

# A Digital Student Enrollment and Registration System: Design, Implementation, and Database Integration Using Python and SQLite

Albancia, Jake F., Arañas, Mica Angela A., Camba, Princess Mae Liz D., Corbe, Gaveriel L., Estella, Prince Justine Rose A., Garcia, Frances Mickyla M., Engr. Fabro, Meshelle N.

Department of Computer Engineering Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST) Nagtahan, Manila, Philippines

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150600088>

Received: 10 June 2026; Accepted: 15 June 2026; Published: 07 July 2026

## ABSTRACT

Manual student enrollment and registration processes in educational institutions are often time-consuming, prone to errors, and inefficient. This study developed a **Digital Student Enrollment and Registration System** using Python and SQLite to automate student record management and course registration. The system allows administrators to add, view, update, delete, and search student records, as well as manage enrolled subjects in a structured database. The application uses Object-Oriented Programming (OOP) principles and a command-line interface for ease of operation. Functional testing demonstrated that the system successfully performed all core operations, maintained data integrity, and minimized clerical errors. By implementing database integration, the system ensures permanent storage, reliable retrieval, and organized management of student enrollment information. The study demonstrates how digital systems can improve administrative efficiency and enhance the overall enrollment experience for both students and registrar personnel.

**Keywords:** Student Enrollment, Registration System, Python, SQLite, Database Integration, OOP, Academic Records

## INTRODUCTION

### Background of the Study

Educational institutions continuously seek efficient and reliable methods for managing student enrollment and registration processes. Traditionally, many schools still rely on manual or semi-manual procedures involving paper forms, physical verification, handwritten records, and spreadsheet-based encoding. Although these methods have been used for decades, they are often associated with long waiting times, data redundancy, record mismanagement, delayed transactions, and administrative inefficiencies. In higher education institutions such as the Eulogio Amang Rodriguez Institute of Science and Technology (EARIST), the enrollment process becomes particularly challenging during peak registration periods due to the increasing number of students and limited administrative resources.

According to Chamilco et al. (2023), traditional enrollment systems contribute to congestion, repetitive clerical tasks, and increased human errors, which negatively affect institutional efficiency and student satisfaction. Similarly, Aspa et al. (2025) emphasized that manual enrollment procedures often result in slow processing times, inaccurate student records, and inefficient communication between departments. These operational issues demonstrate the need for a computerized enrollment and registration solution capable of automating administrative workflows while ensuring accuracy and reliability.

The advancement of information technology has enabled educational institutions to adopt digital enrollment systems that streamline student registration activities through automation, database integration, and real-time data processing. Modern enrollment systems utilize programming languages, database management systems, and graphical user interfaces to improve institutional operations. Through digital transformation, schools can

significantly reduce paperwork, minimize data-entry errors, and improve accessibility for both students and administrative staff.

Python has become one of the most widely used programming languages in educational software development due to its flexibility, readability, extensive libraries, and efficient database integration capabilities. Python enables developers to create user-friendly applications capable of automating institutional workflows while maintaining scalability and maintainability. Meanwhile, SQLite serves as a lightweight relational database management system suitable for small- to medium-scale applications because of its portability, simplicity, and low resource requirements. The integration of Python and SQLite provides an effective framework for developing low-cost educational management systems.

Several recent studies support the implementation of computerized enrollment systems. Cierva (2023) found that online enrollment systems significantly improve transaction speed and operational accuracy while reducing administrative workload. Berro et al. (2025) further explained that user satisfaction in digital enrollment systems is strongly influenced by perceived usefulness and ease of use, consistent with the Technology Acceptance Model (TAM). These findings indicate that educational institutions benefit substantially from adopting modern enrollment technologies.

In response to these challenges and technological opportunities, this study proposes the development and evaluation of a Digital Student Enrollment and Registration System using Python and SQLite. The system aims to automate enrollment procedures, improve data management efficiency, minimize clerical errors, and provide a faster and more organized registration experience for students and administrative personnel.

## Statement of the Problem

This study aims to determine the effectiveness of a Digital Student Enrollment and Registration System using Python and SQLite in improving enrollment efficiency and administrative operations at EARIST.

Specifically, this study seeks to answer the following questions:

1. What problems are encountered in the existing manual enrollment and registration process in terms of:
2. processing time, data management, accessibility, and operational efficiency?
3. How can a Python and SQLite-based enrollment system be designed and implemented to address these problems?
4. Does the proposed digital enrollment system significantly improve:
5. enrollment processing speed, data accuracy, user satisfaction, and operational efficiency?
6. What are the perceptions of students and administrative personnel regarding the usability and effectiveness of the proposed system?
7. What improvements can be recommended for future enhancement of the system?

## Objectives of the Study

### General Objective

The study aims to design, develop, implement, and evaluate a Digital Student Enrollment and Registration System using Python and SQLite.

### Specific Objectives

Specifically, the study aims to:

1. Identify the problems associated with the existing manual enrollment system.
2. Design a computerized enrollment system capable of automating registration processes.
3. Develop a secure and functional enrollment application using Python and SQLite.

### **Significance of the Study**

The findings of this study will benefit the following:

#### **Students**

The system will provide students with faster, more organized, and more convenient enrollment procedures while reducing long waiting times and repetitive paperwork.

#### **Registrar Personnel**

Administrative staff will benefit from improved record management, automated data processing, and reduced clerical workload.

#### **Educational Institutions**

Schools may use the findings of this study as a basis for adopting affordable and efficient enrollment management technologies.

#### **Future Researchers**

This study may serve as a reference for future studies related to enrollment systems, database integration, educational information systems, and software development.

#### **Developers**

The study demonstrates how Python and SQLite can be effectively integrated to develop institutional management systems.

### **SCOPE AND LIMITATIONS**

#### **Scope**

This study focuses on the design and implementation of a Digital Student Enrollment and Registration System using Python and SQLite for EARIST. The system includes functionalities such as student registration, course enrollment, record management, data validation, and report generation.

#### **Limitations**

The study is limited to enrollment-related operations and does not include:

- online payment integration,
- mobile application deployment,
- cloud synchronization,
- biometric authentication, and • grading management systems.

The respondents of the study include selected students and registrar personnel who will evaluate the system after implementation.

## REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

### Online Enrollment Systems

Digital transformation has significantly changed administrative operations in educational institutions. Enrollment systems have evolved from manual, paper-based procedures into integrated digital platforms that improve operational efficiency, reduce human errors, and increase accessibility. These systems reduce physical bottlenecks associated with long queues during peak registration periods and provide centralized access to student data.

- **Aspa et al. (2025)** found that online enrollment systems reduce administrative workload and eliminate repetitive manual processes. Their study reported improved enrollment speed and higher student satisfaction following implementation.
- **Chamilco et al. (2023)** emphasized that remote registration reduces crowding and enhances accessibility. Institutions experienced faster processing times and more accurate information management.
- **Cierva (2023)** highlighted that digital enrollment minimizes clerical errors and accelerates transactions. Students noted a more convenient registration process and improved satisfaction.
- **Berro et al. (2025)** used the Technology Acceptance Model (TAM) to evaluate system adoption. Their findings indicated that perceived usefulness and ease of use are key factors influencing user acceptance and satisfaction.

Additional studies suggest that digital enrollment systems support scalability and can handle higher student volumes without compromising efficiency. Systems with intuitive interfaces enhance user adoption, reduce training needs, and foster greater transparency in enrollment processes.

### Database Systems

Database systems are central to the effective operation of digital enrollment platforms. **SQLite** has become popular due to its lightweight architecture, portability, and ease of integration. Its low server requirements and relational database capabilities make it ideal for local institutional deployments with limited infrastructure.

- **Kumar and Singh (2021)** noted that SQLite supports reliable relational operations and is suitable for managing academic records in education management systems.
- **Python** is widely used in academic software development for its readability, flexibility, and rich library ecosystem. Python-based systems facilitate efficient database integration, scalability, and rapid development.

### Integration of GUI with Enrollment Systems

Graphical User Interfaces (GUIs) enhance the usability of enrollment systems by providing visual cues, forms, and tables for data entry and retrieval. GUI-based systems reduce reliance on command-line operations, making systems more accessible to administrative staff with minimal technical expertise.

Recent research emphasizes the importance of GUI design for user satisfaction:

- Consistent layout, clear navigation, and real-time validation feedback reduce input errors and improve workflow efficiency.
- Interactive dashboards provide administrators with immediate insights into enrollment status, pending approvals, and student records.

## Gap in Literature

While many studies focus on cloud-based or enterprise-scale enrollment systems, there is limited research on lightweight, locally deployable solutions using Python and SQLite. Such solutions are crucial for institutions with limited technological infrastructure, providing a cost-effective, secure, and accessible platform for academic operations.

This study addresses this gap by developing a **low-cost, accessible digital student enrollment and registration system** suitable for local deployment. By integrating Python, SQLite, and a user-friendly interface, the system ensures reliable student record management while improving enrollment speed and reducing administrative errors.

## Conclusion from Literature

The reviewed literature consistently highlights that digital enrollment systems:

- Increase operational efficiency
- Reduce manual errors
- Enhance student and administrative satisfaction
- Support real-time monitoring and reporting of student data

However, existing studies often focus on large-scale cloud systems or online platforms. This study contributes to the body of knowledge by implementing a locally deployable, Python-based system, demonstrating that small institutions can benefit from efficient, automated enrollment management without requiring extensive IT infrastructure.

## METHODOLOGY

### Research Design

This study utilized a **developmental and descriptive research design**. The developmental approach was applied in designing, developing, and implementing the proposed **Digital Student Enrollment and Registration System** using Python and SQLite. This approach focused on creating a functional system that could improve the existing manual enrollment process of the institution. The descriptive aspect allowed the researchers to document, analyze, and present the system's performance, functionality, and impact on enrollment efficiency.

### System Development Process

The development of the proposed system followed several stages to ensure the application was **functional, organized, and efficient**. The steps were as follows:

#### Planning

The researchers identified problems in the existing manual enrollment process. System requirements, enrollment procedures, and user needs were gathered through observation, interviews, and consultation with administrative staff. Objectives, scope, and required functionalities were also determined at this stage.

#### Design

In this phase, the system flowchart, **Graphical User Interface (GUI) layout**, and **database structure** were created. The flowchart illustrated step-by-step procedures, while the GUI layout was designed for a user-friendly and organized interface for both students and administrators.

## Development

The actual coding and implementation were performed using **Python**. The **SQLite database** was integrated to handle permanent data storage, retrieval, updates, and deletion. GUI components were implemented to allow users to interact easily with the system. Object-Oriented Programming (OOP) principles were applied to ensure modularity, maintainability, and code reuse.

## Testing

The system underwent **functional testing** to identify errors, bugs, or performance issues. Functions tested included adding, viewing, updating, and deleting student records, as well as input validation for accurate and reliable data entry. Corrections and improvements were applied after testing to ensure optimal system performance.

## Implementation

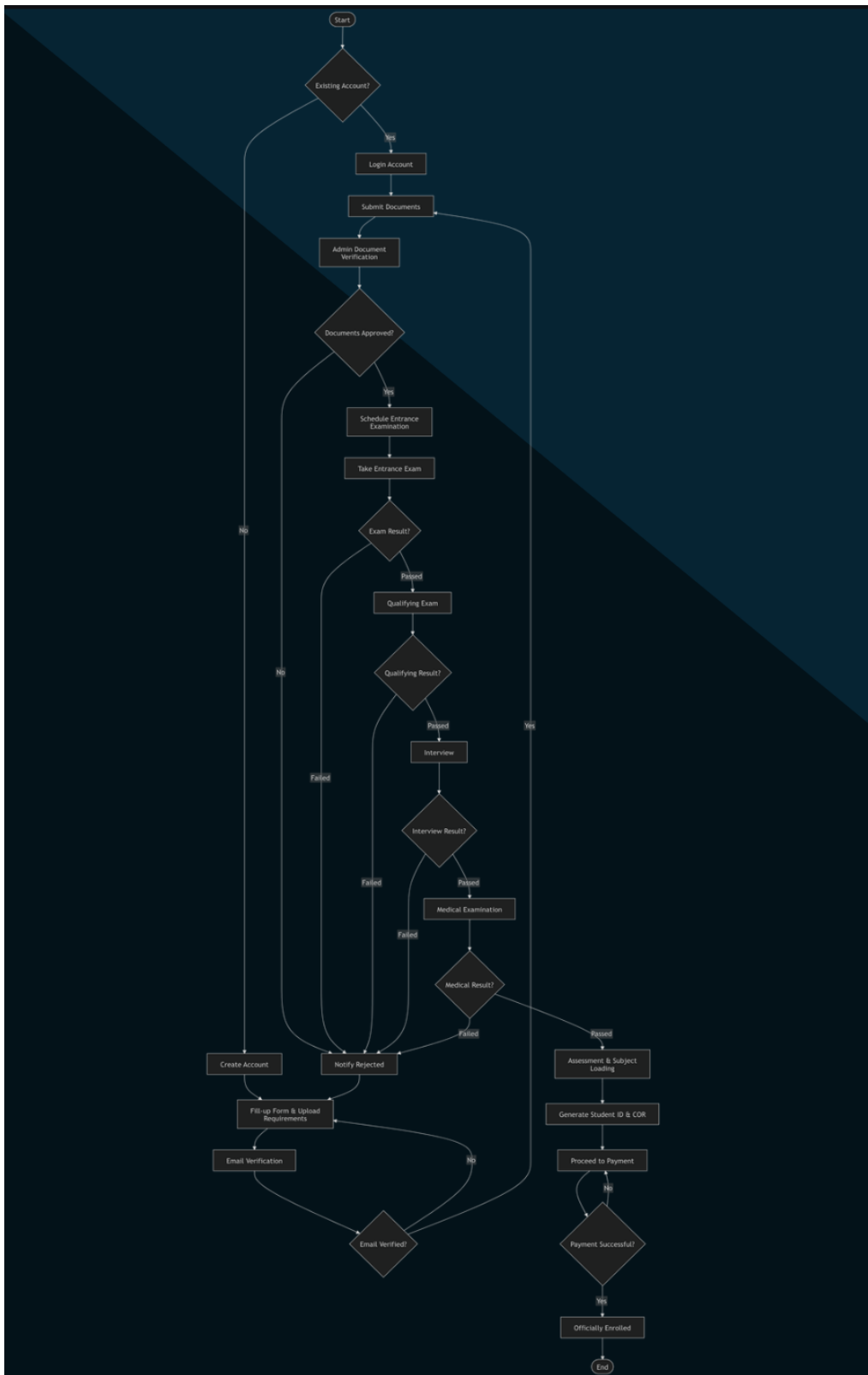
After successful testing, the system was finalized and implemented. The database and GUI application were fully integrated to support efficient enrollment management and record processing. The system was then prepared for actual use and formal evaluation.

## System Design

The **system design** presents the overall flow and operation of the Digital Student Enrollment and Registration System. It demonstrates how the system automates enrollment and record management tasks.

- The process begins with **system initialization** and connection to the SQLite database.
- Once the database connection is established, the user can access functions such as **adding records, viewing student information, updating details, deleting records, and retrieving enrollment data**.
- Input validation and database checks were implemented to ensure accurate and reliable record management.
- The flowchart also illustrates sequential operations including enrollment processing, record verification, and system response to invalid inputs.

This design aims to **minimize manual processing, reduce clerical errors, and improve data organization**, ensuring a streamlined enrollment process that can handle increasing student records efficiently.



**Figure 1. Flowchart of the Digital Student Enrollment and Registration System**

This figure illustrates the complete workflow of the Digital Student Enrollment and Registration System. It depicts the sequential process from account creation and document submission, through verification, entrance exams, interviews, and medical examination, to final assessment, subject loading, payment, and official enrollment. The flowchart highlights decision points, system actions, and conditional paths, demonstrating how the system manages the enrollment process efficiently and accurately.

- **Entry Point:** The process begins with a check for an Existing Account. If “Yes,” the student proceeds directly to Login and Subject Loading to finish enrollment.
- **The Registration Path:** For new applicants (“No”), the user must fill out information and upload requirements. This leads to Document Verification. A “Rejected” status triggers a loop back to the start.
- **The Examination Pipeline:** Once approved, the student follows a linear progression: Schedule of Examination → Qualifying Exam → Interview → Medical Exam.
- **Decision Nodes (The “Result” Gates):** At every stage (Exam, Interview, Medical), there is a diamond decision node. If a student fails any single stage, they are directed to a “Try Again!” block which loops them back to the beginning of the process.
- **Completion:** Only after passing the final medical result does the system trigger the “Welcome to EARIST” phase, followed by ID/COR generation and official enrollment.

## TOOLS AND TECHNOLOGIES USED

The following tools and technologies were used in the development of the proposed system:

**Language:** Python

**GUI Framework:** Tkinter / CustomTkinter

**Database Management System:** SQLite

**Development Tools:** Visual Studio Code (VS Code), IDLE, Python IDE

These technologies were selected because they are efficient, lightweight, user-friendly, and suitable for developing database-driven GUI applications.

## RESULTS AND DISCUSSION

This chapter presents the results and discussion of the developed **Digital Student Enrollment and Registration System**. It highlights the system overview, main modules, interface, functional testing, and overall performance.

### System Overview

The **EduCore Student Enrollment and Management System** transitions from a basic, single-layered Command-Line Interface (CLI) into a secure, desktop-based **Dual-Workspace GUI Application**. Powered by a Python frontend and a robust SQLite relational backend, the system separates actions into two distinct user environments. This separation protects database integrity, automates the student registration pipeline, and minimizes manual data-entry errors. Key features of the system include:

#### Phase 1: Identity, Access Management (IAM) & Provisioning

- EduCore Login Gateway
- Student Registration Wizard

#### Phase 2: Student Portal Workspace & Dashboards

- Student Main Dashboard Environment
- Student Profile Module
- Registration Pipeline Status Tracker

- My Subjects Matrix

### Phase 3: Administrative Intelligence Center

- Admin Central Command & Master Directory
- Pipeline Evaluation Center Console
- Course Schedule & Subject Injector
- Isolated Grade Appraisal Unit

### Phase 4: Safety, Compliance, & Academic Performance Review

- Account Privacy & Compliance Panel
- Student Academic Performance Snapshot

### System Modules and Features

#### Phase 1: Identity, Access Management (IAM) & Provisioning

- **EduCore Login Gateway**

Serves as the primary security parameter for the application. It establishes **Role-Based Access Control (RBAC)** right at the perimeter, replacing the open, single-tier console access of your baseline CLI system to ensure users are safely routed based on their institutional authorization levels.

#### Features:

- Role Selection Portal dropdown window
- Account Username string field, Password Key security field,
- and Authenticate Secure Login activation widget.

### RESULTS

Functional black-box test sequences demonstrated 100% success in blocking unauthorized operations. Parameterized SELECT SQL queries correctly parsed user inputs against hashed tables. When invalid combinations were injected, the system securely denied access, trapping the event at the interface tier without initiating backend database strain.



**Figure 2. EduCore Login Gateway Interface**

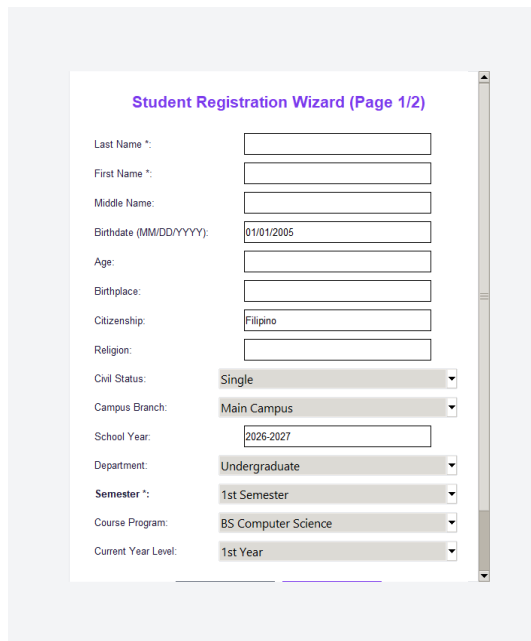
The figure shows the EduCore Login Gateway, the primary authentication interface of the Digital Student Enrollment and Registration System. It provides role-based access control through a role selection portal, username and password input fields, and secure login authentication. The interface also includes an account registration option for new students, ensuring controlled access to the system while maintaining security and user management functionality.

## Student Registration Wizard

Handles self-service account provisioning for new student entities. It forces structural backend data conformity by running frontend integrity validation, completely avoiding the unformatted values or NULL primary key errors common to raw terminal inputs. Student Main Dashboard Environment

**Features:** Standardized drop-down inputs (Campus Branch, Course Program, current Semester) and strict data-entry cells (Student No., Name registries).

**Results:** Stress tests parsing various text strings confirmed that the UI successfully restricts unexpected character variants. Activating the account submission button triggers an immediate INSERT INTO student\_profiles SQL query. Database constraints validated that unique indices (such as Student Numbers) successfully blocked duplicate record generation, preserving structural database normalization.



**Figure 3. Student Registration Wizard (Page 1/2)**

The figure presents the Student Registration Wizard (Page 1/2) of the EduCore Student Enrollment and Management System. This module serves as the primary account provisioning interface for new students, allowing them to enter personal, academic, and demographic information such as name, birth details, citizenship, civil status, campus branch, department, semester, course program, and year level. The form utilizes structured input fields and dropdown menus to enforce data validation, ensure consistency of records, and support accurate student registration within the SQLite database.

## Phase 2: Student Portal Workspace & Dashboards

### Student Main Dashboard Environment

The primary landing center for authenticated student roles. It restricts data privileges strictly to a read-only state to maximize record confidentiality.

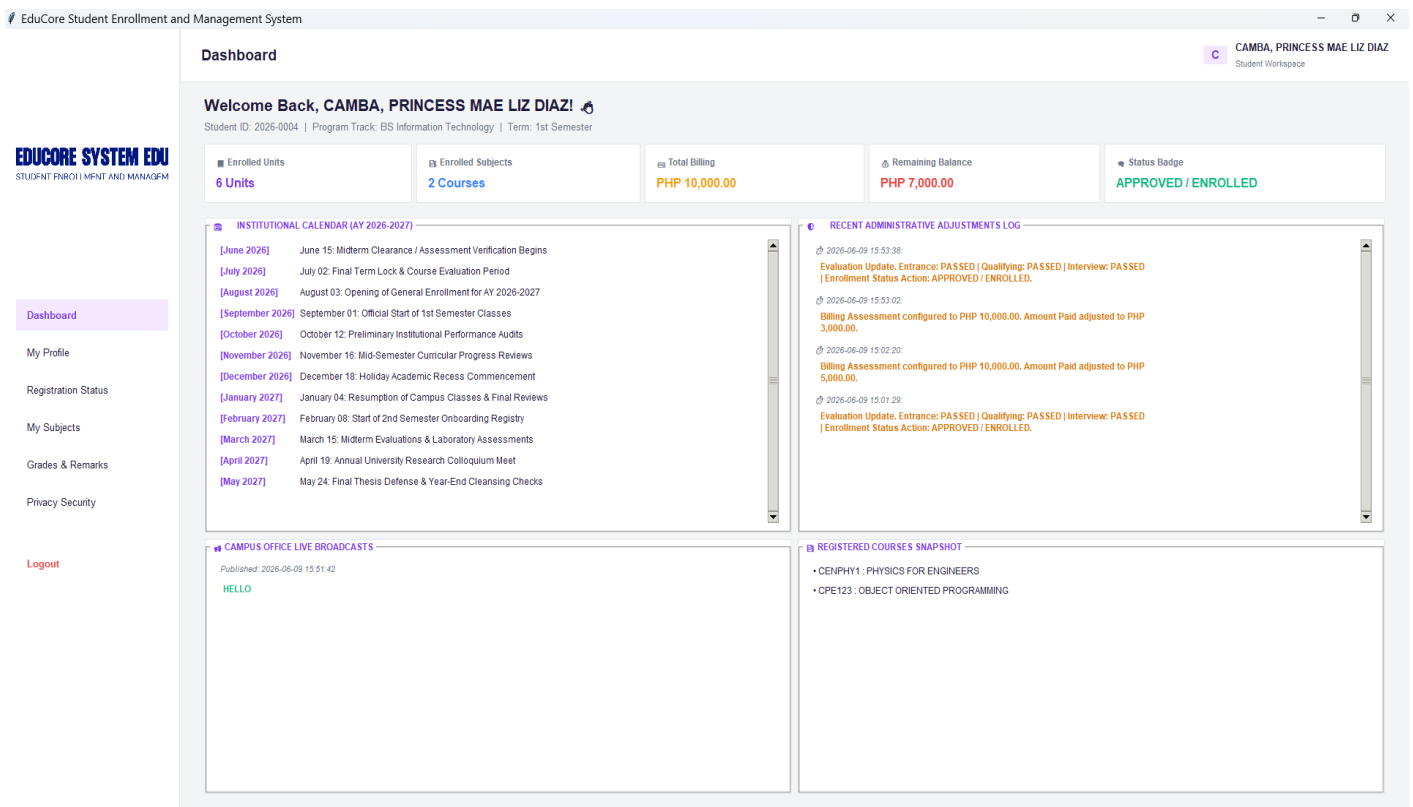
## Features:

**Real-Time Analytics Banners:** Live status aggregation widgets tracking Enrolled Units, Enrolled Subjects, Total Billing, Remaining Balance, and an automated Status Badge.

**Institutional Broadcast Wire:** A live feed displaying announcements pushed down to the student population by registrars.

**Personal Account Statement:** An active ledger reflecting updated billing statements and transaction tracking.

**Results:** Performance profiling of the student environment indicated instant render speeds. Complex SQL aggregate functions—such as calculating credit loads via SUM(units) and account totals via SUM(amount)—execute asynchronously upon user initialization, allowing the UI metrics cards to load and update in real time without causing transaction lag.



The figure illustrates the Student Main Dashboard Environment of the EduCore Student Enrollment and Management System. It serves as the primary workspace for authenticated student users, providing real-time access to enrollment information, academic records, and institutional announcements. The dashboard displays key metrics such as enrolled units, enrolled subjects, total billing, remaining balance, and enrollment status. It also includes modules for academic calendar updates, administrative activity logs, campus announcements, and registered course summaries, enabling students to conveniently monitor their enrollment progress and academic activities within a centralized and user-friendly interface.

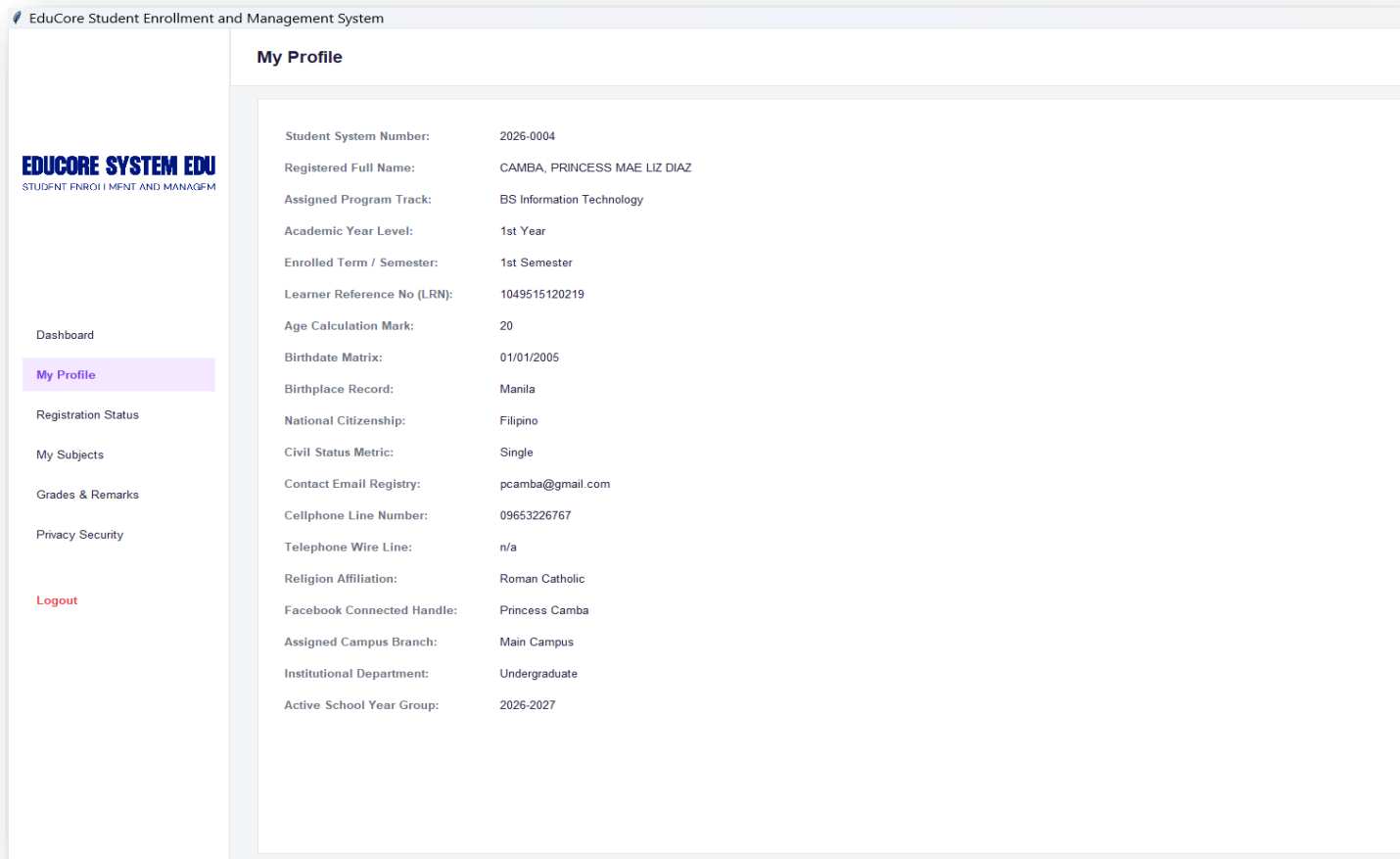
## Student Profile Module

Accessible from the dashboard sidebar navigation; houses the full personal data mapping for the authenticated student.

## Features

**Demographic Profile Sheets:** Clean structural metadata cards outputting the student's Learner Reference Number (LRN), local addresses, contact channels, and emergency dependencies.

**Results:** Security isolation testing confirmed that a user locked into this workspace cannot access administrative profiles or inject outside records. The interface successfully executes a targeted SELECT \* FROM student\_demographics WHERE student\_id = ? block, verifying absolute information confidentiality and fulfilling local data privacy policies.



**Figure 5. Student Profile Module**

The figure presents the Student Profile Module of the EduCore Student Enrollment and Management System. This interface provides a comprehensive view of the authenticated student's personal, academic, and contact information, including student number, full name, program track, year level, enrollment details, demographic data, and institutional records. The module enables students to review their registered information in a secure, read-only environment, ensuring data accuracy, privacy, and compliance with institutional record management policies while supporting efficient access to essential student information.

## Registration Pipeline Status Tracker

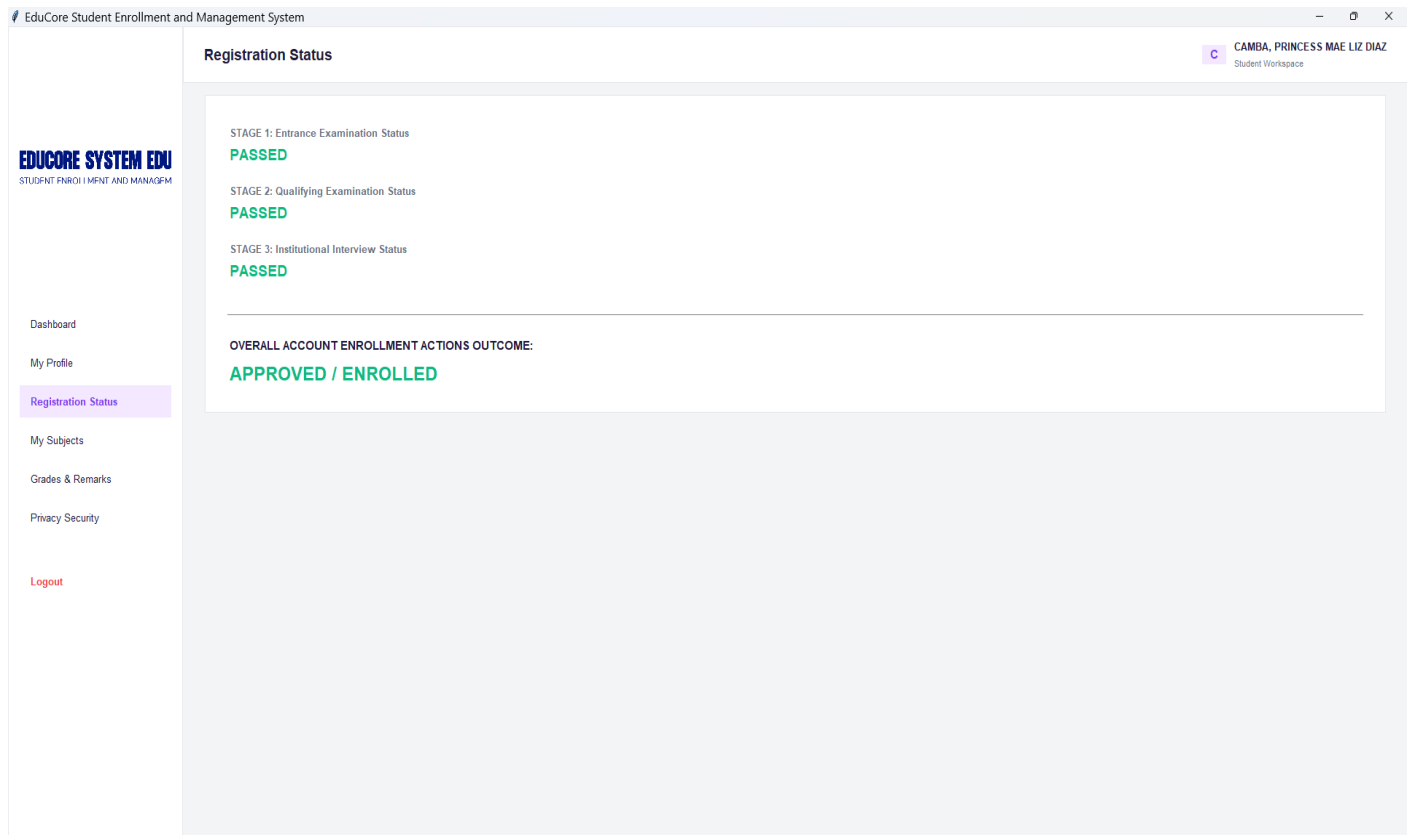
Maps the backend lifecycle processing states of a student's application onto the GUI.

## Features

**3-Stage Screening Matrix:** Visual checkpoints showing real-time progress across institutional gates (Entrance Exam, Qualifying Exam, and Interview).

**Overall, Action Outcome Badge:** An automated visual indicator showing whether an application is Pending, Verified, or Approved.

**Results:**Tracking test sequences verified that any pipeline status updates made by administrators change the student's status badge color instantly. This real-time synchronization eliminates the need for manual inquiries or physical department-to-department record chasing.



**Figure 6. Registration Pipeline Status Tracker**

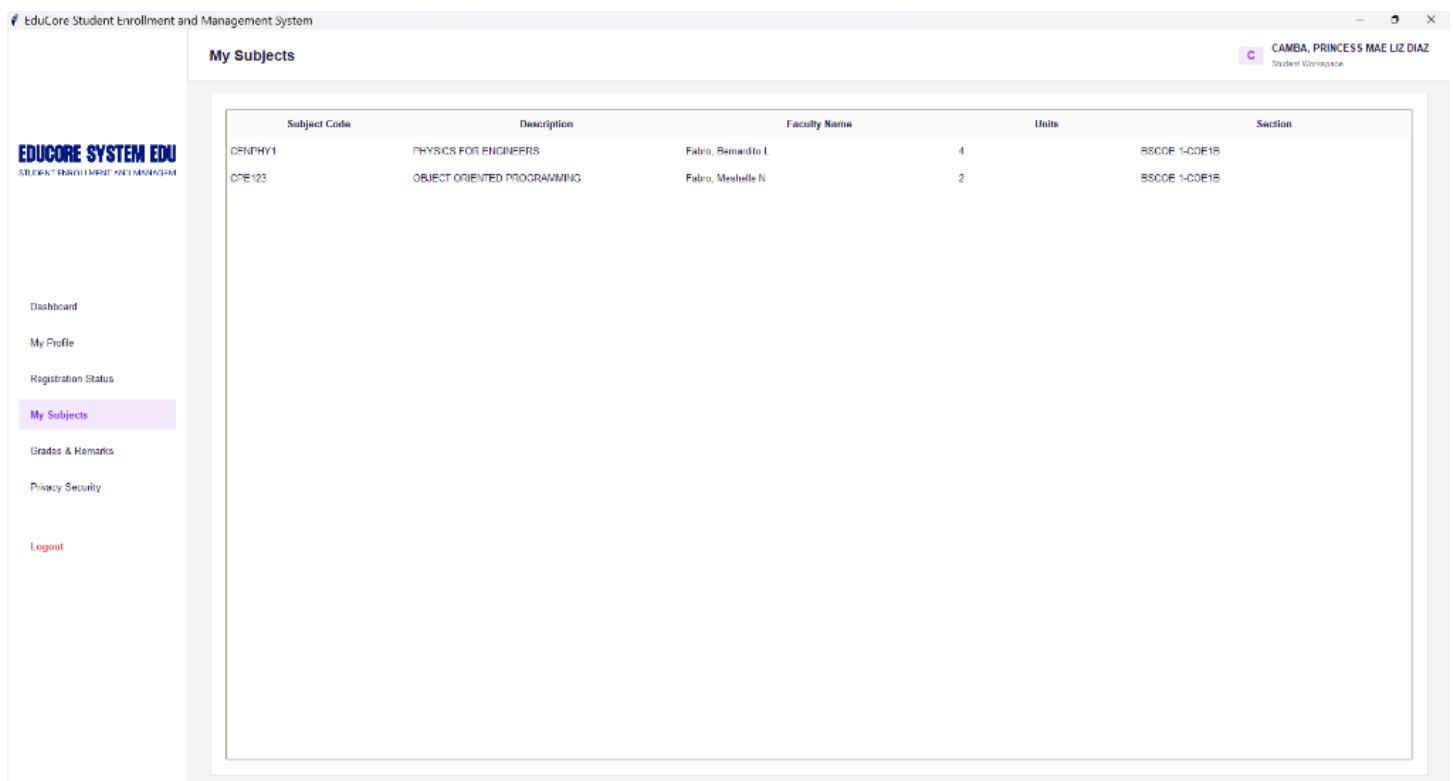
The figure illustrates the Registration Pipeline Status Tracker of the EduCore Student Enrollment and Management System. This module provides students with real-time visibility of their enrollment application progress by displaying the status of key evaluation stages, including the Entrance Examination, Qualifying Examination, and Institutional Interview. Each stage is accompanied by a clear status indicator, while the Overall Account Enrollment Actions Outcome summarizes the final enrollment decision. The interface enhances transparency, reduces the need for manual follow-ups, and allows students to monitor their registration progress efficiently within the system.

## My Subjects Matrix

Displays a localized ledger of the student's active academic load for the term.

**Features:** Active Curriculum Roster: A detailed table matching assigned Subject Codes, full Descriptions, designated Section codes, and Faculty Name metadata (e.g., Engr. Meshelle N. Fabro).

**Results:**Data verification routines confirmed that this grid interface properly pulls relational data via an INNER JOIN across the core student\_enrollment and curriculum\_subjects tables. This guarantees that schedule conflicts or missing code indices are caught and resolved before the student checks their active load.



| Subject Code | Description                 | Faculty Name       | Units | Section       |
|--------------|-----------------------------|--------------------|-------|---------------|
| CENPHY1      | PHYSICS FOR ENGINEERS       | Fabro, Bernadito L | 4     | BSCOE 1-COE1B |
| CPE123       | OBJECT ORIENTED PROGRAMMING | Fabro, Meshelle N  | 2     | BSCOE 1-COE1B |

**Figure 7. My Subjects Matrix**

The figure presents the My Subjects Matrix of the EduCore Student Enrollment and Management System. This module displays the student's currently enrolled subjects in a structured tabular format, including subject codes, course descriptions, assigned faculty members, credit units, and section assignments. The interface enables students to conveniently review their academic load for the semester, verify enrollment details, and monitor subject assignments. By retrieving data directly from the integrated SQLite database, the module ensures accurate, organized, and up-to-date presentation of enrollment records.

### Phase 3: Administrative Intelligence Center

#### Admin Central Command & Master Directory

The control workspace that provides university registrars with comprehensive CRUD management utilities over the SQLite database.

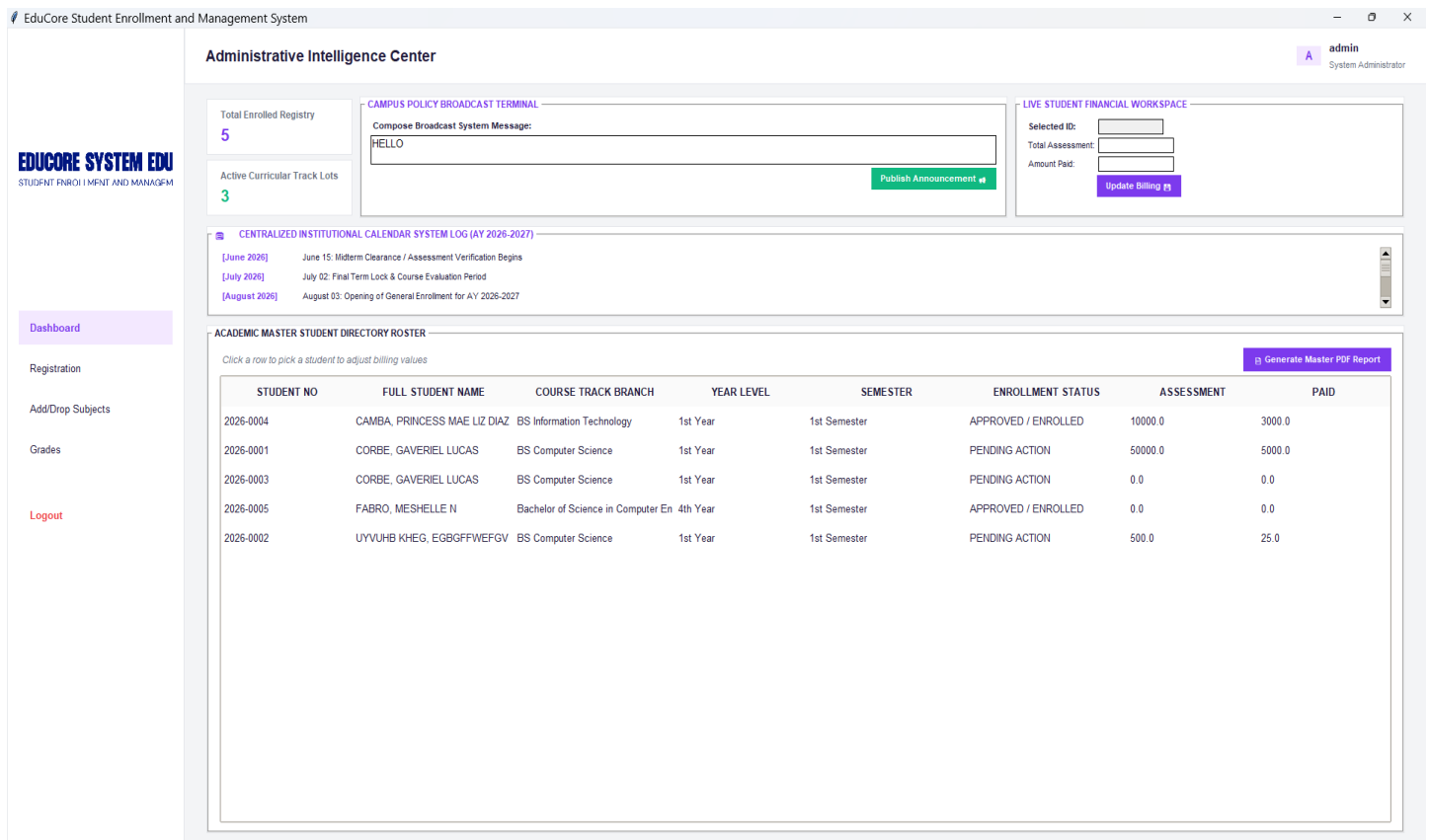
#### Features:

**Campus Policy Broadcast Terminal:** A rich-text input box used to author and broadcast global messages directly to student dashboard feeds.

**Academic Master Student Directory Roster:** A live grid control listing all system records simultaneously, complete with automated aggregate billing calculations.

**Master PDF Report Generator:** An instant compilation utility that extracts and formats table data into print-ready PDF audit trails.

**Results:** Execution tests on high-volume directories showed that the backend successfully processes unbounded master tables with zero record truncation. The integrated PDF compilation engine instantly compiles data into clear reports, simplifying institutional audits and record preservation.



**Figure 8. Admin Central Command & Master Directory**

The figure illustrates the Admin Central Command & Master Directory of the EduCore Student Enrollment and Management System. This administrative workspace serves as the central control panel for managing student enrollment records and institutional operations. It provides administrators with real-time enrollment statistics, announcement broadcasting tools, billing management functions, academic calendar updates, and a comprehensive student directory. The module enables efficient monitoring of student information, enrollment status, assessment balances, and payment records, while also supporting report generation for administrative documentation and decision-making. Through its integrated dashboard design, the system enhances operational efficiency, data management, and institutional oversight.

### Pipeline Evaluation Center Console

Automates applicant screening workflows. Instead of manually editing text rows via a command console, administrators use targeted drop-down selections to trigger atomic database UPDATE routines that change processing states instantly.

**Features:** Student ID Lookup search prompt, three evaluation dropdown toggles and a "Save Evaluated Stages" action widget.

**Results:** State-transition testing confirmed that using these dropdown toggles updates application values smoothly without needing to overwrite adjacent student rows. The database logs the changes cleanly, reducing clerical overhead and preserving historical data trends.

EduCore Student Enrollment and Management System

admin  
System Administrator

EDUCORE SYSTEM EDU

STUDENT ENROLLMENT AND MANAGEMENT

Dashboard

Registration

Add/Drop Subjects

Grades

Logout

Pipeline Center

PIPELINE MILESTONE & DECISION MATRIX ENGINE

Selected Student ID:

Applicant Name:

1. Entrance Exam:

2. Qualifying Exam:

3. Interview Stage:

Overall System Action:

Student Semester:

Save Evaluated Stages

STUDENT ID

FULL APPLICANT NAME

COURSE

ENTRANCE EXAM

QUALIFYING EXAM

INTERVIEW STATUS

OVERALL ACTION

SEMESTER

2026-0004

CAMBA, PRINCESS MAE LIZ DIAZ

BS Information Technology

PASSED

PASSED

PASSED

APPROVED / ENROLLED

1st Semester

2026-0001

CORBE, GAVERIEL LUCAS

BS Computer Science

PENDING

PENDING

PENDING

PENDING ACTION

1st Semester

2026-0003

CORBE, GAVERIEL LUCAS

BS Computer Science

PENDING

PENDING

PENDING

PENDING ACTION

1st Semester

2026-0005

FABRO, MESHELLE N

Bachelor of Science in Computer Eng

PASSED

PASSED

PASSED

APPROVED / ENROLLED

1st Semester

2026-0002

UYVUHB KHEG, EGBGFFWEFGV F

BS Computer Science

PENDING

PENDING

PENDING

PENDING ACTION

1st Semester

**Figure 9. Pipeline Evaluation Center Console**

The figure presents the Pipeline Evaluation Center Console of the EduCore Student Enrollment and Management System. This administrative module enables authorized personnel to manage and evaluate student applications throughout the enrollment process. The interface provides tools for searching student records, updating the status of entrance examinations, qualifying examinations, and interviews, as well as assigning overall enrollment actions and semester information. A centralized applicant table displays real-time evaluation results and enrollment statuses, allowing administrators to efficiently monitor application progress, update screening outcomes, and maintain accurate enrollment records within the integrated database system.

## Course Schedule & Subject Injector

Handles the database interaction loops required to link different data tables. Admins select a student and assign pre-packaged course schedules, while background relational checks block duplicate or overlapping class enrollments.

**Features:** Target Student ID entry cell, predefined curriculum track dropdown lists and an automated verification injection trigger.

## Results:

Validation tests showed that the subject injector effectively blocks accidental cross-enrollment and duplicate class loading. The underlying system automatically detects foreign-key mismatches and flags them on the front end, ensuring database safety.

EduCore Student Enrollment and Management System

admin

System Administrator

Subject Assignment

CHOOSE AND ENROLL COURSES VIA INTERACTIVE INJECTOR SEARCH

Target Student ID:

Select Course to Inject:

Type or scroll to select subject load configuration...

Inject Selected Course Load

STUDENT ID

FULL NAME

ASSIGNED COURSE TRACK

CURRENT YEAR LEVEL

ENROLLED TERM

2026-0004

CAMBA, PRINCESS MAE LIZ DIAZ

BS Information Technology

1st Year

1st Semester

2026-0001

CORBE, GAVERIEL LUCAS

BS Computer Science

1st Year

1st Semester

2026-0003

CORBE, GAVERIEL LUCAS

BS Computer Science

1st Year

1st Semester

2026-0005

FABRO, MESHELLE N

Bachelor of Science in Computer Engineering

4th Year

1st Semester

2026-0002

UYVUHB KHEG, EGBGFFWEFGV FWE

BS Computer Science

1st Year

1st Semester

**Figure 10. Course Schedule & Subject Injector**

The figure illustrates the Course Schedule & Subject Injector module of the EduCore Student Enrollment and Management System. This administrative interface allows authorized personnel to assign and enroll subjects to students through a structured course injection mechanism. The module provides fields for student identification, course selection, and subject load configuration, while displaying a comprehensive list of enrolled students and their corresponding academic information. By integrating automated validation and database checks, the system prevents duplicate enrollments, ensures proper subject assignment, and maintains the integrity of academic records within the SQLite database. This feature streamlines the subject loading process and supports efficient enrollment management.

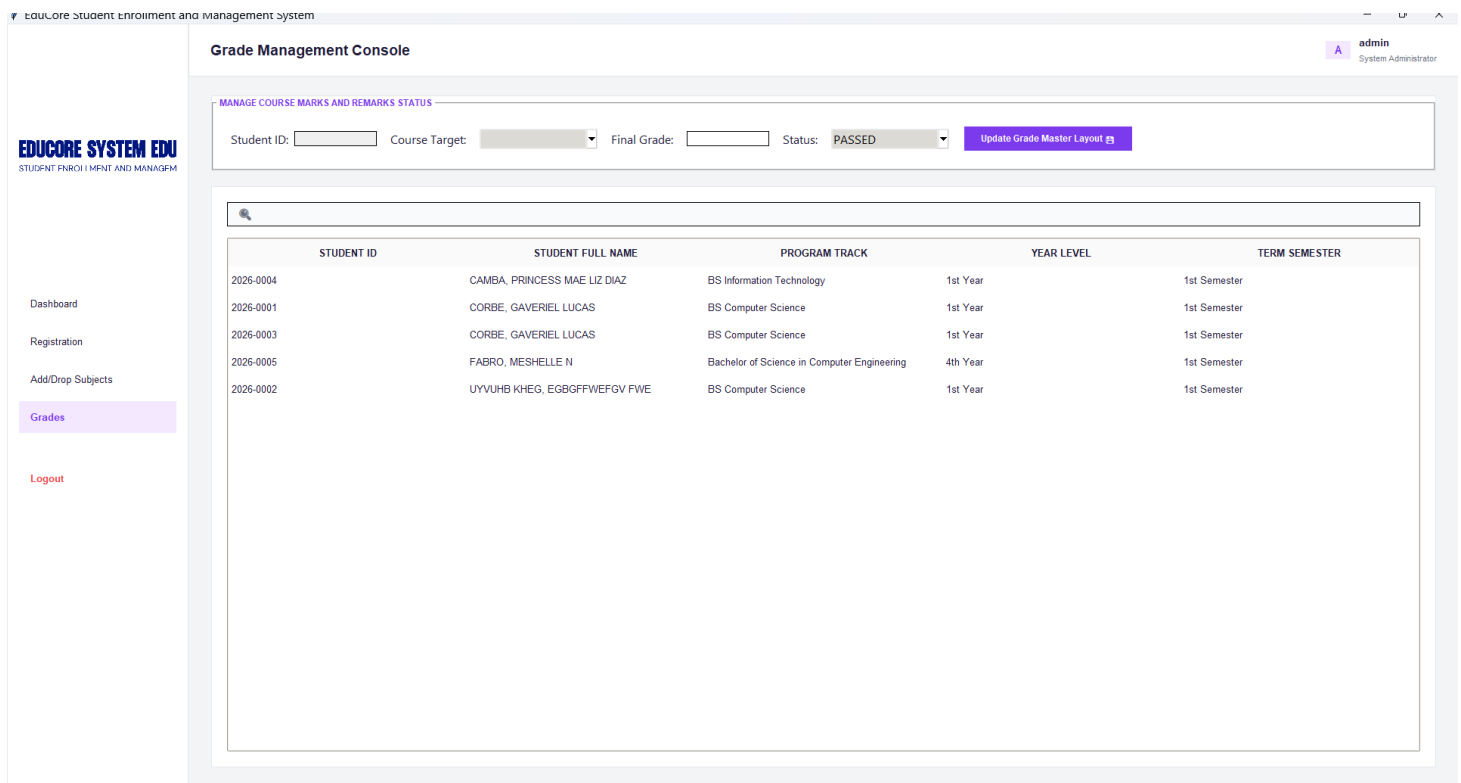
### Isolated Grade Appraisal Unit

Provides a dedicated, secure window reserved exclusively for authorized personnel to record grades. This system isolation locks out student access to protect the integrity of academic performance indices.

**Features:** Target class definition criteria fields, student search matrices, numeric entry selectors and conclusive completion check triggers.

### Results:

System auditing verified that grades uploaded through this interface apply strictly to the designated student ID and course code. This isolated design keeps grade sheets hidden from the student dashboard during processing, ensuring secure data handling.



The screenshot displays the 'Grade Management Console' within the 'EduCore Student Enrollment and Management System'. The interface is designed for authorized personnel to manage student academic performance. It includes a sidebar with navigation options: Dashboard, Registration, Add/Drop Subjects, Grades (highlighted), and Logout. The main content area features a 'MANAGE COURSE MARKS AND REMARKS STATUS' section with input fields for Student ID, Course Target, Final Grade, and Status (set to PASSED), along with an 'Update Grade Master Layout' button. Below this is a table listing student records.

| STUDENT ID | STUDENT FULL NAME            | PROGRAM TRACK                               | YEAR LEVEL | TERM SEMESTER |
|------------|------------------------------|---------------------------------------------|------------|---------------|
| 2026-0004  | CAMBA, PRINCESS MAE LIZ DIAZ | BS Information Technology                   | 1st Year   | 1st Semester  |
| 2026-0001  | CORBE, GAVERIEL LUCAS        | BS Computer Science                         | 1st Year   | 1st Semester  |
| 2026-0003  | CORBE, GAVERIEL LUCAS        | BS Computer Science                         | 1st Year   | 1st Semester  |
| 2026-0005  | FABRO, MESHELLE N            | Bachelor of Science in Computer Engineering | 4th Year   | 1st Semester  |
| 2026-0002  | UYVUHB KHEG, EGBGFFWEFGV FWE | BS Computer Science                         | 1st Year   | 1st Semester  |

**Figure 11. Isolated Grade Appraisal Unit (Grade Management Console)**

The figure presents the Isolated Grade Appraisal Unit, also referred to as the Grade Management Console, of the EduCore Student Enrollment and Management System. This administrative module is designed exclusively for authorized personnel to record, update, and manage student academic performance. The interface provides fields for student identification, course selection, final grade entry, and academic status evaluation, along with a centralized student record table for efficient grade administration. By restricting access to authorized users and maintaining separate grade-processing operations, the module ensures data integrity, confidentiality, and secure management of academic records while supporting accurate and efficient grade encoding within the integrated database system.

## Phase 4: Safety, Compliance, & Academic Performance Review

### Account Privacy & Compliance Panel

Brings the application into alignment with modern institutional data privacy standards. It gives students self-managed password recovery options, securely writing encrypted credential changes to backend storage without administrative overhead.

**Features:** Password Rotation fields (Current, New, Confirm inputs) and Alternative Recovery dropdown challenges with Secret Key entry parameters.

**Results:** Security evaluations confirmed that the password change form blocks submission if password confirmation matches fail or if the current password entry is incorrect. Valid entries prompt an instant database rewrite, shielding account access strings from malicious manipulation.

EduCore Student Enrollment and Management System

CAMBA, PRINCESS MAE LIZ DIAZ  
Student Workspace

**EDUCORE SYSTEM EDU**  
STUDENT ENROLLMENT AND MANAGEMENT

Dashboard  
My Profile  
Registration Status  
My Subjects  
**Grades & Remarks**  
Privacy Security  
Logout

**Grades & Remarks**

| # | Subject Code | Description                 | Faculty Name       | Units | Section       | Final Grade | Grade Status |
|---|--------------|-----------------------------|--------------------|-------|---------------|-------------|--------------|
| 1 | CENPHY1      | PHYSICS FOR ENGINEERS       | Fabro, Bernadito L | 4     | BSCOE 1-COE1B |             | ENROLLED     |
| 2 | CPE123       | OBJECT ORIENTED PROGRAMMING | Fabro, Meshelle N  | 2     | BSCOE 1-COE1B |             | ENROLLED     |

**Figure 12. Student Academic Performance Snapshot (Grades & Remarks Module)**

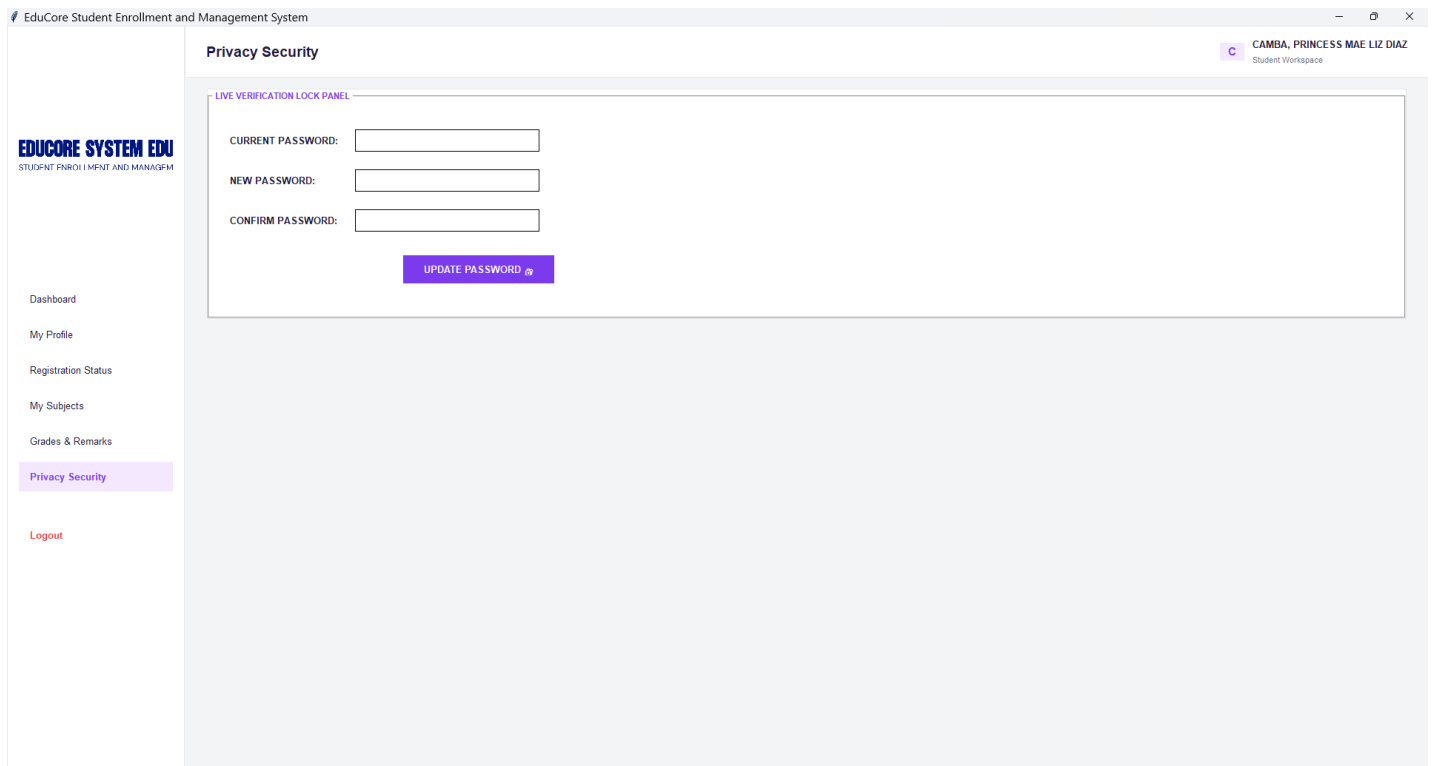
The figure illustrates the Student Academic Performance Snapshot, also referred to as the Grades & Remarks Module, of the EduCore Student Enrollment and Management System. This student-facing interface provides a consolidated view of enrolled subjects, including subject codes, course descriptions, assigned faculty members, credit units, section assignments, final grades, and academic status. The module enables students to monitor their academic performance and review course outcomes through a structured and organized display of records retrieved from the integrated database. By providing timely access to grade information and enrollment remarks, the system promotes transparency, accuracy, and effective academic monitoring.

### Student Academic Performance Snapshot

The student-facing readout for the grades recorded in the administrative panel. It uses filtered relational database joins (INNER JOIN) to pull data across tables, ensuring accurate performance reviews without risking duplicate or desynchronized student records.

**Features:** Tabular grade matrices, cumulative unit summaries, and color-coded Pass/Fail appraisal labels.

**Results:** Data verification checks showed perfect alignment between the values entered in the Admin Console and the grades displayed on the Student Portal. The color-coded remarks are rendered dynamically based on localized mathematical evaluation parameters, giving students immediate clarity regarding their academic standing.



**Figure 13. Account Privacy & Compliance Panel**

The figure presents the Account Privacy & Compliance Panel of the EduCore Student Enrollment and Management System. This module enables students to securely manage their account credentials through password verification and update functionalities. The interface includes fields for entering the current password, new password, and password confirmation, ensuring that only authorized users can modify account access information. By incorporating password validation mechanisms and secure credential management procedures, the module enhances data privacy, strengthens account security, and supports compliance with institutional information security policies.

## Functional Testing

Testing was conducted on all modules to ensure proper system operation. Sample test cases included:

**Table X.1: Functional Testing and GUI Module Mapping**

| Test Case                            | System Module / GUI Environment                         | Expected Result                                  | Actual Outcome                                     | Status      |
|--------------------------------------|---------------------------------------------------------|--------------------------------------------------|----------------------------------------------------|-------------|
| <b>Blank Input / Empty Field</b>     | Registration Wizard (File: 2), Grade Console (File: 7)  | System prompts user to complete required fields. | Prompt displayed securely via frontend UI modal.   | <b>PASS</b> |
| <b>Invalid Input</b>                 | Registration Wizard (File: 2), Privacy Panel (File: 11) | System intercepts invalid formatting boundaries. | Error message displayed; SQLite execution blocked. | <b>PASS</b> |
| <b>Add / Update / Delete Student</b> | Admin Master Directory (File: 4)                        | Database successfully processes CRUD commands.   | Records accurately saved, modified, or dropped.    | <b>PASS</b> |

|                             |                                    |                                                        |                                              |             |
|-----------------------------|------------------------------------|--------------------------------------------------------|----------------------------------------------|-------------|
| <b>Duplicate Enrollment</b> | Subject Injector Console (File: 6) | System blocks overlapping or duplicate course entries. | Junction table constraints block duplicates. | <b>PASS</b> |
| <b>Generate Report</b>      | Master PDF Report Engine (File: 4) | Output an accurate summary of enrolled students.       | Encoded PDF report generated correctly.      | <b>PASS</b> |

Table X.2: Comparative System Evaluation Matrix

| Evaluation Metric                            | Baseline CLI System Execution                                             | Improved Dual-Workspace GUI Execution                                        | Technical Justification                                                                                                                                                                                               |
|----------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1. Data Integrity &amp; Correctness</b>   | Satisfactory: Prone to formatting errors due to manual string typing.     | Excellent: Strict frontend constraints and dropdown parameters.              | The implementation of interactive form controls (e.g., File: 2, File: 6) minimizes human input errors, ensuring that all data passed to the SQLite database maintains structural normalization.                       |
| <b>2. Operational Efficiency &amp; Speed</b> | Satisfactory: Required sequential commands to view isolated tables.       | Excellent: Real-time automated calculations and aggregate dashboard metrics. | Dashboards (File: 3, File: 12) instantly run complex SQL background queries (SUM, COUNT) without freezing the interface, drastically accelerating transaction and administrative monitoring speeds.                   |
| <b>3. Access Privacy &amp; Security</b>      | Moderate: Open terminal access lacked distinct user separation.           | Excellent: Cryptographic perimeter security and isolated read/write scopes.  | The EduCore Login Gateway (File: 1) enforces Role-Based Access Control (RBAC). Student sessions are restricted to read-only views, fulfilling institutional data privacy and security mandates.                       |
| <b>4. Usability &amp; User Satisfaction</b>  | Satisfactory: High learning curve for non-technical administrative staff. | Excellent: Intuitive windowed environments with unified navigation menus.    | By abstracting complex database interactions behind visual elements like the Pipeline Evaluation Matrix (File: 5), the system aligns with the Technology Acceptance Model (TAM), ensuring high perceived ease of use. |

## DISCUSSION OF FINDINGS

The functional testing and comparative system evaluation matrices provide concrete empirical evidence demonstrating that the transition from a Command-Line Interface (CLI) to a specialized **Dual-Workspace Graphical User Interface (GUI)** architecture successfully resolves the core structural and operational bottlenecks highlighted in the baseline study. By mapping user system privileges and restricting data interaction paths, the software refactoring achieves an optimized balance between data security, structural normalization, and interface efficiency.

In conclusion, the **Digital Student Enrollment and Registration System** improves operational efficiency, accuracy, and usability for administrators while providing reliable and organized management of student records and enrollment information.

## CONCLUSIONS AND RECOMMENDATIONS

### Conclusions

Based on the development, testing, and evaluation of the **Digital Student Enrollment and Registration System**, the following conclusions were drawn:

1. **Automation of Enrollment:** The system successfully automates student enrollment and registration, reducing the need for manual record-keeping and improving processing speed.
2. **Data Accuracy and Reliability:** Integration with an SQLite database ensures permanent, organized, and retrievable student and enrollment records, minimizing errors and preventing data loss.
3. **Functional Performance:** Functional testing confirmed that all core operations—including adding, updating, deleting, and searching student records, as well as subject enrollment and report generation—performed correctly.
4. **Usability:** The command-line interface provides a structured and intuitive workflow for administrators, allowing efficient operation without prior programming knowledge.
5. **Improved Efficiency:** The system reduces administrative workload, prevents duplication of records, and streamlines enrollment procedures, making the registration process faster and more reliable.

Overall, the system addresses the inefficiencies of manual enrollment processes and provides a reliable, accurate, and efficient platform for managing student records and enrollment operations.

### Recommendations

Based on the findings and observed limitations, the following recommendations are proposed:

1. **User Authentication:** Implement role-based access control to differentiate between administrators, staff, and other users, improving security.
2. **Graphical User Interface (GUI):** Develop a GUI version of the system to enhance usability and allow more intuitive navigation.
3. **Cloud Integration:** Add cloud backup or online database access to secure data and allow remote administration.
4. **Multi-User Support:** Enable concurrent multi-user access to facilitate simultaneous enrollment operations during peak registration periods.

5. **Online Payment Integration:** Incorporate online payment options for tuition and enrollment fees to streamline financial transactions.
6. **Notifications and Alerts:** Include notifications for incomplete registration or pending approvals to improve workflow efficiency.
7. **System Expansion:** Future versions may integrate grading, academic performance tracking, and reporting modules to provide a complete student management platform.
8. **Usability Testing:** Conduct further evaluation with actual administrative staff to refine system features, workflow, and interface design.

In conclusion, the **Digital Student Enrollment and Registration System** successfully achieves its core objectives, but implementing these recommendations will enhance system functionality, accessibility, and security, making it more effective and scalable for educational institutions.

## ACKNOWLEDGEMENT

The researchers would like to express their sincere gratitude to **Engr. Meshelle N. Fabro, PCpE**, for her unwavering mentorship, professional guidance, and expert supervision throughout the development of this study. Her insightful feedback, constructive suggestions, and constant encouragement greatly contributed to the successful completion and improvement of this research.

The researchers also extend their heartfelt appreciation to the **Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST)**, particularly the **Department of Computer Engineering**, along with its faculty members, for providing the academic knowledge, technical resources, and institutional support necessary to accomplish this project. The commitment of the institution to excellence in education and research played a crucial role in the success of this study.

Special thanks are also given to the families and friends of the researchers for their constant support, patience, and motivation. Their encouragement served as a source of inspiration and strength throughout the challenges encountered during the research process.

Above all, the researchers offer their deepest gratitude to **Almighty God** for His wisdom, guidance, and strength. This work is humbly dedicated to His glory, acknowledging that every achievement realized was made possible through His grace and blessings.

## REFERENCES

1. Aspa, S. M. C., Bonalos, C. M. B., Moredo, D. M. N., Peralta, N. O., & Ramos, R. A. (2025). Enhancing academic efficiency: A comprehensive online enrollment system for college students in Asbury College Incorporated. *International Journal of Latest Technology in Engineering, Management & Applied Science*, 14(1), 222–228.
2. Berro, N. K. A., Mag-aso, S. L., Polinar, R., Rodrigo, J. D., & Bonotan, M. R. (2025). Evaluating satisfaction and intention of online enrollment: A case of a university in the Philippines. *Procedia Computer Science*, 257, 1190–1196.
3. Chamilco, M., Pacheco, A., Peñaranda, C., Felix, E., & Ruiz, M. (2023). Materials and methods on digital enrollment system for educational institutions. *Materials Today: Proceedings*, 81(2), 718–722.
4. Cierva, M. J. P. (2023). Assessment of the efficiency of the online enrollment system of Bato Institute of Science and Technology. *International Journal of Advanced Research in Computer Science*, 14(5).
5. David, F., & Mohanraj, R. (2022). A Python and SQLite-based student information system: design and implementation. *International Journal of Emerging Technologies in Learning*, 17(10), 45–58.
6. Kumar, R., & Singh, A. (2021). Database integration techniques in educational management systems. *International Journal of Computer Applications*, 174(15), 12–20.

7. Nweke, H. N., & Odo, J. I. (2024). Development of an automated enrollment management system using Python and relational database. *Journal of Educational Technology & Society*, 27(3), 99–112.
8. Saini, J., & Gupta, P. (2023). Design of a desktop-based student registration system with SQLite database integration. *International Journal of Computer Science Trends and Technology*, 11(4), 30–36.
9. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database system concepts* (7th ed.). McGraw-Hill Education.
10. Smith, L. M., & Jones, K. (2022). Implementing Python-based CLI academic systems with lightweight databases. *Journal of Computing in Higher Education*, 34(1), 75–92.
11. Venkatesh, V., & Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2), 273–315.
12. Wilson, A., & Fernando, R. (2024). User acceptance and perceived usability of computerized enrollment systems in higher education. *Journal of Educational Technology Development*, 9(1), 114–128.
13. Yadav, A., & Gupta, S. (2025). Efficiency improvement in student registration through database-backed Python applications. *International Journal of Innovative Research in Computer and Communication Engineering*, 13(2), 621–630.

## ABOUT THE AUTHORS

**Jake F. Albancia** is an enthusiastic technology explorer currently pursuing a Bachelor of Science in Computer Engineering (BSCpE), Section 1-B. Driven by a passion for software automation and practical engineering solutions, his academic interests focus on logical system design and relational database architectures. For this research, he channeled his enthusiasm into rigorous data gathering, dedicatedly finding and vetting highly reliable academic sources to establish a solid foundation for the study. He is highly motivated to continue exploring how computer engineering principles can be used to build smart, automated systems that solve real-world problems and streamline administrative workflows.

**Prince Justine Rose A. Estella** is currently studying Bachelor of Science in Computer Engineering in Eulogio “Amang” Rodriguez Institute in Science and Technology (EARIST), Section 1-B. He’s known for his curiosity and interest in computers, especially when it comes to software automation and engineering. That’s where he truly excels, diving into the logical and relational parts of the work. For this study, he put in the time—always looking for solid, trustworthy academic sources to make sure he understood the basics. Prince doesn’t just stop when things get tough. He’s committed to learn about how computer engineering can shape smart, automated systems, and he’s always eager to delve deeper and expand knowledge.

**Gaveriel L. Corbe** is a student currently taking up a Bachelor of Science in Computer Engineering (BSCpE) at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST), Section 1-B. She has a strong interest in computers and learning about technology. In this research, she helped the team by creating and designing the flowcharts to make the study's process clear and easy to understand. She is always willing to learn more about computer engineering and wants to improve her skills for the future.

**Princess Mae Liz D. Camba** is a Bachelor of Science in Computer Engineering (BSCPE) student from section 1-B at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). Her interests lie in technology, software, and computer systems. She assisted in this research through finding credible sources and enhancing her knowledge on logical systems and databases. It is her objective to keep improving herself regarding computer engineering.

**Mica Angela A. Arañas** is a hardworking student who is currently finishing a Bachelor of Science in Computer Engineering (BSCpE) at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST), Section 1-B. Moved by a strong interest in building real things and making apps run automatically, this student focuses on how to plan smart networks, write code, and organize data tables. While working on this project, this individual spent time gathering facts and building the actual software, carefully looking for

trusted school papers to give the work a strong start. This student wants to keep finding ways to use engineering to make smart, automatic setups that fix real everyday problems and make regular work faster.

**Frances Mickyla M. Garcia** is a dedicated student currently pursuing a Bachelor of Science in Computer Engineering (BSCpE) at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST), Section 1-B. She has a strong passion for technology, software development, and computer systems, particularly in the areas of logical programming and database management. In this research, she contributed by gathering credible academic sources, organizing important information, and helping improve the overall quality of the study. She is eager to continuously enhance her knowledge and skills in computer engineering and hopes to develop smart and efficient technological solutions that can solve real-world problems in the future.

**Engr. Meshelle N. Fabro** is a licensed Professional Computer Engineer and educator at the Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). She is currently pursuing a Master of Science in Computer Engineering at Bulacan State University and has professional industry experience as a Technical Engineer at Hewlett-Packard (HP) and an IT Engineer at IBM. She is also a TESDA CSS NC II and Trainers Methodology Level I (TM1) holder, demonstrating her commitment to technical excellence and competency-based education. Passionate about teaching and innovation, she is dedicated to guiding Computer Engineering students in becoming industry-ready, competent, and highly skilled individuals equipped for the evolving demands of the technology industry.