



Tribhuvan University
Faculty of Humanities and Social Sciences

LEARNING MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted To
Department of Computer Application
Ratna RajyaLaxmi Campus

In partial fulfillment of the requirements for the Bachelors in Computer Applications

Submitted By
Gopal Pokhrel (6-2-40-19-2020)
September 2024

Under the Supervision of
Mrs. Kriti Nemkul



Tribhuvan University
Faculty of Humanities and Social Sciences
Ratna RajyaLaxmi Campus

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by “Gopal Pokhrel”, entitled “LEARNING MANAGEMENT SYSTEM” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Mrs. Kriti Nemkul

SUPERVISOR

Department of Bachelors in Computer Applications

Ratna RajyaLaxmi Campus



Tribhuvan University
Faculty of Humanities and Social Sciences
Ratna RajyaLaxmi Campus

LETTER OF APPROVAL

This is to certify that this project prepared by Gopal Pokhrel entitled “LEARNING MANAGEMENT SYSTEM” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

<p>Signature of Supervisor</p> <hr/> <p>Mrs. Kriti Nemkul Department of BCA Ratna RajyaLaxmi Campus Pradarshani Marg, Kathmandu</p>	<p>Signature of HOD/Coordinator</p> <hr/> <p>Mr. Bhupendra Ram Luhar Department of BCA Ratna RajyaLaxmi Campus Pradarshani Marg, Kathmandu</p>
<p>Signature of Internal Examiner</p> <hr/>	<p>Signature of External Examiner</p> <hr/>

ABSTRACT

Learning Management System (LMS) is an online platform that makes teaching and learning easier. It helps instructors upload and organize course materials, and it allows students to access these resources anytime, anywhere.

An LMS is useful for schools, colleges, companies, and any organization that needs to provide training or education. It supports different user roles like administrators, instructors, and students, each with their own set of tools and permissions.

With an LMS, instructors can manage courses and communicate with students through discussion areas if they have any queries. Students can view their courses and see how much of the course they have completed. The system helps keep everything organized and accessible.

Keywords: *E-learning, online, education, course management*

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Mrs. Kriti Nemkul, for giving me the opportunity to work on this "Learning Management System" project. This project has been a valuable learning experience, teaching me about research, new tools and techniques which will also be extremely helpful for my future.

Throughout the project, I faced many challenges and had my share of ups and downs. During these times, my project supervisor and my friends were there to support me with their technical expertise. Their guidance played a crucial role in helping me complete this project within the limited time frame.

This project would not have been possible without the contributions and support of everyone involved. I am truly grateful to all who stood by my side and helped me along the way. Thank you for your kindness and support in making this project a success.

TABLE OF CONTENTS

SUPERVISOR’S RECOMMENDATION	i
LETTER OF APPROVAL.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT.....	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives.....	2
1.4 Scope and Limitation	3
1.4.1 Scope	3
1.4.2 Limitations.....	3
1.5 Development Methodology.....	4
1.6 Report Organization	4
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW	6
2.1 Background Study	6
2.2 Literature Review.....	6
CHAPTER 3: SYSTEM ANALYSIS AND DESIGN	9
3.1 System Analysis	9
3.1.1 Requirement Analysis.....	9
3.1.2 Feasibility Analysis	11
3.1.3 Object Modeling:	13
3.1.4 Dynamic Modeling: State and Sequence Diagram.....	15
3.1.5 Process Modeling: Activity Diagram	18
3.2 System Design.....	21
3.2.1 Refinement of Class Diagram.....	21
3.2.2 Component Diagram.....	22
3.2.3 Deployment Diagram	22
3.3 Algorithm Details.....	23
CHAPTER 4: IMPLEMENTATION AND TESTING	27
4.1 Implementation.....	27

4.1.1 Tools Used	27
4.1.2 Implementation details of modules.....	27
4.2 Testing.....	29
4.2.1 Test Cases for Unit Testing	29
CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS	37
5.1 Lesson Learnt	37
5.2 Conclusion.....	37
5.3 Future Recommendations.....	37
APPENDICES	39
Implementation Code	39
Screenshots of System.....	43
REFERENCES.....	47

LIST OF FIGURES

Figure 1.1: Iterative Waterfall Model of Learning Management System [1]	4
Figure 3.1: Use case Diagram of Learning Management System	10
Figure 3.2: Gantt Chart of Learning Management System.....	12
Figure 3.3: Class Diagram of Learning Management System	13
Figure 3.4: Object Diagram of Learning Management System	14
Figure 3.5: State Diagram of Learning Management System.....	15
Figure 3.6: Sequence Diagram of Learning Management System	17
Figure 3.7: Activity Diagram of Learning Management System for Instructor	18
Figure 3.8: Activity Diagram of Learning Management System for Student.....	19
Figure 3.9: Activity Diagram of Learning Management System for Admin.....	20
Figure 3.10: Refined Class Diagram.....	21
Figure 3.11: Component Diagram for Learning Management System.....	22
Figure 3.12: Deployment Diagram of Learning Management System.....	22

LIST OF TABLES

Table 4.1: Test Case for Register.....	29
Table 4.2: Test Case for Login	30
Table 4.3: Test Case for Course.....	30
Table 4.4: Test Case for Section	31
Table 4.5: Test Case for Cart	32
Table 4.6: Test Case for Payment	33
Table 4.7: Test Case for Review	34
Table 4.8: Test Case for Discussion	35
Table 4.9: Test Case for Recommendation.....	36

LIST OF ABBREVIATIONS

API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
IDF	Inverse Document Frequency
JS	JavaScript
LMS	Learning Management System
NLTK	Natural Language Toolkit
REST	Representational State Transfer
SQL	Structured Query Language
TC	Test Case
TF	Term Frequency
UI	User Interface

CHAPTER 1: INTRODUCTION

1.1 Introduction

The advancement of technology has significantly transformed various sectors, including education. In the past decade, traditional classroom-based education has been increasingly supplemented or even replaced by digital learning solutions. The need for flexible, accessible, and scalable educational platforms has led to the development of Learning Management Systems (LMS). An LMS is a comprehensive software application designed to facilitate the delivery of educational courses, training programs, or learning and development programs.

The shift towards e-learning platforms is driven by several factors. Firstly, there is a growing demand for lifelong learning and professional development, which traditional education systems cannot fully accommodate due to constraints in physical infrastructure and rigid schedules. Secondly, the global reach of the internet allows for the democratization of education, making it possible for learners from various geographical locations and backgrounds to access quality educational resources. Additionally, the COVID-19 pandemic has also shown the importance and need of having online educational systems, further demanding the need of e-learning solutions.

This project aims to develop a Learning Management System that addresses the limitations of existing platforms and enhances the overall learning experience. The platform is designed to cater to asynchronous learning needs. The LMS will allow students and instructors to register, with instructors having the capability to upload various course modules, and other educational content. Students are be able to purchase and enroll in these courses, track their progress, and earn certificates upon completion.

The primary objectives of the Learning Management System (LMS) are to provide tools for instructors to easily create and upload course modules and other educational content, enable students to earn certificates upon course completion, and offer section-specific discussions for students.

In conclusion, this project aims to develop a Learning Management System that meets the current demands of the educational landscape by providing a robust, scalable, and engaging platform for learners and educators alike. This initiative has the potential to significantly

enhance the quality and reach of education, contributing to the broader goal of making learning more accessible for everyone.

1.2 Problem Statement

Even though there are many e-learning platforms available, a critical issue not adequately addressed by existing platforms is the lack of section-dependent discussions. In many courses, students have questions specific to certain sections, but without dedicated discussion spaces, it's challenging for them to seek help from instructors or peers. A system that provides discussions for every course section would significantly enhance student engagement and support, allowing for interaction and more effective problem-solving. This approach would address a major gap in current e-learning solutions, making online education more interactive and responsive to student needs. These problems show the need for a better Learning Management System that solves these issues and plans for future educational needs. This system will improve access to education, keep students engaged and make course management easier. This will greatly improve the quality and reach of online education.

1.3 Objectives

The objectives of Learning Management System are listed below:

- To provide a platform for instructors to create and manage educational content.
- To recommend similar courses to students.
- To provide discussion area in each section, where students can communicate.

1.4 Scope and Limitation

1.4.1 Scope

- Instructor can create, organize, and manage course materials, making it easier to deliver educational content.
- The LMS allows students to access course materials anytime and anywhere, promoting easy learning.
- The LMS can recommend similar courses to students based on what they are viewing.
- Students can see their own progress and how much of the course they have completed.
- The LMS includes discussion areas where instructors and students can communicate and ask questions.
- The LMS helps keep course materials and communications organized and accessible, improving overall efficiency for both instructors and students.

1.4.2 Limitations

- LMS functionality depends on stable internet connectivity which can be a limitation in areas with unreliable internet infrastructure.
- For recommendation system to recommend with perfection, there should be very large number of courses.
- The LMS cannot verify the quality of uploaded courses.

1.5 Development Methodology

For the development of the Learning Management System (LMS), iterative waterfall model is used. This approach ensured that each phase is completed before moving to the next, maintaining a structured and systematic development process and also allowing to give feedbacks.

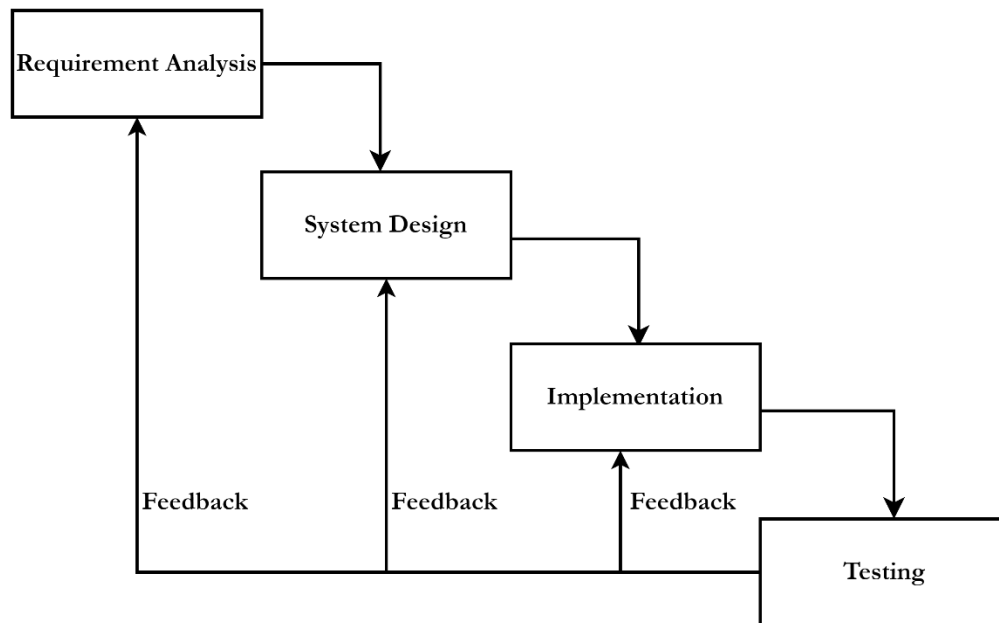


Figure 1.1: Iterative Waterfall Model of Learning Management System [1]

1.6 Report Organization

This project is divided into five major chapters.

Chapter 1 of the project gives the background information on the project and it also tells about the objectives, scope and limitation of the project.

Chapter 2 of the project is Background Study and Literature Review. This chapter provides the background information about topic and provides the known information and theories.

Chapter 3 of the project is System Analysis and Design. In this chapter, system will be analyzed by performing the requirement and feasibility analysis. This chapter also provides the system designs of the project.

Chapter 4 of the project is Implementation and Testing. This chapter includes the implementation details of the design and the tools used while building the project. It also provides information about unit and system testing.

Chapter 5 of the project is Conclusion and Future Recommendation. It is the last chapter which provides conclusion, outcomes and further recommendations related to the project.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

The concept of Learning Management Systems (LMS) has emerged as an important subject in modern education and training. Initially designed to facilitate distance learning, LMS platforms have evolved into versatile digital environments that streamline the administration, delivery, and management of educational content.

LMS platforms serve as centralized hubs where educators can upload course materials, such as lectures, readings, and assignments, making them accessible to students anytime and anywhere with internet access. This accessibility promotes flexible learning opportunities, enabling students to learn at their own pace and convenience.

The adoption of LMS has been driven by advancements in technology and the growing demand for personalized learning experiences. These platforms not only support traditional academic settings but also cater to diverse educational needs, including training programs and continuing education initiatives.

Despite their benefits, the implementation of LMS faces challenges such as technical requirements, integration complexities with existing systems, and the need for adequate training and support for educators and users. Moreover, ensuring data security and privacy within LMS platforms is crucial, given the sensitive nature of educational information.

Looking ahead, the continued evolution of LMS will likely focus on enhancing user experience, expanding accessibility, and integrating innovative features to support collaborative learning environments. Emphasizing these aspects will be essential in leveraging LMS technology to its fullest potential in education and training across various sectors.

2.2 Literature Review

The literature review for this project involved an in-depth study of existing research papers, articles, and case studies related to e-learning platforms and Learning Management Systems (LMS). Additionally, research was done on popular e-learning platforms, like Coursera, Udemy, LinkedIn Learning, Khan Academy, edX, edureka, and Skillshare.

These platforms represent a diverse range of approaches to online education and offer valuable insights into the strengths and weaknesses of current systems.

Many instructors who had no interest in or little experience with virtual teaching found themselves scrambling to move their businesses online due to advancement in technology. They also had to consider important aspects of their business including which online platform they would use to offer classes (e.g., Zoom, Instagram Live, YouTube, or another platform) and how they would handle course building, scheduling and payment processing. The common donation-based model is not effective from a business perspective. [2]

Coursera stands out for its collaboration with renowned universities and organizations to offer high-quality courses, specializations, and degree programs. The platform emphasizes structured learning with scheduled classes, peer-graded assignments, and certification upon completion. [3]

Udemy has a vast library of courses covering a wide range of topics, often created by individual instructors. The platform offers flexibility in course selection and delivery, allowing instructors to upload video content, and assignments. [4]

LinkedIn Learning focuses on professional development and skill-building, integrating seamlessly with users' LinkedIn profiles. The platform offers personalized course recommendations based on users' career interests and goals. While LinkedIn Learning excels in professional integration and personalized recommendations, its course variety may be narrower compared to other platforms. [5]

Khan Academy provides a free, nonprofit educational platform offering instructional videos, practice exercises, and a personalized learning dashboard. The platform covers a wide range of subjects, making it accessible to learners of all ages. [6]

edX is a massive open online course (MOOC) platform founded by MIT and Harvard University, offering a wide range of courses from universities and institutions worldwide. The platform emphasizes high-quality content, interactive learning experiences, and verified certificates for course completion. [7]

Edureka is known for its focus on professional development, particularly in technology and business fields. The platform offers instructor-led live courses, which provide an interactive learning experience. [8]

Skillshare offers a diverse range of courses primarily focused on creative skills, business, and technology. The platform encourages peer interaction and project-based learning, which can enhance the learning experience. [9]

By studying findings from the literature review and analysis of existing systems, I gained valuable insights that helped in the design and development of Learning Management System. These insights include understanding the importance of flexibility, user engagement, content quality, and the need for both structured and self-paced learning options.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

During the system analysis phase, the project's components were examined to identify inefficiencies and ensure alignment with the objectives. Workflows, database structures, and module interactions were analyzed. Key challenges were identified and addressed, resulting in a more efficient system.

3.1.1 Requirement Analysis

The requirement analysis of Learning Management System is done through finding the functional requirements and non-functional requirements for the system.

Functional Requirements

Functional requirements were identified through the creation of use case diagrams, which helped in determining the essential functionalities of the Learning Management System. These functionalities were mapped out to ensure that the system supports users in achieving their goals efficiently.

Use Case Diagram

Student is allowed to register, log in, and log out of their accounts. Students can buy courses, start discussions, give reviews, and view available courses.

Instructor role includes registering, logging in, logging out, and adding and managing their courses. Instructors can also start discussions, view courses, and buy courses from other instructors.

Here both students and instructors can view recommendations based on similar courses. If they are enrolled in any course, they can track their progress as well.

Admin role includes logging in, logging out, and managing all aspects of the platform. Admins can manage courses, view courses, and manage students and instructors.

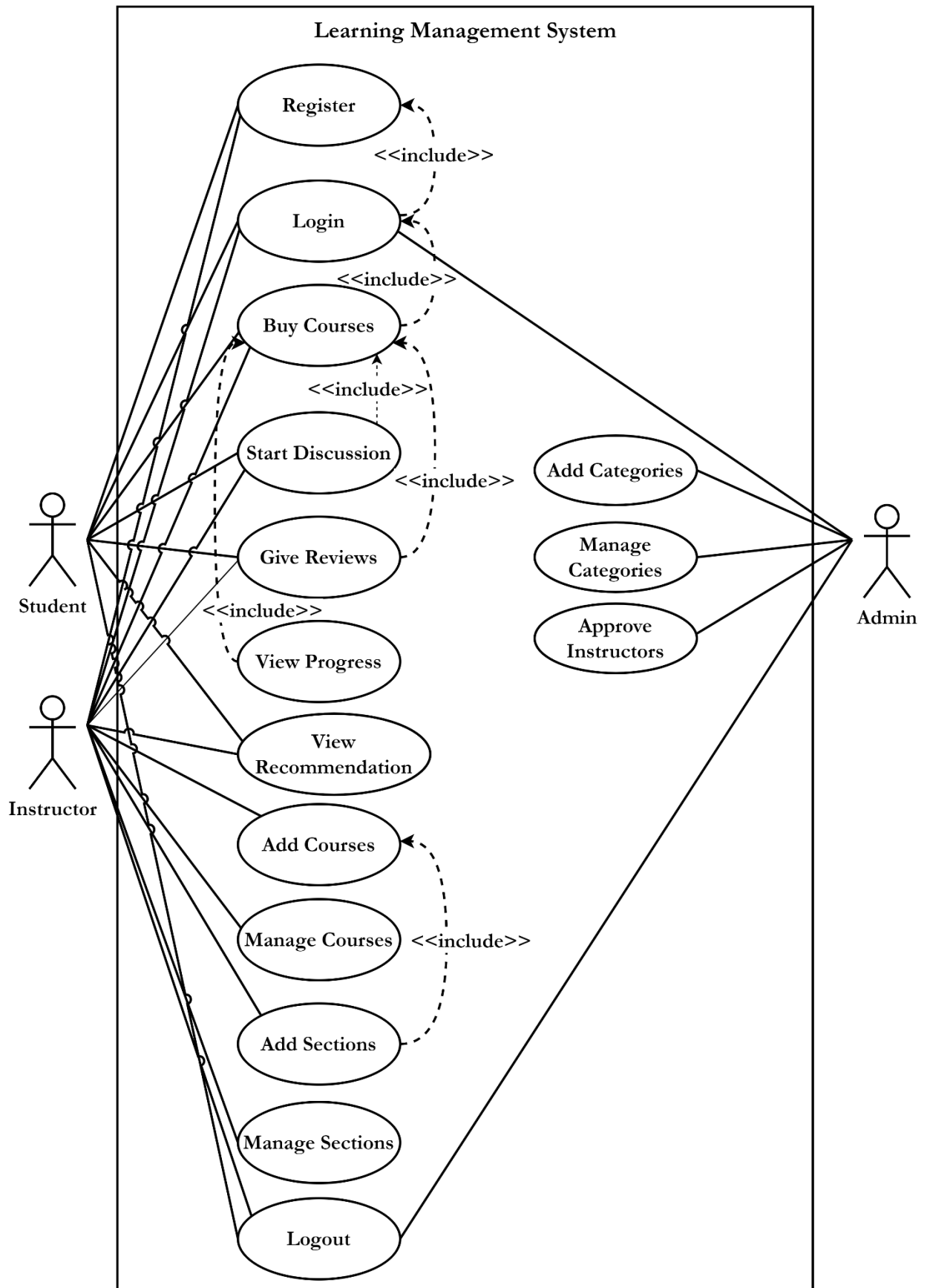


Figure 3.1: Use case Diagram of Learning Management System

Non-Functional Requirements

Non-functional requirements were identified to ensure that the system operates effectively, providing a good user experience, ease of use, and cost efficiency. These attributes were considered critical in enhancing the overall performance and usability of the Learning Management System. Key parameters to ensure non-functional requirements are:

1. Performance

The system is able to handle a large amount of data processing efficiently.

2. Security

The system is highly secure to ensure that sensitive information is protected.

3. Usability

The system is user friendly and interface should be easy to use and understand.

4. Accessibility

The system is accessible to all users, regardless of their location or device and it should support variety of operating systems.

5. Maintainability

The system is easy to maintain and update.

3.1.2 Feasibility Analysis

Technical Feasibility

After assessing the technical feasibility, the chosen technology stack of Django, React, Next.js and PostgreSQL proves to be well-suited for developing the proposed Learning Management System (LMS). These technologies offer robust capabilities for web application development, ensuring smooth implementation.

Operational Feasibility

Operational feasibility was evaluated to determine how effectively the proposed LMS could integrate into the operational environment. This included assessing the user interface design for user-friendliness and identifying any potential training requirements. With a focus on creating an easily usable LMS design, achieving operational feasibility is possible.

Economic Feasibility

Economic feasibility was examined to assess the financial viability of the project. Development costs were estimated, focusing on freely available tools and resources which showed that the project is economically feasible.

Schedule Feasibility

Schedule feasibility was assessed to determine whether the project timeline and deadlines were reasonable. Consideration was given to the constraints of limited time, ensuring that the project could be completed within the specified timeframe, even under mandatory deadlines.

Month	June				July				August				September			
Week	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Planning																
Analysis and Design																
Implementation																
Testing																
Documentation																

Figure 3.2: Gantt Chart of Learning Management System

3.1.3 Object Modeling:

Class Diagram

This class diagram represents the structure and relationships within the database models of Learning Management Systems, including the differentiation of user roles and relationship among various classes.

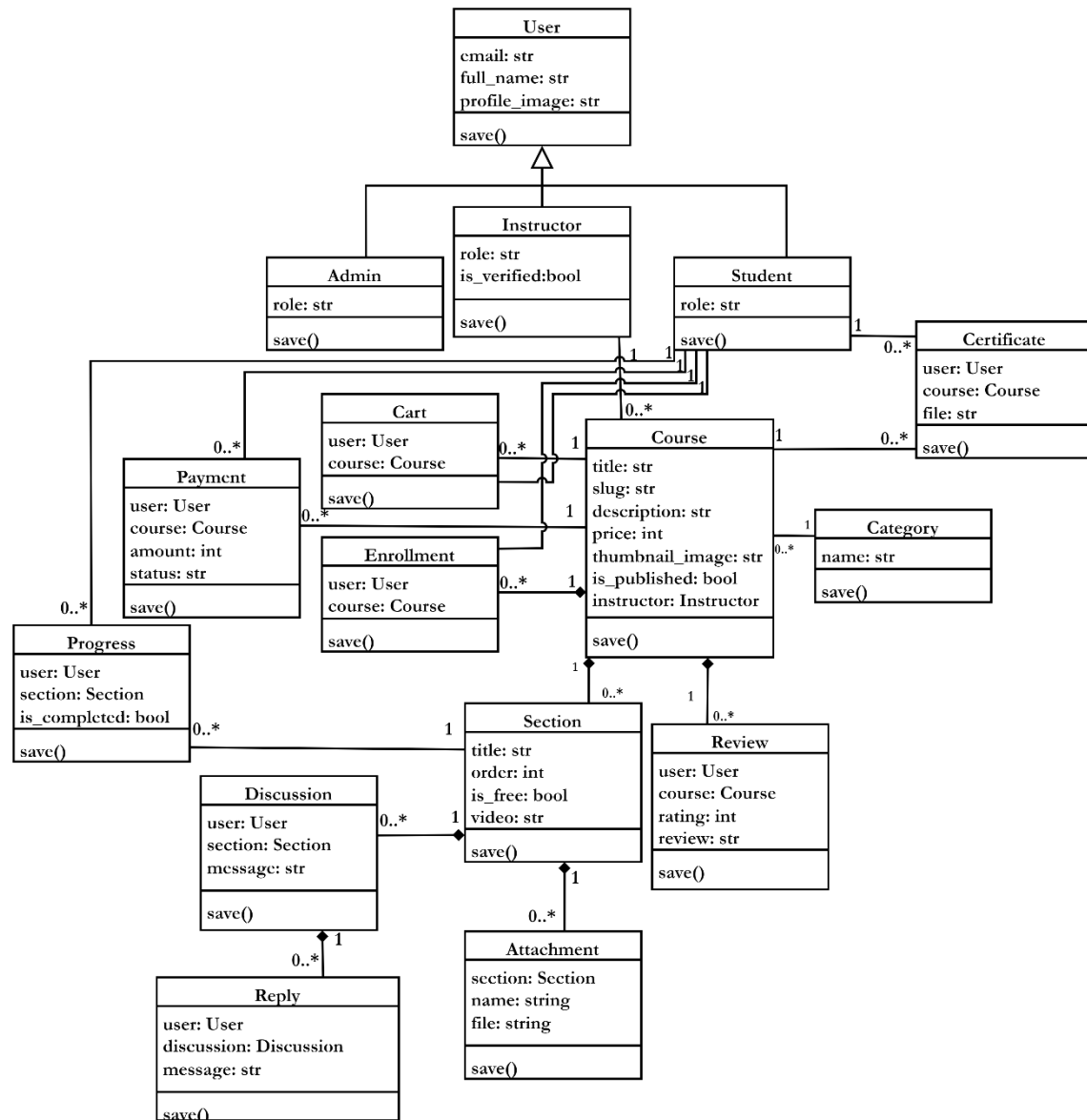


Figure 3.3: Class Diagram of Learning Management System

Object Diagram

The following object diagram illustrates a specific snapshot of Learning Management System (LMS), showcasing the relationships and states of various objects within the system. The diagram provides a clear view of how various entities within the system are interrelated, showing the interactions between users and the courses they engage with along with other entities.

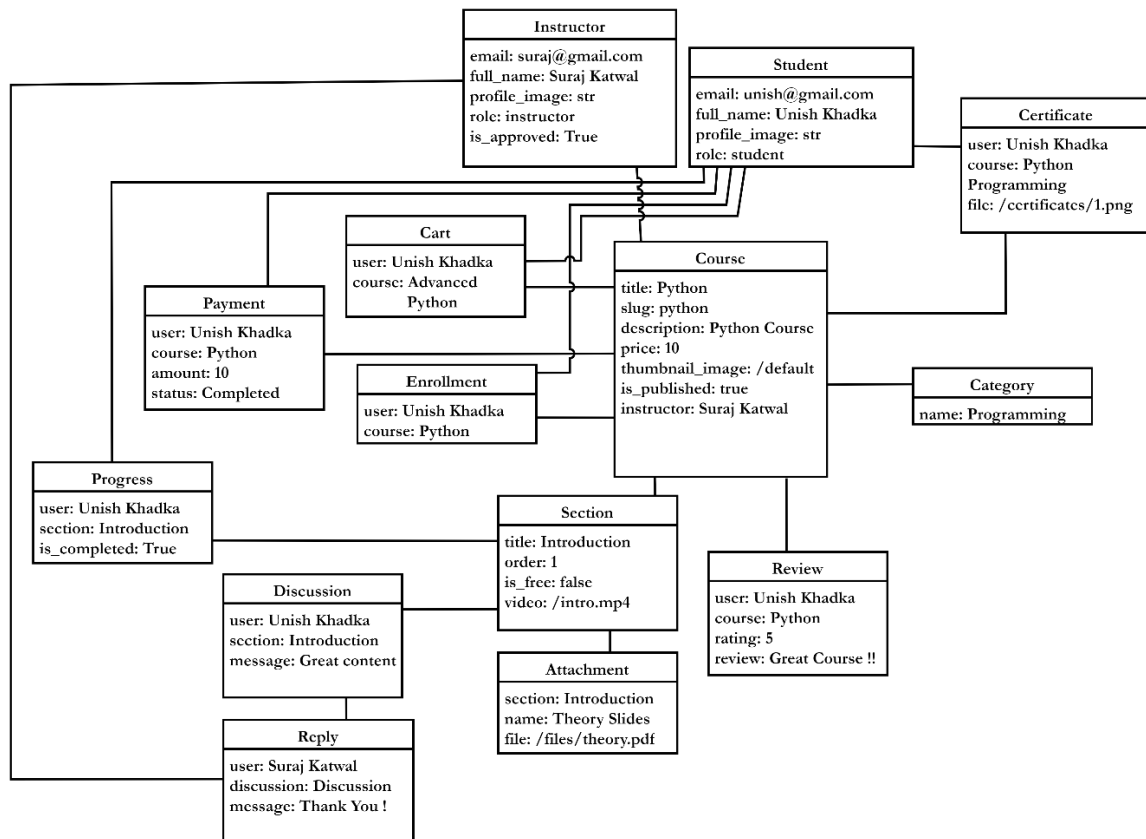


Figure 3.4: Object Diagram of Learning Management System

3.1.4 Dynamic Modeling: State and Sequence Diagram

State Diagram

The following state diagram demonstrates different lifecycle states that the different entities go through.

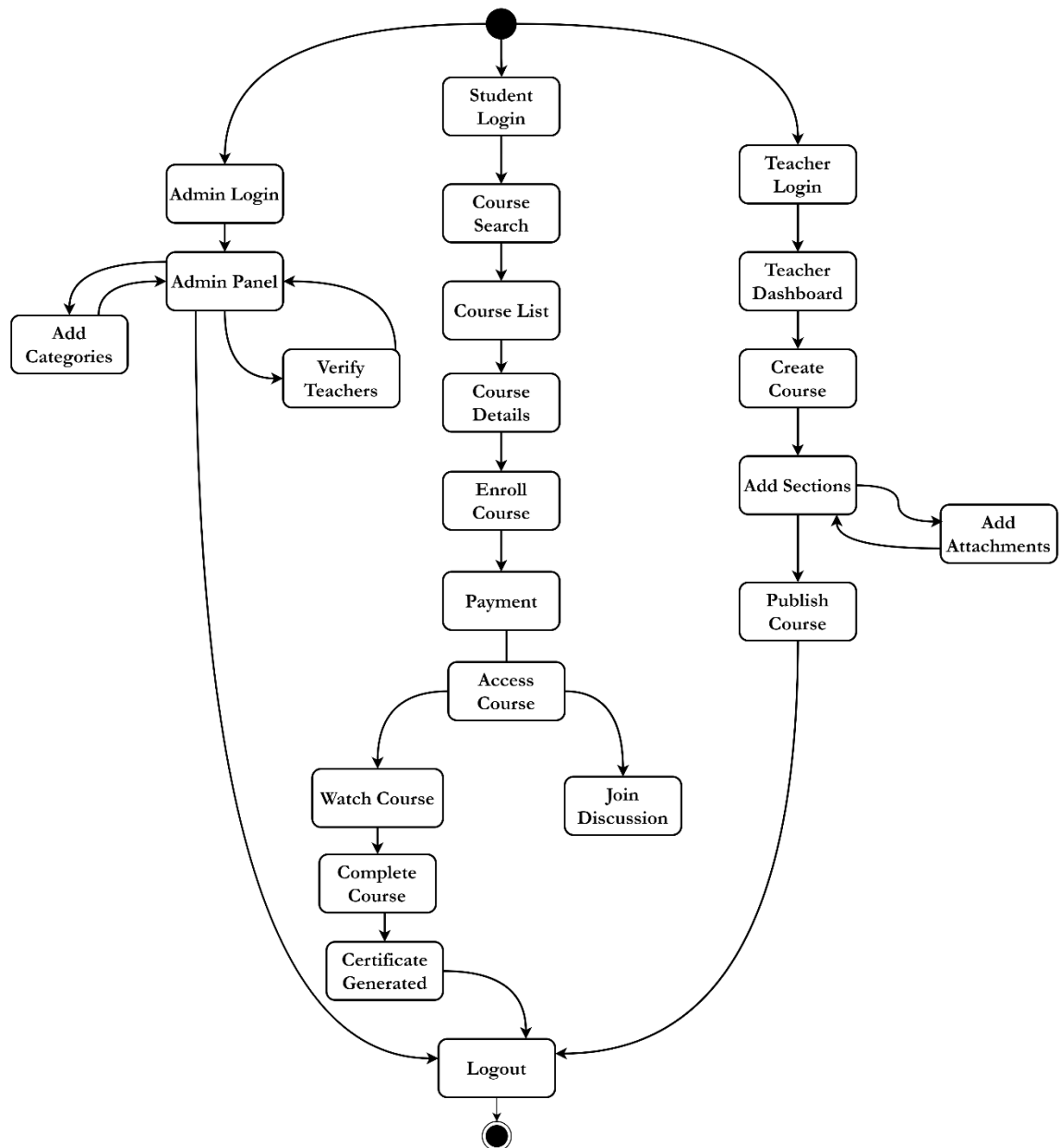
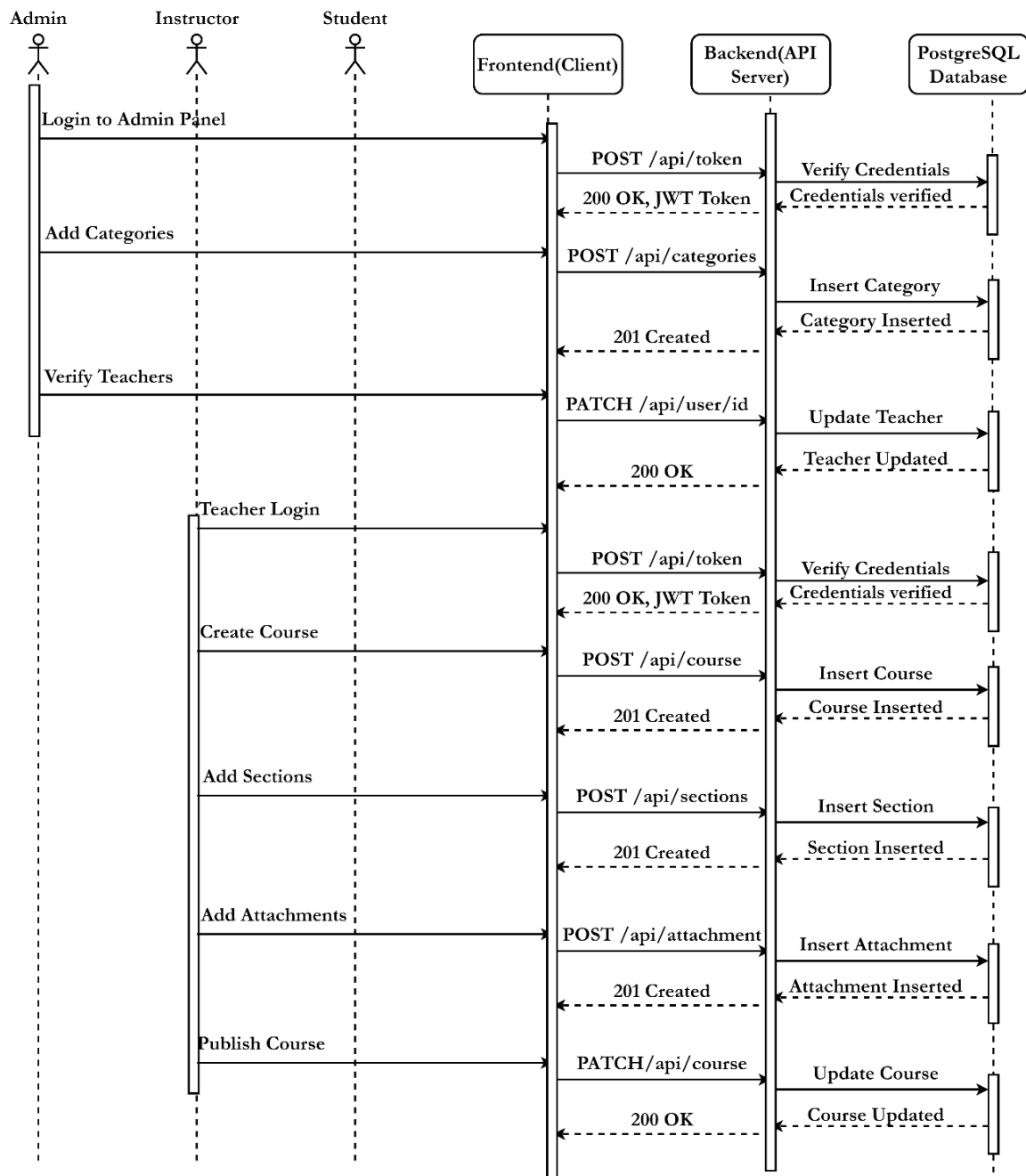


Figure 3.5: State Diagram of Learning Management System

Sequence Diagram

This sequence diagram illustrates the interactions among objects in a sequential order. In the context of our Learning Management System (LMS), the sequence diagram captures the dynamic behavior of the system, depicting how various components interact to perform specific functionalities.



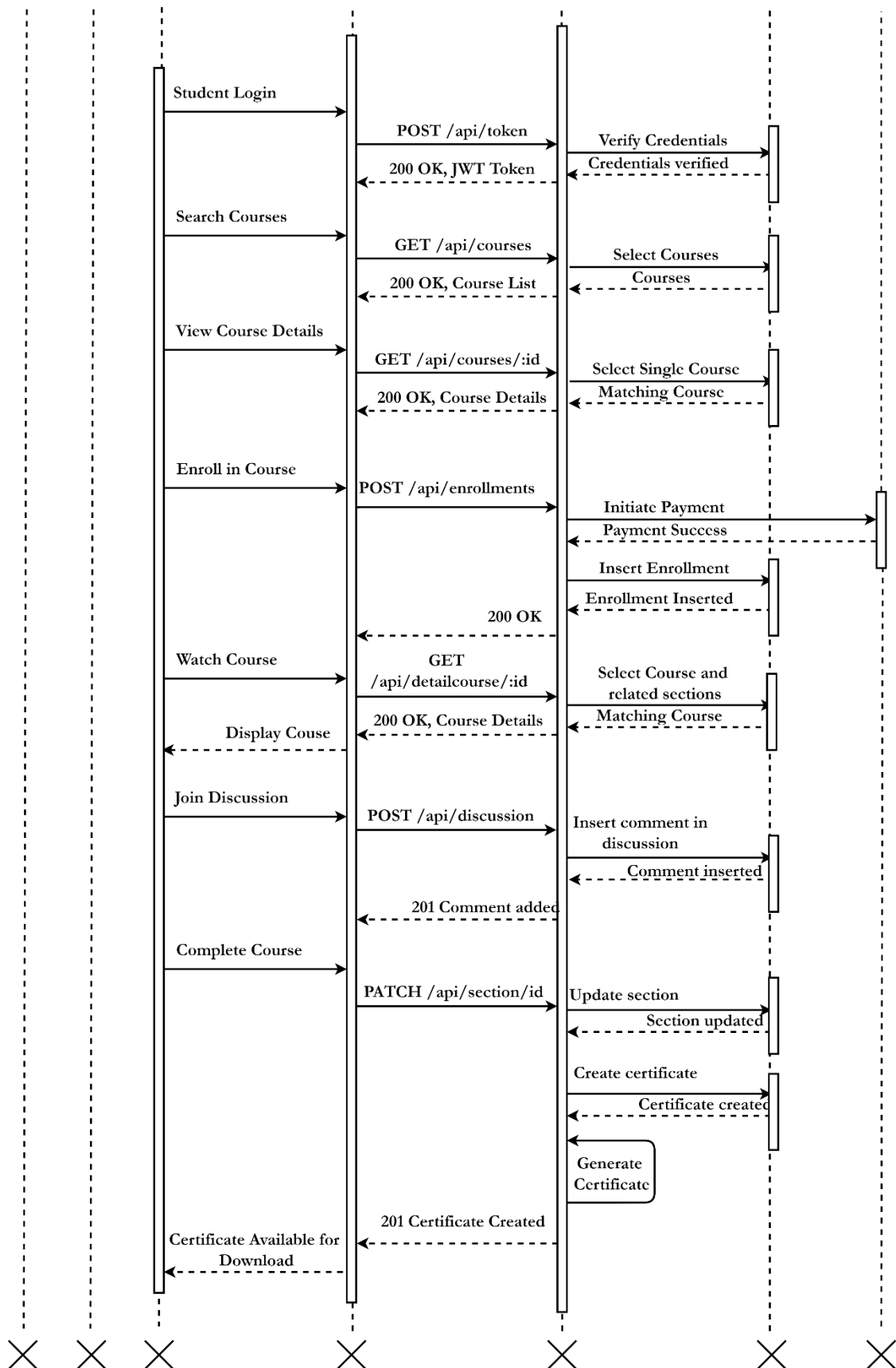


Figure 3.6: Sequence Diagram of Learning Management System

3.1.5 Process Modeling: Activity Diagram

The following two activity diagrams illustrate the processes within a Learning Management System.

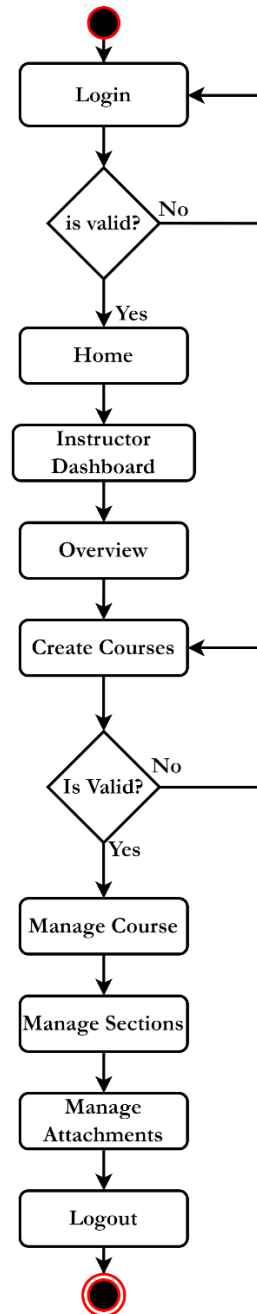


Figure 3.7: Activity Diagram of Learning Management System for Instructor

The above diagram shows the instructor workflow, starting from logging in, accessing the dashboard, creating and updating courses, managing sections and attachments, publishing the course and finally logging out.

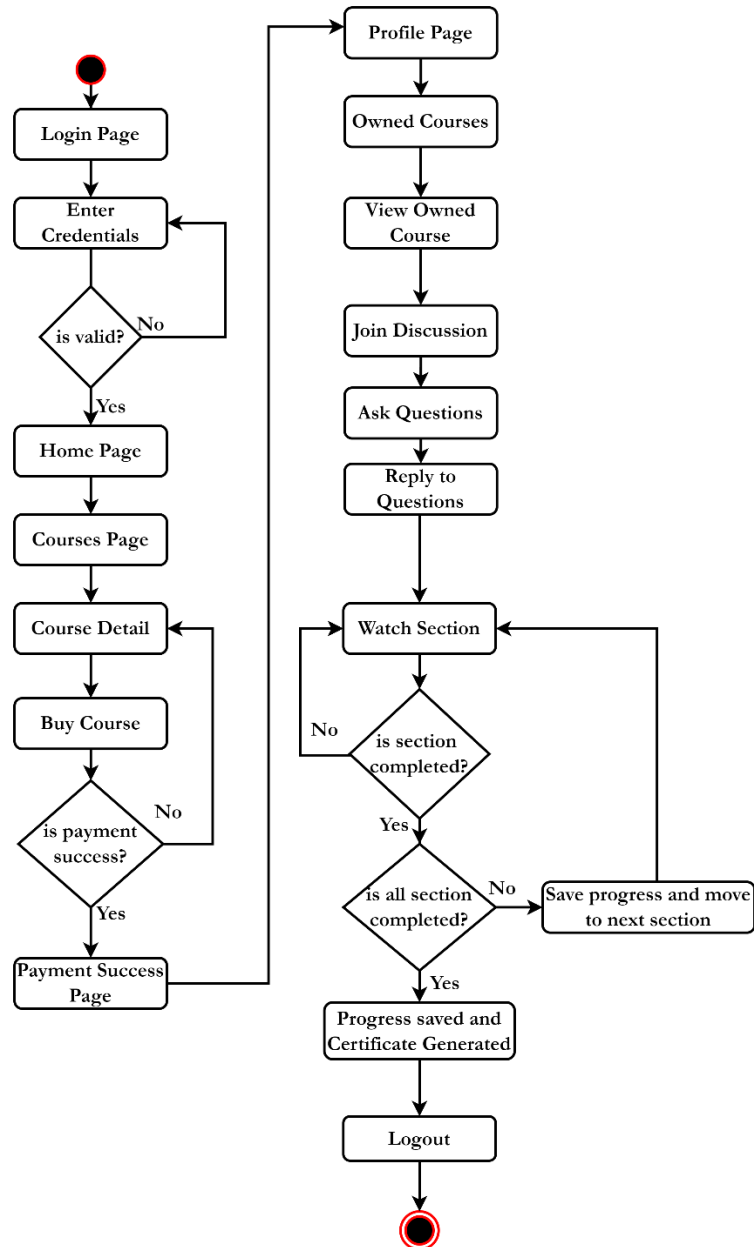


Figure 3.8: Activity Diagram of Learning Management System for Student

This above diagram shows a student's activity flow in a Learning Management System (LMS). It covers login, course purchase, watching sections, engaging in discussions, and completing courses. Upon finishing all sections, progress is saved, and a certificate is generated, followed by logout.

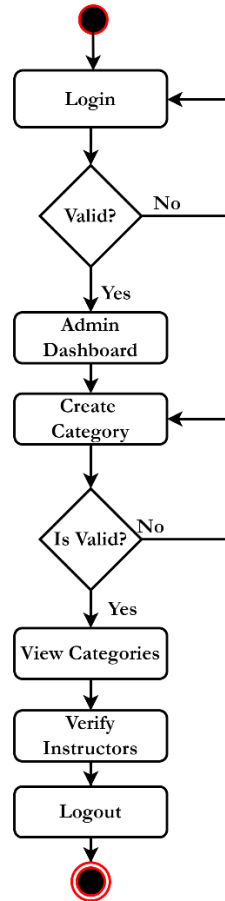


Figure 3.9: Activity Diagram of Learning Management System for Admin

This diagram represents the admin activity flow in an LMS. The admin logs in, and upon successful validation, accesses the dashboard. From there, the admin can create and view categories, verify instructors, and finally log out.

3.2 System Design

During the system design phase, the analysis is taken to another step where existing classes and object diagrams are refined and component and deployment diagram are constructed.

3.2.1 Refinement of Class Diagram

In the following refined diagram, the existing class diagram has been refined by adding more detail. The diagram consists of method signatures, return types and parameter types. While the class diagram showed the overview of the system, the refined class diagram adds more detail for actual implementation.

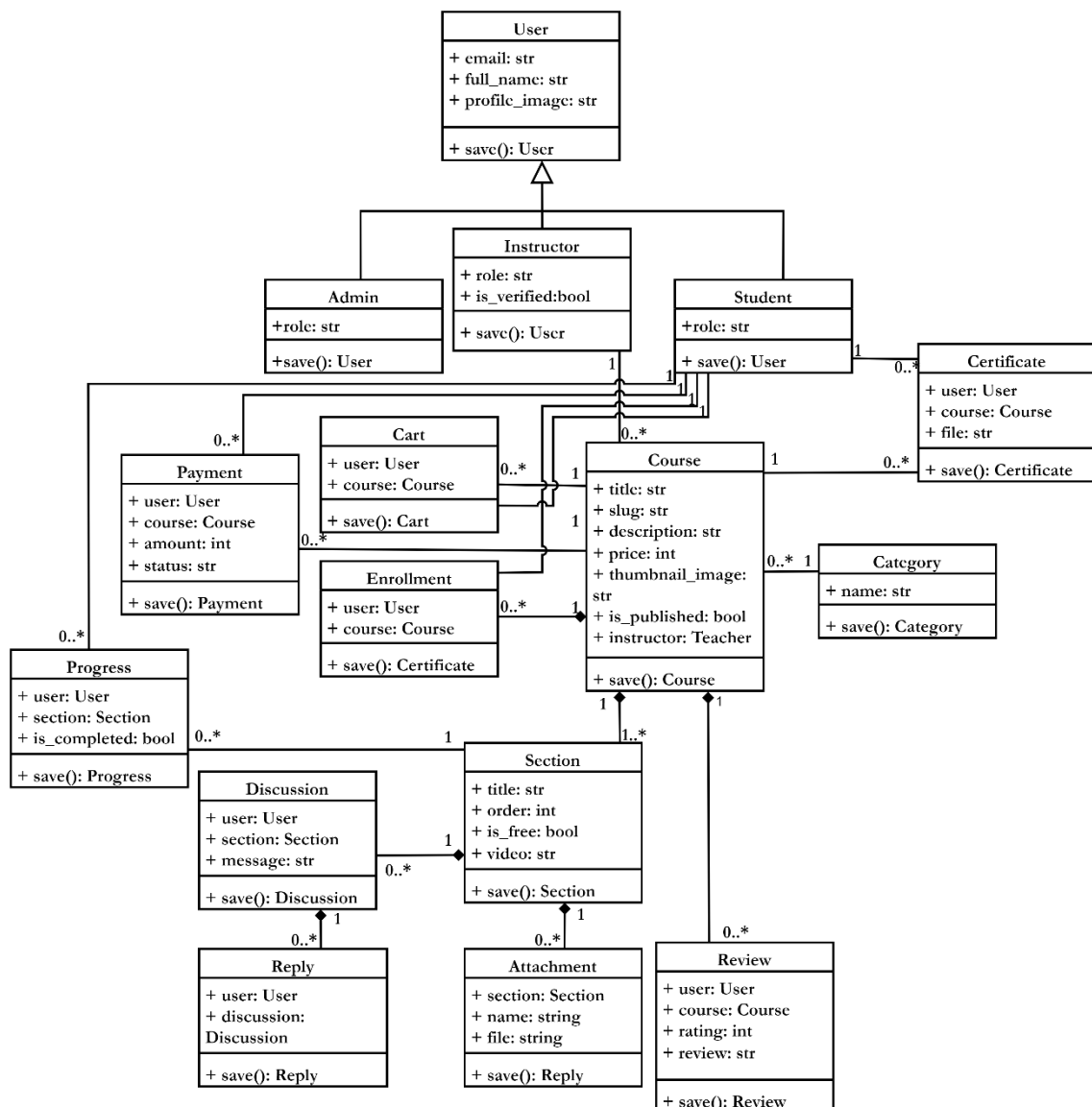


Figure 3.10: Refined Class Diagram

3.2.2 Component Diagram

The component diagram below illustrates the physical components of the Learning Management System and their interactions. With its help we are able to visualize the overall structure and organization of system.

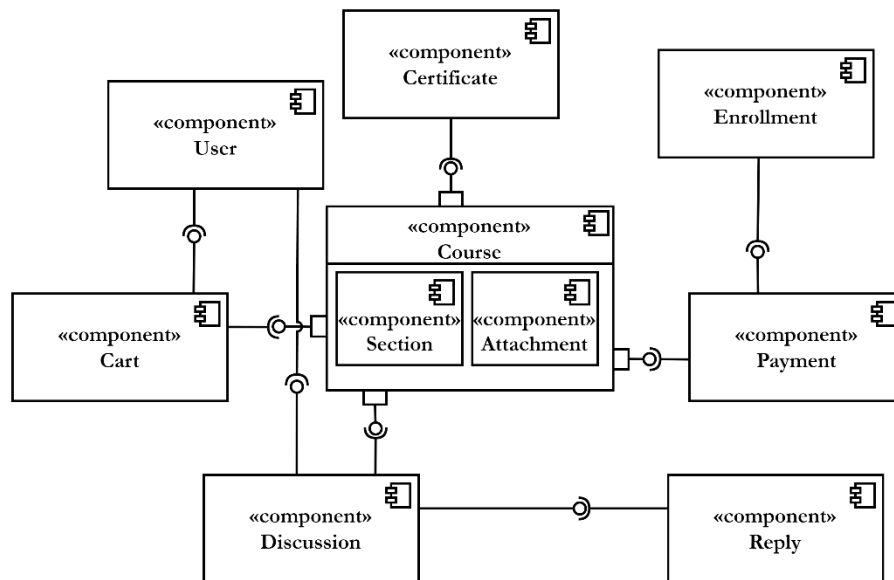


Figure 3.11: Component Diagram for Learning Management System

3.2.3 Deployment Diagram

The following deployment diagram illustrates how software components are distributed across hardware nodes and how they interact with each other in a runtime environment.

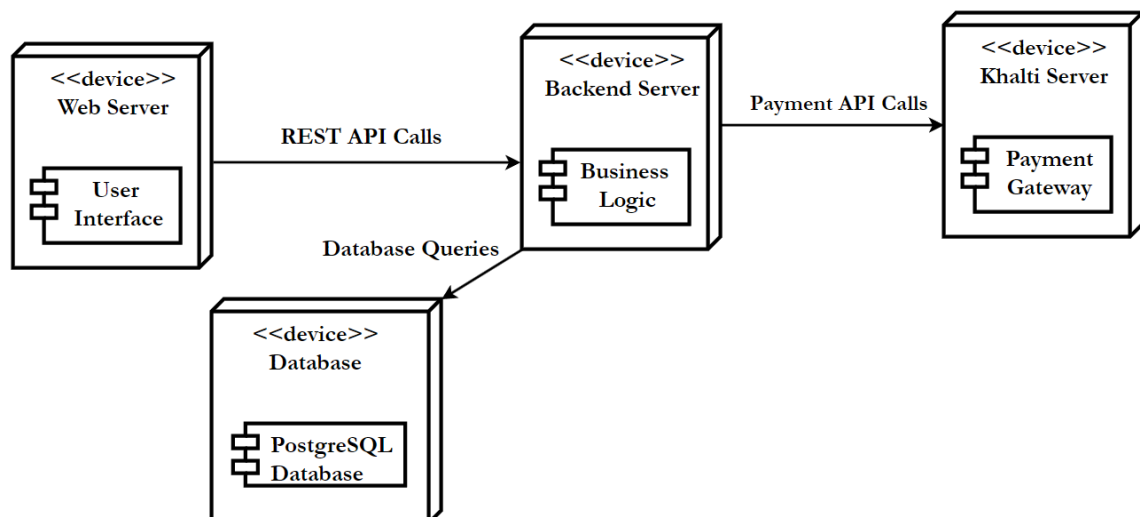


Figure 3.12: Deployment Diagram of Learning Management System

3.3 Algorithm Details

In Learning Management System (LMS), content-based recommendation system has been implemented to assist students in discovering relevant courses based on their interests. This recommendation system utilizes the TF-IDF (Term Frequency-Inverse Document Frequency) technique coupled with cosine similarity to compute the relevance between courses.

TF-IDF Text Vectorization

TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents. It consists of two main components:

- **Term Frequency (TF)**

Measures how frequently a term occurs in a document. It is calculated as the ratio of the number of occurrences of a term to the total number of terms in the document.

$$TF(t, d) = \frac{\text{number of occurrences of term 't' in row 'd'}}{\text{total number of terms in row 'd'}}$$

- **Inverse Document Frequency (IDF)**

Measures the importance of a term by giving less weight to terms that occur frequently across all documents. It is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term.

$$IDF(t) = \log_e \times \frac{\text{total number of rows}}{\text{number of rows with term 't'}} + 1$$

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t)$$

Cosine Similarity

In LMS, each course is transformed into a TF-IDF vector where terms are weighted based on their relevance to the course content. To recommend courses, we calculate the cosine similarity between these TF-IDF vectors across all courses. This metric quantifies how closely aligned two courses are in content, with higher cosine similarity values indicating greater similarity.

$$\text{Cosine Similarity } (\cos\theta) = \frac{A \cdot B}{|A||B|}$$

Steps

1. Fetch Data:

Retrieve list of published courses, instructors, and categories from database

2. Create Data Structures:

Convert retrieved data into data structures for courses, instructors, and categories

3. Preprocess Data:

For each course:

Remove spaces from instructor names and course categories

Concatenate course title, instructor names, and categories to create a tag

4. Remove Stop Words:

For each course tag:

Using NLTK remove common stop words like I, me, my, he, where, etc.

5. Create Document List:

documents = empty list

For each course in courses:

Split course tags into words

Add resulting list of words to documents

6. Compute Term Frequency (TF):

tf_matrix = empty list

For each doc in documents:

doc_tf = empty dictionary

total_words = count words in doc

For each unique word in doc:

$\text{doc_tf}[\text{word}] = (\text{count of word in doc}) / \text{total_words}$

Add doc_tf to tf_matrix

7. Compute Inverse Document Frequency (IDF):

unique_words = set of all unique words across all documents

num_documents = count of documents

idf = empty dictionary

For each word in unique_words:

doc_count = count of documents containing the word

idf[word] = $\log(\text{num_documents} / \text{doc_count})$

8. Compute TF-IDF:

tf_idf_matrix = empty list

For each tf_doc in tf_matrix:

doc_tfidf = empty dictionary

For each word and its tf in tf_doc:

doc_tfidf[word] = $\text{tf} * \text{idf}[\text{word}]$

Add doc_tfidf to tf_idf_matrix

9. Build TF-IDF Matrix:

Create a matrix where:

Each row represents a document

Each column represents a unique word

Cell values are TF-IDF scores

10. Compute Cosine Similarity:

For each pair of documents (courses):

Calculate cosine similarity using their TF-IDF vectors

11. Find Similar Courses:

Given a current_course:

Find index of current_course in the course list

Retrieve similarity scores with other courses

12. Sort Similar Courses:

Sort courses by similarity score in descending order

13. Select Top N Courses:

Return top N most similar courses, excluding the current course

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used

Django

A RESTful API is developed using Django to facilitate communication between the frontend and backend of the LMS. This API defines operations for performing CRUD actions, recommendations, and other functionalities. Django ensures secure user authentication, data validation, and data exchanges, supporting the platform's flexibility and scalability.

Next.js with Tailwind CSS

Next.js and Tailwind CSS has been used to design and develop the user interface of the LMS. Next.js provides efficient server-side rendering and client-side routing, enhancing performance and navigation across the platform. Tailwind CSS facilitates rapid UI development with its utility-first approach, enabling customizable design without the need for custom CSS.

PostgreSQL

PostgreSQL serves as the database for our LMS, storing and managing critical data such as user profiles, course information, and system configurations. Known for its reliability and support for complex queries, PostgreSQL ensures efficient data storage and retrieval, maintaining data integrity and scalability for our platform.

Postman

Postman has been used for testing API endpoints in the LMS. It allows for sending requests, validating responses, and monitoring API performance.

4.1.2 Implementation details of modules

The major functional modules of Learning Management System and their implementation details are explained below.

- **Login Module**

Allows users to securely log into their accounts, providing access to personalized content and features according to their roles.

- **Register Module**
Enables new users to create accounts, collecting necessary information for user profiles according to their roles and allowing access to system resources.
- **Course Module**
Facilitates the creation, and management of courses where instructors can also publish or unpublish courses. Students can also track their progress on a course.
- **Cart Module**
Provides a virtual cart for users to add and manage selected courses before proceeding to payment.
- **Payment Module**
Handles secure payment transactions for course enrollment, supporting Khalti payment method and ensuring financial data protection. After user has successfully paid for course, user will automatically be enrolled in that course and the respective course will also be removed from cart, if it exists there.
- **Review Module**
Allows users to submit reviews and ratings for courses, providing feedback and helping others make informed decisions.
- **Discussion Module**
Provides a platform for course-specific discussions among students and instructors, helping in problem solving and knowledge sharing.

4.2 Testing

In this step, the system is evaluated to test if it performs as expected. Testing in LMS is done by testing every unit and modules.

4.2.1 Test Cases for Unit Testing

The test cases for several modules are as follows

Table 4.1: Test Case for Register

Project Name: Learning Management System					
Test Case 1					
Test Case ID: TC_1			Test Designed by: Gopal Pokhrel		
Module Name: Register			Test Executed by: Gopal Pokhrel		
Description: Testing the Register module of LMS					
SN	Test Case Detail	Input	Expected Result	Actual Result	Status
1	Test password do not match	Password1="12345678" Password2="12345667"	Register Failed	Registered Failed	Pass
2	Test email already taken	Email="existing@gmail.com"	Register Failed	Register Failed	Pass
3	Test invalid email	Email="123email123"	Register Failed	Register Failed	Pass
4	Test invalid name	full_name="User123"	Register Failed	Register Failed	Pass
5	Test valid registration	email = "newuser@example.com" password = "testpass123" full_name = "New User"	Register Success	Register Success	Pass

```
PS C:\Users\ASUS\Desktop\LMS\backend> py .\manage.py test users.tests.RegisterTestCase
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 3.266s

OK
```

Table 4.2: Test Case for Login

Project Name: Learning Management System					
Test Case 2					
Test Case ID: TC_2			Test Designed by: Gopal Pokhrel		
Module Name: Login			Test Executed by: Gopal Pokhrel		
Description: Testing the Login module of LMS					
SN	Test Case Description	Input	Expected Result	Actual Result	Status
1	Test login with invalid credentials	email = invaliduser@example.com password = wrong	Login Failed	Login Failed	Pass
2	Test login with empty data	email = "" password = ""	Login Failed	Login Failed	Pass
3	Test login with correct credentials	email = validuser@example.com password = testpass123	Login Failed	Login Success	Pass

```
PS C:\Users\ASUS\Desktop\LMS\backend> py .\manage.py test users.tests.LoginTestCase
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 2.117s

OK
```

Table 4.3: Test Case for Course

Project Name: Learning Management System					
Test Case 3					
Test Case ID: TC_3			Test Designed by: Gopal Pokhrel		
Module Name: Course			Test Executed by: Gopal Pokhrel		
Description: Testing the Course module of LMS					
Additional Information: The course module can only be used by instructors as they have authority to create and manage courses.					
SN	Test Case Description	Input	Expected Result	Actual Result	Status

1	Test Create Course	Title=Django Instructor= Gopal Category= Coding	Created	Created	Pass
2	Test Reading course	Getting course with title Django	Read	Read	Pass
3	Test Update Course	Title= Django Updated	Updated	Updated	Pass
4	Test Create Course with same name again	Title= Django Updated	Failed	Failed	Pass
5	Test Delete Course	Delete Course with title Django	Deleted	Deleted	Pass

```
PS C:\Users\ASUS\Desktop\LMS\backend> py .\manage.py test api.tests.CourseModelTest
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 2.782s

OK
Destroying test database for alias 'default'...
PS C:\Users\ASUS\Desktop\LMS\backend> █
```

Table 4.4: Test Case for Section

Project Name: Learning Management System					
Test Case 4					
Test Case ID: TC_4			Test Designed by: Gopal Pokhrel		
Module Name: Section			Test Executed by: Gopal Pokhrel		
Description: Testing the Section module of LMS					
Additional Information: The course module can only be used by instructors as they have authority to create and manage courses.					
SN	Test Case Description	Input	Expected Result	Actual Result	Status
1	Test Create Section	Title=Introduction Course=Django Order=1	Created	Created	Pass
2	Test Read Section	Read section where id=section.id	Read	Read	Pass

3	Test Create duplicate section	title=Introduction, course=Django, order=2	Failed	Failed	Pass
4	Test Delete Section	Delete section where id=section.id	Deleted	Deleted	Pass
5	Test Update Section	Title=Introduction to Django	Updated	Updated	Pass

```
PS C:\Users\ASUS\Desktop\LMS\backend> py .\manage.py test api.tests.SectionModelTest
Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 3.062s

OK
```

Table 4.5: Test Case for Cart

Project Name: Learning Management System					
Test Case 5					
Test Case ID: TC_5			Test Designed by: Gopal Pokhrel		
Module Name: Cart			Test Executed by: Gopal Pokhrel		
Description: Testing the Cart module of LMS					
SN	Test Case Description	Input	Expected Result	Actual Result	Status
1	Test create cart	User=student@gmail.com Course=Django Course	Added	Added	Pass
2	Test read cart items	Get cart for logged in user	Read	Read	Pass
3	Test clear cart	Delete cart that belongs to logged in user	Cleared	Cleared	Pass
4	Test add to cart	User=logged in user course=Java course	Added	Added	Pass
5	Test remove from cart	Delete cart item with given id	Removed	Removed	Pass

```

Found 5 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 5 tests in 3.014s

OK

```

Table 4.6: Test Case for Payment

Project Name: Learning Management System					
Test Case 6					
Test Case ID: TC_6			Test Designed by: Gopal Pokhrel		
Module Name: Payment			Test Executed by: Gopal Pokhrel		
Description: Testing the Payment module of LMS					
SN	Test Case Description	Input	Expected Result	Actual Result	Status
1	Test payment with missing fields	"amount": 1000, "purchase_order_id": "PO123", "purchase_order_name": "Test Order", # Missing return_url and website_url	Response Code 400	Response Code 400	Pass
2	Test Payment with required fields	"return_url": "https://example.com/return/", "website_url": "https://example.com/", "amount": 1000, "purchase_order_id": "PO123", "purchase_order_name": "Test Order",	Payment url in response	Payment url in response	Pass
3	Test Payment with invalid amount	Amount: -100	Response Code 400	Response Code 400	Pass

```

PS C:\Users\ASUS\Desktop\LMS\backend> py .\manage.py test api.tests.PaymentModelTest
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 3.334s

OK

```

Table 4.7: Test Case for Review

Project Name: Learning Management System					
Test Case 7					
Test Case ID: TC_7			Test Designed by: Gopal Pokhrel		
Module Name: Review			Test Executed by: Gopal Pokhrel		
Description: Testing the Review module of LMS					
SN	Test Case Description	Input	Expected Result	Actual Result	Status
1	Test give review	user=logged in user, course=django, rating=5, review="Great course!"	Given	Given	Pass
2	Test give review on same course again	user=logged in user, course=Django, rating=5, review="Great course!"	Failed	Failed	Pass

```

$ py manage.py test api.tests.ReviewModelTest
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 1.241s

OK

```

Table 4.8: Test Case for Discussion

Project Name: Learning Management System					
Test Case 8					
Test Case ID: TC_8				Test Designed by: Gopal Pokhrel	
Module Name: Discussion				Test Executed by: Gopal Pokhrel	
Description: Testing the Discussion module of LMS					
Additional Information: This module can be viewed and used by students who are enrolled in any one of the courses.					
SN	Test Case Description	Input	Expected Result	Actual Result	Status
1	Test create discussion	User=logged in user Section= Section of a Django Course Message=Test Message	Created	Created	Pass
2	Test get discussions	Fetch all discussions for that section	Fetches	Fetches	Pass
3	Test update discussion	Message=Updated message	Updated	Updated	Pass
4	Test delete discussion	Delete discussion of a given id	Deleted	Deleted	Pass

```
$ py manage.py test api.tests.DiscussionModelTest
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
....
-----
Ran 4 tests in 4.923s

OK
Destroying test database for alias 'default'...
(venv)
```

Table 4.9: Test Case for Recommendation

Project Name: Learning Management System					
Test Case 9					
Test Case ID: TC_9			Test Designed by: Gopal Pokhrel		
Module Name: Recommendation			Test Executed by: Gopal Pokhrel		
Description: Testing the recommendation algorithm of LMS					
Additional Information: This module can be viewed and used by everyone who is accessing the system.					
SN	Test Case Description	Step	Expected Result	Actual Result	Status
1	Getting recommendation	Visit a single course page	Suggestion of similar courses	Random courses suggested	Failed
2	Getting recommendation after fixing return statement in view code.	Visiting single course page	Suggestion of similar courses	Suggestion of similar courses	Pass

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Lesson Learnt

Developing the Learning Management System was a valuable experience that taught many important lessons. One key takeaway was the importance of focusing on the users. Understanding what instructors and students need and prefer was important for creating an easy-to-use platform. Regular feedback from supervisors and friends helped improve features and usability, ensuring the system met their expectations. Building a system that can grow and perform well was also essential. Using technologies like Next.js for the frontend, Django for the backend, and PostgreSQL for the database allowed the platform to handle more data and users without slowing down.

The project also highlighted the importance of good API design and thorough testing. Creating a strong RESTful API with Django allowed the frontend and backend to communicate smoothly, making the system easier to manage. Testing with tools like Postman ensured the system was stable and reliable, catching issues early and saving time and resources. Additionally, working on web design, backend API creation, and testing provided valuable hands-on experience. These lessons will be very useful for future software development projects, helping to create innovative and effective software applications.

5.2 Conclusion

The Learning Management System has met all its objectives effectively. Instructors can now create and upload courses, modules, and other educational content with ease. Students receive personalized course recommendations, making it easier for them to find relevant courses. Section-specific discussion spaces have been successfully implemented, providing students with areas to collaborate and engage in discussions. Overall, the system enhances the educational experience for both educators and students, fulfilling the intended goals.

5.3 Future Recommendations

Looking forward, enhancing the Learning Management System could involve integrating gamification elements to boost user engagement. Features like quizzes with interactive feedback and a leaderboard system can enrich the learning experience. Improving the recommendation engine with advanced algorithms and personalized learning analytics will

further tailor course suggestions to individual student needs, ensuring a more personalized educational journey. Additionally, creating a mobile application version of the platform would cater to users accessing educational content on smartphones and tablets, enhancing accessibility and user convenience.

APPENDICES

Implementation Code:

```
import pandas as pd

import math

from django.db.models import Q

from nltk.corpus import stopwords

def get_recommended_courses(current_slug):

    if not current_slug:

        return Course.objects.none()

    courses = Course.objects.filter(is_published=True).values_list(

        "id", "title", "instructor", "category", "slug"

    )

    instructors = User.objects.values_list("id", "full_name")

    categories = Category.objects.values_list("id", "name")

    courses_df = pd.DataFrame(

        courses,

        columns=["id", "title", "instructor", "category", "slug"],

    )

    instructors_df = pd.DataFrame(instructors, columns=["id", "full_name"])

    categories_df = pd.DataFrame(categories, columns=["id", "name"])

    instructors_df["full_name"] = instructors_df["full_name"].str.replace(" ", "")

    categories_df["name"] = categories_df["name"].str.replace(" ", "")

    courses_df["instructor"] = courses_df["instructor"].map(

        dict(instructors_df.values.tolist())

    )
```

```

courses_df["category"] = courses_df["category"].map(
    dict(categories_df.values.tolist())
)

stop_words = set(stopwords.words("english"))

def remove_stopwords(text):
    return " ".join(
        word for word in text.lower().split() if word not in stop_words
    )

courses_df["tags"] = (
    courses_df["title"]
    + " "
    + courses_df["instructor"]
    + " "
    + courses_df["category"]
).apply(remove_stopwords)

documents = [tags.split() for tags in courses_df["tags"]]

tf_matrix = []

for doc in documents:
    doc_tf = {}
    total_words = len(doc)
    for word in set(doc):
        doc_tf[word] = doc.count(word) / total_words
    tf_matrix.append(doc_tf)

unique_words = set(word for doc in documents for word in doc)

```

```

num_documents = len(documents)

idf = {}

for word in unique_words:

    doc_count = sum(1 for doc in documents if word in doc)

    idf[word] = math.log(num_documents / doc_count)

tf_idf_matrix = []

for tf_doc in tf_matrix:

    doc_tfidf = {}

    for word, tf in tf_doc.items():

        doc_tfidf[word] = tf * idf[word]

    tf_idf_matrix.append(doc_tfidf)

all_words = sorted(list(unique_words))

matrix = []

for doc_tfidf in tf_idf_matrix:

    row = []

    for word in all_words:

        if word in doc_tfidf:

            row.append(doc_tfidf[word])

        else:

            row.append(0.0)

    matrix.append(row)

similarity_matrix = []

def dot_product(v1, v2):

    return sum(x * y for x, y in zip(v1, v2))

```

```

def magnitude(v):

    return math.sqrt(sum(x * x for x in v))

def cosine_similarity(v1, v2):

    return dot_product(v1, v2) / (magnitude(v1) * magnitude(v2))

for i in range(len(matrix)):

    row = []

    for j in range(len(matrix)):

        similarity = cosine_similarity(matrix[i], matrix[j])

        row.append(similarity)

    similarity_matrix.append(row)

def recommend_courses(course_slug):

    if course_slug not in courses_df["slug"].values:

        return []

    course_index = courses_df[courses_df["slug"] == course_slug].index[0]

    similar_courses = list(enumerate(similarity_matrix[course_index]))

    sorted_similar_courses = sorted(

        similar_courses, key=lambda x: x[1], reverse=True

    )

    recommended_courses = []

    for i in sorted_similar_courses[1:5]:

        recommended_courses.append(courses_df.iloc[i[0]]["title"])

    return recommended_courses

recommended_courses = recommend_courses(current_slug)

if not recommended_courses:

```

```

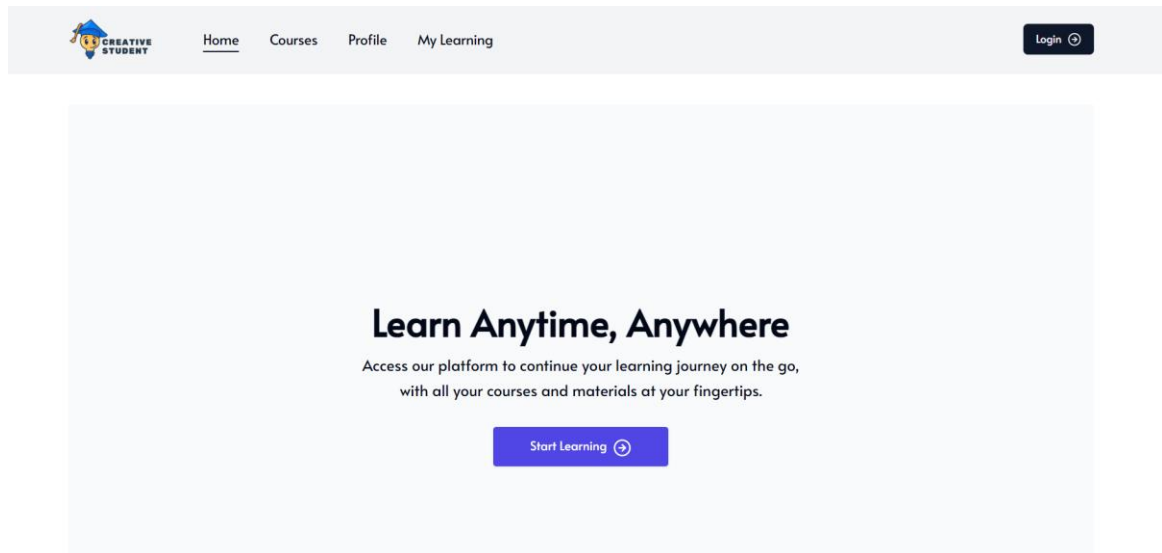

return Course.objects.none()

queryset = Course.objects.filter(title__in=recommended_courses)

return queryset

```

Screenshots of System

Create an account

Full Name


Email

Password

Confirm Password

☒ Student ☐ Instructor

Already have an account? [Login](#)




Login to your account

Email

Password


Don't have an account? [Register](#)



\$ 34.99

Digital Marketing Fundamentals 2024

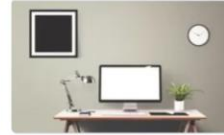
Suraj Katwal



\$ 39.99

Digital Marketing Fundamentals


Suraj Katwal



\$ 59.99

Practical Web Development: 22 Courses in I


Jose Portilla





\$ 49.99


Python for Data Science and Machine Learning Bootcamp


Jose Portilla











Full Course

\$ 15.00

Course Includes :

- Video Lectures
- Downloadable Resources

Rating


2492 Students

★★★★★

880 Ratings

Publisher

View profile



Suraj Katwal
Instructor

JavaScript Mastery

Great Course

Course Contents



33% Completed

Practical Web Development: 22 Courses in I

By Jose Portilla



75% Completed

Digital Marketing Fundamentals


By Suraj Katwal



Get Certificate

Learn Python Programming - Beginner to Master

By Abdul Bari



CERTIFICATE OF COMPLETION

UNISH KHADKA

Has successfully completed


LEARN PYTHON PROGRAMMING - BEGINNER TO MASTER

By continuing to learn, you have expanded your perspective, sharpened your skill, and make your self even more in demand.

Abdul Bari


ABDUL BARI

Certificate Recipient



Unish Khadka
unishkhadka@gmail.com

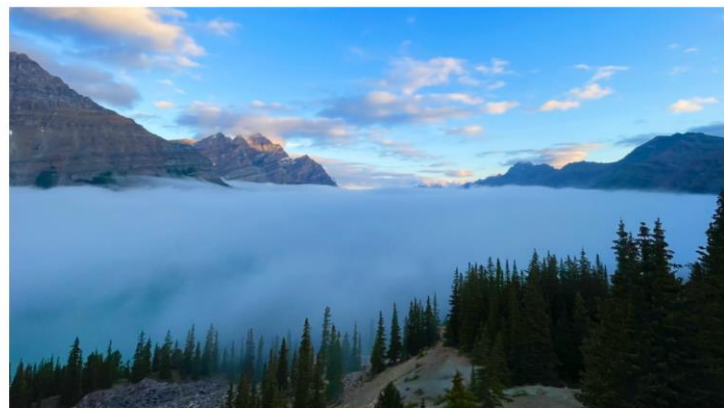
About the course



Learn Python Programming - Beginner to Master
Abdul Bari

[Download](#)

- ✓ Introduction to Python
- ✓ Python Installation and Setup
- ✓ Python Datatypes
- ✓ Regular Expressions




Introduction to Python


Discussions

Type your message


Send




Unish Khadka
Hello instructor, I did not understood about variables.




Abdul Bari
They are variables that holds values like a container.



Unish Khadka
Okay sir thank you !



Abdul Bari
Welcome Student !

 Reply to this message



Dashboard

Courses

Create Course



Suraj Katwal
surajk@gmail.com

Dashboard

**5**

Courses

**1**

Student

**\$ 39.99**

Income



Dashboard

Courses

Create Course



Suraj Katwal
surajk@gmail.com

Course Details

[Delete Course](#)[UnPublish](#)

Course Title

JavaScript Mastery

Course Category

Programming

Description

Great Course


Course Thumbnail



Choose File No file chosen

Price

15.00




Dashboard

Courses

Categories

Instructors



Gopal Pokhrel
pokhrelgopal27@gmail.com

Dashboard

**21**

Total Courses

**7**

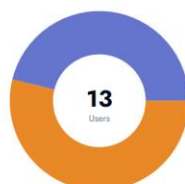
Total Enrollments

**Rs 344.94**

Total Transactions

Registered Users

as of September 24, 2024



This chart shows the number of users who are registered in the website.

Total Earnings each month

As of January - December



REFERENCES

- [1] S. K. Pal, “Software Engineering | Iterative Waterfall Model - GeeksforGeeks,” *GeeksforGeeks*, Mar. 18, 2018. <https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/> (accessed Sep. 23, 2024).
- [2] Azoft, “E-learning Platform Development [Case Study],” Azoft. <https://www.azoft.com/portfolio/e-learning-platform-development/> (accessed Aug. 12, 2024).
- [3] Coursera, “Coursera | Online Courses & Credentials by Top Educators. Join for Free,” Coursera, 2023. <https://www.coursera.org/> (accessed June. 18, 2024).
- [4] Udemy, “Online Courses - Learn Anything, On Your Schedule,” Udemy, 2023. <https://www.udemy.com/> (accessed June. 18, 2024).
- [5] “LinkedIn Learning: Online Training Courses for Creative, Technology, Business Skills,” LinkedIn Learning: Online Training Courses for Creative, Technology, Business Skills. <https://www.linkedin.com/learning> (accessed June. 18, 2024).
- [6] Khan Academy, “Khan academy,” Khan Academy, 2008. <https://www.khanacademy.org/> (accessed June. 27, 2024).
- [7] Edx, “edX,” EdX, 2019. <https://www.edx.org/> (accessed Sep. 18, 2024).
- [8] “Instructor-Led Online Training with 24X7 Lifetime Support | Edureka,” *edureka.co*, 2019. <https://www.edureka.co/> (accessed July. 04, 2024).
- [9] Skillshare, “Online Classes for Creatives | Skillshare,” *www.skillshare.com*, 2023. <https://www.skillshare.com/en/> (accessed July 04. 18, 2024).
- [10] M. Chiny, M. Chihab, O. Bencharef, and Y. Chihab, “Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms,” *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning*, 2021, doi: <https://doi.org/10.5220/0010727500003101/> (accessed Aug. 18, 2024).
- [11] riturajsaha, “Understanding TF-IDF (Term Frequency-Inverse Document Frequency),” *GeeksforGeeks*, Jan. 20, 2021. <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/> (accessed Aug 26. 18, 2024).