VOL. 12 NO. 9,
MAY, 2026
E-ISSN 3026-958X
P-ISSN 3027-1169

African Scholar Publications & Research International
www.africanscholarpub.com

Journal of Science Innovation & Technology Research (JSITR)

Design and Implementation of a Web-Based Management System for Cross-Departmental General Studies Programmes in Nigerian Universities

Daramola Christiana Jumoke[1]; Anieke Somtochukwu Henry[2]; Faji Babasolape Oluwademilade[3]; Ogakwu Okwudilichukwu Oshokha[4]; & Udosen Alfred Akpan[5]

[1]Department of Software Engineering, School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria. [2,3,4,5]Department of Computer Science, School of Computing, Babcock University, Ilishan-Remo, Ogun State, Nigeria.

Corresponding Author: aniekes5782@student.babcock.edu.ng

DOI: https://doi.org/10.70382/ajsitr.v12i9.085

Abstract

Running General Studies (GEDs) courses at a Nigerian university like Babcock University is about running thousands of students from totally different departments in the same courses at the same time. The University Management Information System (UMIS) does reasonably well at managing registrations and computer-based testing, but it was designed around typical departmental programs and simply does not address the unique workflows necessary to administer the GED. This paper presents the design and implementation of a web-based General Studies Management System (GSMS) to address this gap. The backend was developed in Python using the Django framework, the database engine was SQLite, and the frontend was built with standard HTML, CSS, and JavaScript. The system has three different access levels namely the administrator, lecturer and student, each with a dedicated dashboard and permission. All GEDs-related processes like course registration, uploading lecture material, automated grading, mock exams, academic result reporting, all exist in one place. We put it through the paces and verified that the platform withstands real-world operational loads, while ensuring that each user's data is completely isolated from the others. System evaluation involved 20 participants

comprising 15 undergraduate students, two GEDs lecturers, and three administrators. A total of 20 functional test cases were executed across seven system modules, all of which passed successfully, representing a 100% pass rate. The GSMS replaces the patchwork of workarounds students and staff use today like WhatsApp groups, shared drives, manual grade sheets because UMIS was never built for what GEDs actually needs.

Keywords: Course Management System, Django Framework, Educational Technology, General Studies (GEDs), Learning Management System, Role-Based Access Control, Web Application.

Introduction

Every Nigerian undergraduate has to attend a General Study class regardless of what department they are from. Philosophy students take a course with engineering students. Law students are on the same enrolment list as Computer Scientists. Managing such cross-departmental enrolment presents significant administrative complexity that most institutional platforms were not designed to accommodate. In the last decade, Learning Management Systems (LMS) have become a common infrastructure in higher education worldwide (Fibriasari et al., 2023; Rohit et al., 2021). They handle enrolment, grading, and communications all in one place. The problem is that most of them were created for standard departmental programs in which students and courses are in one clear organisational unit and General Studies courses break that assumption completely.

Digital platform use in universities has sharply increased during the COVID-19 pandemic. Sofi-Karim et al. (2022) studied English-language teachers in Iraqi universities and found that even when platforms existed, poor internet access, a lack of devices and no training prevented effective use. This finding translates directly to the Nigerian university context. Huang (2024) and Wang, Y. (2023) both show how well-integrated management systems reduce data duplication and provide administrators with real-time insights – but only if those systems are truly designed for the institution's requirements.

Researchers who have looked at general education and cross-departmental programs specifically keep reaching the same conclusion: standard platforms fall short. Zhang (2024) designed a network-based course system for general education in China and found that role-specific access controls, adaptive assessments, and a centralized resource repository are all prerequisites that generic platforms rarely offer together. Wong-Fajardo et al. (2021) came to the same conclusion studying integrated academic management at a Peruvian university that if a program spans multiple faculties, it needs tools built for that, not a general-purpose platform stretched beyond its original design At Babcock University, GEDs courses are not optional. Under the National Universities Commission's Core Curriculum and Minimum Academic Standards (CCMAS)

framework (National Universities Commission, 2023), every undergraduate must pass modules in communication, entrepreneurship, philosophy, and related subjects before graduating. Thousands of students from every faculty register for the same courses each semester. The university manages these courses through UMIS, its primary institutional platform, which handles registration and computer-based testing adequately, but lacks dedicated infrastructure for GEDs lecture notes, past papers, or grading workflows specific to how GEDs assessment is structured (Umar Ong, 2021).

In the absence of a centralised institutional repository, students routinely resort to unofficial channels such as messaging applications and personal cloud storage to access course materials. Adedoyin et al. (2023) explored how such fragmentation of information leads to access inequality among students in the Nigerian context. Ayanbode et al. (2022) extended on that research using quantitative data collected at Babcock University in Nigeria, revealing that the quality of teacher-student interactions emerged as the key determinant of both learning satisfaction and progression. Such reliance on unofficial distribution channels undermines the institutional conditions required for meaningful student-teacher engagement, directly compromising the academic outcomes that Ayanbode et al. (2022) identify as dependent on that interaction.

This paper presents the design and implementation of a web-based General Studies Management System (GSMS) developed specifically to address the administrative requirements of cross-departmental GEDs administration at Babcock University. The system was created using the following technologies: Python, Django, and SQLite, with access for three types of users: administrators, lecturers, and students. All processes such as registration, material upload, grading, quizzes, and result processing happen within this system exclusively designed for GEDs.

The study has three objectives: (a) to design a GEDs management system built around the cross-departmental structure of the program; (b) to implement a role-based portal with separate, functional interfaces for administrators, lecturers, and students; and (c) to test the system under conditions that reflect actual GEDs operational workflows.

Problem Statement

It should be noted that UMIS itself is not a problematic system – it works exactly as planned. Registration of standard courses and computerized assessment work perfectly. The problem lies in design. The UMIS system was designed with the premise that each of its elements, namely, students, classes, and staff, belong to specific departmental structures. In turn, GEDs does not adhere to this logic. Classes in the field of GEDs involve students from all faculties at once, and accordingly, its data management system, including enrollment lists, class grades, material archives, etc., does not fit into the established logic of any department of UMIS. The absence of alignment in these two areas leads to four distinct drawbacks in UMIS that multiply with each new academic year: (a) GEDs lecturers cannot use institutional platforms to upload any material associated with their lectures, such as lecture notes, tests, and additional

resources; instead, all this is carried out in informal WhatsApp groups and personal clouds; (b) automated scoring of classes conducted by GEDs standards is not implemented in UMIS, which means that the calculation of grades must be done manually by teachers, who may have to grade thousands of students' papers; (c) students cannot have direct communication channel with GEDs staff within any platform, making it hard to get timely answers about course requirements; and (d) real-time result computation and GEDs-specific academic reporting are unsupported, which delays feedback and creates problems for institutional quality assurance reporting.

These gaps affect everyone involved in different ways. Students who do not have direct or reliable access to official materials usually end up falling behind others who are lucky enough to be in active group chats where information is shared faster. Lecturers also have their own challenges, especially when they have to calculate and manage grades manually for large classes, because mistakes are bound to happen and fixing those mistakes later only adds more stress to the process. For administrators, it becomes difficult to get accurate and properly organized GEDs data when everything is scattered across different channels and systems. At this point, a platform built specifically to solve these issues is not just something nice to have. It is something institutions genuinely need if they want to run GEDs properly and efficiently on a larger scale.

Literature Review

General Studies in Higher Education

General Studies programs are compulsory in universities almost everywhere and in Nigeria the NUC's CCMAS framework also made it compulsory for every undergraduate student no matter the faculty (National Universities Commission, 2023). Because of that, the problem is not really just teaching the courses alone. Schools also have to manage registration, course materials, grades and communication for thousands of students at the same time and that can become stressful very quickly.

Bayode (2022) explained that paper methods and scattered digital systems do not really work well anymore once the number of students becomes too much because issues like mistakes, duplicate records and even missing data start becoming common. Ike (2023) also noticed similar problems while designing a student information system for a Nigerian federal university. Both studies were basically pointing to the same thing, that centralized and automated systems with proper databases are not just extra improvements for GEDs administration anymore, schools actually need them for things to run properly.

Zhang (2024) and Wong-Fajardo et al. (2021) also supported this by explaining that automated systems handle communication and data processing across departments much better than manual methods or random general systems. Once the student population keeps increasing, manual processes become harder to control accurately. Looking at all the studies together, it is clear that managing GEDs programs effectively now requires a digital platform that was specifically built for it.

Educational Management Systems and LMS

Educational Management Systems are mainly designed to bring different academic activities like registration, scheduling, grading and attendance into one centralized system instead of having everything scattered across different departments. According to Huang (2024) and Wang, Y. (2023), this helps administrators see information in real time and also reduces the repetition and confusion that usually happens when different departments are using separate systems. Wang, Q. (2023) and Liu (2023) also studied some universities in China and found that centralized modular systems improved data consistency and made responses faster during day-to-day operations.

Learning Management Systems (LMS) take things further by helping with the actual delivery of courses and learning materials. Decuypere et al. (2021) explained that there are different types of LMS platforms. Some are installed directly within the institution which gives the school more control, while some are cloud-based systems that reduce stress on the institution's IT team. There are also open-source platforms like Moodle that schools can modify themselves, and commercial platforms like Blackboard that usually come with professional support but require subscription payments. Each option has its own advantages and disadvantages depending on cost, flexibility and how scalable the system needs to be.

Fibriasari et al. (2023) found that after the pandemic, many students preferred using LMS platforms because they made learning materials easier to access and improved communication between students and lecturers. At the same time, poor internet connectivity was still identified as a major challenge affecting effective usage. This is something that is also very relevant in the Nigerian university environment where internet access and bandwidth can vary a lot between students and campuses.

Information Systems Theories and Educational Technology Frameworks

The adoption of educational information systems is shaped by established theoretical frameworks that explain when and why technology integration succeeds in academic environments. The Technology Acceptance Model (TAM), introduced by Davis (1989), identifies perceived usefulness and perceived ease of use as the two primary determinants of user adoption of information technology. In educational settings, TAM has been widely applied to explain student and faculty acceptance of learning platforms, consistently finding that systems perceived as both administratively useful and easy to navigate achieve higher and more sustained adoption rates (Alia, 2022). These principles directly informed the design approach of the GSMS: by constraining each user interface to the functions relevant to that role only, the platform minimises navigational complexity and increases perceived utility for each user group.

From a development methodology standpoint, the iterative approach adopted in this study aligns with Agile software development principles, which prioritise working software, stakeholder collaboration, and responsiveness to change over rigid upfront specification (Vlasenko et al., 2021). Husain (2023) and Umar Ong (2021) both emphasise that academic

management systems developed through structured, stakeholder-informed processes produce more contextually appropriate solutions than those designed in isolation. The iterative development of the GSMS, incorporating direct feedback from GEDs lecturers during the testing phase, reflects this evidence-based design approach.

Challenges in Managing GEDs Programs

Managing GEDs programs on a large scale comes with a lot of administrative problems, especially when the systems being used are not properly connected. Ayanbode et al. (2022) found at Babcock University that student-teacher interaction plays a major role in how satisfied students feel academically and how much progress they believe they are making. Looking at that finding, the lack of a proper communication system for GEDs becomes an even bigger issue because it reflects problems students are already facing in reality at Babcock University.

When lecturers and administrators are trying to manage registration, attendance and student records across different departments using spreadsheets and scattered tools, mistakes start building up very quickly. Bayode (2022) explained that duplicate records, incorrect data and even missing files become common once decentralized systems are handling large amounts of information across departments. The same thing also affects assessment processes. Once lecturers have to grade very large classes manually, delays in results become normal, feedback becomes less effective and keeping proper records for NUC auditing becomes difficult (Ike, 2023; Zhang, 2024).

Communication is another major problem. Most times, important GEDs information is passed through unofficial channels or random group chats, and once the number of students keeps increasing, the chances of students missing updates also increases. In a system already managing thousands of students at once, those communication gaps can easily create confusion and affect the overall coordination of the program.

Review of Related Systems

Table 1 presents the most relevant related systems reviewed for this study.

Table 1. Summary of Reviewed Related Literature

S/N	Author(s) / Year	Objective	Key Findings / Limitations
1	Zhang (2024)	Design a virtual platform for GE courses with role-specific functions and adaptive assessments	Streamlined teaching; adaptive assessments; role access controls; internet-dependent
2	Wong-Fajardo et al. (2021)	Integrate academic management with LMS for competency-based cross-departmental education	Unified planning; real-time updates; scalability beyond single institution untested
3	Ayanbode et al. (2022)	Investigate how student-teacher interaction affects satisfaction at Babcock University	Student-teacher interaction is the strongest predictor of academic satisfaction; single-institution scope

S/N	Author(s) / Year	Objective	Key Findings / Limitations
4	Ike (2023)	Digitize student data and automate course registration at a Nigerian federal university	Reduced errors; real-time access; mobile-friendly; limited to one department
5	Bayode (2022)	Replace manual school administration with an automated web-based management system	Time-saving automation; all-in-one features; lacks advanced analytics
6	Adedoyin et al. (2023)	Build an integrated system for course creation, grading, and virtual classrooms	Over 90% user satisfaction; poor mobile performance; no team-teaching features
7	Vlasenko et al. (2021)	Design and manage an online educational platform combining software development and pedagogy	High team and user satisfaction; best-practice integration; scalability untested

Research Gap

The systems reviewed in Table 1 all handled different parts of academic management fairly well, but none of them completely addressed the exact problems surrounding GEDs administration in Nigerian universities. Zhang (2024) came very close in terms of the overall idea, but the system was designed around a Chinese university environment and did not really account for the specific structure and challenges of Nigerian GEDs programs. Ike (2023) also showed that a Python-based system could work effectively for managing student information, but the work was limited to a single department and not something operating across an entire university structure. Adedoyin et al. (2023) recorded good satisfaction levels from users, although they still pointed out issues with mobile accessibility and reporting features. Ayanbode et al. (2022) provided the closest institutional context because the study focused on Babcock University itself, but the research was more about interaction and learning experiences rather than designing an actual management platform.

From the studies reviewed, none of the systems fully combined cross-departmental data management, centralized sharing of GEDs materials, automated grading processes for GEDs courses, and separate interfaces for students, lecturers and administrators inside one dedicated platform designed specifically for Nigerian GEDs administration. That is the gap this study is trying to address.

Methodology

Development Approach

The GSMS was developed using an iterative approach instead of trying to design every single feature completely before development started. We built the system module by module, testing and validating each part before moving to the next one. This approach made it easier to adjust certain parts of the architecture as development progressed, especially because the system was being designed around the real operational challenges involved in managing GEDs in a university environment.

Before development started, we first looked at the existing systems already available and identified where UMIS was struggling specifically with GEDs-related processes. We also examined platforms like Moodle and Blackboard as possible alternatives, but it became clear that neither system could properly support the kind of cross-departmental workflows and GEDs-specific grading structure required without heavy customization. At that stage, building a dedicated system became the more practical option both technically and economically.

For development, Django was selected after comparing it with Laravel and Spring Boot. The authentication system, ORM and Django REST Framework already provided a lot of the core infrastructure needed, which reduced the amount of foundational work we would have had to build manually. Before implementation began, the workflows and relationships within the system were first mapped using draw.io, and the database was normalized around 15 core entities from the beginning to reduce redundancy and prevent duplicate records.

Requirement Gathering and Stakeholder Involvement

Prior to system development, requirements were gathered through three complementary methods. First, all three members of the development team conducted direct observation of UMIS usage patterns during routine GEDs administrative activities, systematically documenting specific functional limitations and workflow gaps. Second, semi-structured interviews were conducted with GEDs teaching staff to capture operational pain points from the lecturer perspective. These interviews proved particularly instructive: one GEDs lecturer identified a significant gap in the system's provision for direct lecturer-student communication. This finding was subsequently incorporated into the system design and validated in a later development iteration, resulting in the integration of dedicated in-app communication channels. Third, the development team drew on their own experience as enrolled Babcock University students who had directly encountered the material access and communication challenges the system addresses.

The combination of direct observation, stakeholder interviews, and first-hand institutional experience provided an empirical basis for defining system requirements and validating design decisions prior to implementation, consistent with stakeholder-centred requirement elicitation approaches recommended in the information systems development literature (Husain, 2023; Umar Ong, 2021).

System Requirements

Table 2 lists the system requirements for the GSMS using MoSCoW prioritization, separating what the system must do from what it should do and what could be added later.

Table 2. GSMS System Requirements (MoSCoW)

Req. ID	Requirement	Category	Priority
R1	The system must provide secure role-based login for administrators, lecturers, and students	Functional	Must-have

Req. ID	Requirement	Category	Priority
R2	The system must allow administrators to create, update, and delete courses and user accounts	Functional	Must-have
R3	The system must restrict lecturer-to-course assignment to administrators only	Functional	Must-have
R4	The system must allow lecturers to upload lecture notes, past papers, and supplementary materials	Functional	Must-have
R5	The system must enable students to register for GEDs courses within defined academic periods	Functional	Must-have
R6	The system must deliver quizzes and examinations in-browser with configurable attempt controls	Functional	Must-have
R7	The system must automatically compute total scores, grade points, letter grades, and comments	Functional	Must-have
R8	The system must generate downloadable PDFs for registration slips and academic results	Functional	Should-have
R9	The system must log all administrative actions in a real-time dashboard audit trail	Functional	Should-have
R10	The system must isolate user data so each role only accesses records relevant to them	Non-Functional	Must-have
R11	The system must respond to standard requests within acceptable time under normal load	Non-Functional	Must-have
R12	The system must be compatible with Chrome, Firefox, and Edge browsers	Non-Functional	Must-have
R13	The system should maintain consistent operation with error handling for unexpected failures	Non-Functional	Should-have
R14	The architecture should support future expansion to additional users and courses	Non-Functional	Should-have
R15	The system could incorporate mobile-optimized interfaces for smartphone users	Non-Functional	Could-have

System Architecture

The GSMS was built using a three-tier architecture made up of the presentation layer, business layer and data layer. The presentation layer uses HTML, CSS and JavaScript and handles everything users interact with directly. Students, lecturers and administrators all see different dashboards and features immediately after logging in. The business layer was developed with Python, Django and Django REST Framework, where the application logic, validation and access control are handled. The data layer uses SQLite through Django ORM, with all system data stored around the 15 core entities defined during design.

Separating the system into these layers made the codebase easier to manage and more modular during development. It also means updates or fixes can be made to one module without affecting unrelated parts of the system. The structure also supports future scaling if the number of users increases. Figure 1 shows the relationship between the three layers.

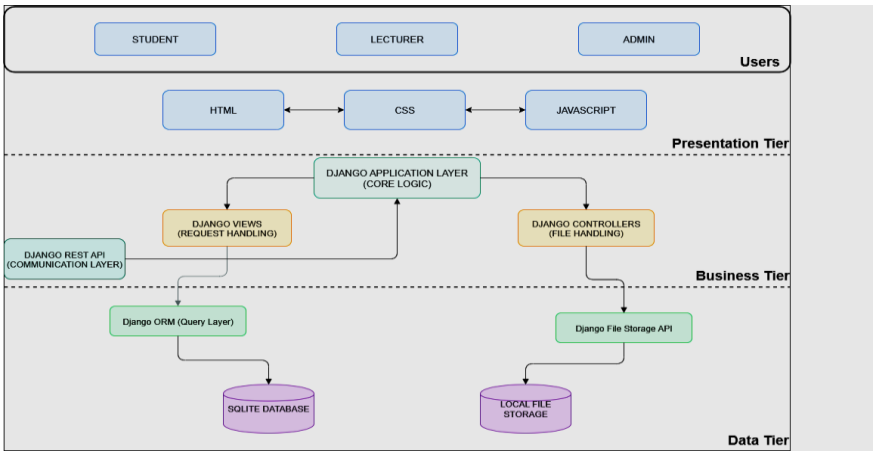


Fig 1. Three-Tier System Architecture Diagram

System Flowchart and Data Flow Diagram

The system flowchart in Figure 2 shows the movement of users through the system starting from login. After authentication, the flow branches based on user role. Students move to registration and course materials, lecturers move to course management and grading, while administrators access system configuration and monitoring features. The flowchart also captures restrictions like registration deadlines, eligibility checks and access control.

The Data Flow Diagram in Figure 3 shows how data moves through the system. Users submit information through frontend forms, the business layer validates the requests and then routes the data to the appropriate database section such as enrollment records, uploaded materials, grades or quiz attempts. Lecturers and administrators can then retrieve reports and summaries from the same centralized database through their dashboards.

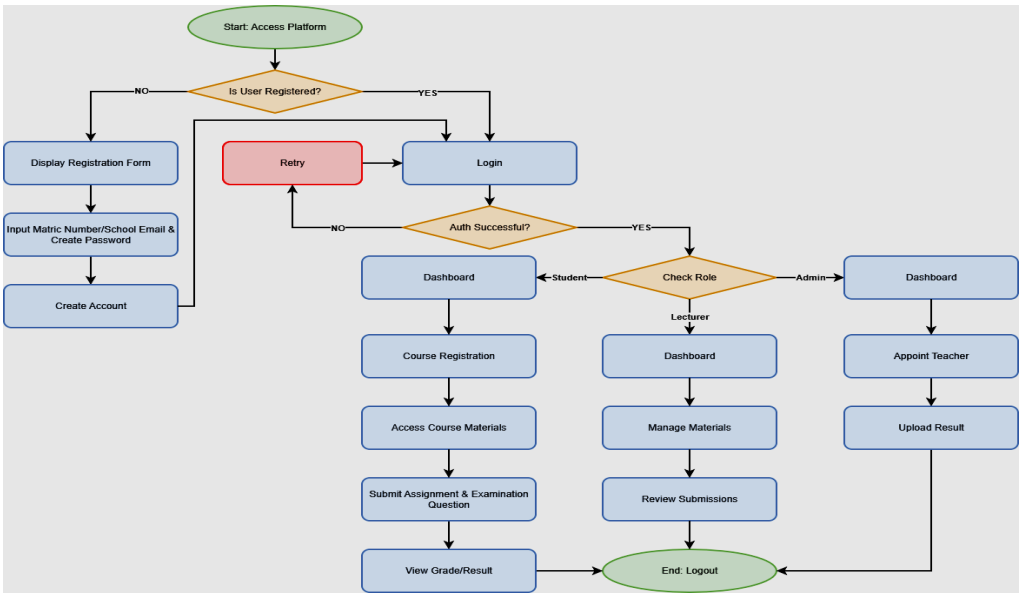


Fig 2. System Flowchart

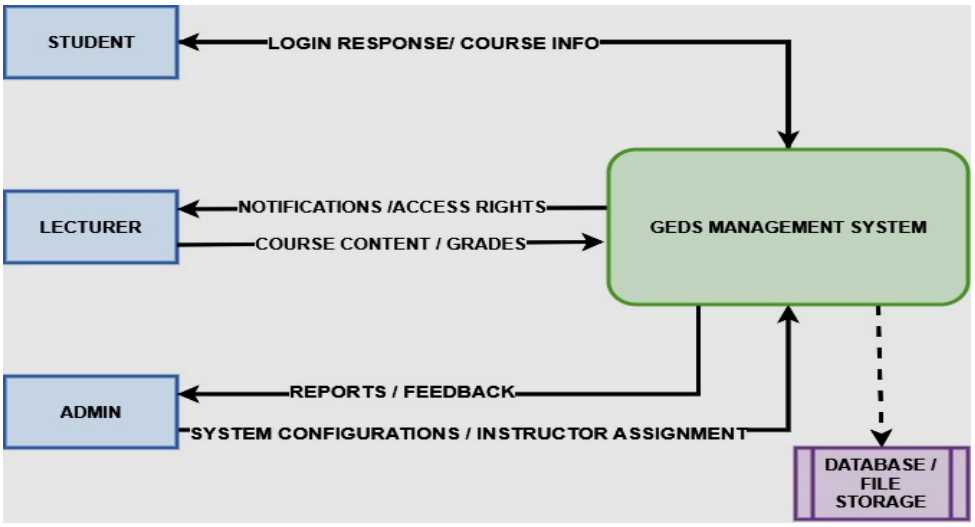


Fig 3. Data Flow Diagram

Database Design

The database schema contains 15 entities including Admin, Lecturer, Student, Course, Semester, Session, Enrolment, Assessment, Grade, Quiz, Question, Choice, QuizAttempt, NewsEvent and Material. The Enrolment entity acts as the central link between students and courses and connects related records like grades and assessments. Quizzes are also linked to courses, questions, choices and quiz attempts within the database structure.

The database was normalized to reduce duplication and improve consistency across the system. Referential integrity was enforced so updates in one entity reflect properly across related records without leaving incomplete data behind. SQLite was chosen because it works smoothly with Django ORM and keeps the system self-contained without needing an external database server. Figure 4 shows the Entity Relationship Diagram.

Regarding data security, Django's ORM enforces parameterised queries throughout the application layer, providing structural protection against SQL injection at the framework level. User credentials are stored as salted hashes using Django's built-in PBKDF2 algorithm, and session data is managed through encrypted cookie mechanisms. Database backups can be scheduled as automated copies of the SQLite file, given its single-file architecture. Regarding scalability, SQLite is well-suited to the current deployment context, where GEDs administrative workflows follow structured semester-based cycles rather than sustaining continuous high-frequency concurrent write operations. Should the platform be extended to serve multiple institutions simultaneously or if concurrent user loads increase significantly, migration to a client-server database engine such as PostgreSQL would be the recommended path forward.

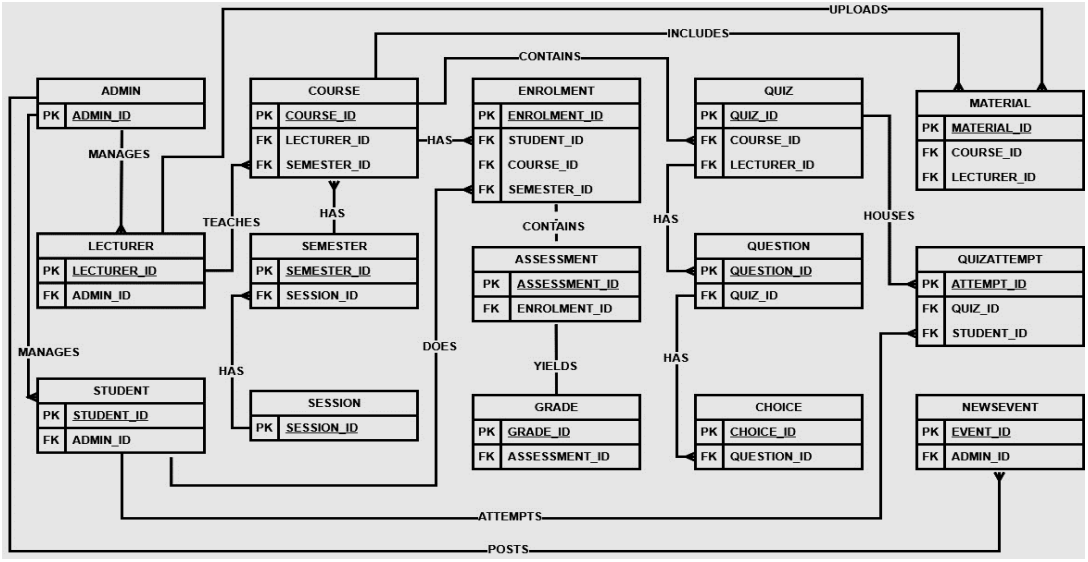


Fig 4. Entity Relationship Diagram

Use Case Summary

The system has three main users: Administrator, Lecturer and Student. The major use cases include account registration, GEDs course enrollment, uploading and downloading materials, submitting and viewing assessments and results, generating reports and managing users and course settings. Figure 5 shows the Use Case Diagram and the access boundaries between each user type.

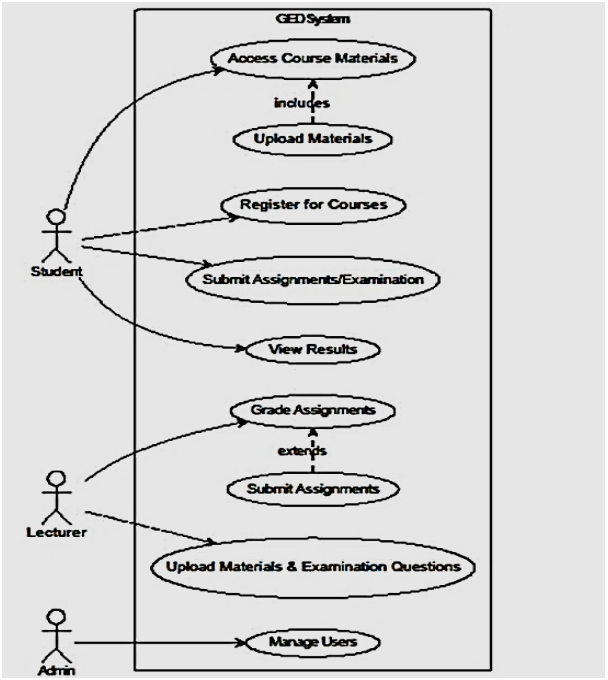


Fig 5. Use Case Diagram

Development Tools

The system was developed using Python 3.x, Django and Django REST Framework for the backend, while HTML5, CSS3 and JavaScript were used for the frontend. SQLite through Django ORM handled data storage. Visual Studio Code was used as the IDE and draw.io was used for the diagrams and workflows before development started.

Implementation and Results

The GSMS interfaces were designed around the principle of role-based simplicity, ensuring that each user group encounters only the functions directly relevant to their operational responsibilities. This approach minimises cognitive load, reduces the risk of accidental modification of records outside a user's assigned scope, and aligns with established usability guidelines that recommend progressive disclosure of system features. Navigation across all three user interfaces follows a consistent structural pattern, reducing the learning curve for first-time users regardless of technical background. Input validation is enforced at both the frontend and backend layers, providing users with immediate error feedback before any submission reaches the database. These design decisions collectively support the perceived ease of use that Davis (1989) identifies through the Technology Acceptance Model as a primary determinant of successful technology adoption, and are reflected in the 100% task completion rate recorded across all user roles during user acceptance testing.

User Authentication and Access Control

The system uses Django's built-in authentication system for login and session management. Users are grouped into administrator, lecturer and student roles, with each role only able to access its assigned pages and modules. Login sessions are maintained securely using tokens and cookies, while administrators can also restrict sensitive configuration panels to admin-only access. Figure 6 shows the login interface.

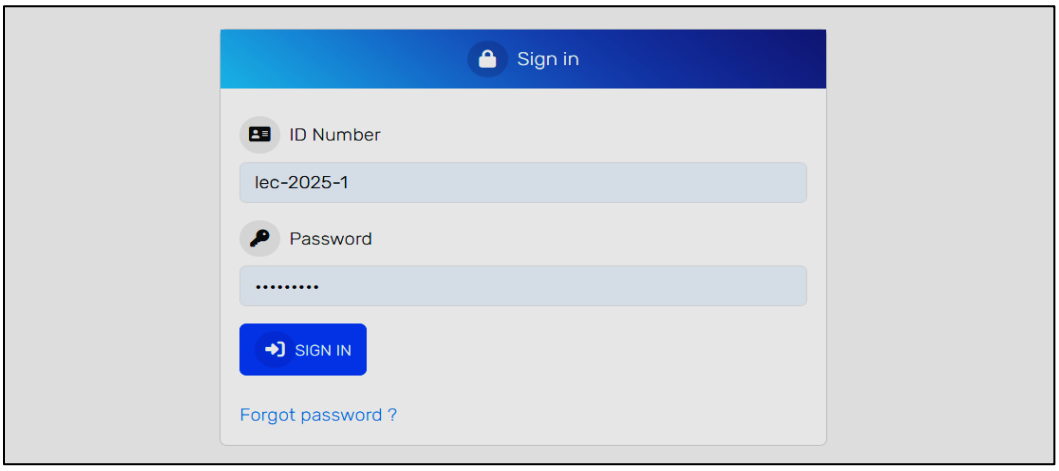


Fig 6. User Authentication Interface

Administrator Module

The administrator module has the highest level of access in the system. After login, administrators see a dashboard showing account statistics and recent system activities like course updates and uploaded materials. The admin panel allows administrators to manage user accounts, create programs and GEDs courses, assign lecturers, define sessions and semesters, monitor examination records and track quiz performance. Figures 7 and 8 show the dashboard and course management interfaces.

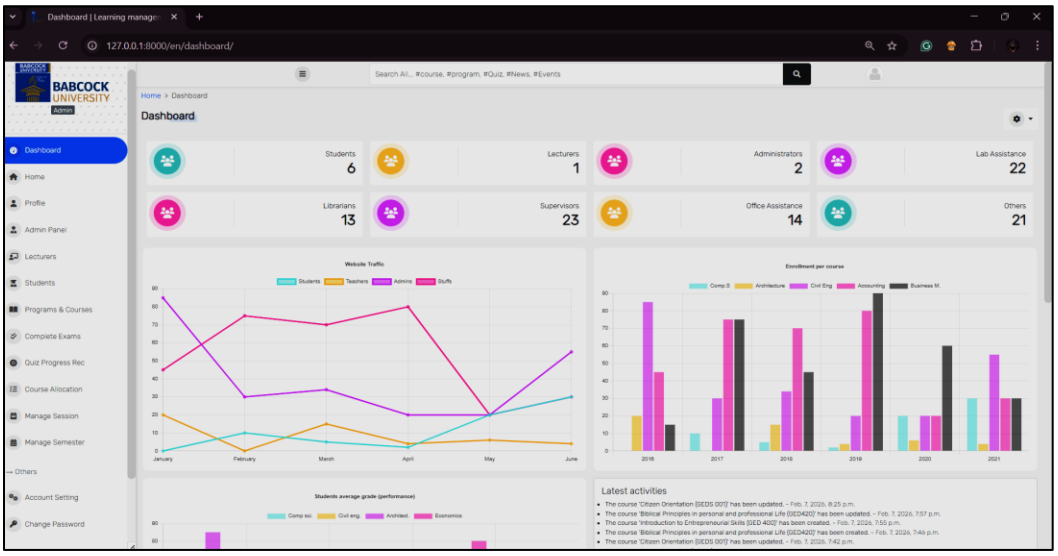


Fig 7. Administrator Dashboard

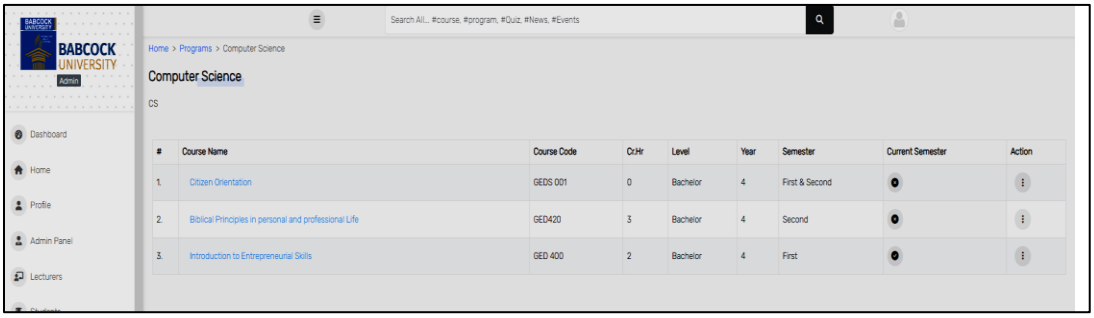


Fig 8. Program and Course Management Interface

Student Module

Students access a homepage containing News and Events updates so important announcements are easier to see instead of relying on unofficial group chats. Through the student panel, users can register courses, download materials, check assessments, view computed grades and take quizzes directly from the browser. Student records are isolated so users cannot access another student's information. Figures 9 and 10 show the materials and quiz interfaces.

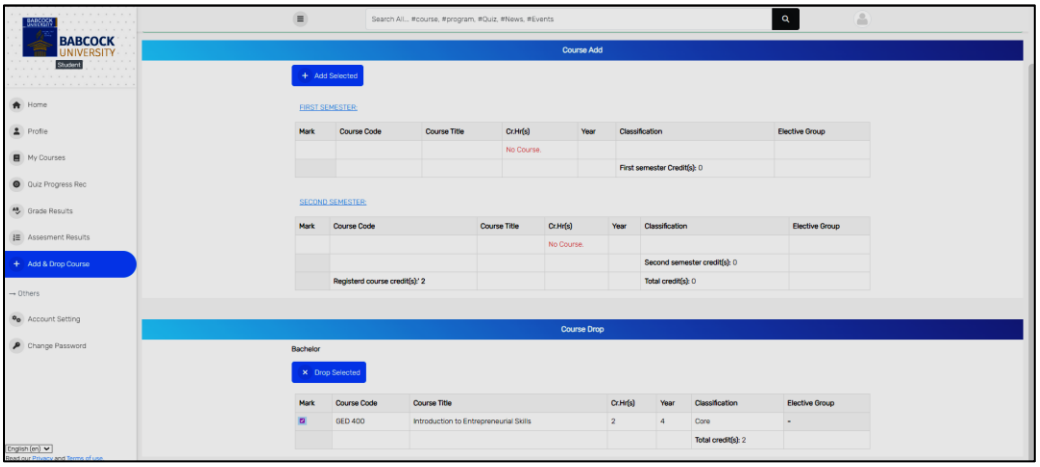


Fig 9. Student Course Materials and Downloads Interface

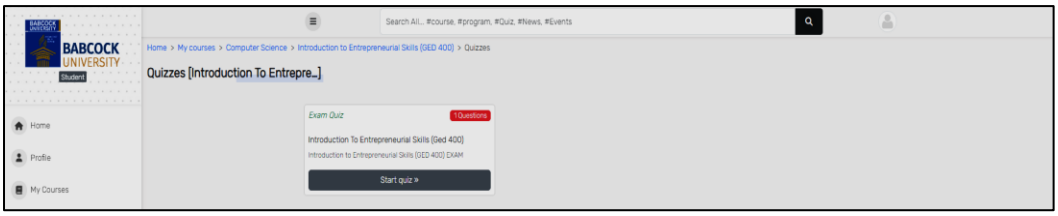


Fig 10. Quiz Progress Record Interface

Lecturer Module

Lecturers have their own dashboard showing assigned courses and management tools. Access is limited only to courses assigned by administrators to avoid accidental modification of another lecturer's records. Lecturers can upload materials, manage assessments, enter scores and review automatically generated grades before final submission. Figures 11 and 12 show the lecturer interfaces.

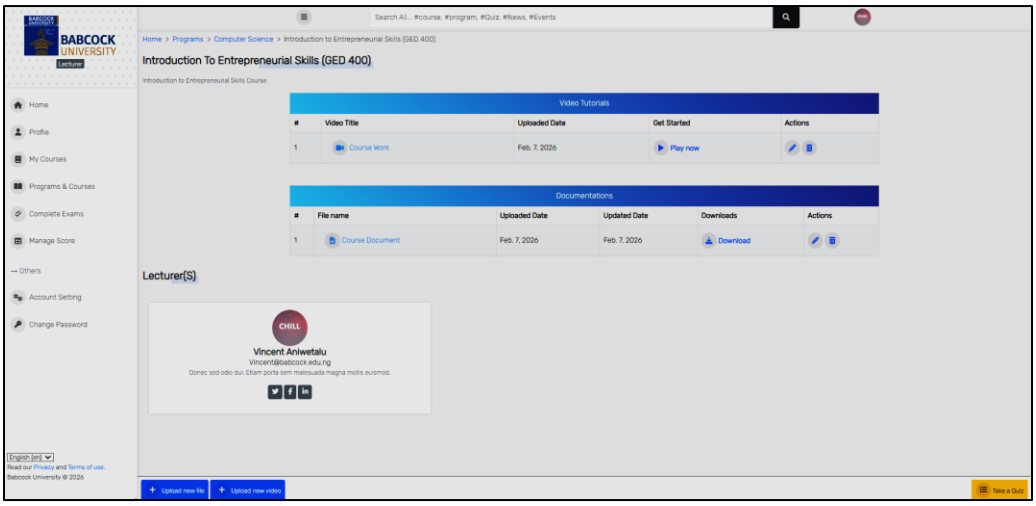


Fig 11. Lecturer Course and Material Management Interface



Fig 12. Student Assessment Score Entry Interface

Assessment and Grading Module

The grading module automates score computation. Lecturers enter raw scores for attendance, assignments, quizzes and exams, while the system calculates totals, grade points, letter grades and comments automatically based on the institution's grading structure. This reduces calculation mistakes and saves lecturers time during result processing. Figure 13 shows the grading interface.

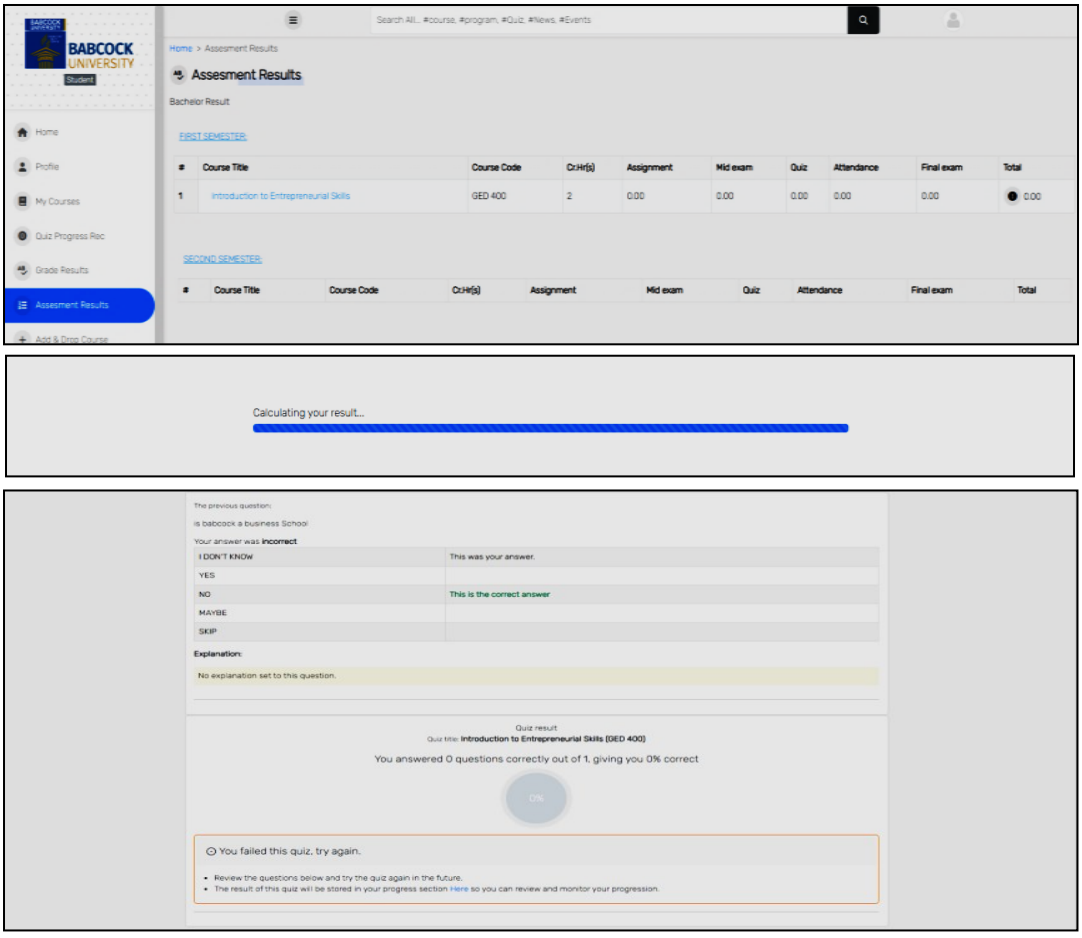


Fig 13. Assessment and Grading Module Interface

Quiz and Examination Module

The quiz module supports multiple-choice and True/False questions with settings for randomization, pass marks and attempt limits. Students take quizzes directly in the browser while all attempts are recorded and organized by session and semester. Essay questions can also be graded manually where automated marking is not suitable. Figure 14 shows the quiz management interface.

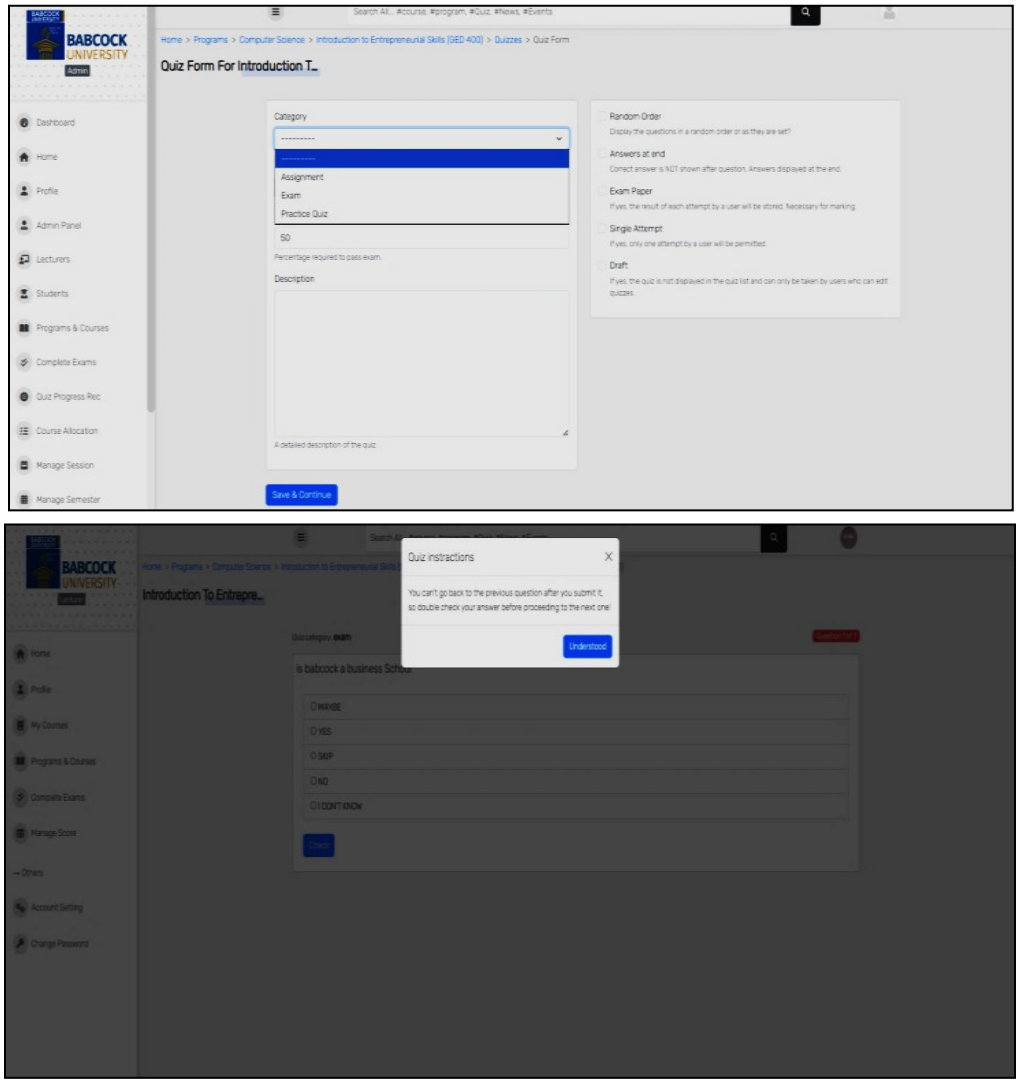


Fig 14. Quiz and Examination Management Interface

Reporting and PDF Generation

Students can generate and download PDF copies of registration slips and result sheets directly from the platform without going through administrators. The PDFs are generated dynamically from the database so the information always stays updated. This also reduces the workload of manual result processing for staff. Figure 15 shows a sample PDF output.

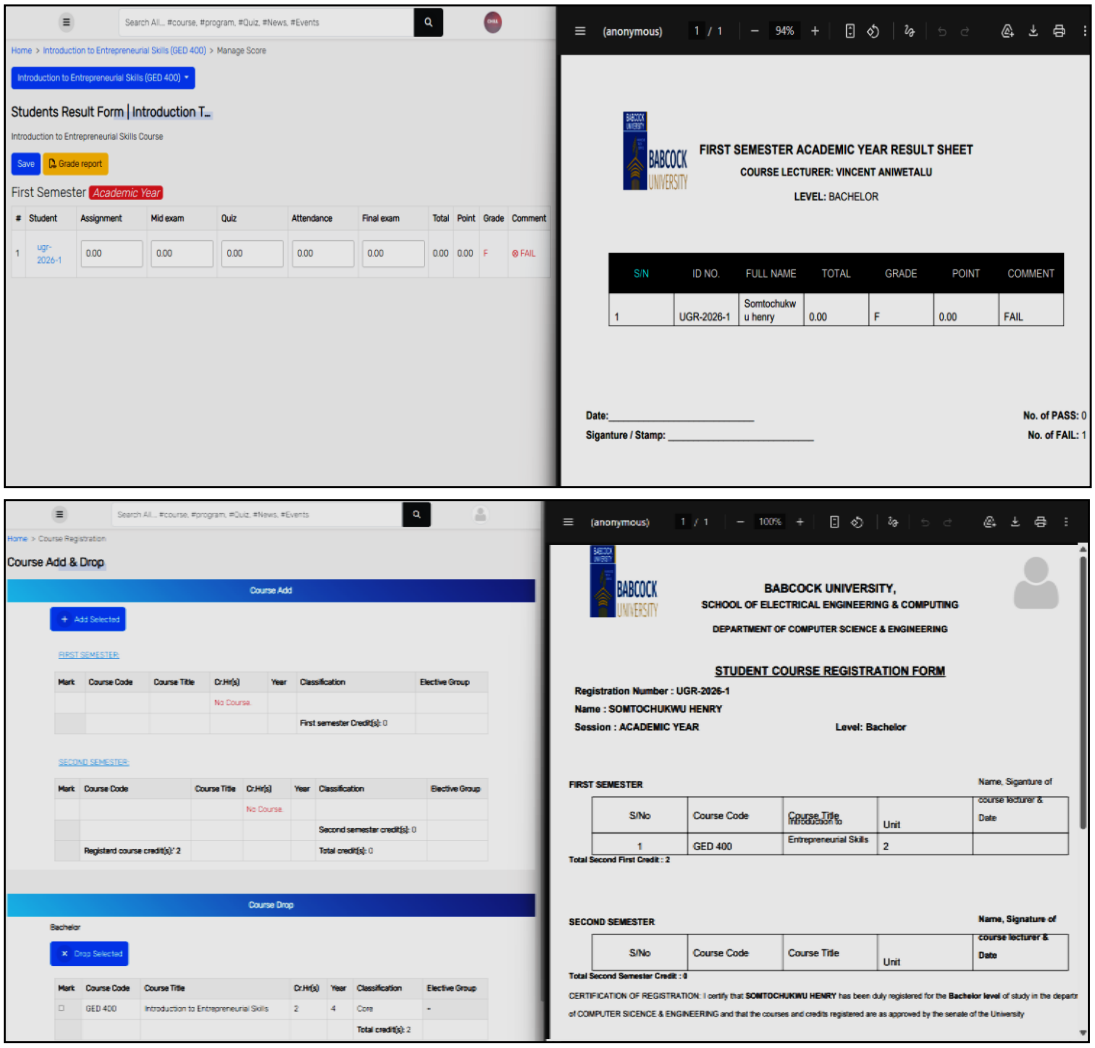


Fig 15. PDF Report Generation and Sample Output

System Testing and Validation

Testing was carried out in three stages. First, unit and integration testing checked individual modules and APIs. Second, full workflow testing covered registration, uploads, quizzes, grading and PDF generation. Finally, user acceptance testing involved students, lecturers and administrators performing their normal tasks within the live system. During testing, grade computation remained accurate, role-based access restrictions worked correctly and no critical system failures were recorded.

A total of 20 functional test cases were designed and executed across seven system modules: Authentication, Administrator Module, Student Module, Lecturer Module, Grading Module, Quiz Module, and Reporting Module. Table 3 presents the complete test case results.

Table 3. GSMS Functional Test Case Results

Test Case ID	Module	Test Description	Input Data	Expected Output	Actual Output	Status
TC001	Authentication	Verify user login with valid credentials	Correct username & password	User logged in and redirected to dashboard	Successful login and dashboard displayed	Pass
TC002	Authentication	Verify login with invalid credentials	Incorrect username/password	Error message displayed	Error message displayed correctly	Pass
TC003	Authentication	Verify role-based access restriction	Student accessing admin page	Access denied message	Access denied enforced	Pass
TC004	Admin Module	Create new user account	Valid user details	New user created	User created successfully	Pass
TC005	Admin Module	Update user details	Modified user info	User updated	Update successful	Pass
TC006	Admin Module	Delete user account	Select user	User removed	User deleted successfully	Pass
TC007	Admin Module	Create course and program	Course title, code	Course created	Course displayed successfully	Pass
TC008	Admin Module	Assign lecturer	Lecturer + course	Lecturer assigned	Assignment successful	Pass
TC009	Student Module	View registered courses	Student login	Courses displayed	Courses displayed correctly	Pass
TC010	Student Module	Add course	Course selection	Course added	Course added successfully	Pass
TC011	Student Module	Drop course	Selected course	Course removed	Course dropped successfully	Pass
TC012	Student Module	View results	Student records	Grades displayed	Results displayed correctly	Pass
TC013	Lecturer Module	Upload scores	Student scores	Scores saved	Scores recorded successfully	Pass
TC014	Lecturer Module	Validate results	Completed assessments	Results finalized	Validation successful	Pass
TC015	Grading Module	Automatic grade computation	Student scores	Correct grades generated	Accurate computation observed	Pass
TC016	Quiz Module	Attempt quiz	Valid attempt	Quiz loads	Quiz executed successfully	Pass
TC017	Quiz Module	Submit quiz	Completed answers	Score calculated	Score displayed correctly	Pass
TC018	Quiz Module	Restrict multiple attempts	Second attempt	Access blocked	Restriction enforced	Pass
TC019	Reporting Module	Generate result PDF	Student data	PDF generated	PDF generated successfully	Pass
TC020	Reporting Module	Generate registration slip	Course data	Slip generated	PDF generated successfully	Pass

All 20 test cases passed successfully, representing a 100% functional pass rate across all system modules. Critical behaviours confirmed include: accurate enforcement of role-based access restrictions across all three user roles; correct automatic computation of weighted totals,

grade points, and letter grades; enforcement of quiz single-attempt restrictions; and successful dynamic PDF generation for both registration slips and result sheets. No functional failures or critical system errors were recorded during the testing process.

User acceptance testing was conducted with 20 participants comprising 15 undergraduate students drawn from multiple departments, two GEDs lecturers, and three administrators who evaluated the system from an operational administrative perspective. Participants were assigned role-specific tasks reflecting their actual operational interactions with GEDs administrative systems. All 20 participants successfully completed their assigned tasks without encountering critical failures, representing a 100% task completion rate across all tested user roles. Feedback gathered iteratively during testing led to one significant improvement: a GEDs lecturer identified the absence of direct lecturer-student communication functionality as a gap in the initial build. This was incorporated in a subsequent development iteration, resulting in the integration of in-app communication channels within the platform.

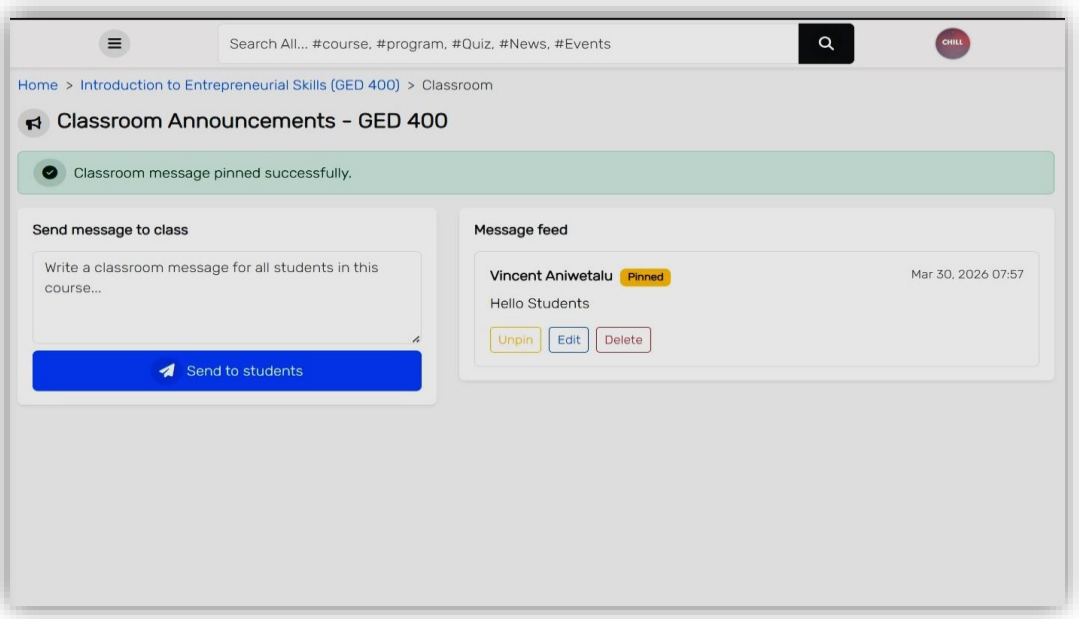


Fig 16. In-App Communication Channel Interface

Discussion

Comparison with Related Systems

The evaluation findings confirm that the GSMS addresses the administrative gaps identified in the problem statement and support a productive comparison with related systems in the literature. Zhang (2024) designed a general education course platform for Chinese universities that similarly emphasises role-differentiated access and centralised resource management, but was developed for a different national curriculum context without the cross-faculty routing demands of the Nigerian GEDs structure. Wong-Fajardo et al. (2021) demonstrated the value

of integrating academic management with LMS functionality at a Peruvian university, achieving unified course planning and real-time updates comparable to those the GSMS provides. The principal distinction is that the GSMS was purpose-built for one programme type rather than designed as a general institutional platform. Ike (2023) employed a Python-based architecture similar to the Django stack used here, confirming the technical viability of this approach for Nigerian university systems, though Ike's implementation was limited to one department. Adedoyin et al. (2023) reported user satisfaction rates exceeding 90% in a comparable Nigerian LMS context; the 100% task completion rate achieved in GSMS user acceptance testing suggests that equivalent outcomes are achievable at full deployment scale.

Addressing the Research Gap

The evaluation results confirm that the GSMS fills the gap identified in the literature review. No reviewed system simultaneously provided cross-departmental data routing, centralised GEDs material distribution, automated role-differentiated grading, and dedicated interfaces for three distinct user groups within a single platform designed for Nigerian GEDs administration. The in-app communication channels integrated in direct response to lecturer feedback further demonstrate the platform's capacity for evidence-driven improvement, directly addressing the communication deficit that Ayanbode et al. (2022) identified as the strongest determinant of student academic satisfaction at Babcock University.

Theoretical Implications

The design and adoption trajectory of the GSMS aligns with the Technology Acceptance Model (Davis, 1989), which positions perceived usefulness and perceived ease of use as the primary determinants of technology adoption. The interfaces were designed to restrict each user role to functions directly relevant to their responsibilities, minimising navigational complexity and maximising functional clarity. The 100% task completion rate across participants with varying levels of technical familiarity provides initial empirical evidence consistent with TAM's predictions regarding the relationship between perceived usability and successful system adoption.

Limitations

Several limitations of the current implementation are acknowledged. SQLite, while sufficient for the current institutional deployment context, may present scalability constraints if the platform is extended to serve multiple institutions or if concurrent write loads increase significantly. The user acceptance testing sample of 20 participants, while representative across roles, was evaluated under controlled conditions rather than full production deployment, which limits the generalisability of the usability findings. The absence of a mobile-optimised interface restricts accessibility for students who access online platforms primarily through smartphones. These limitations represent explicit directions for future development.

Summary, Conclusion and Recommendation

This paper presented the design and implementation of a Specialized General Studies Management System developed specifically for GEDs administration at Babcock University. The main idea behind the system was that GEDs administration already has its own structure and challenges, so trying to force a general university platform to handle everything will always leave gaps somewhere. Because of that, the GSMS was designed as a dedicated platform that brings registration, course materials, grading, mock exams and result processing into one centralized system instead of the scattered methods students and staff currently depend on.

During testing, the system performed reliably under normal operational conditions and maintained proper separation between different user roles and records. The automated grading feature also helped reduce the calculation mistakes that usually happen when lecturers process large volumes of results manually, while the PDF generation feature reduced some of the repetitive administrative work involved in result handling and documentation. Overall, the system addressed several weaknesses that the current UMIS structure does not fully solve when it comes to GEDs administration.

Some recommendations were also identified for future improvements. The first is integrating the GSMS directly with UMIS so enrollment records and student information can stay synchronized automatically without repeated data entry across both systems. Another important improvement would be optimizing the platform more for mobile devices since most Nigerian students access online systems mainly through their phones. The system architecture is also flexible enough that it can later be expanded to support other cross-departmental programs without major structural changes. Finally, monitoring how lecturers continue to use the automated grading tools over time could help guide future updates based on actual usage patterns and operational needs within the university. Future evaluation studies should also be conducted with larger participant samples under live production conditions to generate more robust usability and performance data. Cloud deployment on a platform such as AWS or Google Cloud would eliminate local server infrastructure constraints and improve availability during peak academic periods. Security enhancements including HTTPS enforcement, token-based API authentication, and scheduled automated database backups should be implemented prior to full institutional deployment. Finally, comparative performance testing against established platforms such as Moodle and Blackboard would provide empirical benchmarks that strengthen the case for the GSMS as a specialised solution for GEDs administration.

References

Adedoyin, A. E., Enebe, F. O., Oyekunle, R. A., & Balogun, N. A. (2023). Design and implementation of an online teaching and learning management system. *FUDMA Journal of Sciences*.

Alia, A. A. (2022). The analysis of a learning management system from a design and development perspective. *International Journal of Information and Education Technology, 12*(4), 280–289.

Ayanbode, O. F., Fagbe, A., Owolabi, R., Oladipo, S., & Ewulo, O. R. (2022). Students' interactions, satisfaction, and perceived progress in an online class: Empirical evidence from Babcock University Nigeria. *Cogent Education*.

Bayode, O. A. (2022). Design and implementation of school management system. National Open University of Nigeria.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly, 13*(3), 319–340.

Decuypere, M., Grimaldi, E., & Landri, P. (2021). Introduction: Critical studies of digital education platforms. *Critical Studies in Education*.

Fibriasari, H., Andayani, W., Putri, T. T. A., & Harianja, N. (2023). Learning management system now and in the future: Study case from the Indonesian university students. *International Journal of Information and Education Technology*.

Huang, H. (2024). Design and implementation of student management system of integrated programmable device programming system. *Scientific Reports, 14*, 11873.

Husain, N. A. (2023). A review on academic management system: Bridging programme-SPACE UTM. *Jurnal Teknikal & Kajian Sosial (JUTEKS), 21*(2).

Ike, I. (2023). Design and implementation of a student information system. Federal University Otuoke.

Liu, S. (2023). Design and implementation of management system of university faculty team construction in Internet Plus. In *Proceedings of 2023 4th International Conference on Education, Knowledge and Information Management*. Atlantis Press.

National Universities Commission. (2023). *Core curriculum and minimum academic standards (CCMAS) for Nigerian universities*. NUC.

Rohit, G. P., Grabusts, P., & Teilans, A. (2021). E-learning: Developing tomorrow's education. In *Proceedings of the 13th International Scientific and Practical Conference*. Rezekne Academy of Technologies.

Sofi-Karim, M., Bali, A. O., & Rached, K. (2022). Online education via media platforms and applications as an innovative teaching method. *Education and Information Technologies*.

Umar Ong, I. (2021). An information system design product theory for student academic self-service university management system. *International Journal of Software Engineering and Computer Systems, 7*(1), 47–56.

Vlasenko, K. V., Volkov, S. V., Lovianova, I. V., Chumak, O. O., & Sitak, I. V. (2021). Management of online platform development and support process. *CTE Workshop Proceedings, 8*, 1–14.

Wang, Q. (2023). Design and implementation of college student information management system. In *Proceedings of IEIT-23*. Atlantis Press.

Wang, Y. (2023). Design and development of information-based system for quality evaluation of class collective in colleges and universities. *International Journal of New Developments in Education, 5*(5).

Wong-Fajardo, E. M., Saavedra-Sánchez, H., Mendoza-Rodas, M., & Hernández-Vásquez, R. (2021). Design and implementation of an integrated academic management model with LMS: A Peruvian private university study case. *CEUR Workshop Proceedings*.

Xiong, Z. (2023). Design and implementation of student information management system based on web. *International Journal of New Developments in Education, 5*(13), 30–35.

Zhang, Z. (2024). Design and implementation of general education courses system based on curriculum network. In *Proceedings of the 4th International Conference on Internet, Education, and Information Technology (IEIT 2024)*.