

Campusphere: Smart Academic Management System

- 1) **Aditya Pawar** (Asokrao Mane Group of Institutions, Vathar tarf vadagaon) (CSE)
- 2) **Ritesh Mali** (Asokrao Mane Group of Institutions, Vathar tarf vadagaon) (CSE)
- 3) **Sandesh Kolage** (Asokrao Mane Group of Institutions, Vathar tarf vadagaon) (CSE)
- 4) **Prathmesh Patil** (Asokrao Mane Group of Institutions, Vathar tarf vadagaon) (CSE)
- 5) **Sahil Sawant** (Asokrao Mane Group of Institutions, Vathar tarf vadagaon) (CSE)

ABSTRACT

Campus Sphere is a web-based academic management system designed to improve coordination between students, teachers, and administrators in educational institutions. The system integrates multiple academic and skill development services into a single platform, enabling efficient handling of learning, evaluation, and administrative tasks. The student panel provides access to study notes, mock interviews, live interview sessions, an integrated multi-language code compiler, attendance records, and detailed progress reports to support continuous learning and skill enhancement. The teacher panel allows faculty members to upload notes, manage attendance, conduct mock and live interviews, and monitor student performance through structured evaluation tools. The admin panel ensures smooth system operation by managing student and faculty accounts, maintaining academic records, and overseeing overall platform activities. With role-based access control and centralized data management, Campus Sphere enhances transparency and reduces manual workload. The system promotes interactive learning, improves placement preparation, and strengthens academic tracking. By combining academic resources, evaluation tools, and administrative control, Campus Sphere creates an efficient digital ecosystem for modern education management.

1. INTRODUCTION

In modern educational environments, the need for an integrated and efficient academic management system has become increasingly important due to the growing complexity of student activities, faculty responsibilities, and administrative operations. Traditional methods of managing attendance, academic content, evaluations, and communication often lead to delays, errors, and lack of transparency. To address these challenges, the proposed system, **Campus Sphere**, is designed as a centralized web-based platform that connects students, teachers, and administrators through a unified digital interface.

Campus Sphere is structured into three primary modules: Student Panel, Teacher Panel, and Admin Panel. The **Student Panel** provides access to study notes, attendance records, mock interviews, live interview sessions, an integrated multi-language compiler, and progress tracking features, enabling students to enhance both academic knowledge and technical skills. The **Teacher Panel** facilitates efficient academic management by allowing faculty to upload notes, conduct attendance marking, organize mock and live interviews, and evaluate student performance through systematic progress monitoring tools. The **Admin Panel** ensures overall system governance by managing student and faculty records, controlling user access, and maintaining institutional data integrity.

The system incorporates key concepts such as role-based access control, real-time data management, and centralized academic tracking. It also supports placement preparation through interview simulations and coding practice modules. By integrating learning resources, evaluation systems, and administrative operations into a single platform, Campus Sphere improves communication, reduces manual workload, and enhances transparency in academic processes. This research aims to develop a scalable and efficient solution for modern educational institutions.

2. METHODOLOGY

Campus Sphere is developed as a structured web-based academic management system that integrates student, teacher, and admin operations into a unified platform. The methodology begins with requirement analysis, where academic needs such as attendance tracking, note sharing, interview practice, coding support, and performance evaluation are identified. A role-based authentication system is implemented to ensure secure and restricted access for different users. The system architecture follows a modular design, separating student, teacher, and admin functionalities for better scalability and maintenance. A centralized database is used to store academic records, attendance data, interview results, and progress reports in real time. The development process includes frontend design for user interaction and backend development for data processing and logic handling. Testing is performed to ensure accuracy, security, and smooth performance across all modules. Continuous validation and optimization are applied to improve system efficiency, responsiveness, and usability for academic environments.

The methodology of Campus Sphere is designed to build a centralized and efficient academic management system by integrating student, teacher, and admin functionalities through a structured web-based architecture.

User Authentication and Role-Based Access Control

The system begins with secure login authentication, where users are classified into student, teacher, and admin roles. Each role is granted specific permissions to ensure data security, privacy, and controlled access to academic resources.

Academic Content and Activity Management

Teachers upload notes, manage attendance, and conduct mock and live interviews through the platform. Students can access study materials, attend interviews, use the integrated multi-language compiler, and track their academic progress in real time.

Performance Tracking and Evaluation

The system continuously records student performance through attendance data, interview results, and progress reports. This data is processed to generate analytical insights for both students and faculty.

System Administration and Data Management

Admins oversee user management, including adding or removing students and faculty, maintaining records, and ensuring smooth system operation. Centralized database management ensures consistency, scalability, and efficient retrieval of academic data across all modules.

AI-Powered Learning and Interview Assistance

The system integrates AI-based modules to enhance student learning and placement preparation. The AI chatbot provides instant academic assistance and answers student queries. The automated mock interview system generates department-specific interview questions, evaluates responses based on grammar, confidence, and relevance, and provides detailed performance feedback for improvement.

AI-Powered Learning and Interview Assistance

The system integrates AI-based modules to enhance student learning and placement preparation. The AI chatbot provides instant academic assistance and answers student queries. The automated mock interview system generates

department-specific interview questions, evaluates responses based on grammar, confidence, and relevance, and provides detailed performance feedback for improvement.

Timetable and Academic Scheduling Management

The system provides centralized timetable management where faculty and administrators can upload and update class schedules based on branch and semester. Students can view and download timetables anytime, ensuring organized academic planning and reducing scheduling conflicts between lectures, practical sessions, and examinations

Real-Time Communication and Notification System

The platform includes a real-time communication mechanism using Socket.io and email services. Notifications related to exams, attendance, events, placement activities, and notices are instantly delivered to users. This improves communication efficiency and ensures that students and faculty remain updated with important academic information.

Placement and Career Development Module

The placement module helps students prepare for campus recruitment activities by providing company information, aptitude quizzes, technical tests, and HR interview practice. Students can track their quiz performance and improve weak areas, while administrators can manage placement company records and question databases effectively.

3. MODELING AND ANALYSIS

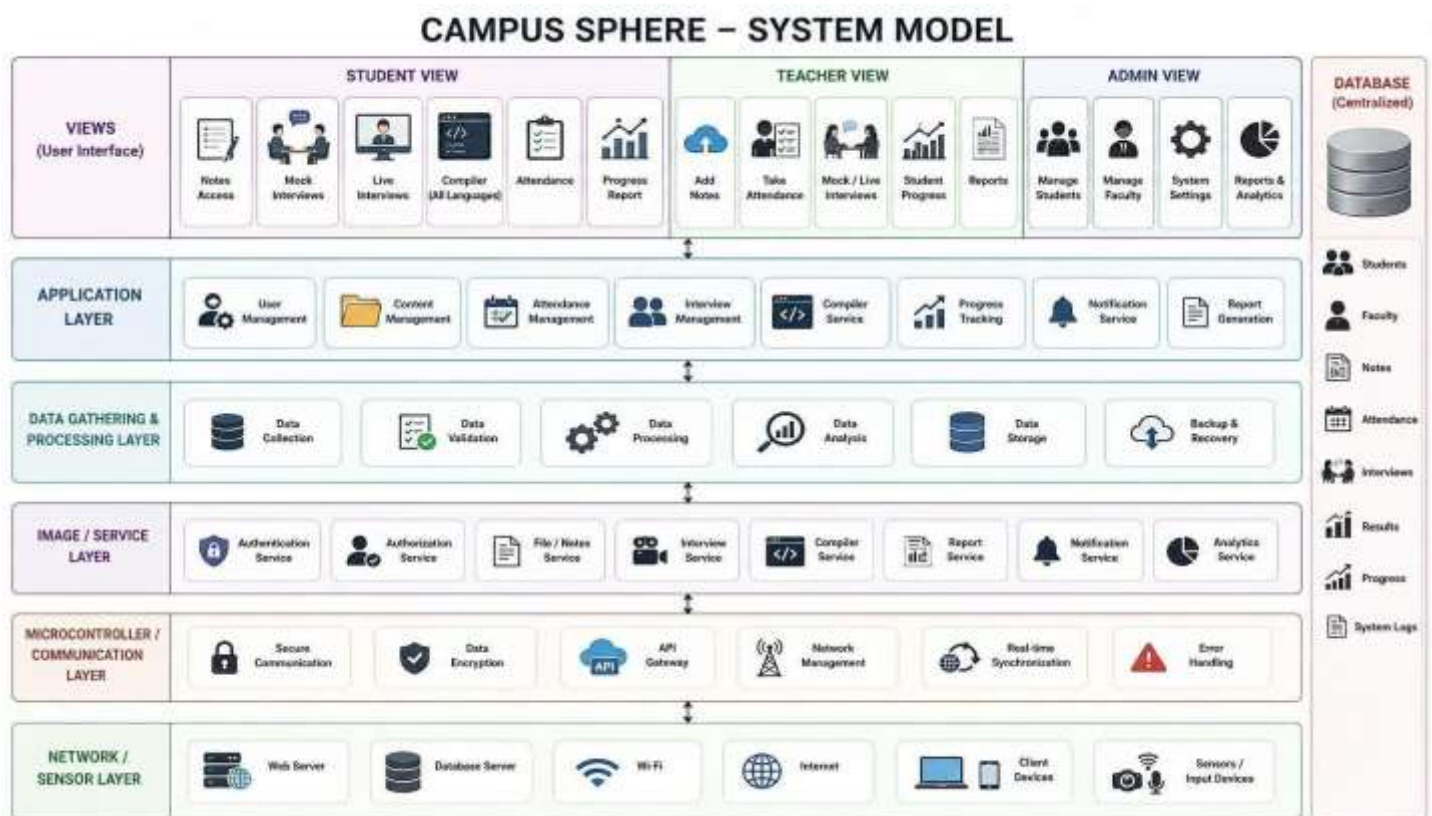


Figure 1: Campus Sphere-system Model

Modeling and analysis in the Campus Sphere system involve designing and evaluating the functionality of different modules such as the student panel, teacher panel, and admin panel. The system architecture is modeled to ensure smooth communication between users, databases, and application services. Various features including attendance management, notes sharing, mock interviews, live interview sessions, progress tracking, and online compiler integration are analyzed for performance, reliability, and usability. Data flow diagrams and module structures are used to represent system operations and user interactions. The analysis process helps identify system requirements, improve security, reduce processing time, and enhance user experience. Through proper modeling and analysis, the Campus Sphere platform ensures efficient campus management and seamless digital interaction within educational institutions.

3.1 UML Diagrams

- Use Case Diagram :

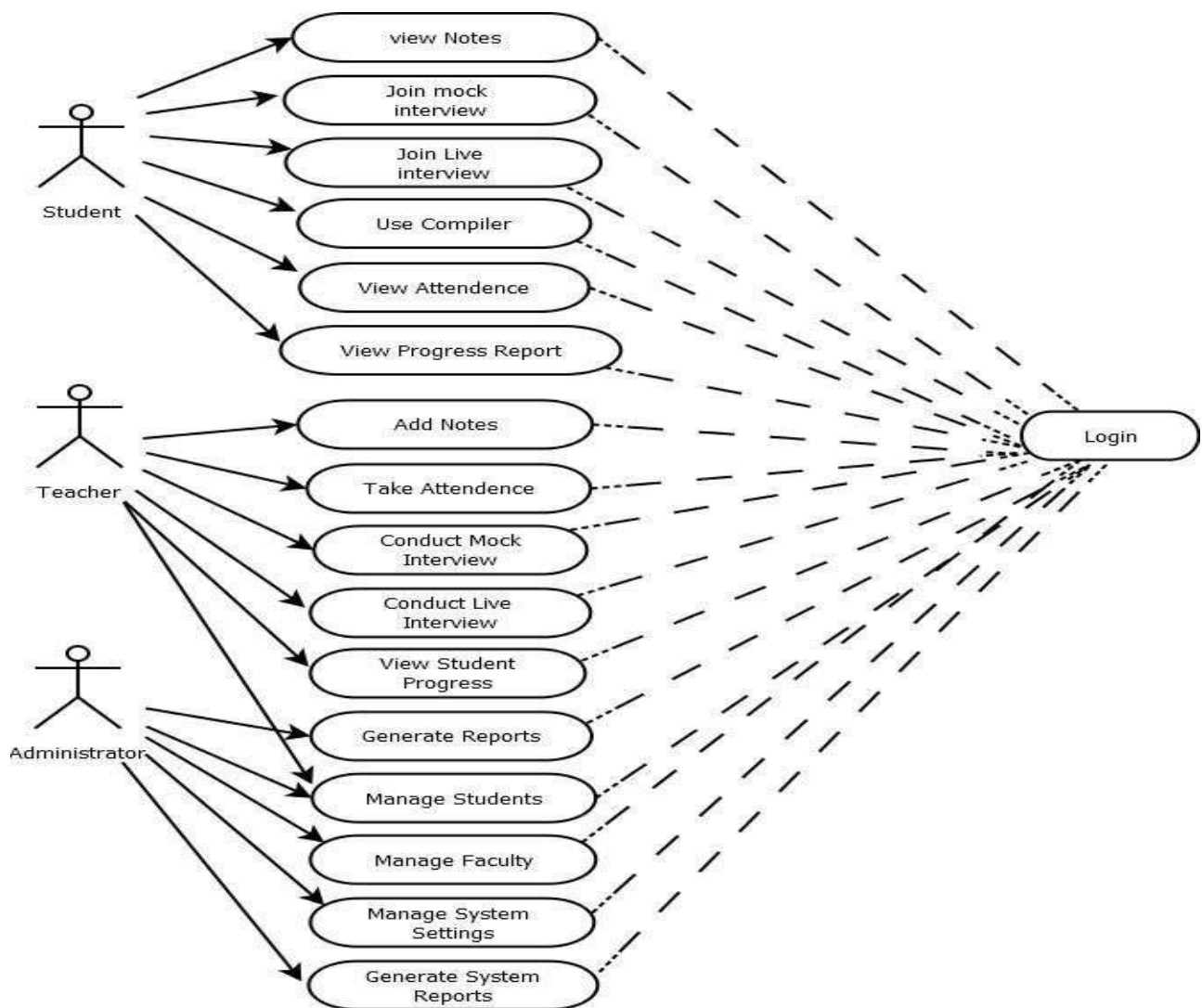


Figure 2: Use case diagram

Use Case Diagram

The Use Case Diagram represents the functional behavior of the College Management System by showing interactions between users and the system. The main actors are Admin, Faculty, and Student. Admin manages users, branches, timetables, notices, placements, and approvals. Faculty uploads notes, takes attendance, creates exams, manages feedback, and adds marks. Students access materials, attempt exams, view attendance, request certificates, and

participate in placements and events. External services such as AI chatbot, payment gateway, and email notification system are also connected. This diagram helps in understanding system functionalities and user responsibilities clearly

• **Class diagram :**

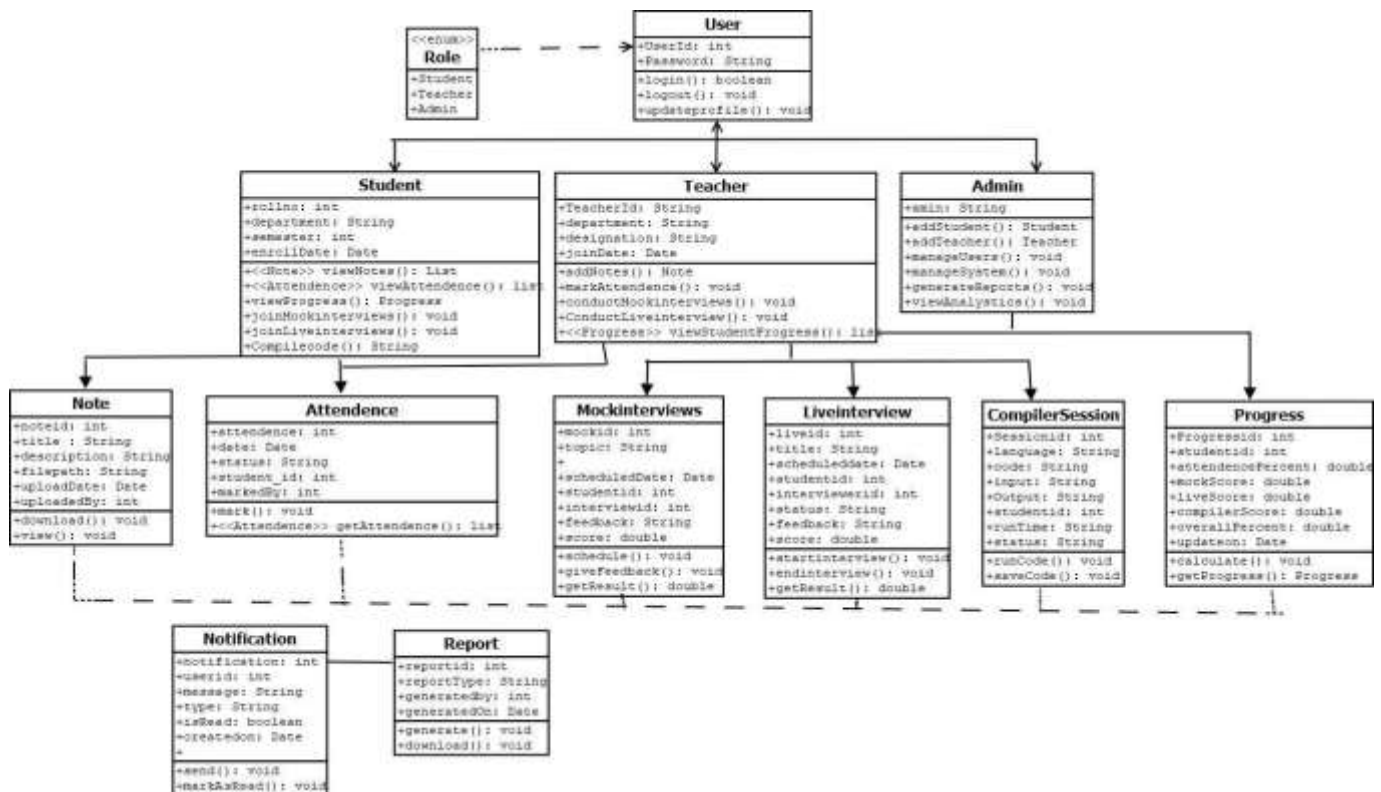


Figure 3: The Class Diagram

The class diagram of Campus Sphere represents the structural relationship between major system components such as Student, Teacher, Admin, Notes, Attendance, Interview, Compiler, and Progress Report. Students access academic resources and performance records, teachers manage academic activities, and admins control system operations. The diagram helps visualize data flow, object interaction, and system organization. The Class Diagram represents the static structure of the College Management System. It contains a base User class with common attributes like userId, name, email, password, and role. Admin, Faculty, and Student inherit from the User class. Admin manages system configuration such as users, branches, and timetables. Faculty manages academic activities like attendance, notes, exams, and marks. Student interacts with learning modules, exams, and feedback. Supporting classes include Subject, Timetable, Exam, Attendance, Feedback, and PlacementCompany. Relationships such as inheritance, association, and aggregation define how objects interact. It shows how real-world entities are organized in an object-oriented structure.

Sequence Diagram :

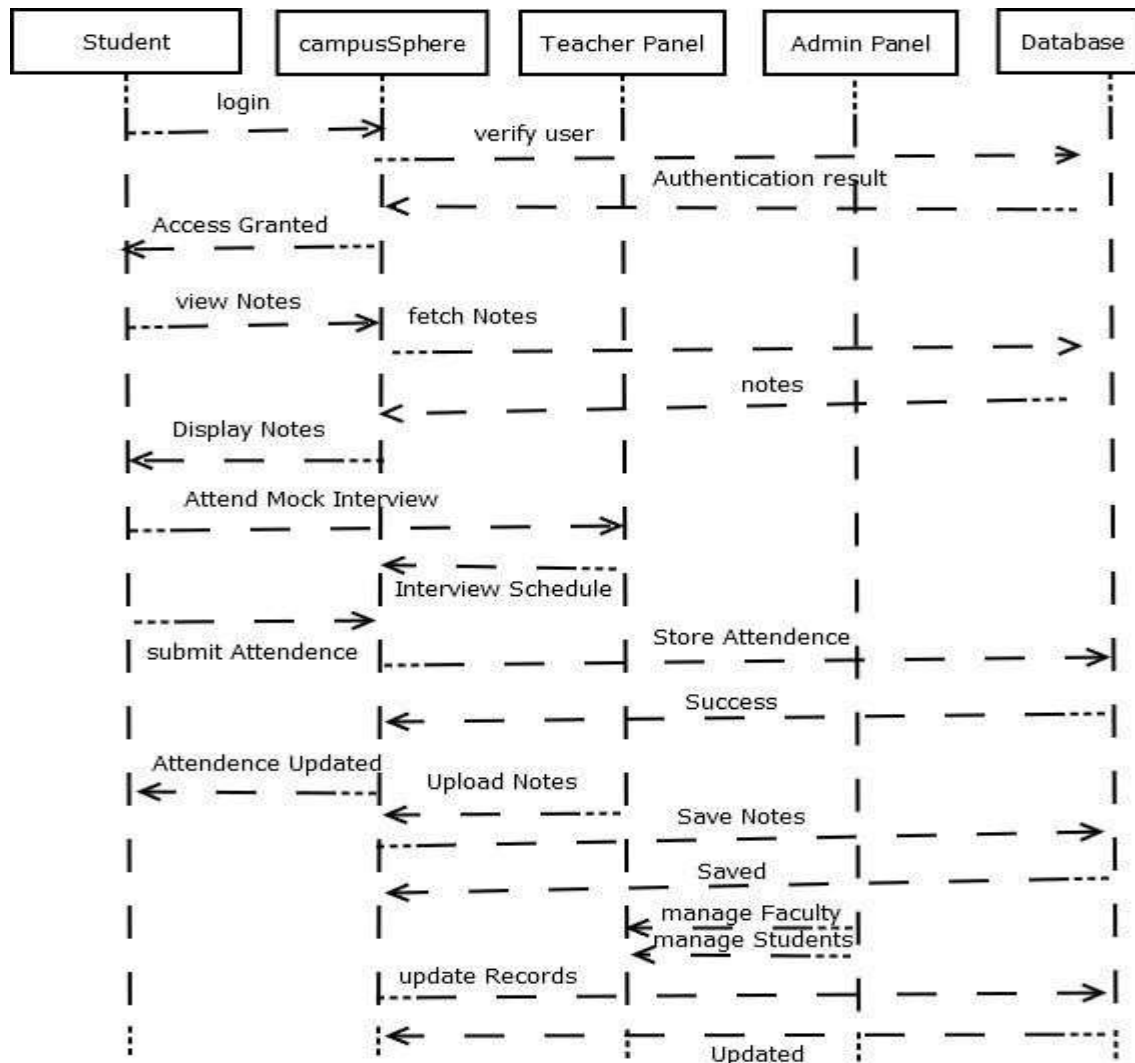


Figure 4: The sequence Diagram

The sequence diagram of CampusSphere shows how different users interact with the system over time, including the Student, Teacher, Admin, and Database. It explains the step-by-step flow of actions such as login authentication, accessing study materials, attending interviews, and managing attendance. All actions are processed through the central system and stored in the database, ensuring real-time updates and smooth coordination between all modules. The diagram highlights how CampusSphere automates academic tasks and improves efficiency, communication, and data management in an educational environment. The Sequence Diagram explains the step-by-step interaction between users and the system. For example, in student login, the student enters credentials which are verified by the backend through the database. If valid, a JWT token is generated and the student is redirected to the dashboard. When accessing notes, the request goes from frontend to backend, which fetches data from the database and returns it to the student. In MCQ exams, questions are loaded, answers are submitted, evaluated, and scores stored. AI-based mock interviews follow API calls to generate questions and evaluate answers. It shows time-ordered system communication flow.

• Activity diagram

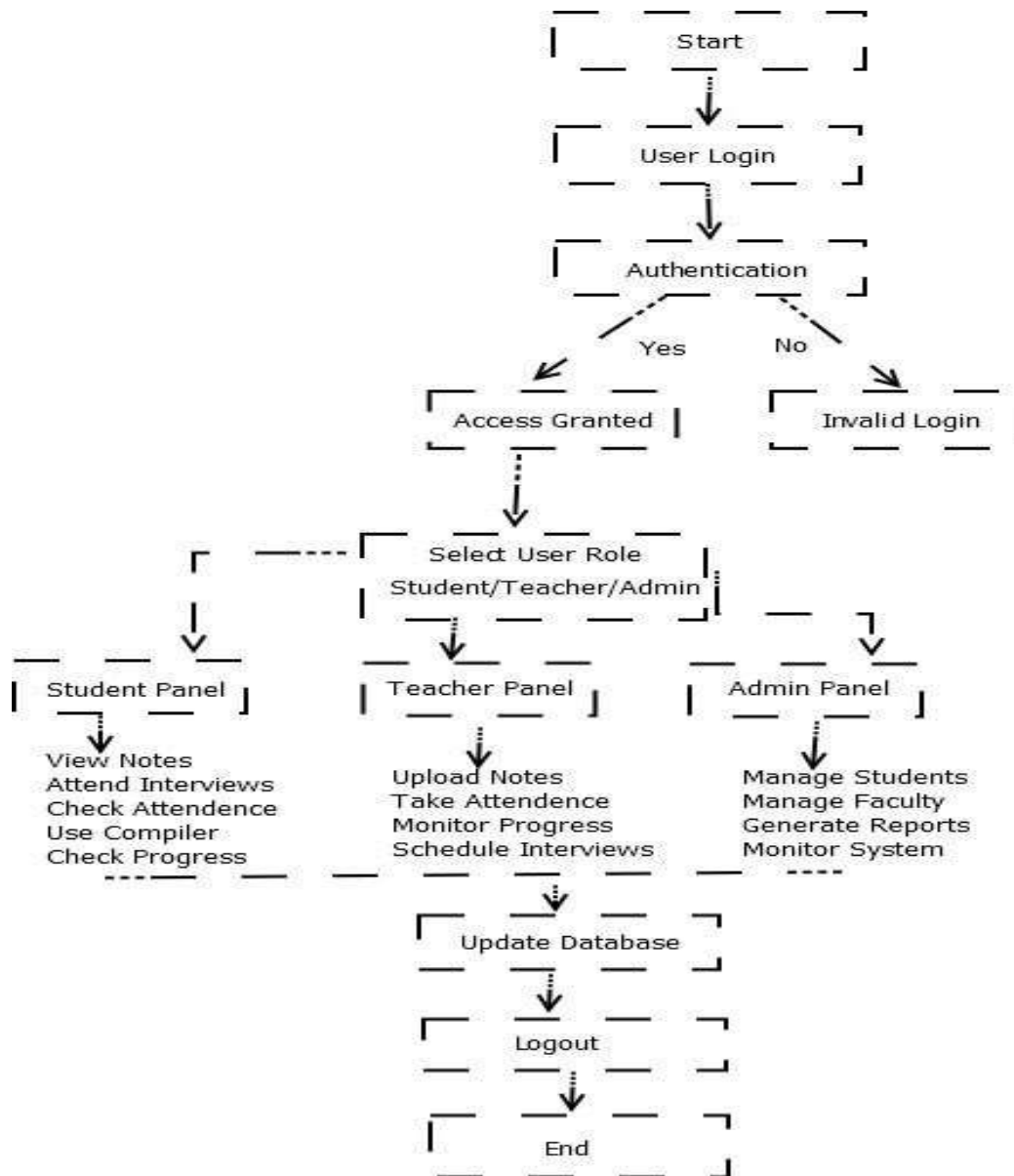


Figure 5: The Activity Diagram

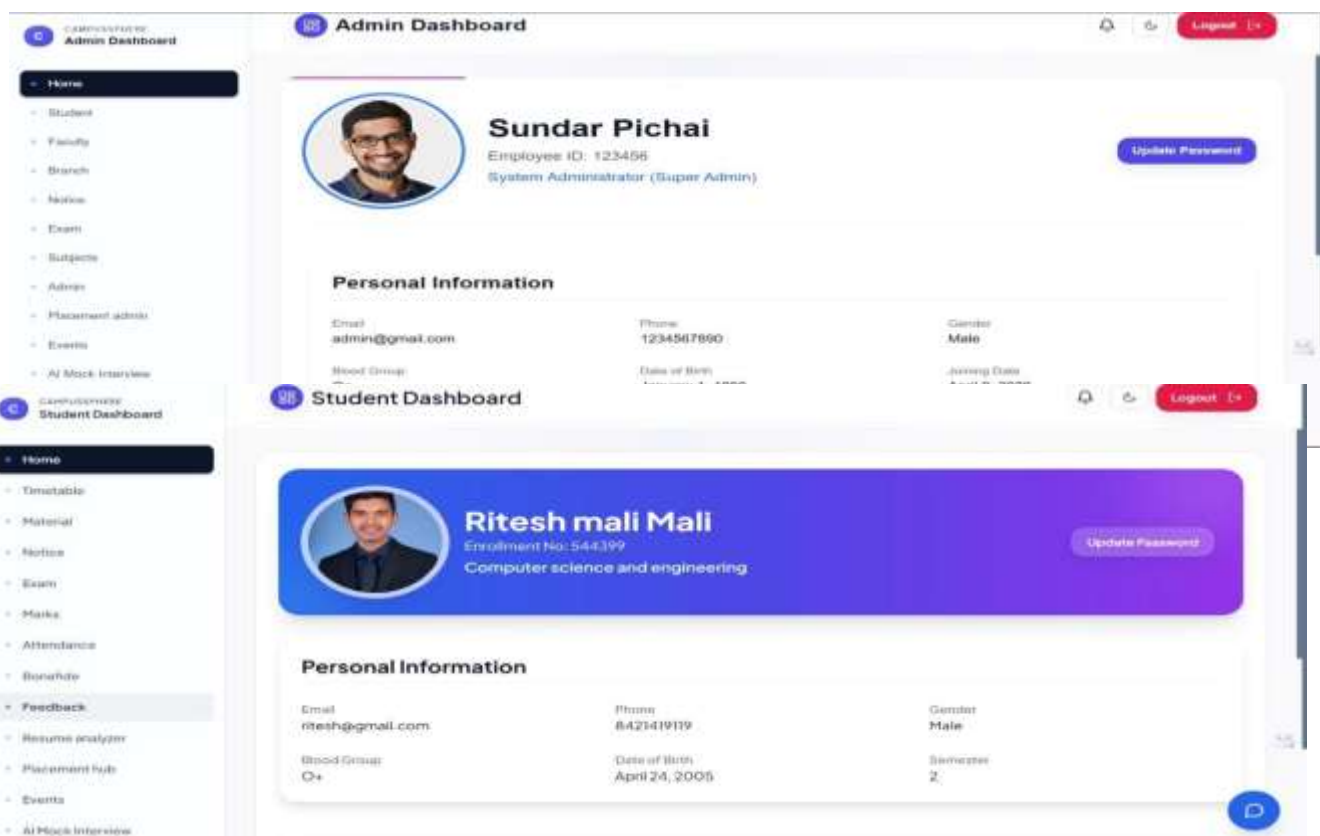
An activity diagram for a smart vehicle system illustrates the flow of operations and decision-making processes involved in real-time vehicle functioning. The diagram begins with the **vehicle startup**, followed by **initial sensor checks**, where IoT sensors like GPS, temperature, and proximity activate and start collecting data. The system then proceeds to **data processing**, where the vehicle controller analyzes sensor inputs. Based on the analysis, the system performs actions such as **obstacle detection**, **speed adjustment**, or **alert generation**. If abnormal behavior or danger is detected, the system may trigger **emergency responses** like braking or notifications to the driver. The data is also sent to the **cloud for logging and remote access**. The process loops continuously during vehicle operation, ensuring dynamic response and safety. This activity diagram highlights the sequential and conditional flow of tasks in a smart, sensor-enabled vehicle environment.

4. RESULTS AND DISCUSSION

The implementation of CampusSphere demonstrated a significant improvement in the management of academic activities by integrating Student, Teacher, and Admin panels into a single unified platform. The system successfully enabled real-time interaction between users and ensured smooth execution of key functions such as notes distribution, attendance tracking, mock interviews, live interview sessions, and online coding practice. During testing, the student module performed efficiently, allowing quick access to study materials, accurate attendance updates, and smooth execution of the online compiler for multiple programming languages. The mock interview and live interview features worked effectively, helping students enhance their communication and technical skills. The progress tracking system provided meaningful insights into student performance, enabling better academic planning. The teacher panel also showed reliable performance, particularly in uploading notes, managing attendance, and monitoring student progress. Data synchronization between teacher inputs and student access was consistent and accurate. The admin panel ensured proper management of users and system data, maintaining overall system stability and control.

However, minor delays were observed during peak usage when multiple users accessed the system simultaneously, indicating a need for better optimization and scalability improvements. Additionally, integrating advanced caching mechanisms and cloud-based load balancing could further enhance system responsiveness. Overall, CampusSphere performed effectively as a smart academic management solution, improving efficiency, communication, and digital learning experience. The results confirm that such integrated systems can significantly modernize educational institutions by reducing manual workload and improving accessibility to academic resources.

Output :





5. CONCLUSION

The CampusSphere project successfully integrates student, teacher, and admin functionalities into a single digital academic platform, improving communication, learning, and administrative efficiency. It reduces manual workload by automating key academic processes like attendance, notes distribution, performance tracking, and interview management. The system enhances learning through features like an online compiler, mock interviews, and progress reports, helping students develop technical and professional skills. Teachers benefit from simplified content management and attendance tracking, while administrators ensure smooth system control and data handling. Overall, CampusSphere promotes a smart and centralized educational environment that improves accessibility, accuracy, and productivity in academic institutions.

Key Takeaways:

1. Centralized system improves academic management efficiency.
2. Automation reduces manual work in attendance, notes, and reports.
3. Student skill development is enhanced through interviews and coding tools.
4. System improves coordination between students, teachers, and admin.

6. REFERENCES

- [1] A. Alammary, S. Sheard and A. Carbone, "Blended learning in higher education: Three different design approaches", Australasian Journal of Educational Technology, vol. 30, no. 4, pp. 440-454, 2014. View Article Google Scholar
- [2] M. M. Rahman and Y. Watanobe, "Development of an intelligent educational platform using artificial intelligence", IEEE Access, vol. 9, pp. 102256-102270, 2021. View Article Google Scholar
- [3] S. Iqbal, M. Z. Khan and A. Malik, "Smart education management system using cloud computing", International Journal of Advanced Computer Science and Applications, vol. 10, no. 6, pp. 215-221, 2019. View Article Google Scholar
- [4] R. K. Sharma and P. Gupta, "Web based student information management system", International Journal of Computer Applications, vol. 97, no. 12, pp. 30-35, 2014. View Article Google Scholar

- [5] J. Singh and R. Kaur, "Online examination and student monitoring system", *Procedia Computer Science*, vol. 125, pp. 307-314, 2018. View Article Google Scholar
- [6] H. F. Atlam and G. B. Wills, "IoT and cloud computing for smart education systems", *Journal of Sensor and Actuator Networks*, vol. 9, no. 3, pp. 1-15, 2020. View Article Google Scholar
- [7] T. Anderson and J. Dron, "Three generations of distance education pedagogy", *International Review of Research in Open and Distributed Learning*, vol. 12, no. 3, pp. 80-97, 2011. View Article Google Scholar
- [8] P. Kumar and S. Bansal, "Automation in educational institutions using management information systems", *International Journal of Engineering Research and Technology*, vol. 6, no. 5, pp. 456-460, 2017. View Article Google Scholar
- [9] N. S. Vaidya and M. Shah, "Digital learning system for higher education", *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, pp. 1256-1261, 2019. View Article Google Scholar
- [10] K. Becker, "The impact of technology on student engagement and learning", *Education and Information Technologies*, vol. 25, no. 3, pp. 1763-1779, 2020. View Article Google Scholar
- [11] A. Al-Fraihat, M. Joy and J. Sinclair, "Evaluating e-learning systems success", *Computers in Human Behavior*, vol. 102, pp. 67-86, 2020. View Article Google Scholar
- [12] S. Goyal and R. Tambe, "Role of online learning platforms in modern education", *International Journal of Scientific Research in Computer Science*, vol. 5, no. 2, pp. 45-51, 2020. View Article Google Scholar
- [13] D. Zhang, J. L. Zhao, L. Zhou and J. F. Nunamaker, "Can e-learning replace classroom learning?", *Communications of the ACM*, vol. 47, no. 5, pp. 75-79, 2004. View Article Google Scholar
- [14] M. Ali, A. H. Kazi and S. Khan, "Cloud integrated smart campus management framework", *IEEE International Conference on Smart Computing*, pp. 120-126, 2021. View Article Google Scholar
- [15] R. Patel and N. Mehta, "Student performance analysis using machine learning techniques", *International Journal of Computer Science and Information Security*, vol. 17, no. 4, pp. 98-104, 2019. View Article Google Scholar
- [16] V. Kumar and P. Singh, "Web based attendance management system using cloud database", *International Journal of Engineering and Advanced Technology*, vol. 9, no. 1, pp. 2301-2306, 2019. View Article Google Scholar
- [17] S. B. Kotsiantis, "Use of machine learning techniques for educational proposes", *Artificial Intelligence Review*, vol. 37, no. 4, pp. 331-344, 2012. View Article Google Scholar
- [18] J. Ellis and R. Goodyear, "Students' experiences of e-learning in higher education", *Routledge Publications*, pp. 1-220, 2016. View Article Google Scholar