    def test_login(self):

        driver = self.driver

        driver.get(self.live_server_url)

        # Wait and enter the username

        WebDriverWait(driver, 10).until(

            EC.visibility_of_element_located((By.NAME, "username"))

        )

        username = driver.find_element(By.NAME, "username")

        username.send_keys("vishal")

        # Wait and enter the password
```

```
WebDriverWait(driver, 10).until(
    EC.visibility_of_element_located((By.NAME, "passname")))
)

password = driver.find_element(By.NAME, "passname")

password.send_keys("zxcvbnmL@123")

# Click the login button

login_button = driver.find_element(By.ID, "submit")

login_button.click()

# Wait for 5 seconds to view the page

time.sleep(5)

if __name__ == '__main__':

    import unittest

    unittest.main()
```

## SCREENSHOT TEST CASE 1

```
PS D:\project\version9\edusphere> python manage.py test
Found 3 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:63777/devtools/browser/47f72ffc-7885-4939-b158-c04bc8ed67ba
Test scenario 'Correct Credentials' passed.
.
DevTools listening on ws://127.0.0.1:63814/devtools/browser/b403bbc4-013d-41b9-bd4c-f2031ca18807
Test scenario 'No Credentials' passed.
.
DevTools listening on ws://127.0.0.1:63843/devtools/browser/a63681a8-ae1f-4505-b795-c12e7d797381
Test scenario 'Wrong Password' passed.
.
-----
Ran 3 tests in 43.720s

OK
PS D:\project\version9\edusphere> █
```



**TEST REPORT**

Test Case 1					
Project Name: EduSphere Fusion					
Login Test Case					
Test Case ID: Test_1			Test Designed By: Vishal C Viswam		
Test Priority(Low/Medium/High): High			Test Designed Date: 27/11/2023		
Module Name: Login			Test Executed By : Ms. Navyamol K T		
Test Title : Login Scenario			Test Execution Date: 01/12/2023		
Description: Test the User login functionality					
Pre-Condition :User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/ Fail)
1	Navigate to login screen		Navigate to login screen	Navigate to login screen	Pass
2	Trying to login without any credentials		Error message should be displayed	Error message displayed	Pass
4	Enter wrong password then press login	vishal vishal123	Invalid credential message should shown	Invalid credential message shown	Pass
5	Enter valid Email and password	vishal zxcvbnmL@123	No error should displayed	No error occured	Pass
Post-Condition: After successful validation with the database, the user is able to log into their account. User can interact with the buttons and functionalities and reset the password.					

**TEST CASE 2:****Code**

```
from django.test import TestCase

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions as EC

import unittest

import time

class Logintest(unittest.TestCase):

    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.implicitly_wait(10)

        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):

        self.driver.quit()

    def fill_form(self, username="", password=""):

        driver = self.driver

        WebDriverWait(driver, 10).until(

            EC.visibility_of_element_located((By.NAME, "username"))

        )

        driver.find_element(By.NAME, "username").send_keys(username)

        WebDriverWait(driver, 10).until(

            EC.visibility_of_element_located((By.NAME, "passname"))

        )

        driver.find_element(By.NAME, "passname").send_keys(password)
```

```
def open_add_department(self):

    # Open the sidebar

    profile_picture = self.driver.find_element(By.CSS_SELECTOR,
"img[onclick='toggleProfileSidebar()']")

    profile_picture.click()

    # Wait for the sidebar to open and click on "Add Departments"

    WebDriverWait(self.driver, 10).until(

        EC.element_to_be_clickable((By.LINK_TEXT, "Add Departments"))

    )

    self.driver.find_element(By.LINK_TEXT, "Add Departments").click()

def fill_department_form(self, name, programs, hod):

    WebDriverWait(self.driver, 10).until(

        EC.visibility_of_element_located((By.ID, "department_name"))

    )

    self.driver.find_element(By.ID, "department_name").send_keys(name)

    self.driver.find_element(By.ID, "offered_programs").send_keys(programs)

    self.driver.find_element(By.ID, "head_of_department").send_keys(hod)

    submit_button = self.driver.find_element(By.XPATH, "//button[@type='submit']")

    submit_button.click()


def test_add_department_without_data(self):

    self.login_with_credentials("amaljyothi", "zxcvbnmL@123")

    self.open_add_department()

    self.fill_department_form("", "", "")

    # Add checks for validation messages or unsuccessful addition

    print("Test scenario 'Add Department without Data' passed.")


def test_add_existing_department(self):
```

```
self.login_with_credentials("amaljyothi", "zxcvbnmL@123")

self.open_add_department()

self.fill_department_form("MCA", "both", "Dr. Smith")

# Add checks for the error message for existing department

print("Test scenario 'Add Existing Department' passed.")

def test_add_new_department(self):

    self.login_with_credentials("amaljyothi", "zxcvbnmL@123")

    self.open_add_department()

    self.fill_department_form("Integrated MCA", "both", "Dr. Johnson")

    # Add checks for successful addition message

    print("Test scenario 'Add New Department' passed.")

if __name__ == '__main__':

    unittest.main()
```

## SCREENSHOT TEST CASE 2

```
(sample) D:\project\version9\edusphere>python manage.py test
Found 3 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:51170/devtools/browser/973b0136-7b42-4f2f-9261-f8d2131b4257
Test scenario 'Add Department without Data' passed.
.
DevTools listening on ws://127.0.0.1:51216/devtools/browser/7c51b723-0bfd-4f27-964c-4cb40f955afd
Test scenario 'Add Existing Department' passed.
.
DevTools listening on ws://127.0.0.1:51289/devtools/browser/319648a3-5289-4ec6-89c1-fd985fc134a3
Test scenario 'Add New Department' passed.
.
-----
Ran 3 tests in 141.067s

OK
```

**TEST REPORT**

Test Case 2					
Project Name: EduSphere Fusion					
Add Department Test Case					
Test Case ID: Test_2			Test Designed By: Vishal C Viswam		
Test Priority(Low/Medium/High): High			Test Designed Date: 04/12/2023		
Module Name: Add Department			Test Executed By : Ms. Navyamol K T		
Test Title: Adding Department			Test Execution Date: 04/12/2023		
Description: Test the add department functionality					
Pre-Condition: User has valid username and password and there should exist a department with same name.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login screen		Navigate to login screen	Navigate to login screen	Pass
2	Trying to add departments with no values		Error message should be displayed	Error message displayed	Pass
4	Enter the name of the department which already exists.	MCA Undergrad Dr.Smith	Departement already exist message should be displayed	Departement already exist message should be displayed	Pass
5	Enter new department that is unique and adding as a new departement	Integrated MCA Both Dr.Johnson	Department added message should be displayed	Department added message should be displayed	Pass
Post-Condition: After successfully adding the department courses and instructors can be add to that department					

**TEST CASE 3:****Code**

```
from django.test import TestCase

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions as EC

import unittest

import time

from selenium.webdriver.support.ui import Select

class Logintest(unittest.TestCase):

    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.implicitly_wait(10)

        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):

        self.driver.quit()

    def fill_form(self, username="", password="):

        driver = self.driver

        WebDriverWait(driver, 10).until(

            EC.visibility_of_element_located((By.NAME, "username"))

        )

        driver.find_element(By.NAME, "username").send_keys(username)

        WebDriverWait(driver, 10).until(

            EC.visibility_of_element_located((By.NAME, "passname"))

        )
```

```
driver.find_element(By.NAME, "passname").send_keys(password)

def login_with_credentials(self, username, password):

    self.driver.get(self.live_server_url)

    self.fill_form(username, password)

    self.driver.find_element(By.ID, "submit").click()

def open_add_instructor_page(self):

    # Log in and open the sidebar to click on "Add Instructor"

    self.login_with_credentials("amaljyothi", "zxcvbnmL@123")

    profile_picture = self.driver.find_element(By.CSS_SELECTOR,
"img[onclick='toggleProfileSidebar()']")

    profile_picture.click()

    WebDriverWait(self.driver, 10).until(

        EC.element_to_be_clickable((By.LINK_TEXT, "Add Instructors"))

    )

    self.driver.find_element(By.LINK_TEXT, "Add Instructors").click()

def test_add_instructor_without_data(self):

    self.open_add_instructor_page()

    submit_button = self.driver.find_element(By.XPATH, "//button[@type='submit']")

    submit_button.click()

    # Add checks for validation messages or unsuccessful addition

    print("Test scenario 'Add Instructor without Data' passed. Failed to add instructor without
giving details")

def test_add_instructor_without_department(self):

    self.open_add_instructor_page()

    self.driver.find_element(By.ID, "instructorName").send_keys("John Doe")

    self.driver.find_element(By.ID, "instructorCourses").send_keys("Mathematics")
```

```
submit_button = self.driver.find_element(By.XPATH, "//button[@type='submit']")

submit_button.click()

# Add checks for validation messages or unsuccessful addition

print("Test scenario 'Add Instructor without Department' passed. Failed to add new
instructor without selecting department")

def test_add_instructor_with_all_details(self):

    self.open_add_instructor_page()

    # Wait for and fill in the instructor's name

    WebDriverWait(self.driver, 10).until(

        EC.visibility_of_element_located((By.ID, "instructorName"))

    )

    self.driver.find_element(By.ID, "instructorName").send_keys("Jane Doe")

    # Wait for and fill in the instructor's courses

    WebDriverWait(self.driver, 10).until(

        EC.visibility_of_element_located((By.ID, "instructorCourses"))

    )

    self.driver.find_element(By.ID, "instructorCourses").send_keys("Physics")

    WebDriverWait(self.driver, 10).until(

        EC.visibility_of_element_located((By.ID, "instructorQualification"))

    )

    self.driver.find_element(By.ID, "instructorQualification").send_keys("MCA")

    # Wait for and select the first department from the dropdown

    WebDriverWait(self.driver, 10).until(

        EC.presence_of_element_located((By.ID, "department"))

    )

    department_select = Select(self.driver.find_element(By.ID, "department")) # Ensure this
ID matches your HTML
```



```
department_select.select_by_index(1)

# Submit the form

submit_button = WebDriverWait(self.driver, 10).until(

    EC.element_to_be_clickable((By.XPATH, "//button[@type='submit']")))

)

submit_button.click()

# TODO: Add checks for successful addition message

print("Test scenario 'Add Instructor with All Details' passed. Successfully added the
instructor")

if __name__ == '__main__':

    unittest.main()
```

### SCREENSHOT TEST CASE 3

```
(sample) D:\project\version9\edusphere>python manage.py test
Found 3 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:52445/devtools/browser/3c019095-98fc-4f3b-abac-36f0227efabf
Test scenario 'Add Instructor with All Details' passed. Successfully added the instructor
.
DevTools listening on ws://127.0.0.1:52482/devtools/browser/2fb22f55-0a24-4f2f-bb28-5e74770e4d6a
Test scenario 'Add Instructor without Data' passed. Failed to add instructor without giving details
.
DevTools listening on ws://127.0.0.1:52519/devtools/browser/04fe6e34-fd50-4be7-ac35-aa99da7867a8
Test scenario 'Add Instructor without Department' passed. Failed to add new instructor without selecting department
.
-----
Ran 3 tests in 82.652s

OK

(sample) D:\project\version9\edusphere>
```

**TEST REPORT**

Test Case 3					
Project Name: EduSphere Fusion					
Add Instructors to Department Test Case					
Test Case ID: Test_3			Test Designed By: Vishal C Viswam		
Test Priority(Low/Medium/High): High			Test Designed Date: 04/12/2023		
Module Name: Add Instructors			Test Executed By : Ms. Navyamol K T		
Test Title : Adding Instructor			Test Execution Date: 04/12/2023		
Description: Test the add instructors functionality					
Pre-Condition :User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to login screen		Navigate to login screen	Navigate to login screen	Pass
2	Trying to add instructor with no values		Error message should be displayed	Error message displayed	Pass
4	Trying to add instructor without selecting the department	John Doe Mathematics	Error message should be displayed	Error message displayed	Pass
5	Add new instructor by giving all the details	Integrated MCA John Does Mathematics MCA	Instructor added message should be displayed	Instructor added message should be displayed	Pass

**Post-Condition:** After successfully adding the instructors these instructors can be selected for courses

## TEST CASE 4:

### Code

```
from django.test import TestCase

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions as EC

import unittest

import time

from selenium.webdriver.support.ui import Select

class Logintest(unittest.TestCase):

    def setUp(self):

        self.driver = webdriver.Chrome()

        self.driver.implicitly_wait(10)

        self.live_server_url = 'http://127.0.0.1:8000/'

    def tearDown(self):

        self.driver.quit()

    def fill_form(self, username="", password=""):

        driver = self.driver

        WebDriverWait(driver, 10).until(

            EC.visibility_of_element_located((By.NAME, "username"))

        )

        driver.find_element(By.NAME, "username").send_keys(username)
```

```
WebDriverWait(driver, 10).until(

    EC.visibility_of_element_located((By.NAME, "passname"))

)

driver.find_element(By.NAME, "passname").send_keys(password)

def login_with_credentials(self, username, password):

    self.driver.get(self.live_server_url)

    self.fill_form(username, password)

    self.driver.find_element(By.ID, "submit").click()

def open_add_course_page(self):

    # Log in and open the sidebar to click on "Add Course"

    self.login_with_credentials("amaljyothi", "zxcvbnmL@123")

    profile_picture = self.driver.find_element(By.CSS_SELECTOR,
"img[onclick='toggleProfileSidebar()']")

    profile_picture.click()

    WebDriverWait(self.driver, 10).until(

        EC.element_to_be_clickable((By.LINK_TEXT, "Add Courses"))

    )

    self.driver.find_element(By.LINK_TEXT, "Add Courses").click()

def test_add_course(self):

    self.open_add_course_page()

    # Fill in the course details

    self.driver.find_element(By.NAME, "course_name").send_keys("Introduction to
Python")

    Select(self.driver.find_element(By.NAME, "department")).select_by_index(1) # Select
the first department

    self.driver.find_element(By.NAME, "course_duration").send_keys("100")

    self.driver.find_element(By.NAME, "course_fee").send_keys("500")
```

```
self.driver.find_element(By.NAME, "course_description").send_keys("This is a course
about Python.")

# Select Languages

# Select Course Level

Select(self.driver.find_element(By.NAME, "course_level")).select_by_value("beginner")

# Submit the form

self.driver.find_element(By.CSS_SELECTOR, "input[type='submit']").click()

# Add checks for successful course addition

print("Test scenario 'Add Course' passed.")

if __name__ == '__main__':

    unittest.main()
```

## SCREENSHOT TEST CASE 4

```
(sample) D:\project\version9\edusphere>python manage.py test
Found 1 test(s).
System check identified no issues (0 silenced).

DevTools listening on ws://127.0.0.1:53092/devtools/browser/ebbed013-6caf-4b76-9732-22253966fa15
[3420:28096:1205/222143.744:ERROR:ssl_client_socket_impl.cc(975)] handshake failed; returned -1, SSL error code 1, net_e
rror -101
[3420:28096:1205/222147.025:ERROR:ssl_client_socket_impl.cc(975)] handshake failed; returned -1, SSL error code 1, net_e
rror -101
Test scenario 'Add Course' passed.
.
-----
Ran 1 test in 50.564s

OK
(sample) D:\project\version9\edusphere>
```

**TEST REPORT**

<b>Test Case 4</b>					
<b>Project Name: EduSphere Fusion</b>					
<b>Add Instructors to Department Test Case</b>					
<b>Test Case ID: Test_4</b>			<b>Test Designed By: Vishal C Viswam</b>		
<b>Test Priority(Low/Medium/High): High</b>			<b>Test Designed Date: 04/12/2023</b>		
<b>Module Name: Add Course</b>			<b>Test Executed By : Ms. Navyamol K T</b>		
<b>Test Title : Adding Course</b>			<b>Test Execution Date: 04/12/2023</b>		
<b>Description: Test the add course functionality</b>					
<b>Pre-Condition :User has valid username and password</b>					
<b>Step</b>	<b>Test Step</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Status(Pass/Fail)</b>
1	Navigate to login screen		Navigate to login screen	Navigate to login screen	Pass
2	Add new course by giving all the details	Introduction to Python MCA 100hr 500rs This is a course about Python beginner	Course added message should be displayed	Course added message should be displayed	Pass
<b>Post-Condition:</b> After successfully adding the course the users can enroll into the course					

## **CHAPTER 6**

### **IMPLEMENTATION**

## 6.1 INTRODUCTION

Project implementation is the phase where plans and strategies transform into tangible actions and outcomes. It marks the transition from theoretical concepts to practical application. This pivotal stage requires meticulous planning, resource allocation, and a dedicated team to execute tasks according to the established timeline and objectives. In this phase, the project team translates the project's blueprints into real-world activities, ensuring that each step aligns with the overarching goals. Effective project implementation demands clear communication, robust leadership, and a keen eye for detail. This introduction sets the stage for a comprehensive understanding of the project implementation process, emphasizing its significance in achieving the envisioned goals. As we delve deeper, we will explore key components, strategies, and best practices that contribute to successful project implementation.

The crux of successful project implementation lies not just in technical proficiency, but also in the art of effective communication. Clear channels of dialogue serve as the lifeblood that sustains the project's momentum, fostering synergy among team members and stakeholders alike. A bedrock of robust leadership provides the necessary guidance and inspiration, steering the ship through uncharted waters with confidence and purpose. Additionally, an unyielding commitment to detail acts as the linchpin that secures the integrity of each executed task.

This introduction sets the stage for a profound comprehension of the project implementation process, underlining its indomitable significance in realizing the envisioned goals. As we embark on this journey of exploration, we will unfurl the tapestry of key components, unveil strategies, and illuminate best practices that form the crucible of triumphant project implementation.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.



## **6.2 IMPLEMENTATION PROCEDURES**

### **6.2.1 USER TRAINING**

User training is a critical component of ensuring the effective utilization of any application software. It involves imparting the necessary knowledge and skills to end-users, enabling them to navigate and utilize the software efficiently. This training equips users with a comprehensive understanding of the software's features, functions, and capabilities. Through hands-on sessions and guided tutorials, users learn how to perform tasks, customize settings, and troubleshoot common issues. Moreover, user training fosters confidence and proficiency, empowering individuals to maximize their productivity while using the application. Regular updates and refresher sessions further enhance user competence, ensuring they stay abreast of new features and functionalities.

### **6.2.2 TRAINING ON THE APPLICATION SOFTWARE**

Training on the application software is a structured program designed to familiarize individuals with the intricacies and functionalities of a specific software application. It encompasses a range of topics, from basic navigation to advanced features, tailored to meet the diverse needs of users. This training often includes interactive demonstrations, hands-on exercises, and Q&A sessions to facilitate effective learning. Trainers may also provide supplemental resources such as user manuals or online guides for reference. By the end of the training, participants are equipped with skills and knowledge required to proficiently utilize the application software in their respective contexts.

### **6.2.3 SYSTEM MAINTENANCE**

System maintenance is a crucial aspect of ensuring the seamless operation and longevity of any software application. It encompasses a series of tasks aimed at monitoring, optimizing, and troubleshooting the underlying infrastructure on which the application runs. This includes activities such as regular software updates, performance monitoring, and data backups. Additionally, system maintenance involves identifying and rectifying any potential vulnerabilities or inefficiencies that may impede the software's performance. Proactive maintenance measures contribute to a stable and secure environment, minimizing the risk of unexpected downtime or data loss.

## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

## 7.1 CONCLUSION

The EduSphere Fusion project stands as a testament to the innovative integration of technology with global education. This ambitious platform, born out of a vision to streamline and enhance the access to worldwide educational resources, aims to bridge the gap between students and top-tier educational institutions. Its development was driven by the need to counter the challenges students face in the realm of foreign education, particularly issues like misinformation and scams.

At its core, EduSphere Fusion offers a multitude of features designed to cater to a diverse array of educational needs. From facilitating online certification courses by reputable foreign colleges to providing a reliable source of updated educational news and events, the platform is a one-stop solution for learners worldwide. Its revenue model, balanced between advertisements and course fees, ensures sustainability while offering value to its users.

The project's commitment to data privacy and user security, coupled with a robust verification system for tutors and colleges, reflects its dedication to authenticity and trust. The platform's design for personalization, particularly through its subscription model, shows an understanding of the varied needs and preferences of learners.

Looking ahead, EduSphere Fusion's plans for expansion and scalability demonstrate foresight and adaptability. The platform's strategy to stay competitive and relevant in the long term, by expanding its offerings and enhancing user engagement, ensures that it will continue to evolve with the changing educational landscape.

In conclusion, EduSphere Fusion is more than just an educational platform; it's a bridge connecting learners with global educational opportunities. Its blend of verified information, user-centric design, and diverse functionalities positions it as a pioneer in the digital education domain. EduSphere Fusion is set to redefine the way students and educators interact with global education, making it more accessible, reliable, and enriching.

## 7.2 FUTURE SCOPE

The future scope of EduSphere Fusion is poised to be both dynamic and innovative, integrating cutting-edge technologies like artificial intelligence (AI) and machine learning (ML) to further revolutionize the educational landscape.

- **Implementing AI and ML for Recommendation Systems:** One of the most exciting prospects is the implementation of AI and ML-driven recommendation systems. These advanced technologies can analyze user data, learning preferences, and past course engagements to provide personalized course suggestions. This not only enhances user experience but also aids in academic growth by aligning course recommendations with individual career goals and learning patterns.
- **Chatbot Integration:** The introduction of AI-powered chatbots can significantly improve user interaction on the platform. These chatbots can provide instant responses to queries, guide users through the platform's features, and offer assistance in course selection and administrative processes. This will ensure a more efficient and engaging user support system, available 24/7.
- **Expanding to IELTS and Competitive Exam Coaching:** Another strategic expansion is incorporating IELTS and other competitive exam coaching into the platform. By offering specialized training modules and practice tests powered by AI algorithms, EduSphere Fusion can cater to a wider audience looking for comprehensive exam preparation. This addition would not only increase the platform's utility but also attract a broader user base.
- **AI Language Training Bot:** Implementing an AI language training bot could revolutionize language learning on the platform. Such a bot can offer personalized language learning experiences, adapt to the learner's pace, and provide interactive, conversational practice - a critical aspect often missing in traditional language learning tools.

Incorporating these advanced technologies will position EduSphere Fusion at the forefront of digital education platforms. The integration of AI and ML not only promises enhanced personalization and efficiency but also opens new avenues for learning and interaction. As the platform evolves, these technologies will be key in maintaining a competitive edge and fulfilling the growing and changing needs of global learners.

## **CHAPTER 8**

### **BIBLIOGRAPHY**

---

**REFERENCES:**

- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Roger S Pressman, “Software Engineering”, 1994.
- PankajJalote, “Software engineering: a precise approach”, 2006.
- Carlo Ginzburg, "Django for Beginners"
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

**WEBSITES:**

- <https://chat.openai.com/>
- <https://docs.djangoproject.com/en/5.0/>
- <https://docs.allauth.org/en/latest/>
- <https://razorpay.com/>
- <https://stackoverflow.com/>
- <https://www.youtube.com/>
- <https://app.diagrams.net/>

## **CHAPTER 9**

## **APPENDIX**

## 9.1 Sample Code

### Views.py

#### Registration and Login

```
def register_normal_user(request):
    error_messages = { }
    if request.method == 'POST':
        fname = request.POST['fname']
        lname = request.POST['lname']
        uname = request.POST['uname']
        email = request.POST['email']
        phone = request.POST['phone']
        password = request.POST['password']
        cpassword = request.POST['cpassword']
        if User.objects.filter(username=uname).exists():
            error_messages['uname'] = 'Username already taken' # Store username error
        if User.objects.filter(email=email).exists():
            error_messages['email'] = 'Email already taken' # Store email error
        if not error_messages: # If there are no errors, proceed with registration
            user = User(username=uname, email=email, is_normal_user=True)
            user.set_password(password)
            #user.is_active=False #make the user inactive
            user.save()
            current_site=get_current_site(request)
            email_subject="Activate your account"
            message=render_to_string('activate.html',{
                'user':user,
                'domain':current_site.domain,
                'uid':urlsafe_base64_encode(force_bytes(user.pk)),
                'token':generate_token.make_token(user)
            })
            email_message=EmailMessage(email_subject,message,settings.EMAIL_HOST_USER,[email],)
```



```
EmailThread(email_message).start()

normal_user = NormalUser(user=user, phone_number=phone, first_name=fname,
last_name=lname)

normal_user.save()

return redirect('loginnew')

return render(request, 'normal_registerpage.html', {'error_messages': error_messages})

def register_college_user(request):
    error_messages = {}
    if request.method == 'POST':
        cname = request.POST['cname']
        uname = request.POST['uname']
        email = request.POST['email']
        phone = request.POST['phone']
        address = request.POST['address']
        password = request.POST['password']
        cpassword = request.POST['cpassword']

        if User.objects.filter(username=uname).exists():
            error_messages['uname'] = 'Username already taken' # Store username error
        if User.objects.filter(email=email).exists():
            error_messages['email'] = 'Email already taken' # Store email error
        if not error_messages:
            user = User(username=uname, email=email, is_college_user=True)
            user.set_password(password)
            user.save()
            college_user = CollegeUser(user=user, college_name=cname, address=address,
contact_email=email, contact_phone_number=phone)
            college_user.save()
            return redirect('loginnew')

        return render(request, 'college_registerpage.html')

def loginnew(request):
```

```
if request.method == 'POST':
    username = request.POST['username']
    password = request.POST['passname']
    user = authenticate(username=username, password=password)
    if user is not None:
        login(request, user)
        if user.is_normal_user:
            messages.success(request, 'You successfully signed in as a normal user.')
            return redirect('home')
        elif user.is_college_user:
            messages.success(request, 'You successfully signed in as a college user.')
            return redirect('college_user_home')
        elif user.is_superuser:
            return redirect('admin_home')
    else:
        return render(request, 'loginpage.html', {'error_message': 'Username or password is incorrect'})
return render(request, 'loginpage.html')
```

### **Add Course, Course Modules, Module Chapters and Chapter Contents**

```
@login_required(login_url='loginnew')
def add_course(request):
    if not request.user.is_authenticated or not request.user.is_college_user:
        return redirect('loginnew')
    if request.method == 'POST':
        course_name = request.POST['course_name']
        course_duration = request.POST['course_duration']
        course_fee = request.POST['course_fee']
        course_description = request.POST['course_description']
        selected_languages = request.POST.getlist('languages')
        selected_exam_types = request.POST.getlist('exam_types')
        selected_course_tags = request.POST.getlist('course_tags')
```

```
course_level = request.POST['course_level']
certificate_available = 'certificate_available' in request.POST
certificate_criteria = request.POST.get('certificate_criteria', '')
exam = 'exam' in request.POST
assignment = 'assignment' in request.POST
total_assignments_str = request.POST.get('total_assignments', '0')
total_assignments = int(total_assignments_str) if total_assignments_str.isdigit() else 0
instructors = request.POST.getlist('instructors')
department_id = request.POST['department']
cover_photo = request.FILES.get('cover_photo')

# Handling the relation with logged in user and department
user = request.user
college_user = CollegeUser.objects.get(user=user)
department = Department.objects.get(pk=department_id)

# Handling the creation of the course
course = Course(
    college=college_user,
    department=department,
    course_name=course_name,
    course_duration=course_duration,
    course_fee=course_fee,
    course_description=course_description,
    languages=",".join(selected_languages),
    exam_types=",".join(selected_exam_types),
    course_tags=",".join(selected_course_tags),
    course_level=course_level,
    certificate_available=certificate_available,
    certificate_criteria=certificate_criteria if certificate_available else "",
    exam=exam,
    assignment=assignment,
```

```
        total_assignments=total_assignments if assignment else 0,
        cover_photo=cover_photo,
    )
    course.save()

    # Handle many-to-many field for instructors
    course.instructors.set(instructors)

    return redirect('add_modules',course_id=course.course_id)
else:
    instructors = Instructor.objects.filter(college__user=request.user)
    departments = Department.objects.filter(college__user=request.user, is_active=True)

    context = {
        'instructors': instructors,
        'departments': departments,
    }
    return render(request, 'course.html', context)

@login_required
def add_modules(request, course_id):
    course = get_object_or_404(Course, course_id=course_id)

    if request.method == 'POST':
        module_names = request.POST.getlist('module_names[]')

        for name in module_names:
            Module.objects.create(course=course, module_name=name)

    return redirect('add_chapters',course_id=course.course_id)

return render(request, 'add_modules.html', {'course': course})
```

```
def add_chapters(request, course_id):
    course = get_object_or_404(Course, pk=course_id)
    modules = course.module_set.all()

    if request.method == 'POST':

        module_ids = request.POST.getlist('module_ids[]')
        num_chapters_list = request.POST.getlist('num_chapters[]')

        for index, module_id in enumerate(module_ids):
            num_chapters = int(num_chapters_list[index])
            chapter_names = request.POST.getlist(f'chapter_names_{module_id}[]')

            if num_chapters == len(chapter_names):
                module = get_object_or_404(Module, pk=module_id)

                for name in chapter_names:
                    Chapter.objects.create(module=module, chapter_name=name)
            else:
                return JsonResponse({'status': 'error', 'message': 'Chapter count mismatch.'})

        return redirect('add_course_material', course_id=course_id)

    else:
        return render(request, 'add_chapters.html', {'course': course, 'modules': modules})

class AddCourseMaterialView(TemplateView):
    template_name = 'add_material.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['course'] = get_object_or_404(Course, pk=self.kwargs['course_id'])
```

```
context['modules'] = Module.objects.filter(course=context['course'])
return context

def add_course_material(request, course_id):
    course = get_object_or_404(Course, pk=course_id)
    completed_modules = request.session.get('completed_modules', [])

    modules = Module.objects.filter(course=course).exclude(module_id__in=completed_modules)

    if request.method == 'POST':
        print(request.POST)
        print(request.FILES)
        module_id = request.POST.get('module_id')
        module = get_object_or_404(Module, module_id=module_id)
        chapters = module.chapters.all()
        for chapter in chapters:
            material_type_key = f'material_type_{chapter.chapter_id}'
            material_type = request.POST.get(material_type_key)
            print(f"Chapter ID: {chapter.chapter_id}, Material Type: {material_type}")
            if not material_type:
                print(f"No material type provided for chapter ID {chapter.chapter_id}")
                continue
            try:
                if material_type == 'reading':
                    title_key = f'reading_title_{chapter.chapter_id}'
                    text_content_key = f'reading_text_content_{chapter.chapter_id}'
                    image_key = f'reading_images_{chapter.chapter_id}'

                    title = request.POST.get(title_key)
                    text_content = request.POST.get(text_content_key)
                    images = request.FILES.get(image_key) if image_key in request.FILES else None
```

```
        ReadingMaterial.objects.create(
            chapter=chapter,
            title=title,
            text_content=text_content,
            images=images
        )
        print(f"Reading material created for Chapter ID: {chapter.chapter_id}")

    elif material_type == 'video':
        video_key = f'video_video_{chapter.chapter_id}'
        transcript_key = f'video_transcript_{chapter.chapter_id}'

        video = request.FILES.get(video_key)
        transcript = request.POST.get(transcript_key)

        VideoMaterial.objects.create(
            chapter=chapter,
            video=video,
            transcript=transcript
        )
        print(f"Video material created for Chapter ID: {chapter.chapter_id}")

    elif material_type == 'multiple_choice':
        # Process each multiple choice question
        question_keys = [key for key in request.POST if
            key.startswith(f'mc_question_text_{chapter.chapter_id}')]

        for question_key in question_keys:
            # Extracting the index using string manipulation
            question_index = question_key.split('[')[-1].rstrip(']')

            # Constructing the keys for choices and the correct answer
            choices_keys = [
```

```
f'mc_choice_1_{chapter.chapter_id}{{question_index}}',
f'mc_choice_2_{chapter.chapter_id}{{question_index}}',
f'mc_choice_3_{chapter.chapter_id}{{question_index}}',
f'mc_choice_4_{chapter.chapter_id}{{question_index}}'
]

# Retrieving the values for choices and the correct answer
choices_values = [request.POST.get(key, "") for key in choices_keys]
correct_answer_key =
f'mc_correct_answer_{chapter.chapter_id}{{question_index}}'
correct_answer_value = request.POST.get(correct_answer_key, "")

# Creating the MultipleChoiceQuestion object
MultipleChoiceQuestion.objects.create(
    chapter=chapter,
    question_text=request.POST[question_key],
    choice_1=choices_values[0],
    choice_2=choices_values[1],
    choice_3=choices_values[2],
    choice_4=choices_values[3],
    correct_answer=correct_answer_value
)

print(f"Multiple choice question {question_index} created for Chapter ID:
{chapter.chapter_id}")

# ... additional elif clauses for other material types ...
except ValidationError as e:
    print(f"Validation Error for chapter {chapter.chapter_id}: {e}")
    return HttpResponse(f"Validation error: {e}", status=400)
except json.JSONDecodeError:
    print("Invalid JSON format for answers or pairs")
    return HttpResponse("Invalid JSON format for answers or pairs", status=400)
completed_modules.append(module_id)
request.session['completed_modules'] = completed_modules
request.session.modified = True
```



```
next_module = modules.exclude(module_id__in=completed_modules).first()

messages.success(request, f"Materials for module '{module.module_name}' saved
successfully.")

if next_module:

    messages.info(request, f"Do you want to add materials to the next module:
'{next_module.module_name}'?")

    redirect_url = f"{reverse('add_course_material',
args=[course_id])}?module_id={next_module.module_id}"
else:

    redirect_url = reverse('course_list_college')

return HttpResponseRedirect(redirect_url)

# If the request method is GET, display the form

return render(request, 'add_material.html', {'course_id': course_id, 'modules': modules})
```

## 9.2 SCREENSHOTS

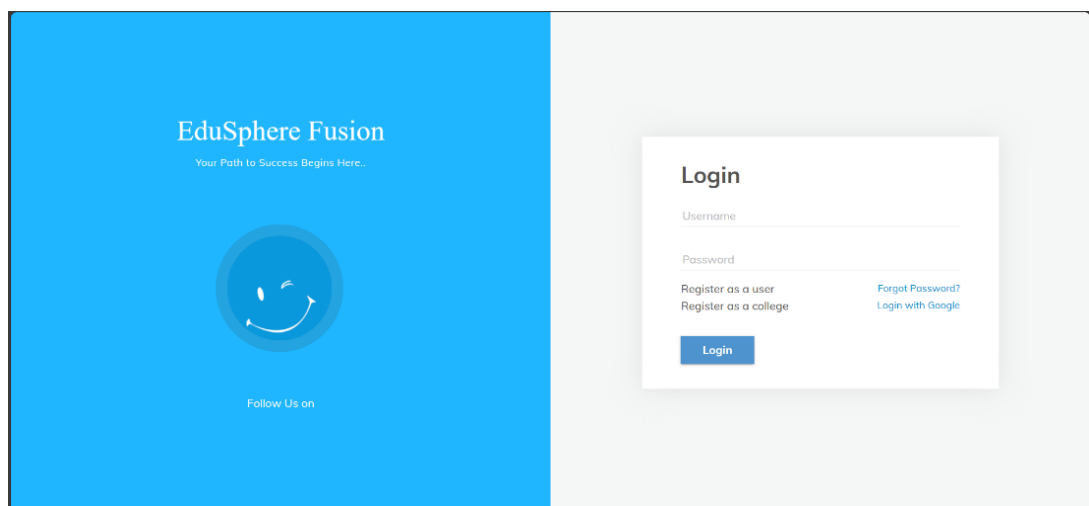
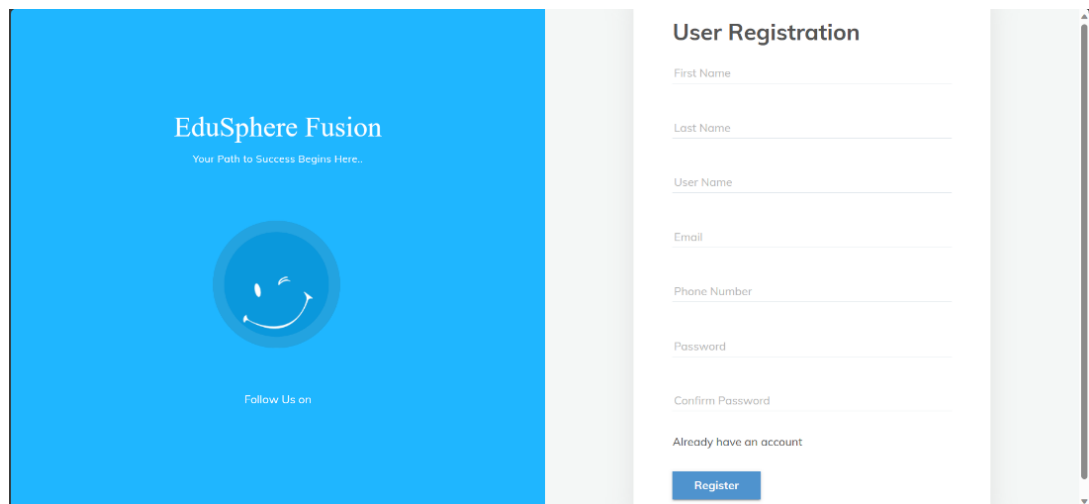
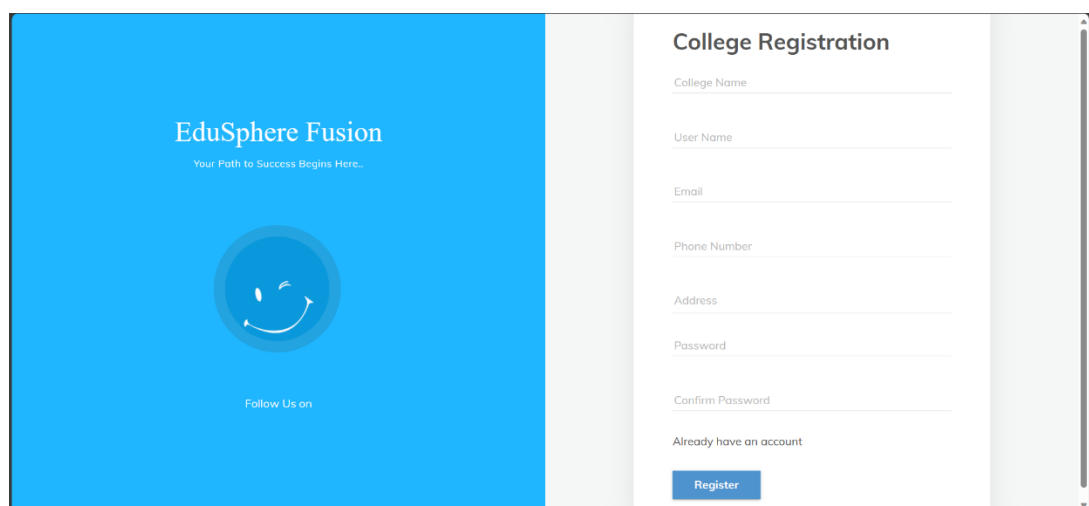


Figure 9.2.1 LOGIN PAGE



The image shows a web interface for user registration. On the left, a blue sidebar contains the 'EduSphere Fusion' logo and the tagline 'Your Path to Success Begins Here..'. Below the logo is a circular icon with a smiley face and the text 'Follow Us on'. The main content area is white and titled 'User Registration'. It contains several input fields: 'First Name', 'Last Name', 'User Name', 'Email', 'Phone Number', 'Password', and 'Confirm Password'. Below these fields is a link 'Already have an account' and a blue 'Register' button.

Figure 9.2.2 USER REGISTRATION



The image shows a web interface for college registration. On the left, a blue sidebar contains the 'EduSphere Fusion' logo and the tagline 'Your Path to Success Begins Here..'. Below the logo is a circular icon with a smiley face and the text 'Follow Us on'. The main content area is white and titled 'College Registration'. It contains several input fields: 'College Name', 'User Name', 'Email', 'Phone Number', 'Address', 'Password', and 'Confirm Password'. Below these fields is a link 'Already have an account' and a blue 'Register' button.

Figure 9.2.2 COLLEGE REGISTRATION



Figure 9.2.3 USER HOME

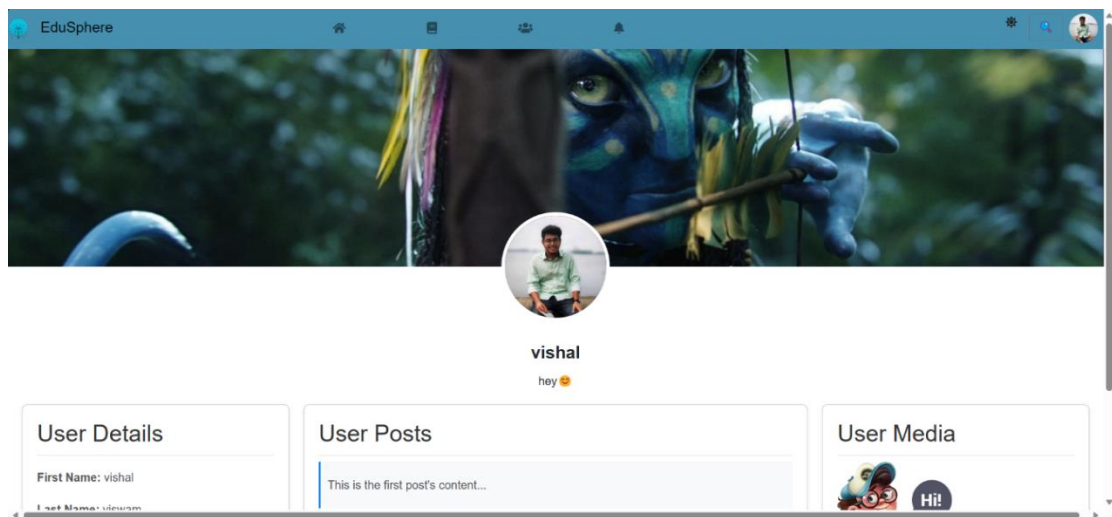


Figure 9.2.4 USER HOME

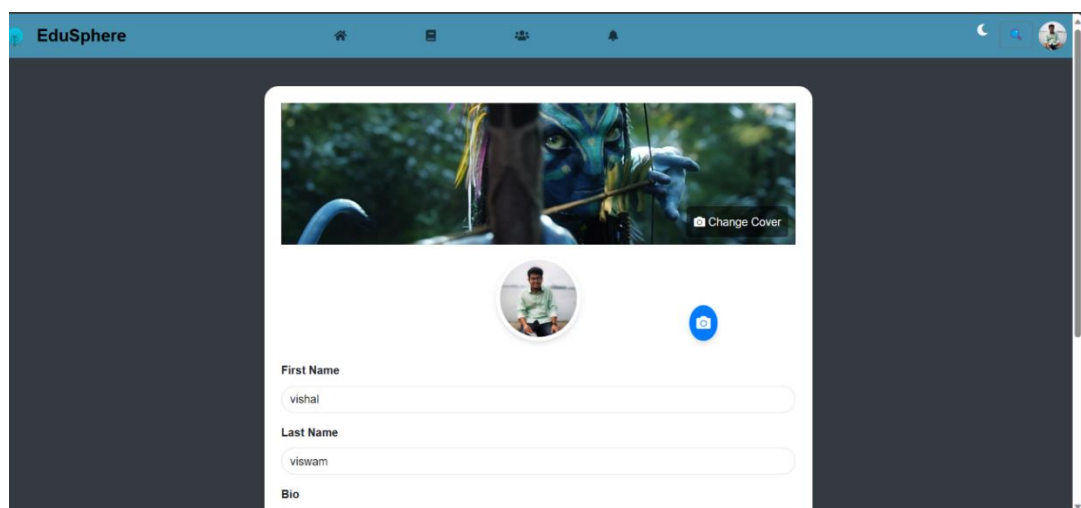


Figure 9.2.5 EDIT PROFILE

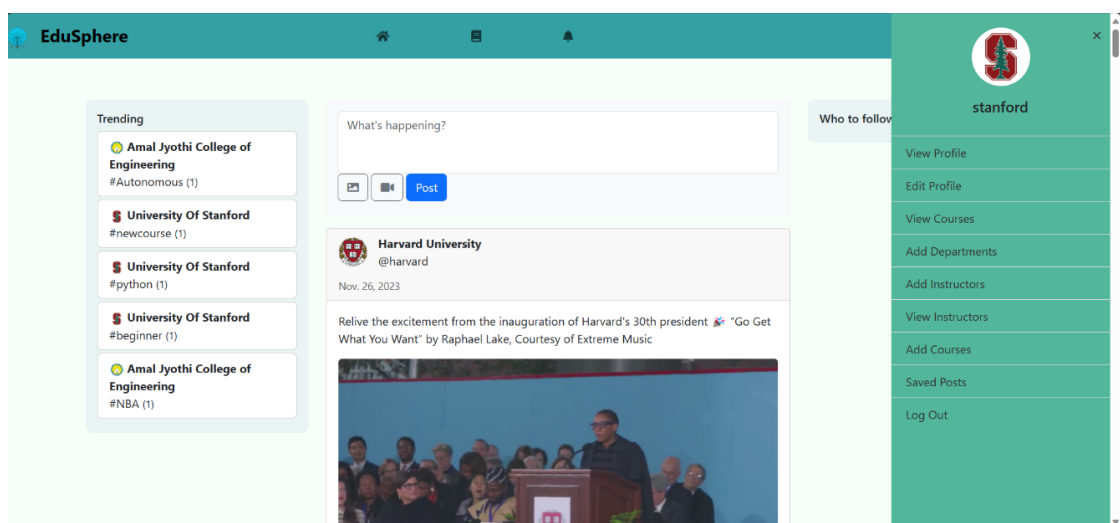


Figure 9.2.6 COLLEGE HOME

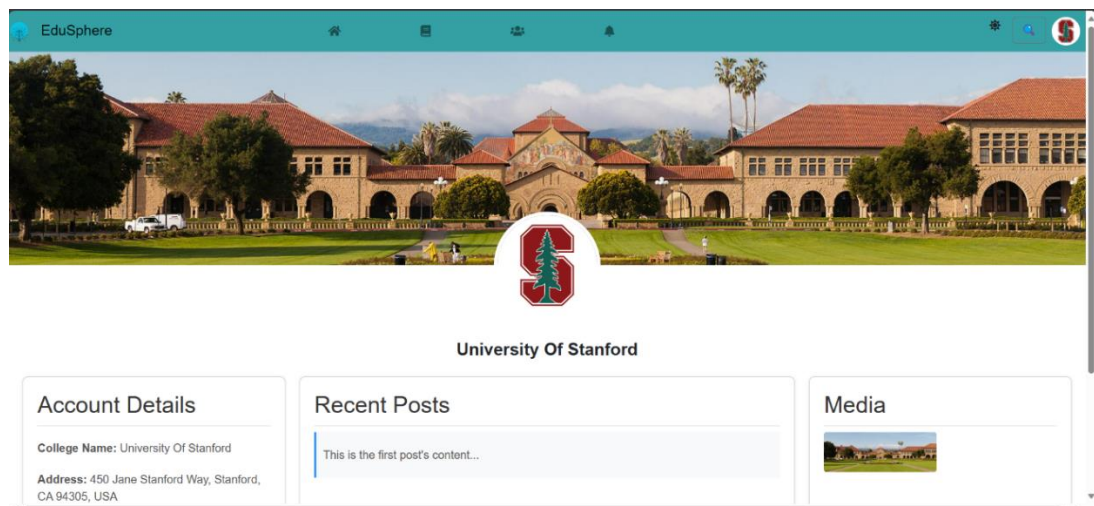


Figure 9.2.7 COLLEGE PROFILE

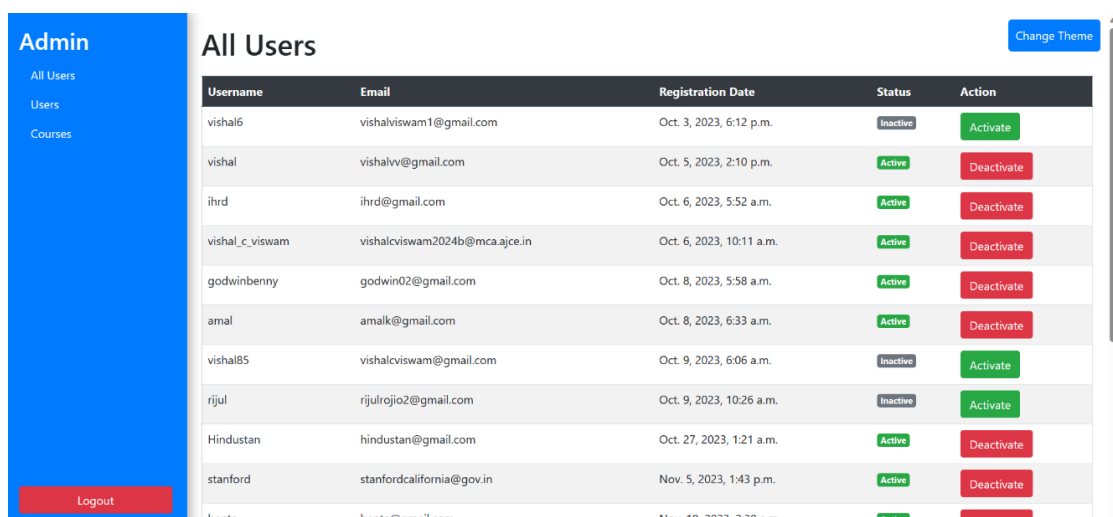


Figure 9.2.8 ADMIN DASHBOARD

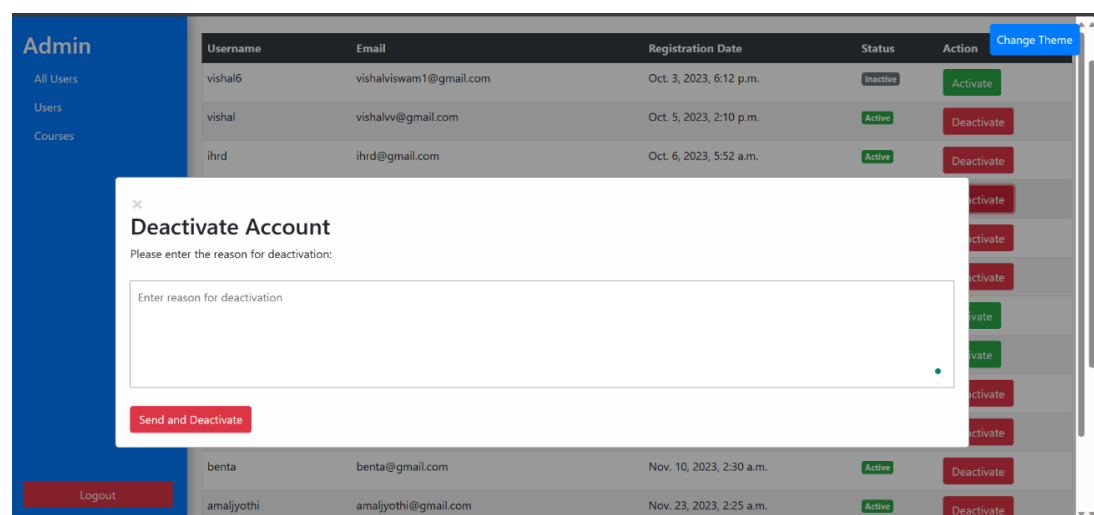
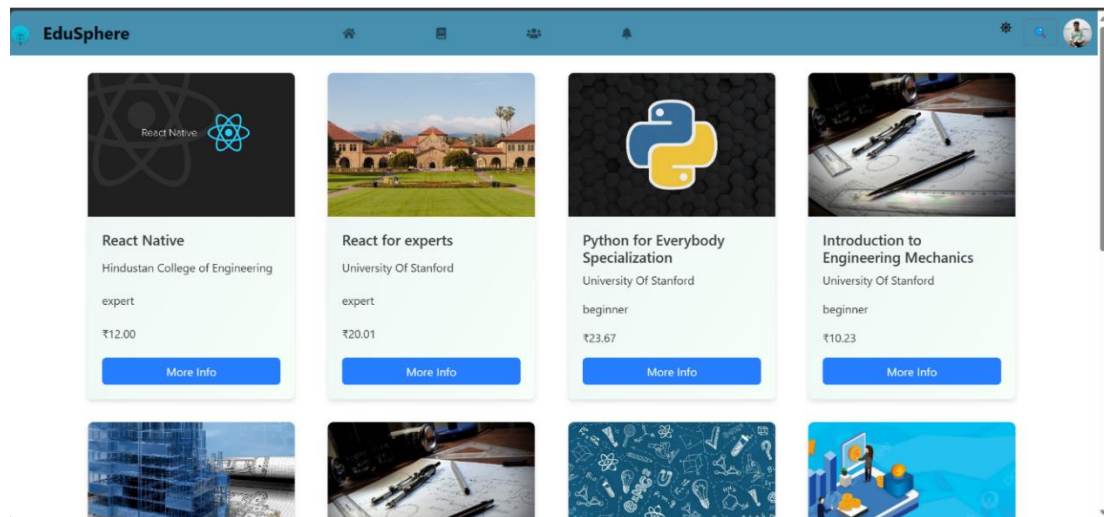
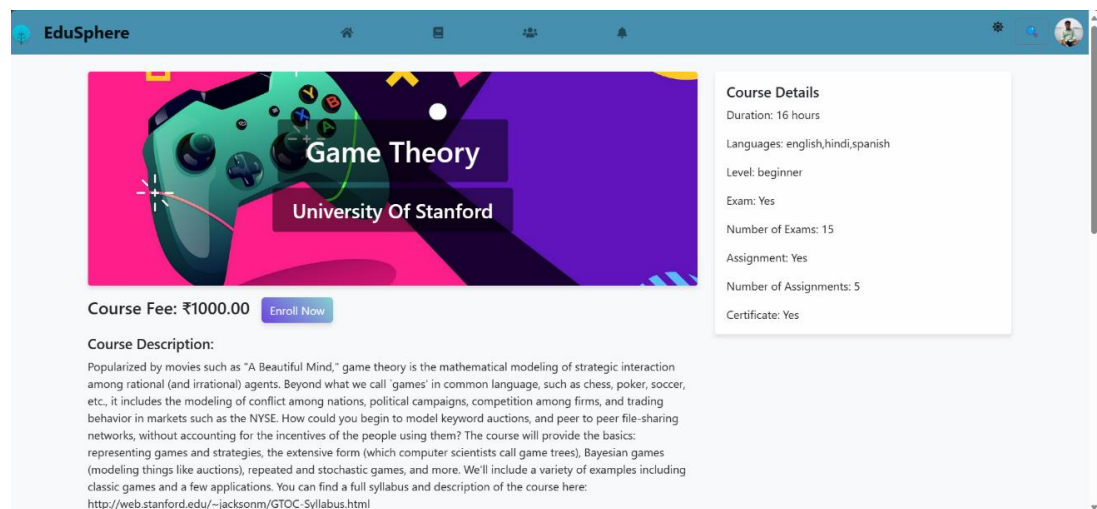


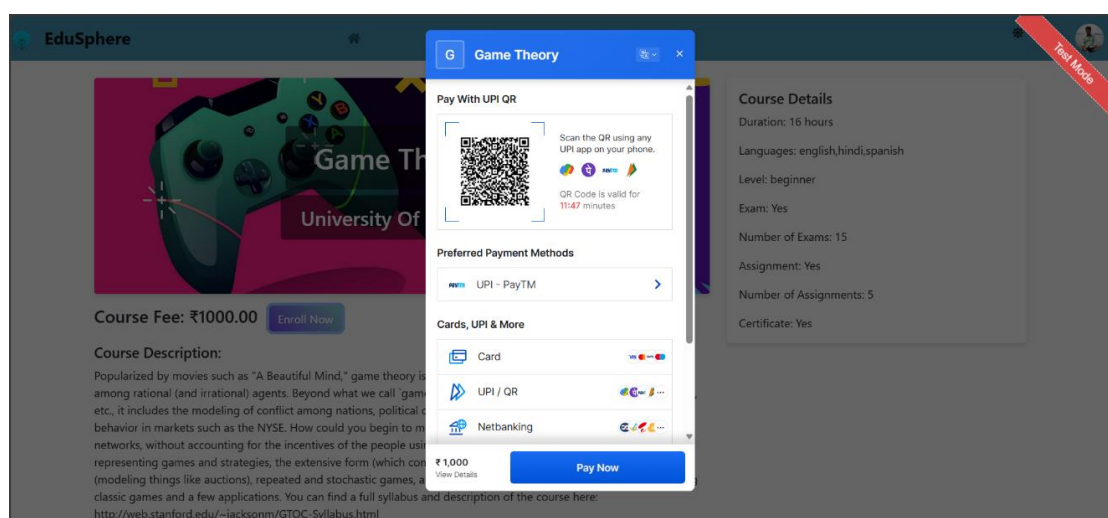
Figure 9.2.9 DEACTIVATE USER ACCOUNT



**Figure 9.2.10 COURSE LIST**



**Figure 9.2.11 COURSE DETAILS**



**Figure 9.2.12 PAYMENT**



Figure 9.2.13 READING MATERIAL

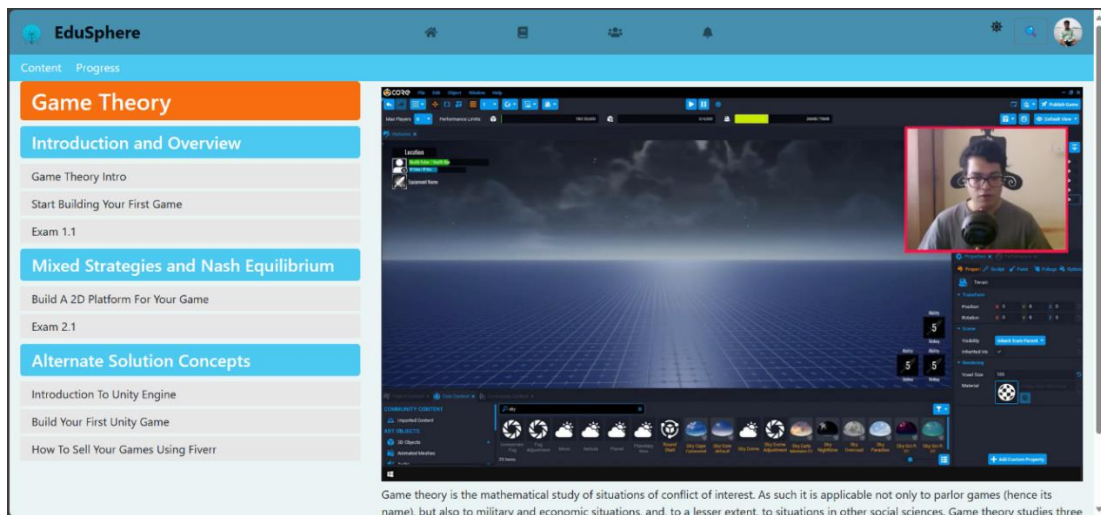


Figure 9.2.14 VIDEO MATERIAL



Figure 9.2.15 EXAM MATERIAL



The screenshot shows the 'Add College Department' form within the EduSphere application. The form is centered on a light gray background. It contains the following fields and controls:

- Department Name:** A text input field with 'Chemical Engineering' entered.
- Offered Programs:** A dropdown menu with 'Undergraduate' selected.
- Head of Department:** A text input field with 'Brenton Tom' entered.
- Add Department:** A blue button at the bottom of the form.

Figure 9.2.16 ADD DEPARTMENT

The screenshot displays the 'College Details' page for the University of Stanford. The page header includes the EduSphere logo and navigation icons. The main content area is titled 'College Details' and 'University Of Stanford', with the location '450 Jane Stanford Way, Stanford, CA 94305, USA'. The page is divided into two main sections:

- Departments:** A list of departments with their offered programs and action buttons:
  - COMPUTER SCIENCE ENGINEERING:** Programs Offered: Undergraduate. Action: Deactivate Department.
  - MECHANICAL ENGINEERING C:** Programs Offered: Postgraduate. Action: Deactivate Department.
  - MASTER OF COMPUTER APPLICATIONS:** Programs Offered: Postgraduate. Action: Deactivate Department.
  - MASTER OF COMPUTER APPLICATIONS12:** Programs Offered: Undergraduate and Postgraduate. Action: Activate Department.
  - MASTER OF COMPUTER APPLICATIONS CC:** Programs Offered: Postgraduate. Action: Activate Department.
- Add Department:** A form on the right side with fields for 'Department Name', 'Offered Programs' (dropdown), and 'Head of Department', followed by an 'Add Department' button.

Figure 9.2.17 VIEW DEPARTMENTS

The screenshot shows the 'Add New Instructor' form in the EduSphere application. The form is titled 'Add New Instructor' and is set against a light gray background. It includes the following fields and controls:

- Department:** A dropdown menu with '--Select Department--' selected.
- Name:** A text input field with the placeholder 'Enter instructor name'.
- Courses Taught:** A text input field with the placeholder 'Enter courses taught'.
- Qualification:** A text input field with the placeholder 'Enter the qualification'.
- Instructor Picture:** A file upload section with a 'Choose File' button and the text 'No file chosen'.
- Add Instructor:** A blue button at the bottom of the form.

Figure 9.2.18 ADD INSTRUCTORS

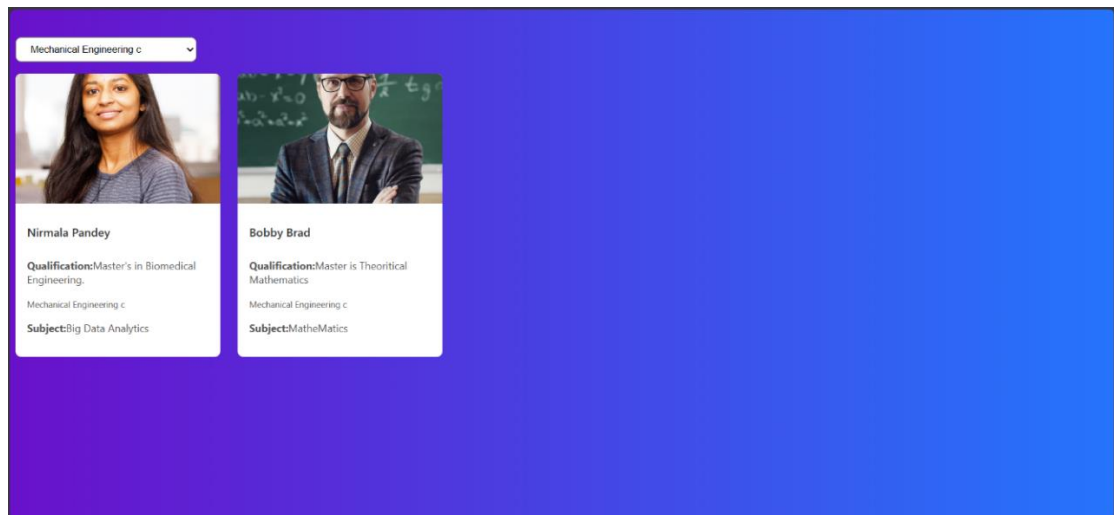


Figure 9.2.19 VIEW INSTRUCTORS

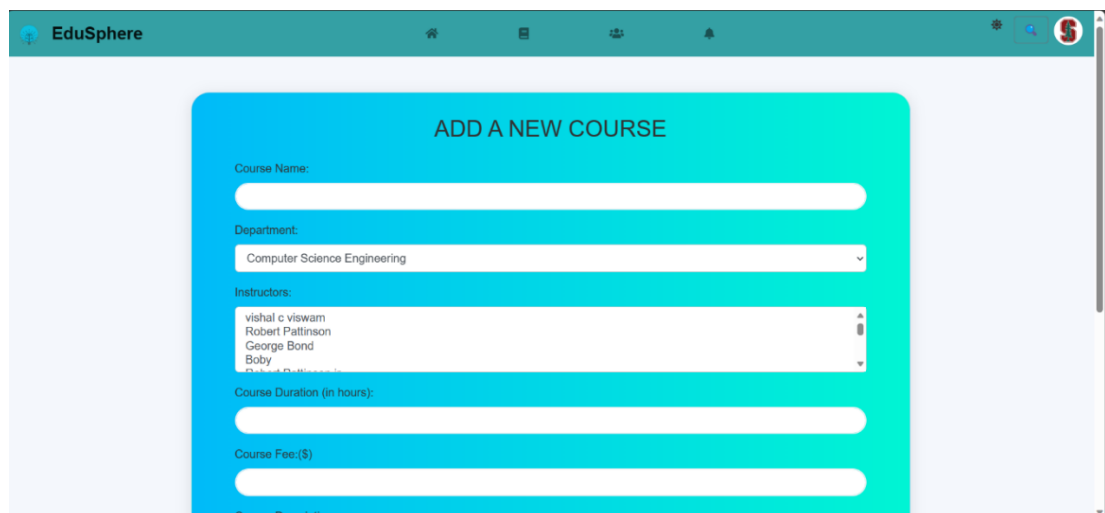


Figure 9.2.20 ADD COURSE DETAILS

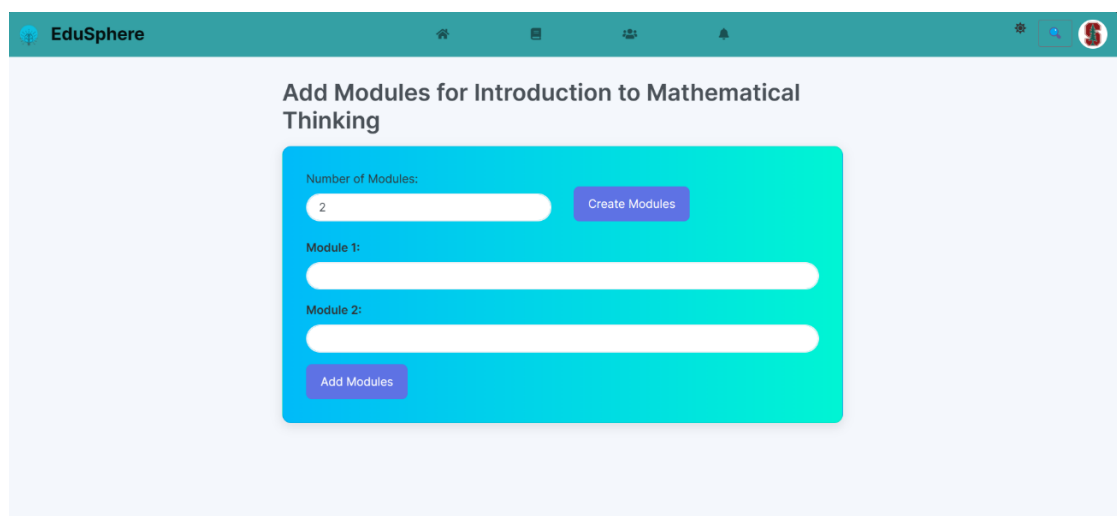


Figure 9.2.21 ADD MODULES



The screenshot shows the 'Add Chapters to Modules for Course: Introduction to Mathematical Thinking' interface. It features two sections for adding chapters to different modules. The first section, 'Add Chapters to: Introduction to MatheMatics', has a 'Number of Chapters' input field. The second section, 'Add Chapters to: More Advanced Maths', also has a 'Number of Chapters' input field. A 'Submit Chapters' button is located at the bottom of the form.

Figure 9.2.22 ADD CHAPTERS

The screenshot shows the 'Add Course Materials' interface. It includes a 'Select Module:' dropdown menu currently set to 'Introduction to MatheMatics'. Below this, there are two sections: 'Addition' and 'Subtraction'. Each section has a 'Select Material Type:' dropdown menu, both currently set to '--Select--'. A 'Save Materials' button is positioned at the bottom of the form.

Figure 9.2.23 ADD COURSE MATERIALS

The screenshot shows the 'Add Reading Materials' interface. It includes a 'Select Module:' dropdown menu currently set to 'Introduction to MatheMatics'. Below this, there are two sections: 'Addition' and 'Subtraction'. The 'Addition' section has a 'Select Material Type:' dropdown menu set to 'Reading Material'. It also includes input fields for 'Title', 'Text Content', and 'Images'. The 'Images' field has a 'Choose File' button and a placeholder text 'No file chosen'. The 'Subtraction' section is partially visible below.

Figure 9.2.24 ADD READING MATERIALS

The screenshot shows the 'Add Video Materials' form in the EduSphere application. At the top, there's a 'Select Module:' dropdown menu with 'Introduction to MatheMatics' selected. Below this is an 'Addition' section with a 'Select Material Type:' dropdown menu set to 'Video Material'. Under the 'Video:' label, there's a 'Choose File' button and a text area showing 'No file chosen'. Below that, under the 'Transcript:' label, there's a text area containing 'Video transcript'. At the bottom of the form, there's a 'Subtraction' section which is currently empty.

Figure 9.2.25 ADD VIDEO MATERIALS

The screenshot shows the 'Add Multiple Choice Questions' form in the EduSphere application. At the top, there's a 'Select Material Type:' dropdown menu set to 'Multiple Choice Questions'. Below this is a 'Question 1:' section with a text area for the question. Underneath, there are four 'Choice' sections, each with a text area for a choice. At the bottom, there's a 'Correct Answer:' section with a text area containing 'correct answer should be one of the choices'.

Figure 9.2.26 ADD MULTIPLE CHOICE QUESTIONS

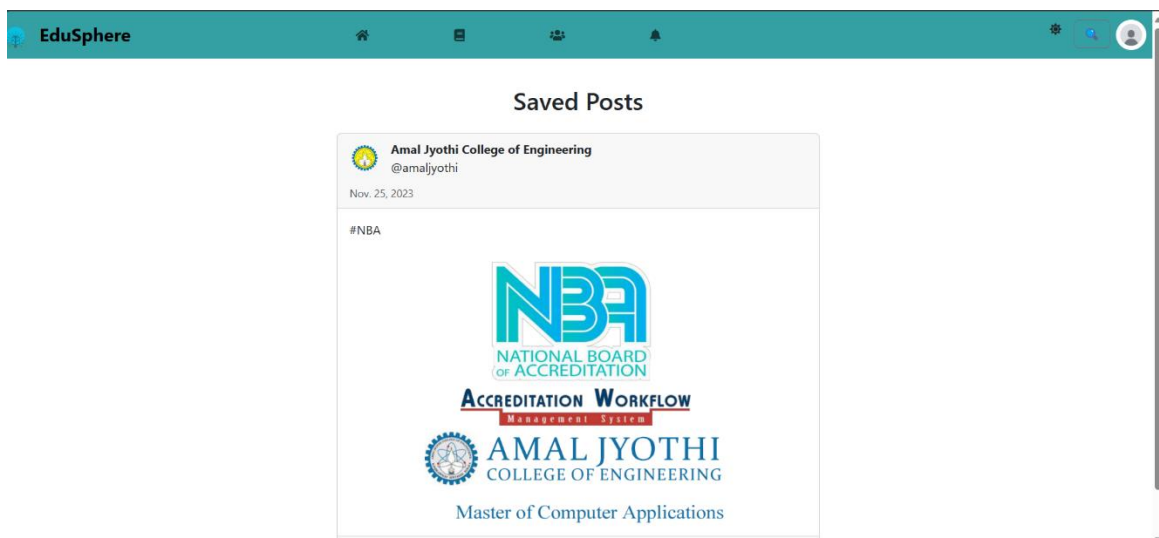


Figure 9.2.27 SAVED POSTS



