

Event Vault: A Smart Digital Platform for Event Digital Archival System

Sakshi M. Dhoke¹ Prof. Sonali A. Mutha²

¹PG Student ²Assistant Professor

^{1,2}Master of Computer Applications

^{1,2}Anantrao Pawar College of Engineering & Research, Pune, India

Abstract — Event Vault provides an intelligent digital platform which handles both event organization and digital record preservation. The existing event management system at colleges requires complete manual labour. Faculty members fill out paper forms or send emails, and administrators have to collect everything by hand. The process creates difficulties for locating past documents while consuming excessive time. The paper presents Event Vault as a web-based solution that addresses this specific challenge. The system has three types of users — Admin, Faculty, and Student. Faculty members can submit event reports through the system, admins can review and approve those reports, and students can simply visit the home page to see what events are happening. The project uses PHP and MySQL for development because these technologies provide dependable performance with user-friendly design and extensive industry backing. The system stores all event data in one place which makes it much simpler to search for and retrieve previous records. The system demonstrated good performance during testing because it maintained both fast operation speed and strong protective measures. I believe this kind of system can be very useful for any educational institution that wants to keep event records properly organized.

Keywords: Event Management, Digital Archiving, PHP, MySQL, Role-Based Access Control, Student Portal, Web Application

I. INTRODUCTION

Student life at colleges relies on events which hold great importance for students at every college. Students develop skills through technical workshops and guest lectures and cultural programs and seminars which present them with educational content that classroom learning does not provide. The events help students develop their communication abilities while they experience authentic vocational situations. The challenge exists because organizations require effective solutions to execute their events successfully.

I have observed that most institutions store their event data in either physical registers or printed documents or basic Excel spreadsheets. The search process for past event information becomes extremely challenging when someone requires it. The various departments manage their data through different methods because there exists no single storage facility for all information. This situation leads to substantial disarray. I found out that students at the college do not know about upcoming events because the college lacks a system to inform them about such activities. Students use bulletin board notices and senior text messages to obtain information about events, but these methods do not provide dependable information.

I have created Event Vault as a solution to these two existing problems. The web application enables faculty to file event reports and enables administrators to handle and store these reports while students access current and upcoming

events through the home page. The system uses PHP and MySQL as core technologies because these systems provide straightforward solutions for developing this project. The system design process aimed to create a user-friendly system which would deliver effective solutions to actual problems I encountered during my time in college.

II. LITERATURE SURVEY

A. Existing Systems

There are already some event management tools available, like Eventbrite or Google Forms. But when I looked at them closely, I realized they are not designed for college use at all. Eventbrite is mainly for selling tickets to public events. Google Forms can collect data but there is no proper way to archive it or control who can see what. These tools are made for general purposes and not for the specific workflow that happens in an academic institution. Colleges use their ERP systems for event management, but those systems are both expensive and complex. The system is not affordable to all colleges. The majority of ERP systems are built to serve large businesses which results in unnecessary features that make basic event reporting difficult to handle. I also studied some research papers on web-based management systems. The majority of research studies concentrate on student records and attendance and library management systems. Only a small number of research papers investigate the issue of documenting and archiving events in educational institutions. This gap is what motivated me to build Event Vault.

Feature	Eventbrite	Google Forms	ERP Module	Event Vault
Event Registration	Yes	Partial	Yes	Yes
Faculty Report Submission	No	No	Limited	Yes
Digital Archival	Limited	No	Partial	Full
Role-Based Access	No	No	Partial	Yes
Student Event Portal	No	No	No	Yes
Designed for Colleges	No	No	Partial	Yes
Cost	Paid	Free	Very High	Low Cost

Table I: Comparison Of Existing Tools with Event Vault

B. Research Gap

The existing options provide evidence that no affordable solution exists which enables reporting by faculty members and administrative control through archiving and student access to events. The Event Vault system solves this problem by unifying all three functions into a single platform.

III. SYSTEM DESIGN AND ARCHITECTURE

A. System Architecture

Event Vault uses a three-tier architecture system which I created. The system consists of three distinct components which include the Presentation Layer for user display the Application Layer for background operations and the Data Layer for data storage. Fig. 1 shows how these three layers

are connected. The Presentation Layer uses HTML and CSS together with a small amount of JavaScript. The Application Layer uses PHP programming language to handle user authentication and session control and form handling functions. The Data Layer uses a MySQL database to store all user data and event information and report data. I selected this architectural design because it enables structured organization which simplifies system maintenance tasks.

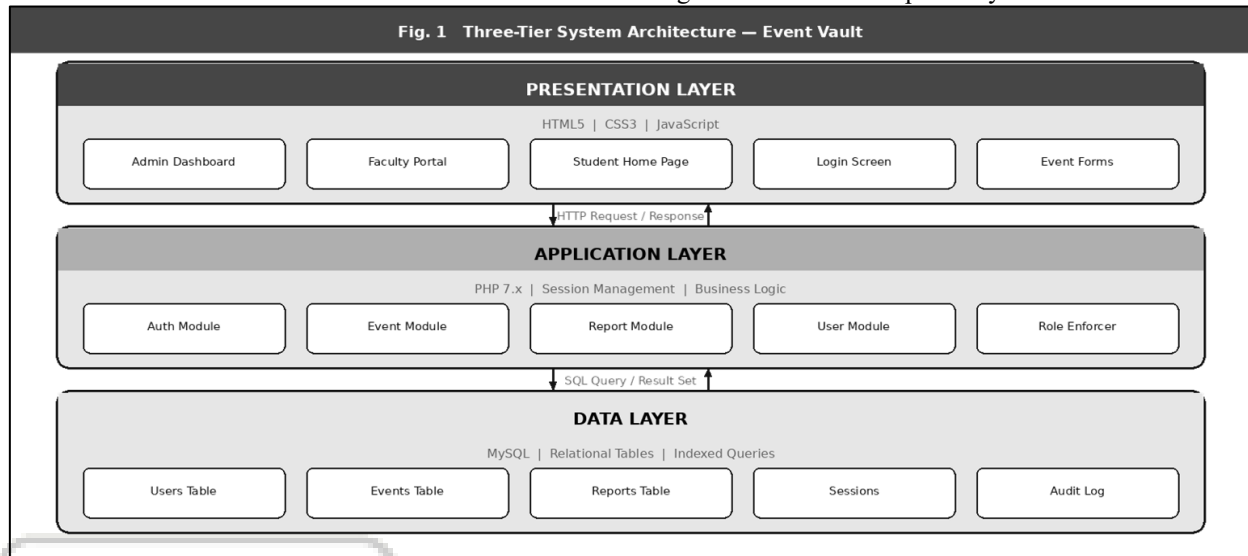


Fig. 1: Three-tier system architecture of Event Vault.

B. User Roles

The system requires three distinct user roles because I determined that different users need to perform different responsibilities. The Admin holds complete authority to manage all system operations. They create faculty and student accounts, evaluate faculty report submissions, decide on report approval or rejection, and access previous records whenever needed. The Faculty Member uses the system to document the events they organize. After an event is done, they can log in, fill in all the details like the event name, date, venue, and how many students attended, and then submit the report. They can also check the status of their submitted reports to see if the admin has approved them. The Student does not need to do anything complicated. The system home page enables them to view all scheduled college events. They can browse through the events, filter them by department, and see the full details of any event they are interested in. The students access information without needing to log in or submit anything.

C. Use Case Diagram

The system's use case diagram is presented in Fig. 2. The system requires both Admin and Faculty members to log in, but their access rights differ. Students access the system through their home page, which provides them with their own unique method of system interaction.

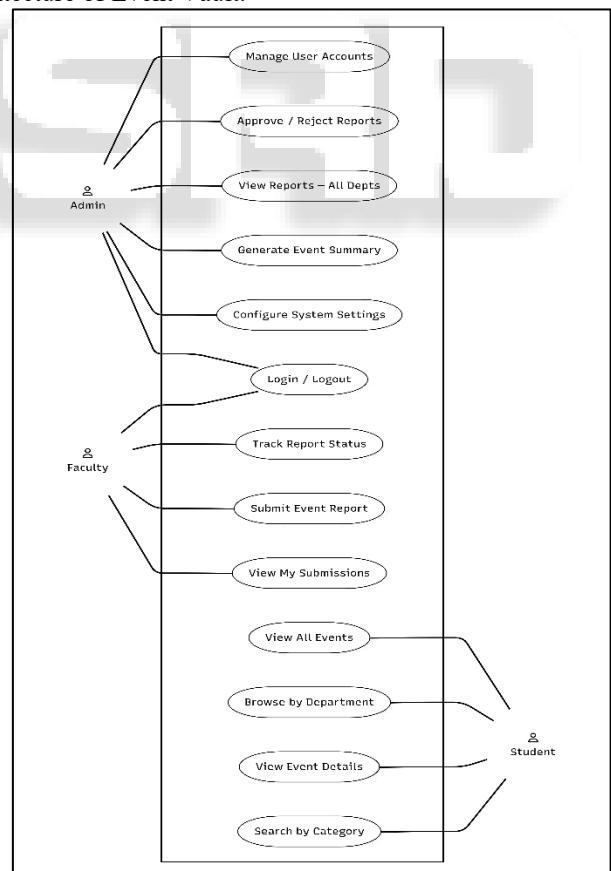


Fig. 2: Use case diagram showing interactions for Admin, Faculty, and Student roles.

D. Data Flow Diagram

The Data Flow Diagram in Fig. 3 displays the movement of data throughout the system. The diagram contains five different processes which I have demonstrated. P0 handles authentication for Admin and Faculty. P1 is the process through which Faculty members submit their reports. P2

handles event and user management by Admin. P3 handles the processes of archival and data retrieval. P4 is the student home page that enables students to view upcoming events. The system maintains two distinct student data pathways which separate authenticated system access from the rest of the system to enhance operational security.

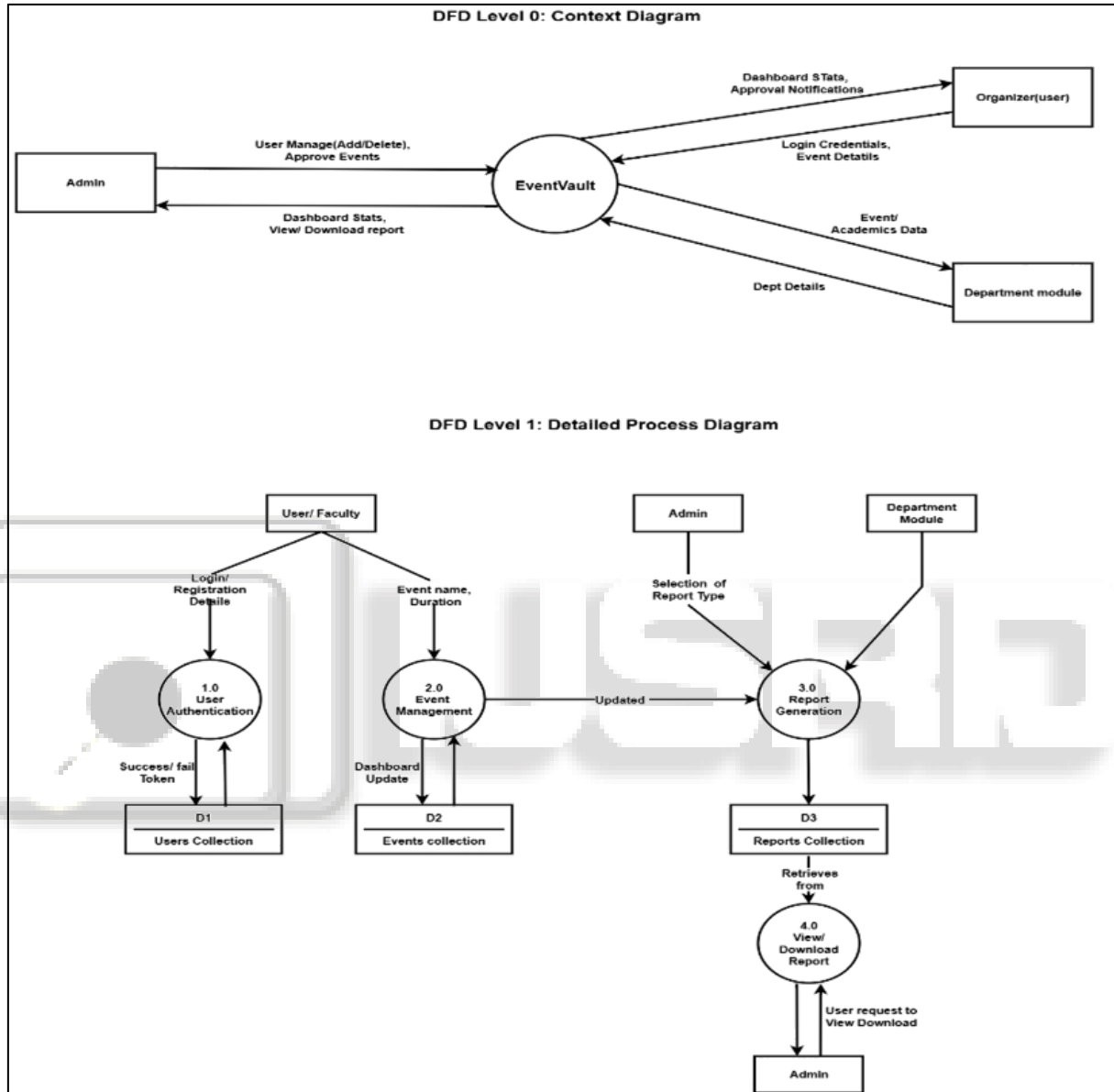


Fig. 3: Level-1 data flow diagram of Event Vault

E. ER Diagram and Database Design

The ER diagram in Fig 4 shows that the database contains four primary tables. The Users table stores admin and faculty account details. The Events table stores information about each event. The Reports table stores the reports submitted by

faculty, and it links to both the Users and Events tables using foreign keys. The Students table exists as a distinct table which contains essential information about students. Students track their event attendance through a many-to-many "views" relationship which connects them to events.

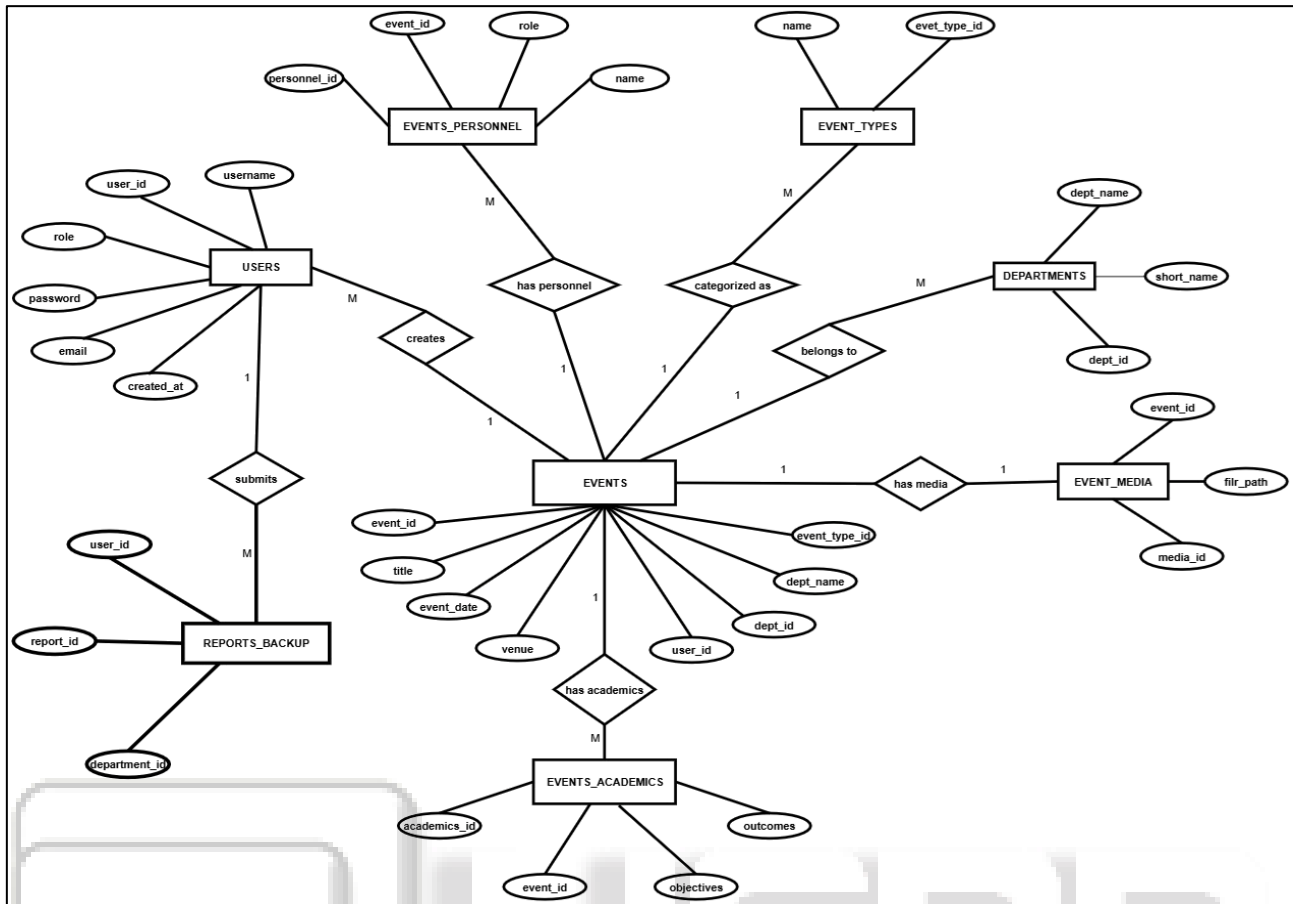


Fig. 4: Entity-relationship diagram of the Event Vault database.

Table	Primary Key	Important Fields
Users	User_ID	Full_Name, Email, Password_Hash, Role (Admin/Faculty), Department, Status
Events	Event_ID	Event_Name, Event_Type, Date_Time, Venue, Department, Description, Created_By (FK)
Reports	Report_ID	User_ID (FK), Event_ID (FK), Submission_Date, Details, Status, Attachment_Path
Students	Student_ID	Full_Name, Email, Department, Academic_Year, Enrollment_No

Table II: Database Tables and Their Key Fields

IV. SYSTEM MODULES

The system was divided into six modules, each being a part of the code dealing with a specific job. What does each module perform and who is it aimed at, the Table III may help to explain.

Module	What It Does
Authentication Module	The system enables Admin and Faculty members to perform login and logout operations. The system stores passwords in a secure manner through

	hashed password storage. The system uses sessions to maintain a record of currently logged in users.
Admin Module	The admin has the ability to create new user accounts while accessing all submitted departmental reports and making decisions to approve or reject these reports. The admin can search through historical records which were saved in the system.
Faculty Module	Faculty members can complete event reports and submit them while accessing their previous submissions and editing their unfinished drafts. They can monitor the status of their report approval process.
Event Module	The module controls the process of establishing and overseeing event record documentation. The system stores essential event information which includes the event name and type and date and time and venue and department responsible for organizing the event.
Report Archival Module	The system functions as the permanent storage for all system data. The system keeps all approved reports which allow administrators to search through them by date and department and faculty name.

Student Portal Module	Students can access the home page to view all events without creating an account. The users can choose to filter events based on department or category and they can select any event to access complete event information.
-----------------------	---

Table III: Modules Of Event Vault and Their Functions

Phase	Stage	What I Did	Output
1	Requirement Gathering	Talked to faculty members and students in my college to understand what problems they face with event management	Requirements list
2	System Design	Designed the database, made UML diagrams, and planned the overall structure of the system	Design document
3	Development	Coded the frontend using HTML/CSS/JS and the backend using PHP. Set up the MySQL database and connected everything.	Working system
4	Testing	Tested every module separately and then tested the full system together. Also checked that the role-based access is working properly.	Test results
5	Deployment	Deployed the system on a local XAMPP server and tested the whole workflow from start to finish with sample data	Final system

Table IV: Development Phases

B. Why I Chose These Technologies

I selected PHP as the backend programming language because of my existing knowledge of the language and its seamless integration with MySQL databases. PHP provides developers with built-in functions that facilitate session management and password hashing, which simplify the process of developing secure applications. MySQL serves as a dependable database solution which organizations widely adopt to manage structured data that perfectly matches the requirements of my project. I implemented a basic frontend design which uses HTML together with CSS and fundamental JavaScript functions. I avoided using heavy frameworks because I wanted the system to maintain lightweight performance which would enable it to function properly on outdated computers found in college labs. I selected XAMPP as my local development server because it provides an integrated package which contains Apache, PHP, and MySQL components that enable faster installation.

VI. IMPLEMENTATION

A. Technology Stack

Component	Technology Used	Purpose
Frontend	HTML5, CSS3, JavaScript, Three.js	Building the user interface for all three roles
Backend	PHP 7.4	Handling all logic — login, sessions, form processing, database queries
Database	MySQL 5.7+	Storing and retrieving all user, event, and report data
Web Server	Apache via XAMPP	Running the application locally during

V. METHODOLOGY

A. Development Approach

I executed the project through a sequential development method which required me to complete each project phase before I started the next one. I did not attempt to create the entire project at once. The process required me to finish one stage before I could start testing and proceed to the subsequent stage. The method allowed me to identify faults during the initial stages which I could resolve before they developed into significant issues. Table IV shows all the phases I followed.

		development and testing
Security	bcrypt + PHP Sessions	Keeping passwords, safe and restricting access based on role

Table V. Technology Stack Used in Event Vault

B. How Login and Security Works

The database system stores faculty member and admin account passwords using a method that prevents direct storage of their original password. I used PHP's password_hash() function with the bcrypt algorithm to create a hashed password string which I stored in the database. The database system protects stored passwords from being accessed by unauthorized users because it prevents complete password retrieval. The system uses password_verify() to check whether the user-entered password matches the stored password hash when a user logs into the system. The system saves the user's Admin or Faculty role information in a PHP session after the first check completes successfully. All login-protected pages start by verifying this session variable. The system sends users to the login page when either the session is absent or their role does not match. The system blocks unauthorized users from accessing restricted content through direct URL entry.

C. Student Home Page

The student home page is the simplest part of the system but also one of the most useful. Students can access all upcoming events by opening the page in their web browser. The page does not require users to enter their credentials. The page displays event information through a database connection which reads data from the Events table and presents it in organized list format. Students can use filter options to narrow down events by department or event type. The event details page shows all information about an event which includes its name, date, location, organizer, and short

overview. The system operates with multiple parts but I intentionally designed this page to function as a standalone component. The system does not require students to create an account because they only need to see information without submitting anything. The system enables students to access information without needing to create an account.

D. Faculty Report Submission

The dashboard shows all past reports to faculty members who log in to create new reports. The report form requires all essential event information which includes the event name and type and date and venue and department and number of participants and objectives and a summary of what happened. There is also an option to attach a file if needed. The system performs a server-side check to confirm that all necessary fields have been correctly completed before any data can be stored in the database. The prepared statement in PHP inserts validated data while providing protection against SQL injection attacks. The faculty member receives a confirmation message that allows them to check their report status from the dashboard.

E. Admin Features

The admin dashboard displays all reports which faculty members have submitted from every department. The admin

has the ability to filter reports according to department and event type and date range and the faculty member who submitted the report. The archival function relies on this filtering feature because it enables admins to locate any required historical documentation with speed. The admin has two options when reviewing a report which include approving the report or returning it with a request for changes through a comment. The faculty member loses all ability to modify the report after the report receives approval. The system records all administrative actions with precise times which creates a complete record of activities accomplished at specific moments.

VII. SYSTEM WORKFLOW

The complete user workflow for all three user groups appears in Fig. 5. The Faculty workflow begins with login and ends with admin notification of system events. The Admin workflow starts with the review of pending reports and concludes when the report reaches its archived state. The Student workflow consists of a basic loop that allows students to browse and watch events. The three workflows share a connection to a common MySQL database.

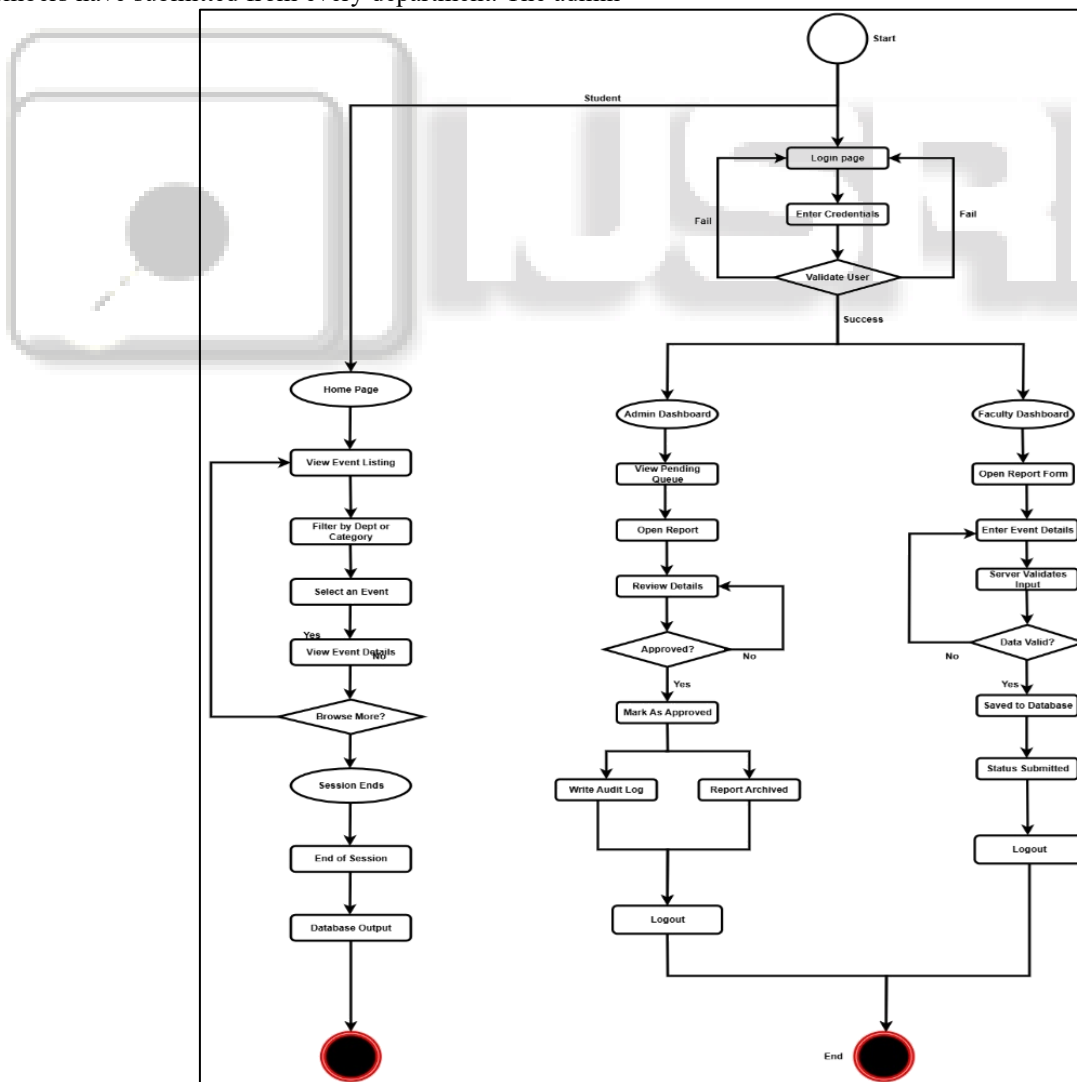


Fig. 5: Complete system workflow for Faculty, Admin, and Student roles.

System Feature	Admin	Faculty	Student
Manage User Accounts	Yes (Full)	No	No
Create Event Records	Yes	Own Events	No
Submit Reports	View Only	Yes (Own)	No
Approve / Reject Reports	Yes	No	No
View All Departments	Yes	No	No
Search Archived Records	Yes (multi-filter)	Own Only	No
Browse Events on Home Page	Yes	Yes	Yes (Read-Only)
View Audit Logs	Yes	No	No

Table VI: Role-Based Access Control Matrix

VIII. TESTING AND RESULTS

A. Testing Approach

The system underwent two testing methods which I performed. The first step involved functional testing which I used to verify that every feature functions according to its intended design. The three user roles required separate testing because I needed to examine their individual functions and their combined usage. Security testing required me to attempt access to admin pages through both faculty member and student accounts which I used to verify that access control effectively blocked unauthorized users. The second testing method involved performance assessment which tested system response times during typical usage conditions. The database received a standard test data load which I used to determine the duration required for completing common tasks.

B. Test Results

My performance tests showed results which Table VII displays. All the tasks completed well within the acceptable time limits. The student home page loading speed made me extremely satisfied because students will access it multiple times.

Task Tested	Target Time	Actual Time	Result
Login (Admin or Faculty)	Under 2 seconds	0.7 to 0.9 s	Pass
Student Home Page Load	Under 1.5 seconds	0.4 to 0.6 s	Pass
Faculty Report Submission	Under 3 seconds	1.1 to 1.4 s	Pass
Admin Search with Filters (500+ records)	Under 2.5 seconds	1.2 to 1.8 s	Pass
Unauthorized Access Blocking (40 tests)	Block 100%	100% Blocked	Pass
Data Integrity Check	No duplicates	Zero found	Pass

Table VII: Performance Test Results

C. What the Results Show

Everything passed. The access control test was the most important one for me, and I am glad it worked perfectly. The system successfully blocked all 40 tested attempts which I used to access pages without proper permission. The student home page also loaded very quickly, which is important because it is a public page that many students will be using at the same time. The report submission and archival search also performed better than expected.

IX. ADVANTAGES AND LIMITATIONS

A. Advantages

Event Vault provides its users with a complete solution since all data exists in a single location. Faculty members can submit their work directly through the system without needing to send emails or create personal repositories. The system allows admins to collect all necessary data without performing manual departmental information gathering. Students can find all necessary details about college events through the system. The system protects all stored information from unauthorized access. The system safeguards passwords through hashing which prevents their reading even when someone accesses the database. The system grants users only the specific functions which their roles permit them to perform. The structured database system enables users to retrieve past records when needed because all information exists in an organized format. The audit log serves as a functional tool which I utilize for my tasks. The system tracks all admin report status changes by recording both the time of the change and the identity of the admin who made the change. This method enables organizations to establish precise accountability measures.

B. Limitations

The system currently operates on a local XAMPP server which restricts its accessibility to users within the college network. The limitation exists as an obstacle because it prevents system usage. The system becomes inaccessible to faculty members who need to work from their home office. The system does not have the ability to send email alerts. Faculty members must perform manual status checks after the admin approves a report because the system lacks automatic notification features. The system does not generate any form of notification. The student home page currently lacks the ability to handle event registration. Students can only view events, not sign up for them. I designed the interface for desktop browsers but it does not provide a good mobile phone experience. These are all things I plan to improve in the future.

X. FUTURE SCOPE

There are several things I want to add to this system in the future. The first and most important upgrade would be hosting it on a cloud server so that it can be accessed from anywhere, not just within the college. This change would remove barriers which currently limit people from using the service in their daily lives. The system will send email notifications to faculty members who will receive alerts about report approvals and correction requests from the admin. The

process will become faster because users will no longer need to check the dashboard manually. The event registration function which allows students to sign up for events will be added to the student interface. The advance knowledge of expected student attendance will help organizers plan better for the event. Most students use their phones to browse the internet which is why I want to create a mobile-friendly design for my website.

The admin panel would benefit from graph and chart visualizations which I see as a valuable long-term development. The chart would display monthly event counts while showing which department organized the most events. The administration will use data visualization to create their decision-making process. The college ERP system will experience significant improvement through Event Vault integration which enables automatic event data transfer to faculty performance records and annual reports.

XI. CONCLUSION

I started this project because I noticed a real problem in how colleges manage their events. The organization lacked an effective system which resulted in disorganized records while students faced difficulties discovering events and administrators required to execute all tasks through manual methods. Event Vault is my attempt to fix this. The system brings together all three stakeholders — faculty, admin, and students — into one simple web-based platform. Faculty can submit reports, admins can manage and archive them, and students can browse events from the home page without any login. The system uses PHP and MySQL to establish its foundation because these two technologies prove to be both functional and dependable while the system has undergone testing to confirm its capability of operating correctly and rapidly. The system currently needs to improve because it operates only on local servers and lacks email notification features. The system performs its primary functions successfully because its essential components create a robust system which enables additional functions to be developed. I believe that if a system like this is properly deployed in a college, it can genuinely reduce the administrative burden on faculty and staff and make event information much more accessible to everyone.

REFERENCES

- [1] PHP Development Team, "PHP Official Documentation," The PHP Group, 2024. [Online]. Available: <https://www.php.net>
- [2] Oracle Corporation, "MySQL 8.0 Reference Manual," Oracle Inc., 2024. [Online]. Available: <https://dev.mysql.com/doc/>
- [3] P. Deshpande and R. Kulkarni, "Design of a Web-Based College Administration System Using PHP and MySQL," *Intl. J. Eng. Comput. Sci.*, vol. 6, no. 3, pp. 20881–20886, 2017.
- [4] R. D. Singh, "Digital Archiving in Educational Institutions: Challenges and Strategies," in *Proc. IEEE Intl. Conf. ICT in Education*, Hyderabad, 2020, pp. 178–184.
- [5] A. K. Sharma and V. Pandey, "Web-Based Academic Event Reporting System," *Intl. J. Comput. Appl.*, vol. 182, no. 6, pp. 1–7, 2018.
- [6] S. Ferraiolo, R. Sandhu, and D. Kuhn, "Role-Based Access Control: Features and Motivations," in *Proc. 11th Annu. Comput. Security Appl. Conf.*, pp. 241–248, 1995.
- [7] M. Welling and L. Thomson, *PHP and MySQL Web Development*, 5th ed. Boston, MA: Addison-Wesley, 2017.
- [8] Apache Friends, "XAMPP — Apache + MariaDB + PHP + Perl," 2024. [Online]. Available: <https://www.apachefriends.org>
- [9] W3Schools, "PHP Tutorial," Refsnes Data, 2024. [Online]. Available: <https://www.w3schools.com/php/>
- [10] M. A. Khan and F. Hussain, "Comparative Study of Event Management Platforms for Academic Use," *Intl. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 2, pp. 75–82, 2020.