CampusMate – A College Companion

Prof. S. R. Karjol[1], Mr. Ramesh Karjol[2], Abhishek Salunke[3], Aftab Faniband[4], Deepak Appayya Patil[5], Prashanth Basappa Pujari[6]

[1]*Assistant Professor, Department of Computer Science and Engineering, B.V.V. Sangha's Basaveshwar Engineering College, Bagalkote*

[2]*Lecturer, Mechanical Department, B.V.V.S. Polytechnic (Autonomous), Bagalkote*

[3,4,5,6]*Students, Department of Computer Science and Engineering, B.V.V. Sangha's Basaveshwar Engineering College, Bagalkote*

***Abstract*— CampusMate, a centralized web-based academic information management system designed to streamline communication and resource sharing among students, faculty, and administrators. The system is implemented using a Python Flask backend and a responsive web interface, incorporating role-based access control to ensure secure and efficient management of academic content. A simplified student dashboard enables quick access to semester-specific information, while faculty and administrators can upload, manage, and organize resources through authenticated modules. Experimental evaluation and usability testing indicate that the proposed system improves accessibility, reduces redundancy in information dissemination, and enhances overall academic workflow efficiency. The results demonstrate that lightweight, scalable web technologies can provide an effective solution for modernizing academic communication systems in educational institutions. The system was evaluated through functional and usability testing involving student and faculty users. Experimental results demonstrate reduced information retrieval time and improved communication efficiency.**

***Index Terms*—Academic Management System, Web Application, Flask, Role-Based Access Control, Educational Technology**

I. INTRODUCTION

Educational institutions generate large volumes of academic information, including notices, assignments, timetables, and learning resources. However, these resources are often distributed across fragmented communication channels such as physical notice boards, emails, and messaging platforms, leading to missed updates, inefficient coordination, and increased administrative workload. The rapid growth of digital communication technologies has transformed information sharing in many sectors, yet academic institutions often rely on fragmented communication channels such as notice boards, messaging groups, and emails. This decentralized approach leads to information asymmetry, missed deadlines, and increased workload for faculty and administrators.

Students frequently need to consult multiple sources to obtain schedules, assignments, and announcements. Faculty members often duplicate efforts by sharing the same resources across different platforms. These inefficiencies highlight the need for a centralized academic information management system.

CampusMate is proposed as a web-based platform that integrates academic notices, assignments, notes, timetables, and events into a single interface. The system aims to simplify access to academic information and improve communication efficiency.

II. MOTIVATION

The motivation for developing CampusMate arises from the increasing demand for efficient academic communication systems. In many institutions, students struggle to track multiple deadlines, locate relevant resources, and verify information authenticity. Faculty members also experience difficulties in managing large volumes of academic content.

Research in assistive and communication technologies has demonstrated the importance of digital platforms in bridging communication gaps and improving accessibility.

Similarly, academic institutions require centralized systems that can:

- Reduce redundancy in communication
- Improve access to learning materials
- Enhance administrative efficiency
- Provide real-time updates

CampusMate aims to address these needs through a centralized web-based solution.

III. OBJECTIVES

The objectives of the proposed system are:

- To design a centralized platform for academic communication.
- To provide role-based access for students, faculty, and administrators.
- To enable efficient uploading, filtering, and retrieval of academic resources.
- To automate maintenance tasks such as deletion of outdated data.
- To ensure usability, responsiveness, and reliability of the system.

IV. METHODOLOGY

The methodology of the proposed CampusMate system focuses on the design and implementation of a centralized academic information management platform that enables efficient data input, processing, storage, and retrieval. The development process follows a structured workflow to ensure system reliability, usability, and security.

The working methodology is divided into four major phases:

A. Data Acquisition

B. Data Processing and Validation

C. Data Storage and Management

D. Information Retrieval and Presentation

These phases ensure seamless communication between students, faculty, and administrators.

A. Data Acquisition

The first stage involves collecting academic data and user inputs. Faculty and administrators upload academic resources such as notices, assignments, notes, timetables, and event details through authenticated dashboards. Each upload includes metadata such as title, description, semester, and division.

Students interact with the system by selecting filtering parameters such as semester and division to retrieve relevant academic information. This approach reduces information overload and ensures that users view only relevant content.

B. Data Processing and Validation

Once data is received, the application layer processes and validates the inputs before storing them in the database. Validation mechanisms ensure:

- Only supported file formats are accepted
- Input fields are sanitized to prevent malicious data entry
- Metadata is verified for correctness

Timestamping mechanisms are implemented to record upload times accurately and ensure consistent deadline tracking.

Additionally, the system assigns unique filenames to uploaded files to prevent overwriting and maintain file integrity.

C. Data Storage and Management

After validation, data is stored in two forms:

- StructuredData--Metadata such as user details, file information, and timestamps are stored in relational database tables.
- FileStorage-- Uploaded documents are stored securely in the server's file system.

This separation of metadata and file storage improves system efficiency and simplifies data retrieval.

To maintain system performance, an automated cleanup mechanism periodically removes outdated notices, assignments, and resources based on predefined time intervals.

D. Information Retrieval and Presentation

When a user requests information, the system retrieves relevant records from the database based on filtering criteria and user roles. The application dynamically generates dashboards and renders content using server-side templates.

Students receive read-only access to academic materials, while faculty and administrators are provided with content management capabilities. Role-based filtering ensures secure and controlled access to information.

Files are delivered through secure routes that determine whether content should be displayed in the browser or downloaded, depending on file type.

Architectural Workflow

Dataflow of CampusMate :

- Users interact with the web interface.
- Requests are sent to the Flask server.
- The server processes requests and interacts with the database.
- Data is retrieved and rendered dynamically to dashboards.

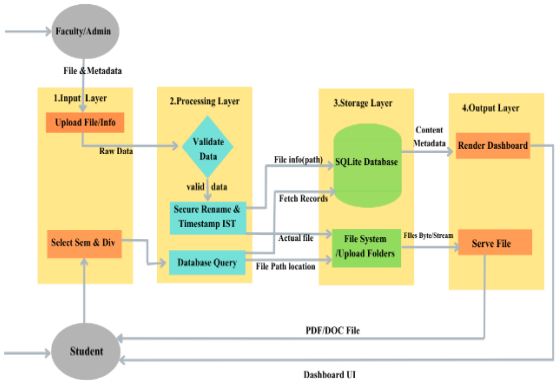

Fig1. Dataflow diagram

V. ARCHITECTURE

The architecture of the proposed CampusMate system is designed to ensure scalability, maintainability, and efficient handling of academic data. The system follows a three-tier Model–View–Controller (MVC) architecture, which separates application logic, user interface, and data management into independent modules. This separation improves system reliability and simplifies future enhancements.

The overall architecture consists of three primary layers:

A. Presentation Layer
B. Application Layer
C. Data Layer

These layers interact to process user requests, store data securely, and deliver information efficiently.

A. Presentation Layer (User Interface)

The presentation layer is responsible for interacting directly with users and displaying academic information in a structured and responsive manner. This layer is implemented using HTML5, CSS3, Bootstrap, and JavaScript to ensure cross-platform compatibility and mobile responsiveness.

The interface provides different dashboards based on user roles:

- Student Dashboard: Students can access notices, assignments, notes, timetables, and events without login by selecting their semester and division.
- Faculty Dashboard: Faculty members can upload academic resources, manage content, and update assignments and deadlines.
- Administrator Dashboard: Administrators can manage user accounts, monitor system activities, and maintain overall system integrity.

The design emphasizes simplicity, accessibility, and quick navigation to improve user experience.

B. Application Layer

The application layer acts as the core processing unit of the system and is implemented using the Flask web framework. This layer handles request processing, authentication, validation, and business logic operations.

Key responsibilities of this layer include:

- Request Handling
  Incoming HTTP requests from users are processed, validated, and routed to appropriate modules.
- Authentication and Authorization
  Faculty and administrators are authenticated using secure login mechanisms, while role-based access control ensures that users can only access permitted resources.
- Data Validation
  Uploaded files are validated for format and security before being stored in the system.
- Automated Maintenance
  The system periodically removes outdated notices, notes, and assignments to maintain database performance and ensure relevance of displayed content.

This layer ensures smooth coordination between the user interface and database.

C. Data Layer (Database and Storage)

The data layer is responsible for storing and retrieving structured and unstructured data. It consists of:

- Relational Database

User credentials, file metadata, timestamps, and role information are stored in relational database tables using SQLAlchemy ORM.

- File Storage System
  Uploaded documents such as PDFs, images, and notes are stored securely in the server file system.
- Data Integrity and Security
  Password hashing, secure file naming, and controlled access mechanisms are implemented to protect sensitive data and prevent unauthorized access.

The separation of metadata and file storage improves performance and simplifies system maintenance.

Model–View–Controller (MVC)

The CampusMate system follows the Model–View–Controller (MVC) architecture, which separates the application into three components: Model, View, and Controller.

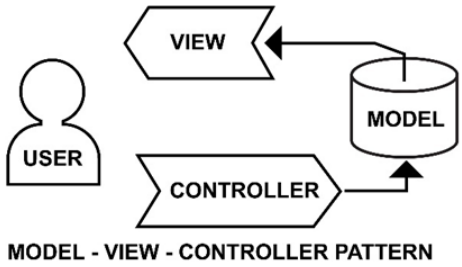

Fig 2. MVC architecture

The Model manages data and database operations, the View handles the user interface and presentation, and the Controller processes user requests and coordinates communication between the Model and View. This architecture improves modularity, maintainability, and scalability of the system.

VI. LITERATURE OVERVIEW

- Recent research has explored digital academic management systems to improve communication, accessibility, and structured information dissemination in higher education institutions. Several campus companion applications have been proposed to centralize academic schedules, assignments, results, and learning resources, aiming to reduce dependency on traditional notice boards and informal communication channels.
- VESP (2024) introduced a centralized mobile-based academic platform integrating schedules, attendance tracking, fee management, and learning resources. The system demonstrated improved communication efficiency and student engagement through real-time synchronization using Firebase. However, the platform primarily focuses on mobile access and requires mandatory user authentication, which may limit quick access to basic academic information.
- The Companion App (2023) presented a student-centric mobile application that enables assignment tracking, academic updates, and push notifications. While the inclusion of real-time updates enhances responsiveness, the system depends heavily on mobile infrastructure and continuous authentication. Additionally, the platform does not incorporate automated lifecycle management for outdated notices or content.
- The College Companion App (2022) proposed a unified digital space for accessing class notes, exam results, and academic announcements. The study highlighted the reduction in communication redundancy and improved access to institutional resources. However, scalability concerns and limited modular architectural design were observed, particularly in managing multi-role user hierarchies.
- Student Companion (STU-COM) (2019) implemented an Android-based academic management solution with modules for attendance, notices, and results. Although it effectively digitized manual academic processes, the system was restricted to mobile platforms and lacked advanced role-based access control mechanisms and automated data maintenance strategies.
- Further studies in digital notice board systems emphasize structured web portals for academic communication. These systems improve transparency and centralized data storage but often require authentication even for read-only access, creating potential friction for students who need quick information retrieval. Moreover, many existing platforms do not clearly separate metadata from file storage, affecting long-term maintainability and system performance.
- Research on Role-Based Access Control (RBAC) in educational information systems demonstrates the importance of structured permission hierarchies in maintaining system security and data integrity. Proper implementation of RBAC enhances administrative efficiency and prevents

unauthorized access. However, integration of RBAC with lightweight web architectures in small-to-medium institutions remains underexplored.

Despite these advancements, several gaps remain in current academic management solutions:

- Most systems prioritize mobile applications over lightweight web-based platforms.
- Mandatory authentication is required even for simple content viewing.
- Limited focus is given to automated content cleanup and lifecycle management.
- Filtering mechanisms are often static rather than dynamically driven by academic attributes such as semester and division
- The proposed CampusMate system addresses these gaps by providing a centralized, web-based academic information platform with role-based access control, simplified no-login access for students, structured metadata separation, and automated removal of outdated content. By combining modular MVC architecture with efficient filtering and maintenance mechanisms, CampusMate contributes a scalable and user-centric solution to modern academic communication challenges.

VII. RESULTS AND DISCUSSION

The CampusMate system was evaluated through functional, integration, and system-level testing to verify performance, usability, and reliability. Testing confirmed that the major system modules, including user authentication, file upload, role-based access control, and content filtering, operated correctly under expected conditions.

File validation and upload mechanisms successfully restricted unsupported file formats and ensured secure storage of academic resources. Filtering features accurately displayed content based on selected semester and division, demonstrating effective data isolation. Automated maintenance functions were also tested and verified to remove outdated records as intended, maintaining database efficiency.

User testing indicated that students were able to locate academic materials quickly and navigate the dashboard with minimal difficulty. The responsive interface functioned effectively across different browsers and devices, confirming the system's usability and accessibility.

Overall, the experimental results demonstrate that the proposed system improves information accessibility, reduces administrative redundancy, and provides a reliable platform for academic communication.

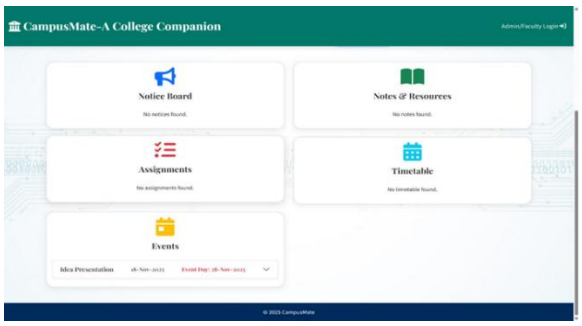

Fig 3. Result (Student Dashboard)

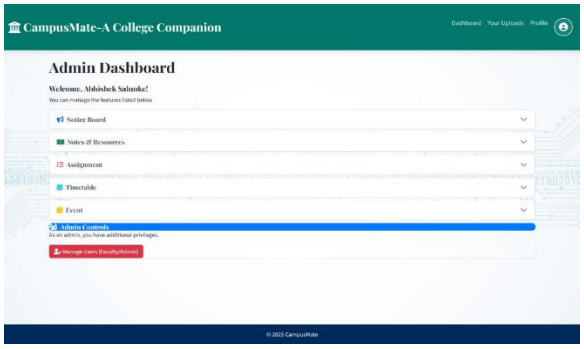

Fig 4: Result (Admin Dashboard)

VIII. CONCLUSION

CampusMate, a centralized academic information management system designed to address the problem of fragmented communication in educational institutions. The system integrates notices, assignments, schedules, and academic resources into a single web-based platform, improving accessibility and efficiency for both students and faculty.

The implementation of role-based access control, automated data maintenance, and responsive interface design ensures secure operation, efficient resource management, and improved user experience. Testing results confirmed that the system meets functional and usability requirements and provides a scalable solution for academic environments.

Future work may include mobile application support, real-time notification services, and cloud-based deployment to further enhance scalability and system performance.

REFERENCES

[1] Vesp: The Ultimate College Companion App — Yash Balotiya, Devang Deokule, Shreya Shukla, Prof. Vidya Lunge, Prof. Meena Talele — IJRPR, Vol. 5, No. 2, Feb 2024.

[2] A Companion Application for Students: Companion App — Ankita Guha, Vanshika Singh, Dr. Anuj Kumar Singh — YMER Journal, Vol. 22, Issue 4, Apr 2023.

[3] A Survey on Campus Companion — Mrs. G. Ramanjinamma — IJRASET, 2023.

[4] The Impact of Digital Deadline Trackers on Student Academic Performance and Stress Levels — T. Lee, J. Kim — Computers & Education, Vol. 145, Jan 2023.

[5] Automating Administrative Workflows in Higher Education Using Python — B. White, D. Clark — Journal of Educational Computing Research, Oct 2022.

[6] College Companion App — M. Sumithra, B. Buvaneswari, T. Vignesh Raj, K. Siddharth, V. Vishnuprasad — Irjet, Vol. 9, Issue 7, Jul 2022.

[7] User Experience (Ux) Design Principles for Student-Centric Dashboards — L. Chen, Y. Zhang — Hci International Conference Proceedings, Sep 2021.

[8] E-Learning and Digital Notice Boards in Universities: A Framework for Paperless Communication — A. K. Sharma, P. Gupta — Ieee Access, Vol. 9, Aug 2021.

[9] GALGOBOT – THE COLLEGE COMPANION CHATBOT — Saksham Saraswat, Siddhartha Mishra, Vikas Mani, Shristi Priya — Proceedings Of ICICCS, 2021.

[10] Role-Based Access Control (Rbac) In Educational Information Systems — M. Johnson, L. Lee — Journal of Systems and Software (Elsevier), Nov 2020.

[11] Implementing Responsive Web Design for Educational Portals — N. Rao, K. V. S. Murthy — IJEAT, Feb 2020.

[12] Student Companion (Stu-Com) — Sajil M. S., Arunsankar V., Dinesh Babu B. — IJERT, 2019.

[13] Student Companion (Stu-Com): A Mobile-Based Solution for Academic Management — R. S. Pressman, T. Maxim — IEEE International Conference on Advanced Learning Technologies, May 2019.

[14] Flask Web Development: Developing Web Applications with Python — Miguel Grinberg — O'reilly Media (Book), Mar 2018.