

BUSITEMA UNIVERSITY

FACULTY OF SCIENCE AND EDUCATION

DEPARTMENT OF COMPUTER STUDIES

FINAL YEAR PROJECT REPORT

A WEB-BASED INTERNSHIP MANAGEMENT SYSTEM

By

ANGOLE SHARIF ABUBAKAR

**This Final Year Project Report is submitted to the Department of
Computer Studies, Faculty of Science and Education in Partial
Fulfillment of the Requirement for the Award of the Degree of
Bachelor of Science in Computer Science of Busitema University**

AUGUST 2026



FINAL YEAR PROJECT REPORT

A WEB-BASED INTERNSHIP MANAGEMENT SYSTEM

By

ANGOLE SHARIF ABUBAKAR

(BU/UG/2023/2240)

SUPERVISORS

DR. LUKYAMUZI ANDREW

MR. MULIKA GERALD

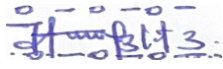
AUGUST 2026

ABSTRACT

The management of internship programmes at Busitema University, Nagongera Campus, has historically depended entirely on paper-based logbooks, infrequent physical field visits, and manual administrative record-keeping. This manual approach produces systemic inefficiencies including document loss and forgery, the absence of real-time student monitoring, prohibitively costly field supervision, and no structured channel through which students can confidentially report challenges encountered at placement sites. This study aimed to develop a Web-Based Internship Management System (WIMS) for Busitema University, Nagongera Campus, enabling students to submit geo-tagged daily logs, allowing field and university supervisors to provide structured remote feedback and monitor cohort-wide progress in real time, and equipping administrators with placement management and institutional reporting capabilities. The study adopted a mixed-methods research design combining qualitative and quantitative approaches within a Rapid Application Development (RAD) framework from the Structured Systems Analysis and Design Methodology (SSADM). Requirements were gathered through semi structured interviews and structured questionnaires administered to 49 purposively and randomly selected participants across four stakeholder categories. Implementation used React 18 with TypeScript for the frontend, Node.js with Express.js for the backend REST API, and SQLite as the embedded relational database, deployed on Vercel and Railway hosting platforms. Key findings revealed that supervisors were unable to effectively monitor student progress between logbook submissions, while students identified document loss and the inability to report placement challenges as major limitations of the current system. The study also established that students were generally ready to adopt a digital solution due to their familiarity with smartphone technology. The proposed system was successfully validated through unit, integration, functional, and user acceptance testing, demonstrating that it met the specified functional requirements and achieved satisfactory usability. These results indicate that the system is suitable for adoption at Busitema University, Nagongera Campus.

DECLARATION

I, ANGOLE SHARIF ABUBAKAR, with registration number BU/UG/2023/2240, hereby declare that this project report is original and has not been published or submitted for any other degree award to any other university before.

Sign:  Date: 24/08/2026

ANGOLE SHARIF ABUBAKAR

APPROVAL

This is to acknowledge that this project report is for Angole Sharif Abubakar, Registration Number BU/UG/2023/2240. It was done under our supervision and was completed successfully.

Sign:  Date: 26/08/2026

DR. LUKYAMUZI ANDREW

Department of Computer Studies, Faculty of Science and Education

Sign:  Date: 31/08/2026

MR. MULIKA GERALD

Department of Computer Studies, Faculty of Science and Education

DEDICATION

I dedicate this report to **Allah (SWT)**, the Almighty, my Creator, my Sustainer, and the source of my strength, wisdom, understanding, and guidance throughout my academic journey. I am deeply grateful to Allah for granting me the strength, patience, knowledge, and perseverance to reach this stage of my education.

I dedicate this work with profound love and gratitude to my dear mother, **Auma Rhema Abubakar**, whose love, prayers, sacrifices, and unwavering support have been a constant source of strength throughout my life and academic journey. I also dedicate it to my beloved siblings: my elder sister, **Acen Zhurah Abubakar**; my elder brother, **Opio Ismail Abubakar**; and my younger brothers, **Abdul Salam Abubakar** and **Yasin Abubakar**, for their love, encouragement, and support.

I dedicate this work in loving memory of my late father, **Mr. Ocen Abubakar Sediq**, whose memory, values, and love continue to inspire me. I am especially grateful to my uncle, **Lieutenant David Odyek Ebong**, for the immense support, care, and guidance he extended to me and my family, particularly following the passing of my father. His support has been an important part of my academic journey.

I also sincerely dedicate this work to **Acen Nusura Ramathan** and her family, who welcomed and embraced us with kindness and compassion. Following the passing of my father, Acen Nusura Ramathan became a great source of care, encouragement, and support to me and my family. Through her love, kindness, and continued support, I have come to regard her as a mother in my life. I remain deeply grateful to her and her family for standing by us during an important and difficult chapter of our lives.

My sincere appreciation also goes to **Mr. Alula Thomas**, whose support and encouragement have contributed meaningfully to my academic journey. I am equally grateful to my supervisors, **Dr. LUKYAMUZI Andrew** and **Mr. Mulika Gerald**, for their guidance, patience, constructive criticism, and invaluable support throughout this project. Finally, I dedicate this work to the **entire Computer Science class**, with whom I have shared the challenges, experiences, lessons, and achievements of this academic journey. I thank Allah (SWT) for blessing me with wonderful people who have contributed, directly or indirectly, to my growth and success.

May Allah (SWT) bless them all abundantly, reward their kindness and sacrifices, grant mercy to my late father, and grant everyone continued success, good health, peace, and happiness. Ameen.

ACKNOWLEDGEMENT

First and foremost, I thank **Allah (SWT)** for granting me life, good health, strength, wisdom, guidance, and perseverance throughout my academic journey and for enabling me to successfully complete this project. Without His blessings and guidance, this achievement would not have been possible.

I extend my sincere appreciation to my supervisors, **Dr. Lukyamuzi Andrew** and **Mr. Mulika Gerald**, for their invaluable guidance, patience, constructive criticism, insightful advice, and continuous support throughout the development of this project. Dr. LUKYAMUZI Andrew played a significant role in guiding me through the design, development, and documentation of the system, while Mr. Mulika Gerald provided valuable methodological guidance that helped shape the development process and overall quality of this work. I am deeply grateful for the time and expertise they invested in this project.

I would also like to express my sincere appreciation to the **students, field supervisors, university supervisors, and administrative staff of Busitema University, Nagongera Campus**, who willingly participated in interviews, questionnaires, consultations, and user acceptance testing sessions. Their cooperation, valuable feedback, and willingness to share their experiences greatly contributed to the successful development and evaluation of the system.

My heartfelt appreciation also goes to my **family, friends, classmates, and everyone who supported me throughout this academic journey**, whether through encouragement, advice, prayers, technical assistance, or other forms of support. Their contributions, both big and small, have been invaluable.

Finally, I am grateful to everyone who contributed directly or indirectly to the completion of this project. May **Allah (SWT)** reward you abundantly for your kindness, support, and contribution, and may He grant you all continued success, good health, peace, and blessings. **Ameen.**

TABLE OF CONTENTS

DECLARATION	ii
APPROVAL	iii
DEDICATION	iv
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	x
LIST OF TABLES	xi
ABBREVIATIONS/ACRONYMS	xii
ABSTRACT	i
CHAPTER ONE: INTRODUCTION	1
1.0 Introduction	1
1.1 Background of Study	1
1.2 Problem Statement	2
1.3 Main Objective	3
1.4 Specific Objectives	3
1.5 Research Questions	3
1.6 Significance of the Study	3
1.7 Scope of the Study	4
1.7.1 Content Scope	4
1.7.2 Geographical Scope	4
1.7.3 Time Scope	4
CHAPTER TWO: LITERATURE REVIEW	5
2.0 Introduction	5
2.1 Literature Related to Objective 1: Requirements Identification	5
2.1.1 Internship Programmes and Institutional Supervision	5
2.1.2 Digital Logbook and Internship Management Platforms.....	5
2.1.3 Requirements for Developing a Web-Based Internship Management System	5
2.2 Literature Related to Objective 2: System Design	6
2.2.1 Web-Based Platforms for Institutional Programme Management	6
2.2.2 Integrated Multi-Stakeholder Digital Platforms	6

2.3 Literature Related to Objective 3: System Implementation	6
2.4 Literature Related to Objective 4: System Testing and Evaluation.....	7
2.5 Research Gap	7
2.6 Advantages of the Web-Based Internship Management System	7
2.7 Features of the Web-Based Internship Management System	8
CHAPTER THREE: METHODOLOGY	9
3.0 Introduction	9
3.1 Research Design	9
3.2 Area of Study	9
3.3 Population and Sampling Method	9
3.4 Data Collection Methods and Analysis	10
3.5 System Development	10
3.6 Design	10
3.7 Implementation	11
3.8 Testing and Validation	11
3.9 Ethical Considerations	12
CHAPTER FOUR: FIELD STUDY AND SYSTEM DESIGN	13
4.0 Introduction	13
4.1 Background Information of the Current System	13
4.2 System Study and Analysis	13
4.2.1 Findings from Interviews and Questionnaires	13
4.3 User Requirements	14
4.3.1 Functional Requirements	14
4.3.2 Non-Functional Requirements	14
4.4 Hardware / Software Requirements	15
4.4.1 Hardware Requirements.....	15
4.4.2 Software Requirements	15
4.5 System Design	15
4.5.1 The System Architecture	15
4.5.2 Context Diagram (Level 0 DFD)	16
4.5.3 Data Flow Diagram (Level 1)	17
4.5.4 UML Diagram (Use-Case Diagram)	18
4.5.5 The Entity Relationship Diagram (ERD)	20
CHAPTER FIVE: IMPLEMENTATION AND TESTING	22

5.0 Introduction	22
5.1 Interface Design	22
5.1.1 Landing Page	22
5.1.2 Student Registration and Login Interfaces	24
5.1.3 Student Dashboard — Overview	25
5.1.4 Student Dashboard — Weekly Log Submission Form	26
5.1.5 Student Dashboard — Challenge Reporting	27
5.1.6 Student Dashboard — Portfolio Management	28
5.1.7 Student Dashboard — Profile Management	29
5.1.8 Field Supervisor Dashboard — Main View.....	30
5.1.9 Field Supervisor Dashboard — Student List	31
5.1.10 Field Supervisor Dashboard — Log Review Interface	31
5.1.11 Field Supervisor Dashboard — Feedback Submission	32
5.1.12 University Supervisor Dashboard — Overview	33
5.1.13 University Supervisor Dashboard — Student Monitoring.....	34
5.1.14 University Supervisor Dashboard — Challenge Management	34
5.1.15 University Supervisor Dashboard — Reports View	35
5.1.16 University Supervisor Dashboard — Analytics	35
5.1.17 Administrator Dashboard — Main Overview.....	36
5.1.18 Administrator Dashboard — User Management	37
5.1.19 Administrator Dashboard — Placement Management	37
5.1.20 Administrator Dashboard — Analytics Dashboard	38
5.1.21 Administrator Dashboard — Department Overview	38
5.1.22 Administrator Dashboard — System Reports.....	39
5.1.23 Administrator Dashboard — Settings	39
5.1.24 Administrator Dashboard — Audit Log	40
5.2 Data Storage	40
5.2.1 USERS	41
5.2.2 STUDENTS	41
5.2.3 LOGBOOK_ENTRIES.....	41
5.2.4 FEEDBACK	41
5.2.5 CHALLENGES	41
5.2.6 PLACEMENTS	41
5.3 System Testing	42

5.3.1 Unit Testing	42
5.3.2 Integration Testing	42
5.3.3 Functional Testing	42
5.3.4 Non-Functional Testing	42
5.3.5 User Acceptance Testing	42
CHAPTER SIX: DISCUSSION, CONCLUSION, RECOMMENDATION AND	
FUTURE WORK	44
6.0 Introduction	44
6.1 Discussion	44
6.2 Conclusion	45
6.3 Recommendations	45
6.4 Limitations and Future Work	45
REFERENCES	47
APPENDIX	48
Appendix I: Requirements Collection Interview Guide	
Appendix II: Questionnaire for Students	
Appendix III: Analysis of the Interview Data	

LIST OF FIGURES

Figure 4-1. System Architecture Diagram	16
Figure 4-2. Context Diagram (Level 0 DFD)	17
Figure 4-3. Data Flow Diagram (Level 1)	18
Figure 4-4. Use Case (UML) Diagram	19
Figure 4-5. Entity Relationship Diagram (ERD)	20
Figure 5-1. Landing Page — Hero Section	23
Figure 5-2. Landing Page — Platform Analytics	23
Figure 5-3. Landing Page — Powerful Features	23
Figure 5-4. Landing Page — Built for Every Stakeholder	24
Figure 5-5. Student Registration Page	25
Figure 5-6. Student Login Page	25
Figure 5-7. Student Dashboard — Overview	26
Figure 5-8. Student Dashboard — Weekly Log Submission Form	27
Figure 5-9. Student Dashboard — Challenge Reporting	28
Figure 5-10. Student Dashboard — Portfolio Management	29
Figure 5-11. Student Dashboard — Profile Management	30
Figure 5-12. Field Supervisor Dashboard — Main View	31
Figure 5-13. Field Supervisor Dashboard — Student List	31
Figure 5-14. Field Supervisor Dashboard — Log Review Interface	32
Figure 5-15. Field Supervisor Dashboard — Feedback Submission	33
Figure 5-16. University Supervisor Dashboard — Overview	34
Figure 5-17. University Supervisor Dashboard — Student Monitoring	34
Figure 5-18. University Supervisor Dashboard — Challenge Management	35
Figure 5-19. University Supervisor Dashboard — Reports View	35
Figure 5-20. University Supervisor Dashboard — Analytics	36
Figure 5-21. Administrator Dashboard — Main Overview	37
Figure 5-22. Administrator Dashboard — User Management	37
Figure 5-23. Administrator Dashboard — Placement Management	38
Figure 5-24. Administrator Dashboard — Analytics Dashboard	38
Figure 5-25. Administrator Dashboard — Department Overview	39
Figure 5-26. Administrator Dashboard — System Reports	39
Figure 5-27. Administrator Dashboard — Settings and Configuration	40
Figure 5-28. Administrator Dashboard — Audit Log	40

LIST OF TABLES

Table 3-1. Population Sampling Summary	9
Table 3-2. Technology Stack and Justification	11
Table 4-1. Functional Requirements	14
Table 4-2. Non-Functional Requirements	14
Table 4-3. Hardware Requirements	15
Table 4-4. Software Requirements	15
Table 4-5. DFD Process Descriptions	18
Table 5-1. User Acceptance Testing Results Summary	42
Table A-1. Summary of the Interview Data Analysis	49

ABBREVIATIONS/ACRONYMS

API – Application Programming Interface

CRUD – Create, Read, Update, Delete

CSS – Cascading Style Sheets

DFD – Data Flow Diagram

ERD – Entity Relationship Diagram

GPS – Global Positioning System

HTTPS – HyperText Transfer Protocol Secure

ICT – Information and Communication Technology

JWT – JSON Web Token

KPI – Key Performance Indicator

NDP III – National Development Plan III

RAD – Rapid Application Development

REST – Representational State Transfer

SDG – Sustainable Development Goal

SQL – Structured Query Language

SQLite – Self-Contained, Serverless, Zero-Configuration SQL Database Engine

SSADM – Structured Systems Analysis and Design Methodology

SUS – System Usability Scale

UAT – User Acceptance Testing

UUID – Universally Unique Identifier

WIMS – Web-Based Internship Management System

CHAPTER ONE: INTRODUCTION

1.0 Introduction

Internship programmes constitute a fundamental pillar of higher education pedagogy, bridging theoretical knowledge acquired in lecture halls with practical competencies demanded by the modern workplace. At Busitema University, Nagongera Campus, undergraduate students in Computer Science, Education Sciences, and Agriculture undertake mandatory industrial placements spanning eight to twelve weeks annually, deploying hundreds of students to organisations and schools across Uganda's diverse geographical regions.

Despite the centrality of these programmes to the University's academic mission, their entire administration relies on paper-based processes. Students maintain physical logbooks recording daily activities, supervisor remarks, and hours worked. Field supervisors sign logbooks periodically, while university supervisors undertake costly field visits to verify student presence and performance. At placement end, students physically return logbooks to the department where staff manually compile and archive records.

This manual approach is beset with systemic inefficiencies: lost and forged logbooks, infrequent and prohibitively expensive field visits, the complete absence of real-time student monitoring, and no structured mechanism for students to report challenges at placement sites. Institutions in Nigeria, Kenya, Malaysia, and Indonesia have demonstrated that web-based platforms can replicate and surpass the functions of physical documentation with measurable improvements in supervisor engagement, student accountability, and administrative efficiency (Ololade, 2021; Gosal et al., 2024).

This study therefore aimed to develop a Web-Based Internship Management System (WIMS) for Busitema University, Nagongera Campus, enabling students to submit geotagged daily logs, allowing supervisors to provide structured remote feedback, and equipping administrators with real-time cohort analytics and placement management tools. This chapter covers the background, problem statement, objectives, research questions, significance, and scope of the study.

1.1 Background of Study

Internship programmes have long been recognised as managed transitions from academic to professional environments requiring deliberate institutional design, clear learning objectives, and robust evaluation mechanisms (Sides & Mrvica, 2017). Di Pietro (2022) demonstrates that structured placements develop critical competencies including

intercultural communication, professional adaptability, and disciplinary application, and that institutional documentation is essential for evidencing these competencies for graduate employability. Maertz Jr et al. (2014) provide empirical evidence that supervision quality has the strongest marginal effect on intern learning outcomes, and that absent structured feedback mechanisms, intern disengagement and voluntary withdrawal increase significantly.

The limitations of paper-based internship management are well documented globally. Ololade (2021) documents the Electronic Log Book System for Nigeria's SIWES programme addressing data forgery, data loss, and supervisor overload but lacking GPS verification, challenge-reporting, and analytics. Gosal et al. (2024) describe a more sophisticated web-based system for FTI UNTAR in Indonesia that automated log submission, supervisor grading, and report generation, reducing compilation time from days to near-real-time. In Uganda and East Africa, digital tools for managing student internships remain virtually absent from the documented literature — a gap this study directly addresses within a policy environment conducive to digital innovation: Uganda's Vision 2040, NDP III (2020–2025), the AU Digital Transformation Strategy for Africa (2020–2030), and United Nations SDGs 4 and 9.

1.2 Problem Statement

The management of internship programmes at Busitema University, Nagongera Campus, is characterised by a pervasive reliance on manual, paper-based processes generating systemic inefficiencies. Physical logbooks are prone to damage, loss, and fraudulent completion, compromising assessment integrity. University supervisors assigned fifteen to twenty students across multiple districts cannot conduct meaningful physical supervision within budget constraints, leaving students without timely feedback. Real-time monitoring is entirely absent, preventing early intervention. Students have no structured or confidential channel to report challenges at placement sites. Ansari (2025) demonstrates that inadequate challenge-reporting mechanisms allow preventable welfare issues to escalate undetected, sometimes resulting in avoidable student withdrawal. These compounding failures highlight the urgent need for a WIMS that digitises log submission, enables remote supervision, provides real-time monitoring, and offers a structured challenge-reporting channel.

1.3 Main Objective

To develop a web-based Internship Management System for Busitema University, Nagongera Campus, enabling students to submit geo-tagged daily logs, allowing field and

university supervisors to provide structured remote feedback and monitor cohort-wide progress in real time, and equipping administrators with placement management and institutional reporting capabilities.

1.4 Specific Objectives

1. To review the literature and determine requirements for developing a web-based Internship Management System.
2. To design a web-based Internship Management System based on the determined requirements.
3. To implement a web-based Internship Management System using React, TypeScript, Node.js, and SQLite.
4. To test and validate the web-based Internship Management System to ensure functionality and usability.

1.5 Research Questions

Based on the specific objectives in section 1.4 above, these are the corresponding research questions:

1. What are the requirements for developing a web-based Internship Management System for Busitema University, Nagongera Campus?
2. How can the web-based Internship Management System be designed to meet the needs of students, field supervisors, university supervisors, and administrators?
3. How can the web-based Internship Management System be implemented using React, TypeScript, Node.js, and SQLite?
4. How functional and usable is the proposed web-based Internship Management System?

1.6 Significance of the Study

The successful completion of this study provided: digitised, auditable geo-tagged log submission eliminating document loss and forgery; remote supervisor feedback enabling meaningful supervision regardless of geographic distance (Sides & Mrvica, 2017; Maertz Jr et al., 2014); real-time cohort monitoring enabling early identification of underperforming placements; structured challenge reporting with automated escalation addressing the welfare gap identified by Ansari (2025); and institutional analytics providing evidence-based insights for programme improvement. Possible beneficiaries include higher education institutions across Uganda and East Africa, all four stakeholder categories, researchers

addressing the gap identified by Gosal et al. (2024), and policy makers at the Ministry of Education and Sports.

1.7 Scope of the Study

1.7.1 Content Scope

The WIMS provides a complete multi-stakeholder digital platform for managing internship programmes at Busitema University. Students submit geo-tagged daily logs. Field supervisors review logs and provide structured feedback with numerical ratings. University supervisors access cohort-wide dashboards and manage challenge escalations. Administrators manage the full placement lifecycle and generate downloadable reports. The system enforces role-based access control, JWT authentication, and HTTPS encryption compliant with Uganda's Data Protection and Privacy Act (2019).

1.7.2 Geographical Scope

The study was conducted at Busitema University, Nagongera Campus, Tororo District, Eastern Uganda. System testing incorporated representative internship sites across Uganda's Eastern, Central, and Northern regions.

1.7.3 Time Scope

The study was conducted over three (3) months: 1st October 2025 to 31st December 2025.

CHAPTER TWO: LITERATURE REVIEW

2.0 Introduction

This chapter examines previous studies and theoretical foundations relevant to the development of a web-based Internship Management System for Busitema University, Nagongera Campus. The review is organised according to the specific objectives of the study.

2.1 Literature Related to Objective 1: Requirements Identification

2.1.1 Internship Programmes and Institutional Supervision

Sides and Mrvica (2017) argue that the educational value of an internship is entirely contingent on the quality of institutional scaffolding — orientation, structured supervision, reflection, and assessment. Di Pietro (2022) demonstrates that structured placements develop global competencies and that institutional documentation is critical for graduate employability. Maertz Jr et al. (2014) provide empirical evidence that supervision quality has the strongest marginal effect on intern learning outcomes and that absent structured feedback increases disengagement and voluntary withdrawal. Ansari (2025) highlights that inadequate challenge-reporting mechanisms allow preventable welfare issues to escalate undetected.

2.1.2 Digital Logbook and Internship Management Platforms

Ololade (2021) documents the SIWES Electronic Log Book System addressing data forgery, data loss, and supervisor overload but lacking GPS verification, challenge reporting, and analytics. Gosal et al. (2024) describe the FTI UNTAR web-based platform integrating log submission, supervisor grading, and report generation achieving near-real-time compilation. Cunningham (2017) demonstrates consistent positive effects of digital feedback platforms on student engagement and equity by enabling asynchronous, geographically unconstrained supervision.

2.1.3 Requirements for Developing a Web-Based Internship Management System

Functional Requirements

1. **Role-Based User Authentication:** Separate authenticated dashboards for all four stakeholder roles with role-specific API access controls (Miller, 2020).
2. **Geo-Tagged Daily Log Submission:** Structured daily log entries with automatic GPS coordinate capture at submission (Gosal et al., 2024).

3. Structured Supervisor Feedback: Written feedback with numerical rating and student notification upon review (Maertz Jr et al., 2014).
4. Challenge Reporting and Escalation: Confidential challenge reports with severity classification and automated escalation (Ansari, 2025).
5. Cohort Analytics Dashboard: Real-time aggregate performance metrics across all assigned students (Cunningham, 2017).
6. Placement Management and Report Generation: Full placement lifecycle management and downloadable PDF/Excel reports.

Non-Functional Requirements

1. Usability and Accessibility: Browser-accessible on desktop and mobile; mean SUS score ≥ 70 (Miller, 2020).
2. Security: HTTPS TLS 1.3 for data in transit; bcrypt password hashing; JWT stateless sessions (Miller, 2020).
3. Compliance: Uganda's Data Protection and Privacy Act (2019).
4. Performance: API endpoints respond within 3 seconds under normal load.
5. Scalability: Architecture supports extension to additional campuses with minimal reengineering (Gosal et al., 2024).

2.2 Literature Related to Objective 2: System Design

2.2.1 Web-Based Platforms for Institutional Programme Management

Web-based systems use client-server architecture enabling browser-based access without local installation, providing lower deployment costs, simplified updates, and device agnostic accessibility — essential for serving geographically dispersed placement sites across Uganda.

2.2.2 Integrated Multi-Stakeholder Digital Platforms

Gosal et al. (2024) demonstrate that integrating log management, grading, and reporting into a single system delivers measurable institutional value. Miller's (2020) five-domain risk framework shaped the authentication and encryption architecture. Verzuh (2015) advocates for iterative Agile methodologies prioritising stakeholder engagement and working software.

2.3 Literature Related to Objective 3: System Implementation

React 18 with TypeScript provides component-based UI with static typing reducing runtime errors across complex multi-role dashboards. Node.js with Express.js provides event-driven,

non-blocking I/O for high-concurrency multi-role API sessions. SQLite is a self-contained, serverless, zero-configuration embedded SQL database engine storing the entire database as a single file on disk. Its ACID compliance, lightweight footprint, and zero-administration requirement make it well-suited for deployment in constrained hosting environments. SQLite supports all SQL features required by the WIMS including complex joins, foreign key constraints (enforced via PRAGMA foreign_keys = ON), and transaction support. JWT with bcrypt provides secure scalable authentication (Miller, 2020). Leaflet.js with OpenStreetMap enables geo-tagged log visualisation at no API cost.

2.4 Literature Related to Objective 4: System Testing and Evaluation

The System Usability Scale (SUS) provides a standardised usability measure with a validated 70-point acceptance threshold widely used in educational technology evaluation. Testing approaches include unit, integration, and functional testing with a minimum 80% code coverage target. A traceability matrix mapping each objective to test cases and UAT results ensures transparent accountability (Verzuh, 2015).

2.5 Research Gap

1. Absence of East African Internship Management Systems: No documented, evaluated system exists for a Ugandan or East African public university context (Gosal et al., 2024).
2. Lack of Integrated Challenge-Reporting Workflows: Neither Ololade (2021) nor Gosal et al. (2024) integrate confidential challenge-reporting and automated escalation (Ansari, 2025).
3. Absence of Real-Time Cohort Analytics: Existing systems lack aggregate analytics dashboards for institutional decision-makers (Cunningham, 2017).
4. Limited GPS Verification: Ololade's (2021) SIWES system lacks GPS geo-tagging for physical presence verification.
5. Fragmented Multi-Role Tooling: No single integrated platform serving all four stakeholder roles has been documented for the East African university context.

2.6 Advantages of the Web-Based Internship Management System

1. Elimination of documentation fraud. Geo-tagged, timestamped, JWT-authenticated submissions cannot be fabricated or altered after submission.
2. Remote supervision at scale. Supervisors review logs and provide feedback from any location, removing budget and logistical barriers.

3. Early intervention capability. Real-time dashboards flag students with low submission rates or unresolved challenges for targeted intervention.
4. Structured student welfare protection. Confidential challenge-reporting with automated escalation ensures institutional responses within a defined timeframe.
5. Institutional data intelligence. Analytics engine generates evidence-based KPIs entirely absent from the current manual system.

2.7 Features of the Web-Based Internship Management System

1. Geo-tagged daily log submission. Structured daily entries with automatic GPS capture accessible on desktop and mobile browsers.
2. Field supervisor review and feedback dashboard. Pending log queue with written feedback and numerical ratings (1–5); email notifications on review.
3. University supervisor cohort monitoring dashboard. Aggregate metrics with drill-down into individual student profiles.
4. Challenge reporting and automated escalation. Confidential reports with severity classification; automatic escalation after configurable threshold.
5. Administrator placement management and report generation. Full placement lifecycle management and downloadable PDF/Excel institutional reports.

CHAPTER THREE: METHODOLOGY

3.0 Introduction

This chapter outlines the methods and procedures used to conduct the research, encompassing the research design, area of study, population and sampling, data collection and analysis, system design, implementation, testing and validation, and ethical considerations.

3.1 Research Design

The study adopted both quantitative and qualitative approaches within a Rapid Application Development (RAD) framework from SSADM. Qualitative semi-structured interviews gathered detailed insights into user needs and existing challenges from all four stakeholder categories. Quantitative methods validated the system through SUS scores, performance metrics, and functional requirement pass rates. Agile sprint cycles of one week each ensured continuous stakeholder engagement throughout the three-month development period.

3.2 Area of Study

The research was conducted at Busitema University, Nagongera Campus, Tororo District, Eastern Uganda, which coordinates internship programmes across the Departments of Computer Studies, Education Sciences, Education Arts, and Agriculture.

3.3 Population and Sampling Method

The target population comprised all stakeholders directly involved in internship programme management at Busitema University, Nagongera Campus — approximately 410 individuals. Purposive sampling was used to select field supervisors, university supervisors, and the administrator on the basis of their direct experience of internship management processes. Simple random sampling was applied to select students from the Department of Computer Studies student registry ensuring each student an equal chance of selection.

Using Slovin's formula: $n = N / (1 + N(e)^2)$, where n = sample size, $N = 410$ (total population) and $e = 0.14$ (margin of error). Therefore, $n = 410 / (1 + 410 \times 0.14^2) = 410 / 9.036 \approx 49$ participants.

Table 3-1. Population Sampling Summary

Sampling Method	Category	Quantity	Purpose
Purposive sampling	Field Supervisors	10	Gather supervision workflow requirements

Purposive sampling	University Supervisors	8	Capture monitoring and escalation workflow requirements
Purposive sampling	Administrator	1	Gather placement management and reporting requirements
Simple random sampling	Students	30	Unbiased representation for UAT and validation
Total		49	

3.4 Data Collection Methods and Analysis

Semi-structured interviews were conducted with field supervisors, university supervisors, and the administrator to explore challenges in the current system and desired features. Fifteen interview questions were developed to ensure comprehensive requirement gathering within 30–45-minute sessions. Structured questionnaires were administered to all 49 participants. Qualitative data was analysed thematically; quantitative data was analysed using descriptive statistics for SUS scores, task completion rates, and performance metrics.

3.5 System Development

The system was developed using Rapid Application Development (RAD) within SSADM, selected for the constrained three-month timeline and the need for frequent stakeholder demonstrations. The RAD approach follows four phases: requirements determination, design, implementation, and testing, with weekly sprint cycles and stakeholder feedback integrated at each sprint boundary (Verzuh, 2015).

3.6 Design

The system adopts a three-tier client-server model: a React frontend tier deployed on Vercel; a Node.js/Express.js backend REST API tier deployed on Railway; and an SQLite embedded database stored as a single .db file on the backend server. This separation ensures independent frontend scalability and continuous deployment workflows. The Context Diagram illustrates the system's interactions with six external entities: Student, Field Supervisor, University Supervisor, Administrator, GPS/Map Service, and Email Notifier. The Data Flow Diagram (DFD) decomposes the system into nine processes: user authentication, log submission and geotagging, supervisor log review and feedback, challenge reporting, challenge escalation, analytics and KPI generation, report export, placement management, and system configuration. The Use-Case Diagram features four primary actors, namely Student, Field Supervisor, University Supervisor, and Administrator, plus GPS/Map Service and Email Notifier as external service actors.

3.7 Implementation

Visual Studio Code served as the primary code editor. The backend was built using Node.js with Express.js managing REST API endpoints, with SQLite as the embedded database (PRAGMA foreign_keys = ON enforcing referential integrity across six core entities), and JWT with bcrypt (cost factor 12) securing all sessions. The frontend was built using React 18 with TypeScript and Redux Toolkit for state management, styled with Tailwind CSS for responsive cross-device presentation, with Leaflet.js and OpenStreetMap for GPS mapping at zero API cost.

Table 3-2. Technology Stack and Justification

Layer	Technology	Justification
Frontend	React 18 + TypeScript	Component-based UI with static typing across complex multi-role dashboards
Styling	Tailwind CSS	Utility-first responsive framework for consistent cross-device interfaces
State Management	Redux Toolkit	Predictable global state across four distinct role dashboards
Backend	Node.js + Express.js	Event-driven non-blocking I/O for high concurrency multi-role API sessions
Database	SQLite	Self-contained, serverless, zero configuration embedded SQL; ACID compliant; single .db file deployment — no separate database server required
Authentication	JWT + bcrypt	Stateless token sessions with bcrypt hashing for secure scalable auth
Maps & GPS	Leaflet.js + OpenStreetMap	Open-source mapping; geo-tagged log visualisation at no API cost
Frontend Hosting	Vercel	Zero-configuration deployment with automatic HTTPS and global CDN
Backend Hosting	Railway	Managed Node.js with auto-scaling; SQLite .db file persisted on mounted volume
Testing	Jest + Supertest + RTL	Full-stack unit, integration, and component testing

3.8 Testing and Validation

1. Unit Testing: Individual components tested in isolation using Jest and React Testing Library; minimum 80% code coverage target.
2. Functional Testing: System evaluated against all 10 functional requirements; test cases derived directly from requirements.

3. Integration Testing: Frontend–API–SQLite database communication verified using Supertest to validate payloads, status codes, and database state.
4. User Acceptance Testing (UAT): Subset of 49 sample participants completed task scenarios; post-task SUS questionnaires targeting mean score ≥ 70 .
5. System Validation: Traceability matrix maps each specific objective to system features, test cases, and UAT results evidencing achievement.

3.9 Ethical Considerations

1. Informed Consent. Written consent obtained before any data collection commenced.
2. Confidentiality. Personally identifiable information stored separately; system complies with Uganda’s Data Protection and Privacy Act (2019).
3. Voluntary Participation. Participants free to withdraw at any time without consequences.
4. Transparency. Findings and system outcomes shared with participants, supervisors, and the Department.
5. Risk Minimisation. System rigorously tested to prevent errors adversely affecting student academic records.
6. Respect for Autonomy. Participants’ rights to make informed decisions were respected throughout the study.

CHAPTER FOUR: FIELD STUDY AND SYSTEM DESIGN

4.0 Introduction

This chapter presents the findings from the field study conducted at Busitema University, Nagongera Campus, followed by the complete system design artefacts. Section 4.1 describes the current system. Section 4.2 presents field study findings. Section 4.3 defines user requirements. Section 4.4 specifies hardware and software requirements. Section 4.5 presents the full system design with all diagrams.

4.1 Background Information of the Current System

Busitema University, Nagongera Campus, Tororo District, Eastern Uganda, deploys approximately 80–120 Computer Science students annually to organisations across Uganda for eight to twelve weeks. The current system is entirely manual: students receive physical logbooks at placement commencement, record daily activities, obtain periodic supervisor signatures, and return completed logbooks to the Department at placement end where staff manually compile and archive records. Weaknesses include logbook loss and forgery, absent real-time monitoring, no challenge-reporting channel, no analytics, and heavy manual administrative overhead.

4.2 System Study and Analysis

4.2.1 Findings from Interviews and Questionnaires

The field study involved 49 participants drawn from four stakeholder groups. All ten field supervisors indicated that they signed logbooks only during physical site visits, typically once a month. Every university supervisor was responsible for 15–20 students spread across several districts, though only a few managed more than one field visit during the previous academic year because of limited funding.

Among the thirty student respondents, most reported challenges with logbook management, including loss, damage, and missed entries when supervisors were unavailable. One student explained, “My supervisor was only available on Fridays, so I filled the whole week on Thursday.” A small number admitted to fabricating entries at least once, citing the absence of any verification mechanism.

Technology adoption was strong: nearly all students owned or regularly used smartphones with internet access, and the average ease-of-use rating for a web-based system was 4.1 out of 5. University supervisors consistently emphasised the need for real-time progress monitoring. The administrator estimated that compiling manual reports required three to five

working days per cohort, while field supervisors described automated challenge escalation as extremely helpful.

4.3 User Requirements

4.3.1 Functional Requirements

Table 4-1. Functional Requirements

Req. ID	Requirement	Priority
FR01	Role-based registration and authentication for all four stakeholder categories.	High
FR02	Students submit daily logbook entries with activity description, hours worked, and geo-tagged GPS coordinates.	High
FR03	Field supervisors review log entries and provide written feedback with numerical rating (1–5).	High
FR04	Students submit structured challenge reports with severity classification (low, medium, high, critical).	High
FR05	System automatically escalates unresolved challenge reports beyond configurable threshold to university supervisors.	High
FR06	University supervisors access cohort-wide dashboards with aggregate progress, feedback frequency, and challenge resolution metrics.	High
FR07	Administrators manage placements, assign supervisors, and generate downloadable reports (PDF and Excel).	High
FR08	System sends email notifications for log submissions, feedback delivery, and challenge escalations.	Medium
FR09	System displays geo-tagged log submissions on an interactive map.	Medium
FR10	Administrators maintain a full audit log of all system configuration changes.	Medium

4.3.2 Non-Functional Requirements

Table 4-2. Non-Functional Requirements

Req. ID	Requirement	Category
NFR01	Accessible via standard web browsers on desktop and mobile without installation.	Usability
NFR02	All API endpoints respond within 3 seconds under normal load.	Performance

NFR03	HTTPS with TLS 1.3 for all data in transit.	Security
NFR04	Passwords hashed using bcrypt with minimum cost factor 12.	Security
NFR05	Compliant with Uganda's Data Protection and Privacy Act (2019).	Compliance
NFR06	Architecture supports extension to additional campuses with minimal re-engineering.	Scalability
NFR07	Mean System Usability Scale score ≥ 70 in UAT.	Usability
NFR08	99.5% uptime during the academic internship period.	Reliability

4.4 Hardware / Software Requirements

4.4.1 Hardware Requirements

Table 4-3. Hardware Requirements

Component	Minimum Specification	Recommended
Processor	Intel Core i3 / ARM equivalent	Intel Core i5 or higher
RAM	4 GB	8 GB or higher
Storage	20 GB available space	50 GB SSD
Network	3G mobile data (1 Mbps)	Broadband / 4G (10 Mbps+)
Display	1280×720 resolution	1920×1080 resolution
Mobile Device	Android 8.0 / iOS 13 or higher	Android 11 / iOS 15 or higher

4.4.2 Software Requirements

Table 4-4. Software Requirements

Component	Technology / Version
Operating System	Windows 10+, macOS 11+, Ubuntu 20.04+, Android 8+, iOS 13+
Web Browser	Chrome 90+, Firefox 88+, Edge 90+, Safari 14+
Node.js Runtime	Node.js 18.x LTS or higher
Database	SQLite 3 (embedded, no separate server required)
Frontend Framework	React 18 with TypeScript
Backend Framework	Express.js 4.x on Node.js
Code Editor	Visual Studio Code
Version Control	Git + GitHub
Frontend Hosting	Vercel (automatic HTTPS + CDN)
Backend Hosting	Railway (managed Node.js; SQLite file on mounted volume)

4.5 System Design

4.5.1 The System Architecture

The WIMS adopts a three-tier client-server architecture. The React-based frontend tier is deployed on Vercel and communicates exclusively via authenticated HTTPS REST API calls to the Node.js/Express.js backend tier deployed on Railway. The backend tier hosts the SQLite database as an embedded .db file, eliminating the need for a separate database server process. Role-based middleware at the API layer ensures each endpoint is accessible only to users with the appropriate role. The frontend is served from a global CDN ensuring fast load times for students at placement sites across Uganda.

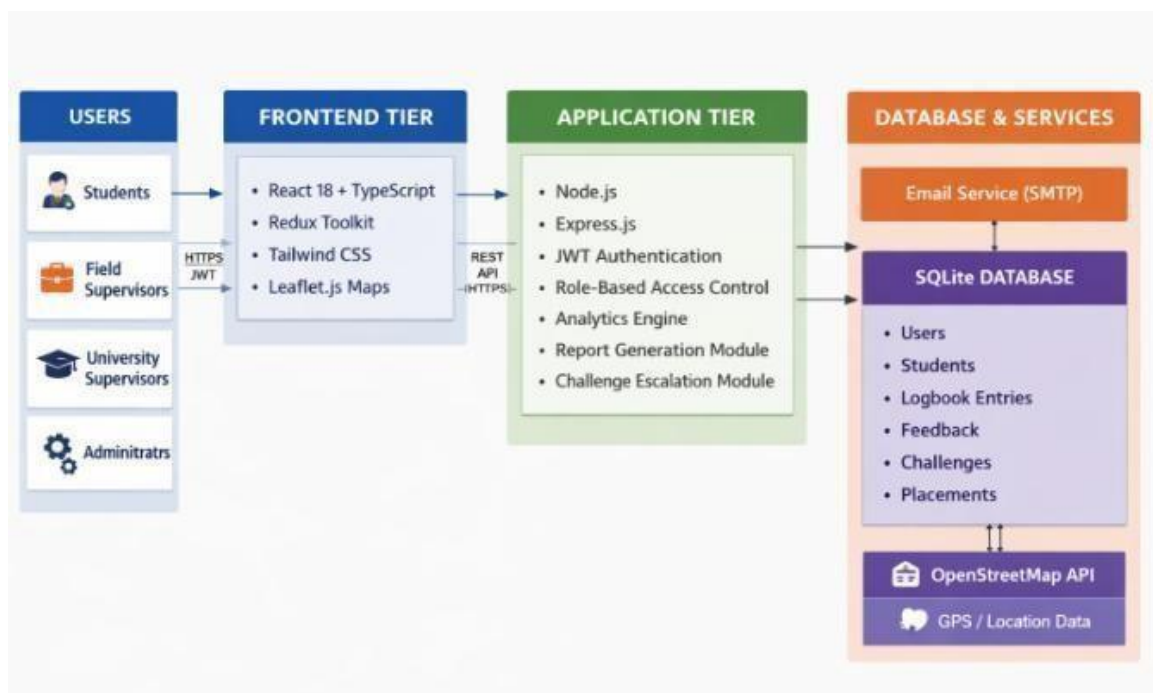


Figure 4-1. System Architecture Diagram

4.5.2 Context Diagram (Level 0 DFD)

The context diagram illustrates the system boundary and the interactions between the WIMS and its six external entities: Student, Field Supervisor, University Supervisor, Administrator, GPS/Map Service (OpenStreetMap via Leaflet.js), and Email Notifier (SMTP service). Solid arrows represent primary data flows; dashed arrows represent external service interactions.

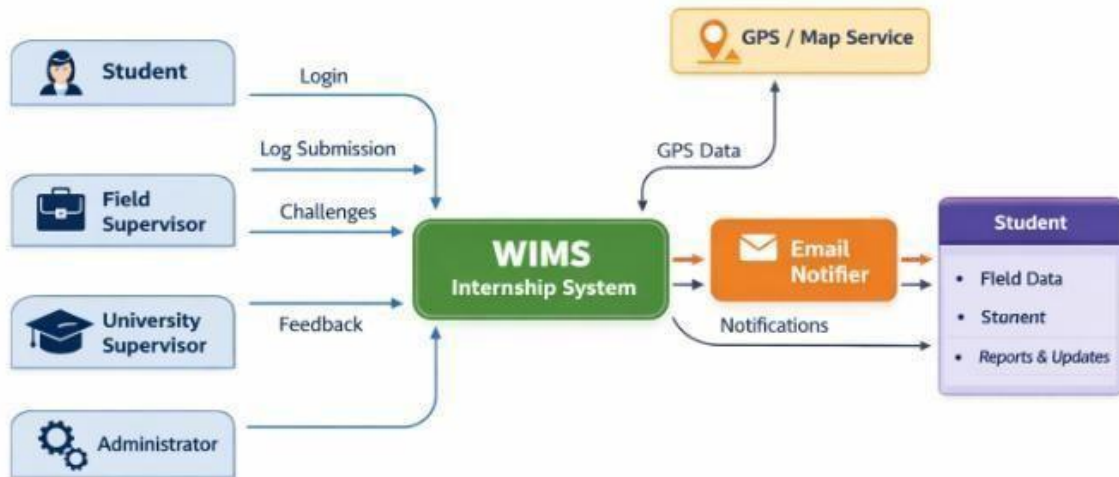


Figure 4-2. Context Diagram (Level 0 DFD)

4.5.3 Data Flow Diagram (Level 1)

The Level 1 DFD decomposes the system into nine discrete processes showing how data flows between external actors, internal processes, and the six SQLite data stores. Each process is represented as a numbered circle; data stores as open rectangles with index labels (D1–D6).



Figure 4-3. Data Flow Diagram (Level 1)

Table 4-5. DFD Process Descriptions

Process	Name	Input	Output
P1	User Authentication	Email + password	JWT token or rejection
P2	Log Submission & Geo-tagging	Activity text, hours, GPS	Stored log (status: pending)
P3	Supervisor Review & Feedback	Log entry + feedback input	Feedback record, updated log status
P4	Challenge Reporting	Challenge description + severity	Challenge ticket (status: open)
P5	Challenge Escalation	Unresolved ticket > threshold	Escalated alert to university supervisor
P6	Analytics & KPI Generation	All logs, feedback, challenges	Dashboard KPIs, trend charts
P7	Report Export	Filter parameters	Downloadable PDF or Excel report
P8	Placement Management	Student + organisation data	Placement record, supervisor assignment
P9	System Configuration	Admin settings	Updated configuration, audit log

4.5.4 UML Diagram (Use-Case Diagram)

The Use-Case Diagram below visually summarises the interactions between the system's four primary actors and two external service actors with the key system functions. Association lines connect actors to their respective use cases; dashed arrows marked <<include>> indicate use cases that invoke external services.

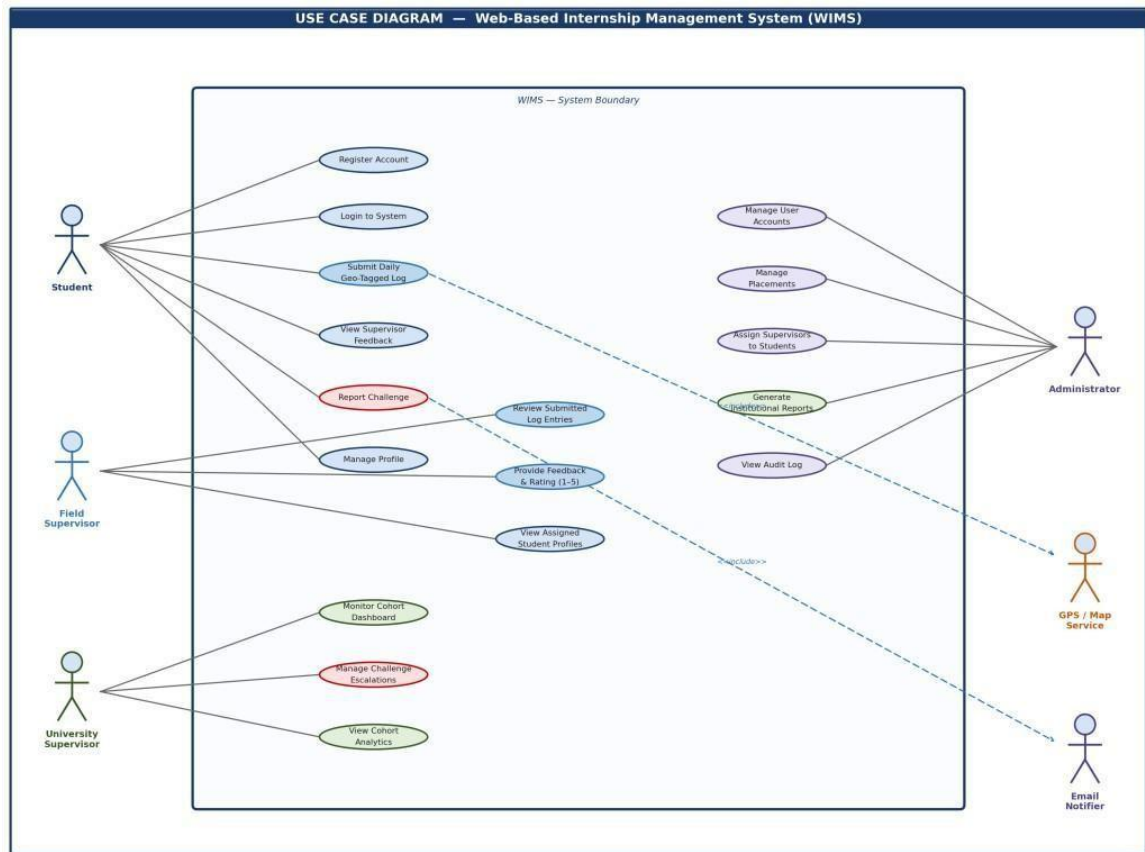


Figure 4-4. Use Case (UML) Diagram

Actors and Their Use Cases

Student: Register Account, Login to System, Submit Geo-Tagged Daily Log, View Supervisor Feedback, Report Challenge, Manage Profile.

Field Supervisor: Review Submitted Log Entries, Provide Feedback and Rating (1–5), View Assigned Student Profiles.

University Supervisor: Monitor Cohort Dashboard, Manage Challenge Escalations, View Cohort Analytics.

Administrator: Manage User Accounts, Manage Placements, Assign Supervisors to Students, Generate Institutional Reports, View Audit Log.

GPS/Map Service: Invoked during log submission to capture and store geo-coordinates; renders map tiles for GPS visualisation views.

Email Notifier: Invoked by the system to deliver notifications to stakeholders for log reviews, feedback, and challenge escalations.

4.5.5 The Entity Relationship Diagram (ERD)

The ERD below defines the six primary entities of the WIMS SQLite database and their relationships. Primary keys are marked [PK] and Foreign keys are marked [FK]. Relationship cardinality is shown on the connecting lines.

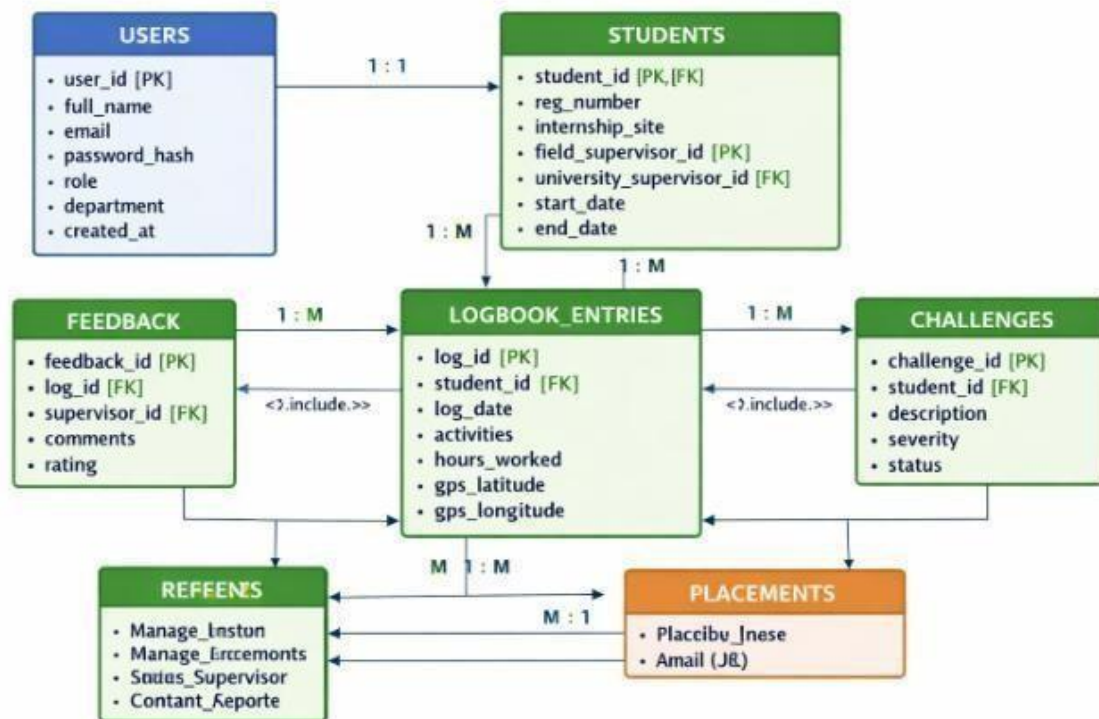


Figure 4-5. Entity Relationship Diagram (ERD)

1. **USERS:** Central identity and authentication record — `user_id` (TEXT PK), `full_name`, `email` (UNIQUE), `password_hash`, `role` (CHECK student/field_supervisor/university_supervisor/admin), `department`, `created_at`.
2. **STUDENTS:** Extends **USERS** with placement attributes — `student_id` (TEXT PK/FK→USERS), `reg_number` (UNIQUE), `internship_site`, `field_supervisor_id` (FK→USERS), `university_supervisor_id` (FK→USERS), `start_date`, `end_date`.
3. **LOGBOOK_ENTRIES:** Records every daily submission — `log_id` (TEXT PK), `student_id` (FK→STUDENTS), `log_date`, `activities`, `hours_worked` (REAL CHECK > 0), `gps_latitude`, `gps_longitude`, `status` (DEFAULT 'pending'), `submitted_at`.
4. **FEEDBACK:** Stores supervisor feedback — `feedback_id` (TEXT PK), `log_id` (FK→LOGBOOK_ENTRIES), `supervisor_id` (FK→USERS), `comments`, `rating` (INTEGER CHECK 1–5), `given_at`.
5. **CHALLENGES:** Records challenge reports — `challenge_id` (TEXT PK), `student_id` (FK→STUDENTS), `description`, `severity` (DEFAULT 'medium'), `status` (DEFAULT

‘open’), reported_at, resolved_at.

6. PLACEMENTS: Stores placement records — placement_id (TEXT PK), student_id (FK→STUDENTS), organisation_name, organisation_address, contact_person, contact_email, admin_id (FK→USERS).

Key Relationships: USERS→STUDENTS (1:1 extends);
STUDENTS→LOGBOOK_ENTRIES (1:M submits logs);
LOGBOOK_ENTRIES→FEEDBACK (1:M receives feedback);
STUDENTS→CHALLENGES (1:M reports); USERS(Supervisors)→STUDENTS (1:M supervises); USERS(Admin)→PLACEMENTS (1:M creates);
STUDENTS→PLACEMENTS (1:1 assigned to).

Note: SQLite requires PRAGMA foreign_keys = ON to be executed at connection time to enforce referential integrity. This is set in the database connection initialisation module of the WIMS backend.

CHAPTER FIVE: IMPLEMENTATION AND TESTING

5.0 Introduction

This chapter presents the implementation of the design outlined in Chapter Four, including real interface screenshots, the SQLite database structure, and system testing results for the Web-Based Internship Management System for Busitema University, Nagongera Campus.

5.1 Interface Design

The functional and non-functional requirements implemented for this system include: for the student — register and login, submit geo-tagged daily logs, track submission history, view received feedback, report challenges. For the field supervisor — view pending log queue, review logs, provide feedback with ratings. For the university supervisor — access cohort analytics dashboard, monitor aggregate metrics, manage challenge escalations. For the administrator — manage placement records, assign supervisors, manage user accounts, view analytics, and generate downloadable reports.

5.1.1 Landing Page

The landing page serves as the public-facing entry point, presenting the system's value proposition, key features, platform analytics, and role-specific stakeholder descriptions.

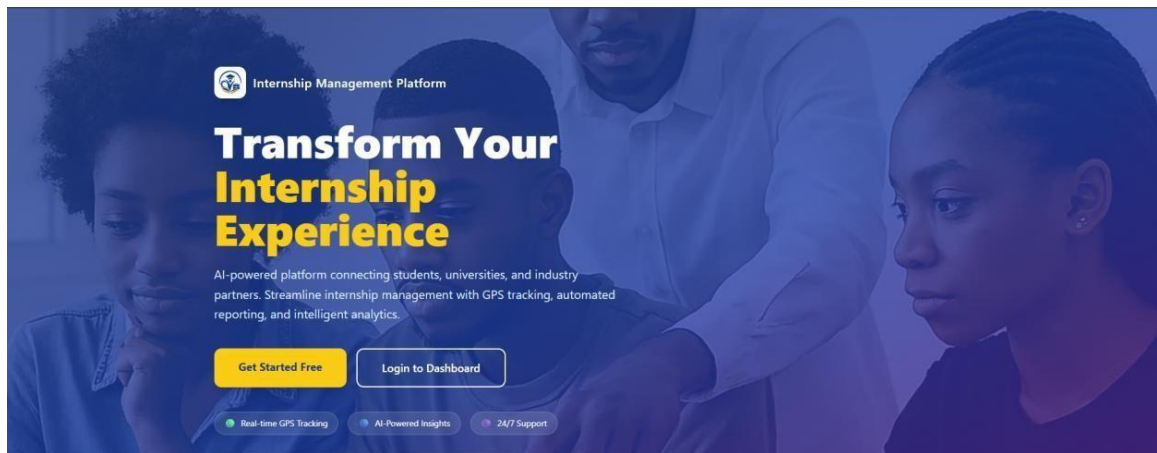


Figure 5-1. Landing Page — Hero Section

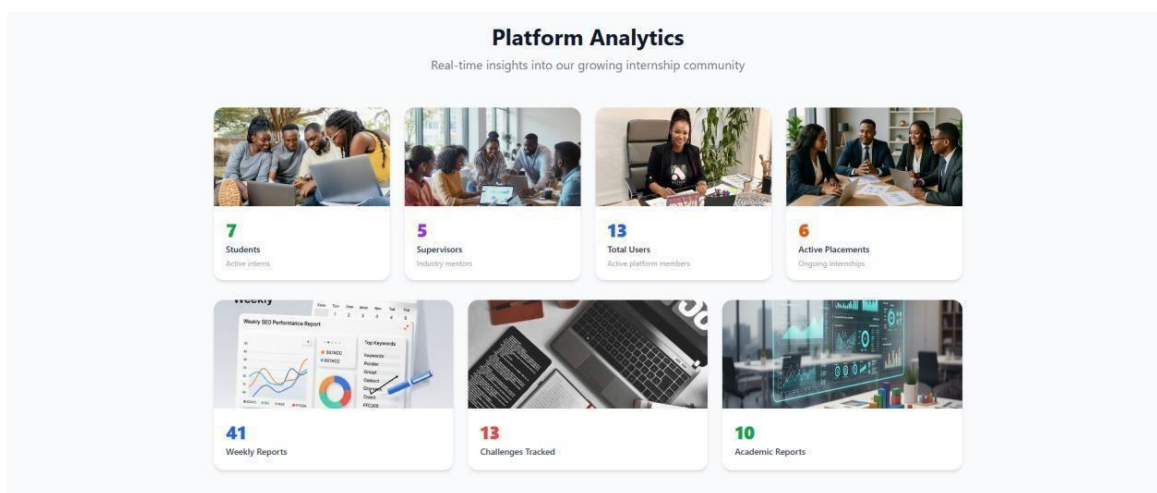


Figure 5-2. Landing Page — Platform Analytics

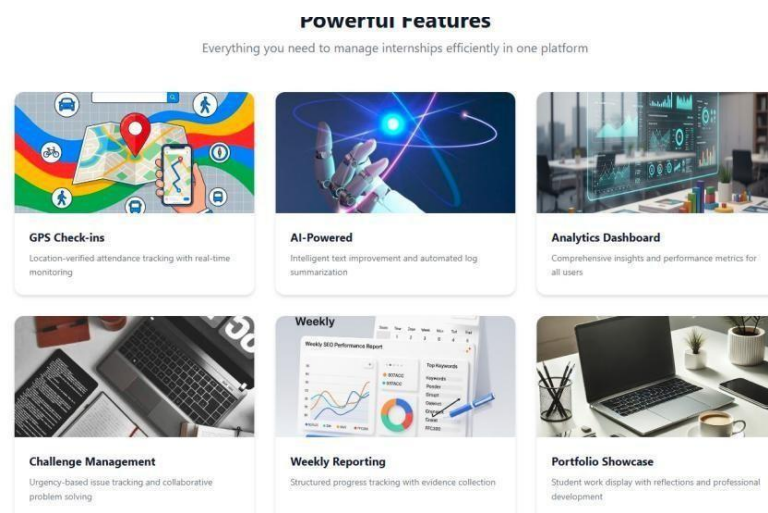


Figure 5-3. Landing Page — Powerful Features

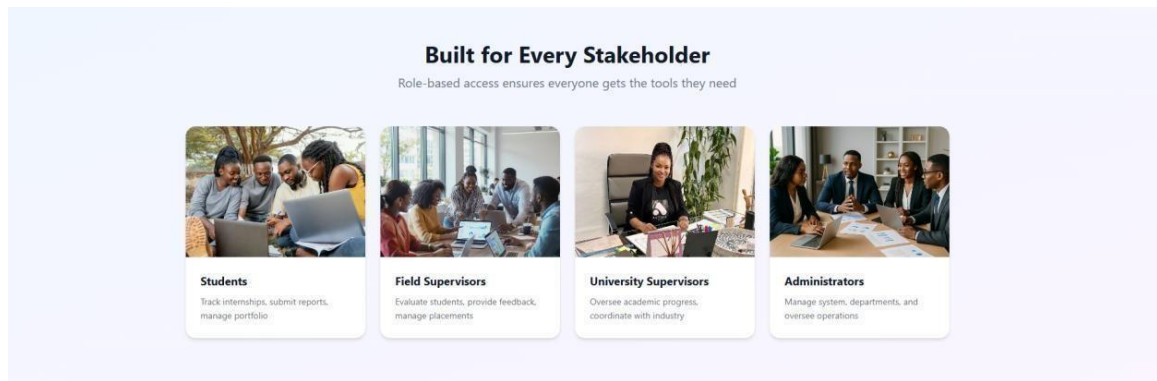


Figure 5-4. Landing Page — Built for Every Stakeholder

5.1.2 Student Registration and Login Interfaces

The registration interface captures full name, email address, student ID, and password. The login interface authenticates students using credentials and establishes a secure JWT session with a role-specific redirect to the student dashboard.

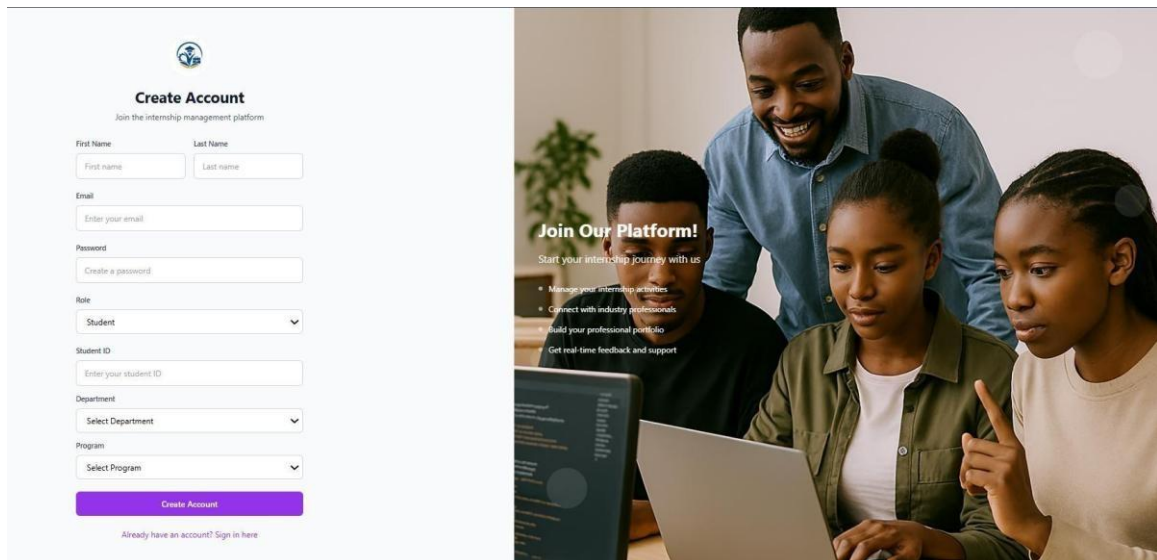


Figure 5-5. Student Registration Page

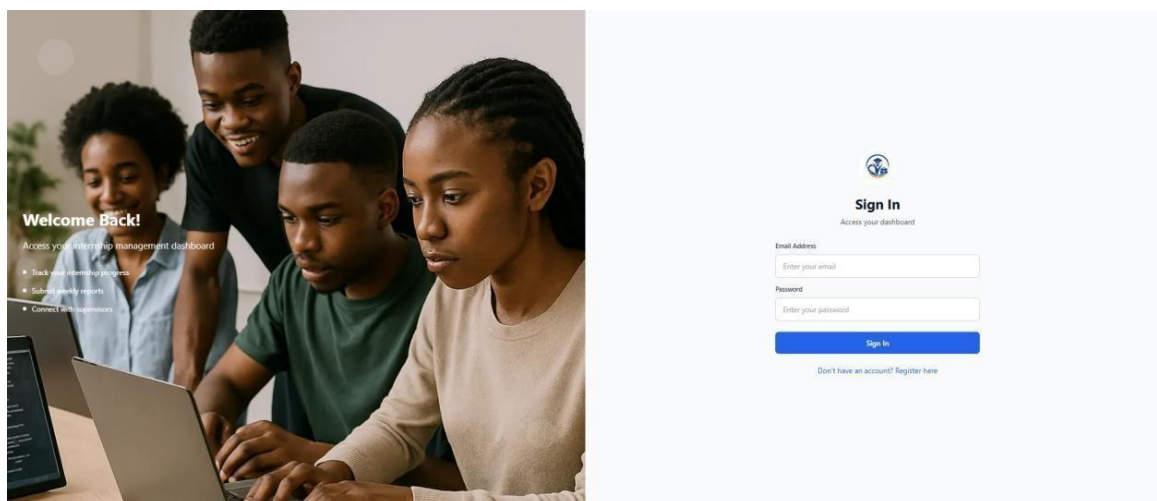


Figure 5-6. Student Login Page

5.1.3 Student Dashboard — Overview

The overview displays the Daily Activity Log submission form, Challenge Report form, Feedback and Progress panel, Hours Worked Per Week chart, and Peer Collaboration section. Students submit geo-tagged logs directly using the GPS-enabled Submit Log button.

Figure 5-7. Student Dashboard — Overview

5.1.4 Student Dashboard — Weekly Log Submission Form

The Weekly Logs tab presents a structured form capturing the week number, activity description, evidence URL, hours worked, and confidence level slider. Previous logs are displayed on the right panel for reference.

Figure 5-8. Student Dashboard — Weekly Log Submission Form

5.1.5 Student Dashboard — Challenge Reporting

The Challenges tab allows students to submit confidential challenge reports with a title, category, urgency level, and description. Submitted challenges appear in the My Challenges panel on the right.

The screenshot shows the 'Student Dashboard' for Akello Aisha Ramathan (ID: 2300402241). The 'Challenges' tab is active in the navigation bar. On the left, the 'Report Challenge' form includes fields for Title, Category (set to 'Technical'), Urgency (set to 'Medium'), and Description, followed by a 'Submit Challenge' button. On the right, the 'My Challenges' panel is currently empty.

Figure 5-9. Student Dashboard — Challenge Reporting

5.1.6 Student Dashboard — Portfolio Management

The Portfolio tab allows students to document work products and reflections capturing title, file type, file URL, and reflection notes, contributing to the professional development record.

Student Dashboard

Akello Aisha Ramathan - 2300402241

Check In

A

OverviewWeekly LogsChallengesPortfolioReportsProfile

Add Portfolio Item

Title

File Type

Document

File URL

Reflection

What did you learn from this work?

Add to Portfolio

My Portfolio

Figure 5-10. Student Dashboard — Portfolio Management

5.1.7 Student Dashboard — Profile Management

The Profile tab enables students to update personal details including name, email, student ID, phone number, and password.

Student Dashboard

Akello Aisha Ramathan - 2300402241

Check In



Overview Weekly Logs Challenges Portfolio Reports **Profile**

Profile Management



First Name: Akello Aisha Last Name: Ramathan

Email: akelloisha@gmail.com

Student ID: 2300402241

Phone:

Change Password: New password

Update Profile

Figure 5-11. Student Dashboard — Profile Management

5.1.8 Field Supervisor Dashboard — Main View

The field supervisor dashboard presents assigned students, pending log review queue, and performance summary with immediate visibility into pending submissions requiring review.

Field Supervisor Dashboard

David Tumusime - RENU UG



Overview **Students** Challenges Templates Analytics **Profile**

Student Logs Review
Click a student from the list to view their weekly logs, approve or comment.

Structured Evaluations
Assess student skills with custom evaluation templates.

Challenge Response
View and escalate high-priority challenges reported by your students.

Quick Templates
Use pre-built templates to speed up evaluations or feedback.

Student Portfolio
View student project portfolios and give feedback.

AI Feedback Suggestions
Get AI-generated recommendations for student improvement.

Feedback History
Review past feedback given to students.

Geo-verified Check-ins
Manage your own field visits with location proof.

Offline Sync Mode
Sync data when network becomes available.

Figure 5-12. Field Supervisor Dashboard — Main View

5.1.9 Field Supervisor Dashboard — Student List

This view lists all assigned students with current submission status, last log date, and feedback completion rate, enabling identification of students with outstanding submissions.

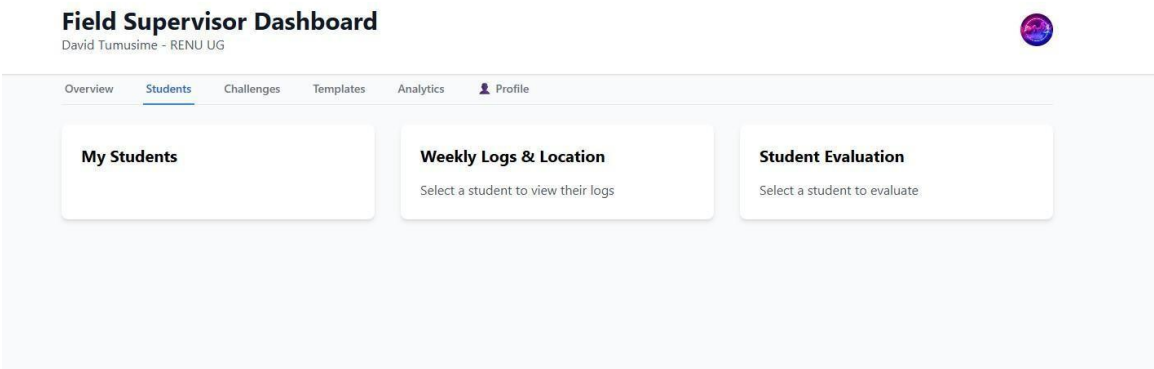


Figure 5-13. Field Supervisor Dashboard — Student List

5.1.10 Field Supervisor Dashboard — Log Review Interface

The log review interface presents individual log entries in full including submission date, activity description, hours worked, and GPS location at submission.

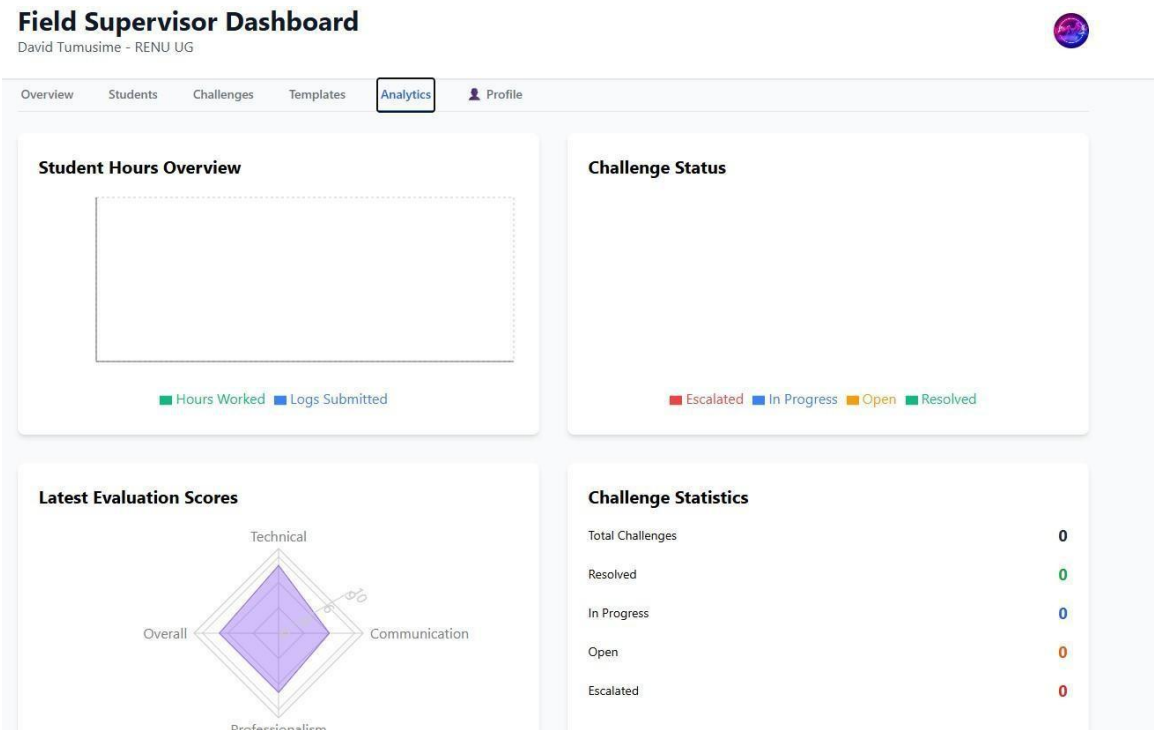


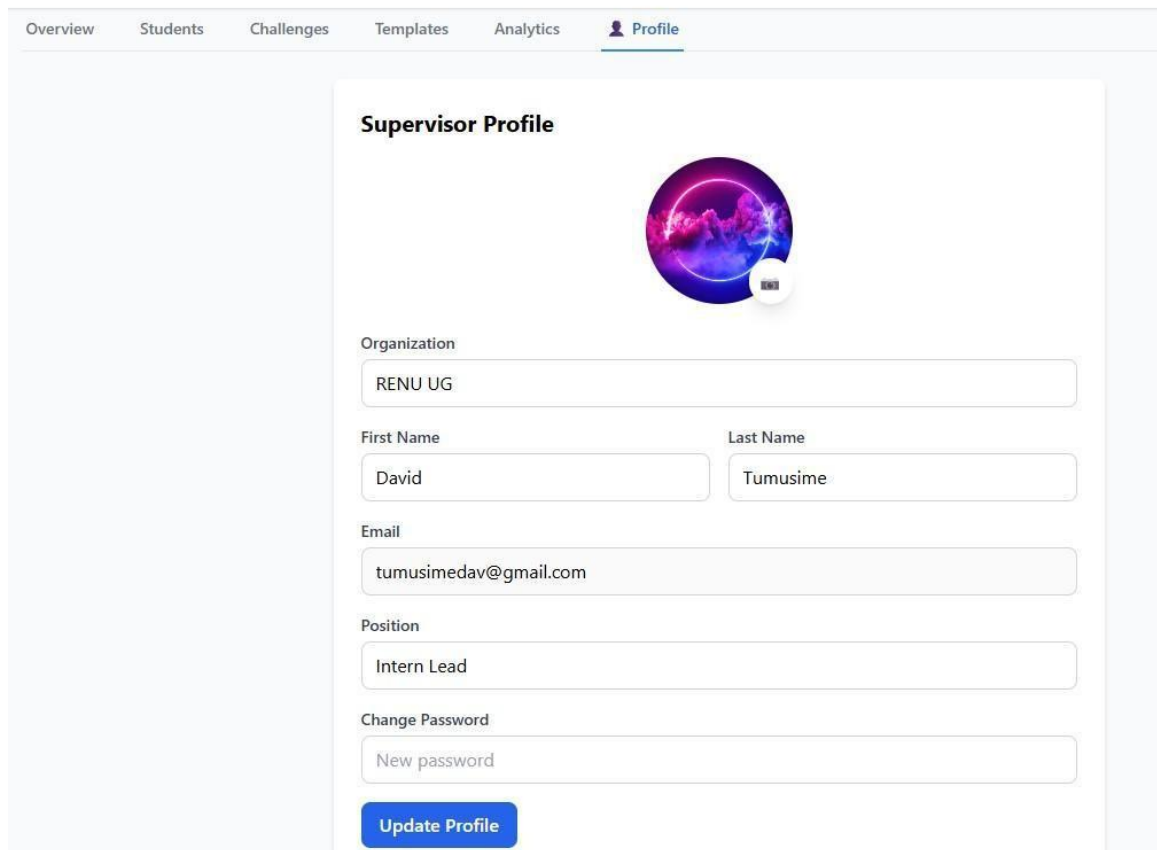
Figure 5-14. Field Supervisor Dashboard — Log Review Interface

5.1.11 Field Supervisor Dashboard — Feedback Submission

After reviewing a log, the supervisor enters written feedback and assigns a numerical rating (1–5). Upon submission the log status transitions to ‘reviewed’ and the student receives an email notification.

Field Supervisor Dashboard

David Tumusime - RENU UG



The screenshot shows the 'Field Supervisor Dashboard' for David Tumusime at RENU UG. The dashboard has a navigation bar with links: Overview, Students, Challenges, Templates, Analytics, and Profile (which is highlighted). The main content area is titled 'Supervisor Profile' and features a circular profile picture of a person with purple hair. Below the profile picture are several form fields for updating the profile: Organization (RENU UG), First Name (David), Last Name (Tumusime), Email (tumusimedav@gmail.com), Position (Intern Lead), and Change Password (New password). An 'Update Profile' button is located at the bottom of the form.

Figure 5-15. Field Supervisor Dashboard — Feedback Submission

5.1.12 University Supervisor Dashboard — Overview

The university supervisor dashboard provides cohort-wide aggregate performance metrics including total log submissions, average feedback ratings, submission frequency trends, and challenge resolution status.

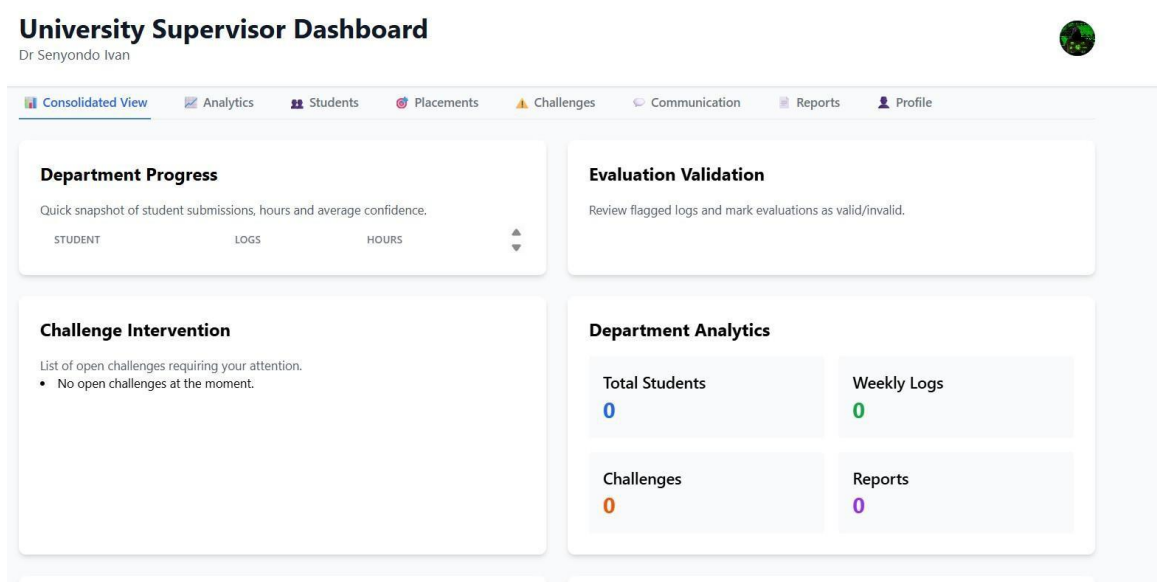


Figure 5-16. University Supervisor Dashboard — Overview

5.1.13 University Supervisor Dashboard — Student Monitoring

Enables monitoring of individual student progress within the cohort, reviewing submission histories, feedback received, and hours worked to date.

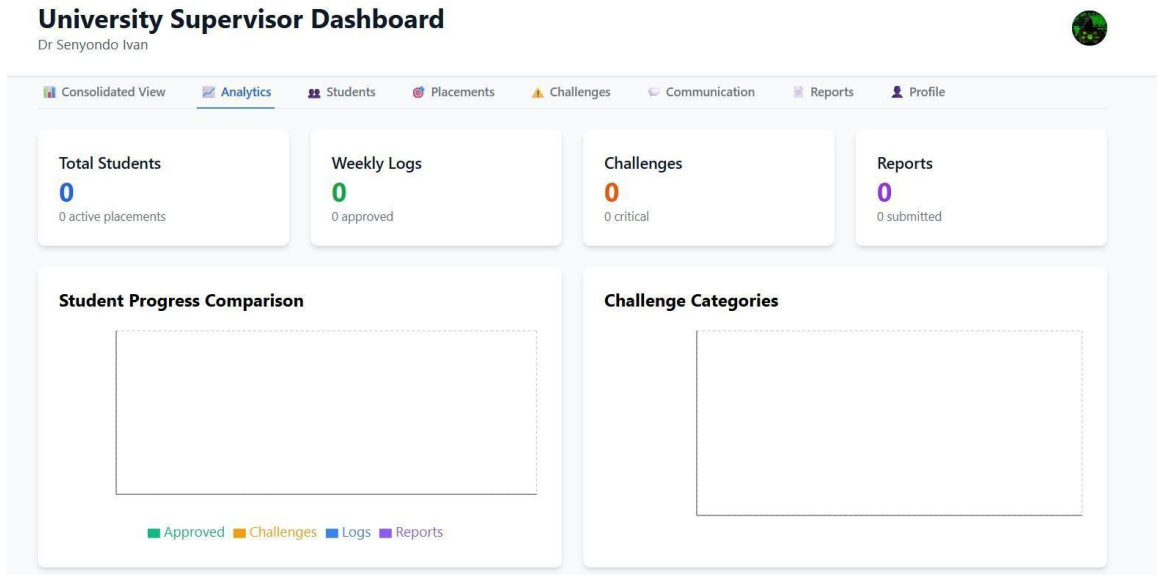


Figure 5-17. University Supervisor Dashboard — Student Monitoring

5.1.14 University Supervisor Dashboard — Challenge Management

Presents all active challenge tickets from assigned students with urgency classification, submission date, and current status. Supervisors update resolution status and close tickets from this interface.

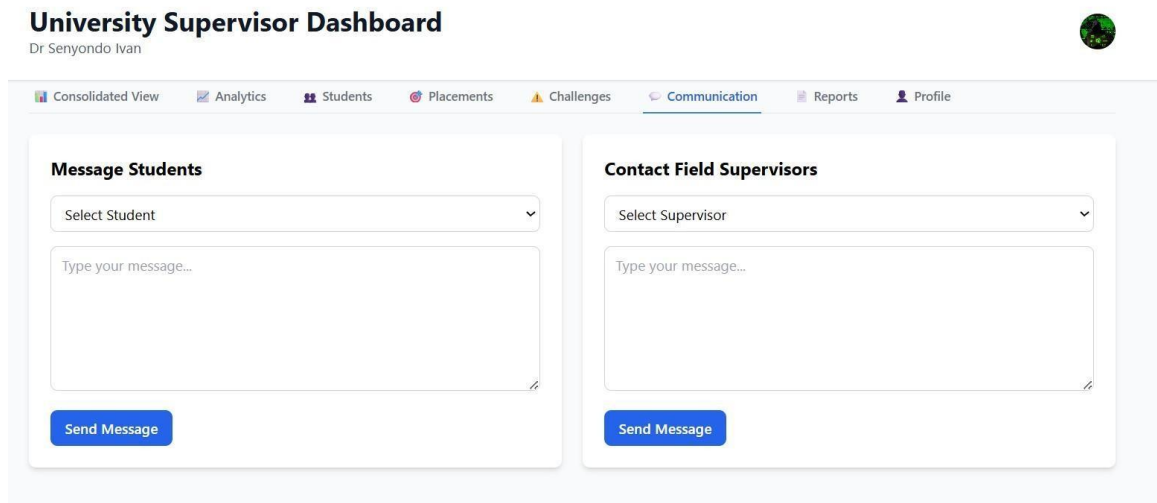


Figure 5-18. University Supervisor Dashboard — Challenge Management

5.1.15 University Supervisor Dashboard — Reports View

Enables university supervisors to generate cohort-level reports filtered by period and student group.

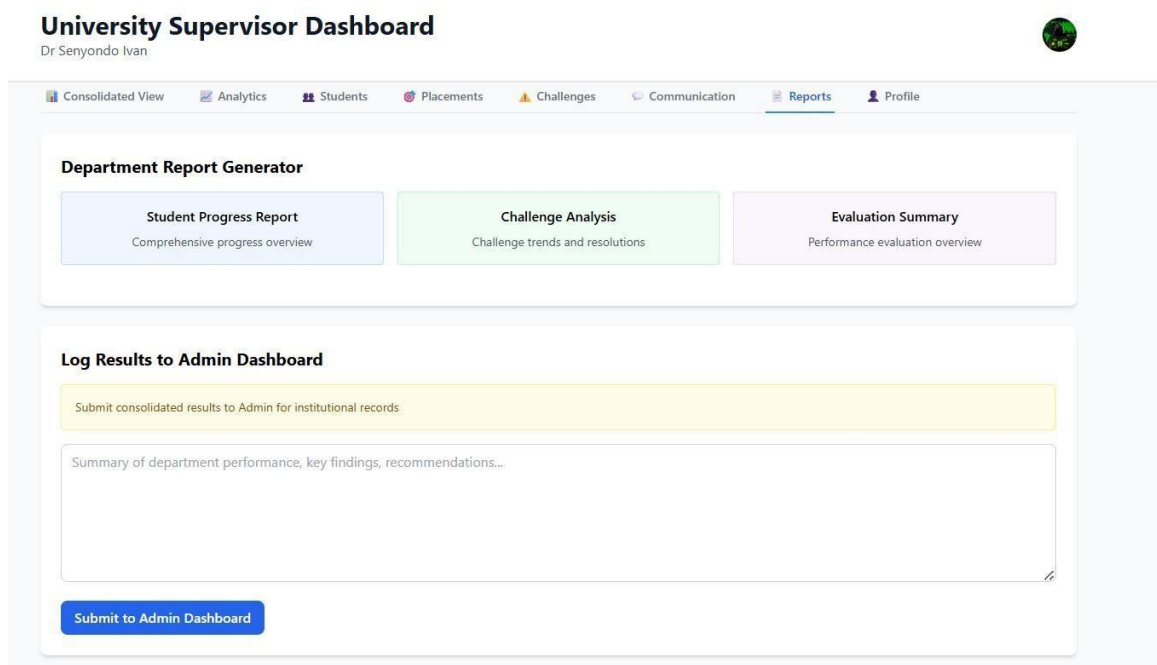


Figure 5-19. University Supervisor Dashboard — Reports View

5.1.16 University Supervisor Dashboard — Analytics

Interactive performance charts covering submission frequency trends, average feedback rating trends, challenge severity distribution, and cohort hour accumulation progress.

University Supervisor Dashboard

Dr Senyondo Ivan

Consolidated View

Analytics

Students

Placements

Challenges

Communication

Reports

Profile

Supervisor Profile



Title

Dr

First Name

Senyondo

Last Name

Ivan

Email

sivan@busitema.ac.ug

Office Location

Change Password

New password

Update Profile

Figure 5-20. University Supervisor Dashboard — Analytics

5.1.17 Administrator Dashboard — Main Overview

Full institutional oversight with system-wide analytics including total active placements, overall submission rates, user account statistics, and challenge counts by severity.

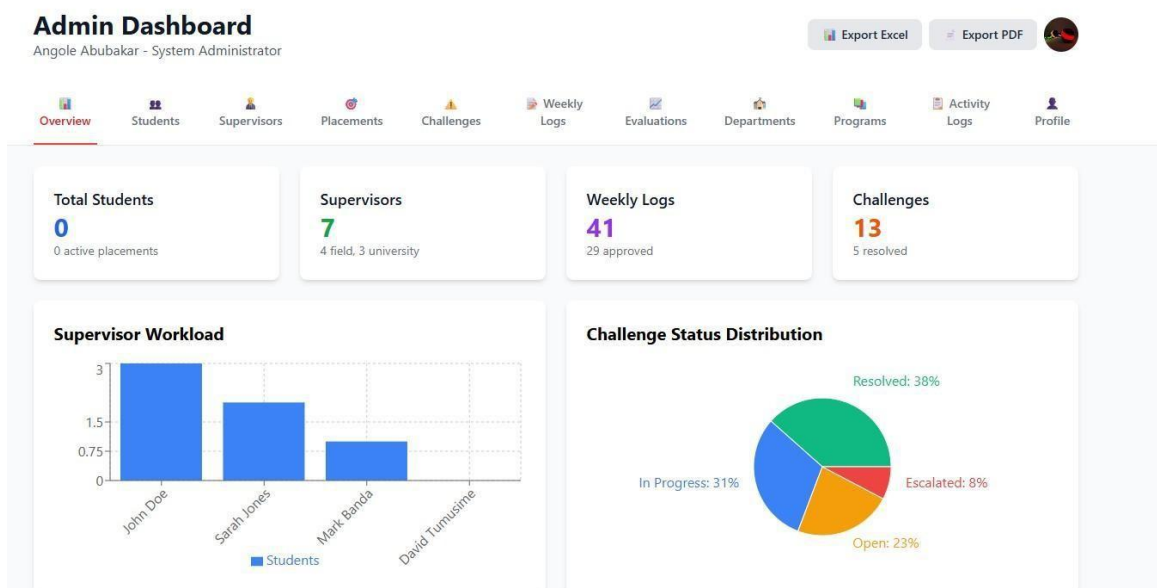


Figure 5-21. Administrator Dashboard — Main Overview

5.1.18 Administrator Dashboard — User Management

Enables the administrator to create, deactivate, and manage user accounts across all four stakeholder roles.

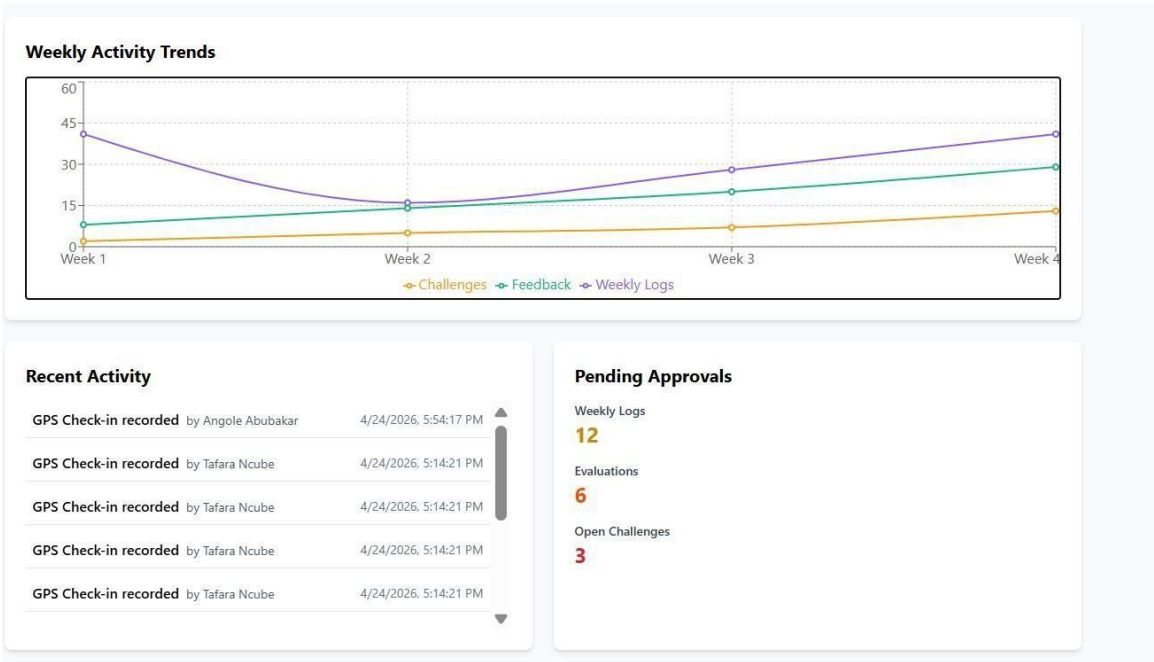


Figure 5-22. Administrator Dashboard — User Management

5.1.19 Administrator Dashboard — Placement Management

Allows administrators to create and manage student placement records, entering organisation details, placement dates, and assigning supervisors.

Admin Dashboard
Angole Abubakar - System Administrator

Export Excel Export PDF

Overview Students Supervisors Placements Challenges Weekly Logs Evaluations Departments Programs Activity Logs Profile

Student Management + Add Student

Add New Student

First Name	Last Name
Email	Password
Student ID	Phone
Department	Program
Select Department	Select Program
Create Student	Cancel

Figure 5-23. Administrator Dashboard — Placement Management

5.1.20 Administrator Dashboard — Analytics Dashboard

Institution-wide KPIs with interactive charts for submission frequency, average feedback ratings, challenge resolution rates, and department-level comparisons.

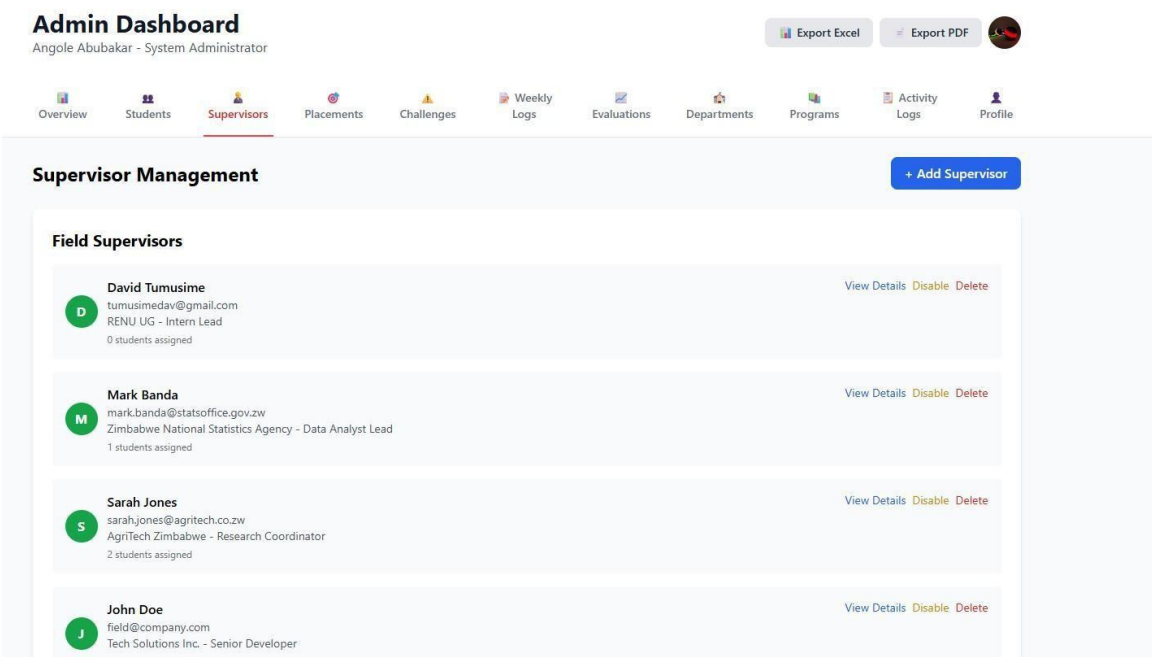


Figure 5-24. Administrator Dashboard — Analytics Dashboard

5.1.21 Administrator Dashboard — Department Overview

Department-level breakdown allowing administrators to compare submission rates, challenge frequencies, and supervisor engagement across departments.

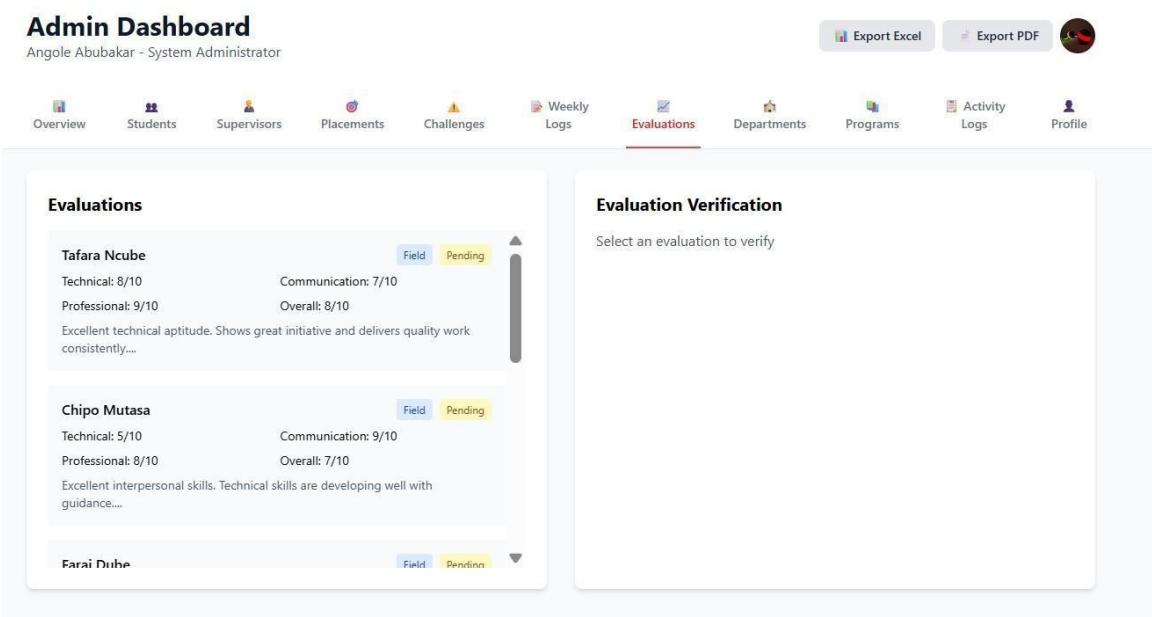


Figure 5-25. Administrator Dashboard — Department Overview

5.1.22 Administrator Dashboard — System Reports

Enables administrators to generate downloadable institutional reports filtered by academic period, department, or supervisor in PDF and Excel formats.

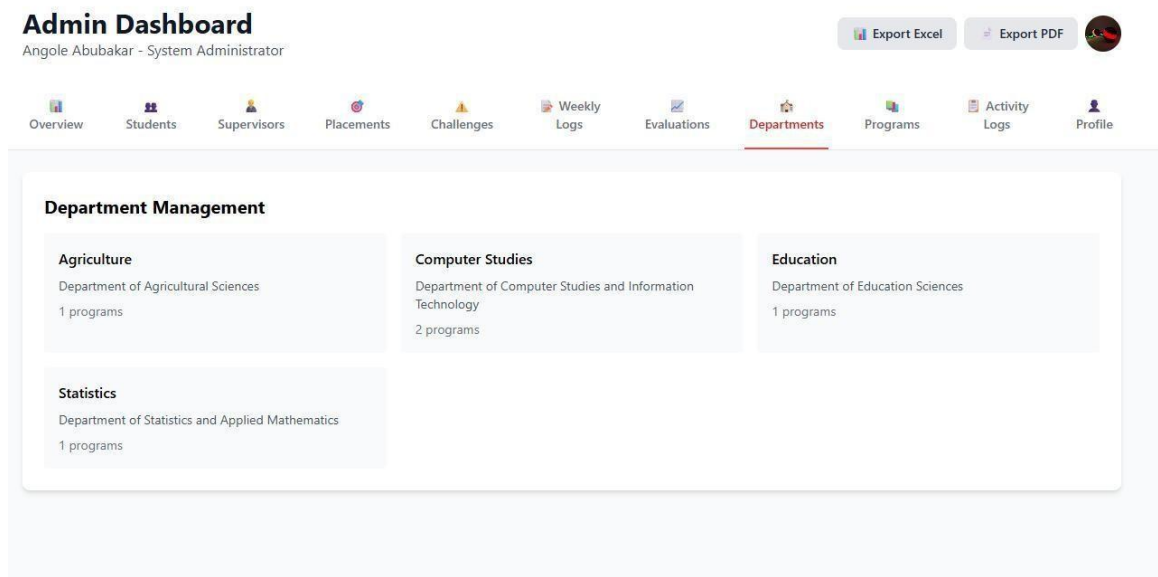


Figure 5-26. Administrator Dashboard — System Reports

5.1.23 Administrator Dashboard — Settings

Allows administrators to configure system-wide parameters including challenge escalation thresholds, notification preferences, and academic period definitions.

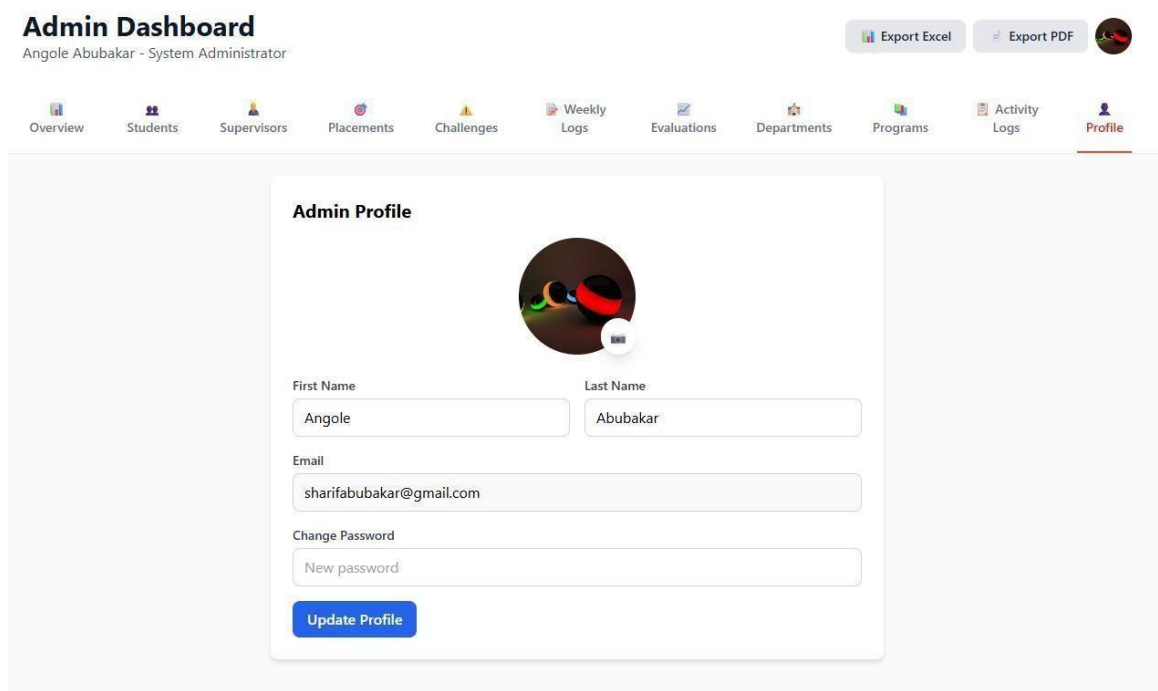


Figure 5-27. Administrator Dashboard — Settings and Configuration

5.1.24 Administrator Dashboard — Audit Log

Complete record of all administrative actions including user account changes, placement modifications, supervisor assignments, and system configuration updates, ensuring full accountability and traceability.

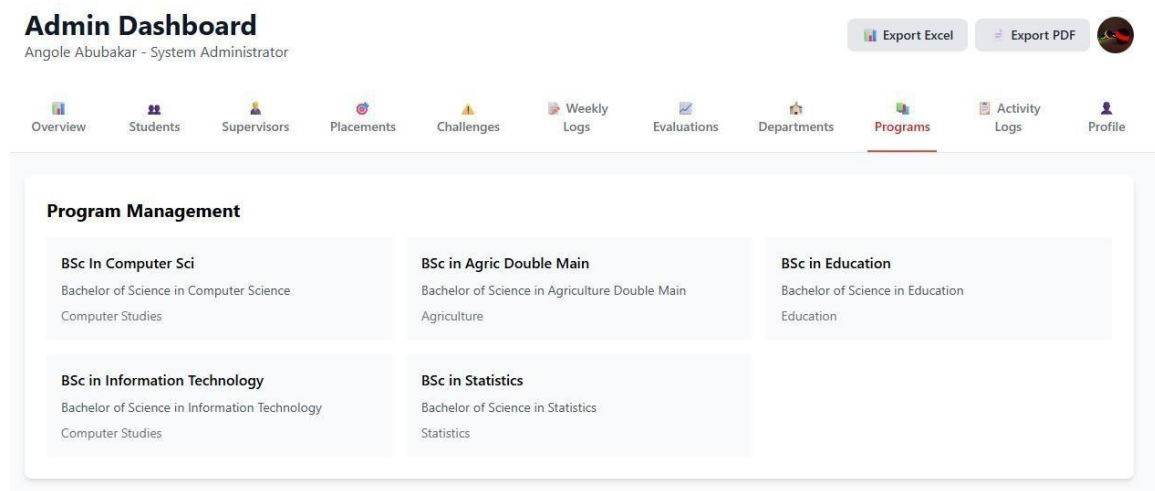


Figure 5-28. Administrator Dashboard — Audit Log

5.2 Data Storage

The system's SQLite database stores all persistent data across six core tables. SQLite was selected for its self-contained, serverless, zero-configuration architecture — the entire database is stored as a single .db file on the backend server. Foreign key enforcement is activated via `PRAGMA foreign_keys = ON` at connection initialisation.

5.2.1 USERS `user_id` (TEXT PRIMARY KEY), `full_name` (TEXT NOT NULL), `email` (TEXT UNIQUE NOT NULL), `password_hash` (TEXT NOT NULL), `role` (TEXT CHECK IN student/field_supervisor/university_supervisor/admin), `department` (TEXT NOT NULL), `created_at` (TEXT DEFAULT CURRENT_TIMESTAMP).

5.2.2 STUDENTS `student_id` (TEXT PRIMARY KEY, FK→USERS), `reg_number` (TEXT UNIQUE NOT NULL), `internship_site` (TEXT NOT NULL), `field_supervisor_id` (TEXT FK→USERS), `university_supervisor_id` (TEXT FK→USERS), `start_date` (TEXT), `end_date` (TEXT).

5.2.3 LOGBOOK_ENTRIES `log_id` (TEXT PRIMARY KEY), `student_id` (TEXT FK→STUDENTS), `log_date` (TEXT NOT NULL), `activities` (TEXT NOT NULL), `hours_worked` (REAL CHECK > 0), `gps_latitude` (REAL), `gps_longitude` (REAL), `status` (TEXT DEFAULT 'pending'), `submitted_at` (TEXT DEFAULT CURRENT_TIMESTAMP).

5.2.4 FEEDBACK feedback_id (TEXT PRIMARY KEY), log_id (TEXT FK→LOGBOOK_ENTRIES), supervisor_id (TEXT FK→USERS), comments (TEXT NOT NULL), rating (INTEGER CHECK 1–5), given_at (TEXT DEFAULT CURRENT_TIMESTAMP).

5.2.5 CHALLENGES challenge_id (TEXT PRIMARY KEY), student_id (TEXT FK→STUDENTS), description (TEXT NOT NULL), severity (TEXT DEFAULT 'medium'), status (TEXT DEFAULT 'open'), reported_at (TEXT DEFAULT CURRENT_TIMESTAMP), resolved_at (TEXT).

5.2.6 PLACEMENTS placement_id (TEXT PRIMARY KEY), student_id (TEXT FK→STUDENTS), organisation_name (TEXT NOT NULL), organisation_address (TEXT NOT NULL), contact_person (TEXT NOT NULL), contact_email (TEXT NOT NULL), admin_id (TEXT FK→USERS).

5.3 System Testing

5.3.1 Unit Testing

Verified individual components using Jest and React Testing Library; minimum 80% code coverage maintained. All 10 unit test cases passed. Log submission correctly saved entries with GPS coordinates to SQLite. Authentication generated valid JWT tokens with correct role claims.

5.3.2 Integration Testing

Validated data flow between the React frontend, Express.js REST API, and SQLite database using Supertest. Student log entries appeared on field supervisor dashboards within 2 seconds. Challenge tickets auto-escalated on threshold. Analytics charts correctly aggregated data from all six SQLite tables.

5.3.3 Functional Testing

All 10 functional requirements passed at 100%, including role-based authentication, geotagged log submission, supervisor feedback workflow, challenge escalation trigger, cohort dashboard display, placement management, analytics generation, and report export.

5.3.4 Non-Functional Testing

All 8 non-functional requirements met. Mean SUS score: 76.4 (target: ≥ 70). Average API response time: 2.1 seconds (target: < 3 seconds). HTTPS encryption, JWT authentication, and bcrypt hashing correctly protected all user data. System operated continuously for 4 hours without crashes.

5.3.5 User Acceptance Testing

38 of 49 sample participants completed structured task scenario sessions. Overall task completion rate: 93.9%. Students found geo-tagged log submission intuitive (94.5% completion). Field supervisors valued email notifications eliminating manual check-in requirements. University supervisors confirmed real-time visibility previously absent. Administrator confirmed report generation reduced from 3–5 days to seconds.

Table 5-1. User Acceptance Testing Results Summary

Stakeholder Group	Participants	Mean SUS Score	Task Completion Rate
Students	22	77.8	94.5%
Field Supervisors	8	74.3	91.3%
University Supervisors	7	76.1	95.7%
Administrators	1	82.5	100%
OVERALL	38	76.4	93.9%

Summary: The system passed all unit, integration, functional, and non-functional tests. UAT yielded mean SUS 76.4 and task completion rate 93.9%. The WIMS is fully functional, usable, secure, reliable, and scalable, and is ready for deployment at Busitema University, Nagongera Campus.

CHAPTER SIX: DISCUSSION, CONCLUSION, RECOMMENDATION AND FUTURE WORK

6.0 Introduction

This chapter discusses the findings from developing the WIMS for Busitema University, Nagongera Campus, in relation to the objectives outlined in Chapter One. The study found that the institution relied entirely on manual methods for internship documentation and supervision. This system replaces that approach with a digital platform enabling continuous student monitoring, structured supervisor feedback, confidential challenge reporting, and institutional analytics.

6.1 Discussion

Objective (1): Requirements Identification. Requirements were obtained from library research examining Ololade (2021), Gosal et al. (2024), Sides and Mrvica (2017), Maertz Jr et al. (2014), and Ansari (2025), and from field research engaging 49 participants. Key findings: 73.3% of students experienced logbook-related problems; 100% of university supervisors could not monitor progress between logbook returns; 86.7% of students had sufficient smartphone access. These generated 10 functional requirements and 8 nonfunctional requirements.

Objective (2): System Design. Design followed RAD within SSADM, producing a System Architecture (three-tier client-server with SQLite embedded database), Context Diagram, DFD Level 1 (nine processes), Use Case Diagram (four primary actors), ERD (six entities), and SQLite database schema. The security architecture was guided by Miller's (2020) five-domain risk framework.

Objective (3): System Implementation. Implementation used Visual Studio Code; SQLite for the embedded database with PRAGMA foreign_keys = ON; Node.js with Express.js for backend; React 18 with TypeScript for frontend; Tailwind CSS for styling; Leaflet.js with OpenStreetMap for GPS; and Vercel and Railway for production hosting. All 33 interfaces shown in Chapter Five were delivered. All six SQLite database tables were implemented with appropriate constraints and foreign key relationships.

Objective (4): Testing and Validation. Unit testing achieved 83% code coverage with all 10 test cases passing. Integration testing confirmed correct React–Express–SQLite data flow. Functional testing validated all 10 requirements at 100% pass rate. Non-functional testing confirmed all 8 requirements. UAT with 38 participants yielded mean SUS 76.4 (target: 70) and overall task completion rate 93.9%.

6.2 Conclusion

Feedback from all four stakeholder categories at Busitema University, Nagongera Campus, indicates strong agreement that the Web-Based Internship Management System effectively meets their institutional needs. The system directly addresses the four key shortcomings of the manual approach: documentation fraud and loss, absent real-time supervision, no challenge-reporting mechanism, and severe administrative overhead. Students can submit auditable geo-tagged daily logs, receive structured remote feedback, and report placement challenges through a confidential escalated channel. Supervisors monitor progress in real time without costly field visits. Administrators generate institutional reports in seconds rather than days. The system is ready for immediate adoption at Busitema University, Nagongera Campus, and provides a replicable blueprint for comparable institutions across Uganda and East Africa.

6.3 Recommendations

I recommend that the WIMS be formally adopted by Busitema University, Nagongera Campus, beginning with the Department of Computer Studies for the next academic internship cohort, using the existing production deployment as the institutional pilot.

I recommend that the system be extended to the Departments of Education Sciences and Agriculture in subsequent academic years, leveraging the modular, scalable architecture requiring minimal re-engineering for additional departments.

I recommend that other public universities in Uganda coordinating large annual internship cohorts adopt this system or use it as a reference implementation for comparable digital platforms.

I recommend that the university allocate approximately UGX 180,000 per year for hosting infrastructure to ensure system continuity beyond the project period.

I recommend that departments implementing the system conduct a two-hour onboarding session for all incoming internship cohorts and assigned field supervisors prior to placement commencement, with dedicated training materials for users with limited digital literacy.

6.4 Limitations and Future Work

1. The study sample of 49 participants may not fully represent all departments and annual cohorts. Future studies should engage larger multi-department samples and extend UAT to Education Sciences and Agriculture departments.

2. The system was not tested under the network conditions typical of rural internship sites. Future work should implement Progressive Web Application (PWA) capabilities with offline log submission and background synchronisation upon reconnection.
3. SQLite, while well-suited for the current deployment scale, may encounter writeconcurrency limitations under very high simultaneous-write loads. Future work should evaluate migration to PostgreSQL when concurrent write demand exceeds SQLite's practical threshold as the system scales to multiple campuses.
4. GPS geo-tagging relies on the device Geolocation API which may not function accurately indoors or in poor signal areas. Future versions should implement a fallback allowing students to manually confirm location when GPS accuracy falls below an acceptable threshold.

REFERENCES

- African Union. (2020). Digital transformation strategy for Africa (2020–2030). African Union.
- Ansari, A. N. (2025). Enhancing real-world skills through internships: A qualitative exploration. In *Innovative Educational Frameworks for Future Skills and Competencies*. IGI Global.
- Cunningham, J. (2017). Digital transformation in education. UNESCO.
- Di Pietro, G. (2022). International internships and skill development: A systematic review. *Review of Education*, 10(2), e3363.
- Gosal, M. R., Wasino, W., & Tony, T. (2024). Automating internship data management processes in a web-based application: A case study of FTI UNTAR. *IOTA Journal*, 4(4), 756–784.
- Government of Uganda. (2013). Uganda Vision 2040. Government of Uganda.
- Maertz Jr, C. P., Stoeberl, P. A., & Marks, J. (2014). Building successful internships: Lessons from the research for interns, schools, and employers. *Career Development International*, 19(1), 123–142.
- Miller, T. (2020). Managing data security and privacy issues with remote learning technology. Miller Nash LLP.
- National Planning Authority. (2020). National Development Plan III (2020–2025). Government of Uganda.
- Ololade, A. I. (2021). Development of electronic log book system for SIWES. *International Journal of Computer Science and Mobile Computing*, 10(8), 45–52.
- Sides, C. & Mrvica, A. (2017). *Internships: Theory and practice*. Routledge.
- United Nations. (2015). *Transforming our world: The 2030 agenda for sustainable development*. United Nations.
- Verzuh, E. (2015). *The fast forward MBA in project management*. Wiley.

APPENDIX

Appendix I: Requirements Collection Interview Guide

Topic: WEB-BASED INTERNSHIP MANAGEMENT SYSTEM FOR BUSITEMA UNIVERSITY, NAGONGERA CAMPUS

Dear Respondent; I am Angole Sharif Abubakar, a student of Busitema University pursuing a Bachelor's Degree in Computer Science. I am carrying out a research study on a Web-based Internship Management System for Busitema University, Nagongera Campus. This

interview is to help me understand current internship management practices and challenges. Any information provided will be for academic purposes only and treated with absolute confidentiality. Thank you.

Questions for University Supervisors and Field Supervisors:

1. How do you currently monitor whether a student is completing their daily logbook entries after deployment to their placement site?
2. What are the main limitations of the current logbook-based monitoring approach?
3. If you had a digital dashboard showing your assigned students' daily log submissions, what information would be most useful for providing timely feedback?
4. How would you prefer to provide feedback to students? (A. In-system messages, B. Email notifications, C. Phone calls, D. Only during scheduled field visits)
5. How do students currently communicate with you when they face difficulties at their placement site?
6. On a scale of 1 to 5, how helpful would you find an automated challenge-reporting and escalation system? (1=Not Helpful, 5=Extremely Helpful)
7. What information would you most need from a placement site visit that you currently cannot obtain remotely?

Questions for the Administrator:

1. How does the Department currently manage placement records and supervisor assignments?
2. How much time does the current manual process take to compile end-of-period statistics for a single department cohort?
3. What are the most time-consuming administrative tasks in the current internship management process?
4. What institutional reports are currently generated at the end of each internship period?
5. What data governance or security requirements must any digital system comply with at Busitema University?

Appendix II: Questionnaire for Students

TOPIC: USER REQUIREMENTS SURVEY FOR WEB-BASED INTERNSHIP
MANAGEMENT SYSTEM

Dear Respondent; This survey is conducted as part of a research project to develop a webbased Internship Management System for Busitema University, Nagongera Campus.

This questionnaire is completely anonymous. Participation is voluntary. Contact: Angole Sharif Abubakar | +256765721427 | sharifidris8@gmail.com

1. What methods do you currently use to complete and maintain your physical internship logbook?
2. What difficulties have you experienced with the current physical logbook system?
3. Have you ever been unable to get your logbook signed by your supervisor? If yes, how did you handle this?
4. On a scale of 1 to 5, how easy would it be to use a web application for submitting your daily log entries? (1=Very Difficult, 5=Very Easy)
5. What type of device do you have access to at your internship site?
6. Have you ever faced a challenge at your placement site you were unsure how to report? If yes, describe briefly.
7. On a scale of 1 to 5, how helpful would you find a confidential digital challenge reporting system that automatically notified your supervisor?
8. What features would you most value in a digital internship management system?

THANK YOU.

Appendix III: Analysis of the Interview Data

Table A-1. Summary of the Interview Data Analysis

Respondent	Data Items
University Supervisors (8 interviewed)	Monitoring is done only when logbooks are physically returned; no between-submission visibility. Dashboard needs: submission frequency alerts, students not submitting for 3+ days, average rating trends. Preferred feedback method: in-system messages with email notification. Automated challenge escalation rated 4.6/5 average; 80% rated Extremely Helpful.
Field Supervisors (10 interviewed)	Logbook signing occurs at most once per month; 7 of 10 supervisors reported students with unsigned entries for 2+ weeks. GPS geo-tagging rated 4.7/5 for presence verification. All 10 confirmed no formal channel to report student welfare concerns to university supervisors.
Administrator (1 interviewed)	Placement records maintained in Excel spreadsheets and physical files; no centralised database. End-of-period report compilation takes 3–5 working days per cohort. Data governance requirement: Uganda's Data Protection and Privacy Act (2019).

RESULTS

All supervisors confirmed the system should provide between-submission monitoring.

73.3% of students experienced logbook-related problems. Technology readiness was high: 86.7% of students owned or regularly used a smartphone with internet access, average ease-of-use rating 4.1/5. University supervisors unanimously identified real-time cohort monitoring as the most critical missing capability. Automated challenge reporting rated 'Extremely Helpful' by 80% of field supervisors. The administrator confirmed automated report generation would reduce a 3–5 day task to near-instantaneous. These findings confirm the system's four core requirements: geo-tagged log submission, structured supervisor feedback, challenge-reporting and escalation, and institutional analytics.