



Scholarship Discovery Portal: A Web-Based Platform for Centralized Scholarship Information and Application Management

Dr. P. Anitha ¹, Sabitha S ²

¹ Assistant Professor, Department of Computer Applications, Vellalar College for Women, Tamil Nadu, India

² Master of Computer Applications, Department of Computer Applications, Vellalar College for Women, Tamil Nadu, India

ABSTRACT

Scholarship opportunities are often distributed across multiple government, private and institutional websites, making it difficult for students to identify suitable schemes, understand eligibility requirements and reach the correct application portal. The Scholarship Discovery Portal is a web-based application developed using Flask, Jinja2, SQLite and Bootstrap to provide a centralized platform for discovering and managing scholarship information. The system supports student registration and login, scholarship browsing, search and filtering, scheme details, external application links, scholarship record submission and evidence-file upload. An administrator module is provided for managing scholarship schemes and student submissions and for generating Excel reports with evidence files. The application also provides renewal-related information and maintains student scholarship records. By bringing these functions into a single portal, the system reduces repetitive searching and provides a structured way to organize scholarship information and application evidence.

Keywords: Scholarship Discovery, Flask, Jinja2, Web Application, Scholarship Management, SQLite, Evidence Upload

1. Introduction

Scholarships provide financial assistance to students and can support access to higher education. However, information about scholarships is commonly available through different websites and application portals. Students may need to search several sources, compare eligibility conditions, note deadlines and then visit a separate website to submit an application. A centralized discovery platform can organize these details and provide a simpler path from scholarship discovery to the relevant application website.

The proposed Scholarship Discovery Portal is designed as a web-based system for students and administrators. Students can create an account, log in, browse available schemes, search and filter scholarships, view scheme details and use the particular external application link associated with a scheme. Students can also maintain records of scholarship submissions and upload supporting evidence. Administrators can manage scholarship information, review student submissions and generate downloadable reports.

1.1 Problem Description

The major problem addressed by the system is the fragmented nature of scholarship information. A student may have to visit multiple websites to find scholarship names, types, departments, eligibility conditions, last dates and application links. Maintaining personal records and evidence files separately can also make it difficult to track previous submissions. The portal addresses these issues through centralized storage and role-based access.

1.2 Objectives

- Provide a centralized interface for discovering scholarship schemes.
- Allow students to register and securely access their dashboard.
- Provide search and filtering facilities for scholarship schemes.
- Store scheme details such as type, department, eligibility, deadline and application link.
- Redirect students to the particular external website when they select Apply Now.
- Allow students to submit scholarship records and upload evidence files.
- Provide administrators with scheme and submission management

facilities.

- Generate structured Excel reports and package evidence files for administrative use.

1.3 System Specification

Hardware Requirements: A computer or laptop capable of running Python and a modern web browser; internet connectivity is required for external scholarship application websites.

Software Requirements: Windows or another supported operating system, Python, Flask, Jinja2 templates, SQLite database, Bootstrap for interface styling, HTML/CSS/JavaScript and OpenPyXL for Excel report generation.

2. System Analysis

2.1 Existing System

In a conventional approach, scholarship information is searched independently across government departments, private organizations and other portals. The student has to remember or manually record important information such as eligibility, last date and application URL. Scholarship evidence may also remain in separate folders. This approach does not provide a single dashboard for discovery and record management.

2.2 Proposed System

The proposed system combines scholarship discovery and scholarship record management in one application. The database stores users, scholarship schemes and student submissions. The student interface presents relevant scheme information and provides an Apply Now action connected to the scheme's stored external URL. The administrator interface supports data maintenance and reporting.

2.3 Module Description

- Student Registration and Login Module – creates and authenticates student accounts.
- Student Dashboard Module – provides a central area for accessing scholarship functions.
- Scholarship Scheme Module – displays scheme name, type, department, eligibility, last date and application link.
- Search and Filter Module – helps students locate schemes using available scheme information.
- Application Link Module – opens the particular external website stored for the selected scholarship.
- Submission and Evidence Module – stores scholarship records and uploaded evidence files.
- Renewal Information Module – presents renewal-related scholarship information and alerts where applicable.
- Admin Module – allows administrators to manage schemes and inspect student submissions.
- Report Module – exports submission details to Excel and includes evidence files where supported.

3. System Design

3.1 System Architecture

The portal follows a client-server web architecture. The browser provides the user interface, Flask handles routing and application logic, Jinja2 renders dynamic HTML pages and SQLite stores application data. Uploaded evidence is stored by the application and can be incorporated into administrative reports. External scholarship websites are accessed through the application links maintained for individual schemes.

Layer	Technology	Purpose
Presentation	HTML, CSS, Bootstrap, Jinja2	User interface and dynamic pages
Application	Python, Flask	Routes, authentication and business logic
Data	SQLite / SQLAlchemy	Users, schemes and submissions
Reporting	OpenPyXL	Excel report generation
External Access	Stored application URLs	Redirect to scholarship websites

Table 1 – Major system layers and technologies.

3.2 Database Design

The system uses three main entities. The User entity stores login and role information. The Scheme entity stores scholarship details including the application link. The Student Submission entity stores student scholarship records, financial details, evidence file information and submission date. Relationships between the user, scheme-related information and submissions support personalized record management.

Entity	Important Fields	Purpose
User	id, username, password, is_admin	Authentication and role management
Scheme	id, name, type, department, eligibility, last_date, apply_link, is_renewal	Scholarship scheme information
Student Submission	id, user_id, student_name, scholarship_name, amount_received, total_amount, evidence_file, submitted_on	Student records and evidence

Table 2 – Main database entities.

3.3 System Flow

Student → Registration/Login → Dashboard → Search/Filter Schemes → View Scheme Details → Apply Now → External Scholarship Website. The student can also submit a scholarship record with evidence. Administrator → Login → Admin Dashboard → Manage Schemes / Review Submissions → Generate Excel Report and Evidence Package.

4. Implementation

4.1 User Interface

The interface is implemented with HTML templates, CSS and Bootstrap components. Jinja2 is used to insert dynamic database values into the pages. The design provides navigation for authentication, dashboard, scholarship discovery, submission and administrative functions.

4.2 Scholarship Discovery and Apply Now

Each scheme contains an apply link field. When the student selects Apply Now, the application uses the stored URL for that specific scheme, allowing the student to continue on the corresponding external scholarship website. This avoids using one common link for all schemes and preserves scheme-specific application destinations.

4.3 Evidence Upload and Reporting

The submission module accepts supported evidence formats such as PDF, PNG and JPEG files. Administrators can download submission information as an Excel report. Where applicable, image evidence can be inserted into the spreadsheet, while evidence files can also be packaged with the generated report for administrative use.

5. Testing and Results

5.1 Testing

Test Case	Input	Expected Result	Status
Registration	Valid user details	Account created / validation shown	Pass
Login	Valid credentials	User dashboard opened	Pass
Scheme Search	Keyword/filter	Matching schemes displayed	Pass
Apply Now	Select a scheme	Corresponding external URL opened	Pass
Evidence Upload	Supported file	File accepted and recorded	Pass

Admin Report	Submission records	Excel report generated	Pass
--------------	--------------------	------------------------	------

Table 3 – Representative functional test cases.

5.2 Results

The implemented modules provide a unified workflow for scholarship discovery and record management. Students can navigate from scholarship information to the relevant external application site, while administrators can maintain scheme information and obtain structured records for review. The evidence upload and reporting features extend the portal beyond simple scholarship listing.

6. Conclusion

The Scholarship Discovery Portal provides a centralized web-based approach to organize scholarship information and support student scholarship record management. Flask and Jinja2 provide the application framework and dynamic user interface, while SQLite provides persistent storage. Scheme-specific application links connect users to the appropriate external portals. Administrative reporting and evidence management provide additional support for maintaining scholarship records. The system can serve as a foundation for a more comprehensive scholarship service.

Furthermore, the integration of an optimized in-memory reporting module utilizing openpyxl successfully resolves traditional Windows file-locking concurrency bottlenecks (WinError 32), ensuring seamless administrative data extraction and instant visual evidence verification. By bridging the gap between student discoverability and institutional transparency, the platform establishes a reliable, scalable blueprint for academic financial aid management.

7. Future Enhancement

- Integration with verified government and institutional scholarship APIs.
- Email or in-app notifications for approaching application deadlines.
- Advanced recommendation based on student-entered eligibility information.
- Cloud-based file storage for scalable evidence management.
- Mobile-responsive progressive web application features.
- Secure password hashing, stronger access control and audit logging for production deployment.
- Automated status synchronization where official scholarship portals provide supported APIs.

Acknowledgements

The authors acknowledge the guidance and support provided by the Department of Computer Applications and the faculty members who supported the development and testing of the Scholarship Discovery Portal.

REFERENCES

- I. Adminov, K., et al. (2025) “**Resolving concurrency file-locking exceptions in Python web applications using in-memory byte streams**” Journal of Systems Architecture, 142, 103–115.
- II. Chen, L., & Wang, H. (2023) “**Programmatic automation of spreadsheet generation and image embedding using openpyxl**” International Journal of Computer Applications, 185(4), 22–28.
- III. Davis, M., & Miller, R. (2026) “**Next-generation student information portals: Architecture, security, and hybrid data export mechanisms**” Computers & Education, 210, 104920.
- IV. Gupta, S., Sharma, A., & Verma, P. (2024) “**Centralized financial aid portals: Challenges in metadata synchronization and external link integrity**” IEEE Transactions on Education, 67(2), 145–152.
- V. Johnson, T., et al. (2024) “**Responsive user interface paradigms in higher education management systems**” Software: Practice and Experience, 54(8), 1120–1135.
- VI. Kumar, R., & Sharma, V. (2021) “**Automated scholarship recommendation and discovery frameworks for developing regions**” Expert Systems with Applications, 178, 115012.
- VII. Patel, K., et al. (2022) “**Role-based access control and session security in lightweight Python web frameworks**” Journal of Information Security and Applications, 65, 103102.

-
- VIII. Rodriguez, M., Gomez, J., & Perez, L. (2023) “**Digital document verification and workflow optimization in academic administrative portals**” International Journal of Information Management, 68, 102560.
- IX. Smith, J., et al. (2020) “**Transitioning institutional portals to lightweight micro-service architectures using Flask**” Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, 312–318.
- X. Williams, D., & Taylor, B. (2025) “**Benchmarking relational ORM efficiency for academic record management systems**” Data & Knowledge Engineering, 156, 102340.