Online Learning Management System

24-FYP-201

Session 2020-2024

BACHELOR OF SCIENCE IN

SOFTWARE ENGINEERING

SUBMITTED BY

Muhammad Sami 20-NTU-CS-1127

Syed Nad E Ali Naqvi 20-NTU-CS-1136

SUPERVISED BY

Dr. Isma Hamid (supervisor)

Mr. Arsal Mahmood (Co. supervisor)

Department of Computer Science

National Textile University, Faisalabad

CERTIFICATION

This Document, which has been written by **Muhammad Sami(20-NTU-CS-1127)** and **Syed Nad e Ali Naqvi(20-NTU-CS-1136)** under the directions of our supervisor and co-supervisor and approved by all members of the thesis committee, has been presented to and accepted Chairman of, the Department of Computer Science, in fulfillment of the requirement of the degree of Bachelor of Science in Software Engineering.

Supervisor: Dr. Isma Hamid	**Date**
Co-Supervisor: Mr. Arsal Mahmood	**Date**
Internal Examiner (1)	**Date**
Internal Examiner (2)	**Date**
FYP-Convener Mr. Muhammad Shahid	**Date**
Head of Department Dr. Asif Malik	**Date**

DECLARATION

We hereby declare that we completely write this document, and it is totally our effort and none of anyone from outside of our group has copied it. This report is purely written in a technical way in accordance with our project.

Group Members:

Muhammad Sami	_____	_____
20-NTU-CS-1127	Signature	Date
Syed Nad e Ali Naqvi	_____	_____
20-NTU-CS-1136	Signature	Date

ACKNOWLEDGEMENT

More than anything else, For the sake of Allah, the Most Advantageous, the Most Benevolent. Every one of the gestures of recognition also, thanks be to Allah. A ton of adoration for our cherished Heavenly Prophet MUHAMMAD (S.A.W), his direction assists us with getting the correct way. From that point onward, we additionally gladly announce that the achievement boundaries of this venture are the vast supplications and backing of our folks. We likewise need to recognize the endeavors of our supervisor, who helped us a lot in the project, particularly in the documentation. Each time we need to meet him, he is continuously there for us with good thoughts, conversation and ideas that help us a great deal in fruition of our project and documentation. We likewise need to thank the individuals who helped us in various phases of the venture. We are exceptionally appreciative to the Program Director, Bachelor of Software Engineering, and final year project convener for outrageous strength, mindful, persistence, delicacy, liberality, extraordinary mindfulness and for propelling us to do something remarkable and progressive.

Presently venturing ahead, we wish to pay our glow of appreciation and gratefulness to our noteworthy supervisors who delivered their benevolent administrations to help us in various parts of the trouble spot. We are appreciative to our most regarded educators and Dr.Isma Hamid and Mr.Arsal Mehmood for being dependably some assistance in all respects of programming standards, this aided us in all advancement levels of the undertaking also, particularly for making us ready to push ahead in the field of computer programming furthermore, plan. We gladly proclaim that the main driver of our outcome in the field of programming designing are our instructors by their strong disposition for continuously being a making a difference hand for us. We cannot overlook the help of the IT administrations division. Particularly the IT administrations staff for giving us a spot to sit and work in the IT division and the related help on request. Their aiding demeanor was amazing. Finally, yet all the same not the least, we additionally need to thank our class colleagues particularly for sacrificial assistance, counsel, and ideas in various stages going from elicitation to organization.

ABSTRACT

This introduces a new Online Learning Management System (LMS) designed to address our current challenges with outdated technology and limited functionality. The LMS features a master panel for efficient user access control, allowing administrators to manage permissions and ensure security. A user-friendly interfaces to easily create, update, and publish course content, manage assessments, quizzes, and facilitate real-time chat with instructors. This streamlined approach aims to enhance productivity, ensure timely updates, and provide a more efficient and engaging online learning experience for our Educational community.

TABLE OF CONTENT

CERTIFICATION ii

DECLARATION iii

ACKNOWLEDGEMENT iv

ABSTRACT v

TABLE OF FIGURES x

TABLE OF TABLE xi

Chapter 1: Introduction 12

1.1: Introduction : 13

1.2: Problem Statement: 14

1.3: Purpose: 14

1.4: Project goal: 14

1.5: Project Objectives: 15

1.6: Project Scope: 15

1.7: Types of Risks: 16

1.7.1: Cost Risk: 16

1.7.2: Performance Risk: 16

1.7.3: Scheduling and Time Risk: 16

1.7.4: Unforeseen Circumstances Risk: 17

1.7.5: Risk in Development step: 17

1.8: Project Planning: 17

1.9: Gant Chart: 19

Chapter 2: Literature Review 20

2.1: Related Work: 21

i. React Multi-App Manager: 21

2.2: Area of Studies: 21
2.3: Reason of development: 22
Chapter 3: Methodology 23
3.1: Methodologies for Software Development: 24
3.2: Existing Methodologies: 25
Chapter 4: System Requirement 29
4.1: Functional Requirements: 30
4.1.1 User Authentication and Authorization: 30
4.1.2 Course Management: 30
4.1.3 User Interaction: 30
4.1.4 Content Management: 30
4.1.5 Assessment and Grading: 31
4.1.6 Reporting and Analytics: 31
4.1.7 Mobile Accessibility: 31
4.1.8 Integration with External Systems: 31
4.2: Non-Functional Requirements for Online Learning Management System 32
4.2.1: Performance Requirements 32
4.2.2: Security and Reliability 32
4.2.3: Compliance with Regulations 33
4.2.4: Usability 33
4.3: Application Users: 34
4.3.1:Administrators: 34
4.3.2:Instructors / Teachers: 34
4.3.3:Students: 35
4.3.4:System Support / Helpdesk: 35

4.3.5:External Auditors / Inspectors (if applicable): 35
4.3.6:Guest Users : 36
Chapter 5: Architecture Design 37
5.1: Introduction to System Design: 38
5.2: Behavioral UML Diagram: 38
5.2.1: Use Cases: 39
5.2.2:Sequence Diagram: 44
5.2.3: Class Diagram: 47
5.2.4: ERD: 48
Chapter 6: Prototype 49
6.1: Home Page: 50
6.2: Sign Up Page: 50
6.3: Login Page: 51
6.4: Forget Password: 51
6.5: Reset Password: 52
6.6: Contact: 52
6.7: View Course: 53
6.8: Single Course Detail: 53
6.9:Quiz: 54
6.10: Quiz Result: 55
6.11: User chat with instructor: 56
6.12: User course enrolled: 56
6.13: Watch course page: 57
6.14:Search courses: 57
6.15: Our Instructors: 58

6.16: Search Instructors: 58
6.17: Admin login: 59
59
6.18: Admin Home: 59
6.19: Admin Course Creation: 60
6.20: Admin Quiz Creation: 60
Chapter 7: Implementation and Testing 62
7.1 White Box Testing: 63
7.2 Black box Testing: 63
7.3: Verification 64
7.4: Validation 65
7.5: Adopted Methodology 65
7.5.1: Unit testing 65
7.5.2: Module Testing 65
References 67

TABLE OF FIGURES

Figure 3.1: Iterative water fall model 25
Figure 3.2: Agile(Scrum) 26
Figure 5.1:Administrator Use Case 39
Figure 5.2:InstructorAuthor(Teacher) 41
Figure 5.3:Students Panel 42
Figure 5.4 :Administrator sequence diagram 44
Figure 5.5:Teacher sequence diagram 45
Figure 5.6:Student sequence diagram 46
Figure 5.7: Class Diagram 47
Figure 5.8.:ERD 48
Figure 6.1:Home Page 50
Figure 6.2:Sign Up Page 51
Figure 6.3:Login Page 51
Figure 6.4:Forget Password 51
Figure 6.5:Reset Password 52
Figure 6.6:Contact 52
Figure 6.7: View Course 53
Figure 6.8:Single Course Detail 53
Figure 6.9 :Quiz 54
Figure 6.10 :Quiz Page1 54
Figure 6.11:Quiz Page 2 55
Figure 6.12: Quiz Result 55
Figure 6.13:User chat with instructor 56
Figure 6.14:User course enrolled 56
Figure 6.15:Watch course page 57
Figure 6.16:Search courses 57
Figure 6.17:Our Instructor 58
Figure 6.18:Search Instructors 58
Figure 6.19:Admin Login 59
Figure 6.20:Admin Home 59
Figure 6.21: Admin Course Creation 60
Figure 6.22:Admin Quiz Creation 60
Figure 6.23:Quiz Question Added by admin 61
Figure 7.1:Black box testing 64

TABLE OF TABLE

Table 1.1 :Project Planning 18
Table 1.2 :Gant Chart 19
Table 5.1 :Administrator 40
Table 5.2:Instructor(Teacher) 41
Table 5.3:Students Panel 43

Chapter 1: Introduction

1.1: Introduction:

In the ever-evolving landscape of education, the demand for a seamless and comprehensive Online Learning Management System (LMS) has become increasingly imperative for academic institutions and colleges alike. The multifaceted responsibilities, both administrative and academic, can be efficiently streamlined and automated through the incorporation of a robust LMS. Designed with various modules.

Much like their campus-based counterparts, Online Learning Management Systems provide a centralized platform to oversee the entirety of academic operations, spanning from financial and administrative duties to the intricacies of academic and research pursuits. Beyond the fundamental features like

1. Online Course registration,
2. Grade management,
3. Student record tracking

The transformative impact of LMS goes beyond organizational efficiency, playing a pivotal role in elevating the experiences of students, teachers, and staff. By offering an integrated and streamlined approach to online education, these systems contribute to increased productivity, reduced administrative workloads, and a more cohesive learning environment. The adaptability of LMS ensures that institutions of all sizes, from modest colleges to expansive universities, can harness the benefits of cloud-based solutions, making these systems increasingly affordable and accessible.

Moreover, Online Learning Management Systems serve as invaluable tools for data reporting and analysis, empowering educational organizations to make informed, data-driven decisions. This not only enhances immediate performance but also facilitates continuous improvement over time. As the realm of cloud-based solutions continues to expand, the accessibility and affordability of these systems broaden, marking a pivotal shift in the way institutions approach online education. In essence, the introduction of an LMS heralds a new era in education management, fostering efficiency, collaboration, and excellence in the digital realm.

1.2: Problem Statement:

The current Online Learning Management System (LMS) suffers from disorganized content, limited accessibility, and inadequate tracking of learner progress. Outdated technology and communication bottlenecks reduce system effectiveness. Compliance with mandatory training and scalability issues add to the challenges. A modern, user-friendly, and scalable LMS is essential to improve the learning experience, streamline content management, and adapt to the evolving online education landscape. Developing such an LMS will offer a comprehensive solution for educators, learners, and administrators, even **without chat functionality and useability factors as well in technologies**

1.3: Purpose:

The purpose of creating a online Learning management system is to overcome theshortcomings of the existing one, which is based on antiquated technology and lacks key components needed for contemporary content management. The new system will offer an updated and more effective platform for handling and distributing digital content, enabling better user experiences, more efficient processes, and enhanced cooperation and communication among campus resident

1.4: Project goal:

To replace and re-create a dynamic website utilizing server-side programming languages, a new Online Learning management system is being developed. The new system will offer a more adaptable, configurable, and effective platform for storing and providing digital information to the university community, allowing for a better user experience, and streamlined workflows. It will do this by utilizing contemporary technologies and programming languages like **Python (Django) or Django Rest Framework**.

1.5: Project Objectives:

Following are the main objectives

I. To develop a system that can work efficiently.

II. To perform all activities in less time.

III. To minimize the whole workload

IV. To give the campus community a centralized and effective platform for managing digital content.

V. To host and manage learning materials, courses, and resources in one platform.

VI. To use current technologies and programming languages to build a flexible and adaptable system.

VII. To incorporate new features and functionality including online chat, and customized landing pages to get beyond the constraints of the outdated system.

VIII. To improve the user experience for teachers, staff, and students by making it simpleand quick to access crucial data, resources, and services via an engaging website.

IX. To create a more effective and productive work environment by streamlining workflows and enhancing collaboration and communication among campus residents.

1.6: Project Scope:

The project scope for the Online Learning Management System (LMS) encompasses the development and implementation of a centralized platform aimed at facilitating seamless access to educational content. **This system will provide a user-friendly interface for managing various learning materials, including multimedia resources, courses, and assessments.** The scope extends to ensuring scalability to accommodate a growing user base and diverse content. Personalized learning paths will be a key feature, allowing customization based on individual learner needs. The security standards and regulations is integral to the scope, as is the provision of training resources and support for administrators, instructors, and learners .The scope also encompasses cost optimization measures, **mobile compatibility(means you can run this website on your mobiles)** user role management.

1.7: Types of Risks:

Risks are very important factors in the projects that are related to software or building architecture. Some of the risks involved in this project are mentioned below:

- Cost Risk
- Performance Risk
- Scheduling and Time Risk
- Unforeseen Circumstances
- Risk in Development step
- Requirement elicitation
- Architecture and Design
- Coding and Implementation

1.7.1: Cost Risk:

Project cost exceeding the budget introduces cost risk, a crucial factor in project development. During project planning, meticulous calculations and approximate cost estimates are recorded for each step. To mitigate cost risk, ongoing cost analysis is essential throughout the project development phase. This helps maintain project performance, ensuring that scope and quality align with the baseline budget.

1.7.2: Performance Risk:

In our project, performance is key. We aim to create an integrated solution that efficiently executes various functions. From course transactions to resource management, all activities will be optimized to enhance overall system performance. Recognizing the importance of performance, we prioritize it throughout the entire development process.

1.7.3: Scheduling and Time Risk:

In our project, we consider scheduling and time risk as crucial factors. Scheduling risk arises when project activities take longer than planned, potentially causing delays in project delivery. To manage this, we've divided the system into modules with specific development timelines, aiming to schedule the entire project

effectively. Scheduling risk is common in any plan, and while it's challenging to predict due to unforeseen events, we're mindful of addressing it throughout the development process.

1.7.4: Unforeseen Circumstances Risk:

In our project refers to unexpected events. These could include team members taking sick leaves, technical failures, the introduction of new technology, or even natural disasters. While these events may or may not have a significant impact on the project development, we acknowledge their unpredictability and plan to address them as they arise.

1.7.5: Risk in Development step:

In developing our System, we're taking a step-by-step approach by breaking it into modules. This method not only saves time but also helps prevent potential issues later on. Throughout the process, a dedicated project manager oversees the work of the team members, ensuring everyone stays on track. Clear communication is maintained to address any challenges promptly and keep the development process smooth.

1.8: Project Planning:

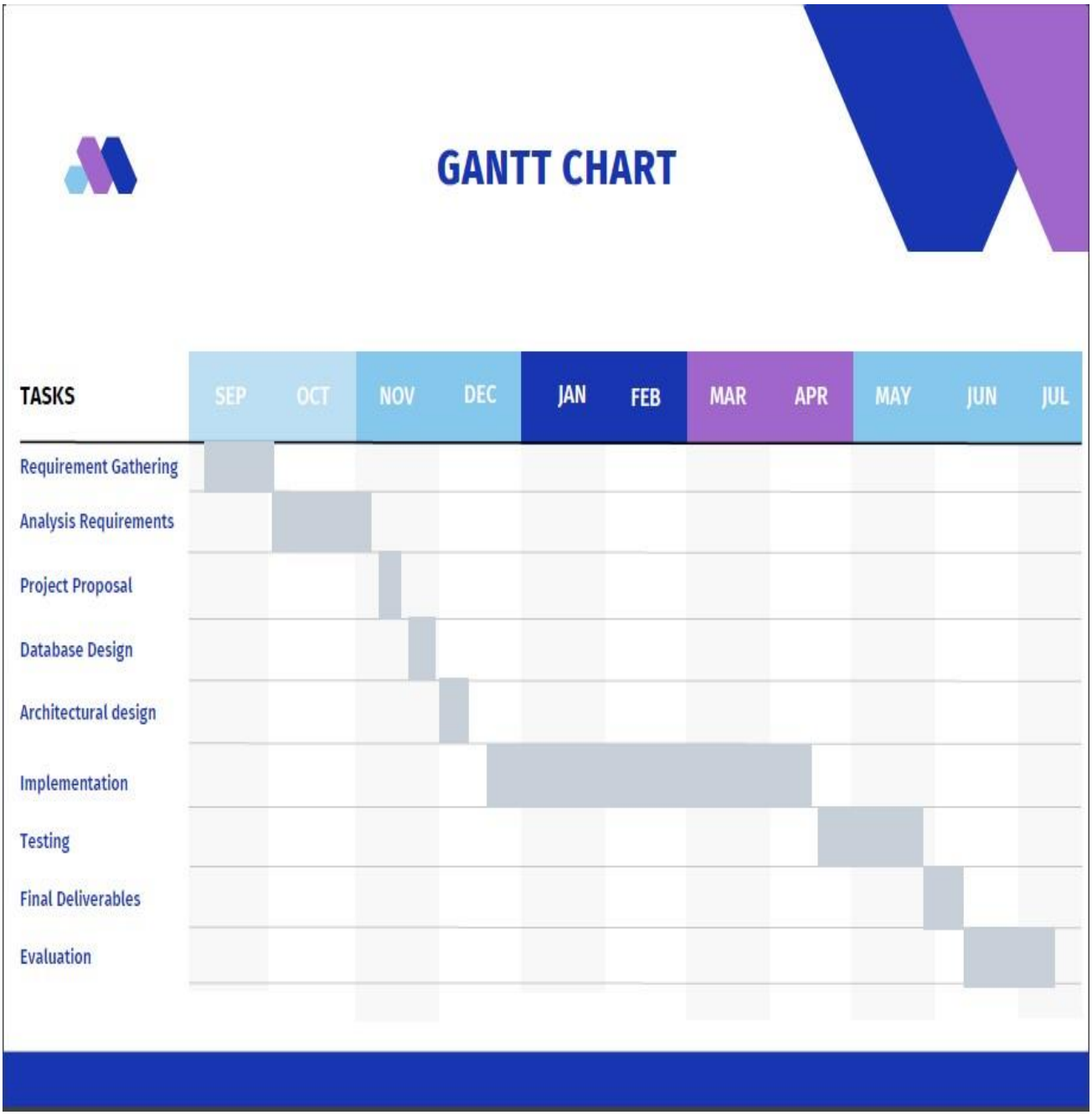
Project planning is done by using tool Microsoft Project. We will achieve our task completion within given time which is written in given table and display in the figure:

Table 1.1 Project Planning

Task Name	Duration	Start	Finish	Predecessors
Requirement Gathering	20 days	Tue 03/08/23	Mon 30/08/23	
Analyses Requirement	25 days	Tue 31/08/23	Mon 04/10/23	1
Project Purposal	7 days	Tue 05/10/23	Wed 13/10/23	1,2
Database Design	8 days	Thu 14/10/23	Mon 25/10/23	1,2,3
Architectural Design	10 days	Tue 26/10/23	Mon 08/11/23	1,2,3,4
Implementation	90 days	Tue 09/11/23	Mon 14/03/24	1,2,3,4,5
Testing	30 days	Tue 15/03/24	Mon 25/04/24	1,2,3,4,5,6
Final Deliverable	10 days	Tue 26/04/24	Mon 09/05/24	1,2,3,4,5,6,7
Evaluation	30 days	Tue 10/05/24	Mon 20/06/24	1,2,3,4,5,6,7,8

1.9: Gant Chart:

Table 1.2 Gant Chart



Chapter 2: Literature Review

2.1: Related Work:

There are several related works to a Online management system that create dynamic websites through code. Here are a few examples:

- **JavaScript Multi-App Manager:** A single installation of the JavaScript Multi-App Manager empowers developers to create and manage multiple websites seamlessly. Using JavaScript, it becomes feasible to build a network of applications that share a common database and codebase. This architecture simplifies administration tasks, allowing for efficient updates across all applications simultaneously. The modular and component-based nature of JavaScript facilitates the creation of dynamic and interactive user interfaces, enhancing the overall user experience. With the JavaScript Multi-App Manager, developers can harness the power of JavaScript to streamline the management and development of a diverse portfolio of web applications.
- **Django LMS:** The Django LMS is a Learning management system that is partof the Django web framework. It offers a user-friendly interface for content management and permits the coding of dynamic websites.

- **Django Rest Framework:** A rest API for managing and producing website content is offered by the Learning management system. It makes it possible for programmers to create dynamic websites and makes it simple to collaborate and Apply the fetching data on front-end.

2.2: Area of Studies:

Our project on developing an **Online Learning Management System (LMS)** draws upon various domains of computer science to ensure a comprehensive and well-executed solution. Software engineering principles have been instrumental in guiding the project's development, particularly in determining the workflow and adopting the agile methodology. This approach has proven effective in promoting adaptability and responsiveness throughout the development lifecycle. Additionally, software engineering practices have been crucial in time management and scheduling, enabling us to meet project milestones efficiently. The project also extensively utilizes **Software Architecture and Design methodologies** to document and structure the LMS effectively. This involves creating

a robust architectural framework that not only supports the current functionalities but also allows for future scalability and enhancements. By integrating these diverse aspects of computer science, **our Online Learning Management System project is underpinned by sound engineering practices, ensuring a well-organized, agile,** and scalable solution for the dynamic landscape of online education.

2.3: Reason of development:

Embarking on the development of an Online Learning Management System (LMS) as a senior project offers students a unique opportunity to translate their acquired knowledge and skills into a tangible, real-world application. This project not only serves as a culmination of theoretical learning but also allows students to engage with the practical complexities of the ever-evolving field of development. By delving into the creation of an LMS, students have the chance to investigate and implement cutting-edge technologies, crucial for staying abreast of the rapid advancements in the web development landscape. The significance of this endeavor extends beyond individual skill enhancement, as the resulting LMS could potentially make a substantial contribution to the academic community. Online Learning Management Systems are indispensable tools for educational organizations, facilitating streamlined content delivery and management. Thus, by undertaking the development of an LMS, students not only enhance their own capabilities but also contribute to the broader mission of advancing educational technology and providing practical solutions for modern learning environments.

Chapter 3: Methodology

3.1: Methodologies for Software Development:

Software development is a crucial step in ensuring projects are completed successfully. It breaks down the development process into smaller, manageable phases such as system and program design, implementation, various levels of testing, and maintenance. This structured approach helps in enhancing the quality and effectiveness of the project.

The selection of a development methodology is a critical decision that influences how well the project's challenges can be addressed. Each project has its unique nature, which necessitates a specific approach to tackle its complexities efficiently. Therefore, the choice of methodology is based on the specific requirements and characteristics of the project at hand.There's a wide array of methodologies available, each with its own set of principles and focus areas. For example, some methodologies prioritize risk management, while others emphasize on extensive testing to ensure the project meets its objectives. Among the most commonly used methodologies are:

- **V-model:** It's a sequential process where each phase must be completed before the next begins, with a strong emphasis on planning and testing at various stages.
- **Spiral model:** This model combines elements of both design and prototyping in stages, making it flexible and allowing for risk assessment at each cycle.
- **Scrum:** Part of the Agile framework, Scrum is iterative and incremental, focusing on collaboration, regular feedback, and small, manageable chunks of work known as sprints.
- **Incremental model:** This approach divides the project into small parts, allowing teams to deliver segments of functionality gradually, leading to a more manageable workflow.
- **Rapid development:** It aims to speed up the development process, often through the use of software development tools, less formal team communications, and rapid prototyping.
- **Agile method:** Known for its flexibility and responsiveness to change, Agile encourages continuous feedback, iterative development, and adaptability throughout the project lifecycle.

Each of these methodologies offers different benefits and suits different project types. The selection

process involves understanding the project's requirements, team dynamics, client expectations, and potential risks to choose the most appropriate approach that ensures project success.

3.2: Existing Methodologies:

The **Software Development Life Cycle (SDLC)** is crucial for the efficient and high-quality development of software, encompassing various steps and methodologies. Among the methodologies mentioned, the Spiral Model and Iterative Waterfall Model are briefly introduced. However, after careful consideration, a hybrid methodology is chosen for the effective development of the system, incorporating the Iterative Waterfall Model and Agile Model, while excluding the Spiral Model.

Selected Methodology: Hybrid Methodology

1. Iterative Waterfall Model:

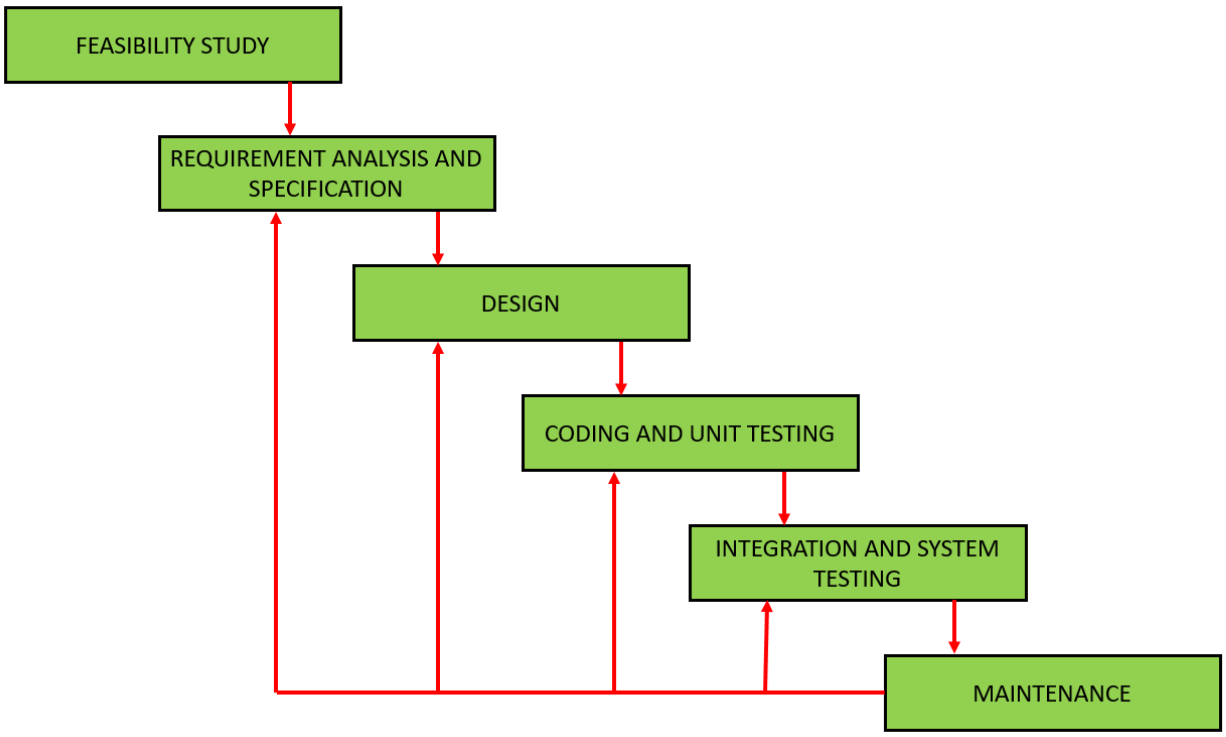


Figure3.1: Iterative water fall model

The Iterative Waterfall Model is chosen because it allows flexibility by enabling developers to revisit previous steps or phases. This flexibility becomes essential when there are changes in requirements, as the model supports modifications even after initial phases are completed. This iterative approach reduces time consumption and efforts needed for error detection and correction.

2. Agile Model (Scrum):

The Agile Model is selected, specifically adopting the Scrum framework. Scrum aligns well with the iterative and flexible nature of the project. It emphasizes incremental and continuous development, ensuring that the software evolves through collaboration between cross-functional teams and end-users. Scrum's iterative cycles, known as sprints, allow for gradual releases and refinements, making it adaptable to changing requirements.

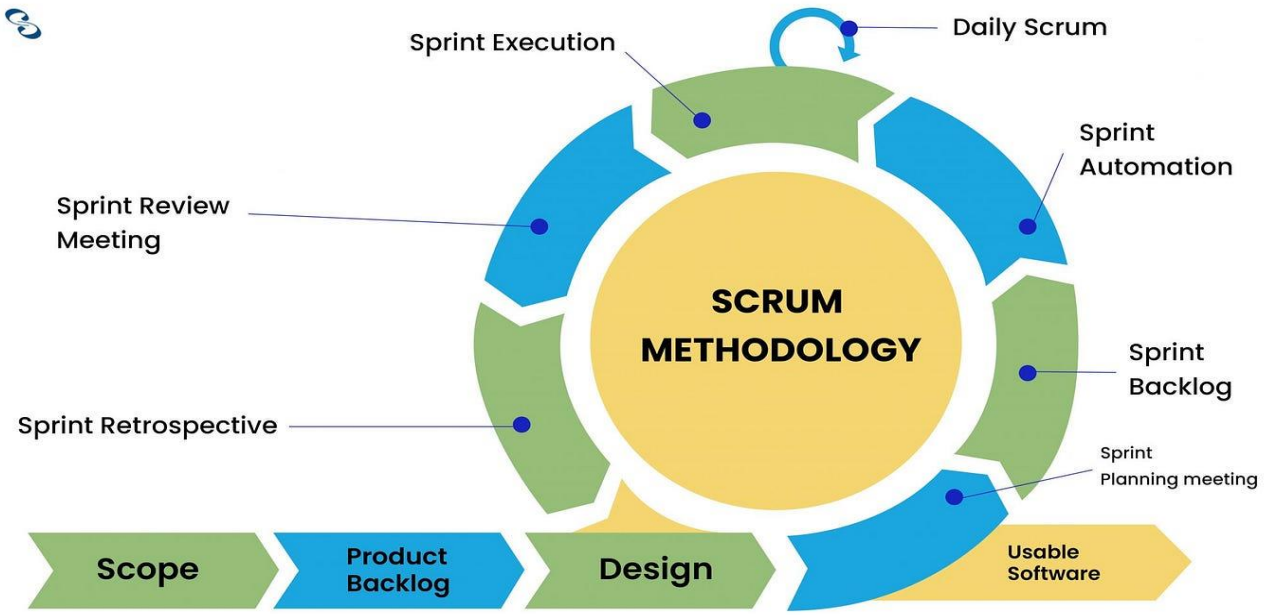


Figure 3.2: Agile(Scrum)

Why Scrum in the Hybrid Methodology?

The Scrum framework within the Agile Model introduces structured project management practices. Scrum's key elements, including roles (Scrum Master, Product Owner, and Development Team), artifacts (Product Backlog, Sprint Backlog, and Increment), and events (Sprint Planning, Daily Stand-ups, Sprint Review, and Sprint Retrospective), enhance project transparency, collaboration, and adaptability.

- **Sprint Planning:** Enables the team to define the scope of work for the upcoming sprint.
- **Daily Stand-ups:** Facilitates daily communication, addressing challenges and ensuring alignment.
- **Sprint Review:** Involves stakeholders in evaluating the increment and adjusting the product backlog.
- **Sprint Retrospective:** Allows the team to reflect on the sprint and identify areas for improvement.

Why Hybrid Methodology?

The hybrid methodology combines the strengths of the Iterative Waterfall Model, Agile Model (Scrum), and incorporates the benefits of Scrum's project management practices. It allows for flexibility and modifications in response to changing requirements, as provided by the Iterative Waterfall Model. Simultaneously, the Agile (Scrum) Model brings in continuous collaboration, incremental development, and structured project management practices.

Benefits of the Hybrid Methodology:

- **Flexibility:** Ability to revisit and modify previous steps if requirements change.
- **Continuous Collaboration:** Involves regular communication with stakeholders, ensuring alignment with evolving project goals.

- **Incremental Development:** Enables gradual releases and refinements, providing tangible results at each stage.
- **Structured Project Management:** Scrum's framework enhances project transparency, collaboration, and adaptability.

Chapter 4: System Requirement

4.1: Functional Requirements:

4.1.1 User Authentication and Authorization:

Objective: Ensure secure and controlled access to the LMS.

Sub-requirement 1: Implement a robust user authentication system

Sub-requirement 2: Define user roles (administrator, instructor, student) with specific access permissions.

4.1.2 Course Management:

Objective: Facilitate efficient creation, organization, and delivery of courses.

Sub-requirement 1: Enable instructors structure courses, including modules, lessons, and assessments.

Sub-requirement 2: Include version control for course content and updates.

4.1.3 User Interaction:

Objective: Foster effective communication and collaboration among users.

Sub-requirement: Implement for user easy and efficent interface.

4.1.4 Content Management:

Objective: Support various types of educational content delivery.

Sub-requirement 1: Allow the upload and management of diverse content formats, including videos, and interactive media.

Sub-requirement 2: Integrate with external content repositories or cloud storage services.

4.1.5 Assessment and Grading:

Objective: Facilitate fair and efficient evaluation of student performance.

Sub-requirement 2: Enable automated grading where applicable, with options for manual review.

Sub-requirement 3: Implement a my learning page with show for student progress.

4.1.6 Reporting and Analytics:

Objective: Offer insights into system usage and student performance.

Sub-requirement 1: Generate customizable reports on course participation, completion rates, and assessment results.

Sub-requirement 2: Implement Panel tool for administrators to assess system performance and identify areas for improvement.

4.1.7 Mobile Accessibility:

Objective: Ensure users can access and interact with the LMS on mobile devices.

Sub-requirement: Design a responsive user interface that adapts to various screen sizes.

4.1.8 Integration with External Systems:

Objective: Enable seamless data exchange with other educational tools and systems.

Sub-requirement: Support integration with external learning tools and applications.

4.2: Non-Functional Requirements for Online Learning Management System

4.2.1: Performance Requirements

4.2.1.1: Scalability

- Define the specified number of simultaneous users the system should handle. This involves understanding the expected user load and ensuring that the system can scale up or down to accommodate varying levels of demand.
- Consider load balancing mechanisms to distribute user requests evenly across servers or resources.

4.2.1.2: Response Time

- Establish specific benchmarks for the minimal acceptable response time for user interactions within the system. This could include actions such as loading a page, submitting a form, or accessing content.
- Performance testing should be conducted to measure and optimize response times under various conditions.

4.2.2: Security and Reliability

4.2.2.1: Secure Authentication Protocols

- Detail the specific authentication mechanisms, such as multi-factor authentication, OAuth, or other secure login methods, to ensure only authorized users access the system.

- Regularly update and patch authentication systems to mitigate potential vulnerabilities.

4.2.2.2: Regular System Backups and Data Encryption

- Specify the frequency of system backups and the mechanisms used for data encryption during storage and transmission.
- Define the recovery time objectives (RTO) and recovery point objectives (RPO) to ensure quick system restoration in case of data loss or system failures.

4.2.3: Compliance with Regulations

4.2.3. 1: Data Privacy Regulations

- Clearly outline the specific data privacy regulations that the system must comply with, such as GDPR, HIPAA, or other relevant regional laws.
- Implement data anonymization and pseudonymization techniques where necessary to protect sensitive user information.

4.2.3.2: Ethical Considerations

- Articulate ethical guidelines for content delivery and user interactions within the system.
- Define how the system will address potential biases in content or algorithms to ensure fair and unbiased user experiences.

4.2.4: Usability

4.2.4. 1: User Interface Design

- Define user interface guidelines to ensure a consistent and user-friendly design.

- Conduct usability testing to gather feedback on the system's ease of use and make iterative improvements.

4.2.4. 2: Accessibility

- Ensure the system complies with accessibility standards (e.g., WCAG) to make it usable for people with disabilities.
- Provide alternative text for images, ensure keyboard navigation, and use semantic HTML for better accessibility.

4.3: Application Users:

In an Online Learning Management System (LMS), various user roles contribute to the overall functionality and success of the platform. Here are some common application users and their roles in an LMS:

4.3.1:Administrators:

➢ Responsibilities:

- Administrators is in charge of setup and system configuration.
- They manage user accounts, adding, removing, or modifyin' them as needed.
- Administrators oversee courses, creating, updating, and archiving.
- Accessing analytics and reporting' features to keep an eye on system usage.

4.3.2:Instructors / Teachers:

➢ Responsibilities:

- Chat with the Users on the Real time Integrations

4.3.3:Students:

Responsibilities:

- They access course materials like Video lectures, and multimedia content links.
- Participate in the Checking their Lessons as well.
- Check their semester results.

4.3.4:System Support / Helpdesk:

➢ Responsibilities:

- Provide technical support to users facin' issues with the platform.
- Addressin' user inquiries related to functionality, access, or technical difficulties is the task.
- Offerin' guidance on usin' various features within the LMS is what they're here for.

4.3.5:External Auditors / Inspectors (if applicable):

➢ Responsibilities:

- Conductin' assessments to ensure compliance with educational standards is their gig.
- Reviewin' data and records related to courses, student performance, and system usage is also in their wheelhouse.
- Ensuring the LMS adheres to regulatory requirements and standards is a crucial aspect.

4.3.6: Guest Users / just visiting users :

- Responsibilities:

- Explore available courses and program offerings.
- Access demo content or introductory materials.

Chapter 5: Architecture Design

5.1: Introduction to System Design:

The process of system design is pivotal in crafting the blueprint of a system, ensuring that its architecture, components, modules, interfaces, and data are all aligned to meet specified criteria. This task is inherently interdisciplinary, drawing on systems analysis for understanding and

specification, systems architecture for high-level design, and systems engineering for holistic integration and problem-solving. The architectural design of software plays a critical role in this process, offering a macro view of the system's structure and functionality.

5.2: Behavioral UML Diagram:

In the context of an Online Learning Management System (LMS), a Use Case Diagram serves as a crucial tool for visually representing the interactions between users (actors) and the system, highlighting the system's functionalities from a user's perspective. This diagram helps in understanding how different users engage with the LMS and outlines the services the system provides to each user type.

This system consists of three actors:

- Students (Learners)
- Instructors(Teachers)
- Administrators

The below section shows the use case diagrams of these three actors:

5.2.1: Use Cases:

5.2.1.1: Administrator Use case:

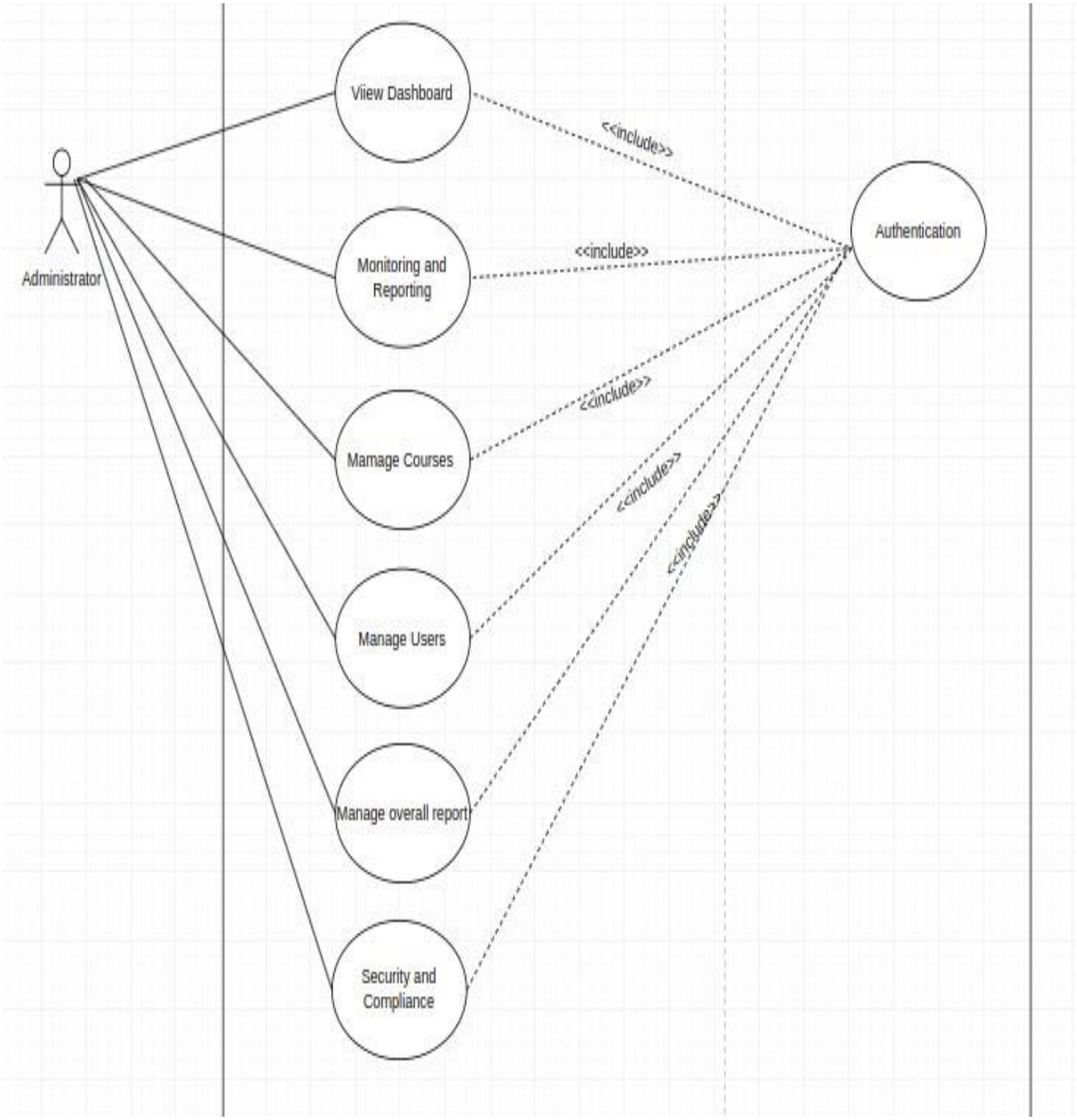


Figure 5.1:Administrator Use Case

Table 5.1 Administrator

Use case Id & Title	Description
UCI 1: View Dashboard Actor: Administrator	Administrator can view the dashboard of the website and can manage all the activities on the dashboard.
UCI 2: Manage any Transaction	Configure and customize the LMS settings.
UCI 3: Manage Courses	Administrator can manage the Courses which are being added or droped.
UCI 4: Manage Users	Administrator can manage all the users include(teachers, Students).
UCI: Manage overall report	Administrator can manage all the Report and Queries against the system or from the students and teachers.

5.2.1.2: Instructor(Teacher) Use case:

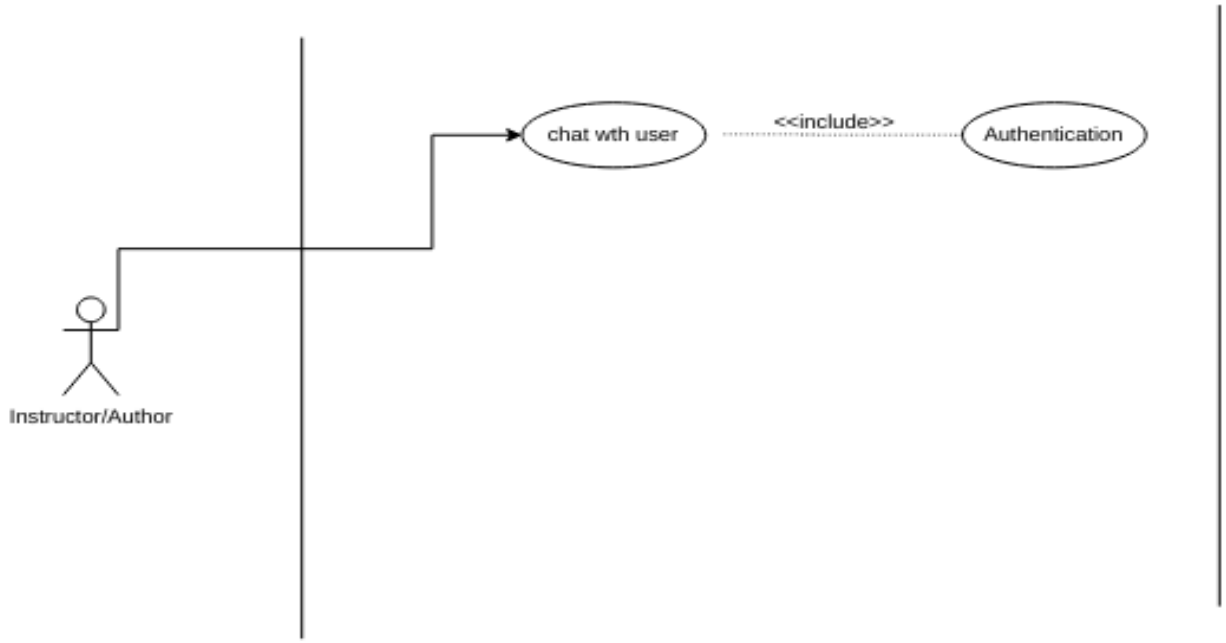


Figure 5.2:InstructorAuthor(Teacher)

\

Table 5.2 Instructor/Author(Teacher)

Use case Id & Title	Description
UCI 1: Direct chat with user	The Teacher have facility to direct Chat with the User or Student who can Enrolled in the course that the teacher has made.

5.2.1.3: Students Use case:

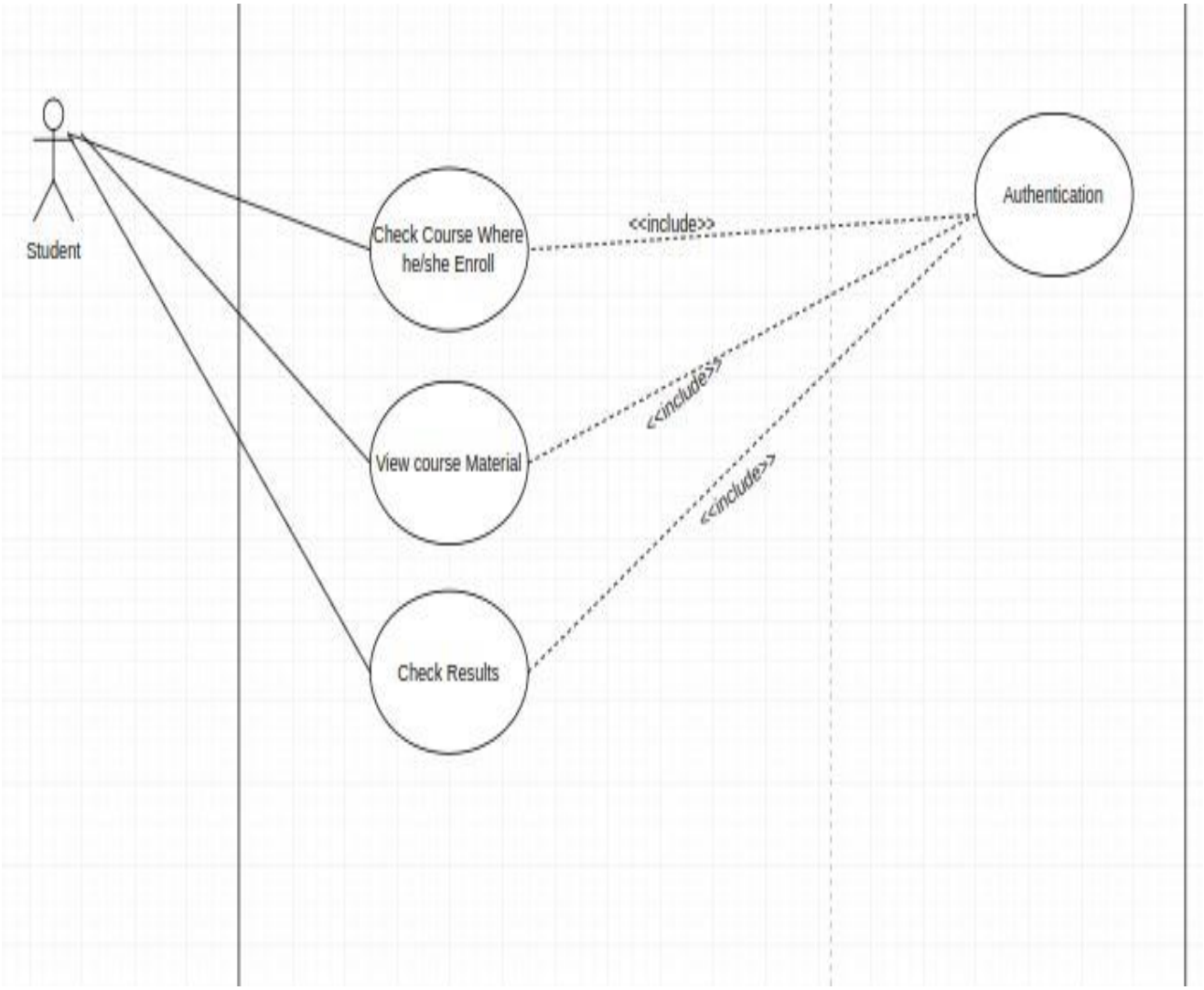


Figure 1.3:Students Panel

Table 5.3 Students Panel

Use case Id & Title	Description
UCI1: Check courses Where he/she Enrolled	Students can view all courses.
UCI 3: View Course Material	Students can check their material also in the course.
UCI 4: Check Results	Student can check their Result as well.

5.2.2:Sequence Diagram:

5.2.2.1: Administrator Sequence Diagram:

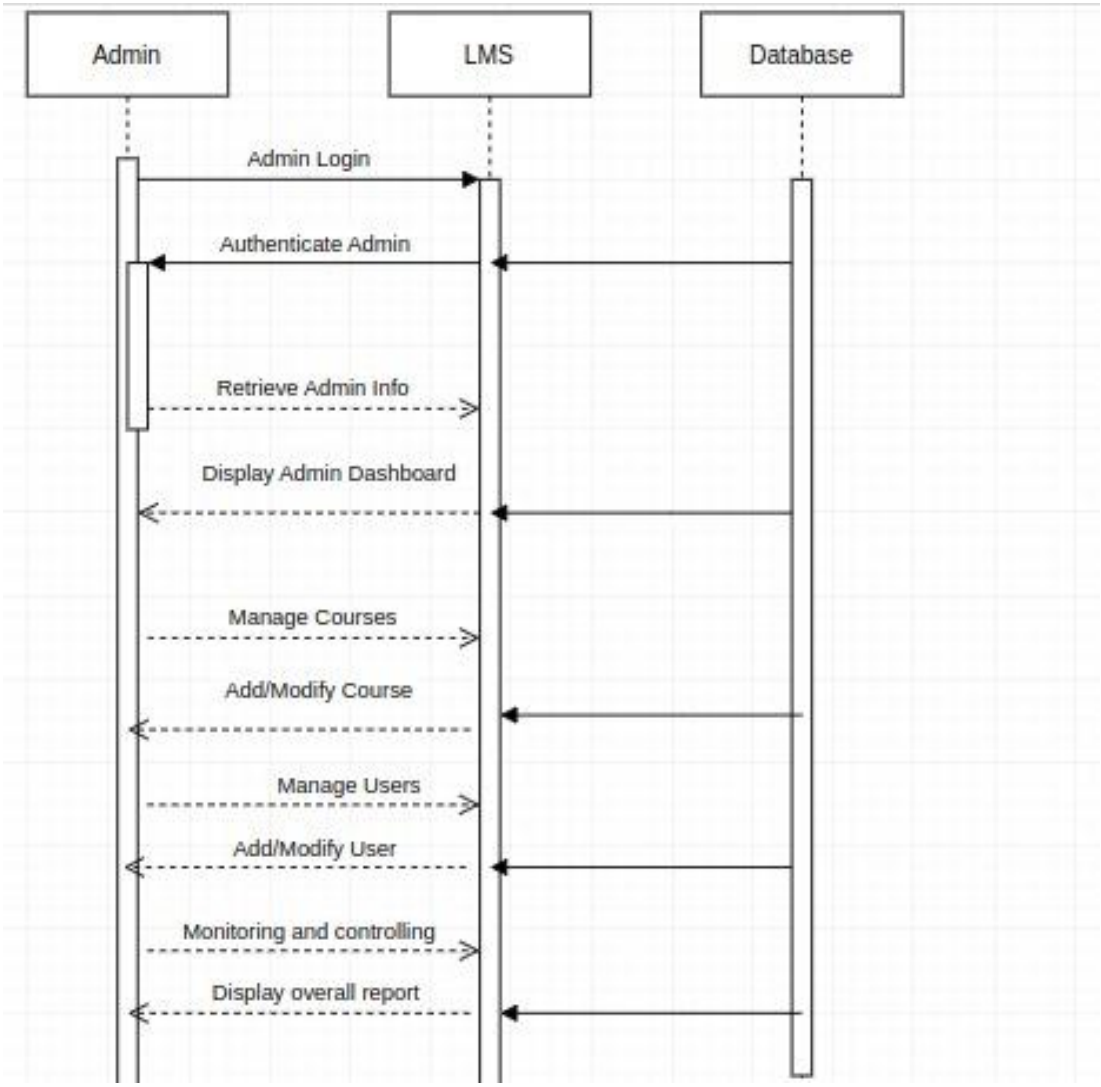


Figure5.4:Administrator sequence diagram

\5.2.2.2: Teacher Sequence Diagram:

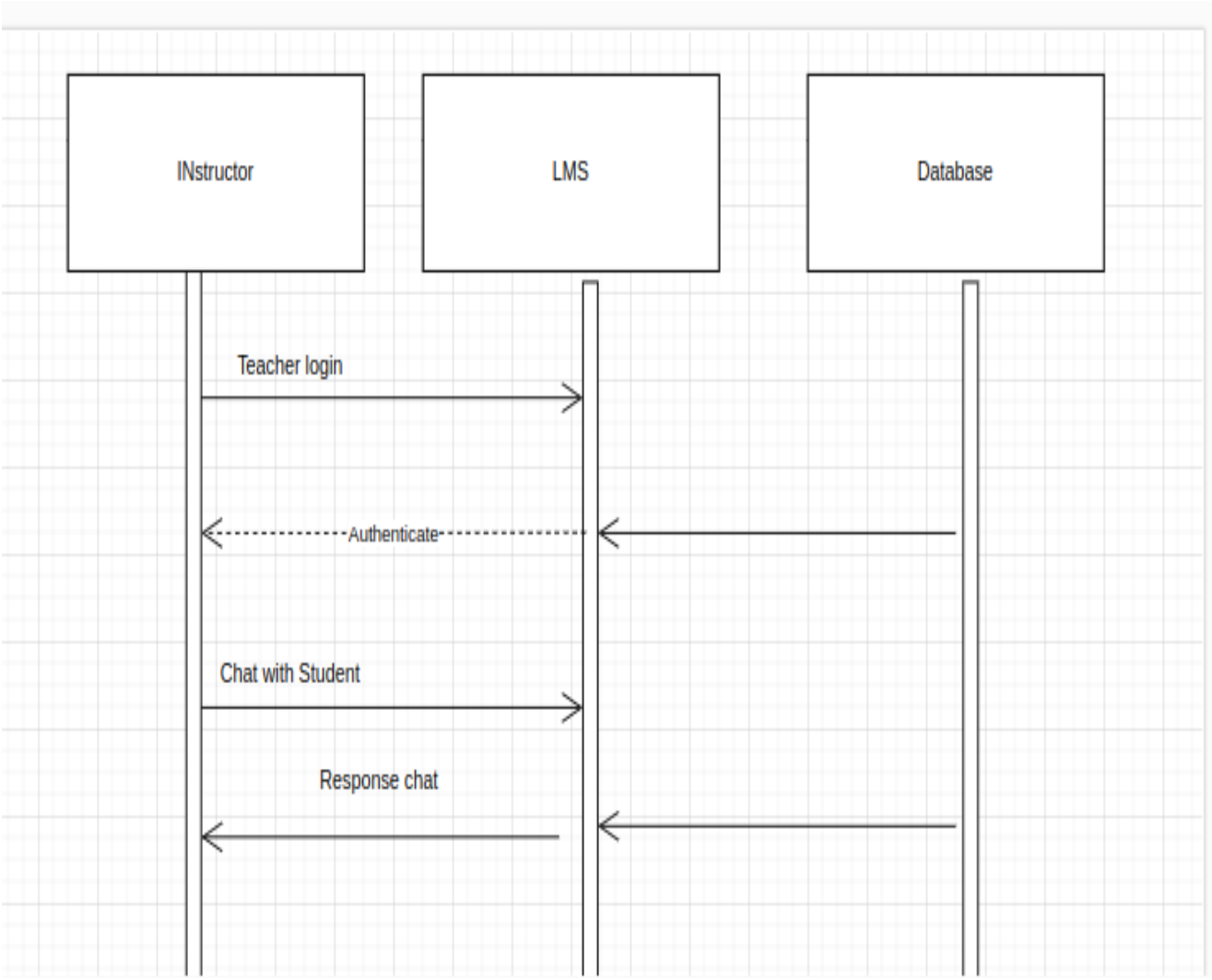


Figure5.5:Teacher sequence diagram

5.2.: Student Sequence Diagram:

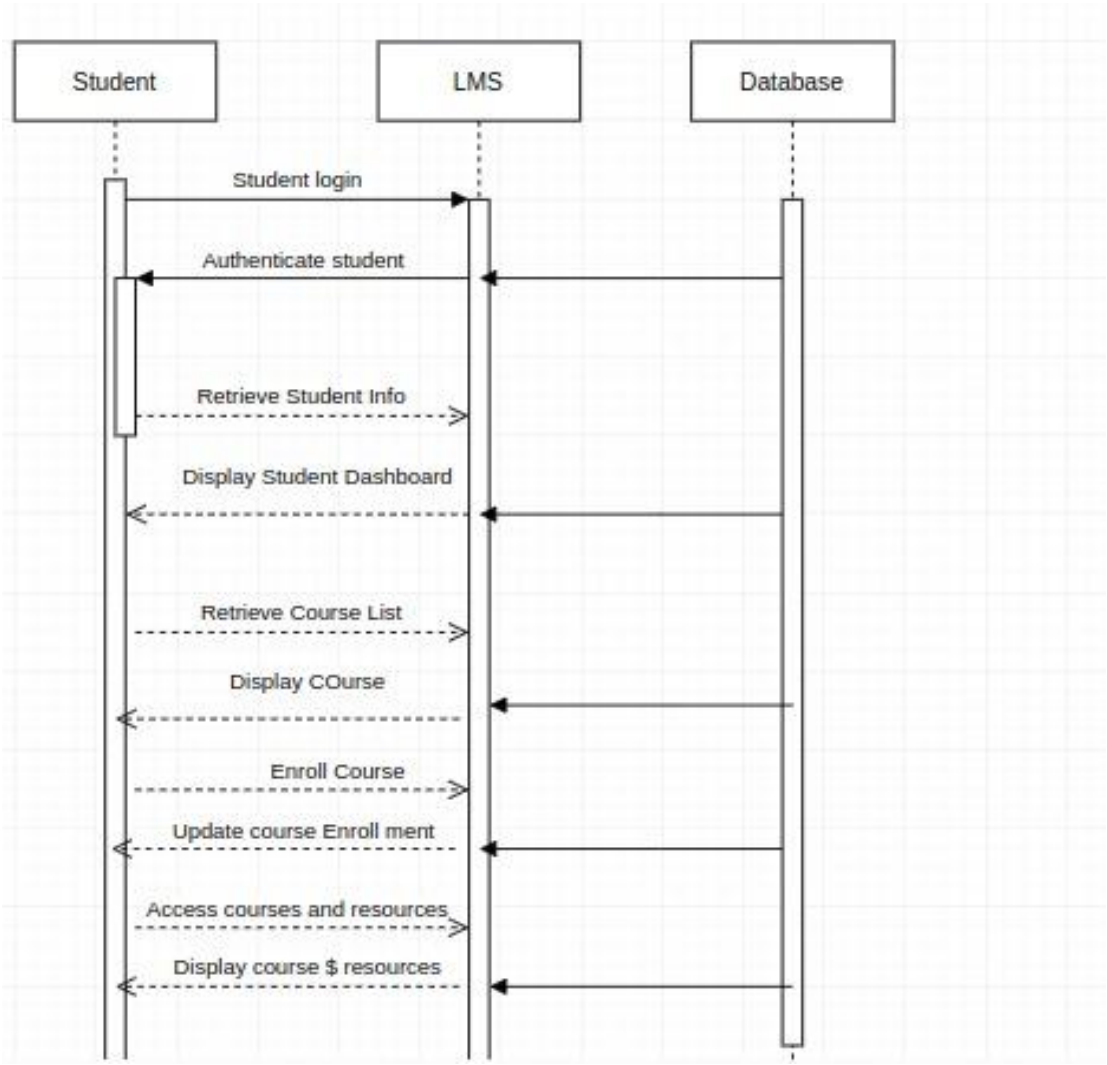


Figure5.6:Student sequence diagram

5.2.3: Class Diagram:

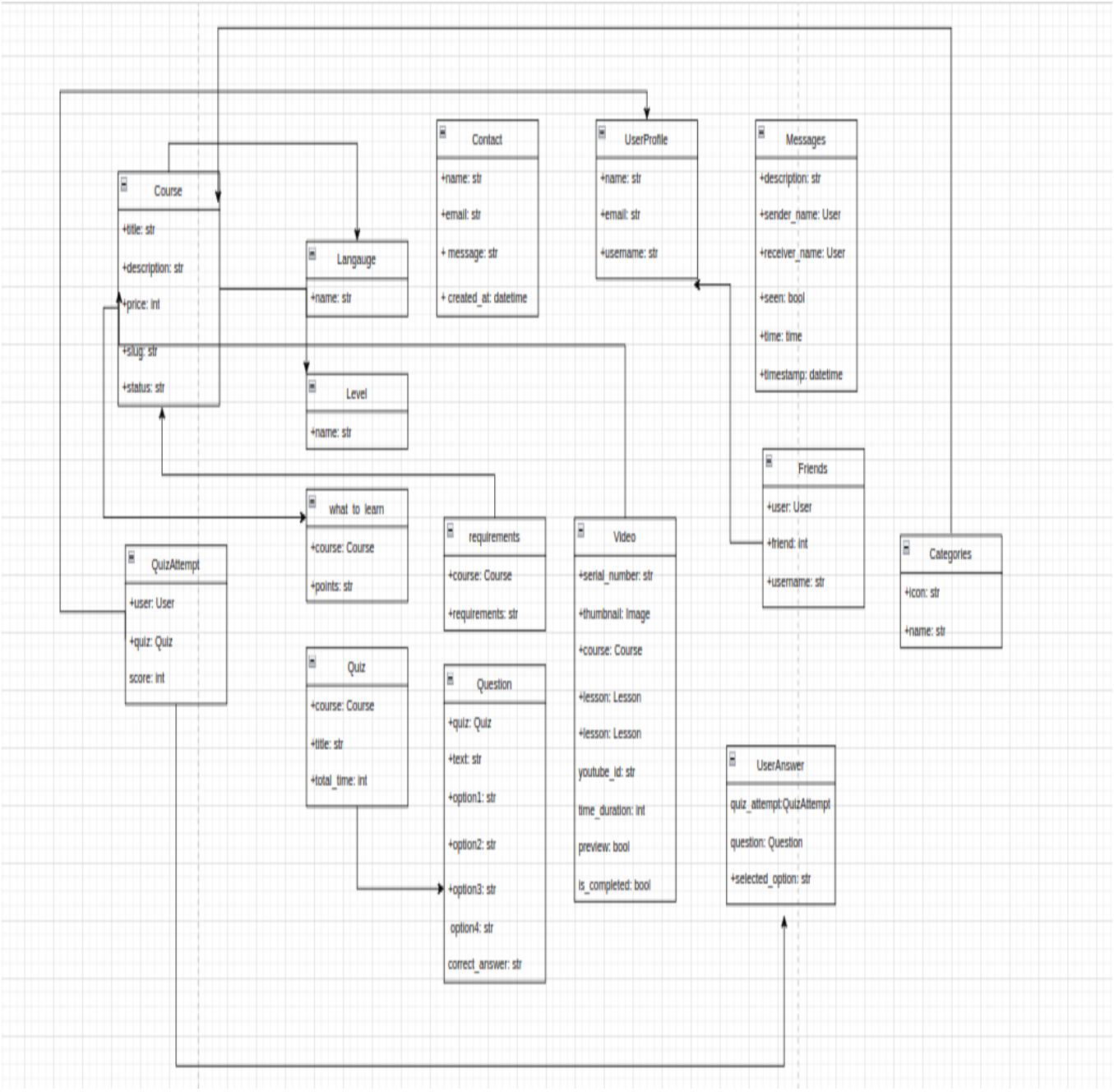


Figure 5.7: Class Diagram

5.2.4: ERD:

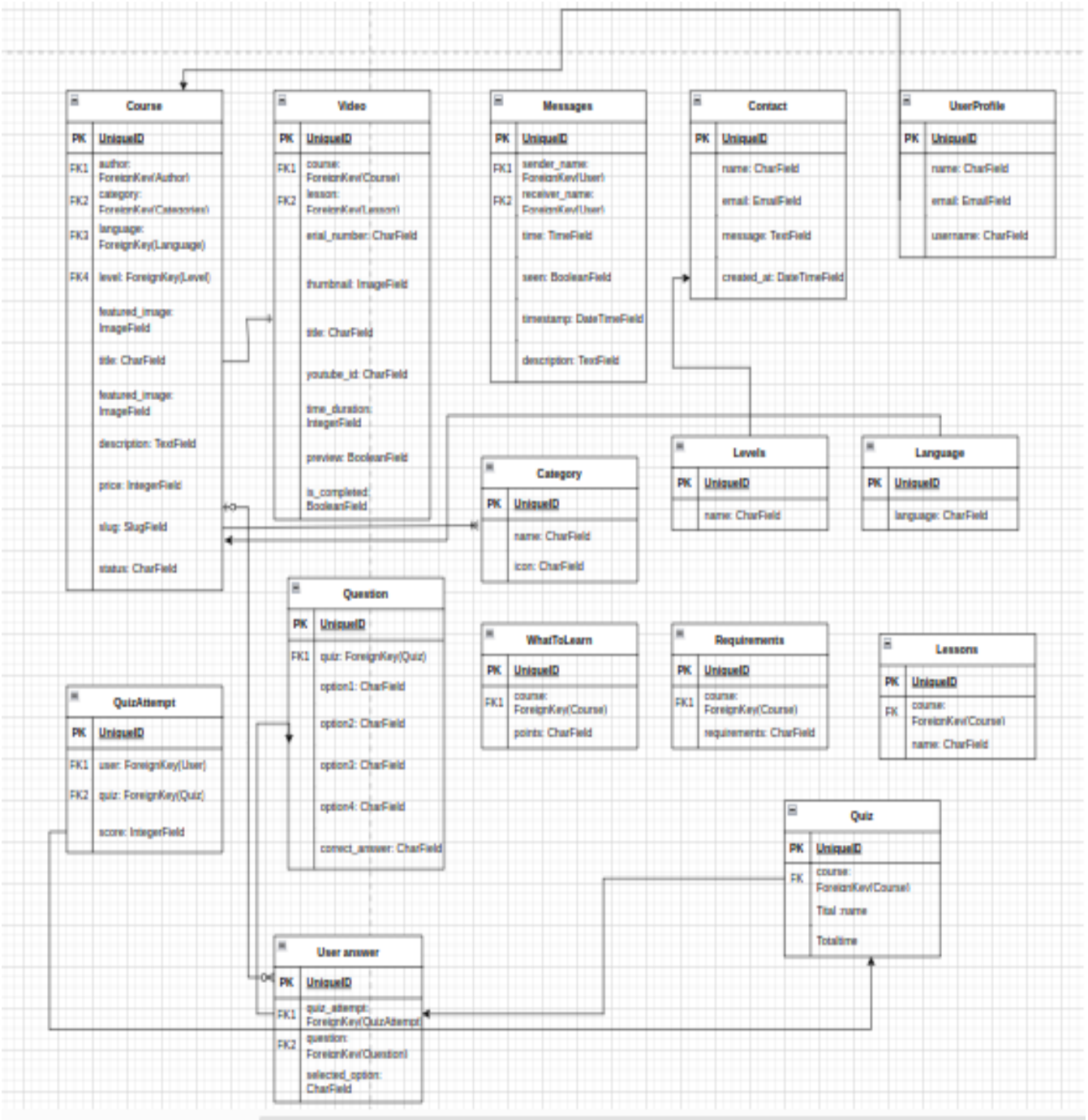


Figure5.8:ERD

Chapter 6: Prototype

6.1: Home Page:

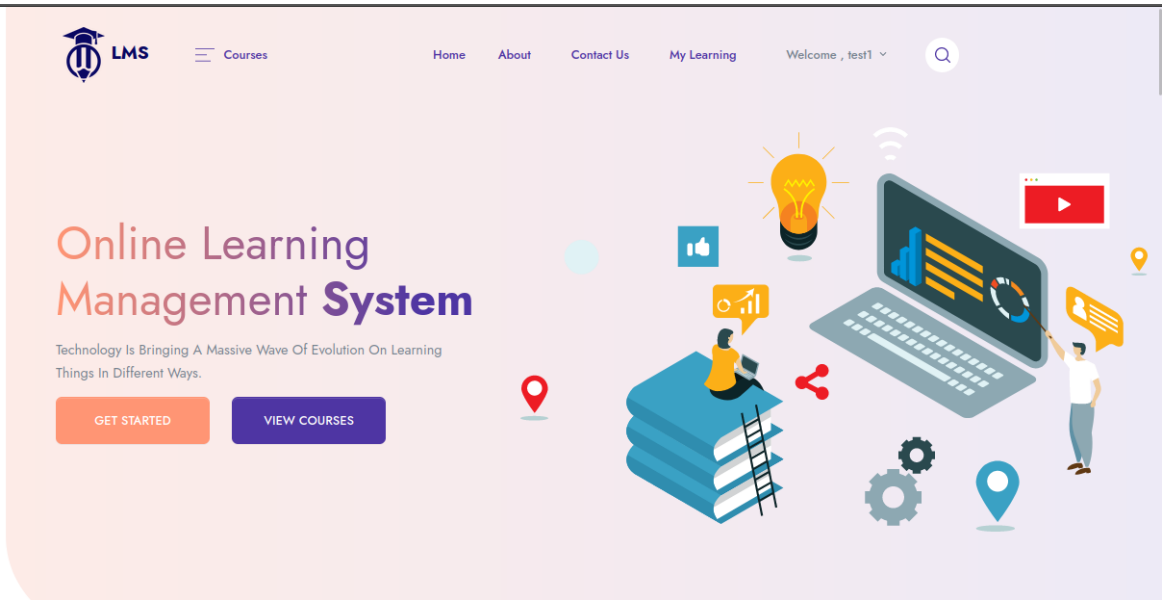


Figure 6.1:Home Page

6.2: Sign Up Page:

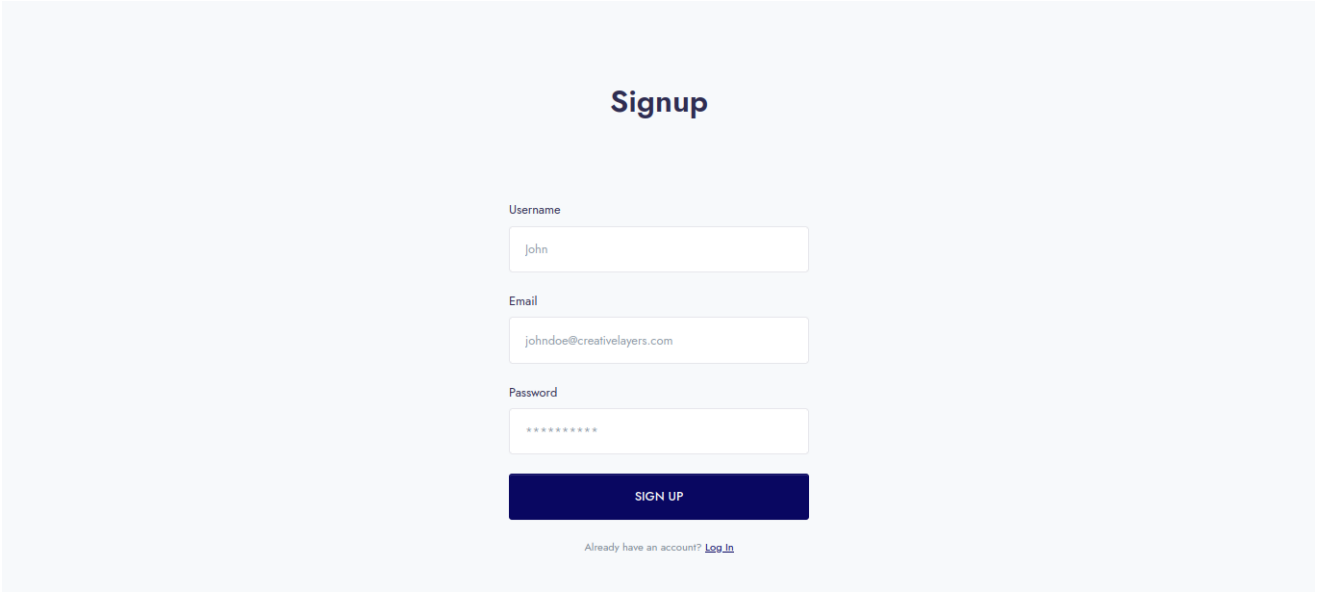


Figure 6.2:Sign Up Page

6.3: Login Page:

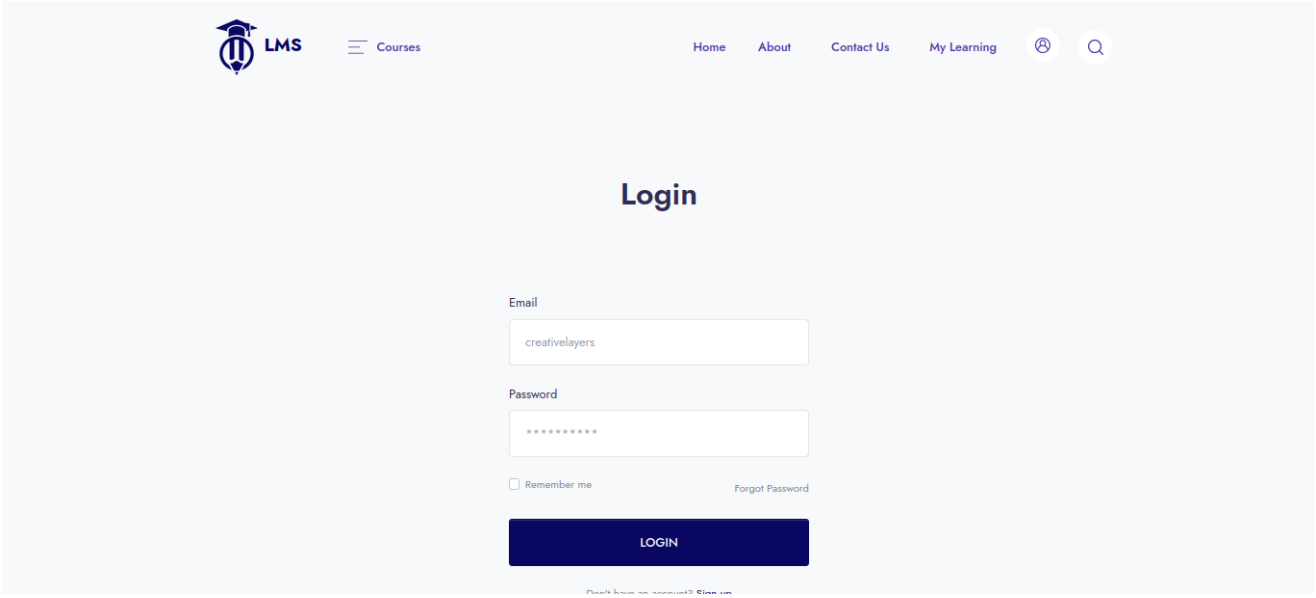


Figure 6.3:Login Page

6.4: Forget Password:

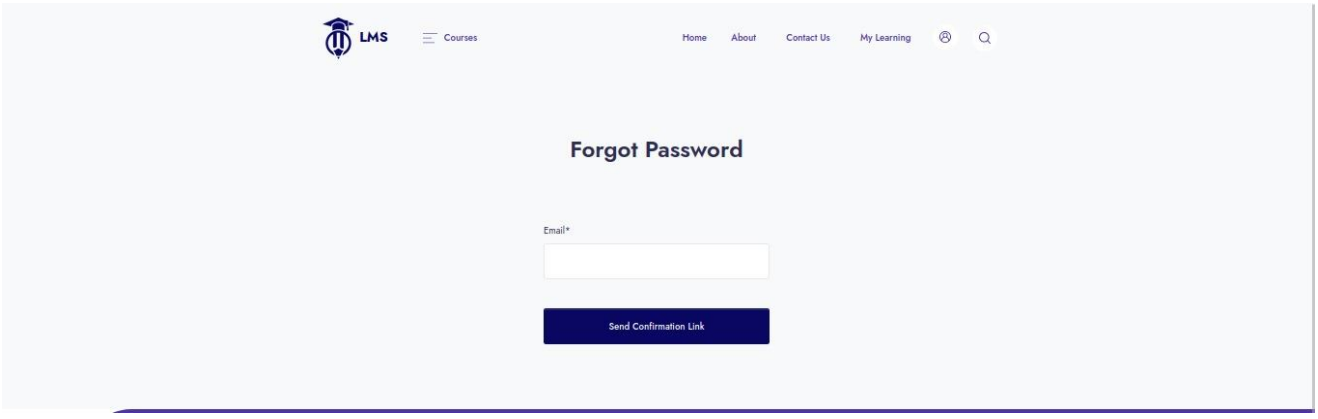


Figure 2:Forget Password

6.5: Reset Password:

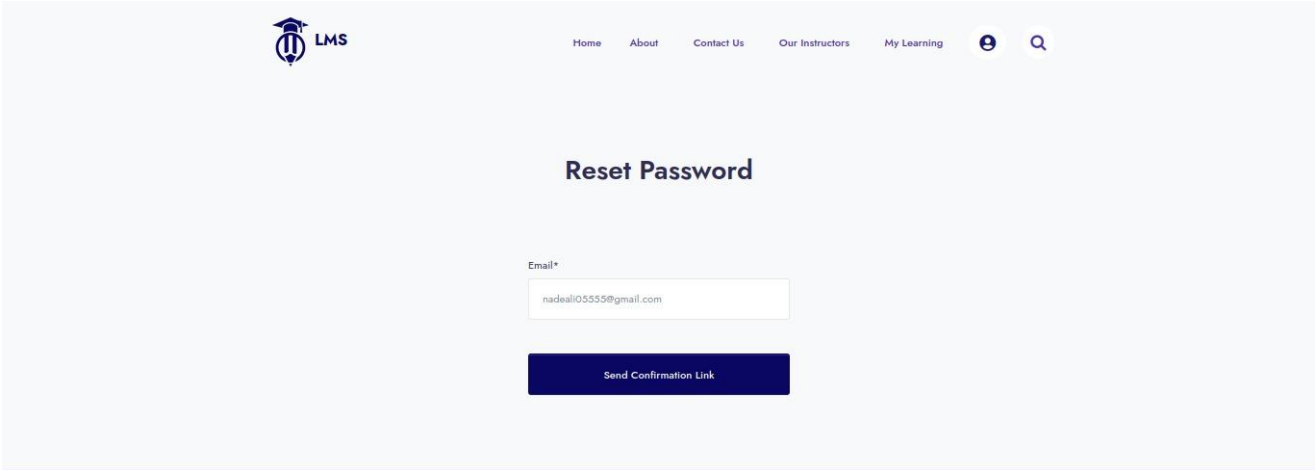


Figure 3:Reset Password

6.6: Contact:

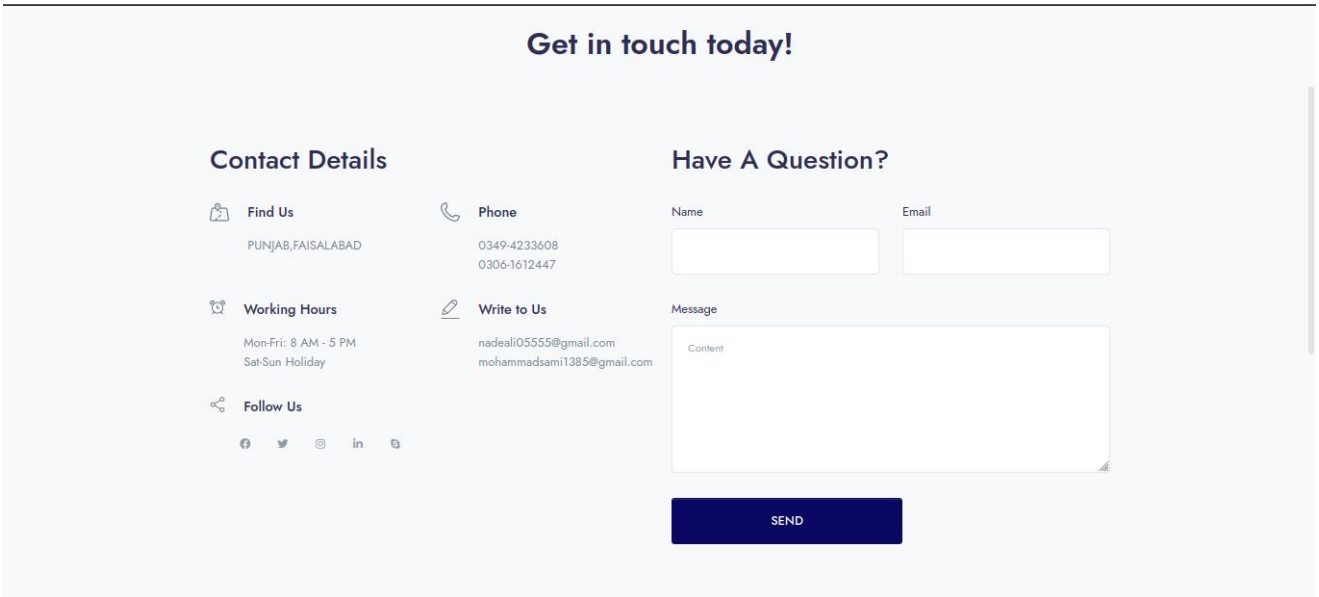


Figure 4:Contact

6.7: View Course:

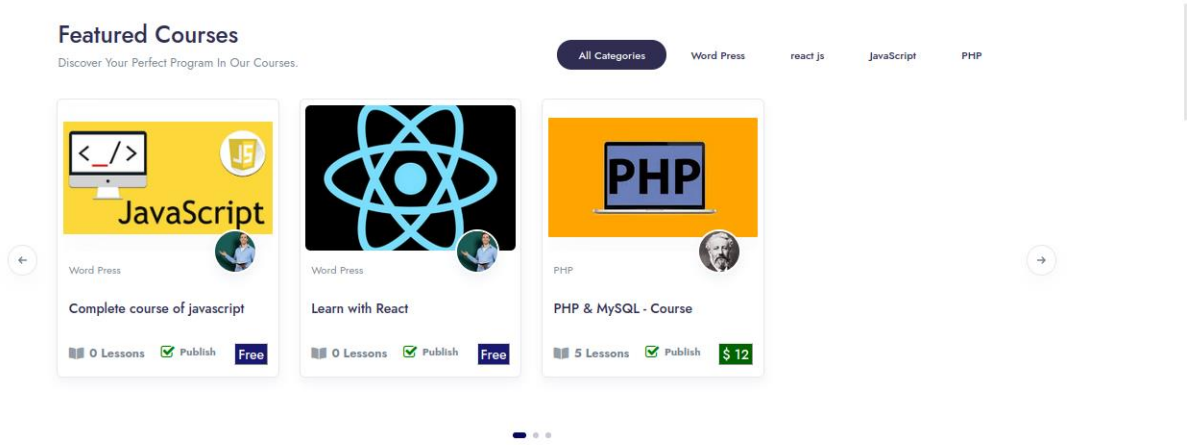


Figure 5: View Course

6.8: Single Course Detail:

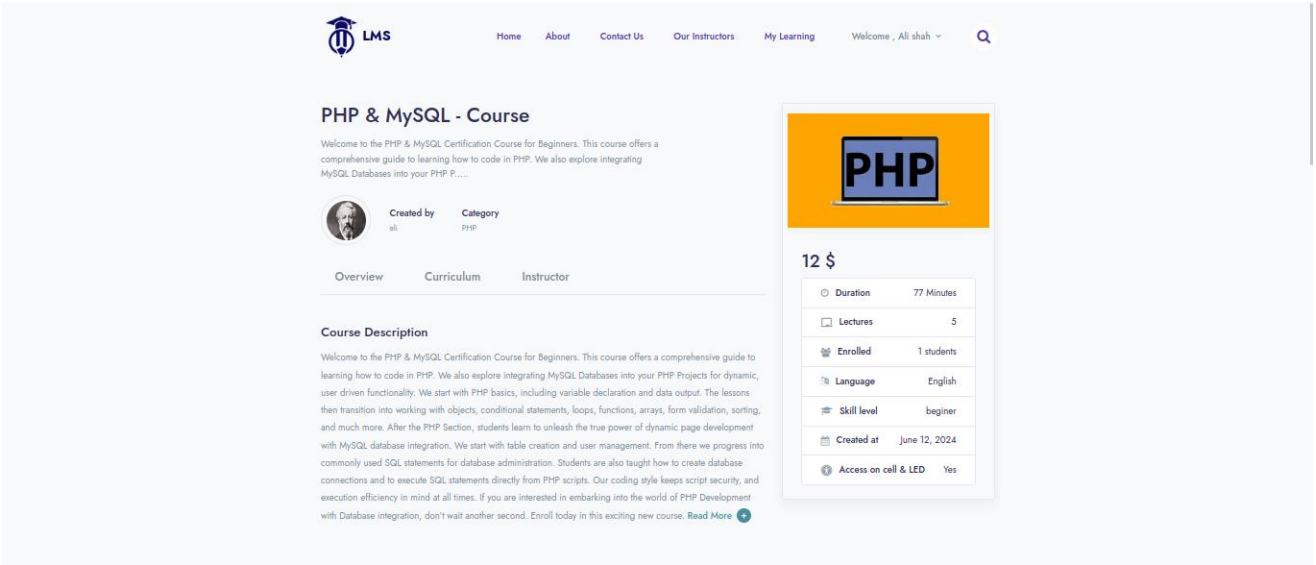


Figure 6:Single Course Detail

6.9:Quiz:

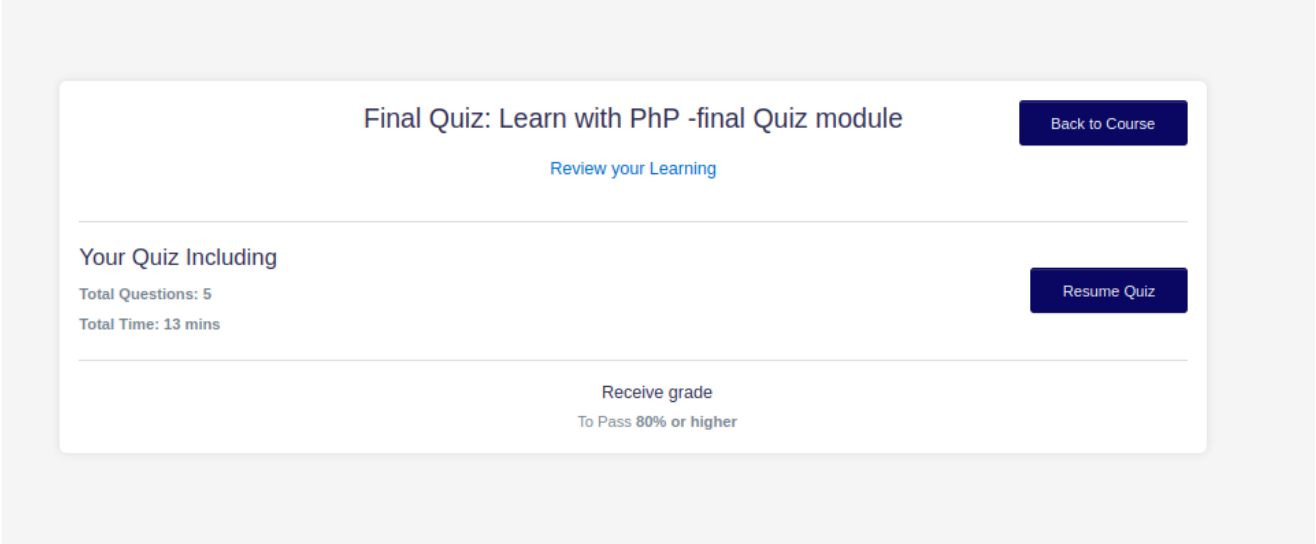


Figure 7 :Quiz

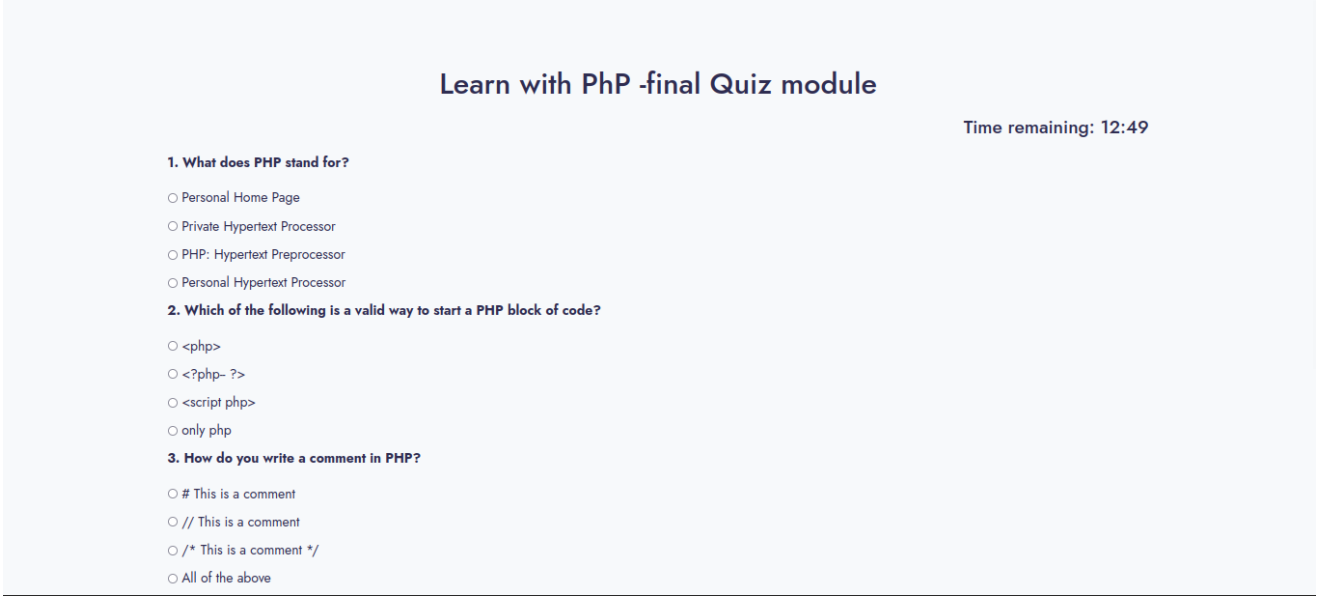


Figure6.10:Quiz Page1

Figure 6.11:Quiz Page 2

6.10: Quiz Result:

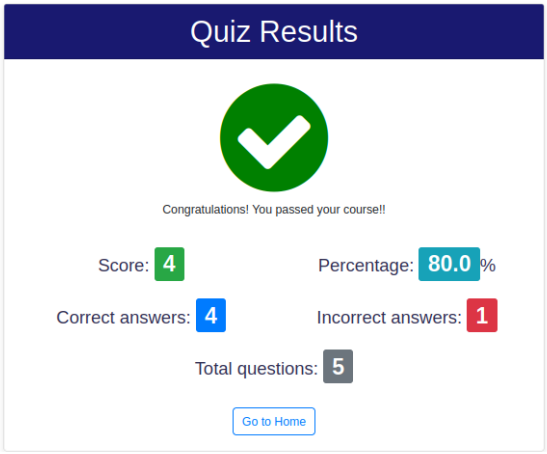


Figure 6.12: Quiz Result

6.11: User chat with instructor:

Figure6.13: User chat with instructor

6.12: User course enrolled:

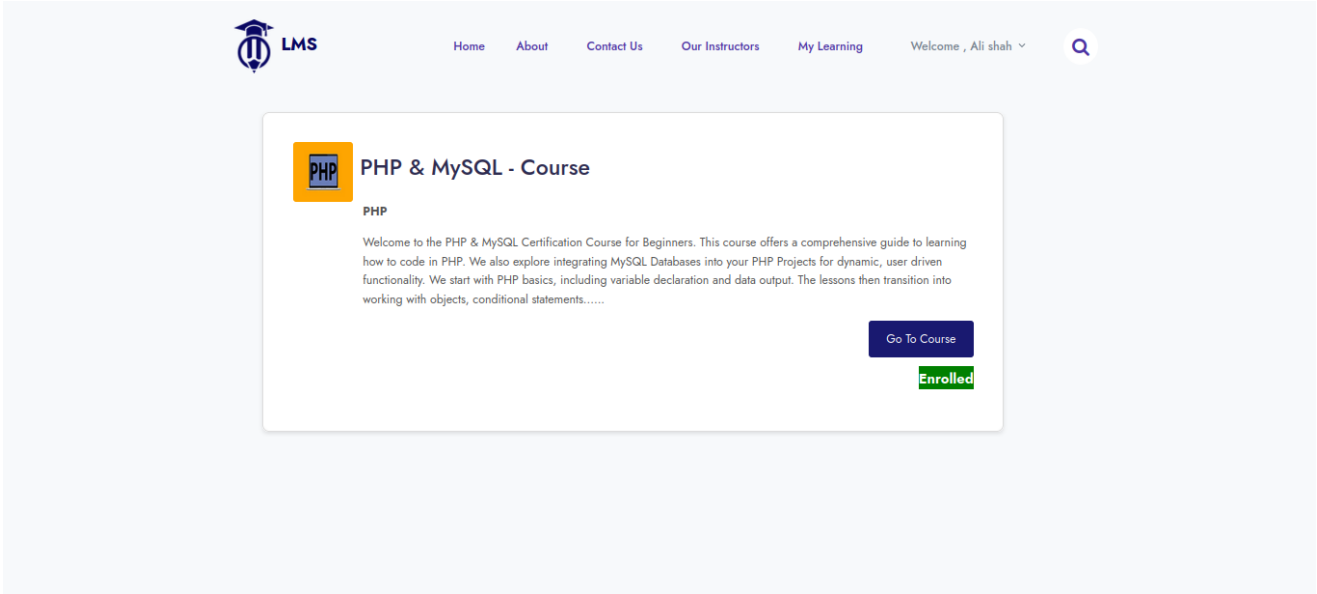


Figure6.14: User course enrolled

6.13: Watch course page:

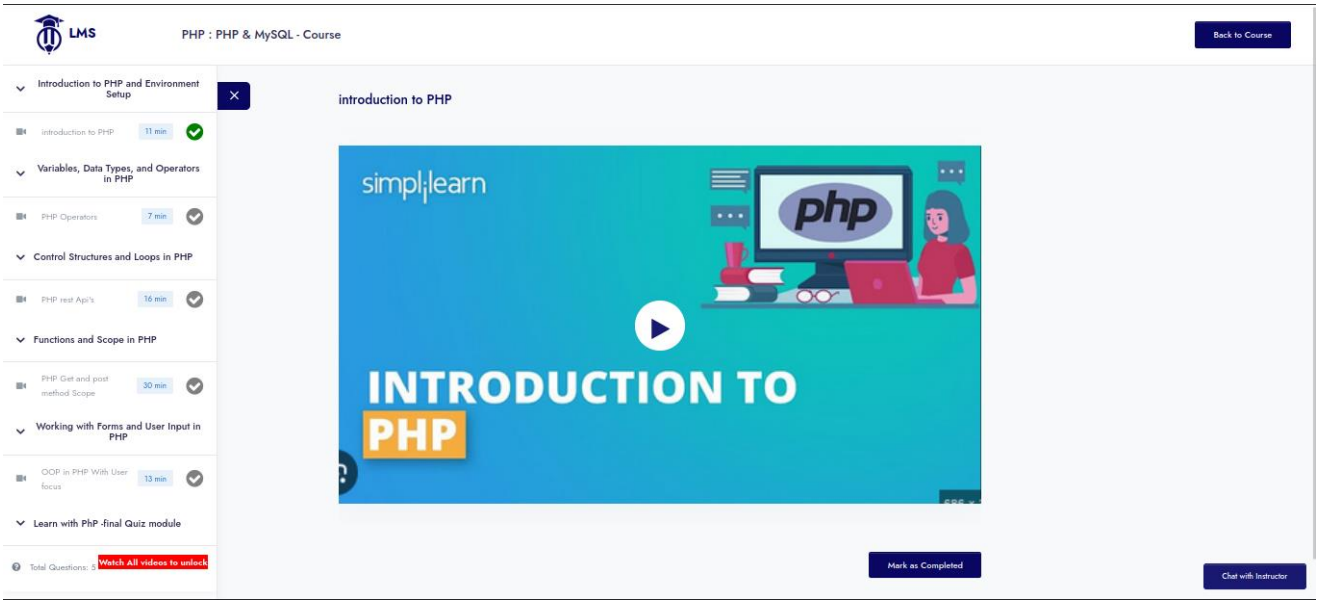


Figure6.15: Watch course page

6.14:Search courses:

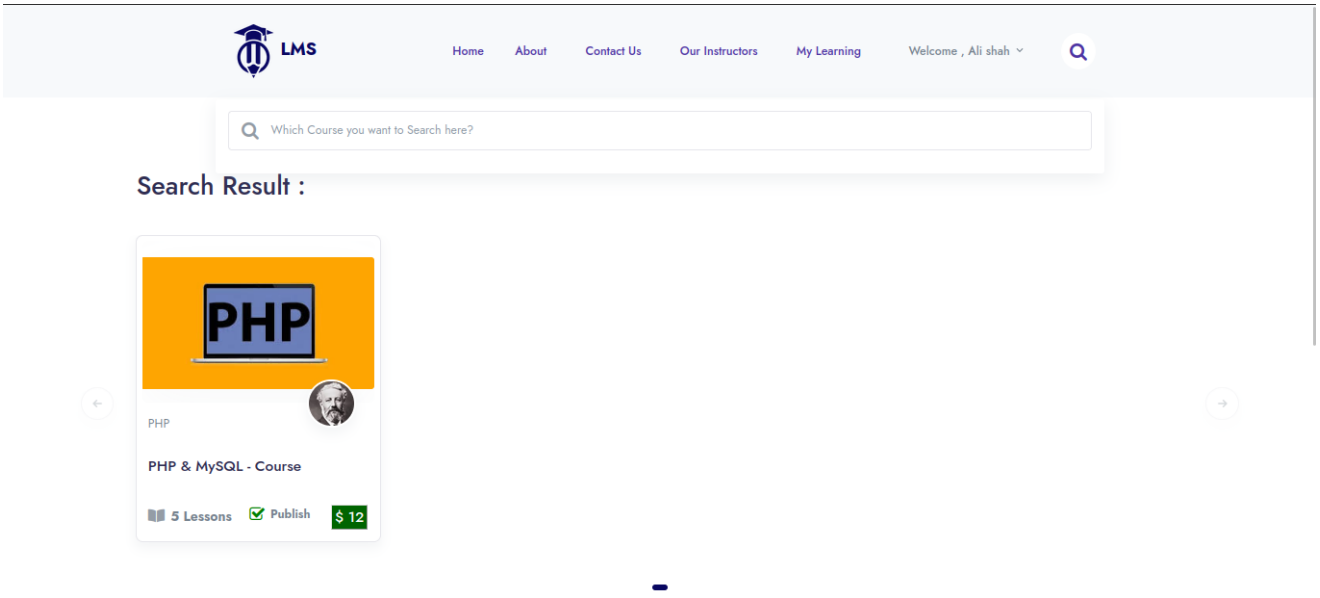


Figure6.16: Search courses

6.15: Our Instructors:

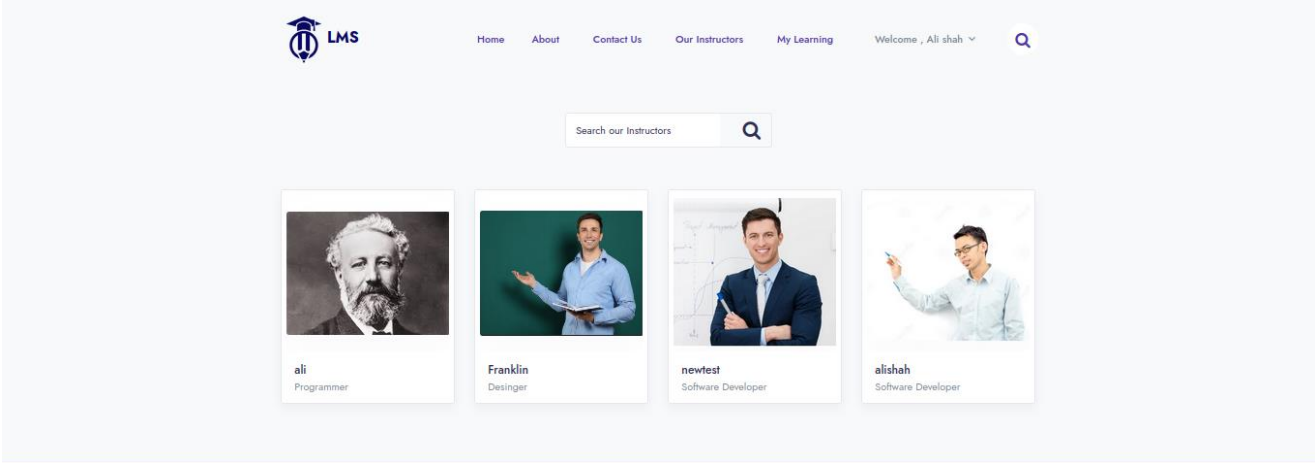


Figure6.17: Our Instructor

6.16: Search Instructors:

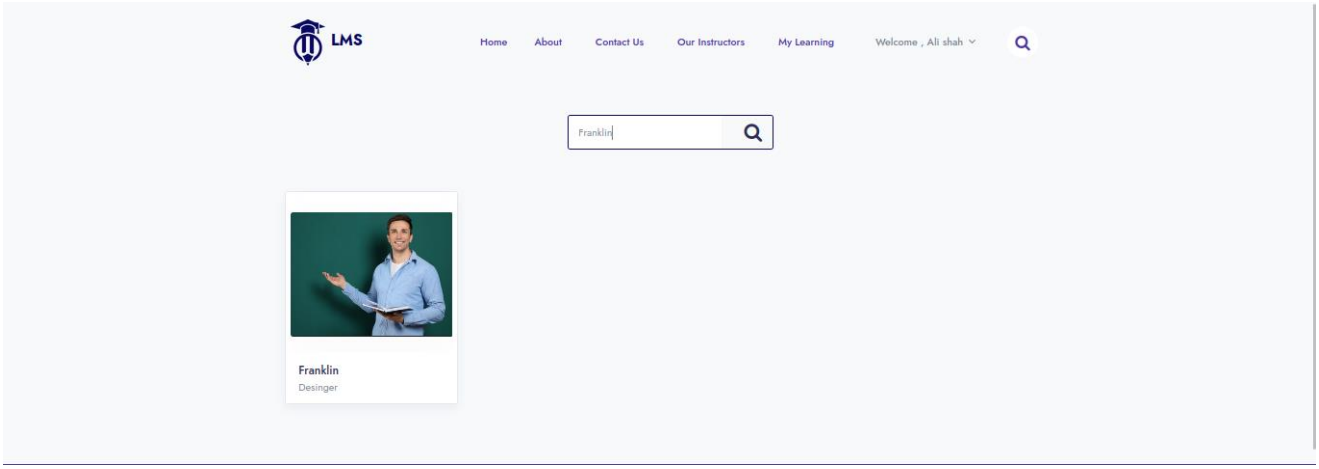


Figure6.18: Search Instructors

6.17: Admin login:

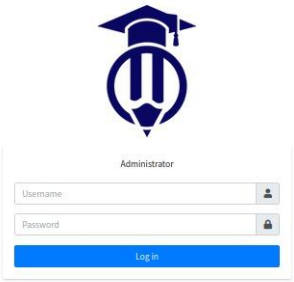


Figure6.19:Admin Login

6.18: Admin Home:

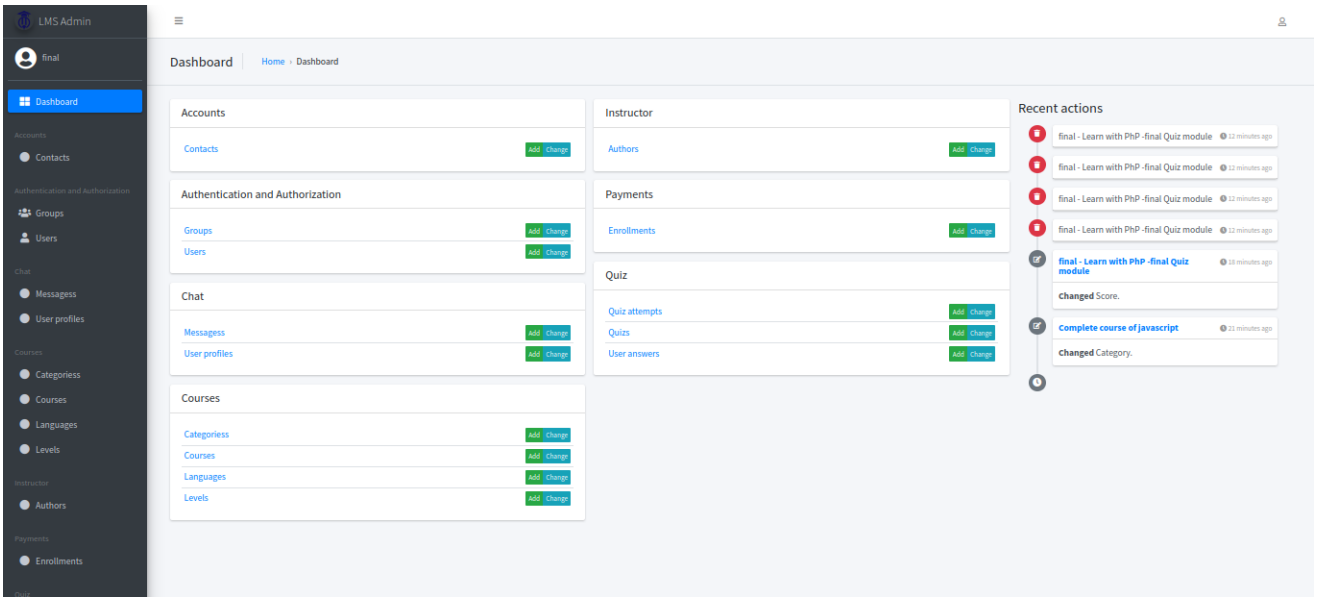


Figure6.20:Admin Home

6.19: Admin Course Creation:

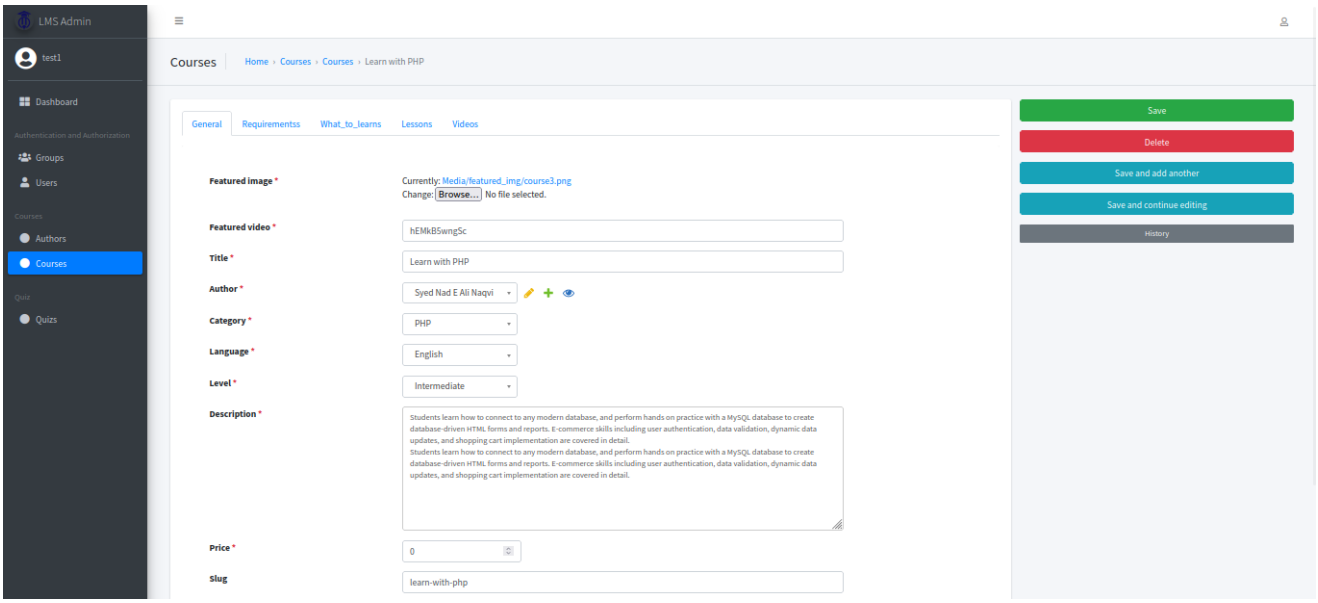


Figure6.21: Admin Course Creation

6.20: Admin Quiz Creation:

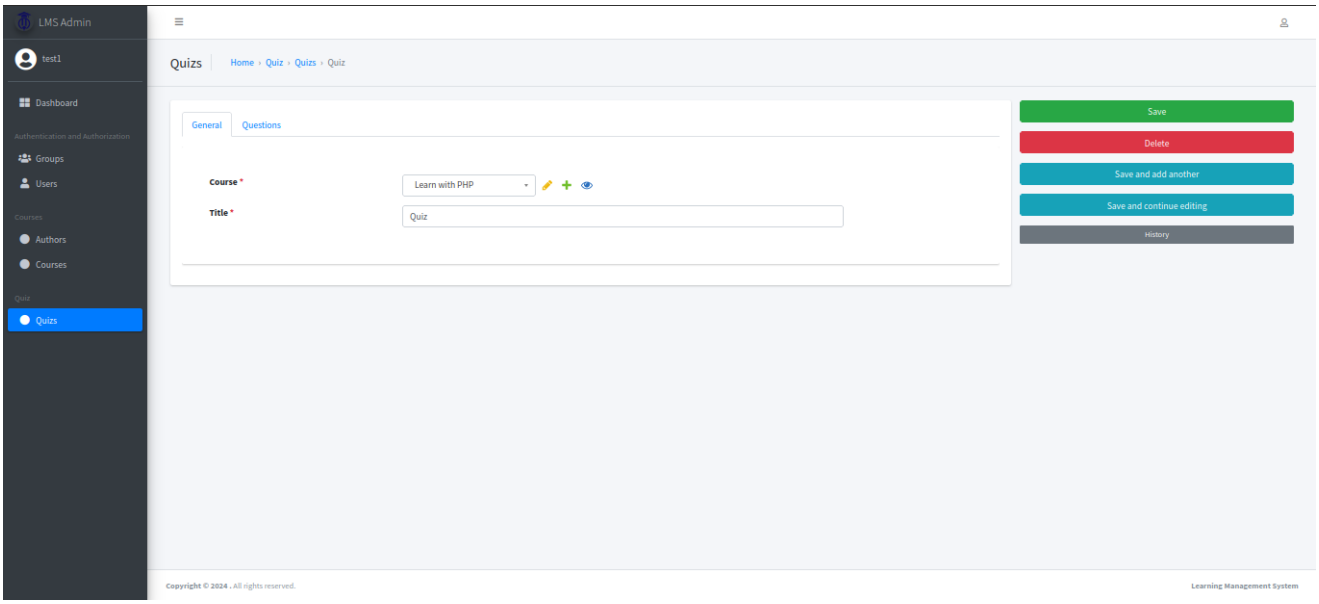


Figure6.22: Admin Quiz Creation

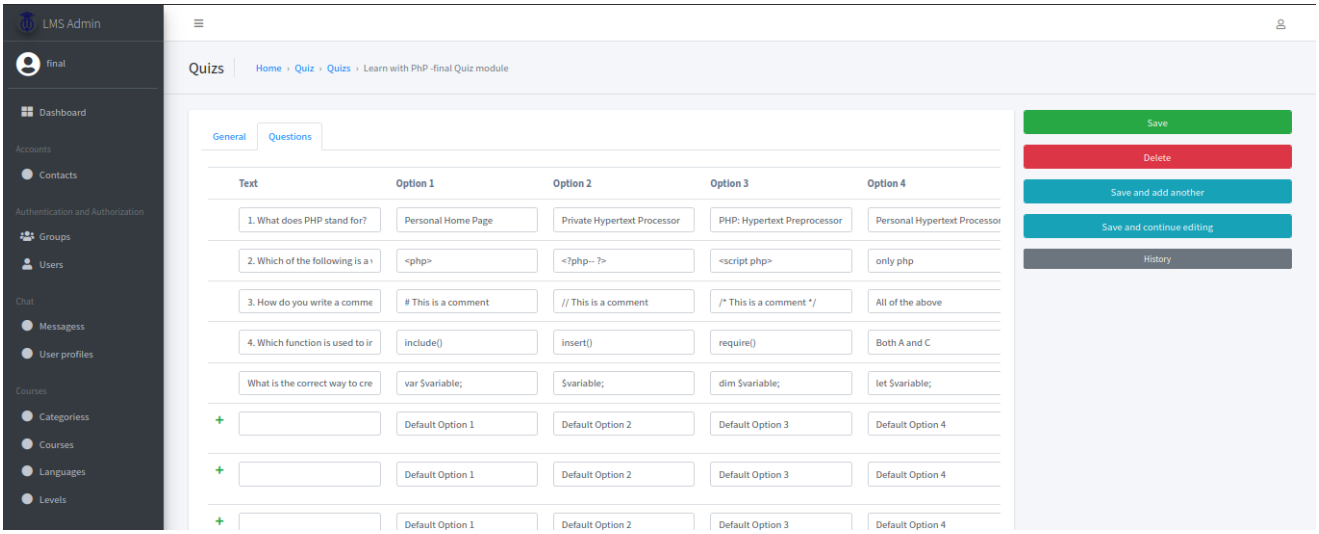


Figure6.23: Quiz Question Added by admin

Chapter 7: Implementation and Testing

Software testing is performed to verify that the software system does what it is supposed to do and to ensure that the software system is defect free by using finite test cases. Each test case is selected by the software tester. Test case is a set of actions that are executed to verify a particular functionality of the software system under different circumstances. They can apply to any software system, can use manual testing, can use automated testing and can also use test case management tool. Software testing is performed to identify errors, bugs and any missing requirement in the system in contrast to actual requirement.

7.1 White Box Testing:

White box testing or glass box testing is an approach that allows tester to verify the internal structure of the software like coding and integration with internal system. Basic goal of white box testing is to verify the working flow for a software system rather than just like functionality in the black box testing.

As white box testing has advantage like it cover almost the whole system and provide coverage for testing. But also has disadvantages like it is very expensive, very time-consuming and require professional resources. To perform white box testing testers must have code or programming knowledge. Following are the white box testing type:

- Unit testing
- Testing for memory leaks
- White box penetration testing
- White box mutation testing

7.2 Black box Testing:

In black box testing no prior knowledge of internal system is needed during testing. As it is based on user perspective, so concerned with input and output of the program. Testers give input than evaluate the output of the system.

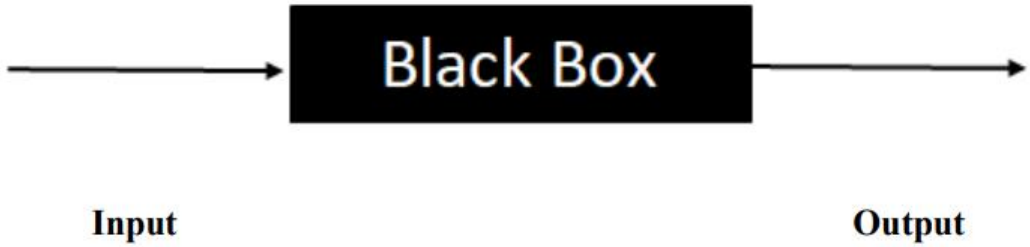


Figure 7.1:Black box testing

Input Output Black box testing is also called behavioral testing. Test cases used for black box testing can be functional and non-functional. But mostly use functional test case.

- Functional testing
- Non-functional
- Regression testing

Following are the other type of testing:

- Scenario based Testing.
- Equivalence Based Testing
- Boundary valued Testing
- Accessibility Testing
- State Transition Testing

7.3: Verification

Verification is a process in which we check about the product meet all the system specifications. And We built the right product? We get all the answer about these questions in the verification

process. This is an often-internal process. Four methods we are going to implement on our product for the verification process are as follows:

- Inspection
- Demonstration
- Test
- Analysis

7.4: Validation

Validation is a process in which we get analysis of the data gathered throughout the design and manufacturing of the product to confirm that each process will produce a product with standard quality. We will perform validation process on our product to confirm that the product meet user expectations and concern about we built the right product.

7.5: Adopted Methodology

7.5.1: Unit testing

Form the start, we test the whole system in term of testing the small units of system with respect to its intended functionality. Unit testing is done during the development phase. It is regarded in term of system Verification. After unit testing, we move to next level of testing.

7.5.2: Module Testing

After unit testing, we move to next testing that is called module testing where we test each test object independently as a module without integrating with other objects like class, module and program. We can test login and logout module at the same time because they are dependent on each other. After completing each module, we test it to avoid any further problems. After module testing, we move to next testing.

7.5.3: Integration testing

Integration testing is a phase in which we collect all the individual module of the system as a group then perform testing. Integration testing help in catching system issues such as broken database schema and mistaken cache integration. It is regarded in term of system verification. After completing each module, we performed integration testing to check their compatibility with each other.

7.5.4: System testing

After completing all the sub-level testing, we perform system testing on the whole system at once. System testing is also called black box testing. System testing generally focuses on the system validation. We performed system testing on our system after completing all its modules to check that it meets all its requirement.

7.5.5: Acceptance testing

Acceptance testing is the most important phase of testing in which the system is validated by the stakeholders. They ensure that the system is according to their expectations or requirements. After successfully passing the acceptance testing, the system is deployed after a short time.

References

In the world of online education and professional training, Learning Management Systems (LMS) have become vital tools. They facilitate seamless interaction between instructors, administrators, and learners, acting as a centralized hub for delivering content, conducting assessments, and fostering collaboration. Here are two well-known LMS platforms similar to Coursera and Udemy:

Moodle: Moodle is celebrated for its adaptability and extensive customization options. As an open-source platform, it empowers educators to develop dynamic and engaging online courses. With a comprehensive set of tools, Moodle helps create a secure and integrated learning environment that meets the diverse needs of both teachers and students.

Canvas: Canvas is praised for its intuitive interface and robust features. This cloud-based LMS simplifies the management of courses and enhances the learning experience with user-friendly tools and workflows. It enables educators to provide high-quality online education effortlessly, making it a favorite among many institutions and organizations.

Additional Reference:

[1] Ellis, Ryann K. (2024), "Field Guide to Learning Management", sourced from Quora, archived on 24 August 2014, retrieved on 5 July 2012.

[2] Davis, B., Carmean, C., & Wagner, E. (2009). "The Evolution of the LMS : From Management to Learning". The ELearning Guild Research. 24.

[3] Reports, Valuates. "Learning Management System (LMS) Market to Grow USD 40360 Million by 2028 at a CAGR of 17.1% | Valuates Reports".

[4] "LMS Data – The First Year Update". Edutechnica. 23 September 2014. Retrieved 1 February 2015.

[5] Reports, Valuates. "Learning Management System (LMS) Market to Grow USD 40360 Million by 2028 at a CAGR of 17.1% | Valuates Reports".

[6] Santos, Lara Cristina de Paiva Lourenço dos Santos (2008). Adaptability support in a learning management system (masterThesis).

[7] Connolly, P. J. (2001). A standard for success. InfoWorld, 23(42), 57-58. EDUCAUSE Evolving Technologies Committee (2003). Course Management Systems (CMS). Retrieved 25 April 2005, from http://www.educause.edu/ir/library/pdf/DEC0302.pdf

[8] Demian, Peter; Morrice, James (15 December 2015). "The Use of Virtual Learning Environments and Their Impact on Academic Performance". Engineering Education. 7: 11–19. doi:10.11120/ened.2012.07010011.

[9] Wiley, D. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), The instructional use of learning objects: Online version. Retrieved 13 March 2005, from http://reusability.org/read/chapters/wiley.doc.

[10] A field guide to learning management systems. (2005). Retrieved 12 November 2006, from http://www.learningcircuits.org/NR/rdonlyres/BFEC9F41-66C2-42EFBE9D-E4FA0D3CE1CE/7304/LMS_fieldguide1.pdf.