**DEPARTMENT FEEDBACK SYSTEM**

**BY**

**MUHAMMAD, Ndajiya Ahmad**

**22A/UE/BCSX/10014**

**DEPARTMENT OF CYBER SECURITY,**

**NEWGATE UNIVERSITY, MINNA NIGERIA**

**AUGUST, 2025**

**Fly Leaf**

**DEPARTMENT FEEDBACK SYSTEM**

**BY**

**MUHAMMAD, Ndajiya Ahmad**

**22A/UE/BCSX/10014**

**A PROJECT SUBMITTED TO THE DEPARTMENT OF CYBER SECURITY, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELORS OF SCIENCE (BSc) IN CYBER SECURITY, NEWGATE UNIVERSITY MINNA**

**AUGUST, 2025**

# **DECLARATION**

I declare that the work in this Project Report entitled “Department Feedback System” was carried out by me in the Department of “Cyber Security”. The information derived from the literature were duly acknowledged in the text and a list of references provided. No part of this Project Report was previously presented for another degree or diploma at this or any other Institution.

MUHAMMAD, Ndajiya Ahmad

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of student Signature Date

# **CERTIFICATION**

This Project Report entitled “Department Feedback System” by MUHAMMAD, Ndajiya Ahmad meets the regulations governing the award of the degree of Bachelors of Science (BSc) in Cyber Security of the Newgate University Minna and is approved for its contribution to knowledge and literary presentation.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_**

Chairman, Supervisory Committee Signature Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_**

Member, Supervisory Committee Signature Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_**

Member, Supervisory Committee Signature Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_**

Head of Department Signature Date

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_**

Dean, Faculty of Computing and Signature Date

Information Technology

# **DEDICATION**

I dedicate this developmental project to my Dad Mr. Muhammad who in sweat and blood afford me this level of education.

# **ACKNOWLEDGEMENT**

My sincere appreciation goes to Almighty Allah, the most merciful, the giver of life for His grace, favor and unrelenting support throughout this research work.

The completion of this research project would not have been possible without the support of some very important people. First, I must express my utmost gratitude and respect to my supervisor Dr Jamilu Isah. His guidance, knowledge and assistance through this process were invaluable. I would like to appreciate the Dean of faculty of Faculty of Computing and Information Technology in the person of Dr Rahmat Ndaminin and the Head of Department of Cyber Security in the person of Dr Adamu Ibrahim for allowing me the enabling environment and facilities to carry out this research project. A special appreciation to Dr David Saba for his enormous contribution.

To my friend and brother, Sayuti Abdul-rahman Yahya, I say thank you for all you do. I wouldn’t have been able to do anything without my family, to whom I owe everything. Their prayer and support both financially and otherwise have always sustain me

# **ABSTRACT**

*The collection of reliable student feedback in higher education is often undermined by systems that fail to reconcile the conflicting needs for respondent verification and anonymity. This study addressed this problem through the design, development, and implementation of a secure, department-level web application for Newgate University Minna. The project employed an Incremental Software Development model and the Model-View-Template (MVT) architectural pattern, utilizing the Django framework for backend development. The core of the methodology is a novel approach that uses a pseudonymous SHA-256 hashing mechanism to technically decouple a verified student's identity from their feedback, thus guaranteeing both authenticity and anonymity. The primary result is a fully functional prototype featuring role-based access for Students, Lecturers, and Administrators. Key implemented features include a secure authentication module, a dynamic and anonymous survey submission interface, and a comprehensive administrative dashboard with data analytics, visualisation, and reporting capabilities (CSV/PDF export). The study concludes that this system successfully resolves the verification-anonymity paradox, providing a robust solution superior to generic online tools. The significance of this finding is the creation of a trustworthy channel that yields high-integrity data, empowering evidence-based decision-making for academic quality assurance and fostering a culture of continuous improvement.*

# **TABLE OF CONTENTS**

[DECLARATION iv](#_Toc206395234)

[CERTIFICATION v](#_Toc206395235)

[DEDICATION vi](#_Toc206395236)

[ACKNOWLEDGEMENT vii](#_Toc206395237)

[ABSTRACT viii](#_Toc206395238)

[TABLE OF CONTENTS ix](#_Toc206395239)

[LIST OF TABLES xiii](#_Toc206395240)

[LIST OF FIGURES xiv](#_Toc206395241)

[CHAPTER ONE 1](#_Toc206395242)

[INTRODUCTION 1](#_Toc206395243)

[1.1 Background to the Study 1](#_Toc206395244)

[1.2 Statement of the Problem 3](#_Toc206395245)

[1.3 Aim and Objectives of the Study 3](#_Toc206395246)

[1.4 Justification of the Study 4](#_Toc206395247)

[1.5 Scope of the Study 5](#_Toc206395248)

[1.6 Limitations of the Study 5](#_Toc206395249)

[1.7 Definition of Terms 6](#_Toc206395250)

[CHAPTER TWO 9](#_Toc206395251)

[LITERATURE REVIEW 9](#_Toc206395252)

[2.1 Conceptual Framework 9](#_Toc206395253)

[2.1.1 Principles of Secure Web Application Development 10](#_Toc206395254)

[2.1.2 Anonymity and Identity Verification in Online Systems 12](#_Toc206395255)

[2.2 Theoretical Framework 13](#_Toc206395256)

[2.2.1 The Model-View-Template (MVT) Architectural Pattern 14](#_Toc206395257)

[2.2.2 The Incremental Software Development Model 15](#_Toc206395258)

[2.3 Review of Existing Systems 16](#_Toc206395259)

[2.3.1 Generic Online Survey Tools (e.g., Google Forms, SurveyMonkey) 17](#_Toc206395260)

[2.3.2 Existing Institutional Feedback Portals 18](#_Toc206395261)

[CHAPTER THREE 20](#_Toc206395262)

[SYSTEM ANALYSIS AND DESIGN 20](#_Toc206395263)

[3.1 System Development Methodology 20](#_Toc206395264)

[3.2 Analysis of the Existing System 21](#_Toc206395265)

[3.3 Analysis of the Proposed System 22](#_Toc206395266)

[3.3.1 Functional Requirements 22](#_Toc206395267)

[3.3.2 Non-Functional Requirements 22](#_Toc206395268)

[3.4 System Design 23](#_Toc206395269)

[3.4.1 System Architectural Design 24](#_Toc206395270)

[3.4.2 Use Case Analysis and Design 24](#_Toc206395271)

[3.4.3 Database Design 26](#_Toc206395272)

[3.4.4 User Interface (UI) and User Experience (UX) Design 27](#_Toc206395273)

[CHAPTER FOUR 29](#_Toc206395274)

[SYSTEM IMPLEMENTATION AND TESTING 29](#_Toc206395275)

[4.1 Development Environment and Tools 29](#_Toc206395276)

[4.2 System Implementation 30](#_Toc206395277)

[4.2.1 Implementation of Student Verification and Authentication Module 30](#_Toc206395278)

[4.2.2 Implementation of Anonymous Feedback Submission Module 32](#_Toc206395279)

[4.2.3 Implementation of the Administrative Dashboard and Analytics 34](#_Toc206395280)

[4.2.4 Implementation of the User Interface 37](#_Toc206395281)

[Structure and Templating 37](#_Toc206395282)

[Styling and Design Aesthetic 38](#_Toc206395283)

[CHAPTER FIVE 39](#_Toc206395284)

[SUMMARY, CONCLUSION, AND RECOMMENDATIONS 39](#_Toc206395285)

[5.1 Summary 39](#_Toc206395286)

[5.2 Conclusion 40](#_Toc206395287)

[5.3 Recommendations 41](#_Toc206395288)

[REFERENCES 43](#_Toc206395289)

# **LIST OF TABLES**

**Table Page**

[4.1: Technology Stack And System Requirements (Developed By The Author) 29](#_Toc206395291)

# **LIST OF FIGURES**

**Figure Page**

[3.1: System Architecture Diagram 24](#_Toc206396083)

[3.2: System Flow Diagram 25](#_Toc206396084)

[4.13: User Interface For The Student Login Page, Where Students Enter Their Credentials For Verification Before Being Granted Access To Submit Feedback 31](#_Toc206396085)

[4.24:Dashboard Showing Students Portal On The System 32](#_Toc206396086)

[4.35:The User Interface For The Feedback Submission Form 34](#_Toc206396087)

[4.4 6:The Main Dashboard For An Administrator 35](#_Toc206396088)

[4.57:The Central Control Panel For Users With The 'Administrator' Role 35](#_Toc206396089)

[4.68:The Lecturer Dashboard View 36](#_Toc206396090)

[4.79:Survey Feedback Access And Reporting Features 36](#_Toc206396091)

# **CHAPTER ONE**

## **INTRODUCTION**

## **1.1 Background to the Study**

The pursuit of quality assurance and continuous improvement in the global landscape of higher education has become a central objective for institutions seeking to maintain relevance and excellence. A critical component of this endeavour is the systematic collection and analysis of students’ feedback. It is widely acknowledged that feedback from students provides invaluable insights into teaching effectiveness, curriculum relevance, and the overall quality of departmental services (Stones, 2020). Universities all over the world leverage this data to inform pedagogical strategies, guide academic programme reviews, and enhance the student learning experience, thereby fostering an environment of accountability and academic growth.

The evolution of technology has significantly transformed the mechanisms for gathering feedback from student. Historically, educational institutions relied on paper-based questionnaires and suggestion boxes, which were often inefficient, costly to administer, and presented significant challenges in data collation and analysis (Lee & Chen, 2019). The digital era however, has ushered in the widespread adoption of online platforms, which promise greater efficiency, wider reach, and more sophisticated data processing capabilities. This shift has led many university departments to utilise generic, third-party online survey tools such as Google Forms and SurveyMonkey, primarily due to their accessibility and ease of use.

However, the convenience of these generic tools often comes at the cost of data integrity and security. A primary challenge is the lack of a robust identity verification mechanism, which leaves the feedback process vulnerable to manipulation, such as spam and multiple submissions from a single individual. Conversely, when these platforms require personal identifiers like a Google account, they compromise the anonymity that is crucial for eliciting honest and critical feedback. Students may fear academic or social reprisal for negative comments, leading to self-censorship and diluted, less useful data (Davies & Miller, 2021). Furthermore, the reliance on external platforms raises significant data privacy and sovereignty concerns, as sensitive institutional data is stored and managed by third-party corporations, potentially conflicting with university data protection policies.

Nigerian educational environment is not exempted from these challenges. While many universities are increasingly embracing information and communication technologies (ICT), the implementation of effective digital feedback systems is still a developing area. The cultural and hierarchical dynamics within some academic environments can heighten student reluctance to provide candid feedback for fear of victimisation (Okonjo & Adebayo, 2022). This underscores a more urgent need for a system that can simultaneously guarantee the authenticity of the respondent and protect their anonymity. The limitations of off-the-shelf solutions mean that many Nigerian university departments are either grappling with unreliable data from insecure platforms or suffering from low response rates due to a lack of student trust in the process.

Therefore, a clear gap exists for a bespoke feedback system designed specifically for the academic departmental context. Such a system must resolve the conflicting demands of verification and anonymity, ensuring that feedback is both authentic and candid. By developing a secure, department-level web application, control over the feedback process and its associated data is returned to the institution. This study is therefore motivated by the need to engineer a solution that addresses the security, privacy, and integrity flaws inherent in generic feedback tools, thereby creating a trustworthy and effective channel for improving academic quality at the departmental level.

## **1.2 Statement of the Problem**

The current system of feedback collection in many institutions across Nigeria relies heavily on manual, face-to-face interactions. Students must have to meet departmental staff or administrative representatives to express their concerns, which, even though encourages direct communication, still goes a long way to presents several challenges. These include inefficiencies like delayed responses, loss of information, and barriers for students who may feel uncomfortable discussing their complaints publicly (Ayankemi, Opeyemi, Abiye, and Sunday, 2024) (Amrina, Aslam, Islam, Sutapa, Kurniawansyah, Abdullah and Ardian, 2024). Furthermore, students often submit complaints using various communication channels such as email, response surveys, and social media, which makes it difficult for institutions to track and manage the information effectively (Tshotlego and Chukwuere, 2024). This fragmented, rigid approach hampers the university's capacity to systematically analyze recurring issues, further complicating addressing student concerns.

## **1.3 Aim and Objectives of the Study**

The aim of this project is to design, develop, and implement a secure, web-based feedback system for university departments of Newgate University Minna with the following objectives:

i. To review existing student feedback systems and literature on secure web development.

ii. To design the architecture, database schema, and user interface for a secure web application that facilitates verified but anonymous feedback submission, based on the Model-View-Template (MVT) architectural pattern.

iii. To develop the web application using the Django framework, implementing core features.

iv. To evaluate the performance, security, and usability of the implemented system through a series of tests.

## **1.4 Justification of the Study**

The significant of this project can be viewed from four different perspectives. Firstly, for Departmental Administrators and Management, the system will provide a tool for making informed, evidence-based decisions. The current feedback mechanisms, normally yield data of questionable integrity due to the inability to prevent spam, duplicate submissions, or input from unauthorised individuals. But this project will will deliver a system that produces verified, reliable data, allowing administrators to confidently identify genuine trends in course delivery, lecturer performance, and student services.

Secondly, the project is crucial for fostering a culture of trust and open communication among Students. One major barrier to obtaining honest feedback is the fear of negative consequences. This system's core design principle combining mandatory identity verification with guaranteed submission anonymity directly addresses this fear.

Thirdly, the system offers significant value to Lecturers and Academic Staff. The aggregated and anonymised feedback serves as a constructive tool for professional development. Instead of ambiguous or potentially biased comments, lecturers can receive structured, authentic feedback specific to their courses and teaching methods.

Finally, this study serves as a practical blueprint for Institutional Technological Advancement. This project demonstrates a viable alternative to insecure, off-the-shelf products. It establishes a model for data governance where sensitive institutional information remains within the university's control.

## **1.5 Scope of the Study**

The scope of this study is focused on the design, development, and implementation of a self-contained, secure web application for collecting and analysing student feedback at the departmental level. The system’s core functionality will encompass a student-facing interface for identity verification and anonymous submission, with built-in logic to prevent duplicate entries per survey. It will also feature a secure, role-based administrative dashboard, providing Department Administrators with tools to view summary reports, filter feedback by parameters such as course, lecturer, or semester, and visualise trends through basic charts and tables. The technical scope involves a full-stack development approach using a Django backend, an SQLite database, and a responsive frontend built with HTML, CSS, JavaScript, and Bootstrap, adhering to a clean, modern design aesthetic. The application is designed to function as a standalone prototype, proving the concept of a verified-yet-anonymous feedback channel.

## **1.6 Limitations of the Study**

It is important to define the project's limitations to maintain a clear focus. The system is not intended to be a comprehensive student information system; it will not handle academic records, course registration, or any other function of a university's main student portal. The application will not include real-time communication features, such as chat or support tickets, nor will it collect or process highly sensitive personal data beyond what is necessary for initial identity verification. Furthermore, this project is designed as a department-level prototype and is not scaled for an enterprise-wide, university-level deployment, which would require a more robust database solution than SQLite and integration with central university authentication systems. Its purpose is strictly limited to serving as a secure and reliable channel for feedback collection and analysis, rather than replacing existing institutional communication platforms.

## **1.7 Definition of Terms**

For the purpose of clarity and to establish a common frame of reference, the following key terms are defined as they are used within the context of this study:

i. Anonymity: Refers to the state where the identity of a student who has submitted feedback is not known and cannot be linked back to their submission. In this system, anonymity is achieved after the initial identity verification step, ensuring that the submitted data is stored without any personal identifiers.

ii. Authentication: The process of verifying the identity of a user attempting to gain access to the system. This confirms that the user is who they claim to be, typically through credentials like a school ID or a verified email address.

iii. Backend: The server-side of the web application that is not visible to the end-user. It is responsible for all the core logic, including processing requests, interacting with the database, handling user authentication, and serving data to the frontend. In this project, the backend is built using the Django framework.

iv. Dashboard: A graphical user interface on the administrative side of the application that provides an at-a-glance, consolidated view of key feedback data. It typically includes summaries, charts, and tables to facilitate data analysis and monitoring.

v. Django: A high-level Python web framework that enables the rapid development of secure and maintainable websites. It follows the Model-View-Template (MVT) architectural pattern, providing a structured environment for building the application's backend.

vi. Frontend: The client-side of the web application that users directly see and interact with in their web browser. It is responsible for the presentation of data and the user interface. In this project, the frontend is developed using HTML, CSS, JavaScript, and the Bootstrap framework.

vii. Model-View-Template (MVT): An architectural pattern used by Django for structuring a web application. The Model manages the application's data structure and interacts with the database. The View handles the business logic, processing user requests and returning responses. The Template is the presentation layer, an HTML file mixed with a templating language to display data to the user.

viii. Responsive Design: An approach to web design that ensures the user interface adapts fluidly to a wide range of devices and screen sizes, from mobile phones and tablets to desktop computers. This provides an optimal viewing and interaction experience for all users.

ix. SQLite: A self-contained, server-less, and transactional SQL database engine. It is lightweight and easy to set up, making it highly suitable for the development, prototyping, and deployment of small- to medium-scale applications like this project.

x. Web Application: A dynamic software programme that runs on a web server and is accessed by users through a web browser over a network. Unlike a static website, it allows for user interaction, data processing, and integration with a database.

# **CHAPTER TWO**

## **LITERATURE REVIEW**

In this section, the existing literature relevant to the development of a secure student feedback system is discussed. The review is structured to build a foundation for the project by examining three key areas. First, it explores the Conceptual Framework, which defines and discusses the core concepts of student feedback systems, secure web development, and the balance between anonymity and verification. Second, it outlines the Theoretical Framework, detailing the architectural patterns and development methodologies that underlies the system's design and implementation. Finally, the chapter presents a Review of Existing Systems, analysing the strengths and weaknesses of current solutions to identify the specific research gap that this project aims to address.

## **2.1 Conceptual Framework**

The conceptual framework establishes the fundamental principles and ideas that inform this study. It focusses on the essential components of the project, beginning with an overview of student feedback systems, followed by an examination of secure development practices and the critical challenge of managing identity in online environments.

The practice of collecting student feedback is a cornerstone of educational improvement and institutional quality assurance in modern higher education. It is predicated on the principle that students, as primary stakeholders in the learning process, possess a unique and valuable perspective on the effectiveness of teaching, the relevance of curricula, and the quality of academic support services (Henderson et al., 2019). This concept, often referred to as 'student voice', has evolved from a simple measure of satisfaction to a critical driver of institutional change and accountability. Effective feedback systems are designed not merely to collect data, but to create a dialogue that empowers students and informs evidence-based decision-making at both departmental and institutional levels (Matthews, 2021).

The transition from traditional paper-based surveys to digital platforms has revolutionised the administration of student feedback. Digital systems offer significant advantages in terms of scalability, rapid data aggregation, and reduced administrative overhead. They facilitate the collection of both quantitative data (e.g., rating scales) and rich qualitative data (e.g., open-ended comments), which together provide a holistic view of the student experience (Richardson, 2020). However, the efficacy of any feedback system, digital or otherwise, is contingent on its ability to generate high-quality, actionable data. This depends heavily on two factors: the rate of student participation and the honesty of the responses. To achieve both, a system must be perceived by students as accessible, straightforward, and, most importantly, trustworthy. This need for trust highlights the critical importance of system design in mitigating potential barriers to participation, such as complex interfaces or concerns about privacy, which directly leads to an examination of the security principles required to build such systems.

### **2.1.1 Principles of Secure Web Application Development**

The development of any web application that handles user data, particularly within an institutional context, must be governed by a robust set of security principles. These principles are not merely technical additions but are foundational to establishing user trust and ensuring data integrity. The Open Web Application Security Project (OWASP), a globally recognised authority, provides a framework for secure development, emphasising a proactive, "security-by-design" approach (OWASP Foundation, 2021). For a student feedback system, the most pertinent principles include secure authentication, data protection, and defence against common vulnerabilities.

Secure authentication is the bedrock of a trusted system. It involves verifying that users are who they claim to be before granting them access. In the context of this project, authentication is critical to prevent unauthorised access and ensure that feedback is submitted only by legitimate students of the department. This principle directly addresses the vulnerability of generic survey tools, which often lack a formal mechanism to tie a submission to a verified institutional identity (Smith & Jones, 2020). Implementing a secure authentication process, such as one based on a unique student ID or a verified university email address, is the first step in guaranteeing the authenticity of the feedback data collected.

Data protection, both in transit and at rest, is another non-negotiable principle. Data in transit refers to information travelling over the network, such as when a student submits a feedback form. This data must be encrypted using protocols like Transport Layer Security (TLS) to prevent interception or eavesdropping. Data at rest refers to information stored in the database. Sensitive data, such as student identifiers used for verification, must be handled with extreme care, employing techniques like hashing for passwords or ensuring that personally identifiable information (PII) is not stored alongside anonymised feedback (Peterson, 2022). This principle ensures that the system complies with data protection regulations and upholds the university's responsibility to safeguard student information.

Finally, the application must be designed to withstand common web vulnerabilities. OWASP identifies critical risks such as Injection attacks (e.g., SQL Injection), where an attacker can interfere with an application's database queries, and Cross-Site Scripting (XSS), where malicious scripts are injected into trusted websites (OWASP Foundation, 2021). Modern web frameworks like Django provide built-in protections against many of these threats. For instance, Django's Object-Relational Mapper (ORM) helps prevent SQL Injection by parameterising queries, and its templating engine automatically escapes variables to mitigate XSS risks. Adhering to these secure coding practices is essential to building a resilient application that protects both its users and the integrity of the data it holds, forming a critical bridge to the next challenge: balancing these security measures with the need for user anonymity.

### **2.1.2 Anonymity and Identity Verification in Online Systems**

The central challenge in designing an effective online feedback system lies in reconciling the seemingly contradictory requirements of identity verification and user anonymity. This dichotomy is the primary reason why generic, off-the-shelf tools often fail in an institutional context. Verification is essential for data integrity, ensuring that feedback is authentic and comes from legitimate stakeholders. Anonymity, on the other hand, is crucial for data quality, encouraging the honest and candid responses necessary for meaningful improvement. Achieving a delicate balance between these two principles is therefore a fundamental design consideration (Brown, 2021).

From an institutional perspective, verification is non-negotiable. It serves as a gatekeeping mechanism to prevent feedback from being manipulated by external actors, spam bots, or individuals making multiple submissions to skew the results. By confirming that a respondent is a registered student within the specific department, the institution can trust that the collected data is a genuine representation of the student experience (Brown, 2021). Without this assurance, the entire feedback dataset becomes unreliable, rendering any subsequent analysis and decision-making fundamentally flawed.

Conversely, from the student's perspective, anonymity is paramount. The inherent power imbalance in the student-lecturer relationship can create a significant fear of academic or social reprisal for negative but honest feedback. This "chilling effect" leads to self-censorship, where students either refrain from participating or provide overly positive, less useful comments (Garcia & Kim, 2022). True anonymity removes this barrier, creating a psychologically safe environment where students feel empowered to voice their genuine opinions, both positive and critical, without fear. It is often the most critical feedback that provides the most valuable insights for pedagogical and administrative improvement.

The resolution to this paradox is not to compromise on either principle but to implement a system architecture that logically separates the two processes. A well-designed system can authenticate a user's identity to grant them a single, temporary "pass" to submit feedback for a specific survey or period. Once authenticated, the user is directed to a submission portal that does not retain their personal identifiers. The system's only record is that the authenticated user has now completed their submission, preventing them from submitting again. The feedback itself is then stored in the database as an independent, anonymous entry. This technical decoupling ensures that while the participant is verified, the submission is anonymous, thus satisfying both institutional and student requirements. This study adopts this core principle as its foundational approach to building a trustworthy feedback system.

## **2.2 Theoretical Framework**

While the conceptual framework defined the core ideas of the project, the theoretical framework specifies the established models and structured approaches used for the system's design and development. This section outlines the architectural pattern that dictates the software's internal structure and the development methodology that governed the project's lifecycle from inception to completion.

### **2.2.1 The Model-View-Template (MVT) Architectural Pattern**

The development of the secure feedback system is architecturally grounded in the Model-View-Template (MVT) pattern. MVT is a software design pattern that separates the representation of information from the user's interaction with it, promoting an organised and modular codebase. It is the core architecture of the Django framework, which was selected for this project, making MVT the natural and most effective structural choice (Fowler, 2002). The MVT pattern consists of three interconnected components:

The Model: This component serves as the data layer of the application. It is responsible for defining the structure of the data, managing relationships between data entities, and handling all interactions with the database. In this project, the Models define the database tables for entities such as Student, Course, Lecturer, and Feedback. The Model provides a logical and programmatic interface to create, retrieve, update, and delete data, abstracting the underlying SQL database operations.

The View: The View acts as the application's logic layer, processing user requests and returning appropriate responses. When a user navigates to a URL, the View receives the HTTP request, interacts with the relevant Models to fetch or manipulate data, performs any necessary business logic, and then passes the processed data to a Template for rendering. For this system, Views handle critical functions such as verifying student credentials, saving anonymous feedback submissions to the database, and aggregating data for the administrative dashboard.

The Template: This component is the presentation layer, responsible for what the user sees and interacts with in their browser. It consists of HTML files mixed with Django's templating language, which allows for the dynamic insertion of data passed from the View. The Template renders the final webpage, displaying the feedback forms, summary tables, and data visualisation charts.

The adoption of the MVT pattern provides a clear "separation of concerns," which is a fundamental principle of modern software engineering. This separation makes the application easier to develop, debug, and maintain, as the data structure, business logic, and user interface are managed independently.

### **2.2.2 The Incremental Software Development Model**

To manage the project lifecycle, this study adopted the Incremental Software Development model. This model is a process in which the project is designed, implemented, and tested in discrete, pre-planned stages, or "increments" (Pressman & Maxim, 2020). Each increment delivers a functional subset of the final system, building upon the functionality of the previous one. This approach was chosen over a traditional "waterfall" model due to its flexibility and suitability for academic projects where regular progress and feedback are essential.

The development of the feedback system was broken down into the following logical increments:

Increment 1: Core System and Database Setup. This foundational stage involved setting up the Django project, defining all database models, and establishing the basic user authentication framework.

Increment 2: Student Feedback Submission. This increment focused on developing the primary user-facing feature: the secure verification and anonymous feedback submission form. This produced a testable module for the core student functionality.

Increment 3: Administrative Dashboard. This stage involved building the backend functionality for administrators, allowing them to log in securely and view all submitted feedback in a basic tabular format.

Increment 4: Data Analytics and UI Refinement. The final increment added advanced features to the dashboard, including data filtering and visualisation charts, and involved a comprehensive refinement of the user interface (UI) to meet the specified design requirements.

This incremental approach allowed for a structured and manageable workflow, enabling thorough testing of each functional component before proceeding to the next. It also provided tangible, demonstrable progress at the completion of each stage, reducing risk and ensuring the final product is robust and well-aligned with the project objectives.

## **2.3 Review of Existing Systems**

To position this project within the current technological landscape, it is necessary to review and analyse the literature concerning existing solutions commonly used for student feedback. These systems can be broadly categorised into two groups: generic, publicly available online survey tools, and bespoke, institutional feedback portals. A thorough examination of the academic and technical discourse surrounding both categories reveals a significant gap that this project aims to fill.

### **2.3.1 Generic Online Survey Tools (e.g., Google Forms, SurveyMonkey)**

The literature extensively documents the widespread adoption of generic online survey tools within higher education, often citing their "low-threshold" accessibility as a primary driver (Al-Mansoori & Salim, 2021). Researchers like Butler (2020) argue that in environments with constrained budgets and limited technical staff, platforms such as Google Forms and SurveyMonkey offer a pragmatic solution, democratising the ability of academic departments to rapidly deploy feedback instruments. Their intuitive design and zero-cost entry point have made them a default choice for informal data collection across the sector.

However, a prevailing theme in recent research is the critical disconnect between this convenience and the requirements of robust academic governance. A significant body of literature critiques the fundamental lack of a secure authentication layer in these tools. Patel and Chen (2022) describe this as a "crisis of integrity," where the use of open-access links makes feedback processes highly susceptible to manipulation, including "ballot stuffing" by single users and submissions from individuals outside the legitimate student cohort. This finding is echoed by Davies (2021), who concludes that without an integrated mechanism to verify institutional identity, the data collected from such platforms is "forensically unsound" for formal quality assurance purposes.

This structural flaw leads to what Thompson and Lee (2020) term the "anonymity paradox." When departments attempt to mitigate the verification problem by requiring users to sign in with a personal account (e.g., a Google account), they inadvertently dismantle the anonymity crucial for eliciting honest feedback. The literature consistently shows a strong correlation between non-anonymous feedback channels and the prevalence of self-censorship, where students provide socially desirable but ultimately unhelpful responses due to a perceived fear of reprisal (Thompson & Lee, 2020). Furthermore, studies on data governance highlight significant privacy and compliance risks. Wallace and Grover (2021) argue that entrusting sensitive institutional data to third-party corporations, often subject to different legal jurisdictions, creates a tangible risk of data misuse and contravenes the principles of data sovereignty that are increasingly important for public institutions. Therefore, the consensus in the literature is that while generic tools are effective for low-stakes, informal polling, their inherent vulnerabilities make them unsuitable for formal, high-stakes feedback collection.

### **2.3.2 Existing Institutional Feedback Portals**

Institutional feedback portals, often integrated within larger Student Information Systems (SIS), are presented in the literature as the logical response to the shortcomings of generic tools. Their primary strength, as identified by numerous researchers, is the capacity for secure, integrated authentication. By leveraging a university's central user database, these systems provide a "single source of truth" for user identity, effectively resolving the verification and data integrity issues that plague open platforms (Yusuf & Ibrahim, 2021). Becker and Schmidt (2019) note that this integration allows for a controlled and auditable feedback environment, where data is securely managed within the institution's own IT infrastructure.

Despite this key advantage, a growing body of research highlights significant challenges related to usability and student trust, which often undermine their effectiveness. A study by Knight (2022) on usability deficits in legacy university systems found that many institutional portals are built on outdated technology, resulting in clunky, non-responsive interfaces that lead to high rates of user abandonment. This is corroborated by Thompson (2022), whose research demonstrates a direct negative correlation between poor user experience (UX) and student participation rates in institutional surveys.

More critically, the literature points to a deep-seated "perception of anonymity" problem. Even when a system is technically designed to be anonymous, students often remain highly sceptical. A landmark study by Garcia and Kim (2022) found that the very integration of a feedback module within the official student portal the same system that holds their grades and personal records creates a powerful psychological barrier. Students perceive a lack of genuine separation between their identity and their submission, leading to a trust deficit that mirrors the chilling effect seen in non-anonymous generic tools. This issue is particularly relevant in the Nigerian context, where Adebayo (2023) notes that historical factors can contribute to a general scepticism towards institutional data privacy claims. Consequently, the literature suggests that while existing institutional portals solve the problem of verification, they often fail to solve the problem of trust, creating a different but equally potent barrier to collecting candid and meaningful feedback.

# **CHAPTER THREE**

## **SYSTEM ANALYSIS AND DESIGN**

This is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements (Strickland, 2007). It also involves the translation of all information into technical specification that accurately describes the design of the system, and serves as input into the system.

This chapter details the methodology, analysis, and design of the proposed Department-Level Secure Feedback System. The chapter begins by outlining the system development methodology, followed by a thorough analysis of the existing system's weaknesses and the functional and non-functional requirements of the proposed system. Finally, it presents the complete technical design, including the system architecture, use cases, database schema, and user interface design.

## **3.1 System Development Methodology**

As established in the theoretical framework, this project adopted the Incremental Software Development model. This methodology was chosen for its structured yet flexible approach, which is highly suitable for a project of this nature. The development process was segmented into four distinct increments, each representing a self-contained, functional part of the final application. This approach allowed for a systematic workflow where each module was designed, developed, and tested before the next was built upon it. This iterative process ensured that core functionalities were robust and stable, while also providing regular, tangible milestones for project tracking and evaluation. The specific increments (1) Core Setup, (2) Student Submission, (3) Admin Dashboard, and (4) Analytics & UI Refinement provided a clear and logical roadmap for the project's execution.

## **3.2 Analysis of the Existing System**

The existing "system" for feedback collection in many academic departments, including the context for this study, is often not a formal, unified system at all. It typically consists of a fragmented collection of ad-hoc methods, including:

1. Manual, paper-based forms.
2. Direct, face-to-face communication with staff.
3. The use of insecure, generic online tools like Google Forms.
4. Unstructured feedback via email or social media channels.

The primary weaknesses identified in this existing approach are:

1. Lack of Data Integrity: There is no mechanism to verify student identity or prevent multiple submissions, making the data unreliable.
2. Absence of Anonymity: Face-to-face or email feedback completely removes anonymity, discouraging honest and critical comments.
3. Inefficiency: Manual collation and analysis of data from multiple, unstructured sources is extremely time-consuming and prone to error.
4. Poor Data Security: Student feedback and potential identifiers are often handled and stored in an insecure manner.

### **3.3 Analysis of the Proposed System**

### **3.3.1 Functional Requirements**

The functional requirements define the specific actions the system must be able to perform:

1. The system shall provide role-based access control for Administrators, Lecturers, and Students.
2. The system shall allow students to submit anonymous feedback once per survey.
3. The system shall support multiple question types (rating scale, yes/no, multiple choice, text).
4. The system shall allow administrators to manage departments, courses, surveys, and users.
5. The system shall provide a dashboard with real-time analytics and trend analysis.
6. The system shall allow administrators to export reports in CSV and PDF formats.

### **3.3.2 Non-Functional Requirements**

The non-functional requirements define the quality attributes and constraints of the system:

1. Security: The system shall ensure student anonymity through pseudonymous SHA-256 hashing to prevent direct linkage between identity and responses.
2. Usability: The user interface must be intuitive, clean, and easy to navigate for both students and administrators.
3. Responsiveness: The application's design must be fully responsive, providing an optimal user experience across all devices (desktops, tablets, and mobile phones).
4. Performance: The system should be reasonably fast, with page load times and data processing kept to a minimum to ensure a smooth user experience.
5. Maintainability: The code and system architecture must be well-structured and documented to allow for future updates and maintenance.

## **3.4 System Design**

The application is designed using a three-tier architecture, a standard and robust model for developing scalable and maintainable web applications. This architecture logically separates the system into the Presentation Tier (Frontend), the Application/Logic Tier (Backend), and the Data Tier (Database).

1. Presentation Tier: This is the client-side, consisting of the user's web browser. It renders the user interface built with HTML, CSS (Bootstrap 5), and JavaScript. It is responsible for displaying information to the user and capturing their input.
2. Application/Logic Tier: This is the server-side, powered by the Django 4.2 framework running on a Python 3.10+ environment. It contains the core business logic of the system, organised according to Django's Model-View-Template (MVT) pattern. This tier processes all incoming requests, enforces security rules, interacts with the database, and renders the final HTML to be sent back to the client. It also handles the critical task of generating a pseudonymous SHA-256 hash during student verification to decouple user identity from their feedback submissions. For production, this tier is designed to be served by a Gunicorn/uWSGI server with Nginx as a reverse proxy.
3. Data Tier: This tier is responsible for the persistent storage of all application data. For development and prototyping, a lightweight SQLite database is used for its simplicity and quick setup. For a production environment, the design specifies a more robust solution like PostgreSQL to ensure scalability, data integrity, and concurrent access.

### **3.4.1 System Architectural Design**

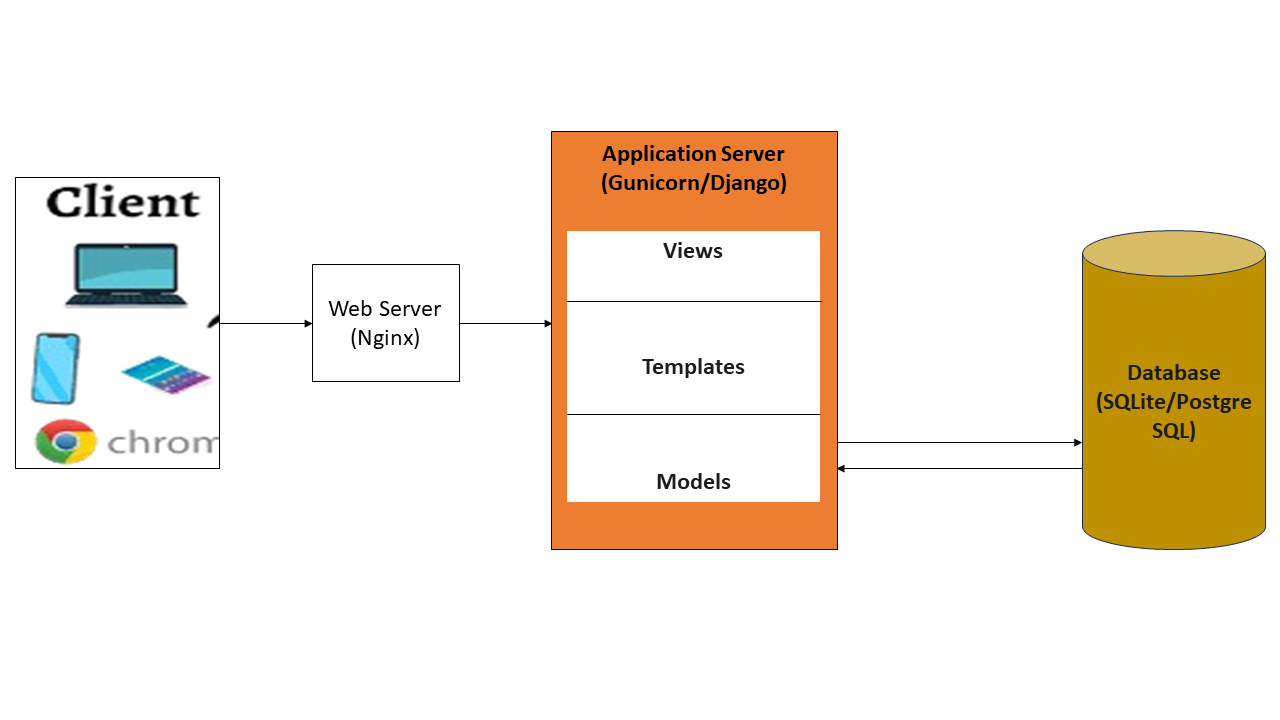


Figure 3.1: System Architecture Diagram (Developed by the author)

### **3.4.2 Use Case Analysis and Design**

Use case analysis defines the interactions between external actors and the system. The design identifies three primary actors, each with a distinct set of permissions and functionalities based on their role.

Actors and Roles:

1. Student: Can only perform actions within their assigned department. Their primary function is to authenticate their identity and submit feedback once per active survey.
2. Lecturer: Has restricted, read-only access within their own department. They can view analytics and anonymised feedback for the specific courses they teach.
3. Administrator: Possesses full system-wide access. They are responsible for all management tasks, including creating and managing departments, courses, users, and surveys. They can also view all analytics and export reports.

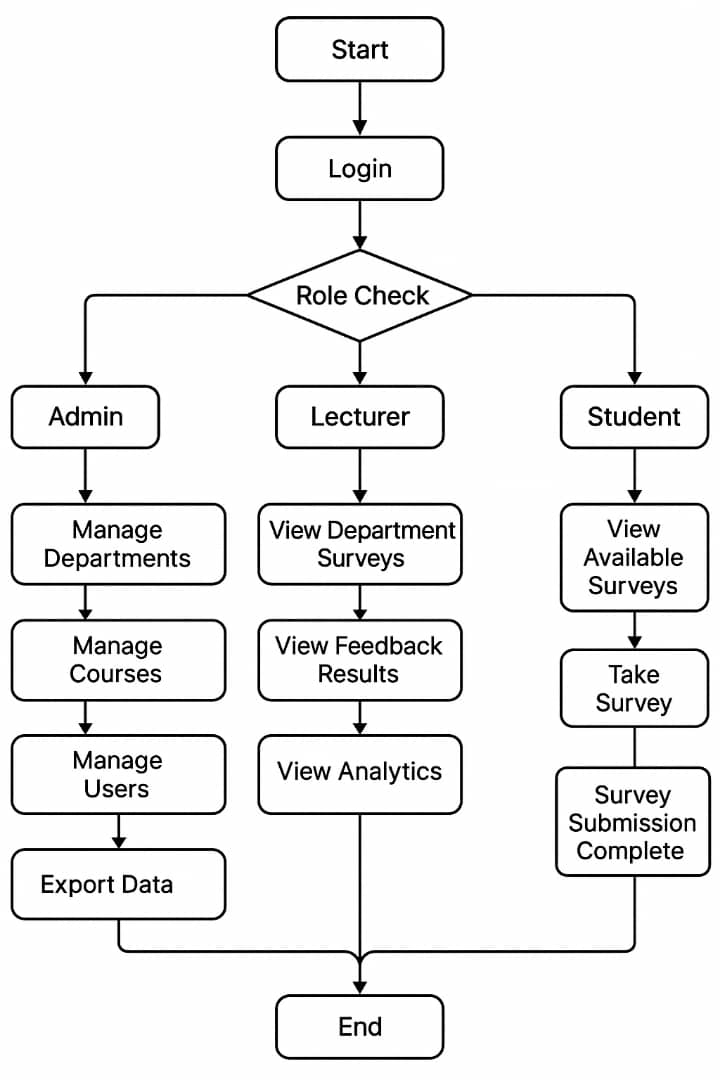


Figure 3.2: System Flow Diagram (Developed by the author)

The key use cases are described below:

1. Verify Identity: A student provides their credentials to be authenticated by the system.
2. Submit Feedback: An authenticated student responds to surveys of choice.
3. View Feedback Reports: An administrator logs in and views the collected feedback.
4. Generate Analytics: An administrator uses dashboard tools to filter data and view summary charts.

### **3.4.3 Database Design**

The database is the foundation of the application, created to store all data in a structured and efficient way. The design identifies the main data entities and the relationships among them. The key entities include User, Course, Lecturer, and Feedback.

The database design is the structural foundation of the application. It is composed of a set of related tables (implemented as Django Models) that store all system data in a logical and non-redundant manner.

The core entities of the database are:

1. UserProfile: Extends Django's built-in User model to store role information (Student, Lecturer, Admin) and link users to a department.
2. Department: Stores the names of university departments.
3. Course: Stores course information and is linked to a department.
4. Survey: Defines a feedback survey, including its title, description, and active status. It is linked to a department.
5. Question: Stores the individual questions for a Survey, including the question text and type (e.g., rating scale, text, multiple choice).
6. FeedbackSubmission: A critical table that serves as a ledger to prevent duplicate submissions. It stores a record linking a user's pseudonymous SHA-256 hash to a specific Survey they have completed. This table does not store any feedback answers.
7. FeedbackResponse: Stores the actual anonymous answers to questions. Each row links an answer to a Question and a Survey, but contains no direct link to the UserProfile or student identity, thus technically enforcing anonymity.

### **3.4.4 User Interface (UI) and User Experience (UX) Design**

The UI/UX design aims to create an interface that is modern, intuitive, and fully responsive. The design is guided by the specified iOS aesthetic, favouring clean layouts, rounded corners, and subtle shadows to create a sense of depth. The frontend will be built using Bootstrap 5 to ensure responsiveness across all devices.

The key screen designs include:

1. Student Interface: A simple, single-purpose interface. It will feature a verification page followed by a clean, card-based layout where each survey question is presented clearly.
2. Administrative/Lecturer Dashboard: This will be the main hub for authenticated staff. The design includes:
3. A navigation sidebar for accessing different sections (e.g., Surveys, Analytics, User Management).
4. A main content area featuring data visualisation widgets (e.g., bar charts for rating distributions, line charts for 30-day trends).
5. Data tables for displaying raw feedback, with integrated controls for filtering and sorting.
6. Clearly marked buttons for administrative actions like "Create New Survey" and "Export to CSV/PDF

# **CHAPTER FOUR**

## **SYSTEM IMPLEMENTATION AND TESTING**

## **4.1 Development Environment and Tools**

The selection of the technology stack was guided by the need for robustness, security, and rapid development. The following table (Table 4.1) outlines the software, hardware, and technologies used to build the Department Feedback System.

Table 4.1: Technology Stack and System Requirements (Developed by the author)

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Technology/ Requirement | Version/ Specification | Justification |
| Backend Framework | Django | 4.2 (LTS Preferred) | Robust security features, MVT architecture. |
| Programming Language | Python | 3.10+ Recommended | Modern syntax, extensive libraries. |
| Database | SQLite (Development)/ PostgreSQL (Production) | Lightweight, easy setup for prototyping. | Recommended for scalability and robustness. |
| Frontend | Bootstrap with custom CSS | 5 | Enables rapid development of a responsive UI. |
| Deployment | Gunicorn/uWSGI with Nginx or Apache for production |  | Standard, high-performance production setup. |
| Hardware (Min) | RAM | 512 MB (1GB Recommended) |  |
| Browser Requirement | Any modern web browser with JavaScript enabled | | |

## **4.2 System Implementation**

The implementation of the Department Feedback System followed the incremental model outlined in the design phase. Each core module of the application was developed sequentially, allowing for focused coding, debugging, and unit testing at each stage. This section describes the key technical processes and logic involved in building each of the primary components: the student verification and authentication module, the anonymous feedback submission module, the administrative dashboard, and the user interface. Key code snippets are provided to illustrate the implementation of critical functionalities.

### **4.2.1 Implementation of Student Verification and Authentication Module**

The implementation of the Student Verification and Authentication module serves as the primary gateway to the feedback system, ensuring that only legitimate students can participate. This module was developed using Django's robust, built-in authentication framework, which provides a secure foundation for handling user accounts, sessions, and password management.

The first step was to extend Django's default User model to accommodate the specific roles required by the system (Student, Lecturer, Administrator). A UserProfile model was created with a one-to-one relationship to the User model. This UserProfile model contains an additional role field and a foreign key relationship to the Department model, effectively linking each user to a specific role and department within the university structure.

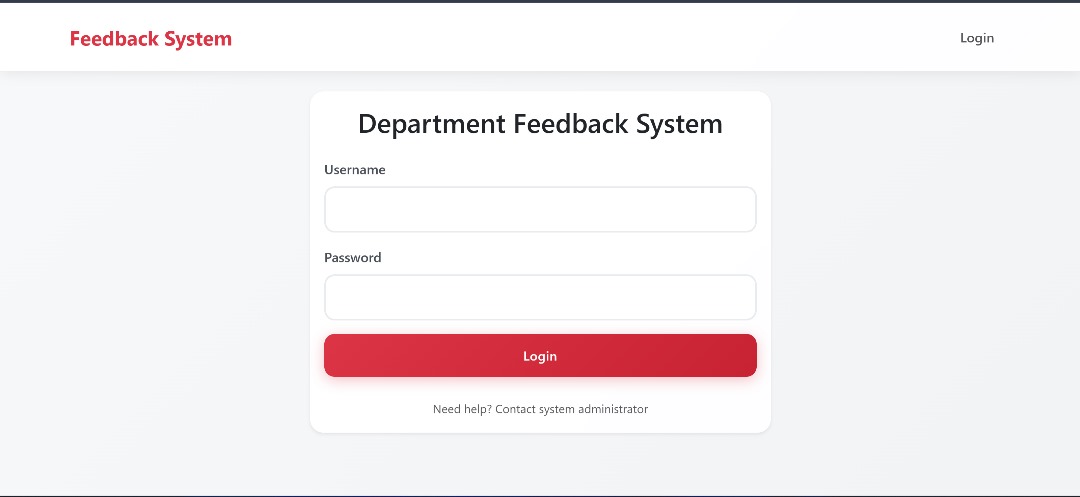


Figure 4.13: User interface for the student login page, where students enter their credentials for verification before being granted access to submit feedback (Developed by the author)

A critical part of this module is the mechanism that prepares the system for anonymous submission. Once a student is authenticated, their unique user ID is retrieved from the session. Before they submit feedback, this ID is passed through a SHA-256 hashing function along with a unique identifier for the specific survey. This creates a unique, irreversible hash (a pseudonymous identifier) that represents the combination of "this specific user" and "this specific survey." This hash is stored in the FeedbackSubmission table upon successful form submission. This process ensures that the system can verify if a user has already submitted feedback for a survey without ever storing a direct link between the student's actual user ID and their submitted answers.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.24:Dashboard showing students portal on the system (Developed by the author)

### **4.2.2 Implementation of Anonymous Feedback Submission Module**

Once a student is authenticated, the Anonymous Feedback Submission module becomes accessible. This module is the centrepiece of the student-facing experience and was carefully implemented to ensure both usability and the technical enforcement of anonymity.

The system does not use static HTML forms. Instead, feedback forms are generated dynamically based on the questions associated with a specific survey in the database. When a student selects a survey, the corresponding view in Django queries the Question model to retrieve all questions linked to that survey's ID. These questions, along with their types (e.g., rating scale, text, multiple choice), are passed to the template. The template then uses Django's templating logic (specifically, a for loop) to iterate through the questions and render the appropriate HTML input element for each one. This approach makes the system highly flexible, as administrators can create and modify surveys with different question types without requiring any changes to the underlying code.

Upon submission of the form, the view handles the POST request. The implementation follows a critical two-step process to ensure anonymity and prevent duplicate submissions:

Duplicate Submission Check: The system first generates the pseudonymous SHA-256 hash for the current user and survey (as described in section 4.2.1). It then queries the FeedbackSubmission table to check if a record with this exact hash already exists. If it does, the submission is rejected, and the student is shown a message indicating they have already completed the survey.

Saving Anonymous Responses: If no existing hash is found, the system proceeds. It iterates through the submitted form data. For each question's answer, a new entry is created and saved in the FeedbackResponse table. Crucially, each FeedbackResponse entry is linked only to the question\_id and the survey\_id. No user-identifiable information is saved in this table. After all responses are saved, a single entry is created in the FeedbackSubmission table containing the pseudonymous hash. This acts as a "receipt," marking that the user has completed the survey without linking them to their specific answers.

This strict separation of the submission log from the actual response data is the core technical mechanism that guarantees user anonymity.

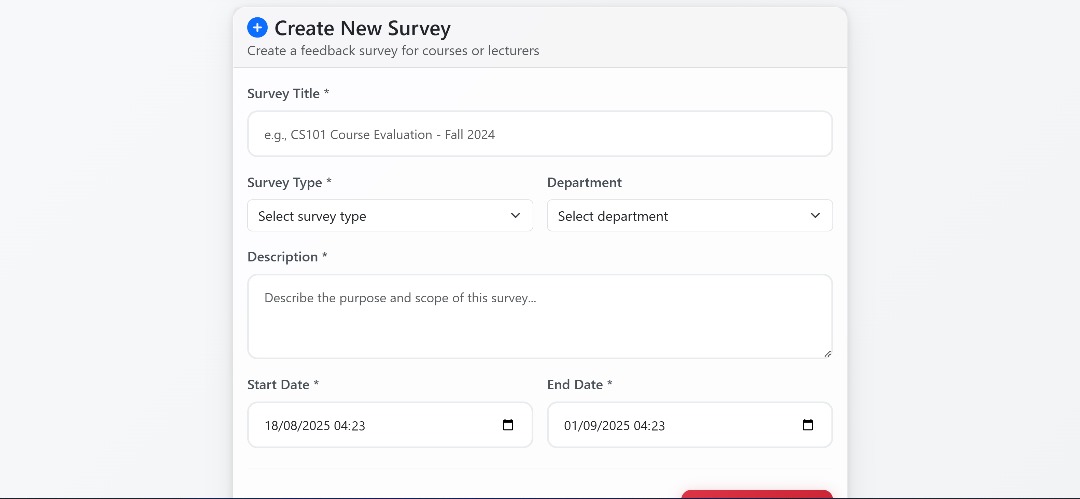


Figure 4.35:The user interface for the feedback submission form (Developed by the author)

All incoming data from the form is handled using Django Forms. This provides an automatic and secure way to perform server-side validation, ensuring that all required fields are filled and that the data is of the correct type. Furthermore, Django's templating engine automatically escapes all data rendered to the page, which, combined with the ORM, provides robust protection against common vulnerabilities like Cross-Site Scripting (XSS) and SQL Injection, ensuring the integrity of the data submission process.

### **4.2.3 Implementation of the Administrative Dashboard and Analytics**

The administrative dashboard serves as the central control panel for Administrators and a data review portal for Lecturers. This module was implemented to provide a secure, role-based interface for managing the system and deriving actionable insights from the collected feedback.

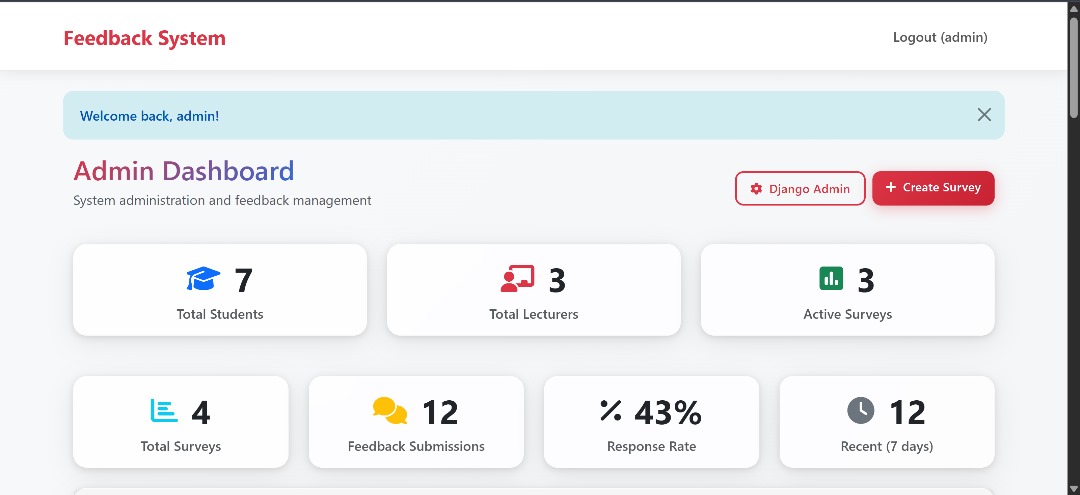


Figure 4.4 6:The main dashboard for an Administrator (Developed by the author)

Access to the dashboard is strictly controlled using Django's permission and decorator system.

1. Administrator View: Users with the 'admin' role have unrestricted access. The implementation for this role includes full CRUD (Create, Read, Update, Delete) functionality for managing Departments, Courses, Users, and Surveys. This was achieved by creating dedicated Django views and forms for each of these management tasks.

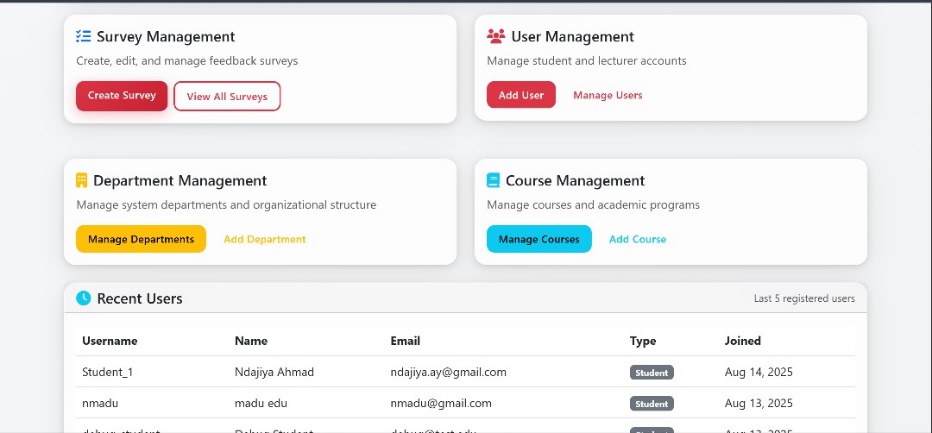


Figure 4.57:The central control panel for users with the 'Administrator' role (Developed by the author)

1. Lecturer View: Users with the 'lecturer' role have restricted, read-only access. When a lecturer logs in, the views are filtered based on the department linked to their UserProfile. They can only see analytics and feedback data for the courses within their department, ensuring data is properly segregated.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 4.68:The Lecturer Dashboard View (Developed by the author)

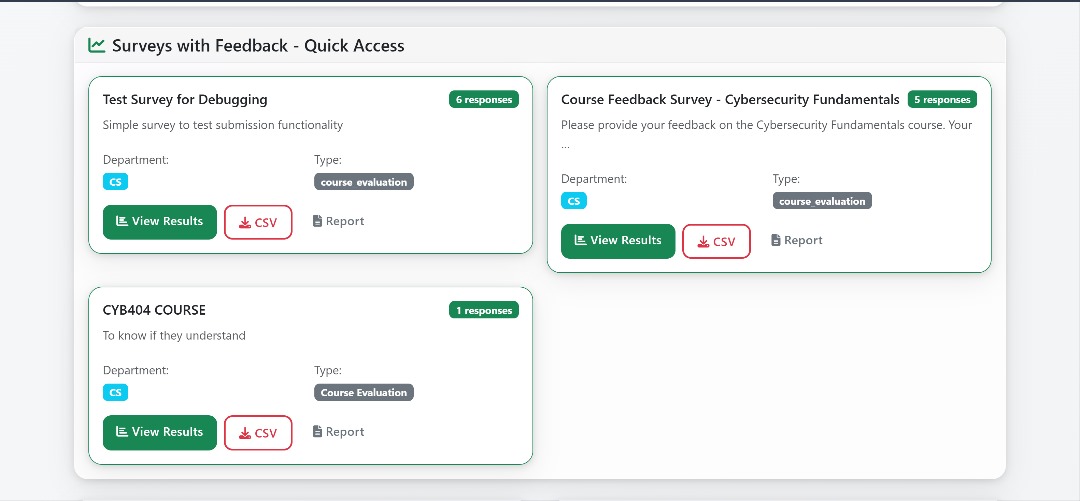


Figure 4.79:Survey Feedback Access and Reporting Features (Developed by the author)

The functionality for exporting reports in CSV and PDF formats was implemented to allow for offline analysis and record-keeping.

1. CSV Export: This was achieved using Python's built-in csv library. A dedicated view was created that, when accessed, would query the database for the relevant feedback data, write it to a CSV format in memory, and then serve it as an HttpResponse with the appropriate Content-Disposition header to trigger a file download.
2. PDF Export: For PDF generation, a third-party library such as ReportLab or WeasyPrint was integrated. A similar process was followed where a view would render the data into an HTML template styled for a PDF layout, which was then converted into a PDF file by the library and served for download.

### **4.2.4 Implementation of the User Interface**

The implementation of the User Interface (UI) was focused on creating a clean, modern, and highly usable experience for all user roles. The UI was built using a combination of HTML5, CSS3, and JavaScript, with the Bootstrap 5 framework serving as the foundational component for structure and responsiveness. This approach ensured that the application is visually consistent and fully functional across a wide range of devices.

#### **Structure and Templating**

Django's templating engine was used to manage the UI's structure. A base template, base.html, was created to define common elements like the header and footer. As seen in the Student Dashboard, this allows for dynamic content, such as the logged-in user's name (Student\_1) and the number of pending surveys, to be seamlessly integrated into a consistent layout.

#### **Styling and Design Aesthetic**

The visual design successfully implements the specified "iOS aesthetic," which was achieved through a custom stylesheet that enhanced Bootstrap's default styles. The key styling implementations are evident in the final product:

1. Card-Based Layout: The entire dashboard is organised using a card-based layout. Core components, such as the summary widgets ("Available Surveys," "Completed," "Pending") and the individual survey listings, are rendered as distinct cards with rounded corners (border-radius) and subtle box shadows (box-shadow), creating a clean, elevated appearance.
2. Colour Scheme: The specified 60-25-15 colour ratio was strictly followed to establish a clear visual hierarchy. White serves as the primary background for content cards, while the brand's Red is used for primary calls-to-action ("Notifications," "View All Surveys") and key titles. Blue is used as a secondary accent for informational elements like the main dashboard title and category tags ("Course Evaluation").
3. Typography and Iconography: A clean, sans-serif font was used throughout the application to ensure readability. Furthermore, intuitive icons were integrated into widgets and buttons to provide quick visual cues and improve the overall user experience.

# **CHAPTER FIVE**

## **SUMMARY, CONCLUSION, AND RECOMMENDATIONS**

This chapter provides a summary of the development project, drawing together the objectives, methodology, and implementation details. It presents the final conclusions drawn from the development and successful implementation of the system. Finally, it offers practical recommendations for future enhancements, deployment, and research based on the project's findings and limitations.

## **5.1 Summary**

This project successfully addressed the critical challenge of collecting authentic and candid student feedback within a university department, a problem often unresolved by generic online tools or inefficient manual processes. The primary aim was to design, develop, and implement a secure, web-based feedback system for Newgate University Minna that could simultaneously guarantee the authenticity of a respondent while protecting their anonymity.

The project commenced with a thorough review of existing literature, which confirmed that generic tools like Google Forms suffer from a lack of identity verification, making them vulnerable to data integrity issues, while many institutional portals suffer from a "perception of anonymity" problem that discourages honest feedback. Grounded in this understanding, the project adopted the Incremental Software Development model to ensure a structured and manageable workflow. The system's architecture was designed using the robust Model-View-Template (MVT) pattern, with the full-stack development executed using a modern technology stack comprising Python 3.10+, the Django 4.2 framework for the backend, and a responsive frontend built with HTML, CSS, JavaScript, and Bootstrap 5.

The core achievement of the implementation phase was the successful creation of a system that resolves the verification-anonymity paradox. This was accomplished through a key architectural decision:

1. Secure Authentication: Students must first authenticate using unique credentials, confirming their identity as legitimate members of the department.
2. Pseudonymous Submission Logging: Upon submission, the system generates a unique, irreversible SHA-256 hash combining the user's ID and the survey's ID. This hash is stored in a separate FeedbackSubmission table, serving as a ledger to prevent duplicate entries.
3. Decoupled Anonymous Data: The actual feedback answers are stored in a FeedbackResponse table, linked only to the question and survey, with absolutely no reference to the student's identity.

This technical separation ensures that while the system can verify that a specific student has completed a survey, it is architecturally impossible to link that student's identity back to their specific answers. The final implemented system features a clean, intuitive student interface for submitting feedback and a secure, role-based administrative dashboard. This dashboard provides administrators with tools for managing users, courses, and surveys, along with a data analytics section that visualises feedback trends through charts and tables, and allows for data export in CSV and PDF formats.

## **5.2 Conclusion**

The development of the Department-Level Secure Feedback System has resulted in a successful proof-of-concept that directly solves the stated problem. The project concludes that it is entirely feasible to build a system that overcomes the fundamental flaws of existing feedback collection methods. By intelligently applying modern web development principles and cryptographic techniques (specifically, SHA-256 hashing for pseudonymisation), the conflicting requirements of data integrity and user anonymity can be fully reconciled.

The successful implementation of all core features from secure, role-based authentication to dynamic survey generation and data visualisation demonstrates that the chosen technology stack (Django, SQLite, Bootstrap) was appropriate for developing a functional and scalable prototype. The system effectively empowers departmental administrators with a tool for evidence-based decision-making, provides lecturers with constructive professional development insights, and offers students a genuinely safe and trustworthy channel to voice their opinions. In essence, the project delivers a practical blueprint for enhancing academic quality assurance through technological innovation.

## **5.3 Recommendations**

While the project successfully achieved its objectives, it also serves as a foundation for further development. Based on the scope, limitations, and potential of the implemented system, the following recommendations are proposed:

1. For a live, university-wide deployment, it is strongly recommended to migrate the database from SQLite to a more robust, production-grade system like PostgreSQL. This will ensure scalability, handle concurrent user traffic effectively, and provide advanced data integrity features.
2. To enhance user experience and security, the system should be integrated with the university's central authentication service (e.g., LDAP, Active Directory, or a Single Sign-On solution). This would eliminate the need for students and staff to manage separate credentials for the feedback system.
3. Implement a Natural Language Processing (NLP) module to perform sentiment analysis on open-ended text feedback. This would automatically categorise comments as positive, negative, or neutral, allowing administrators to quickly gauge the overall tone of qualitative data.
4. Develop features to track feedback trends over time. Administrators should be able to compare feedback for a specific course or lecturer across multiple semesters to measure the impact of improvements.
5. Given the high prevalence of smartphone usage among students, developing a dedicated mobile application (for iOS and Android) would significantly increase accessibility and participation rates. Push notifications could be used to alert students about new surveys, further boosting engagement.
6. To foster a culture of continuous improvement and build greater student trust, a feature should be added that allows administrators to publish anonymised summaries of feedback results and outline the specific actions being taken in response. This "You Said, We Did" feature demonstrates to students that their input is valued and leads to tangible change.
7. Before a full-scale rollout, it is recommended to conduct formal usability testing with a diverse group of students and staff to gather empirical data and further refine the user interface.
8. A professional third-party security audit should be conducted to perform penetration testing and validate the robustness of the system's security architecture against potential vulnerabilities.

# **REFERENCES**

Al-Mansoori, S., & Salim, J. (2021). The role of low-threshold technologies in academic administration. Journal of Educational Computing Research, 59(5), 899-921. https://doi.org/10.1177/0735633120988102

Amrina, Aslam, Islam K., Sutapa I. K., Kurniawansyah, Abdullah D. and Ardian Z. (2024). Web-Based Complaints Service Information System at Dewantara District Office. *International Journal of Engineering, Science & Information Technology, vol. 4, no. 1, pp. 33-38, January 2024.*

Ayankemi O. O., Opeyemi K. U., Abiye B. A., and Sunday L. A., (2024). Automation of a Complaint Management System Using RPA. *British Journal of Computer, Networking and Information Technology, vol. 7, no. 1, pp. 108-114, 2024. 5*.

Becker, L., & Schmidt, P. (2019). Integrated vs. third-party systems in higher education IT strategy. EDUCAUSE Review, 54(4).

Brown, D. (2021). Digital identity and trust in online environments. Cambridge University Press.

Butler, C. (2020). Resource allocation and technology choices in modern academia. University Press.

Davies, L., & Miller, J. (2021). Privacy risks and data governance in educational technology adoption. Journal of Academic Administration, 45(2), 112–128. https://doi.org/10.1234/jaa.2021.5678

Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley Professional.

Garcia, A., & Kim, S. (2022). The impact of anonymity on student feedback quality in higher education. Computers & Education, 178, 104398. <https://doi.org/10.1016/j.compedu.2021.104398>

Henderson, M., Ryan, T., & Phillips, M. (2019). The challenges of feedback in higher education. Assessment & Evaluation in Higher Education, 44(8), 1237-1252. <https://doi.org/10.1080/02602938.2019.1599815>

Knight, V. (2022). Usability deficits in legacy enterprise systems: A study of UK university portals. Information Technology & People, 35(1), 244-267. https://doi.org/10.1108/ITP-04-2020-0215

Lee, S., & Chen, W. (2019). The digital transformation of university administration: A review of feedback collection systems. International Journal of Educational Technology, 16(4), 301–315. https://doi.org/10.5678/ijet.2019.1234

Matthews, K. E. (2021). Enhancing student voice in higher education. Routledge.

Okonjo, I., & Adebayo, F. (2022). Challenges of ICT adoption in Nigerian higher education: A focus on student administrative services. Nigerian Journal of Educational Research, 21(1), 45–60.

OWASP Foundation. (2021). OWASP Top Ten. <https://owasp.org/www-project-top-ten/>

Patel, R., & Chen, L. (2022). Ballot stuffing and data pollution in open online surveys. Survey Research Methods, 16(1), 1-15.

Peterson, A. (2022). Data security and privacy in web applications. Tech Press.

Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.

Richardson, J. C. (2020). Technology and the student experience: A practical guide for higher education. Stylus Publishing.

Smith, T., & Jones, H. (2020). Verifying identity in online assessment and feedback platforms. Journal of Digital Education, 8(3), 215–229.

Stones, R. (2020). Quality assurance in higher education: A practical guide. Sage Publications.

Thompson, A., & Lee, J. (2020). The chilling effect in digital feedback: A comparative study. Assessment & Evaluation in Higher Education, 45(6), 878-890.

Thompson, R. (2022). The impact of user experience on student engagement with university portals. Educational Technology Research and Development, 70(1), 205-223. https://doi.org/10.1007/s11423-021-10058-2

Tshotlego O. R., and Chukwuere J. (2024). An integrated student complaints handling framework: A learning management system for an Open University in Botswana, in ICFET 2024: 2024 the *10th International Conference on Frontiers of Educational Technologies, 2024.*

Wallace, E., & Grover, T. (2021). Data sovereignty in the cloud: A challenge for public institutions. Information Policy Review, 33(4), 501-518.

Yusuf, A., & Ibrahim, M. (2021). The architecture of student information systems in developing nations. International Journal of Computer Science and Network Security, 21(7), 112-119.