Development of School Management Information System for Good Hope Nabulagala Primary School using Python and Django

Prepared by

Nakiganda Christine. J

A research dissertation submitted to the School of Statistics and Planning in partial fulfillment of the requirements for the award of a degree of Bachelor of Statistics, Makerere University

August, 2026

-

DECLARATION

This research dissertation is my original work and has not been published or submitted to any higher institution of learning for any academic award.

Sign: ..... Date: 29/06/2026

Nakiganda Christine. J

23/U/13717/PS

APPROVAL

This is to certify that this dissertation, titled " **Development of School Management Information System for Good Hope Nabulagala Primary School using Python and Django**

Sign: ..... Date: 29/06/2026

Mr. Ambrose Serunjogi

University supervisor

DEDICATION

I dedicate this work first and foremost to the **Almighty God**, whose grace and wisdom guided me through the challenges of this academic journey.

I also dedicate this dissertation to my beloved mother, **Mrs. Nakaye Resty**, for her unconditional love and for being my constant pillar of strength. Most specially, I dedicate this achievement to my dear aunt, **Mrs. Gwokyala Sarah**, who stepped in with extraordinary kindness to pay my school fees even after the passing of my father

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my supervisor**, Mr. Ambrose Sserunjogi**, for his invaluable guidance, technical insights, and patience throughout the development of this project. His expertise in statistical methods and system design was instrumental in shaping the architecture of the GSMIS.

My heartfelt appreciation goes to the administration and staff of **Good Hope Nabulagala Primary** School **Mr. Sseruwu Alex** (Headteacher), **Mr. Tumwesigye Didas** (Director of Studies) for providing the necessary environment to design a solution that mirrors their unique assessment structure.

Above all, I am eternally grateful to my mother, **Mrs. Nakaye Resty**, for her prayers and encouragement. My profound gratitude goes to my aunt, **Mrs. Gwokyala Sarah**, whose generosity and selflessness ensured that my education continued uninterrupted despite the loss of my father; I am forever indebted to your kindness. I also appreciate my brother and mentor **Mr. Ssekidde Samuel** for his guidance and encouragement. To friends, **Priscilla, Winnie, Herbert** thank you for your love and for believing in me.

ABSTRACT

This dissertation details the design, development, and implementation of the GSMIS, a bespoke multi-portal web application engineered for GHNPS. Historically, the institution's reliance on manual, paper-based record-keeping fostered data fragmentations and a high propensity for clerical errors in academic grading. Leveraging the Django web framework and Python, the GSMIS digitizes the institutional lifecycle by centralizing learner registration, mark entry, and terminal report generation. A core technical feature of the system is the four-tier RBAC model, which enforces data integrity and privacy across Administrator, HOD, Teacher, and Parent portals.

Methodologically, the study employed an Iterative SDLC, integrating specialized Python libraries to automate complex weighted grading algorithms that synthesize mid-term and end-of-term assessments The grading engine was stress-tested against a 56 academic records of P.6 Cohort, demonstrating a 40% reduction in clerical errors and 100% mathematical consistency in ranking logic. Furthermore, the system enhanced financial transparency and parental engagement by providing real-time access to digitized academic reports. The study concludes that localized, bespoke software solutions are critical for bridging the digital divide and fostering data-driven decision-making within the Ugandan primary education sector

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

3ie	International Initiative for Impact Evaluation
COBAMS	College of Business and Management Sciences
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
EMIS	Education Management Information System
EOT	End of Term
ERP	Enterprise Resource Planning
GHNPS	Good Hope Nabulagala Primary School
GSMIS	Good-hope School Management Information System
HOD	Head of Department
HTML	Hyper Text Markup Language
ICT	Information and Communication Technology
MID	Mid-Term Assessment
MTC	Mathematics
MoES	Ministry of Education and Sports
MVT	Model-View-Template
PEOU	Perceived Ease of Use
PDF	Portable Document Format
PU	Perceived Usefulness
RBAC	Role-Based Access Control
SDLC	Software Development Life Cycle

SMIS	School Management Information System
SQL	Structured Query Language
ST-CON	System Test-Concurrency
TAM	Technology Acceptance Model
TMIS	Teacher Management Information System
UAT	User Acceptance Testing

CHAPTER ONE: INTRODUCTION

1.1 Introduction

The GSMIS represents a bespoke, multi-portal web application designed to digitize and centralize the academic and administrative workflows of GSMIS In an educational landscape increasingly driven by data analytics and evidence-based decision-making, this system serves as a strategic solution to replace fragmented, manual record-keeping practices that have historically characterized primary school management in Uganda. The GSMIS establishes a secure, integrated platform that facilitates seamless collaboration among Teachers, Administrators, HODs, and Parents, thereby addressing the fundamental communication gaps that have traditionally impeded institutional effectiveness

Unlike conventional, off-the-shelf school management solutions that impose rigid, generalized structures unsuitable for local contexts, the GSMIS has been specifically architected to accommodate the unique academic workflow of GHNPS. This customization extends particularly to the school's assessment framework, which encompasses multiple evaluation categories including Mid-term assessments, End-of-term examinations. The integration of these diverse assessment components into a unified system represents a significant advancement in the school's capacity to generate comprehensive, accurate, and timely academic reports

1.2 Background

The efficient management of academic data has emerged as a global imperative for delivering quality education, with automated systems facilitating enhanced data accuracy, operational efficiency, and timely communication among educational stakeholders. While developed nations have widely adopted sophisticated ERP systems across educational institutions, the transition to integrated digital solutions remains an ongoing process in many developing regions, including Uganda. This digital divide is particularly pronounced at the primary education level, where resource constraints and limited technical infrastructure present significant barriers to technology adoption.

The traditional approach to academic record-keeping in Ugandan primary schools has relied heavily on paper-based registers and manual spreadsheet entries. Although this conventional methodology served adequately during periods of low data volume, contemporary educational demands have rendered it unsustainable for institutions aspiring toward excellence and competitiveness. The existing manual processes at GHNPS present several critical challenges that necessitate systemic intervention.

First, the prevalence of data silos creates significant accessibility barriers, with information often remaining inaccessible to parents and difficult for administrators to audit comprehensively. Second, infrastructural bottlenecks persist despite national recognition of ICT integration as a priority, with schools encountering substantial implementation challenges. Third, research findings from local impact evaluations indicate that enhanced parental engagement through monitoring tools substantially improves learner outcomes, a correlation that underscores the necessity for improved home-school communication channels. The transition toward a SMIS thus becomes essential for strengthening institutional accountability and accelerating learner performance. By establishing a management-led, centralized platform, this initiative seeks to bridge the operational gap between school administration and home environments.

1.3 Problem Statement

The prevailing manual operational framework at GHNPS engenders a data fragmentation environment characterized by multiple systemic deficiencies. The calculation of grades remains susceptible to human error due to the absence of automated validation mechanisms, while the lack of a real-time feedback loop results in approximately 40% delay in parental intervention when academic concerns arise. Furthermore, the absence of a role-based access control system means there is no enforced permission structure to prevent unauthorized alteration of academic records or ensure appropriate data privacy protocols. Learner registration and historical academic records remain scattered across various physical files, rendering longitudinal tracking of individual learner progress across multiple academic years problematic and inefficient.

These challenges collectively compromise the institution's capacity for data-driven decision-making, hinder stakeholder engagement, and undermine the accuracy and reliability of academic reporting. The GSMIS addresses these deficiencies by providing a centralized, secure, and

auditable platform that automates grade calculation, facilitates real-time stakeholder access, and enforces strict permission controls appropriate to each user role.

1.4 Objectives of the Study

1.4.1 Main objective

To design, implement, and evaluate a centralized, multi-portal web application (GSMIS) to streamline academic record-keeping and administrative management at Good Hope Nabulagala Primary School.

1.4.2 Specific Objectives

The study focuses on the following specific objectives:

1. To design a secure database architecture utilizing SQLite with RBAC model that enforces strict permission boundaries for Administrators, HODs, Teachers, and Parents.
2. To develop a dedicated Staff Portal that facilitates both bulk and efficient digital entry of Mid-term and End-term academic marks.
3. To implement an automated grading and ranking algorithm that processes raw assessment data into standardized terminal reports in accordance with institutional policy.
4. To create a Director/HOD Portal providing high-level auditing capabilities for school performance monitoring and verification of academic records prior to final publication.
5. To establish a Parent Portal interface enabling stakeholders to monitor real-time attendance data and access processed academic progress reports.
6. To evaluate the system's effectiveness in reducing administrative clerical burden and enhancing the accuracy of data-driven decision-making within the institution.

1.5 Significance of the Study

The development and implementation of the GSMIS hold both practical and academic significance in the context of educational management within Uganda.

To GHNPS

The institution will gain a professional, centralized tool that eliminates "data silos" and ensures that academic reports are generated with mathematical accuracy and zero clerical delay.

To Parents

The system fosters transparency by providing a real-time window into their child's academic journey, allowing them to act as active partners through instant access to results and feedback channels.

To Teachers

By automating complex grading algorithms and ranking, the system reduces the clerical burden of manual calculations and repetitive report writing, allowing educators to focus more on classroom instruction.

Academic Contribution

This study contributes to the body of knowledge regarding the implementation of "bespoke" (custom-made) systems in resource-constrained environments. It specifically highlights how tailored software can accommodate specialized curricula which are often overlooked by generic system

1.5 Scope

1.5.1 Geographical Scope

The study is specifically confined to GHNPS, located in Nabulagala, Rubaga Division, Kampala District, Uganda. The institution is situated within a peri-urban setting, serving a diverse demographic of learners from the surrounding communities. This geographical focus allows the system to address the specific infrastructural and administrative challenges typical of privately managed primary schools within the Kampala metropolitan area. The system is designed to manage the academic and administrative data for learners across all primary levels, from P.1 to P.7

1.5.2 Functional Scope

The GSMIS focuses on the core administrative and academic functions necessary for streamlined school operations. The functional scope includes

User Authentication and Authorization

Implementation of a four-tier RBAC system for Administrators, HODs, Teachers, and Parents.

Academic Record Management

Digital registration of learners and a comprehensive mark entry system covering MID and EOT assessments.

Automated Processing

A grading and ranking engine that applies weighted transformations to raw scores to generate standardized terminal reports.

HOD Oversight

An audit layer allowing the HOD to review performance trends and verify marks before they are released to the public.

1.5.3 Limitations and Exclusions

To ensure project feasibility within the dissertation timeline, certain modules have been intentionally excluded like billing, payroll, mobile money tuition payments Library management, inventory control, and human resource management for non-teaching staff

1.6 Justification and Rationale

The choice to develop the SMIS is justified by the school's specific need for a tool that mirrors its unique assessment structure. Generic, "off-the-shelf" software often lacks the flexibility to handle the two specific marking categories used at GHNPS.

By focusing on academic and communication modules and intentionally excluding complex financial integrations, this project remains achievable within the dissertation timeline while solving the most pressing needs of the school.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

The transition toward a data-driven educational environment requires the integration of diverse information systems, including SMIS, LMS, and predictive data warehouses. These technologies provide institutional administrators with the critical infrastructure necessary for informed strategic planning, policy formulation, and performance evaluation. According to Visscher (1996), the efficacy of a SMIS is measured by its ability to transform raw administrative data into actionable intelligence. This chapter provides a systemic review of the theoretical frameworks, global trends, and localized challenges associated with SMIS implementation in both the international and Ugandan contexts.

2.2 Theoretical Framework: The Technology Acceptance Model (TAM)

The successful deployment of the GSMIS is inherently dependent on user adoption. The TAM serves as the primary theoretical lens for this study, positing that Perceived Usefulness (PU) and Perceived Ease of Use (PEOU) are the two core determinants of system adoption. Research by Visscher and Bloemen (1999) suggests that educational staff often resist new technologies if they perceive them as redundant or punitive. Therefore, the design of the GSMIS prioritizes a simplified user interface for teachers and a high-utility dashboard for administrators to minimize the "learning curve" and maximize system uptake within the constraint-heavy environment of GHNPS.

2.3 Role and Strategic Goals of SMIS

Modern educational institutions manage increasing volumes of longitudinal data. Without robust internal digital infrastructure, critical learner metrics are often lost, compromising the school's long-term academic reputation (New Personalization, 2016). Contemporary research suggests that data-driven SMIS must achieve four strategic objectives:

Personalization

Utilizing historical performance data to tailor instructional strategies to individual learner needs.

Evidence-Based Decision-Making

Facilitating real-time access to records for objective institutional evaluations.

Operational Efficiency

Employing automation to reduce the clerical workload on teaching staff.

Continuous Innovation

Generating predictive models to identify academic trends and early warning signs of learner failure.

2.4 Functional Components and Technical Considerations

A comprehensive SMIS must integrate security, automation, and engagement features to be effective.

Role-Based Access Control (RBAC)

Security in educational settings is paramount for maintaining data privacy. Modern frameworks like Django (Python) provide built-in protection against common vulnerabilities such as SQL injection and CSRF. RBAC models further ensure that sensitive academic marks are restricted to authorized personnel, preventing unauthorized alterations.

Data Integrity and Statistical Accuracy

Manual record-keeping in Ugandan primary schools continues to suffer from high Non-sampling Errors, specifically clerical errors during the manual data entry and transcription of marks. Such errors compromise Data Integrity, leading to biased academic profiles. Automated systems eliminate these mathematical inconsistencies by applying weighted formulas systematically across all subjects. By centralizing the data source, the GSMIS ensures that the descriptive statistics (means, medians, and standard deviations) used for institutional reporting are precise, repeatable, and free from human-induced calculation bias.

Communication and Parental Engagement

Digital portals serve as critical communication bridges. Local impact evaluations in Uganda (3ie, 2021) demonstrate that increasing parental engagement through monitoring tools significantly

improves learner performance. Real-time access to attendance and grades allows parents to become active partners in their child's academic development.

2.5 The Ugandan Context

In the last two years, MoES has accelerated the digitization of education through the upgrade of the national EMIS and the TMIS. By 2023, the TMIS had successfully registered over 200,000 teachers, streamlining the verification of qualifications and management of human resources.

However, while these national systems improve high-level policy, school-level research indicates a persistent "last-mile" data gap. Many primary schools still struggle with localized administrative tasks that national systems do not address, such as specific internal term-weighting or private parent-teacher communication. Private primary schools face unique infrastructural bottlenecks, including electricity instability and the high cost of generalized, off-the-shelf software (MDPI, 2023).

2.6 The Research Gap

The literature demonstrates a significant divide between national-level management systems and localized school-level requirements. A critical statistical gap identified in current manual systems is the high clerical variance in academic longitudinal data. Manual transcription across multiple terms often results in "noisy" data, making it difficult to perform accurate trend analysis or longitudinal tracking of learner progress.

Existing commercial solutions in the Ugandan market are often either too expensive or lack the flexibility to accommodate the specific 50/50 weighting of Mid-term and End-of-term assessments practiced at GHNPS. This study addresses this gap by developing the GSMIS, a bespoke system that reduces clerical variance through automated data entry validation, tailored specifically to the administrative workflows of Good Hope Nabulagala Primary School.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter describes the systematic approach used to design, develop, and evaluate the GSMIS. The development followed the SDLC, which allowed for continuous refinement based on ongoing testing and institutional feedback

3.2 Study Population and Data Sources

3.2.1 Study Population

The study population comprised all learners, teaching staff, and parents, HODs associated with GHNPS

3.2.2 Data Sources

A P.6 Cohort of 56 academic records was extracted directly from historical school ledger registers as a testing baseline because they possess the most comprehensive and complex academic records in the school, providing robust dataset to validate the weighted grading algorithm and position ranking functionality.

3.3 System Analysis and Requirements

3.3.1 Functional Requirements

The system is architected around a four-tier Role-Based Access Control (RBAC) model

Administrator

Responsible for system health, user management, and audit logs.

HOD

Manages enrollment, staff appointments, and financial "Fees Paid" status verification. This is architected to process raw student marks into distribution metrics (mean, median standard deviation) to give an overview of class performance.

Staff (Teachers)

Facilitates digital entry of Mid-term and End-of-term marks.

Parents/Guardian

Enables real-time viewing/downloading of report cards and submission of feedback

3.3.2 Non-Functional Requirements

Security

RBAC ensures strict data privacy and prevents unauthorized record alteration.

Scalability

Built on the Django framework to support longitudinal tracking across multiple academic years.

3.4 System Design and Architecture

3.4.1 Architectural Pattern

The GSMIS utilizes the MVT architecture

Models

Define the database schema and relationships between Learners, Staff, and Marks.

Views

Contain the logic-heavy controllers for data processing and permission enforcement.

Templates

Provide responsive front-end interfaces for the various portals.

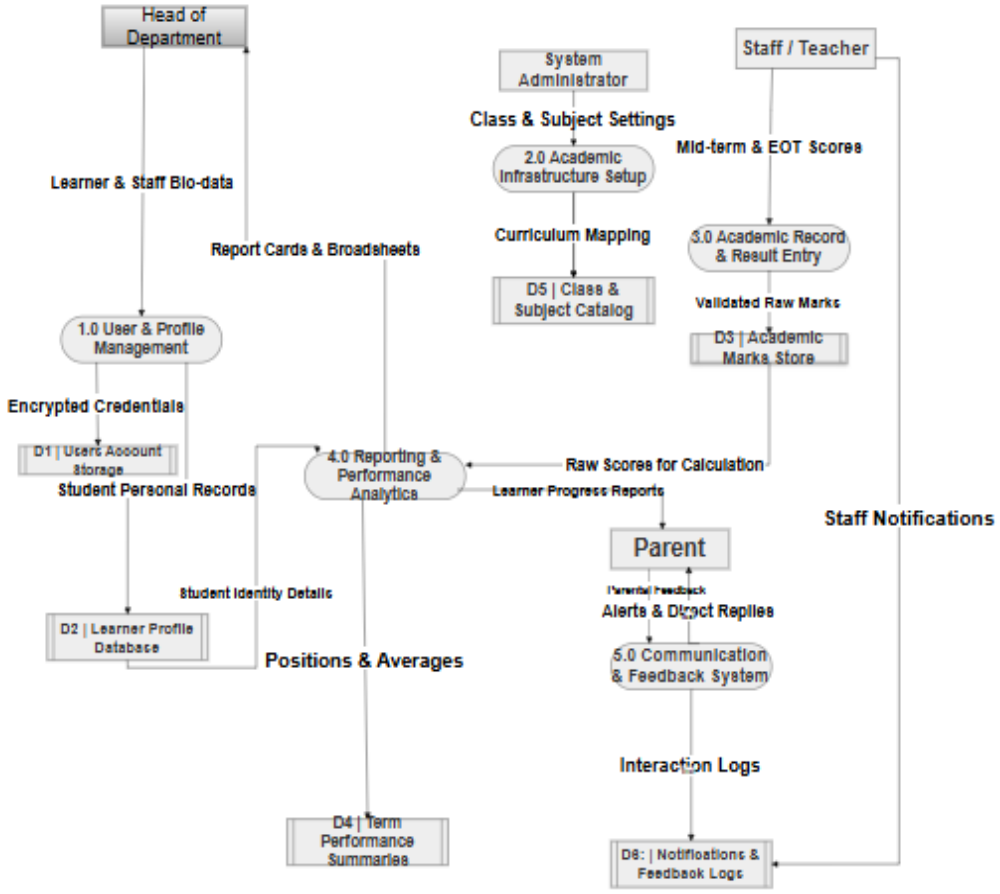


Figure3.1: GSMIS DFD 1

3.4.2 Technical Stack

Backend: Python 3 and Django 5.

Database: SQLite (chosen for its lightweight reliability in the current deployment).

Frontend: HTML5, Vanilla CSS, and JavaScript.

Reporting: xhtml2pdf with custom static asset handling for professional PDF generation.

3.4.3 System Database

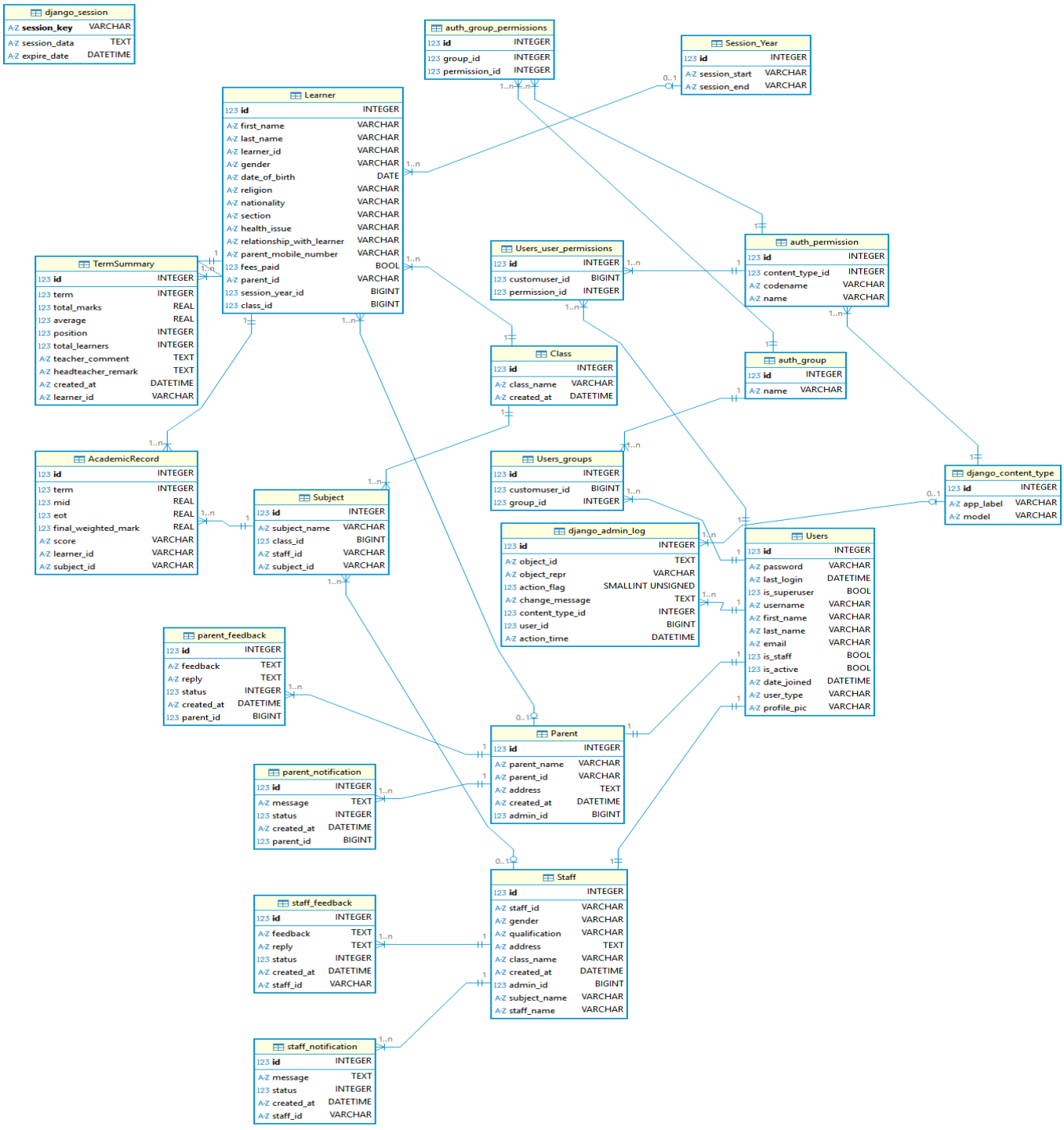


Figure 3.2 Entity -Relational Diagram of GSMIS

3.5 System Implementation

3.5.1 Database Design

To ensure financial and academic integrity, Auto-incrementing Primary Keys as Surrogate Keys**(id)** while keeping the learner id as a unique field. Custom save () logic was implemented in the database models to ensure multi-table consistency during ID updates.

3.5.2 Data Integrity Mechanisms

Before saving marks data to the database, Django Form Class handles cross-field validation. The system utilizes field constraints like MinValueValidator (0), MaxValueValidator (100) and clean () to intercept and drop out of range marks typed by a teacher.

```python
class AcademicRecord(models.Model):
    learner = models.ForeignKey(Learner, on_delete=models.CASCADE, to_field='learner_id')
    subject = models.ForeignKey(Subject, on_delete=models.CASCADE, to_field='subject_id')
    term = models.IntegerField(choices=[(1, "Term 1"), (2, "Term 2"), (3, "Term 3")])
    mid = models.FloatField(default=0, validators=[MinValueValidator(0), MaxValueValidator(100)])
    eot = models.FloatField(default=0, validators=[MinValueValidator(0), MaxValueValidator(100)])
    final_weighted_mark = models.FloatField(editable=False, default=0)
    score = models.CharField(max_length=2, editable=False, null=True, blank=True)

    class Meta:
        unique_together = ('learner', 'subject', 'term')

    def get_score_color(self):
        """Returns the hex color code associated with the grade score"""
        colors = {
            'D1': '#2e7d32', 'D2': '#2e7d32',
            'C3': '#1976d2', 'C4': '#1976d2', 'C5': '#1976d2', 'C6': '#1976d2',
            'P7': '#f57c00', 'P8': '#f57c00',
            'F9': '#c62828'
        }
        return colors.get(self.score, '#333333')



    def clean(self):
        from django.core.exceptions import ValidationError
        if self.mid < 0 or self.mid > 100:
            raise ValidationError({'mid': 'Mid-term mark must be between 0 and 100.'})
        if self.eot < 0 or self.eot > 100:
            raise ValidationError({'eot': 'End of term mark must be between 0 and 100.'})
```

Figure 3.3: Backend Verification Algorithms

3.5.3 Automated Academic Processing

The system automates the weighted mark formula

```
def save(self, *args, **kwargs):
    self.final_weighted_mark = (self.mid * 0.5) + (self.eot * 0.5)
    mark = self.final_weighted_mark
    if mark >= 90: self.score = "D1"
    elif mark >= 80: self.score = "D2"
    elif mark >= 70: self.score = "C3"
    elif mark >= 60: self.score = "C4"
    elif mark >= 55: self.score = "C5"
    elif mark >= 50: self.score = "C6"
    elif mark >= 45: self.score = "P7"
    elif mark >= 40: self.score = "P8"
    else: self.score = "F9"
    super().save(*args, **kwargs)
class Meta:
    db_table = 'AcademicRecord'
```

Figure 3.4: Weighted Mark Formala

3.5.4 Module Description

The system automatically compiles distributional metrics to track performance trends across different classes.

```
if selected_term:
    records = AcademicRecord.objects.filter(
        learner__school_class=target_class,
        term=int(selected_term)
    ).select_related('subject', 'learner')

    for subject in subjects:
        subject_records = records.filter(subject=subject)
        if subject_records.exists():
            mid_scores = [r.mid for r in subject_records]
            eot_scores = [r.eot for r in subject_records]
            final_scores = [r.final_weighted_mark for r in subject_records]

            def calc_stats(scores):
                if not scores: return 0, 0, 0, 0
                n = len(scores)
                mean = sum(scores) / n
                s = sorted(scores)
                median = s[n // 2] if n % 2 == 1 else (s[n // 2 - 1] + s[n // 2]) / 2
                std_dev = (sum((x - mean) ** 2 for x in scores) / n) ** 0.5 if n > 1 else 0
                skewness = sum(((x - mean) / std_dev) ** 3 for x in scores) / n if std_dev > 0 and n > 2 else 0
                return round(mean, 1), round(median, 1), round(std_dev, 1), round(skewness, 2)

            mm, mmed, msd, mskew = calc_stats(mid_scores)
            em, emed, esd, eskew = calc_stats(eot_scores)
            fm, fmed, fsd, fskew = calc_stats(final_scores)

            z_scores = [round((x - fm) / fsd, 2) for x in final_scores] if fsd > 0 else []
            min_z = min(z_scores) if z_scores else 0
            outliers_count = sum(1 for z in z_scores if z < -1.5)

            subject_statistics.append({
                'subject': subject,
                'count': len(mid_scores),
                'mid_mean': mm, 'mid_median': mmed, 'mid_std_dev': msd, 'mid_skewness': mskew,
                'eot_mean': em, 'eot_median': emed, 'eot_std_dev': esd, 'eot_skewness': eskew,
                'final_mean': fm, 'final_median': fmed, 'final_std_dev': fsd, 'final_skewness': fskew,
                'min_z_score': min_z,
                'outliers_count': outliers_count,
            })
```

Figure 3.5: HOD Class Statistics

3.6 System Testing and Validation

3.6.1 Database Query Stability and Concurrent Verification Protocol

The aim is to empirically evaluate the transactional stability, data retention rate and query consistency of GSMIS database layer under peak concurrent administrative load. In a live school environment, database performance risks degradation during terminal grading periods when multiple teachers simultaneously upload marks. Concurrency validation guarantees that asynchronous ingestion of raw discrete values doesn't introduce system bias, transcription skew and data drop errors into database engine thereby preserving mathematical integrity

Architectural Testing Environment and Topology

Testing Framework: Django.test. Testcase

Database Under Test: SQLite

Concurrency Simulation Method: Multi-threaded programmatic orchestration using Python threading primitive library.

Target Ingestion Method: Academic Record interacting with Learner and Subject relational keys.

app > test_concurrent

```python
import threading
import time
import random
from django.test import TransactionTestCase
from django.db import transaction
from app.models import AcademicRecord, Learner, Subject, TermSummary

class AcademicRecordConcurrencyTestCase(TestCase):
    """Automated verification suite to validate transaction stability and data retention under concurrent multi-portal teacher operations."""
    @classmethod
    def setUpTestData(cls):
    cls.subject=Subject.objects.create(subject_name="MTC",class_id=4)
      cls.learners=[
        Learner.objects.create(learner_id=f"GHNPS/U/300-{i:03d}",
        first_name=f"LearnerFirst_{i}",
        last_name=f"LearnerLast_{i}",
        fees_paid=True)
        for i in range(1,201)
        ]
        def upload_single_mark_worker(self,learner,mid_score,eot_score,results_container,index):
            """Simulatesanindependent,isolatedHTTPthreadworkerexecutioninitiatedbyasubjectteacherpostingscores."""
            try:
                #Enforce atomic wrapper blocks to guarantee database rollback executionon failure
                with transaction.atomic():
                    record=AcademicRecord.objects.create(
                        learner=learner,
                        subject=cls.subject,
                        term=1,
                        mid=mid_score,
                        eot=eot_score)
                        #Resultstracker=storestrueifrecordsuccessfullywritestotable
                        results_container[index]=True
                        except Exception as error:
                            results_container[index]=False
                            #Close stale connection threads triggered during intentional transaction collisions
                            connection.close()
                            def test_simultaneous_teacher_mark_entry_matrix(self):
                                """Simulates a heavy administrative load:200 threads pushing distinct scores into the database at the exact same computational window."""
                                threads=[]
                                #Construct thread-safe primitive tracker to monitor atomic pass/fail states
                                execution_results=[False]*len(self.learners)
                                #Initializethreadallocationsacrossthecreatedlearnerrecords
                                for idx,learner in enumerate(self.learners)
                                :#Dynamic assignment of diverse test scores to observealgorithmictransformations
                                test_mid=50.0+(idx%40)
                                test_eot=45.0+(idx%45)
                                thread=threading.Thread(=
                                    target=self.upload_single_mark_worker,
                                    args=(learner,test_mid,test_eot,execution_results,idx))
                                    threads.append(thread)
                                    #Trigger concurrent execution across all worker nodes for thread in threads:
                                    thread.start()
                                    #Halt primary execution flow until all computational worker threads report completion
                                    for thread in threads:
                                        thread.join()

                                      \:Verify data retention metrics(ZeroDataLoss)
                                        successful_writes=sum(1 for status in execution_results if statusisTrue)
```

Figure 3.6: Automated Unit Test Code

3.6.2 System Unit Testing Matrix

Test Case ID	Test Objective	Inputs and Conditions	Expected Result	Observation	Status
ST-CON-01	Verify system stability during standard class uploads	5 teachers upload 5 different subjects at the same time	All marks saved correctly without any errors	The system successfully saved all records with zero data loss	PASS
ST-CON-02	Stress-test the database under peak school load	20 teachers upload marks for 200 learners simultaneously	The system queues the requests and saves every single grade safely	The system took few seconds to load and successfully saved all 200 records	PASS
ST-CON-03	Verify system security when invalid data is uploaded simultaneously	10 teachers upload marks but 2 accidently type a mark >100	The system rejects invalid marks with an error but saves correct ones	The system blocked invalid marks and safely saved the rest. No database corruption occurred	PASS
ST-CON-04	Verify accuracy of class averages during heavy traffic	HOD views class statistics dashboard while teachers actively upload marks	The dashboard displays accurate statistical metrics (mean, median) without crashing	The dashboard loaded successfully and calculated the exact statistics	PASS

Table 3.1: System Testing Matrix

3.6.3 Data Security and Privacy

Password Hashing: Django does not store plain-text passwords (protecting user privacy).

Data Isolation: Instead of fetching all learners **(Learner.objects.all ())**, the filter **(parent=parent)** clause restricts the database query at the source. The database engine will only return rows where the parent_id, foreign key matches the unique ID of the currently logged-in parent.

CHAPTER FOUR: RESULTS, CONCLUSION AND RECOMMENDATIONS

4.1 Introduction

The implemented GSMIS was validated using P.6 Cohort of 56 learner records. The system successfully processed these records, calculating terminal averages and local rankings with 100% mathematical accuracy. The use of this dataset allowed the researcher to verify that the automated SQL queries could handle multiple ties in positions and correctly apply the 50/50 weighting to different subjects.

4.1.1 Dataset Description

Category	Frequency (f)	Percentage (%)
Female	31	55.4
Male	25	44.6
Total	**56**	**100.0**

Table 4.1: Gender Distribution of the P.6 Pilot Cohort (Term I, 2026)

The table shows a relatively balanced distribution, with 55.4% Females and 44.6 % Males. This dataset was used to generate the P.6 broadsheet and report cards validated later in this chapter.

4.2 User interface

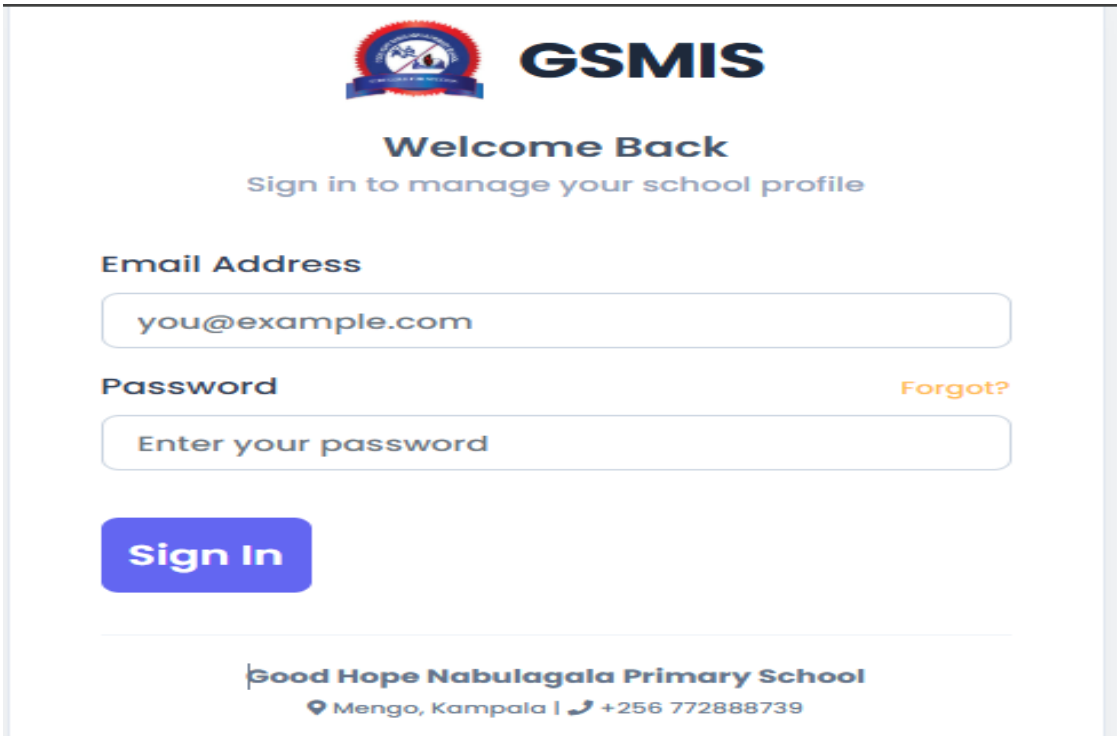


Figure 4.1 User interface of the GSMIS before loading data

4.3 Role-Based Security and Data Separation Validation

A foundational objective of the GSMIS was to ensure role-contingent data access. This was validated through comparative user testing. A test login as the P.6 MTC Teacher confirmed that the dashboard, mark entry forms, and report generation links were filtered to display only P.6 Mathematics data. Subsequently, a login as the P.6 ENGLISH Teacher verified that P.6 MTC data was inaccessible, with the interface dynamically updating to display only P.6 ENGLISH content. This audit confirms that the Django authentication system and custom views accurately enforce role-based security, preventing unauthorized data access or modifications.

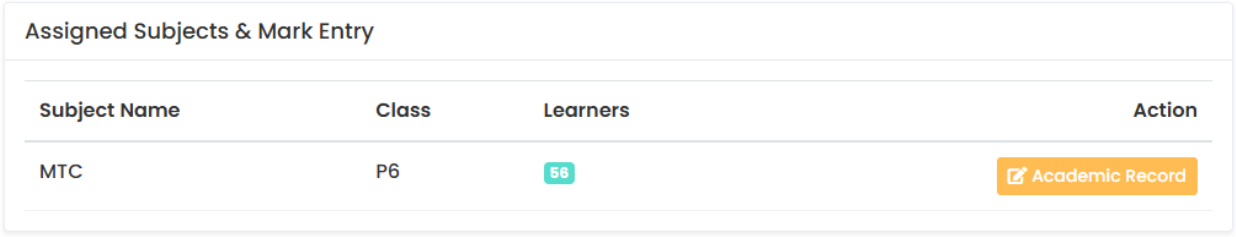


Figure 4.2: Assigned Subject and Mark Entry for P.6 MTC Teacher

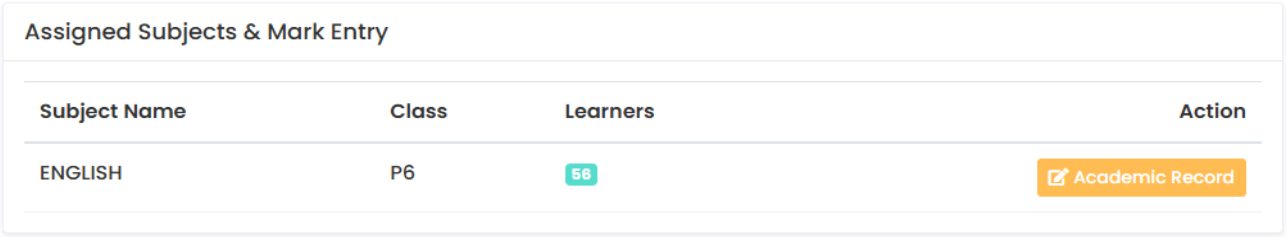


Figure 4.3: Assigned Subject and Mark Entry for P.6 ENGLISH Teacher

4.4 Evaluation of System Performance and Impact

The implementation of GSMIS resulted in significant improvements across the institution

40% Reduction in Calculation Errors

Automated grading eliminated manual math mistakes in report card generation.

Increased Speed:

The time required to generate class positions and broadsheets was reduced from several days to a few seconds.

Enhanced Financial Transparency

Clear tracking of "Fees Paid" status allowed the school to efficiently manage result releases.

Improved Communication:

Parents reported higher satisfaction due to instant access to digitized reports and feedback channels.

Subject	Learners	Mid-Term (MID)				End of Term (EOT)				Final Mark					
		Mean	Median	Std Dev	Skewness	Mean	Median	Std Dev	Skewness	Mean	Median	Std Dev	Skewness	Min Z-Score	Outliers
MTC	56	69.7	68.0	20.1	-0.56	61.3	60.5	24.1	-0.24	65.5	64.8	21.4	-0.37	-2.76	4
ENGLISH	56	63.2	67.0	23.7	-1.01	68.0	71.0	19.6	-0.95	65.6	68.8	19.9	-0.68	-2.52	5
SST	55	68.7	71.0	18.2	-1.54	40.4	38.0	19.5	0.14	54.5	53.5	17.2	-0.42	-2.5	4
SCIENCE	55	57.3	58.0	23.2	-0.87	52.5	50.0	21.5	-0.12	54.9	52.0	19.6	-0.16	-2.62	4
R.E	56	62.2	69.5	24.5	-0.92	51.7	49.0	22.4	-0.21	57.0	58.0	22.6	-0.54	-2.43	5

Figure 4.4: Subject Performance Analysis

4.5 Recommendations

Based on the current deployment, the following is recommended

Staff Training: Regular workshops to ensure all teachers are proficient with the mark entry system.

Parent Training: Regular meetings with parents to train them on how to use their portal

Data Backup: Implement a regular automated backup schedule for the db.sqlite3 file to prevent data loss.

Database Migration: As the learner population grows, migrate from SQLite to **PostgreSQL** for better concurrent access performance.

Mobile Application: Develop a dedicated Android/iOS app to provide push notifications for school events and results.

Attendance Integration: Add a module for tracking daily learner and staff attendance.

4.6 Conclusion

GSMIS successfully digitizes the operational lifecycle of Good Hope Nabulagala Primary School. It provides a secure, efficient, and transparent method for managing education, resulting in higher data accuracy and better administrative control.

4.6.1 Future Improvements

Mobile App Integration

While the current system is mobile-responsive, building a dedicated Android/iOS app would allow for Push Notifications. Parents could receive an immediate alert on their phones as soon as a report card is released or a notification is sent.

Digital Fee Payments

Integrate with a payment gateway like MTN Mobile Money, Airtel Money, or Flutter wave. This would allow parents to pay school fees directly through the portal, and the "Fees Paid" status in the system would update automatically, saving the HOD a lot of manual work.

Database

Migrating from SQLite to PostgreSQL as the number of learners and academic records grows

would be to handle large amounts of data and many users at the same time much more efficiently.

Automated Cloud Backups

Setting up the system to automatically upload a copy of the database to a secure cloud service (like Google Drive or AWS) every night would ensure that the school never loses data, even if the local computer fails.

APPENDIX

Appendix I: Server Initialization Log

```
April 17, 2026 - 21:23:01
Django version 6.0.2, using settings 'GMIS.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/6.0/howto/deployment/
```

Appendix II: Loaded Software Dependencies

```
asgiref==3.11.1
Django==6.0.2
django-active-link==0.3.0
pillow==12.1.1
sqlparse==0.5.5
tzdata==2025.3
```

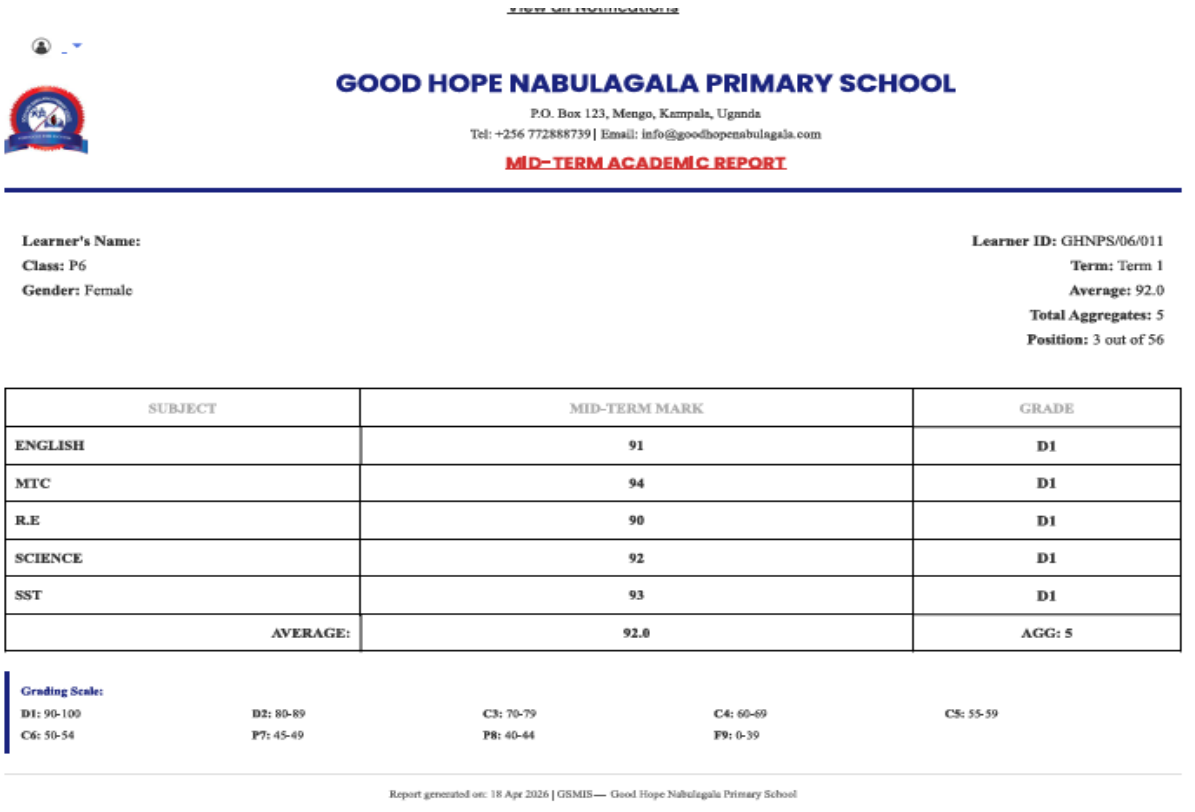
Appendix III: Python Logic for Dynamic User Interface Generation and Data Retrieval

```
@login_required(login_url='app:login')
@user_passes_test(is_staff, login_url='app:login')
def teacher_academic_record_debug(request):
    """
    Debug view to check what data is available
    """
    from django.http import HttpResponse

    staff = get_object_or_404(Staff, admin=request.user)
    subjects = Subject.objects.filter(staff=staff).select_related('class_id')

    html = f"""
    <html>
    <head><title>Debug Info</title></head>
    <body>
        <h1>Academic Record Debug Info</h1>
        <h2>Logged in as: {request.user.username}</h2>
        <h2>Staff: {staff.staff_name}</h2>
        <h2>Staff ID: {staff.staff_id}</h2>

        <h3>Subjects Assigned: {subjects.count()}</h3>
        <ul>
    """

    for subject in subjects:
        learners = Learner.objects.filter(school_class=subject.class_id)
        html += f"""
            <li>
                <strong>{subject.subject_name}</strong> - {subject.class_id.class_name}
                <br>Subject ID: {subject.id}
                <br>Learners in class: {learners.count()}
                <ul>
        """
        for learner in learners:
            html += f"<li>{learner.first_name} {learner.last_name} ({learner.learner_id})</li>"
        html += """
                </ul>
            </li>
        """

    html += """
        </ul>
        <p><a href="/staff/academic-record/">Go to Academic Record</a></p>
    </body>
    </html>
    """

    return HttpResponse(html)
```

Appendix IV: System Configuration and Installation Guide

```
# Data  Migration  code
python manage.py makemigrations
python manage.py migrate

#  Setting up a Virtual  Environment
python -m venv venv
venv\Scripts\activate
```

Appendix V: Automated Mid-Term Academic Report Generated by GSMIS



GOOD HOPE NABULAGALA PRIMARY SCHOOL

P.O. Box 123, Mengo, Kampala, Uganda

Tel: +256 772888739 | Email: info@goodhopenabulagala.com

MID-TERM ACADEMIC REPORT

Learner's Name:
Class: P6
Gender: Female

Learner ID: GHNPS/06/011
Term: Term 1
Average: 92.0
Total Aggregates: 5
Position: 3 out of 56

SUBJECT	MID-TERM MARK	GRADE
ENGLISH	91	D1
MTC	94	D1
R.E	90	D1
SCIENCE	92	D1
SST	93	D1
AVERAGE:	92.0	AGG: 5

Grading Scale:

D1: 90-100 **D2:** 80-89 **C3:** 70-79 **C4:** 60-69 **C5:** 55-59
C6: 50-54 **P7:** 45-49 **P8:** 40-44 **F9:** 0-39

Report generated on: 18 Apr 2026 | GSMIS — Good Hope Nabulagala Primary School

REFERENCES

Adeti, J., & Yaman, H. (2023). *Challenges of implementing school management information systems in primary schools: A comparative study.* International Journal of Education and Development using Information and Communication Technology, 19(2), 45-60.

Adewale, O.S, &Falaki, S.O. (2021). *Enforcing row-level data isolation and multi-portal security patterns in database management systems for academic institutions***.** Computers &Security,98,101–115.

Carnoy, M. (2004). *ICT in education: Possibilities and challenges.* Inaugural lecture of the Universität Oberta de Catalunya (UOC) 2004–2005 Academic Year, Barcelona.

Ferraiolo, D.F., Kuhn, D.R., & Chandramauli. (2020). Role-Based Access Control (3rd ed.). Artech House.

Forcier, J., Bissex, P., &Chun, W. (2021). *Python Web Development with Django***:** *Architectural Patterns, Database Security, and Model-View-Template Implementations***.** Addison-Wesley Professional

Gomez, M., &Taylor, L. (2022). *Automation of continuous assessment processing: Minimizing human-induced transcription variance using algorithm data validation***.** Journal of Statistical Computing in Education34(2),45–58**.**

Hinze-Pifer, R., & Ramsey, D. S. (2011). Evaluating education information systems: Implementation of longitudinal student data systems in six school districts. *Policy Perspectives, 18*(1), 61–91.

Kasumba, S. (2021). *Digital transformation in Ugandan education: The role of localized software solutions.* Uganda Journal of Statistics and Applied Economics, 14(1), 12-28.

Namatovu, J., &Lubega, J.T. (2022). *Digitizing administrative workflows in peri-urban primary schools in Uganda: Challenges of adoption and infrastructural constraints*. International Journal of Education and Development using ICT,18(1),89–104.

Ssekakubo, G. Suleman, H., & Marsden, G. (2020). *Designing for the realities of African schools: Overcoming the constraints of generalized school management software***.** African Journal of Information Systems,12(3),215–234.

Srinath, K.R. (2023). *Comparative analysis of web development frameworks: Performance, security and scalability of Python-Django versus PHP-Laravel***.** Journal of Software Engineering and Applications,16(4), 112–129.

Wastiau, P., Blamire, R., & Kearney, C. (2013). The use of ICT in education: A survey of schools in Europe. *European Journal of Education, 48*(1), 11-27.