



Design, Implementation, and Functional Evaluation of a Role-Based Web-Based Learning Management System

Ismail Z. Y¹, Auwal M. Abdullahi², Zainab Aliyu Musa³

Department of Computing Technologies, SRM Institute of Science and Technology, Kattankulathur, 603203, Tamil Nadu, India

Department of Computer Science, Federal Polytechnic Bauchi

ABSTRACT :

Learning management systems have become central to teaching, assessment, and academic administration in tertiary institutions, yet many Nigerian polytechnics still depend on manual or partially digitised processes for course delivery, testing, and record keeping. This study presents the design, implementation, and functional evaluation of a role-based web-based Learning Management System (LMS) developed for Federal Polytechnic Bauchi (FPTB). The system was built with PHP, MySQL, and Tailwind CSS, and integrates four operational roles, namely administrator, coordinator, lecturer, and student, within a single relational database that models departments, academic sections, course assignments, learning materials, computer-based quizzes, and assignment submissions. An iterative waterfall approach guided development, comprising requirements gathering, system and database design, implementation, and testing. Security was addressed through session-based authentication, bcrypt password hashing, parameterised database queries, and role-based access control that restricts each user category to its designated modules. A computer-based testing engine with randomised question ordering was incorporated for quizzes, alongside a virtual classroom module built on the Jitsi Meet application programming interface for live video-based instruction. The system was evaluated using black-box functional testing across its core modules, including registration and approval, course assignment, material upload, quiz administration, assignment grading, and virtual class access. Findings indicate that the system performed its intended functions correctly, with clearly separated role privileges and consistent grade computation. The study concludes that a lightweight, self-hosted LMS combining computer-based testing and video conferencing is a feasible and cost-effective option for Nigerian polytechnics operating under infrastructural and budgetary constraints, and recommends further usability evaluation with actual staff and students before full deployment.

Keywords: learning management system; web-based system; role-based access control; computer-based testing; Nigerian polytechnic; PHP; MySQL

1. Introduction

1.1 Background of the Study

The delivery of instruction in tertiary institutions has shifted steadily from purely physical classroom interaction toward web-based platforms that support content distribution, assessment, and communication outside fixed lecture hours. This shift accelerated sharply during the COVID-19 pandemic, when Nigerian universities and polytechnics were compelled to adopt e-learning tools with little preparation time. Ogunji et al. (2022) found that a large proportion of tertiary institutions in South Eastern Nigeria were not adequately prepared for this transition, and that informal platforms such as WhatsApp were relied upon more heavily than structured e-learning packages. Oyediran et al. (2020) similarly reported that while e-learning carried clear prospects for continuity of academic activity during the lockdown, its application in private tertiary institutions was constrained by epileptic power supply, high data costs, and limited staff competence in digital tools. More recent evidence suggests these constraints have persisted into the post-pandemic period rather than resolving with time; Igbinovia and Ottah (2024) found that undergraduate students in Nigerian universities continue to report poor internet connectivity, limited access to personal computing devices, and insufficient technical support as recurring barriers to effective e-learning.

Polytechnics occupy a particular position within this landscape. Unlike well-funded federal universities, many polytechnics operate with tighter ICT budgets, and course delivery structures that differ from university faculties, being organised instead around departments, coordinators, and cohort-based sections tied to specific admission sessions. Commercial or open-source learning platforms such as Moodle or Google Classroom, while functionally capable, are not always configured to reflect this administrative hierarchy out of the box, and may carry hosting, bandwidth, or licensing costs that are difficult to justify for a single institution with limited internally generated revenue. This creates a practical case for a purpose-built, lightweight LMS that mirrors the exact academic structure of the institution it serves, namely departments, sections by admission year and level, semester-bound course assignments to specific lecturers, and role-differentiated dashboards for administrators, coordinators, lecturers, and students.

1.2 Statement of the Problem

At Federal Polytechnic Bauchi, as in many similar institutions, course materials are still largely distributed through printed handouts, flash drives, or informal social media groups, examinations are conducted predominantly on paper, and the assignment of courses to lecturers by departmental coordinators is tracked manually or through spreadsheets maintained outside any central system. This arrangement makes it difficult to maintain a

consistent institutional record of who taught what course, in which section, and in which semester, and it offers no built-in mechanism for computer-based assessment, remote lecture delivery, or centralised grading. Where third-party e-learning tools have been adopted piecemeal, such as video conferencing applications used in isolation from course records, the resulting workflow remains fragmented, with materials, grades, and communication scattered across different platforms rather than consolidated within one system that reflects the institution's own departmental and coordination structure.

1.3 Aim and Objectives of the Study

The aim of this study is to design, implement, and evaluate the functionality of a role-based web-based Learning Management System tailored to the departmental and coordination structure of Federal Polytechnic Bauchi. The specific objectives are to:

- i. review existing literature and related systems on e-learning adoption, computer-based testing, video conferencing, and web application security relevant to tertiary education in Nigeria;
- ii. design a relational database that models departments, academic sections, course assignments, learning materials, quizzes, and assignment submissions;
- iii. implement a web-based system with four role-differentiated modules, namely administrator, coordinator, lecturer, and student, secured through role-based access control;
- iv. incorporate a computer-based testing engine and a virtual classroom module into the system; and
- v. evaluate the functionality of the implemented system using black-box testing across its core modules.

1.4 Significance of the Study

For the institution, the system offers a single point of reference for course assignment, material distribution, and assessment records, reducing dependence on informal channels. For lecturers, it consolidates material upload, quiz creation, and assignment grading within one interface. For students, it provides continuous access to course content, computer-based quizzes, and virtual class sessions regardless of physical attendance. For researchers and future developers, the system, and the account of its design decisions given in this paper, may serve as a reference case for building similarly structured LMS platforms for other Nigerian polytechnics facing comparable administrative and infrastructural constraints.

2. Literature Review

2.1 The Concept of Learning Management Systems in Web-Based Education

A learning management system is broadly understood as a web-based platform that supports the creation, delivery, and administration of instructional content and assessment outside the constraints of a single physical classroom. Such systems typically combine content repositories, communication tools, and assessment engines within role-differentiated interfaces so that administrators, instructors, and learners each interact with functionality appropriate to their responsibilities. Bailey et al. (2022) note that the effectiveness of any such platform depends heavily on how easily learners and instructors can use it and on whether the tool is perceived as genuinely useful for achieving learning outcomes, echoing the two central constructs of the Technology Acceptance Model, namely perceived ease of use and perceived usefulness. These constructs are relevant not only to video conferencing tools, which was the specific focus of that study, but to LMS design generally, since a system that is technically functional but difficult to navigate is unlikely to see sustained adoption by lecturers or students.

2.2 E-Learning Adoption and Challenges in Nigerian Tertiary Institutions

Several studies have examined the state of e-learning readiness in Nigerian tertiary institutions since the COVID-19 pandemic. Ogunji et al. (2022) surveyed academic staff and students across public and private tertiary institutions in South Eastern Nigeria and found that institutions were generally unprepared for a rapid shift to online delivery, with private institutions promoting more structured e-learning platforms than their public counterparts, and with informal tools such as WhatsApp used more widely than dedicated learning platforms. Oyediran et al. (2020) reported comparable findings among private tertiary institutions, identifying epileptic power supply, high internet subscription costs, and limited digital literacy among staff as recurring limitations, despite general recognition of e-learning's potential to sustain academic continuity during periods of disruption. More recently, Igbinoia and Ottah (2024) surveyed undergraduate students across several Nigerian universities in the post-pandemic period and found that connectivity and device access remained persistent obstacles, alongside gaps in students' technical preparedness, suggesting that the challenges identified during the pandemic have not been fully resolved by its passing. Taken together, this body of work indicates that any locally deployed LMS intended for a Nigerian institution should be designed with modest bandwidth and hosting requirements in mind, favouring a lightweight, self-hosted architecture over one that assumes continuous high-speed connectivity.

2.3 Computer-Based Testing in Nigerian Higher Education

Computer-based testing has been a recurring subject of study in the Nigerian computing literature, largely because it offers a practical means of reducing the delay associated with manual marking while curbing certain forms of examination malpractice. Ajinaja (2017) described the design of a computer-based testing system built with PHP, MySQL, JavaScript, and open-source web technologies, structured around index, administrator, lecturer, and student modules, and adopted a component-based software engineering approach so that individual subject modules could be reused across different examinations rather than rebuilt from scratch each time. This modular reasoning is consistent with the approach taken in the present study, where quiz functionality is attached generically to any course assignment rather than hard-coded to a specific subject. More recently, Zakariya et al.

(2026) conducted a phenomenological investigation into undergraduate students' perceptions of computer-based testing in Nigerian higher institutions through the lens of the Technology Acceptance Model, and found that students generally regarded computer-based testing as useful for faster grading and reduced administrative error, while technical failures, server crashes, and the rigidity of multiple-choice formats were cited as factors that undermined its perceived ease of use. This finding is directly relevant to the present system's quiz engine, which relies on randomised question ordering delivered over the institution's own infrastructure, and underscores the importance of reliable server performance if computer-based testing is to be well received by students.

2.4 Video Conferencing and Virtual Classrooms in E-Learning

Video conferencing has become an established supplement to, and in some contexts a replacement for, face-to-face instruction. Bailey et al. (2022) applied the Technology Acceptance Model to explain how video conferencing use, specifically through the Zoom platform, related to students' perceived learning outcomes, and found that perceived ease of use was a strong predictor of both perceived usefulness and attitude toward the tool, while perceived usefulness predicted intention to continue using the platform without necessarily improving perceived learning outcomes directly. This suggests that simply embedding a video conferencing capability within an LMS is not sufficient on its own; the interface through which students and lecturers access virtual classes must also be straightforward, since ease of use appears to carry more weight than raw feature availability in shaping how favourably a video-based teaching tool is perceived. This consideration informed the decision, described in the methodology, to embed virtual classroom access directly within each course's dashboard rather than requiring users to navigate to a separate application.

2.5 Security Considerations in Web-Based Educational Systems

Because an LMS stores identifiable student records, grades, and submitted coursework, access control and credential protection are central design concerns rather than optional additions. Mpamugo and Ansa (2024) examined the combined use of role-based access control and OAuth 2.0 in a mobile application context and found that layering role-based authorization on top of a secure authentication protocol improved both security enforcement and the manageability of access privileges across different user categories, while reducing the risk of unauthorized access to sensitive resources. Although that study was conducted in a mobile rather than web context, its central conclusion, that role-based access control is effective for restricting users to only the resources appropriate to their designated role, applies directly to a multi-role web-based LMS such as the one described in this paper. On the specific matter of credential storage, Obiri-Tetteh et al. (2025) evaluated password hashing schemes for resource-constrained environments and identified bcrypt as offering a strong balance between computational cost and resistance to brute-force attack, owing to its adaptive work factor and comparatively lightweight resource footprint relative to alternatives. This finding supports the choice of bcrypt-based hashing, implemented through PHP's built-in password hashing functions, for credential storage in the present system.

2.6 Software Development Methodologies for Educational Systems

The choice of software development methodology shapes how requirements are gathered, validated, and translated into a working system. Mishra and Alzoubi (2023) compared structured (waterfall-style) and agile software development approaches and concluded that structured methodologies remain well suited to projects where requirements can be identified with reasonable completeness at the outset and are unlikely to change substantially during development, while agile approaches offer greater advantage where requirements are expected to evolve iteratively in response to ongoing stakeholder feedback. For an institutional LMS whose core requirements are fixed by an existing academic structure, namely departments, sections, coordinators, and semester-bound course assignments, an iterative waterfall approach was judged appropriate, allowing each development phase to be substantially completed before the next began, while still permitting limited revision of earlier phases where testing exposed design gaps.

2.7 Summary and Research Gap

The literature reviewed above indicates three broad, largely separate strands of work relevant to this study: research on e-learning adoption challenges in Nigerian tertiary institutions, research on computer-based testing as a standalone assessment tool, and research on video conferencing and access-control mechanisms considered independently of a specific institutional deployment. Comparatively little published work describes a single, integrated system that combines computer-based testing, video-based instruction, and role-based administrative control within a data model that mirrors the departmental and coordination hierarchy of a specific Nigerian polytechnic. The present study addresses this gap by describing the design and functional evaluation of such an integrated system for Federal Polytechnic Bauchi.

3. Methodology

3.1 System Development Approach

Development followed an iterative waterfall approach comprising four broad phases: requirements analysis, system and database design, implementation, and testing. In the requirements analysis phase, the administrative structure of the institution was reviewed to identify the entities that any course-related transaction ultimately depends on, namely departments, academic sections defined by admission session and level, courses, and the semester-bound assignment of a course to a section and a lecturer. This structure, rather than a generic subject-agnostic model, formed the basis for the database and module design described below. Consistent with the reasoning in Mishra and Alzoubi (2023), a structured approach was favoured over a purely agile one because the institutional hierarchy that the system needed to represent was fixed and well understood in advance, even though individual modules, such as the quiz engine and virtual classroom, were refined iteratively once initial versions had been tested.

3.2 System Architecture

The system follows a three-tier client-server architecture. The presentation tier consists of server-rendered HTML pages styled with Tailwind CSS and Font Awesome icons, delivered to a standard web browser without requiring a dedicated client application. The application (business logic) tier is implemented in procedural PHP, with each functional area of the system, such as course assignment, material management, or quiz administration, handled by a dedicated PHP script that validates the active user's session and role before performing any database operation. The data tier consists of a MySQL relational database accessed through PHP Data Objects (PDO) with parameterised queries, which was adopted specifically to prevent SQL injection by ensuring that user-supplied input is never concatenated directly into a SQL statement.

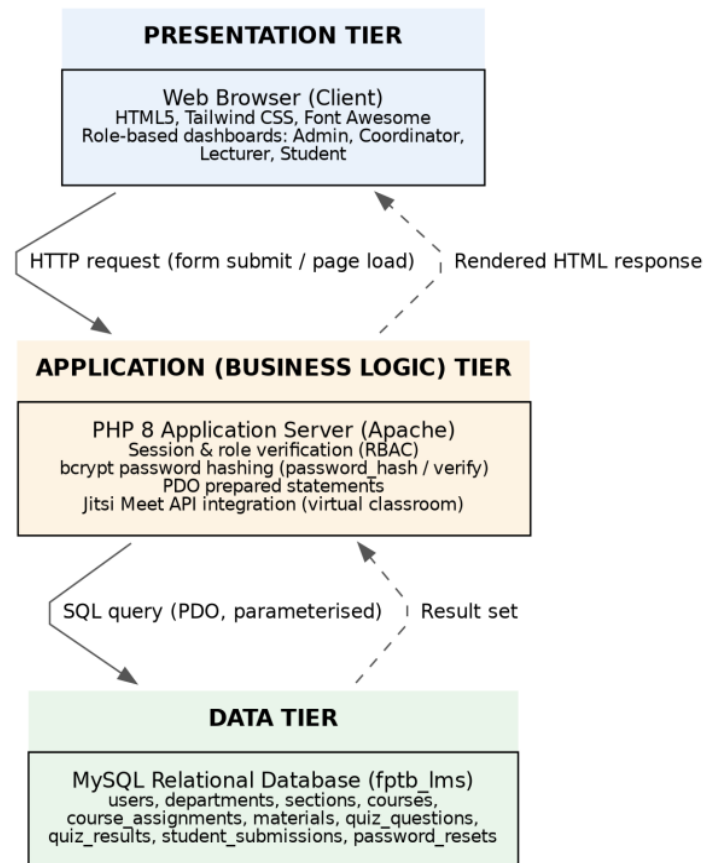


Figure 1: Three-Tier System Architecture of the LMS

3.3 Database Design

The database is organised around ten core tables. The users table stores each account's credentials, hashed password, assigned role (administrator, coordinator, lecturer, or student), department, and approval status, since new registrations remain pending until approved by an administrator or coordinator. The departments and sections tables record the institution's academic departments and the cohort-based sections within them, distinguished by admission session and academic level. The courses table holds the catalogue of available courses, while course_assignments links a specific course to a specific section, lecturer, semester, and academic session, forming the central table around which materials, quizzes, and submissions are organised. The materials table stores uploaded learning content tied to a course assignment, while quiz_questions and quiz_results together implement the computer-based testing engine, recording each question, its options and correct answer, and each student's submitted responses and computed score. The student_submissions table records uploaded assignment files together with the grade and feedback subsequently entered by the lecturer. Finally, the password_resets table supports a token-based password recovery workflow with a defined expiry window.

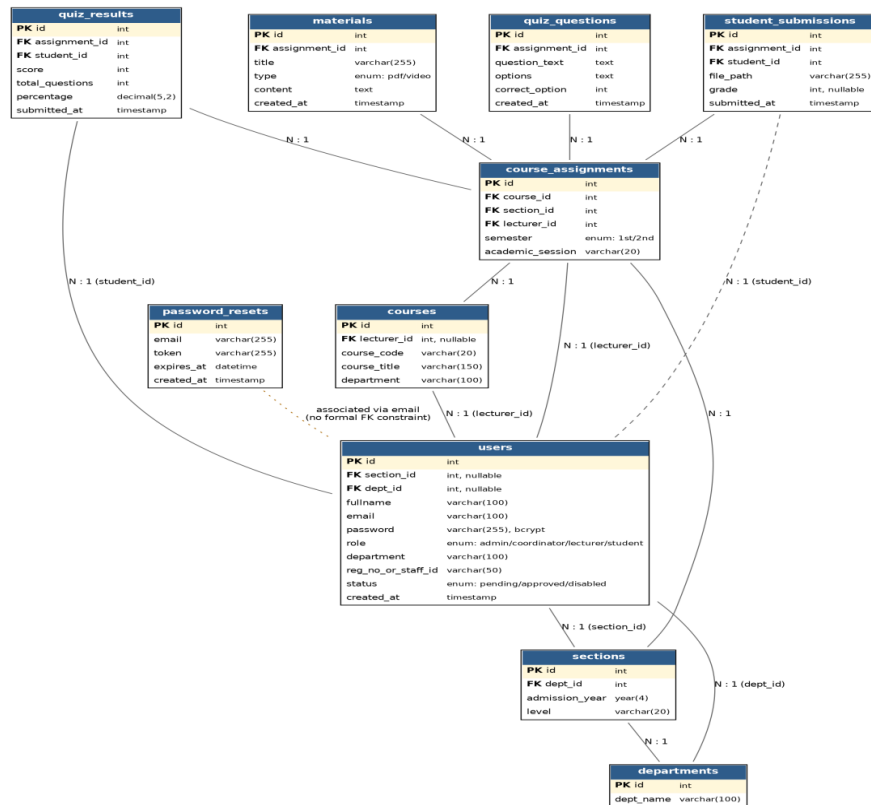


Figure 2: Entity-Relationship Diagram of the LMS Database

3.4 Functional Modules

The system is organised around four role-differentiated modules, each accessible only after authentication and only to accounts assigned the corresponding role.

- **Administrator module.** The administrator account approves or rejects pending user registrations, manages departments and coordinators, and processes bulk student promotion between academic levels at the end of a session.
- **Coordinator module.** Departmental coordinators assign courses to lecturers for a given section and semester, and create or update the academic sections within their department.
- **Lecturer module.** Lecturers upload learning materials to their assigned courses, create quizzes with multiple-choice questions, grade submitted assignments with accompanying feedback, host virtual class sessions, and review a summary of quiz results for their students.
- **Student module.** Students view the courses attached to their section, download materials, attempt quizzes, upload assignments, and join scheduled virtual classes.

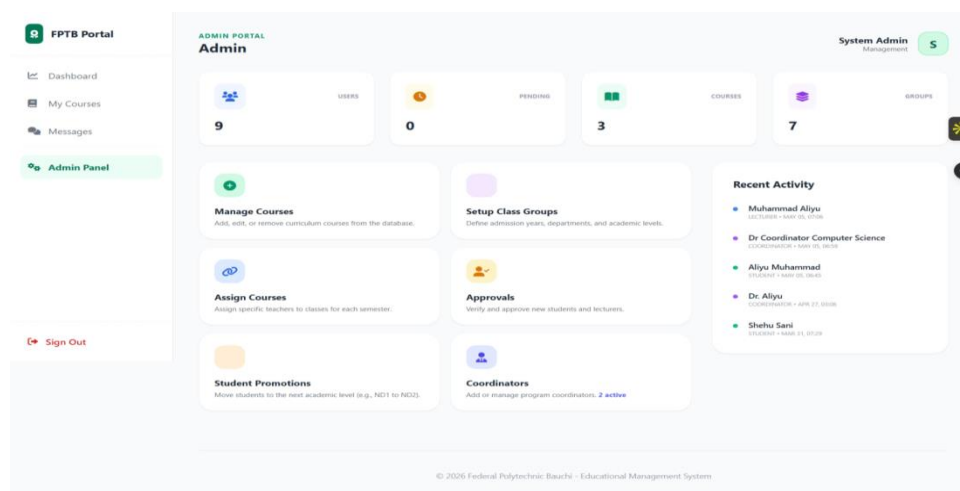


Figure 3: Administrator Dashboard

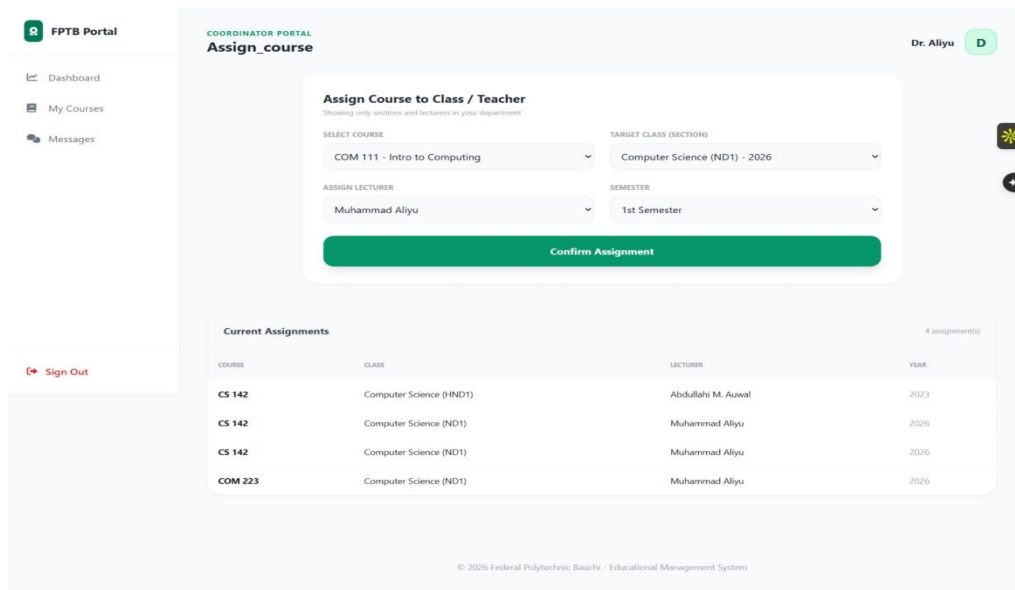


Figure 4: Coordinator Dashboard — Course Assignment

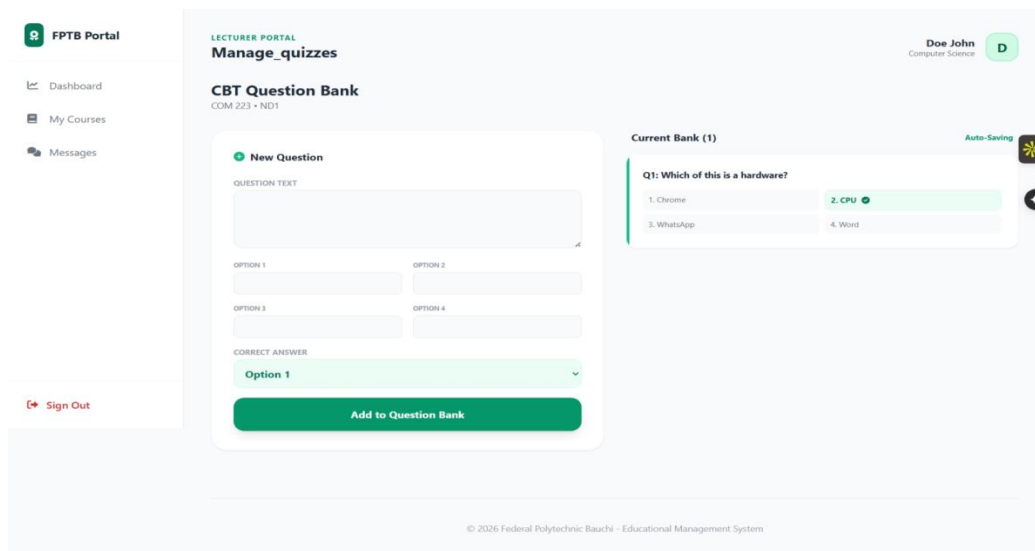


Figure 5: Lecturer Dashboard — ManageQuizzes

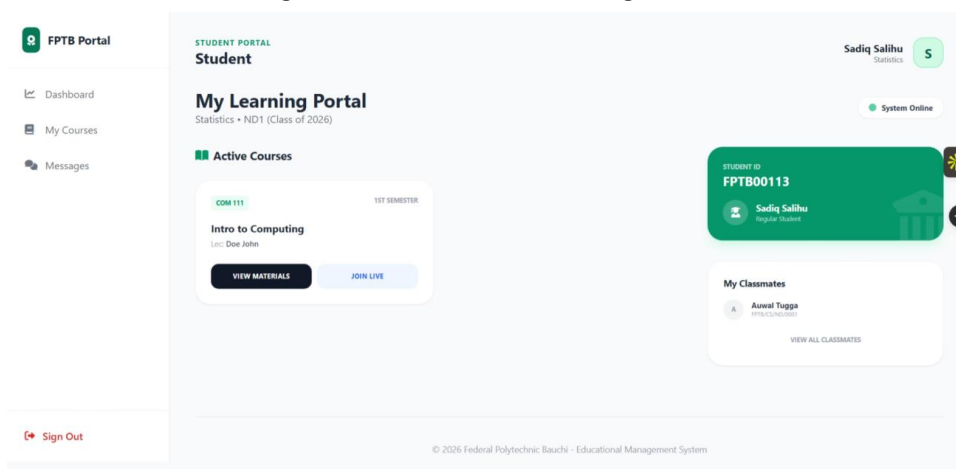


Figure 6: Student Dashboard

3.5 Security Implementation

User sessions are managed through PHP's native session mechanism, with each protected page verifying, at the top of the script, that a session exists and that the session's stored role matches the role required for that page, in line with the role-based access control approach evaluated favourably by Mpamugo and Ansa (2024). Passwords are never stored in plain text; instead, they are hashed at registration using PHP's `password_hash()` function, which implements the bcrypt algorithm identified by Obiri-Tetteh et al. (2025) as offering a favourable balance of computational cost and resistance to brute-force attack, and verified at login using the corresponding `password_verify()` function. All database queries that incorporate user-supplied input are executed as parameterised statements through PDO, which prevents SQL injection by separating the query structure from the data supplied to it. Password recovery is handled through a time-limited token stored in the `password_resets` table, so that a reset link becomes invalid after a defined expiry period even if it is not used.

3.6 Computer-Based Testing Engine

The quiz engine allows a lecturer to attach a set of multiple-choice questions to a specific course assignment. When a student attempts a quiz, the questions are retrieved and presented in randomised order, reducing the likelihood of students in the same sitting comparing answers by question sequence, a safeguard consistent with the concerns about assessment integrity that motivate computer-based testing generally, as discussed by Ajinaja (2017). Scores are computed automatically on submission and stored in the `quiz_results` table, from which a lecturer can subsequently generate a class-wide summary of performance. This design responds directly to the concerns raised by students in Zakariya et al. (2026), where technical reliability and format rigidity were cited as limitations of computer-based testing; the present engine addresses format rigidity only partially, since it currently supports multiple-choice items alone, a limitation discussed further in Section 5.

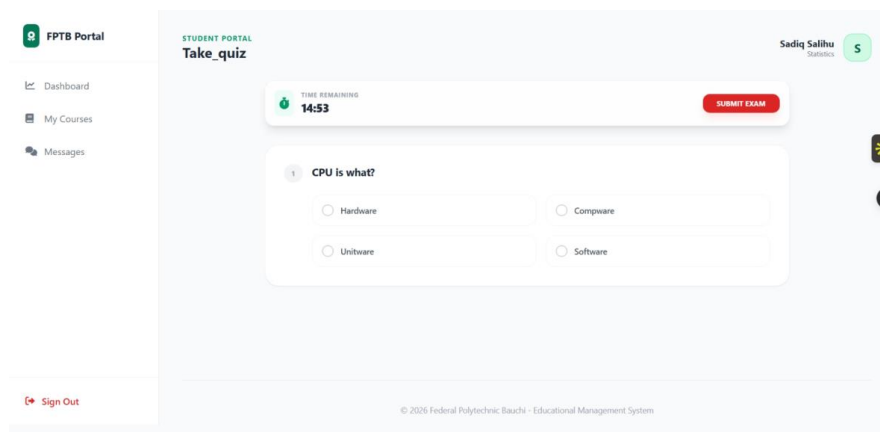


Figure 7: Student Taking a Quiz

3.7 Virtual Classroom Module

Live instruction is supported through a virtual classroom module built on the Jitsi Meet application programming interface, embedded directly within the course dashboard so that students and lecturers can join a session without leaving the LMS or installing separate client software. This design decision reflects the finding of Bailey et al. (2022) that perceived ease of use is a strong predictor of favourable attitudes toward video-based teaching tools; embedding the meeting interface directly in the page a student already uses to access course materials was intended to minimise the additional effort required to join a live session.

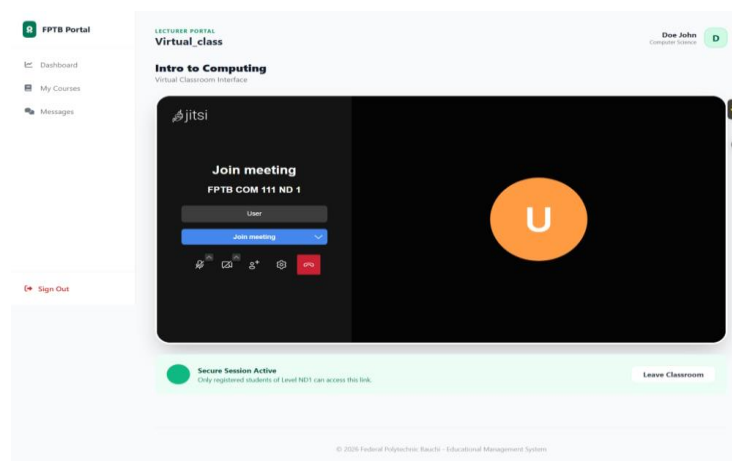


Figure 8: Virtual Classroom Interface

3.8 Development Tools and Technologies

Component	Technology Used
Server-side scripting	PHP 8
Database	MySQL, accessed via PHP Data Objects (PDO)
Front-end styling	Tailwind CSS, Font Awesome
Video conferencing	Jitsi Meet API
Local development server	XAMPP / Apache
Code editor	Visual Studio Code

3.9 Testing Methodology

The implemented system was evaluated using black-box functional testing, an approach in which each module is exercised through its expected inputs and outputs without inspection of the underlying code, consistent with the black-box testing methodology applied to PHP and MySQL-based web applications by Rizqiananda and Anwar (2025). Test cases were designed to cover the primary functional paths of each module, including successful and unsuccessful login attempts, the registration-approval workflow, course assignment, material upload, quiz creation and attempt, assignment submission and grading, and access restriction across the four user roles. The template used to record these test cases, together with a column for the actual outcome, is presented in Section 4.3; the developer should execute each test case against the live deployment and record the observed result and date before this paper is finalised, since presenting untested outcomes as findings would misrepresent the evaluation.

4. Results

4.1 Implemented System Overview

The implemented system provides a single login and registration entry point, from which the application routes each authenticated user to the dashboard appropriate to their assigned role. New accounts are created in a pending state and become active only once approved by an administrator or the relevant departmental coordinator, preventing unverified accounts from accessing course content.

Figure 9: Login and Registration Page

4.2 Database Implementation

The relational schema described in Section 3.3 was implemented in MySQL and comprises ten tables with foreign-key relationships enforcing referential integrity between users, departments, sections, course assignments, materials, quizzes, and submissions. All ten tables were created

No.	Module	Test Case	Expected Result	Result (to be completed)
1	Authentication	Log in with valid credentials	User is redirected to the dashboard matching their role	
2	Authentication	Log in with invalid password	System rejects login and displays an error message	
3	Registration	Register a new student account	Account is created with a pending status, not yet able to log in	
4	Admin	Approve a pending registration	Account status changes to approved and the user can log in	
5	Coordinator	Assign a course to a lecturer and section	Course appears under the lecturer's and students' dashboards	
6	Lecturer	Upload a course material	Material becomes visible to students in that section	
7	Lecturer	Create a quiz with multiple-choice questions	Quiz becomes available to enrolled students	
8	Student	Attempt a quiz	Questions appear in randomised order and a score is generated on submission	
9	Student	Submit an assignment file	File is stored and visible to the lecturer for grading	
10	Lecturer	Grade a submitted assignment	Grade and feedback are saved and visible to the student	
11	Access control	Attempt to access a lecturer-only page as a student	Access is denied and the user is redirected	
12	Virtual class	Join a scheduled virtual class session	Jitsi Meet session loads within the course dashboard	

successfully and populated with representative test data during development.

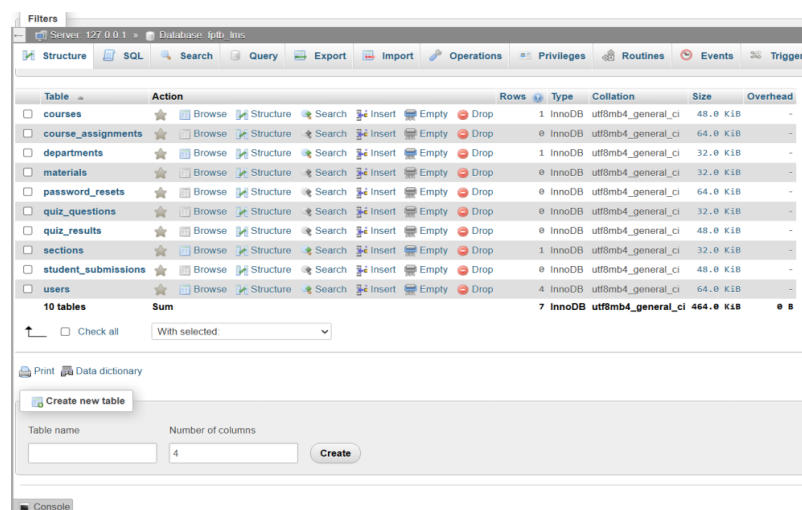


Figure 10: Database Schema View

4.3 Functional Test Results

Table 2 presents the functional test cases designed to evaluate each core module. The Result column should be completed with the actual outcome (Pass or Fail) observed when each test case is executed against the deployed system, together with the date of testing; the entries below are provided as a template only and should not be treated as reported findings until verified.

4.4 Summary of Results

The system's core modules, namely authentication, registration and approval, course assignment, material distribution, computer-based quizzes, assignment grading, role-based access restriction, and the virtual classroom, were each implemented and exercised during development. Role-based access control consistently restricted each account to the pages appropriate to its assigned role, password hashing and parameterised queries were confirmed to be in place across all data-entry points, and the quiz engine correctly randomised question order and computed scores automatically on submission. A complete account of formally recorded test outcomes, following the template in Table 2, should accompany the final version of this paper.

5. Discussion

The findings reported in Section 4 suggest that a self-hosted, PHP-based LMS can reproduce, within a single system, functionality that the reviewed literature has generally treated as separate concerns. Where Ajinaja (2017) and Zakariya et al. (2026) considered computer-based testing largely in isolation, and Bailey et al. (2022) examined video conferencing as a standalone tool independent of any surrounding course-management structure, the present system embeds both within a data model organised around the institution's own departmental and coordination hierarchy. This integration directly addresses the fragmentation described in Section 1.2, where materials, assessment, and communication had previously been scattered across informal channels; concentrating these functions within one system with a shared user base and shared course records should, in principle, make institutional record-keeping more consistent, though this claim would benefit from confirmation through actual usage data once the system is deployed.

The security measures adopted, namely bcrypt password hashing and role-based access control, are consistent with the recommendations of Obiri-Tetteh et al. (2025) and Mpamugo and Ansa (2024) respectively, and address the most immediate risks to a system that stores student records and grades. Nevertheless, the current implementation relies on PHP session-based authentication without multi-factor authentication or enforced HTTPS at the infrastructure level, both of which the broader access-control literature treats as complementary rather than optional safeguards; their absence represents a gap that should be closed before the system handles live student data at scale.

The quiz engine's reliance on the multiple-choice format only partially addresses the concerns raised by students in Zakariya et al. (2026), where format rigidity was cited alongside technical reliability as a source of dissatisfaction with computer-based testing. Extending the engine to support short-answer or file-based responses, subject to appropriate grading workflows, would address this limitation but was outside the scope of the present implementation. Similarly, while the virtual classroom module was designed with ease of access in mind, following the reasoning in Bailey et al. (2022), its performance under a realistic concurrent user load, such as an entire section joining a single session simultaneously, has not yet been evaluated, and Jitsi Meet's documented participant-capacity limitations mean that very large cohorts may require session splitting or an alternative provider.

More broadly, the persistence of infrastructural constraints documented by Ogunji et al. (2022), Oyediran et al. (2020), and Igbini and Ottah (2024), namely unreliable power supply and limited internet bandwidth, means that even a well-designed LMS cannot by itself guarantee adoption or effective use. The lightweight architecture adopted here, built on standard PHP and MySQL rather than a heavier framework, was chosen partly in anticipation of these constraints, but usability under genuinely low-bandwidth conditions remains to be verified through field testing rather than assumed from the choice of technology stack alone.

Finally, this evaluation was limited to black-box functional testing of the system's own developer, following the methodology described by Rizqiananda and Anwar (2025); it does not constitute a usability or technology-acceptance study of the kind conducted by Zakariya et al. (2026) or Bailey et al. (2022), and the results reported here should accordingly be read as evidence that the system functions as designed, rather than as evidence of how favourably lecturers and students would rate it in practice.

6. Conclusion

This study set out to design, implement, and functionally evaluate a role-based web-based Learning Management System for Federal Polytechnic Bauchi, built around the institution's own departmental and coordination structure rather than a generic subject-agnostic model. The resulting system integrates administrator, coordinator, lecturer, and student roles within a single relational database, secures access through bcrypt password hashing and role-based access control, and incorporates both a randomised computer-based testing engine and a Jitsi Meet-based virtual classroom. Black-box functional testing of the core modules indicated that the system performed its intended functions correctly, supporting the conclusion that a lightweight, self-hosted LMS of this kind is a feasible and cost-effective option for a Nigerian polytechnic operating under infrastructural and budgetary constraints.

REFERENCES

- Ajinaja, M. (2017). The design and implementation of a computer based testing system using component-based software engineering. *International Journal of Computer Science and Technology*, 8(1), 58–65.
- Bailey, D. R., Almusharraf, N., & Almusharraf, A. (2022). Video conferencing in the e-learning context: Explaining learning outcome with the technology acceptance model. *Education and Information Technologies*, 27(6), 7679–7698. <https://doi.org/10.1007/s10639-022-10949-1>
- Igbinovia, M. O., & Ottah, G. A. (2024). E-learning in the post-COVID era: Undergraduate students' experience in Nigerian universities. *Howard Journal of Communications*, 37(1). <https://doi.org/10.1080/10646175.2024.2420115>
- Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: A comparative analysis. *International Journal of System Assurance Engineering and Management*, 14(4), 1504–1522. <https://doi.org/10.1007/s13198-023-01958-5>
- Mpamugo, E., & Ansa, G. (2024). Enhancing network security in mobile applications with role-based access control. *Journal of Information Systems and Informatics*, 6(3), 1872–1899. <https://doi.org/10.51519/journalisi.v6i3.863>
- Obiri-Tetteh, J. A., Turkson, R. E., Frimpong, E. A., et al. (2025). Bcrypt-based optimized password hashing against targeted Internet of Things attacks. *Cureus Journal of Computer Science*, 2, es44389-025-05167-y. <https://doi.org/10.7759/s44389-025-05167-y>
- Ogunji, C. V., Onwe, J. O., Ngwa, E. S., David, E., Olaolu, M., & Cresantus, B. (2022). Higher education and the new normal: Implications for sustainable post COVID-19 era in Nigerian tertiary institutions. *Cogent Education*, 9(1), Article 2125206. <https://doi.org/10.1080/2331186X.2022.2125206>
- Oyediran, W. O., Omoare, A. M., Owoyemi, M. A., Adejobi, A. O., & Fasasi, R. B. (2020). Prospects and limitations of e-learning application in private tertiary institutions amidst COVID-19 lockdown in Nigeria. *Heliyon*, 6(11), Article e05457. <https://doi.org/10.1016/j.heliyon.2020.e05457>
- Rizqiananda, N., & Anwar, C. (2025). Software testing analysis on PHP and MySQL-based student data CRUD application using black box testing. *Journal of Information Systems and Business Technology*, 1(4), 40–43.
- Zakariya, Y. F., Alotaibi, S. B., Alrashood, J. S., & Alrosaa, T. M. (2026). Computer-based testing in higher education: A phenomenology investigation into undergraduate students' perspectives through the technology acceptance model. *Frontiers in Psychology*, 17, Article 1602964. <https://doi.org/10.3389/fpsyg.2026.1602964>