DIGITAL STUDENT RECORD AND PERFORMANCE TRACKING SYSTEM

Akash D[1]

[1]B.Tech, Artificial Intelligence And Machine Learning, Sri Shakthi Institute Of Engineering And Technology, Coimbatore, Tamilnadu, India.

ABSTRACT

This paper presents a Smart Centralized Student Portfolio Management System designed to streamline the management of student records, performance, and achievements in academic institutions. The system enables administrators to maintain and monitor comprehensive student data, including academic performance (CGPA and attendance), technical skills, certifications, and achievements, in a centralized digital portal. Students can log in to view their personal profiles and progress, while administrators can update and manage data through an easy-to-use interface. Built using Python, Tkinter, and SQLite, the system ensures efficient data storage, secure authentication, and intuitive interaction. The platform eliminates manual recordkeeping, minimizes redundancy, and provides a sustainable digital alternative for department-level student management.

Keywords: Student Portfolio, Academic Record Management, Tkinter, SQLite, Digital Campus System, CGPA Tracking, Attendance Management, Skill Mapping, Certificate Management, Student Achievements.

1. INTRODUCTION

In many educational institutions, maintaining student records manually is time-consuming and prone to inconsistencies. Data such as CGPA, attendance, and achievements are often scattered across multiple files, making retrieval and updates difficult. To address these challenges, the Smart Centralized Student Portfolio Management System offers a unified digital solution for storing and managing student data. This system provides separate logins for administrators and students. The admin can add, edit, or remove student records, manage semester-wise CGPA and attendance, and upload details of skills, certificates, and achievements. Students can log in using their credentials to access their profiles, view performance, and track academic growth.

The project uses Python (Tkinter) for the graphical interface and SQLite as the local database, ensuring reliability, portability, and simplicity. The system enhances transparency, saves time, and promotes a paperless approach to record management.

2. METHODOLOGY

The proposed system follows a modular methodology to ensure organized data flow and secure operations.

2.1 Authentication System:

The system includes two login roles: Admin and Student.

- Admin Login: Access to add, modify, and delete records.
- Student Login: Access to view personal portfolio details.

Default student credentials are generated automatically upon record creation.

2.2 Student Information Management:

Admins can input essential student details, including roll number, name, department, year, email, and phone number. Each student's data is linked with their performance, skills, and achievements.

2.3 Academic Performance Module:

This module records semester-wise CGPA and attendance data. The performance table is linked by student ID, ensuring accurate mapping of each student's academic progress.

2.4 Skills and Achievements Module:

The admin can add students' skills, certificates, and achievements with supporting details such as title, issuing authority, and year. This forms the foundation of each student's digital portfolio.

2.5 Database Integration:

The backend uses SQLite, a lightweight and self-contained relational database. It handles student records, authentication details, and performance data efficiently.

2.6 Portfolio Generation:

Students can view their complete portfolio — including academic details, skills, and achievements — and optionally export it as a PDF or CSV file for reporting or placement purposes.

3. MODELING AND ANALYSIS

The proposed Smart Centralized Student Portfolio and Academic Record Management System follows a modular architecture consisting of three main layers: the Interface Layer, Logic Layer, and Database Layer. The frontend, developed using Python's Tkinter, enables administrators and students to log in securely, manage personal profiles, update academic records such as CGPA and attendance, and upload details about skills, certifications, and achievements. The interface ensures a user-friendly experience for data entry and retrieval. The processing layer, built in Python, handles authentication, CRUD operations, and data validation. It ensures that all inputs are verified before being stored, and automates processes such as student credential generation and report creation. This layer acts as the communication bridge between the interface and the database. The database layer utilizes SQLite, providing lightweight and reliable data storage for all student information. It maintains records of academic performance, attendance statistics, and portfolio achievements in a structured and easily retrievable format. The analysis of the system shows improved efficiency in managing departmental data, reduced manual record-keeping errors, and enhanced accessibility through centralized storage. Overall, the model delivers a secure, scalable, and transparent solution for digital academic record management within educational institutions.
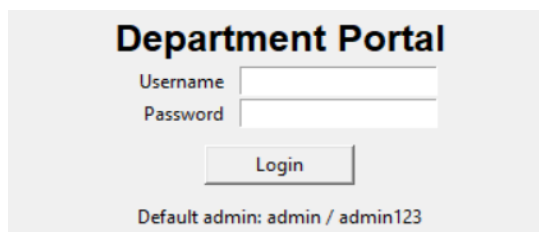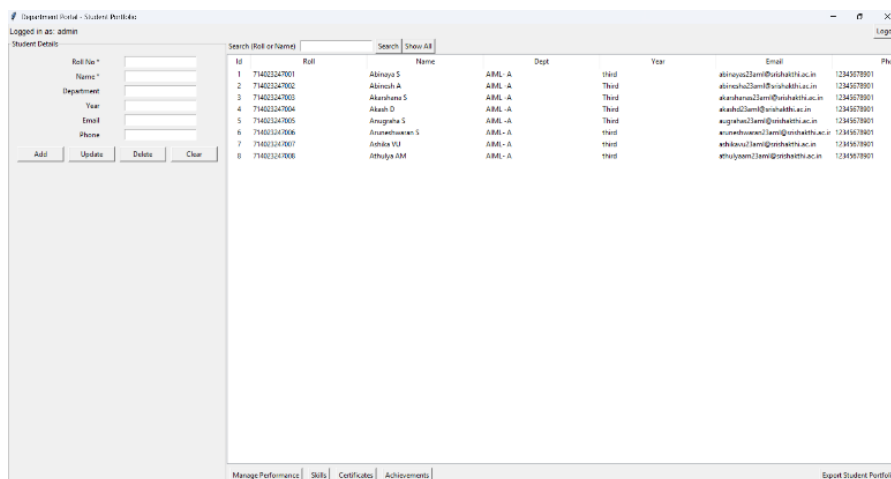

Figure 3.1: Login page
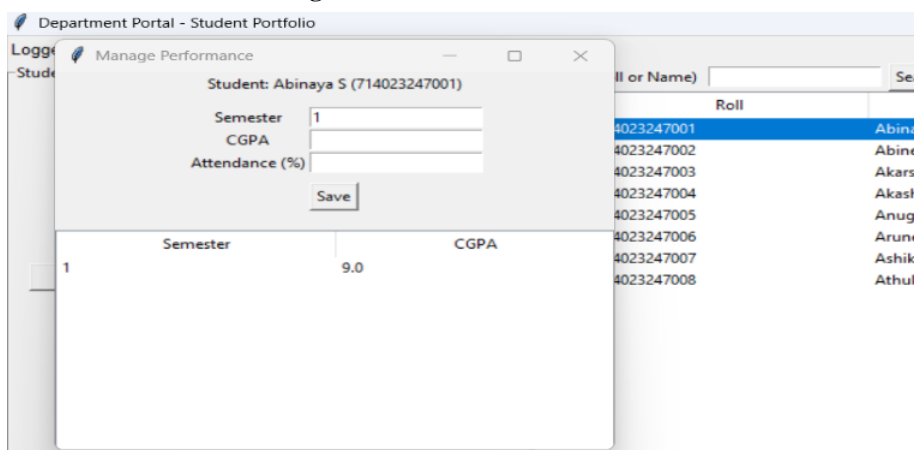

Figure 3.2: Main admin interface
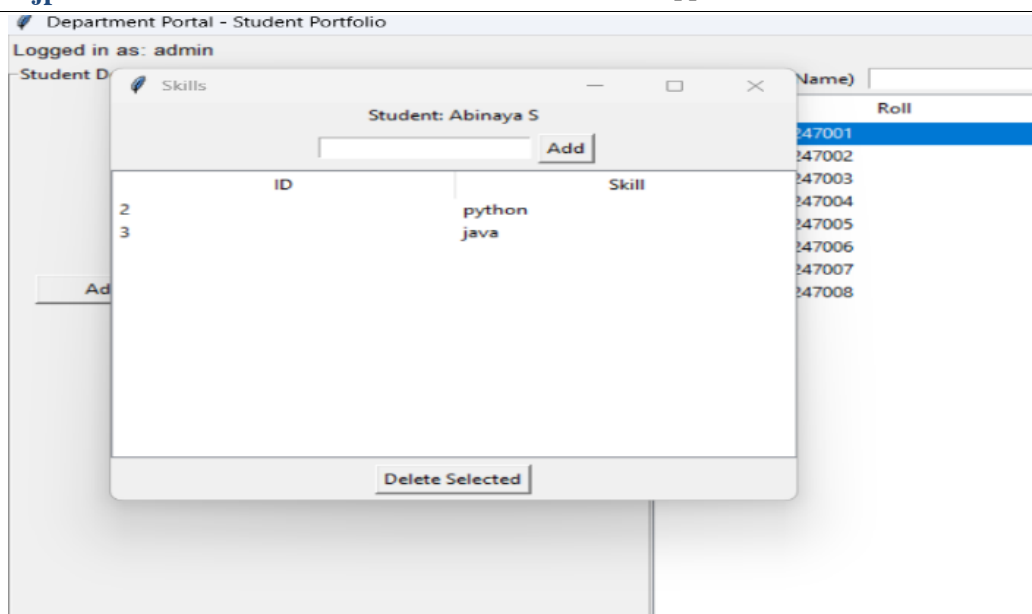

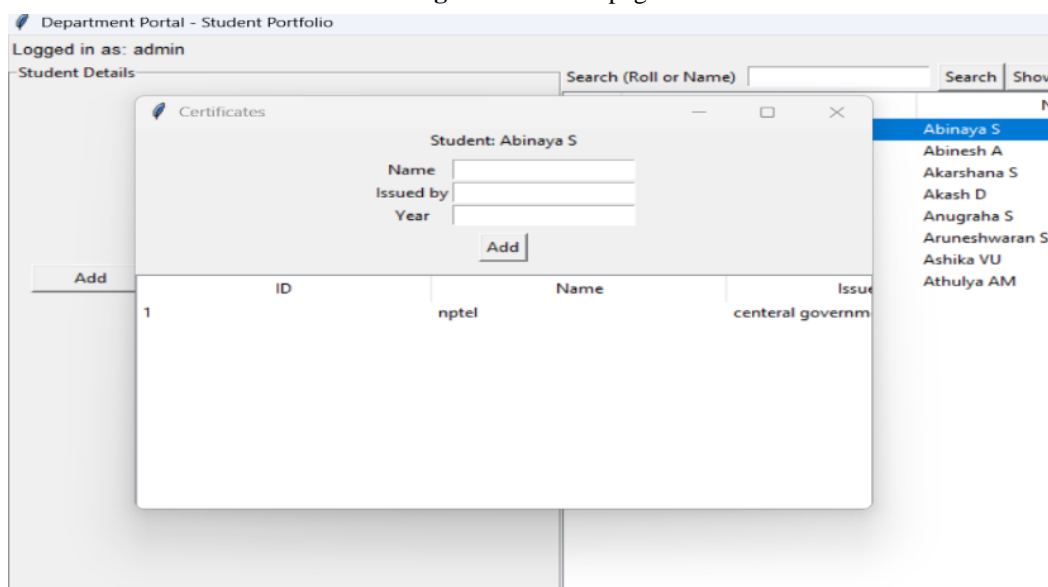**Figure 3.3:** Manage performance page

Figure 3.4: Skills page

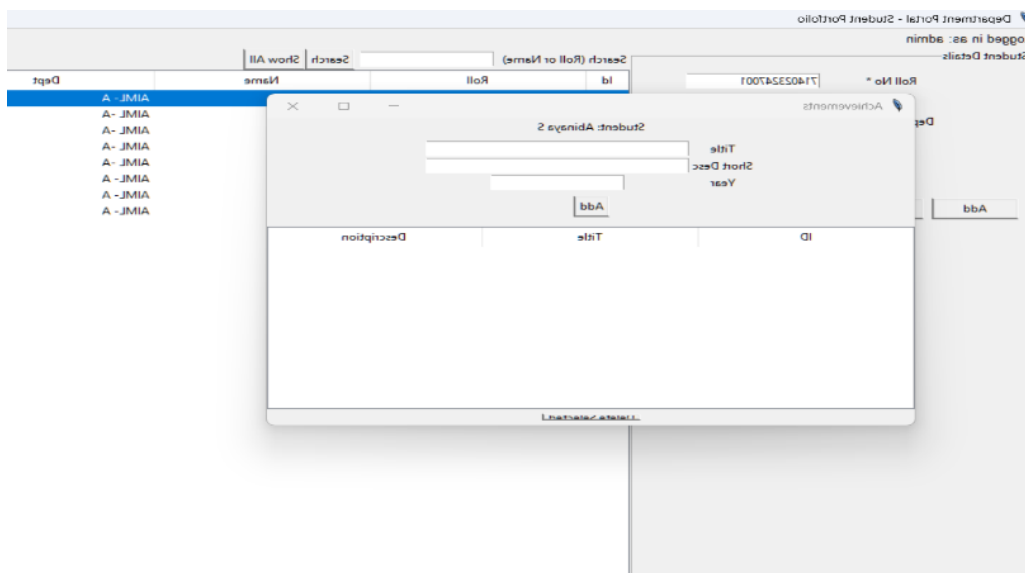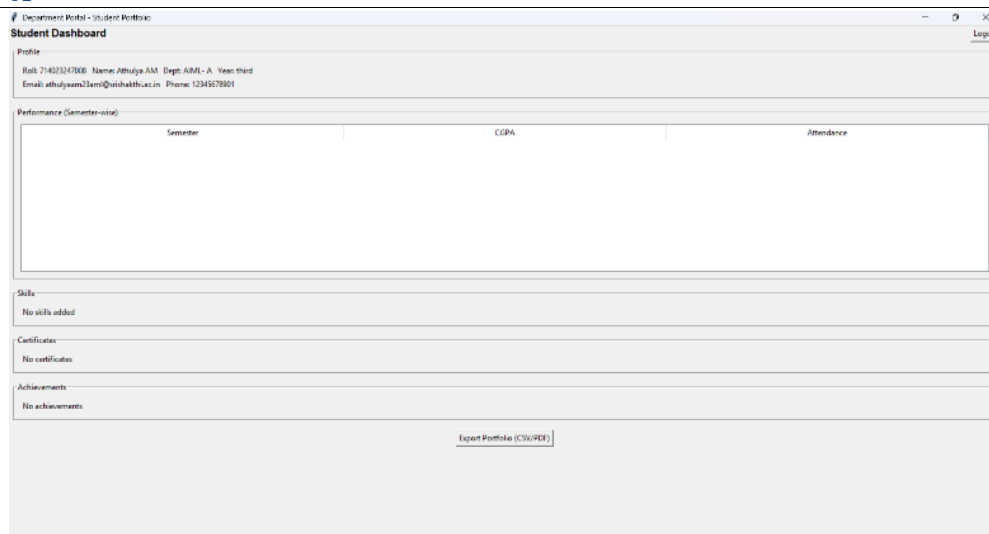
Figure 3.5: Certificate page

Figure 3.6: Achievement

Figure 3.7: Individual student portfolio

4. CONCLUSION

The Smart Centralized Student Portfolio and Academic Record Management System effectively digitalizes student data handling by integrating academic performance, attendance, skills, and achievements into a unified platform. The system improves data accuracy, transparency, and accessibility while reducing manual work for administrators. Built using Python Tkinter and SQLite, it ensures secure, reliable, and scalable management of student information. Overall, the project enhances institutional efficiency and provides a foundation for future upgrades like cloud integration and AI-based analytics.

5. REFERENCES

[1] A. Sharma and R. Gupta, "Digital Academic Record Management System Using Database Integration," International Journal of Computer Applications, Vol. 185, Issue 12, 2019, pp. 25–31.

[2] S. Kumar and P. Reddy, "Smart Student Information Management System Using Python and SQLite," International Journal of Advanced Research in Computer Science, Vol. 11, Issue 5, 2020, pp. 48–55.

[3] M. Priya and N. Karthik, "Automation of Academic Data Handling Using GUI-Based Applications," International Research Journal of Engineering and Technology (IRJET), Vol. 7, Issue 6, 2020, pp. 2105–2112.

[4] R. Bansal and T. Singh, "Design and Implementation of Centralized Student Database System," IEEE Conference on Smart Computing and Informatics, 2021, pp. 550–556.

[5] P. S. Rao and A. Thomas, "Institutional Record Management Using Cloud and Local Databases," International Journal of Engineering and Technology (IJET), Vol. 9, No. 4, 2021, pp. 300–306.

[6] Government of Tamil Nadu, "Higher Education Digitalization Initiative – Smart Campus and Data Portals," Department of Higher Education Report, 2023.

[7] S. Ankit and M. Raj, "Smart Portfolio Tracking System for Academic Institutions," International Journal of Emerging Technologies and Innovative Research (JETIR), Vol. 10, Issue 2, 2022, pp. 550–557.

[8] Python Software Foundation, "Tkinter Library Documentation – GUI Development in Python," Python.org Official Documentation, 2023.

[9] W. Smith and L. Brown, "Enhancing User Experience in Student Information Portals Using Modern UI Frameworks," ACM Digital Library, Vol. 17, Issue 3, 2023, pp. 110–117.

[10] Ministry of Education, Government of India, "Digital Campus and e-Governance in Higher Education," Official Report, 2023.