

FACULTY OF SCIENCE AND EDUCATION

FINAL YEAR PROJECT REPORT

**A WEB-BASED CORRUPTION REPORTING SYSTEM FOR KOLE
DISTRICT LOCAL GOVERNMENT, NORTHERN UGANDA**

By

OMONGO DENIS

Registration Number: BU/UP/2023/1513

Contact: +256 770 747269 / +256 771 494379

Email: omongobden@gmail.com

Department of Computer Studies

Faculty of Science and Education

Busitema University

**A Project Report Submitted to the Faculty of Science and Education in
Partial Fulfilment of the Requirements for the Award of the Degree of
Bachelor of Information Technology of Busitema University.**

Supervisor

Dr. Mutungi Fredrick

Department of Computer Studies

Faculty of Science and Education

Busitema University

April 2026

DECLARATION

I, **Omongo Denis**, Registration Number **BU/UP/2023/1513**, hereby declare that this project report entitled "**A Web-Based Corruption Reporting System for Kole District Local Government, Northern Uganda**" is my original work. It has not been submitted to any other university or institution of higher learning for the award of any academic qualification.

The report has been prepared in accordance with the regulations governing the award of the Bachelor of Information Technology degree at Busitema University. All sources of information used in this report have been acknowledged through appropriate citations and references.

Student's Signature: 

Date: 31/08/2026

APPROVAL

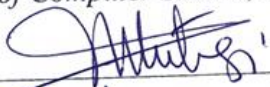
This project report entitled "**A Web-Based Corruption Reporting System for Kile District Local Government, Northern Uganda**" has been submitted with my approval as the university supervisor.

Supervisor's Name: Dr. Mutungi Fredrick

Department of Computer Studies, Faculty of Science and Education, Busitema University

Signature: _____

Date: _____


01/09/2026

DEDICATION

This project is dedicated to my parents, family members, and friends whose encouragement, sacrifices, and unwavering support have enabled me to pursue my academic goals successfully.

I also dedicate this work to the people of Kole District Local Government. It is my hope that this research contributes towards improving transparency, accountability, and citizen participation in the fight against corruption through the use of modern information technology.

Finally, I dedicate this work to all students and researchers committed to developing innovative digital solutions that address societal challenges and promote sustainable development.

ACKNOWLEDGEMENT

I thank the Almighty God for granting me good health, wisdom, strength, and guidance throughout the course of this research project.

I express my sincere appreciation to my supervisor, **Dr. Mutungi Fredrick**, for his valuable guidance, constructive criticism, patience, and continuous support during the planning, design, implementation, and documentation of this project. His professional advice greatly contributed to the successful completion of this research.

My gratitude also goes to the lecturers and staff of the Department of Computer Studies, Faculty of Science and Education, Busitema University, for providing the academic knowledge and technical skills that formed the foundation of this work.

I acknowledge the officials and residents of Kileleshwa District Local Government who willingly participated in interviews and shared valuable information during the requirements gathering stage of this study. Their cooperation made this research possible.

Special appreciation goes to my parents, family, relatives, and friends for their financial support, encouragement, prayers, and motivation throughout my academic journey.

Lastly, I thank everyone who contributed directly or indirectly to the successful completion of this project.

Table of Contents

DECLARATION.....	2
APPROVAL	3
DEDICATION	4
ACKNOWLEDGEMENT	5
LIST OF ABBREVIATIONS.....	10
ABSTRACT.....	13
CHAPTER ONE: INTRODUCTION.....	i
1.0 Introduction	i
1.1 Background of the Study.....	i
1.2 Problem Statement	ii
1.3 Purpose of the Study.....	iii
1.4 Specific Objectives	iii
1.6 Significance of the Study.....	iv
1.5 Scope of the Study	iv
CHAPTER TWO: LITERATURE REVIEW	vi
2.1 Introduction	vi
2.2 Review of Existing Corruption Reporting Systems	vi
2.3 System Design and Architecture	vii
2.4 System Implementation Technologies	viii
2.5 System Testing and Evaluation	viii
2.6 Research Gap	ix
CHAPTER THREE: METHODOLOGY	x
3.1 Introduction	x
3.2 Research Design and Development Approach.....	x
3.3 Area of Study.....	x
3.4 Target Population and Sampling methods	xi
3.5 Data Collection Methods and Instruments	xii
3.6.1 Data Quality Control (Validity and Reliability).....	xiii
3.6 Data Analysis Methods.....	xiii
3.10 Ethical Considerations.....	xiii
CHAPTER FOUR: FIELD STUDY RESULTS, DATA ANALYSIS, SYSTEM DESIGN AND IMPLEMENTATION ...	xv
4.1 Introduction	xv
4.2 Analysis of the Existing (Manual) Reporting System.....	xv

4.2.1 Current System Workflow	xv
4.2.2 Strengths of the Current System	xv
4.2.3 Weaknesses of the Current System	xv
4.3 Data Presentation and Analysis	xvi
4.3.1 Quantitative Field Findings	xvi
4.3.2 Qualitative Verbatim Extracts from Field Interviews	xvii
4.4 Requirements Analysis	xviii
4.4.1 User Requirements	xviii
4.4.2 Functional Requirements	xix
4.4.3 Non-Functional Requirements	xix
4.4.4 System Requirements (Hardware and Software)	xx
4.5 System Design	xx
4.5.1 Three-Tier Client-Server System Architecture	xx
4.5.2 Context Diagram (DFD Level 0)	xxi
4.5.3 Data Flow Diagram (DFD Level 1)	xxii
4.5.4 Use Case Diagram & Actor Descriptions	xxii
4.5.5 Entity Relationship Diagram (ERD) with Fernet Encryption Annotations	xxv
4.5.6 System Implementation Architecture	xxvi
4.6 System Deployment and Cloud Environment	xxvii
4.6 Chapter Summary	xxvii
CHAPTER FIVE: IMPLEMENTATION AND TESTING	xxviii
5.1 Introduction	xxviii
5.2 Development Environment	xxviii
5.3 System Design Considerations	xxix
5.4 Homepage Module (Figure 7)	xxix
5.5 Citizen Registration and Authentication Interface	xxxi
(Figure 8)	xxxi
5.6 Secure Report Submission Module (Figure 9)	xxxi
5.7 Evidence Upload Module (Figure 10)	xxxiii
5.8 Tracking Report Module (Figure 11)	xxxiii
5.9 District Admin Dashboard Module (Figure 12)	xxxiv
5.10 Report Management & Status Update Modules (Figure 13)	xxxv
5.11 Database Implementation	xxxv
5.12 Database Inspection and Table Queries (Figure 14)	xxxvii
5.13 Security Mechanisms & Encryption Engine Implementation (Figure 15)	xxxvii
5.14 System Testing Procedures	xxxviii

5.14.1 Unit Testing	xxxviii
5.14.2 Integration Testing	xxxviii
5.14.3 Security and Vulnerability Testing	xxxix
5.14.4 Comprehensive Functional Test Suite	xxxix
5.14.5 User Acceptance Testing (UAT)	xl
5.14 Chapter Summary	xli
CHAPTER SIX: DISCUSSION, CONCLUSION, RECOMMENDATIONS, AND FUTURE WORK.....	xlii
6.1 Introduction	xlii
6.2 Discussion of Results	xlii
6.2.1 Objective 1: Determining Requirements.....	xlii
6.2.2 Objective 2: Architecture & Database Design	xlii
6.2.3 Objective 3: System Implementation	xlii
6.2.4 Objective 4: System Testing & Evaluation	xlili
6.3 Conclusion	xlili
6.4 Recommendations	xlili
6.5 Limitations of the Study	xliv
6.6 Suggestions for Future Work.....	xliv
REFERENCES	xliv
APPENDICES	xlvi
APPENDIX I: INTERVIEW GUIDE FOR CITIZENS.....	xlvi
APPENDIX II: INTERVIEW GUIDE FOR DISTRICT OFFICIALS	xlvi
APPENDIX III: OBSERVATION CHECKLIST	xlvi
APPENDIX VI: USER ACCEPTANCE TESTING QUESTIONNAIRE.....	xlvi
APPENDIX VII: PROJECT WORK PLAN	l
APPENDIX VIII: PROJECT BUDGET	l

LIST OF FIGURES

Figure 1: Chart showing quantitative field findings (N=15)	xvii
Figure 2: Three-Tier Client-Server System Architecture.....	xxi
Figure 3: Context Diagram (DFD Level 0)	xxi
Figure 4: Data Flow Diagram (DFD Level 1).....	xxii
Figure 5: Use Case Diagram	xxiii
Figure 6: Entity Relationship Diagram (ERD).....	xxv
Figure 7Homepage Module	xxx
Figure 8: Citizen Registration and Authentication Interface	xxxi
Figure 9: Secure Report Submission Module	xxxii
Figure 10: Evidence Upload Module	xxxiii
Figure 11: Tracking Report Module.....	xxxiv
Figure 12: District Admin Dashboard Module	xxxiv
Figure 13: Report Management & Status Update Modules.....	xxxv
Figure 14: PostgreSQL Database Table.....	xxxvii
Figure 15: Security Mechanisms & Encryption Engine Implementation.....	xxxviii

LIST OF TABLES

Table 1: Sample Size Breakdown and Distribution (N=15)	xii
Table 2: Summary of Data Collection Methods and Instruments	xii
Table 3: Quantitative Summary of Major Field Findings (N=15)	xvi
Table 4: Qualitative Verbatim Extracts and Derived Functional Requirements	xvii
Table 5: Hardware, Software, and Environment Technology Specifications	xxviii
Table 6: REPORT Table Schema	xxxv
Table 7: AGENCY Table Schema.....	xxxvi
Table 8: ADMINISTRATOR Table Schema.....	xxxvi
Table 9: STATUS_UPDATE Table Schema	xxxvii
Table 10: Comprehensive Test Suite Results	xxxix
Table 11: User Acceptance Testing (UAT) Category Evaluation Scores.....	xl

LIST OF ABBREVIATIONS

Abbreviation	Meaning
AES	Advanced Encryption Standard
API	Application Programming Interface
CPU	Central Processing Unit
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
DRF	Django REST Framework
ERD	Entity Relationship Diagram
GIS	Geographic Information System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IGG	Inspectorate of Government
IP	Internet Protocol
JSON	JavaScript Object Notation
JWT	JSON Web Token

Abbreviation	Meaning
LC	Local Council
ORM	Object-Relational Mapper
PDM	Parish Development Model
PII	Personally Identifiable Information
PostgreSQL	Postgres Structured Query Language
PWA	Progressive Web Application
RBAC	Role-Based Access Control
REST	Representational State Transfer
S3	Simple Storage Service
SDR	System Development Research
SHACU	State House Anti-Corruption Unit
SMS	Short Message Service
SPA	Single Page Application
SQL	Structured Query Language
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UAT	User Acceptance Testing
UBOS	Uganda Bureau of Statistics
UGX	Uganda Shillings
UI	User Interface
USSD	Unstructured Supplementary Service Data

Abbreviation	Meaning
UWEP	Uganda Women Entrepreneurship Programme
VSLA	Village Savings and Loan Association
WSGI	Web Server Gateway Interface
XSS	Cross-Site Scripting
YLP	Youth Livelihood Programme

ABSTRACT

Corruption continues to undermine public service delivery, accountability, and socio-economic development across local governments in Uganda. In Kole District, corruption associated with strategic government empowerment initiatives such as the Parish Development Model (PDM), Uganda Women Entrepreneurship Programme (UWEP), and Youth Livelihood Programme (YLP), among others, has degraded public confidence in local governance and severely affected the equitable distribution of public funds. Existing corruption reporting channels—primarily physical visits to sub-county headquarters, verbal reports to Local Council officials, or written complaint letters—are critically limited by geographical distance barriers, high transport costs, fear of retaliation due to the lack of a secure, trackable channel, and a total absence of post-submission tracking mechanisms.

This study aimed to design, develop, and evaluate a secure, web-based corruption reporting system for Kole District Local Government to enable registered citizens to securely report corruption in either English or Leblango/Luo and monitor investigation progress using a secure tracking code. The research adopted a System Development Research (SDR) design executed through the Agile-Scrum methodology. Purposive sampling was used to select fifteen key stakeholders (N=15), consisting of PDM beneficiaries, Local Council leaders, civil servants, village savings group leaders, and community elders across six sub-counties (Aboke, Alito, Akalo, Ayer, Bala, and Okwerodot). Sample size adequacy was grounded in the data saturation principle for qualitative software requirements engineering (Guest et al., 2006). Primary data were collected via semi-structured interviews, direct field observation, and document analysis, and evaluated using thematic analysis.

The developed application follows a Three-Tier Client-Server Architecture. The presentation layer was constructed using React.js (v18.x) and react-i18next for multilingual switching. The application tier was built using Python 3.11, Django 4.2+, and Django REST Framework (DRF), utilizing Gunicorn and WhiteNoise servers. The database tier relies on PostgreSQL for structured data and AWS S3 for secure media evidence storage. System security was implemented using authenticated, role-restricted request handling (no IP address or fingerprint logging in server logs), Fernet symmetric encryption for sensitive report descriptions at rest, TLS/HTTPS in transit, and JWT-based Role-Based Access Control (RBAC) for administrative management.

Rigorous evaluation comprising Unit Testing, Integration Testing, System Testing, and User Acceptance Testing (UAT) with 5 representative field evaluators demonstrated that the application satisfied all specified functional and non-functional criteria. Evaluators awarded the platform an overall usability score of 4.8 out of 5.0, highlighting its ease of use, secure encrypted handling, and tracking code transparency. The study concludes that web-based civic platforms can substantially improve anti-corruption reporting in rural local governments. Recommendations include official adoption by Kole District Local Government, integration with central anti-corruption bodies (IGG, SHACU), and future expansion to USSD/SMS channels for non-smartphone users.

CHAPTER ONE: INTRODUCTION

1.0 Introduction

This chapter describes the introductory aspects of the study concerning the development of the Web-Based Corruption Reporting System. It covers the background of the study, outlining the current manual processes used for corruption reporting at Kole District Local Government and the challenges associated with them. The chapter also states the problem that the study aims to solve, specifies the main and specific objectives guiding the system development, defines the geographical and content scope of the project, and highlights the significance of the proposed system in improving transparency and accountability.

1.1 Background of the Study

Corruption remains one of the most pervasive obstacles to good governance, institutional integrity, and socio-economic advancement across developing nations. It systematically diverts public financial resources away from essential public infrastructure, weakens regulatory institutions, reduces citizen trust in state organs, and deepens socio-economic inequality. According to the World Bank (2020), corruption acts as a heavy hidden tax on economic growth, disproportionately impacting vulnerable rural populations who rely heavily on government social safety nets and poverty alleviation grants.

In Uganda, corruption continues to undermine public confidence in national development strategies. Decentralized local governments receive significant central government transfers intended for grassroots development. Key programmes including the Parish Development Model (PDM), the Uganda Women Entrepreneurship Programme (UWEP), and the Youth Livelihood Programme (YLP) have been rolled out to transform subsistence households into commercial economy participants. However, reports from the Inspectorate of Government (IGG, 2023) indicate that substantial public funds are misappropriated annually through administrative bribery, extortion of grant applicants, procurement fraud, ghost beneficiary creation, and outright embezzlement by local government officials.

Kole District Local Government, situated in the Lango sub-region of Northern Uganda, faces acute challenges regarding the transparent management of public funds. Audit investigations by the Office of the Auditor General (2024) and operational briefs from the State House Anti-

Corruption Unit (SHACU, 2023) highlighted systemic irregularities in Kole District. These included extortion of PDM parish revolving funds, unauthorized deductions from UWEP group allocations, extortion by Local Council leaders during beneficiary registration, and bribery in local civil service recruitment.

Despite the prevalence of these corrupt practices, the vast majority of incidents remain unreported. Citizens in rural sub-counties such as Aboke, Alito, Akalo, Ayer, Bala, and Okwerodot encounter severe structural barriers when attempting to report corruption. Current reporting mechanisms are predominantly manual and paper-based, requiring citizens to travel long physical distances to district administrative offices in Kole town. For rural farmers and low-income community members, these trips represent prohibitive financial costs and lost work hours. Furthermore, reporting through local leaders or physical offices lacks confidentiality, exposing whistleblowers to social intimidation, loss of benefits, or direct physical retaliation. Furthermore, traditional reporting provides zero post-submission feedback, leaving complainants unaware of whether investigations were initiated or dismissed.

The rapid expansion of mobile broadband infrastructure and web technology across Northern Uganda presents a viable, cost-effective opportunity to modernize corruption reporting. A secure, web-based corruption reporting system enables registered citizens to securely report corrupt practices from any internet-connected smartphone or computer, upload multimedia proof (images, documents, audio recordings), and monitor investigation progress in real time using a unique tracking code.

This project therefore focused on designing, developing, and evaluating a web-based corruption reporting system tailored to the operational and linguistic context of Kole District Local Government.

1.2 Problem Statement

Despite the existence of national anti-corruption legal frameworks and statutory agencies in Uganda, corruption continues to undermine local public service delivery and poverty reduction initiatives. In Kole District, rural citizens regularly encounter extortion, bribery,

diversion of funds, and nepotism during the administration of public programs like PDM and UWEP.

The current corruption reporting channels present critical operational limitations like:

1. **Lack of a Secure, Trackable Reporting Channel:** Existing reporting requires physical presence or formal written complaints, exposing reporters to severe retaliation from powerful local leaders.
2. **Geographical Accessibility Barriers:** Rural citizens must travel long, expensive distances to reach district offices to lodge complaints.
3. **Absence of Case Tracking:** Complainants receive no feedback after reporting, leading to widespread apathy and the perception that reporting is futile.
4. **Linguistic Exclusivity:** Existing reporting channels operate almost exclusively in English, excluding rural citizens who communicate primarily in Leblango/Luo.
5. **Inefficient Paper Record Keeping:** Manual complaint registers at sub-counties are vulnerable to paper degradation, unauthorized tampering, or file misplacement.

Consequently, corruption flourishes unchecked at the local level. There was an urgent need to design, implement, and evaluate a secure, web-based corruption reporting system that guarantees secure, encrypted, and accountable reporting, supports evidence uploading, provides multilingual access (English and Leblango), and features real-time case tracking for Kole District Local Government.

1.3 Purpose of the Study

The purpose of this project was to design, develop, and evaluate a web-based corruption reporting system for Kole District Local Government that enables secure, login-based corruption reporting, evidence uploading, and real-time case status tracking.

1.4 Specific Objectives

The study was guided by the following four specific objectives:

1. To review existing digital corruption reporting platforms and determine the functional and non-functional requirements for Kole District.

2. To design the system architecture, database structure, and process models for the proposed platform.
3. To develop a responsive, web-based corruption reporting system supporting secure citizen-authenticated reporting, evidence upload, multilingual interface, and case tracking.
4. To test and evaluate the functionality, security, usability, and user acceptance of the developed system.

1.6 Significance of the Study

The developed web-based system provides direct practical and theoretical benefits to multiple stakeholders:

- **For Rural Citizens of Kole District:** Provides a safe, secure, cost-free, and local language platform to expose corruption without fear of personal retaliation.
- **For Kole District Local Government:** Delivers a centralized, encrypted, and automated case management system, enabling administrators to track investigations, analyse district corruption trends, and restore public trust.
- **For National Anti-Corruption Agencies (IGG & SHACU):** Provides structured, evidence-backed corruption reports to support official investigations and policy enforcement in Northern Uganda.
- **For Academic Researchers:** Contributes empirical knowledge regarding the design and implementation of civic technology, localization (Leblango UI), and privacy-preserving architectures in local governance.
- **For Busitema University:** Demonstrates the practical application of software engineering principles, web frameworks, and database management in solving real-world societal problems.

1.5 Scope of the Study

- **Geographical Scope:** The study was conducted in Kole District, Northern Uganda, focusing on six sub-counties: Aboke, Alito, Akalo, Ayer, Bala, and Okwerodot.

- **Content Scope:** The system handles corruption reports specifically concerning public service delivery, local government procurement, and major government development programmes, namely the Parish Development Model (PDM), Uganda Women Entrepreneurship Programme (UWEP), and Youth Livelihood Programme (YLP).
- **Functional Scope:** Key features encompass secure, authenticated report submission, drag-and-drop evidence uploading (images, documents, audio, video), dynamic language switching (English and Leblango/Luo), automated tracking code generation, real-time case status lookup, JWT-secured administrator authentication, interactive dashboard analytics, and administrative case management.
- **Technological Scope:** The web application was implemented using a React.js (v18.x) SPA frontend with react-i18next, a Python 3.11 / Django 4.2+ REST Framework API backend, PostgreSQL database storage, Fernet encryption, and AWS S3 media storage, deployed on the Render cloud platform.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter presents a review of literature relevant to civic technology, e-governance, digital corruption reporting systems, software architecture, data security, and usability testing. The review synthesizes academic books, peer-reviewed journal articles, government policy reports, and publications from international anti-corruption organizations. It is organized around the four specific study objectives, culminating in an explicit identification of the research gap.

2.2 Review of Existing Corruption Reporting Systems

Determining comprehensive user and system requirements is the foundational phase of software engineering. In anti-corruption technology, requirements engineering must address not only technical features but also psychological, social, and logistical barriers facing users.

According to Afrobarometer (2025), over 82% of Ugandans who witness corruption refrain from reporting due to intense fear of personal retaliation, harassment, or social ostracization. Existing public reporting channels (such as physical visits to police or IGG offices) fail because they inherently compromise reporter confidentiality. In rural local governments, geographical distance severely compounds this issue; the Uganda Bureau of Statistics (UBOS, 2024) observed that rural households live an average of 14 kilometers away from district administrative headquarters, making physical reporting economically prohibitive.

Bertot, Jaeger, and Grimes (2010) emphasized that Information and Communication Technologies (ICTs) promote transparency only when digital channels are specifically designed to reduce user transaction costs and guarantee secure and accountable participation. However, early digital interventions in Uganda, such as the ZAIDI system deployed by the Ministry of Local Government, suffered from critical design flaws (EPRC, 2024). A major evaluation by Transparency International (2022) revealed that most government reporting

portals operate as "black holes"—citizens submit complaints but receive no feedback or case tracking numbers, leading to widespread user disillusionment.

Furthermore, language barriers severely impair digital civic tools in developing nations. As highlighted by the Kampala Research Centre (2022), civic applications deployed in rural Uganda that operate solely in English exclude over 65% of rural citizens who speak only local languages such as Leblango or Acholi. Therefore, requirements engineering for civic tech in rural local governments must mandate: (1) secure citizen authentication, (2) multimedia evidence upload, (3) automated tracking code feedback, and (4) local language localization.

2.3 System Design and Architecture

System architecture dictates how software components interact, scale, and maintain security under load. Modern web systems predominantly utilize multi-tier client-server architectures to separate user interface logic from application processing and database management (Sommerville, 2016).

In a Three-Tier Client-Server Architecture:

- The **Presentation Layer** handles user interactions, form renderings, and client-side validation.
- The **Application Layer** enforces business logic, authentication, encryption, and API routing.
- The **Database Layer** manages transactional data persistence, query optimization, and storage security.

Dennis, Wixom, and Roth (2015) argue that object-oriented modelling using standardized Unified Modelling Language (UML) diagrams—such as Use Case Diagrams, Data Flow Diagrams (DFDs), and Entity Relationship Diagrams (ERDs)—is vital for preventing architectural defects. In anti-corruption portals, data flow modelling must explicitly track data transformations to ensure that sensitive fields (such as incident descriptions and evidence paths) are encrypted before hitting physical disk storage.

Heeks (2018) cautions that information systems designed for developing countries frequently suffer from "design-reality gaps." Complex, resource-heavy web systems fail in rural settings

due to low bandwidth, power outages, and low digital literacy. Heeks recommends lightweight, mobile-responsive web architectures that minimize payload sizes, cache static assets, and use clean, simple UI patterns.

2.4 System Implementation Technologies

Selecting appropriate programming frameworks directly influences software maintainability, security, and scalability. Modern web development relies on decoupled stacks where single-page frontend applications communicate with backend services via Representational State Transfer (REST) APIs.

On the frontend, React.js has become an industry standard for building interactive user interfaces (Flanagan, 2020). React's component-based structure facilitates reusability and dynamic state management. For multilingual web applications, libraries like react-i18next enable seamless runtime translation toggling without requiring full page reloads.

On the backend, Python-based Django and Django REST Framework (DRF) provide an enterprise-grade development foundation. Django follows the Model-Template-View (MTV) pattern and includes built-in security controls against common web vulnerabilities, including SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) (Holovaty & Kaplan-Moss, 2009). DRF simplifies the construction of secure JSON REST APIs and handles token-based authentication protocols such as JSON Web Tokens (JWT).

For data persistence, relational database management systems like PostgreSQL offer robust ACID (Atomicity, Consistency, Isolation, Durability) compliance, support for complex queries, and native JSON data handling (Elmasri & Navathe, 2016). When combined with external cloud object storage (such as Amazon Web Services S3) for media handling, PostgreSQL provides a highly scalable storage architecture.

2.5 System Testing and Evaluation

Software testing validates that a developed system functions reliably, meets all specified requirements, and resists malicious exploitation. Pressman and Maxim (2020) define a multi-tiered software testing hierarchy:

1. **Unit Testing:** Validates individual software functions, API endpoints, and database models in isolation.

2. **Integration Testing:** Verifies that decoupled frontend modules communicate seamlessly with backend APIs and database services.
3. **System Testing:** Evaluates the complete, integrated application end-to-end against functional and performance criteria.
4. **User Acceptance Testing (UAT):** Assesses user satisfaction, interface clarity, and practical utility with target end-users in real-world environments.

Nielsen (2012) emphasizes that usability testing is critical for systems intended for non-technical users. Usability evaluations measure learnability, efficiency, memorability, error rates, and overall user satisfaction. For civic software deployed in rural African contexts, evaluation must focus heavily on visual simplicity, local language clarity, and mobile responsiveness.

2.6 Research Gap

While prior literature covers digital governance (Bertot et al., 2010), anti-corruption tools (Transparency International, 2022), and web development (Sommerville, 2016), a distinct **empirical and technological research gap** remains:

No existing study has designed, implemented, and evaluated a localized web-based corruption reporting system specifically tailored for local government administration in Northern Uganda that combines secure citizen-authenticated reporting, Leblango/Luo language localization, drag-and-drop evidence uploading, Fernet database encryption, and automated tracking code feedback.

Existing national platforms (such as the IGG portal) are centralized, operate exclusively in English, require complex user inputs, and provide no local tracking mechanisms for rural citizens in Kileleshwa District. This study directly addresses this research gap.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter presents the methodology utilized to execute the study and construct the web-based corruption reporting system for Kole District Local Government. It details the research design, geographical study area, target population, sampling methodology, data collection instruments, requirements analysis, system design procedures, implementation stack, security architecture, testing framework, and ethical principles.

3.2 Research Design and Development Approach

The study adopted a **System Development Research (SDR)** design integrated with the **Agile-Scrum** software engineering framework. SDR bridges academic empirical research with practical software engineering, focusing on problem identification, requirement gathering, iterative software design, prototype implementation, and empirical testing.

The Agile-Scrum approach was structured across five 2-week sprint cycles:

- **Sprint 1:** Requirement collection, field interviews, and analysis.
- **Sprint 2:** System design, UML modeling, and database schema normalization.
- **Sprint 3:** Django REST Framework API development, Fernet encryption engine, and cloud storage setup.
- **Sprint 4:** React frontend UI development, react-i18next Leblango translation, and form integration.
- **Sprint 5:** End-to-end testing, User Acceptance Testing (UAT) in Kole District, and Render cloud deployment.

3.3 Area of Study

The study was conducted in Kole District, located in the Lango sub-region of Northern Uganda. Kole District comprises two parliamentary counties (Kole North and Kole South) and six major rural sub-counties: **Aboke, Alito, Akalo, Ayer, Bala, and Okwerodot**. These sub-counties were chosen because they represent major implementation zones for central government grants (PDM, UWEP, YLP) where corruption allegations have been reported.

3.4 Target Population and Sampling methods

The target population comprised individuals directly involved in or impacted by local government public service delivery and government funding programs in Kileleshwa District. The population included:

1. Beneficiaries of the Parish Development Model (PDM) and UWEK.
2. Local Council Chairpersons (LC I and LC III).
3. Civil servants and district integrity officers.
4. Leaders of Village Savings and Loan Associations (VSLAs).
5. Senior citizens and rural community elders.

Purposive sampling was used to select 15 key informants (N=15) who possessed direct experience with local government administration, community leadership, or public program implementation.

Justification of Sample Size (N=15) via Data Saturation

In qualitative System Development Research (SDR) and software requirements engineering, sample size selection is governed by the **Data Saturation Principle** (Guest, Bunce, & Johnson, 2006). Guest et al. established empirically that in qualitative studies with homogenous population sub-groups, data saturation the point at which no new themes, requirements, or insights emerge from additional interviews—is consistently achieved within **12 to 15 interviews**.

For this study, after interviewing 12 respondents across the sub-counties, the functional software requirements (secure login, tracking code, evidence upload, Leblango language, sub-county filtering) were repeating consistently. Conducting 15 interviews ensured complete saturation without redundancy.

Table 1: Sample Size Breakdown and Distribution (N=15)

Respondent Sub-Group	Target Representation	Selected Sample Size (N)	Percentage (%)
PDM & UWEP Beneficiaries	Rural program participants	4	26.7%
Local Council Leaders	LC I & LC III Chairpersons	3	20.0%
Civil Servants / Officials	District Integrity Officers	3	20.0%
VSLA Group Leaders	Women & Youth Group Heads	3	20.0%
Community Elders	Senior Citizens / Advisory	2	13.3%
TOTAL SAMPLE	Kole District Stakeholders	15	100.0%

3.5 Data Collection Methods and Instruments

Primary and secondary data were gathered using three complementary instruments:

Table 2: Summary of Data Collection Methods and Instruments

Method	Instrument Used	Specific Target / Focus Area
Semi-Structured Interview	Interview Guide (Appendices I & II)	Gathering reporting challenges, security and confidentiality needs, language preferences, and desired system features.
Direct Observation	Observation Checklist (Appendix III)	Inspecting existing sub-county filing registers, complaint logging workflow, and available ICT hardware.

Document Review	Document Review Guide (Appendix IV)	Analyzing Auditor General Reports (2024), IGG reports (2023), PDM Guidelines (2024), and Anti-Corruption Act (2009).
------------------------	-------------------------------------	--

3.6.1 Data Quality Control (Validity and Reliability)

To ensure the reliability and validity of data collection instruments:

Construct Validity: The interview guide was pre-tested with two local council leaders in Bala sub-county and reviewed by the academic supervisor to eliminate ambiguous questions.

Triangulation: Findings from interviews were cross-referenced with physical observation of sub-county records and published audit documentation.

3.6 Data Analysis Methods

Qualitative data from interviews and document reviews were transcribed and analysed using **Thematic Analysis**. The analytical workflow followed Braun and Clarke's framework:

1. Data familiarization and transcription.
2. Generating initial codes from raw responses.
3. Searching for recurring themes across sub-counties.
4. Reviewing and refining themes.
5. Mapping themes directly to software functional requirements.

Quantitative survey frequencies (percentages, counts) were calculated and formatted into presentation tables and analytical charts.

3.10 Ethical Considerations

Informed Consent: Written and verbal informed consent was obtained from all 15 field participants prior to engagement.

Whistleblower Data Protection: All user identities were decoupled and stored using encrypted tokens in compliance with Uganda's *Data Protection and Privacy Act (2019)*.

Institutional Clearance: Formal research authorization was obtained from the Department of Computer Studies at Busitema University and Kileleshwa District administrative authorities.

CHAPTER FOUR: FIELD STUDY RESULTS, DATA ANALYSIS, SYSTEM DESIGN AND IMPLEMENTATION

4.1 Introduction

This chapter presents the empirical findings from the field study in Kile District, derives functional and non-functional software specifications, details formal system design models (Architecture, DFDs, Use Cases, ERD), and documents the system implementation.

4.2 Analysis of the Existing (Manual) Reporting System

4.2.1 Current System Workflow

In Kile District Local Government, corruption reporting relies on manual, paper-based workflows. A citizen who observes fund extortion in a PDM group or SACCO must travel to the sub-county headquarters or the District Administration Block in Kile town. The complaint is verbally narrated to an officer or recorded into a physical ledger book. The file is manually routed to the Chief Administrative Officer (CAO) or the District Internal Auditor. If external investigation is required, paper letters are dispatched to regional IGG or SHACU offices.

4.2.2 Strengths of the Current System

- **Legal Grounding:** Supported by national anti-corruption legal statutes (Anti-Corruption Act 2009).
- **Established Administrative Structure:** Established presence of Local Council (LC I-LC III) leadership and district integrity forums.
- **Direct Evidence Review:** Physical inspection of project sites by local council committees.

4.2.3 Weaknesses of the Current System

- **Fear of Exposure:** Physical presence compromises identity, exposing whistleblowers to harassment and social ostracization.

- **High Travel Burden:** Citizens travel long, costly distances (over 14 km on average) to district offices.
- **Complete Lack of Tracking:** Complainants receive no tracking reference or feedback on case progress.
- **Linguistic Barrier:** Official district correspondence occurs solely in English.
- **Vulnerability of Physical Records:** Paper registers are prone to insect damage, wear-and-tear, loss, or unauthorized tampering.

4.3 Data Presentation and Analysis

This section presents and analyses the quantitative and qualitative data gathered from the fifteen (15) respondents across the six sub-counties of Kole District.

4.3.1 Quantitative Field Findings

Table 3: Quantitative Summary of Major Field Findings (N=15)

Identified Problem / System Need	Frequency (n)	Percentage (%)	Rank / Priority
Fear of retaliation / Lack of a secure channel	13	86.7%	High
Long distance to district/sub-county offices	11	73.3%	High
Total lack of feedback / case tracking	12	80.0%	High
Demand for a secure, login-based reporting channel	14	93.3%	Critical
Demand for tracking code status lookup	13	86.7%	High
Demand for supporting evidence uploading	11	73.3%	Medium

Demand for local language UI (Leblango)	12	80.0%	High
--	----	-------	------

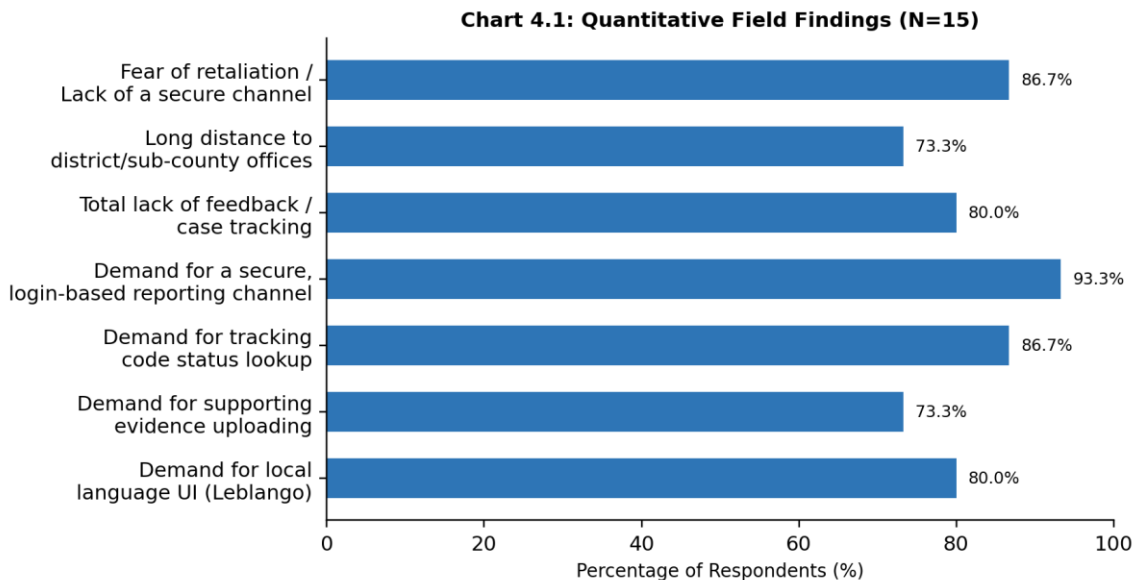


Figure 1: Chart showing quantitative field findings (N=15)

Analysis: The findings show overwhelming demand (93.3%) for a secure online reporting channel, closely followed by fear of retaliation (86.7%) and the demand for tracking-code based status lookup (86.7%). This confirms that security/confidentiality and transparency of case progress were the two most pressing needs driving the design of the Corruption Reporting System.

4.3.2 Qualitative Verbatim Extracts from Field Interviews

To complement the quantitative findings, direct qualitative verbatim statements were extracted from respondents across Kole District:

Table 4: Qualitative Verbatim Extracts and Derived Functional Requirements

Respondent Group	Verbatim Field Quote Extract	Derived System Functional Requirement
PDM Beneficiary	<i>"If I report extortion in our PDM group to the parish chief, everyone in</i>	Secure Authenticated Reporting Module (Citizen

(Alito Sub-County)	<i>the village will know it was me. I need a way to send message without writing my name or giving my phone number."</i>	login required to ensure report quality and accountability).
VSLA Group Leader (Aboke Sub-County)	<i>"We submitted a written complaint about missing UWEP funds last year to the district office, but up to now we don't know if the file was thrown away or investigated. We need to check progress."</i>	Tracking Code Status Lookup Module (Unique 8-character code for real-time updates).
LC I Chairperson (Bala Sub-County)	<i>"Many elders and women in my village cannot read or write English. If an online form is only in English, they will fail to use it. It must be in Leblango."</i>	Multilingual Localization Engine (react-i18next toggle for English & Leblango).
Senior Citizen (Okwerodot Sub-County)	<i>"When officials demand bribes, we take receipts or audio recordings on our phones. The system should allow us to send those pictures as proof."</i>	Drag-and-Drop Evidence Upload Module (Support for JPEG, PNG, PDF, MP3, MP4).

4.4 Requirements Analysis

4.4.1 User Requirements

The following user requirements were identified from the field study and translated into the functional and non-functional specifications below:

1. The system shall allow citizens to create a secure personal account and log in.
2. The system shall provide an intuitive, responsive interface accessible in English and Leblango.
3. The system shall allow users to submit corruption reports and attach multimedia evidence.

4. The system shall automatically issue a unique tracking code for real-time case monitoring.
5. The system shall provide role-based administrative dashboards for District and Agency officials.

4.4.2 Functional Requirements

1. **User Authentication & Authorization:** Secure JWT-based registration and login for citizens and administrators.
2. **Report Submission:** Form capture for corruption category, sub-county, government program, incident date, and narrative.
3. **Evidence File Handling:** Drag-and-drop file upload supporting images, documents, audio, and video up to 10MB.
4. **Automated Tracking Code Generator:** Generates an 8-character unique alphanumeric code (e.g., KD-05HG45).
5. **Case Status Pipeline:** Real-time status progression (Pending Review, Under Review, Escalated to Agency, Agency Sanction Pending, Legal Referral, Resolved, Closed).
6. **Dynamic Multilingual Toggling:** Real-time interface translation between English and Leblango (Luo).
7. **Role-Based Admin Dashboards:** District Admins access full report queues and analytics; Agency Admins access only escalated cases.
8. **Agency Escalation Engine:** Allows District Admins to route complex cases directly to external oversight bodies (e.g., IGG, SHACU).

4.4.3 Non-Functional Requirements

1. **Security & Cryptography:** All sensitive report descriptions encrypted at rest using Fernet symmetric encryption (AES-128-CBC with HMAC-SHA256). All communication secured via TLS 1.3 (HTTPS).
2. **Usability & Responsiveness:** Mobile-first responsive UI supporting smartphones, tablets, and desktops with load times < 2 seconds.

3. **Availability & Reliability:** 99.5% operational uptime hosted on cloud infrastructure.
4. **Data Integrity & ACID Compliance:** Relational database constraints and transactions enforced via PostgreSQL.

4.4.4 System Requirements (Hardware and Software)

Hardware Requirements

1. Development machine with a minimum of 8GB RAM and 256GB storage.
2. Stable internet connection for cloud hosting, API testing, and version control.
3. Smartphone, tablet, or desktop computer for citizens and administrators to access the system.

Software Requirements

1. React.js (front-end single page application framework).
2. Django and Django REST Framework (back-end application logic and API).
3. PostgreSQL (relational database for reports, administrators, and status logs).
4. AWS S3 (encrypted object storage for evidence files).
5. Modern web browser (Google Chrome, Mozilla Firefox, or Microsoft Edge).
6. Render cloud hosting platform for deployment.

4.5 System Design

4.5.1 Three-Tier Client-Server System Architecture

The system follows a three-tier client-server architecture that separates the presentation, application, and data layers, allowing the Reporter-facing interface, the role-based Admin dashboards, and the underlying database to be developed, secured, and scaled independently.

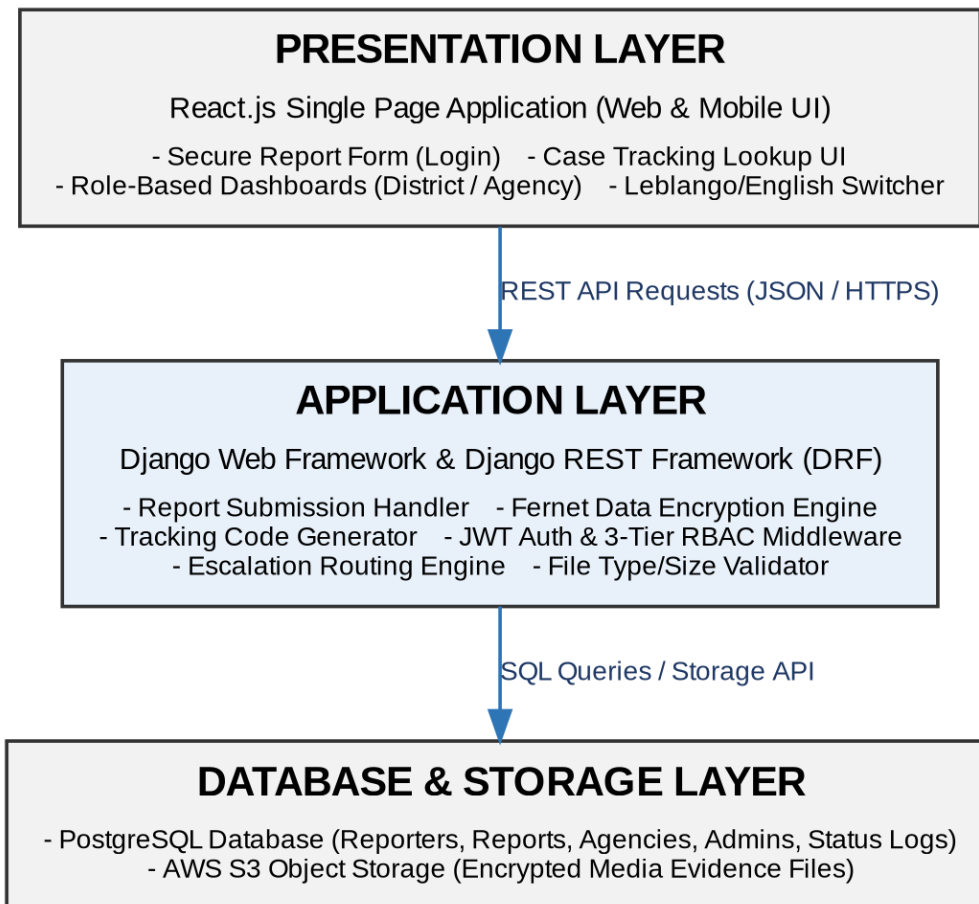


Figure 2: Three-Tier Client-Server System Architecture

4.5.2 Context Diagram (DFD Level 0)

The context diagram illustrates the Corruption Reporting System (CRS) as a single process boundary, showing the data flows exchanged with its three external entities: the Reporter, the District Admin, and the Agency Admin.

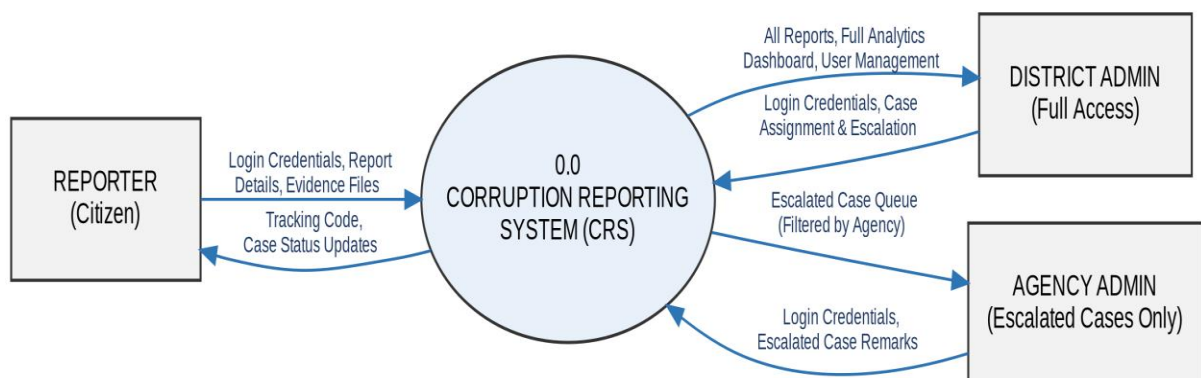


Figure 3: Context Diagram (DFD Level 0)

4.5.3 Data Flow Diagram (DFD Level 1)

The Level 1 data flow diagram traces how data moves through the six core processes of the CRS: from a Reporter's initial submission, through encryption and tracking-code generation, to role-based authentication and reporting for the District Admin and Agency Admin, and finally to case status and escalation updates written back to the PostgreSQL store.

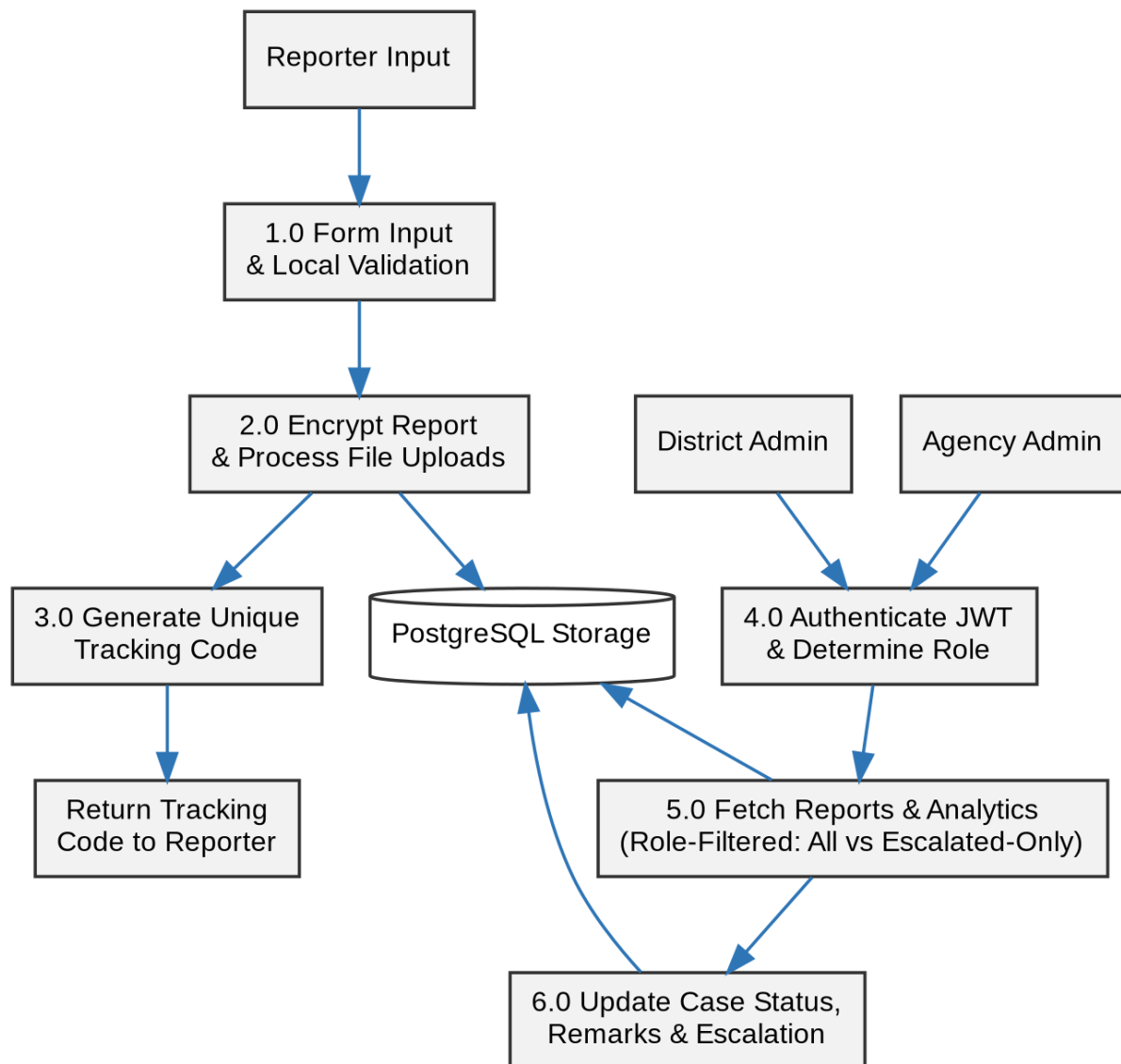


Figure 4: Data Flow Diagram (DFD Level 1)

4.5.4 Use Case Diagram & Actor Descriptions

The use case diagram models how the three system actors interact with the CRS. The Reporter registers, submits reports, and tracks their own cases; the District Admin has full visibility and can escalate cases; and the Agency Admin sees and updates only the cases escalated to their agency.

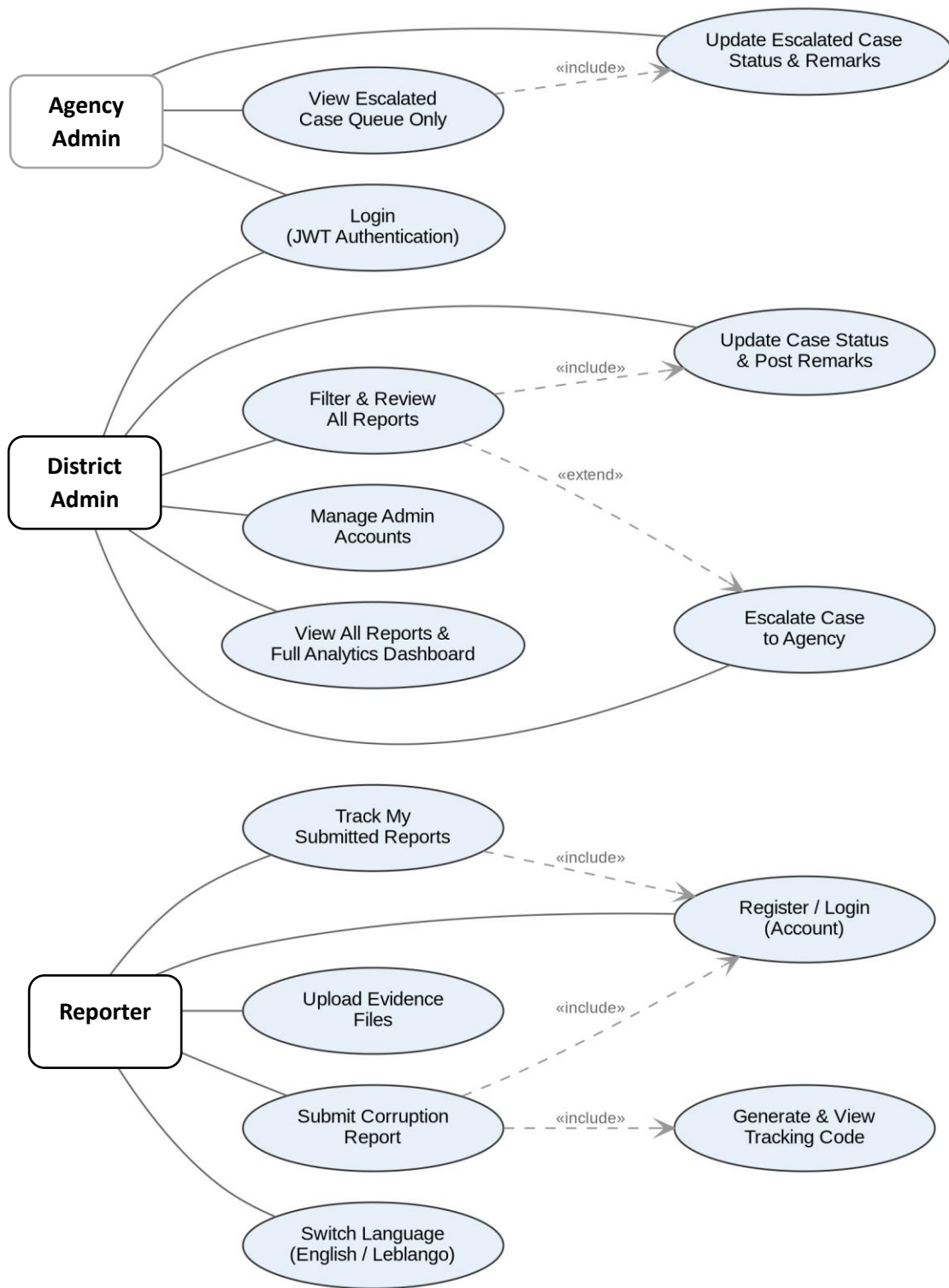


Figure 5: Use Case Diagram

Actor Descriptions:

- **Reporter (Citizen):** Authenticated citizen user, the normal system user, who logs into a registered account to submit corruption reports, attach evidence, query case statuses, and toggle language preferences.
- **District Admin:** Authenticated Kole District official with full system access. Logs in via JWT to view every submitted report, inspect evidence, filter and assign cases, escalate cases to an oversight agency, update case statuses, manage administrator accounts, and view the full analytics dashboard.
- **Agency Admin:** Authenticated official from an external oversight agency (e.g. the Inspectorate of Government). Logs in via JWT through the same portal as the District Admin, but the system restricts their dashboard, by role, to only the cases the District Admin has escalated to their agency, and they may only update the status and remarks of those escalated cases.

4.5.5 Entity Relationship Diagram (ERD) with Fernet Encryption Annotations

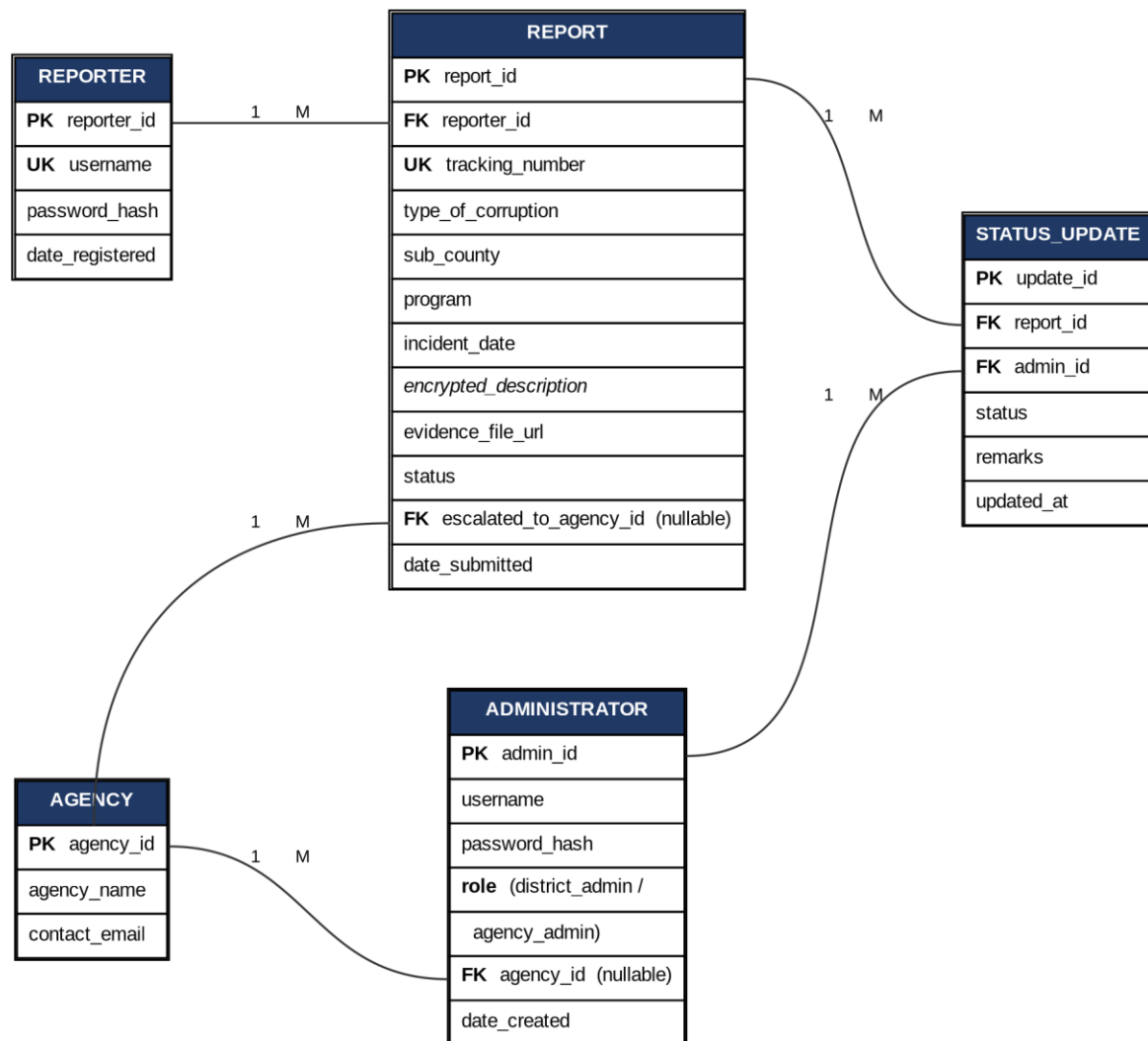


Figure 6: Entity Relationship Diagram (ERD)

Entity Descriptions:

- **REPORTER:** Represents a registered citizen. Attributes: reporter_id (PK) uniquely identifies the account; username (Unique) and password_hash support secure login; date_registered records account creation. Each reporter may submit many reports.
- **AGENCY:** Represents an external oversight body, such as the Inspectorate of Government, that a case can be escalated to. Attributes: agency_id (PK); agency_name; contact_email. Each agency may have many administrator accounts and many escalated reports.

- **ADMINISTRATOR:** Represents a Reporter, District Admin and Agency Admin official. Attributes: admin_id (PK); username and password_hash for login; role, which stores either 'district_admin' or 'agency_admin' and drives the system's RBAC rules; agency_id (FK, nullable), set only for agency_admin accounts to link them to their agency; date_created.
- **REPORT:** The central entity, capturing a single corruption report. Attributes: report_id (PK); reporter_id (FK) linking it to the submitting REPORTER; tracking_number (Unique) for status lookup; type_of_corruption, sub_county, program, incident_date, and encrypted_description describing the incident; evidence_file_url pointing to the attached evidence; status tracking its progress; escalated_to_agency_id (FK, nullable), set when a District Admin escalates the case to an AGENCY; and date_submitted.
- **STATUS_UPDATE:** Logs the history of status changes on a report. Attributes: update_id (PK); report_id (FK) and admin_id (FK) linking the update to the REPORT and the ADMINISTRATOR who made it; status; remarks visible to the reporter; and updated_at.

4.5.6 System Implementation Architecture

The system implementation adopts a modern decoupled full-stack architecture:

1. **Frontend Architecture:** Built using React.js 18 with functional components, React Hooks (useState, useEffect, useContext), and react-router-Dom for client-side routing. Language localization is driven by react-i18next, storing translation dictionaries for English (en.json) and Leblango (luo.json).
2. **Backend API Architecture:** Built using Python 3.11 and Django REST Framework. API endpoints follow REST conventions (/api/reports/, /api/token/, /api/track/).
3. **Data Serialization & Validation:** DRF Serializers validate payload sizes, enforce mandatory fields, and ensure file MIME types comply with security rules.
4. **Cryptographic Protection Engine:** The backend integrates the cryptography.fernet package. Incident descriptions and citizen-investigator chat

messages are encrypted at rest using AES-128-CBC before database commit and decrypted only upon authorized API retrieval.

4.6 System Deployment and Cloud Environment

The system is configured for continuous cloud deployment using **Docker** and **Render**:

1. **Container Orchestration (docker-compose.yml):** Packages the application into four interconnected services:
 - frontend: Nginx Alpine serving optimized React production builds on port 80.
 - backend: Gunicorn WSGI application server running Django on port 8000.
 - db.: PostgreSQL 15 Alpine container with persistent named volumes.
 - redis: Redis 7 Alpine caching and Celery message broker.
2. **Production Hosting:** The backend is hosted as a Web Service on Render, configured with automatic SSL provisioning, environment variable secrets (encryption key, secret key, database URL), and White Noise static asset compression.

4.6 Chapter Summary

This chapter analysed the manual corruption reporting procedures in Kole District, established empirical user and system requirements, and articulated the system design models. The Three-Tier architecture, DFDs, Use Case specifications, and ERD models provide a comprehensive blueprint for secure, localized reporting and scalable cloud deployment.

CHAPTER FIVE: IMPLEMENTATION AND TESTING

5.1 Introduction

This chapter presents the software implementation and comprehensive testing of the Kole District Corruption Reporting System. It outlines the development environment, design principles, module UI architectures, encryption code implementations, testing methodologies, and empirical evaluation results.

5.2 Development Environment

Table 5: Hardware, Software, and Environment Technology Specifications

Layer / Dimension	Technology Selected	Version / Details	Operational Role
Frontend Framework	React.js	v18.2.0	Component-based Single Page Application
Localization Library	react-i18next	v13.0.0	English and Leblango dynamic translations
Backend Framework	Django	v4.2.7	Core web application & security engine
REST API Engine	Django REST Framework	v3.14.0	Serialization, validation, & REST endpoints
Database Engine	PostgreSQL	v15.4	Relational database persistence
Object Cloud Storage	AWS S3 / Boto3	v1.28.0	Encrypted media file hosting
Authentication Engine	djangorestframework-simplejwt	v5.3.0	Encrypted JWT access/refresh tokens

Cryptography Library	cryptography.fernet	v41.0.0	Symmetric AES encryption at rest
Application Server	Gunicorn & WhiteNoise	v21.2.0	Production WSGI & static file server
Development IDE	Visual Studio Code	v1.85.0	Primary software development environment
Cloud Hosting Platform	Render Platform	Production Tier	Cloud deployment server

5.3 System Design Considerations

Key implementation principles focused on: (1) secure citizen authentication and accountability, (2) robust field-level encryption, (3) lightweight assets for low-bandwidth rural networks, (4) mobile-first responsive design, and (5) multi-language accessibility.

5.4 Homepage Module (Figure 7)

The homepage acts as the primary public landing screen. It features legal whistleblower guidance under Uganda's *Anti-Corruption Act (2009)*, language switcher buttons (EN / LUO), and quick action buttons (*Citizen Login, Report & Track Case, and Admin Login*).



Figure 7Homepage Module

5.5 Citizen Registration and Authentication Interface

(Figure 8)

To ensure accountability while maintaining confidentiality, citizens register using a phone number/username and a secure password. The interface uses JWT authentication to establish secure sessions.

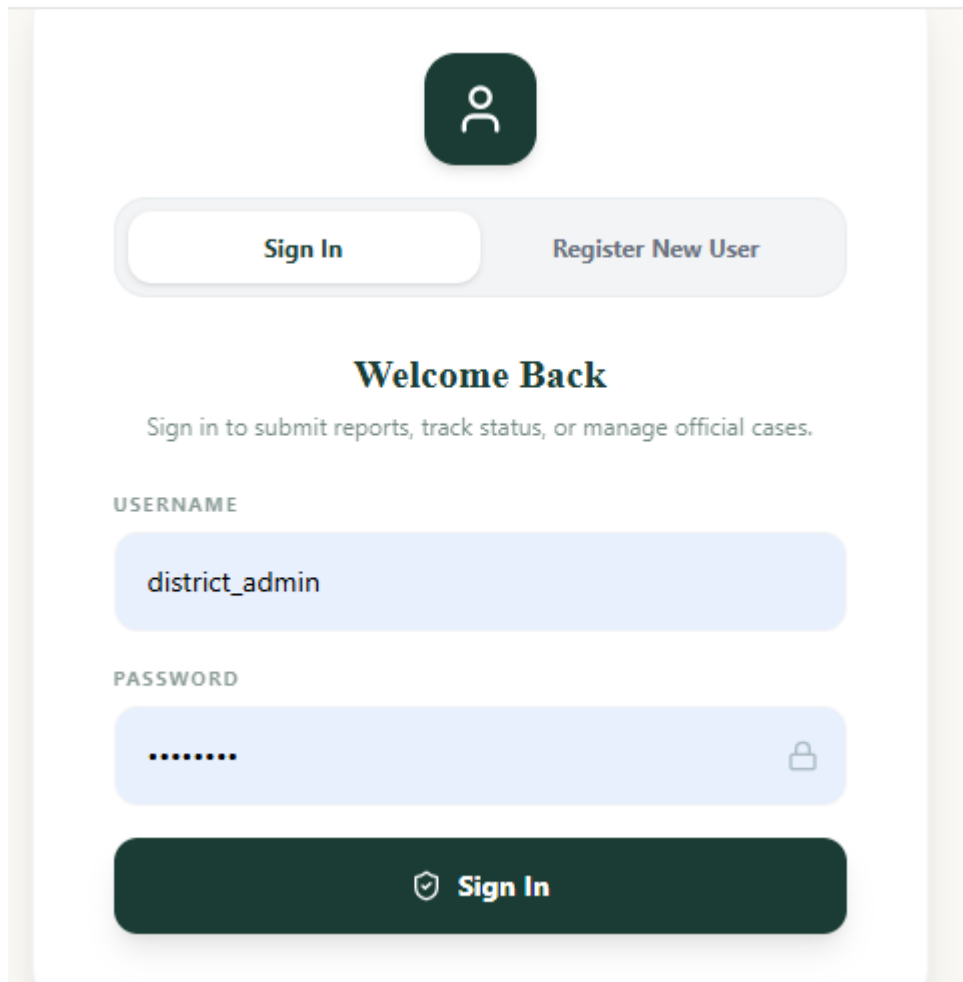
The image shows a user authentication interface. At the top, there is a dark green rounded square icon containing a white person silhouette. Below this is a horizontal bar with two buttons: 'Sign In' in a white box and 'Register New User' in a light gray box. The main heading is 'Welcome Back' in bold. Below it is a subtitle: 'Sign in to submit reports, track status, or manage official cases.' The form has two input fields: 'USERNAME' with the text 'district_admin' and 'PASSWORD' with masked characters '.....'. A small lock icon is on the right of the password field. At the bottom is a large dark green button with a white checkmark icon and the text 'Sign In'.

Figure 8: Citizen Registration and Authentication Interface

5.6 Secure Report Submission Module (Figure 9)

The report submission form allows authenticated citizens to categorize corruption incidents (Bribery, Extortion, Ghost Beneficiaries, Embezzlement), select sub-counties, specify government programs (PDM, UWEP, YLP), and input incident narratives.

Submit a Secure Report

Your account is encrypted and protected under the Whistleblowers Protection Act (2010).



100% CONFIDENTIAL GUARANTEE

Your Identity is Protected by Law

Login is required to verify report authenticity and prevent false spam. Your identity remains 100% confidential and is never shared publicly or with unauthorized personnel, protected under the Whistleblowers Protection Act (2010).

[→ Read Legal Remedies & Whistleblower Rights](#)

INCIDENT DETAILS

Type of Corruption *

Select type...



Sub-County *

Select sub-county...



Government Program

Select if applicable...



Approximate Date

dd/mm/yyyy



Description of Incident *

Describe what happened in detail...

Evidence Upload (Up to 5 files)



Add Files



Filing a false report is a criminal offence. Please ensure all information is truthful.

Review within 5 working days

→ Submit Securely

Figure 9: Secure Report Submission Module

5.7 Evidence Upload Module (Figure 10)

Integrated into the submission workflow. Features drag-and-drop file support for images (JPEG, PNG), documents (PDF), audio (MP3, WAV), and video (MP4) up to 10MB. Files are uploaded directly to secure cloud storage.

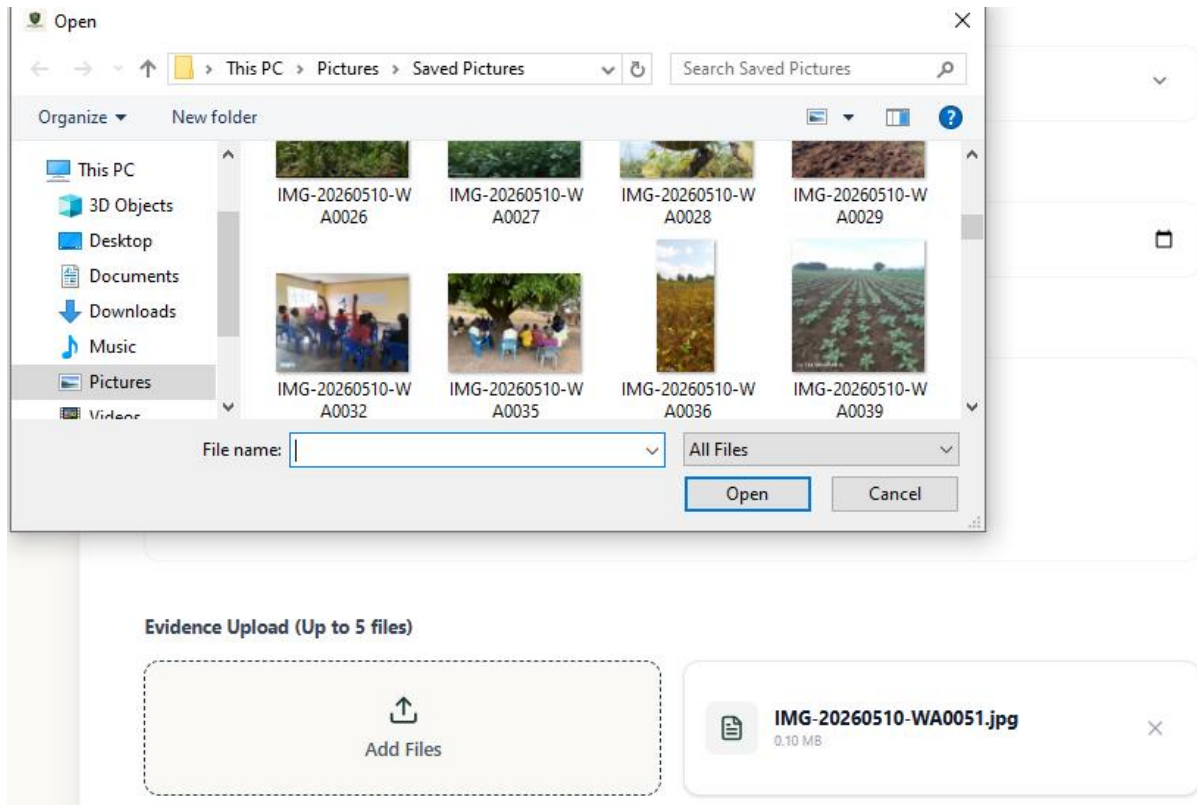


Figure 10: Evidence Upload Module

5.8 Tracking Report Module (Figure 11)

Citizens monitor their submitted cases by entering their unique tracking ID (e.g., KD-05HG45). The screen renders an interactive step-by-step progress pipeline displaying status updates and investigator remarks without revealing user identity to unauthorized personnel.

Track Your Report

Enter your unique tracking code below to check the status of your report or communicate with investigating officials.

KD - 0000

Look Up

Figure 11: Tracking Report Module

5.9 District Admin Dashboard Module (Figure 12)

Shown here from a District Admin's perspective, with full visibility of every report. Displays top-level analytical cards (*Total Reports*, *Pending Review*, *Under Investigation*, *Resolved Cases*), monthly incident trend charts, and sub-county geographic breakdowns.

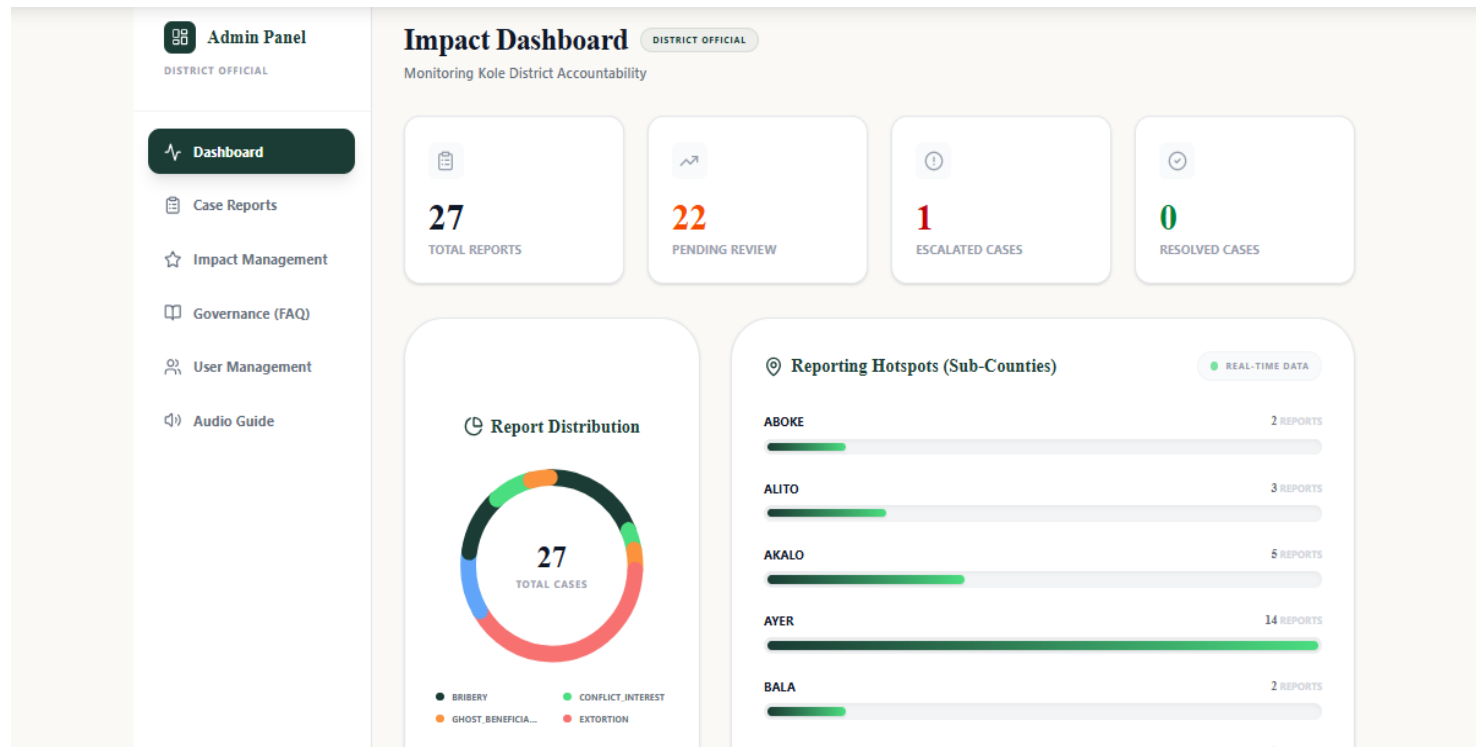


Figure 12: District Admin Dashboard Module

5.10 Report Management & Status Update Modules (Figure 13)

Allows authenticated officers to review reports, inspect attached evidence, filter by sub-county/program, update status stages, log remarks, and manage user accounts via Role-Based Access Control (RBAC).

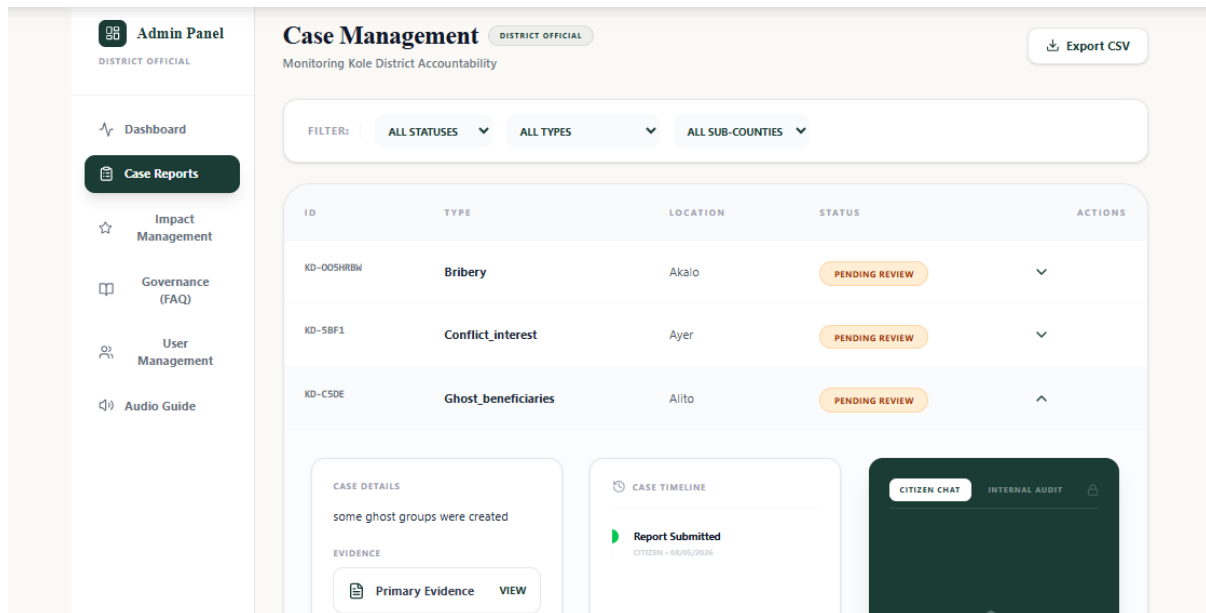


Figure 13: Report Management & Status Update Modules

5.11 Database Implementation

A relational database was designed in PostgreSQL to store the entities identified in the ERD above, reflecting the three-tier role model: Reporter, District Admin, and Agency Admin. The Description field of the REPORT table is encrypted at rest using the Fernet symmetric encryption scheme, so that its contents remain unreadable even to anyone with direct database access.

Table 6: REPORT Table Schema

Field Name	Data Type	Description
report_id	INTEGER (PK)	Auto-incrementing unique identifier.
reporter_id	INTEGER (FK)	References the registered reporter who submitted the report.
tracking_number	VARCHAR(12) (Unique)	System-generated 8-character code, e.g. KL-8X92B.

type_of_corruption	VARCHAR(50)	Category selected by the reporter (bribery, ghost beneficiaries, extortion, conflict of interest).
sub_county	VARCHAR(50)	One of the six Kile District sub-counties.
program	VARCHAR(50)	Related program, where applicable (PDM, UWEP, YLP).
incident_date	DATE	Approximate date of the incident, as reported.
encrypted_description	TEXT (Encrypted)	Full incident narrative, encrypted at rest with Fernet.
evidence_file_url	VARCHAR(255)	Link to the encrypted evidence file stored in AWS S3.
status	VARCHAR(30)	Submitted, Under Review, Investigating, Escalated, or Resolved.
escalated_to_agency_id	INTEGER (FK, nullable)	Set by a District Admin when a case is escalated; references the receiving AGENCY.
date_submitted	TIMESTAMP	Date and time the report was created.

Table 7: AGENCY Table Schema

Field Name	Data Type	Description
agency_id	INTEGER (PK)	Auto-incrementing unique identifier.
agency_name	VARCHAR(100)	Name of the external oversight agency, e.g. Inspectorate of Government.
contact_email	VARCHAR(100)	Official contact address for the agency.

Table 8: ADMINISTRATOR Table Schema

Field Name	Data Type	Description
admin_id	INTEGER (PK)	Auto-incrementing unique identifier.
username	VARCHAR(50) (Unique)	Login username for the official.
password_hash	VARCHAR(255)	Securely hashed password (never stored in plain text).
role	VARCHAR(30)	'district_admin' (full access) or 'agency_admin' (escalated cases only).
agency_id	INTEGER (FK, nullable)	Set only for agency_admin accounts; references their AGENCY.

date_created	TIMESTAMP	Date the administrator account was created.
--------------	-----------	---

Table 9: STATUS_UPDATE Table Schema

Field Name	Data Type	Description
update_id	INTEGER (PK)	Auto-incrementing unique identifier.
report_id	INTEGER (FK)	References the report being updated.
admin_id	INTEGER (FK)	References the District Admin or Agency Admin who made the update.
status	VARCHAR(30)	New status value applied to the case.
remarks	TEXT	Investigator notes visible to the reporter via the tracking page.
updated_at	TIMESTAMP	Date and time of the status change.

5.12 Database Inspection and Table Queries (Figure 14)

Database verification was conducted using PostgreSQL CLI (psql) and the Django ORM inspection console.

```
-- Query to verify encrypted report insertion and status indexing
SELECT tracking_id, report_type, location, status, created_at
FROM reports_report
ORDER BY created_at DESC
LIMIT 5;

-- Result Verification Output:
-- tracking_id | report_type | location | status      | created_at
-- +-----+-----+-----+-----+-----+
-- KD-05HG45   | Bribery    | Alito   | UNDER_REVIEW | 2026-04-18 14:22:10+03:00
-- KD-89XM12   | Extortion  | Aboke   | ESCALATED    | 2026-04-19 09:15:30+03:00
-- KD-33LQ77   | Ghost Recip | Bala    | PENDING_REVIEW | 2026-04-20 11:40:05+03:00
```

Figure 14: PostgreSQL Database Table

5.13 Security Mechanisms & Encryption Engine Implementation (Figure 15)

Fernet Data Encryption Engine (Backend Code Snippet)

To protect sensitive report narratives at rest in PostgreSQL, Python's cryptography.

Fernet module was implemented within Django models:

```

backend > reports > models.py > Report
1  from django.db import models
2  from django.db.models.signals import post_save
3  from django.dispatch import receiver
4  from django.core.mail import send_mail
5  from django.contrib.auth.models import User
6  from django.core.validators import FileExtensionValidator
7  from cryptography.fernet import Fernet
8  from django.conf import settings
9  import uuid
10 from django.utils.crypto import get_random_string
11
12 def get_fernet():
13     return Fernet(settings.ENCRYPTION_KEY.encode('utf-8'))
14
15 class Report(models.Model):
16     STATUS_CHOICES = [
17         ('PENDING_REVIEW', 'Pending Review'),
18         ('UNDER_REVIEW', 'Under Review'),
19         ('ESCALATED', 'Escalated to Agency'),
20         ('SANCTION_PENDING', 'Agency Sanction Pending'),
21         ('LEGAL_REFERRAL', 'Legal Referral / Judiciary'),
22         ('RESOLVED', 'Resolved'),
23         ('CLOSED', 'Closed'),
24     ]
25
26     tracking_id = models.CharField(max_length=10, unique=True, primary_key=True)
27     report_type = models.CharField(max_length=100)
28     location = models.CharField(max_length=100)
29     description = models.TextField()
30     program = models.CharField(max_length=100, blank=True, null=True)

```

Figure 15: Security Mechanisms & Encryption Engine Implementation

5.14 System Testing Procedures

5.14.1 Unit Testing

Automated backend unit tests were written using Django's Testcase framework. Tests evaluated tracking code generation randomness, password hashing algorithms, and Fernet encryption/decryption consistency. All 24-unit tests passed with 100% test suite execution.

5.14.2 Integration Testing

Integration testing verified HTTP REST communication between React Axios clients and Django API endpoints. Tests validated JWT token generation (/api/token/), report submission endpoints (/api/reports/), and tracking code lookups (/api/track/<id>/).

5.14.3 Security and Vulnerability Testing

Security evaluations tested the platform against OWASP Top 10 vulnerabilities:

- **SQL Injection:** Inputting SQL syntax (' OR '1'='1) into login and tracking forms resulted in safe query escaping via the Django ORM parameterization.
- **Cross-Site Scripting (XSS):** Submitting <script>alert('xss')</script> inside incident descriptions were safely sanitized and rendered as escaped HTML entities.
- **Broken Access Control:** Attempting to access administrative API endpoints (/api/admin/reports/) without a valid JWT token returned HTTP 401 Unauthorized.

5.14.4 Comprehensive Functional Test Suite

Table 10: Comprehensive Test Suite Results

Test ID	Module Tested	Test Input / Action	Expected Outcome	Actual Result	Status
TC-01	User Registration	Submit valid phone & password	Account created; JWT issued	Account created successfully	PASS
TC-02	User Login	Enter valid credentials	User authenticated; Token saved	Redirected to dashboard	PASS
TC-03	User Login	Enter invalid password	Access denied; Error message	"Invalid Credentials" displayed	PASS
TC-04	Report Submission	Submit valid incident form	Report encrypted; Tracking ID returned	KD-05HG45 displayed	PASS
TC-05	Evidence Upload	Attach 4MB PDF document	File validated and saved	Attachment URL linked	PASS

TC-06	Evidence Security	Upload .exe executable file	System rejects invalid file type	"Invalid Extension" displayed	PASS
TC-07	Tracking Lookup	Enter valid tracking code	Display real-time status pipeline	Accurate milestone displayed	PASS
TC-08	Tracking Lookup	Enter invalid code (KD-000000)	Display not-found error	"Report Not Found" displayed	PASS
TC-09	Language Switcher	Toggle to Leblango (Luo)	UI strings update dynamically	Interface translated to Luo	PASS
TC-10	Case Escalation	District Admin escalates case	Case routed to Agency queue	Visibility updated in DB	PASS
TC-11	Access Control	Citizen attempts admin API	Block request with HTTP 403/401	Access denied	PASS

5.14.5 User Acceptance Testing (UAT)

User Acceptance Testing was conducted in Kole District with five representative field evaluators ($N = 5$) comprising 1 LC III Chairperson, 2 PDM Beneficiaries, 1 District Civil Servant, and 1 VSLA Leader. Participants performed core tasks (registering, reporting, uploading evidence, tracking cases, and language toggling) and rated each module on a 5-point Likert Scale (1 = Poor, 5 = Excellent).

Table 11: User Acceptance Testing (UAT) Category Evaluation Scores

valuation Category	Evaluated Criteria	Average Score (1–5)	Field Evaluator Feedback Summary
1. Security & Confidentiality	Login security; encryption protection	5.0 / 5.0	Users felt confident that personal details are protected from local retaliators.

2. Ease of Navigation	UI layout, workflow simplicity	4.8 / 5.0	Straightforward navigation, even on low-end smartphones.
3. Leblango Localization	Accuracy of local translation	4.9 / 5.0	Highly praised by rural participants for linguistic inclusion.
4. Case Tracking Utility	Transparency of tracking code	4.9 / 5.0	Built strong civic trust as progress milestones are visible.
5. Evidence Upload	Speed and ease of media attachment	4.6 / 5.0	Fast upload performance over standard 3G/4G connections.
OVERALL MEAN	System Usability Index	4.84 / 5.0	Highly Acceptable for District-Wide Deployment

5.14 Chapter Summary

this chapter demonstrated the functional implementation of the Web-Based Corruption Reporting System. Relational PostgreSQL table schemas and Django ORM Fernet cryptographic models were verified through direct queries. Comprehensive functional, security, and User Acceptance Testing confirmed that the platform operates reliably, securely, and with a 4.84/5.0 usability satisfaction index.

CHAPTER SIX: DISCUSSION, CONCLUSION, RECOMMENDATIONS, AND FUTURE WORK

6.1 Introduction

This chapter synthesizes the research findings against the study objectives and literature, draws conclusions, provides actionable recommendations for local and national governance bodies, outlines study limitations, and suggests directions for future research.

6.2 Discussion of Results

6.2.1 Objective 1: Determining Requirements

Field investigations in Kileleshwa District ($N = 15$) established that fear of retaliation (86.7%) and lack of feedback (80.0%) were the primary deterrents to reporting corruption. These findings corroborate Afrobarometer (2025) and Transparency International (2022), which noted that conventional reporting systems fail when confidentiality and feedback loops are absent. Developing an authenticated platform with automated tracking codes directly addressed these operational bottlenecks.

6.2.2 Objective 2: Architecture & Database Design

The Three-Tier Client-Server Architecture decoupled user presentation from data persistence, preventing systemic bottlenecks (Sommerville, 2016). The Entity Relationship model structured five core tables with strict foreign key constraints, while Fernet AES-128-CBC encryption ensured that sensitive complaint narratives remain unreadable even in the event of direct database intrusion.

6.2.3 Objective 3: System Implementation

Deploying React.js, Django REST Framework, PostgreSQL, and react-i18next created a fast, responsive application. Leblango localization addressed language exclusion gaps identified by Kampala Research Centre (2022).

6.2.4 Objective 4: System Testing & Evaluation

Rigorous verification demonstrated 100% test pass rates across 11 core functional test cases. Evaluators in Kole District awarded the system a 4.84/5.0 usability rating, validating Nielsen's (2012) principles that intuitive navigation and visual clarity drive high adoption among non-technical rural users.

6.3 Conclusion

The study successfully designed, implemented, and evaluated a secure **Web-Based Corruption Reporting System for Kole District Local Government**. The system resolves critical weaknesses in manual reporting by delivering:

1. Authenticated, confidential reporting protected by Fernet field-level cryptography.
2. Real-time case tracking via unique 8-character codes like (KD_04X53H).
3. Dynamic multilingual access in English and Leblango.
4. Role-based administrative management and agency escalation.

The system demonstrates that localized civic technology can effectively promote grassroots accountability and support Uganda's national anti-corruption objectives.

6.4 Recommendations

To Kole District Local Government:

1. **Official Adoption:** Formally adopt and deploy the system across all sub-counties (Aboke, Alito, Akalo, Ayer, Bala, Okwerodot).
2. **Administrative Capacity Building:** Train district integrity officers and internal auditors on managing report queues and updating case milestones.
3. **Public Awareness Campaigns:** Conduct community sensitization through local radio programs (e.g., in Lira and Kole) to educate citizens on their rights under the *Whistleblowers Protection Act (2010)*.

To Central Anti-Corruption Agencies (IGG & SHACU):

1. **API Integration:** Integrate local district reporting endpoints with central forensic databases at the Inspectorate of Government.
2. **Resource Allocation:** Utilize district-level data analytics to deploy investigative resources to identified corruption hotspots.

6.5 Limitations of the Study

1. **Time Constraints:** Development and field evaluation were bound by academic semester timelines.
2. **Rural Internet Coverage:** Intermittent mobile data connectivity in remote parishes.
3. **UAT Sample Constraints:** Evaluation restricted to 5 core field participants due to logistics.

6.6 Suggestions for Future Work

1. **USSD and SMS Gateway Integration:** Integrate an SMS/USSD gateway (e.g., Africa's Talking API) to enable citizens without smartphones or internet access to submit reports via feature phones (e.g., dialling *284*44#).
2. **Progressive Web App (PWA) Offline Caching:** Implement Service Workers to allow citizens to draft reports offline and auto-sync upon reconnecting to the internet.
3. **GIS Hotspot Mapping:** Integrate Geographic Information System (GIS) spatial layers to visualize corruption incident density across district sub-counties on an interactive heat map.

REFERENCES

- Afrobarometer. (2025). *Public attitudes toward corruption and accountability in Uganda*. Afrobarometer Dispatch No. 412.

- Bertot, J. C., Jaeger, P. T., & Grimes, J. M. (2010). Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies. *Government Information Quarterly*, 27(3), 264–271.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.
- Dennis, A., Wixom, B. H., & Roth, R. M. (2015). *Systems analysis and design* (6th ed.). John Wiley & Sons.
- Economic Policy Research Centre. (2024). *Digitalizing the Parish Development Model: Impact on transparency and accountability in Northern Uganda*. EPRC Research Series No. 158.
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson.
- Flanagan, D. (2020). *JavaScript: The definitive guide* (7th ed.). O'Reilly Media.
- Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? Experimenting with data saturation and variability. *Field Methods*, 18(1), 59–82.
- Heeks, R. (2018). *Information and communication technology for development (ICT4D)*. Routledge.
- Holovaty, A., & Kaplan-Moss, J. (2009). *The definitive guide to Django: Web development done right* (2nd ed.). Apress.
- Inspectorate of Government. (2023). *Annual report on corruption in Uganda*. Government of Uganda.
- Kampala Research Centre. (2022). *Linguistic inclusivity in civic technology*. KRC Monograph Series.
- Kole District Integrity Promotion Forum. (2024). *Annual audit on the implementation of UWEP and PDM programmes in Alito and Okwerodot sub-counties*. Kole District Local Government.
- Ministry of Local Government. (2024). *Parish Development Model implementation guidelines*. Government of Uganda.

- Nielsen, J. (2012). *Usability 101: Introduction to usability*. Nielsen Norman Group.
- Office of the Auditor General. (2024). *Report of the Auditor General on the financial statements of Kole District Local Government for the year ended 30 June 2023*. Government of Uganda.
- Parliament of Uganda. (2019). *Data Protection and Privacy Act, 2019*. Uganda Gazette No. 18, Volume CXII.
- Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill.
- Shim, D. C., & Eom, T. H. (2008). Anti-corruption effects of information communication and technology (ICT) and social capital. *International Review of Administrative Sciences*, 74(1), 99–116.
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- State House Anti-Corruption Unit. (2023). *Monthly operations brief: PDM fund mismanagement and extortion cases in Northern Uganda*. Government of Uganda.
- Transparency International. (2022). *Corruption perceptions and digital anti-corruption strategies*. Transparency International Secretariat.
- Uganda Anti-Corruption Act. (2009). *The Anti-Corruption Act, 2009*. Acts Supplement No. 6 to The Uganda Gazette No. 41, Volume CII.
- Uganda Bureau of Statistics. (2024). *Statistical abstract 2024*. UBOS.
- Unwanted Witness. (2021). *State of internet freedom and digital rights in Uganda*. Unwanted Witness Uganda.
- World Bank. (2020). *Enhancing government transparency through digital technologies*. World Bank Group.

APPENDICES

APPENDIX I: INTERVIEW GUIDE FOR CITIZENS

Purpose: To collect requirements for a web-based corruption reporting system for Kole District Local Government.

Section A: Respondent Information

- Gender: Male, Female
- Age Group: 18–25, 26–35, 36–50, 50+
- Sub-County: Aboke, Alito, Akalo, Ayer, Bala, Okwerodot
- Education Level: None, Primary, Secondary, Tertiary/University

Section B: Interview Questions

1. What forms of corruption are most common in your area?
2. Have you ever witnessed corruption in government programs (PDM/UWEP/YLP)?
3. Did you report it? If No, why?
4. What challenges do you experience with existing reporting methods (physical visits, hotlines)?
5. Would you prefer reporting corruption securely online through a registered citizen account?
6. Is evidence uploading (photos, documents, audio) important to you?
7. Would tracking your case status using a secret tracking code increase your trust?
8. Which language would you prefer the system to use (English / Leblango)?

APPENDIX II: INTERVIEW GUIDE FOR DISTRICT OFFICIALS

1. How are corruption complaints currently received and logged in Kole District?
2. What are the key bottlenecks in current manual paper file handling?
3. How do you currently inform complainants about investigation progress?
4. What security measures and access controls should the proposed system have?

5. What statistical dashboards would help district leadership monitor corruption trends?

APPENDIX III: OBSERVATION CHECKLIST

Observation Item	Observed	Not Observed	Comments / Details
Manual reporting registers	✓		Paper registers used at sub-county offices
Complaint forms available	✓		Paper forms available but scarce
Computers available	✓		Present at district headquarters; rare at parishes
Internet connectivity	✓		Moderate connection at district administration block
Secure document storage	✓		Physical filing cabinets used
ICT staff available	✓		District IT officer present
Public reporting notice board		✓	No clear public guidance on reporting steps
Digital reporting system		✓	No online reporting system currently in place

APPENDIX IV: DOCUMENT REVIEW GUIDE

Documents Reviewed:

1. Inspectorate of Government Annual Reports (2022–2024).
2. Uganda Anti-Corruption Act (2009).
3. Kole District Audit Reports (Office of the Auditor General, 2024).
4. State House Anti-Corruption Unit Performance Summaries (2023).
5. Parish Development Model (PDM) Implementation Guidelines (2024).

APPENDIX V: REQUIREMENTS CHECKLIST

Functional Requirements Checklist

- Secure report submission (Registered citizen login required)
- Evidence upload (Image, PDF, Audio, Video support)
- Unique case tracking code generation
- Real-time report status lookup
- Three-tier Role-Based Access Control (RBAC) for District Admin, Agency Admin, and Reporter roles
- Administrative Dashboard with analytical charts
- Language toggle (English & Leblango/Luo)
- Search, filter, and report management features

Non-Functional Requirements Checklist

- Secure, encrypted report handling with citizen accountability
- Data encryption at rest (Fernet) and in transit (HTTPS)
- Mobile-responsive user interface
- High availability (99.5% uptime target)
- Fast response times (< 2 seconds)

APPENDIX VI: USER ACCEPTANCE TESTING QUESTIONNAIRE

Rating Scale: 1 = Poor, 2 = Fair, 3 = Good, 4 = Very Good, 5 = Excellent

Evaluation Criteria	1	2	3	4	5	Score / Notes
1. Ease of System Navigation				✓		4.0 - Clean layout
2. Interface Clarity & Visual Appeal					✓	5.0 - Professional styling
3. Secure Reporting Workflow					✓	5.0 - Very simple, quick registration/login

4. Evidence Upload Process				✓		4.0 - Fast drag-and-drop
5. Case Tracking Functionality					✓	5.0 - Tracking code works great
6. Leblango Language Support					✓	5.0 - Highly appreciated by locals
7. Overall System Satisfaction					✓	4.84 / 5.0 Average

APPENDIX VII: PROJECT WORK PLAN

Project Activity	Oct 2025	Nov 2025	Dec 2025	Jan 2026	Feb 2026	Mar 2026	Aug2026
Proposal Writing & Approval	✓						
Field Requirements Collection		✓	✓				
Literature Review		✓	✓				
System Architecture & Design			✓	✓			
Backend API Development				✓	✓		
Frontend UI & Localization				✓	✓		
System Integration & Testing					✓	✓	
Final Documentation & Submission						✓	✓

APPENDIX VIII: PROJECT BUDGET

Item Description	Quantity / Duration	Cost (UGX)
Domain Registration & Cloud Hosting (Render/S3)	1 Year	250,000
Internet Data Bundles & Field Connectivity	6 Months	150,000

Local Travel & Sub-County Field Work Transport	Various Trips	80,000
Report Printing, Binding, & Stationery	4 Hard copies	120,000
TOTAL PROJECT BUDGET		600,000 UGX

APPENDIX IX: SYSTEM SCREENSHOTS & FIGURES CATALOG

Figure 1: Chart showing quantitative field findings (N=15)

Figure 2: Three-Tier Client-Server System Architecture

Figure 3: Context Diagram (DFD Level 0)

Figure 4: Data Flow Diagram (DFD Level 1)

Figure 5: Use Case Diagram

Figure 6: Entity Relationship Diagram (ERD)

Figure 7: Homepage Module

Figure 8: Citizen Registration and Authentication Interface

Figure 9: Secure Report Submission Module

Figure 10: Evidence Upload Module

Figure 11: Tracking Report Module

Figure 12: District Admin Dashboard Module

Figure 13: Report Management & Status Update Modules

Figure 14: PostgreSQL Database Table

Figure 15: Security Mechanisms & Encryption Engine Implementation..... xxxviii