Campusview: IQAC Automation

A PROJECT REPORT (PHASE-2)

by

ALAN SHAJI(VJC21CS013)
ANIRUDDH AJAY (VJC21CS028)
KRISHNADEV P MELEVILA (VJC21CS084)
ROBIN GEORGE (VJC21CS104)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM
APRIL 2025

Campusview: IQAC Automation

A PROJECT REPORT (PHASE-2)

by

ALAN SHAJI(VJC21CS013)
ANIRUDDH AJAY (VJC21CS028)
KRISHNADEV P MELEVILA (VJC21CS084)
ROBIN GEORGE (VJC21CS104)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

under the guidance

of

Mr.Andrews Jose

Assistant Professor, CSE Dept.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

APRIL 2025

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

Department of Computer Science and Engineering

Vision

Moulding socially responsible and professionally competent Computer Engineers to adapt to the dynamic technological landscape

Mission

1. Foster the principles and practices of computer science to empower life-long learning and build careers in software and hardware development.
2. Impart value education to elevate students to be successful, ethical and effective problem-solvers to serve the needs of the industry, government, society and the scientific community.
3. Promote industry interaction to pursue new technologies in Computer Science and provide excellent infrastructure to engage faculty and students in scholarly research activities.

Program Educational Objectives

Our Graduates

1. Shall have creative aid critical reasoning skills to solve technical problems ethically and responsibly to serve the society.
2. Shall have competency to collaborate as a team member and team leader to address social, technical and engineering challenges.
3. Shall have ability to contribute to the development of the next generation of information technology either through innovative research or through practice in a corporate setting
4. Shall have potential to build start-up companies with the foundations, knowledge and experience they acquired from undergraduate education

Program Outcomes

1. **Engineering knowledge**:Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**:Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design / development of solutions**:Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**:Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**:Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:**Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:**Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:**Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9. **Individual and team work:**Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **Communication:**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and unread in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**:Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes

1. Ability to integrate theory and practice to construct software systems of varying complexity
2. Able to Apply Computer Science skills, tools and mathematical techniques to analyse, design and model complex systems
3. Ability to design and manage small-scale projects to develop a career in a related industry.

VISWAJYOTHI COLLEGE OF ENGINEERING AND TECHNOLOGY, VAZHAKULAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report (phase-2) entitled " **CAMPUSVIEW: IQAC AUTOMATION"** is a bonafide record of the work done by **ALAN SHAJI (VJC21CS013), ANIRUDDH AJAY (VJC21CS028),KRISHNADEV P MELEVILA (VJC21CS084),ROBIN GEORGE (VJC21CS104) in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of APJ Abdul Kalam Technological University.**

Internal Supervisor **External Supervisor**

Project Coordinator **Head of Department**

ACKNOWLEDGEMENT

First and foremost, We thank **God Almighty** for His divine grace and blessings in making all these possible. May he continue to lead us in the years. It is our privilege to render our heartfelt thanks to our beloved Manager, **Rev. Msgr. Dr. Pius Malekandathil** , our director **Rev. Dr. Paul Parathazham** and our Principal **Dr. K K Rajan** for providing us the opportunity to do this project report (phase-1) during the Fourth year (2023) of my B.Tech degree course. We are deeply thankful to my Head of the Department, **Dr. Amel Austine** for his support and encouragement. We would like to express my sincere gratitude and heartfelt thanks to our Project Guide **Mr. Andrews Jose**,Assistant Professor, Department of Computer Science and Engineering for her motivation, assistance and help for the project. We also express sincere thanks our Project Coordinator **Mr. Andrews Jose**,Assistant Professor, Department of Computer Science and Engineering for her guidance and support. We also thank all the staff members of the Computer Science Department for providing their assistance and support. Last, but not the least, We Would like to thank all our friends and family for their valuable feedback from time to time as well as their help and encouragement.

DECLARATION

We undersigned hereby declare that the project report "CAMPUSVIEW: IQAC AUTOMATION", submitted for partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology of the APJ Abdul Kalam Technological University is a bonafide work done by myself under the supervision of Mr.Andrews Jose. This submission represents ideas in my own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Vazhakulam

Date: 28-03-2025

ALAN SHAJI

ANIRUDDH AJAY

KRISHNADEV P MELEVILA

ROBIN GEORGE

ABSTRACT

The CampusView project is a web-based platform designed to automate and streamline Internal Quality Assurance Cell (IQAC) operations for educational institutions. It aims to replace traditional manual processes involved in accreditation workflows, document management, and report generation with a more efficient, centralized solution. The system leverages technologies such as Node.js, MongoDB, and Handlebars.js to provide a robust backend and user-friendly interface. Key features include secure role-based access, real-time analytics dashboards, and automated report generation for NAAC and NBA accreditation requirements. By simplifying complex IQAC operations and ensuring data accuracy, CampusView enhances productivity, reduces administrative overhead, and supports institutions in maintaining high-quality educational standards. The project also emphasizes scalability, making it adaptable to the evolving needs of educational institutions. Index Terms-IQAC Automation, Accreditation Workflow, Campus Management System, Document Management, Report Generation, NAAC Accreditation, NBA Accreditation, Node.js, MongoDB, Web-Based Platform, Education Technology

Contents

List of Figures

List of Abbreviations

IQAC	Internal Quality Assurance Cell
NAAC	National Assessment and Accreditation Council
NBA	National Board of Accreditation
CMS	College Management System
LMS	Learning Management System
SIS	Student Information Systems
IOT	Internet of Things
UAT	User Acceptance Testing
GDPR	General Data Protection Regulation

Chapter 1

INTRODUCTION

In today's academic environment, effective quality assurance is essential for maintaining and improving educational standards. For higher education institutions in India, securing accreditations such as NAAC (National Assessment and Accreditation Council) and NBA (National Board of Accreditation) is crucial, as these affirm an institution's commitment to quality in education and student outcomes. The IQAC (Internal Quality Assurance Cell) operations play a pivotal role in collecting, organizing, and analyzing data required for these accreditations. However, most institutions rely on traditional, manual, and often complex Excel-based systems for these tasks, which can be cumbersome, prone to errors, and limited in scalability.

This project aims to transform the existing macro-enabled Excel application, which has been used for IQAC operations, into a streamlined web-based application built with Node.js. Moving to a web application format provides numerous advantages, such as improved accessibility, better data management, and easier integration with other institutional systems. Additionally, a web-based system reduces dependency on specific software, as it can be accessed from any device with an internet connection, making data handling and reporting more efficient and centralized.

By leveraging modern web technologies, this project seeks to create a solution that not only simplifies the accreditation process but also ensures data accuracy, security, and ease of use. The web application will empower IQAC members to perform operations seamlessly, improve collaboration across departments, and provide automated features that reduce manual effort. Ultimately, this project will serve as a valuable tool for institutions aiming to streamline their accreditation workflows and uphold educational quality standards in line with national requirements.

1.1 Problem Statement

Institutions aiming for NAAC and NBA accreditation rely heavily on the Internal Quality Assurance Cell (IQAC) to maintain and report on various academic and administrative quality standards.

However, most IQAC operations are managed through complex, macro-enabled Excel applications, which are not only difficult to maintain but also lack accessibility, scalability, and robust data security. These traditional methods increase the risk of human error, limit collaboration, and require significant manual effort to handle large volumes of accreditation-related data. Consequently, there is a critical need for a web-based solution that can automate, streamline, and secure these processes, making accreditation management more efficient and reliable

1.2 Objective

1. Streamline IQAC Data Management: Develop a web-based application to simplify the collection, storage, and organization of data required for NAAC and NBA accreditation, reducing reliance on complex Excel-based systems.

2. Enhance Accessibility and Collaboration: Enable seamless access to IQAC data for authorized users across devices, fostering real-time collaboration among departments involved in accreditation processes.

3.Improve Data Accuracy and Security: Implement automated features and robust security measures to minimize human error, ensure data integrity, and protect sensitive institutional information throughout the accreditation lifecycle.

1.3 Scope

1.Conversion of Excel-Based System to Web Application: Transform the existing macro-enabled Excel application used for IQAC operations into a scalable, interactive web application using Node.js to improve accessibility and ease of use.

2. User-Friendly Interface and Dashboard: Design a clean, intuitive interface that allows IQAC members to easily navigate and manage accreditation data, with a dashboard providing quick insights and status updates on accreditation metrics.

3. Data Entry, Management, and Reporting Automation: Enable automated data entry, management, and report generation features, reducing manual effort and increasing the speed and accuracy of data processing for accreditation needs.

4. Role-Based Access and Security: Implement role-based access control to ensure that only authorized users can view, edit, or approve sensitive data, enhancing overall data security and compliance.

5.Scalability and Cross-Platform Accessibility: Ensure that the application is responsive and accessible across various devices, allowing IQAC members and administrators to access information anytime, from anywhere, on different platforms.

Chapter 2

LITERATURE SURVEY

2.1 Research Paper on College Management System

College management systems (CMS) are increasingly valuable in educational institutions, aiming to streamline and automate various administrative and academic functions. These systems support operations like attendance tracking, scheduling, course management, and reporting, making it easier for institutions to maintain and access records efficiently. CMS typically involves modules tailored to handle tasks unique to a college environment, such as student profiles, faculty data, course scheduling, and examination records. By automating these functions, CMS reduces paperwork and administrative burdens, enabling institutions to focus on enhancing educational quality and compliance with accreditation standards(paper1).

IQAC Automation in Accreditation Readiness For institutions pursuing NAAC and NBA accreditations, automated systems like CampusView enable centralized management of IQAC-related data, essential for institutional quality assurance. The CMS provides tools to monitor compliance, manage institutional data, and generate essential reports for accreditation assessments. Systems designed for accreditation often integrate with Learning Management Systems (LMS) and data analytics platforms, creating a cohesive infrastructure for data collection, reporting, and continuous quality improvement. CampusView also incorporates role-based access controls, ensuring that data is securely accessible to authorized users only

2.1.1 Methodology

The methodology for the development of a College Management System (CMS) follows a structured approach to address the challenges faced by educational institutions in managing their operations efficiently. The process begins with identifying the problem through a thorough literature review and gathering insights from surveys and interviews with stakeholders (students, faculty, and administrators). This helps define the system requirements and ensures the CMS meets the needs of the institution.

The system design involves creating a modular architecture with a user-friendly interface and robust backend, incorporating technologies like MySQL for the database and frameworks such as Node.js or Django for development. The implementation phase focuses on developing and integrating modules like student management, fee processing, and examination scheduling. Once developed, the system undergoes rigorous testing, including performance and security evaluations, to ensure it meets the desired goals. A pilot test in a real-world environment allows for feedback and further refinement. Finally, data analysis is conducted to assess the system's effectiveness in improving efficiency, accuracy, and user satisfaction, with suggestions for future improvements and scalability.

2.1.2 Advantages

1. Improved Efficiency in Data Management: Web-based systems streamline the collection and management of vast datasets necessary for accreditation. These platforms automate workflows, reducing manual data entry and improving the overall efficiency of the accreditation process

2. Real-Time Monitoring and Reporting: With integrated dashboards and real-time data analytics, institutions can monitor their compliance status continuously, ensuring they are always prepared for audits or accreditation visits. Such systems reduce the risk of non-compliance and ensure timely reporting

2.1.3 Disadvantages

1. HIS High Initial Setup and Maintenance Costs: Implementing a sophisticated web-based system for accreditation can be expensive. The costs are often associated with system design, infrastructure setup, integration with existing platforms (such as ERP or LMS), and ongoing maintenance.

2. HIS Data Privacy and Security Concerns: Storing sensitive accreditation data on web-based platforms raises security and privacy issues. These systems need to comply with data protection regulations (e.g., GDPR) and ensure secure access control, which can be a challenge when dealing with large volumes of student and institutional data.

2.2 Towards Smart Campus Management: Defining Information Requirements for Decision Making through Dashboard Design

The concept of a ”smart campus” is gaining traction globally as universities face a range of operational and developmental challenges. Modern campuses require flexibility to adapt to fluctuating student numbers, varying space demands, and funding challenges, leading institutions to seek technological solutions that enhance efficiency and decision-making capabilities. Smart campus tools, especially IoT-enabled systems, are valuable in optimizing space, saving energy, and enhancing the experience for students and staff alike. However, while these tools offer significant potential for supporting campus managers, there is limited research on how data from these systems can be effectively integrated into organizational decision-making processes.This study addresses the need to define information requirements for campus management decision-making, specifically through the use of customized dashboards. Dashboards serve as powerful tools by consolidating and visually displaying essential campus data in a single interface, allowing decision-makers to monitor real-time information and respond effectively. The research presents a structured approach to designing dashboards that align with the decision-making needs of campus management. By connecting IoT data with strategic processes, this study demonstrates how dashboards can facilitate better management of campus spaces and resources.

2.2.1 Methodology

The methodology for designing a smart campus management system (SCMS) emphasizes a user-centered, iterative design approach that aligns IoT data with the strategic needs of campus managers. The development process begins with problem identification through a literature review and needs assessment with stakeholders, such as campus managers, faculty, and operational staff. This approach ensures that the SCMS accurately reflects the information requirements and management goals of the institution. The system design involves developing modular dashboards with intuitive interfaces and integrating IoT data sources through a centralized platform. Technologies such as data visualization frameworks and IoT protocols (e.g., MQTT) are commonly used to collect and display information in real time. During implementation, specific campus management modules, such as space utilization and energy monitoring, are tested and refined with stakeholder feedback. Rigorous testing, including usability and security assessments, ensures the system’s effectiveness and data security. The final system is piloted within the campus environment, with follow-up assessments to measure its impact on decision-making efficiency, sustainability, and user satisfaction.

2.2.2 Advantages

Enhanced Decision-Making and Resource Allocation: Smart campus systems with IoT integration enable campus managers to make data-driven decisions. Dashboards provide a consolidated view of critical data, supporting efficient space management and optimal allocation of campus resources, thus improving overall campus operations.

Continuous Monitoring and Strategic Insights: By offering real-time updates through interactive dashboards, SCMS facilitates ongoing monitoring of campus performance metrics. This enables managers to maintain compliance with institutional standards, quickly address emerging issues, and continuously improve campus strategies based on data trends.

2.2.3 Disadvantages

High Setup and Maintenance Costs: Establishing an IoT-based SCMS involves substantial initial costs for IoT devices, data infrastructure, and software development. Additionally, ongoing maintenance and periodic updates to the system require considerable resources, which can be challenging for institutions with limited budgets.

Data Privacy and Security Risks: IoT-enabled campus management systems collect and store sensitive operational data. Ensuring data privacy and compliance with regulations (such as GDPR) requires robust security protocols, role-based access controls, and regular audits. These measures are essential to safeguard data integrity but can also increase the complexity and cost of system maintenance.

2.3 COLLEGE MANAGEMENT WEB APPLICATION SYSTEM USING MEAN STACK

As educational institutions grow in size and complexity, the need for efficient, automated solutions for campus management has become essential. Traditional methods for managing campus resources, disseminating information, and handling administrative tasks often involve time-consuming, paper-based processes that lack scalability and flexibility. With advancements in technology, web-based applications are emerging as viable solutions to streamline and centralize these processes, enhancing communication, coordination, and data accessibility among students, faculty, and administrative staff.The College Management Web Application System, built using the MEAN stack (MongoDB, Express.js, Angular, and Node.js), addresses these needs by providing an integrated platform for managing various aspects of campus operations. This application allows stakeholders to manage student data, track attendance, access academic materials, and communicate important updates in real time. Through automation and centralization, this system minimizes the manual workload for faculty and staff, reduces errors in data handling, and improves the efficiency of daily operations. It also incorporates security measures, such as encryption, to ensure the safety of sensitive information shared within the campus network.

This system offers a modular design, catering to different user roles, including students, teachers, and administrators, with functionalities tailored to each group's specific needs. For example, faculty can use mobile devices to take attendance, upload assignments, and notify students of upcoming exams, while students can access their academic records and participate in discussion forums. By focusing on ease of use, accessibility, and security, the College Management Web Application System aims to provide a reliable digital infrastructure that enhances institutional management and promotes effective communication across the campus community. This transition to a digital campus environment reflects a broader trend in educational technology, where institutions leverage modern software solutions to achieve operational excellence and create a more connected, informed campus culture.

2.3.1 Methodology

The methodology for developing the College Management Web Application System using the MEAN stack (MongoDB, Express.js, Angular, and Node.js) follows a structured and iterative approach to meet the needs of educational institutions effectively. The process began with a comprehensive needs analysis, which involved reviewing current literature on college management systems and conducting surveys and interviews with stakeholders, including students, faculty, and administrative staff. This initial phase was essential for identifying the key functional re-

quirements and understanding the specific challenges faced by each user group within the college environment.

2.3.2 Advantages

1. Efficiency in Data Management: The web-based system significantly improves data management by centralizing and automating processes that were previously manual, such as attendance tracking, notifications, and assignment uploads. This reduction in paperwork and manual data entry saves time and minimizes errors, allowing faculty and administrators to focus on more critical tasks.

2. Real-Time Communication and Updates: The notification module allows administrators to broadcast real-time updates to students and staff, ensuring that critical information—such as schedule changes, upcoming events, or examination reminders—is accessible instantly. This feature enhances responsiveness and keeps everyone informed, fostering a more connected campus environment

3. Customizable and Modular Design: The system's modular design accommodates different functionalities for each user group (students, teachers, administrators), providing a tailored experience. This approach enables scalability and flexibility, allowing additional modules or features to be incorporated as institutional needs evolve.

4. Improved Access to Academic Resources: Through the discussion forum and assignment modules, students can easily access notes, assignments, and course materials anytime, promoting better engagement and self-directed learning. Faculty can also use the discussion forum to answer questions and provide additional support outside of classroom hours.

5. Secure and Role-Based Access Control: By implementing role-based access and encryption, the system ensures that sensitive information is accessible only to authorized users, enhancing data security. This feature aligns with data protection standards and helps prevent unauthorized access to student and institutional data

2.3.3 Disadvantages

1. High Initial Setup and Maintenance Costs: Developing and deploying a web-based college management system requires considerable resources, including software development, server infrastructure, and ongoing maintenance. For institutions with limited budgets, these upfront and operational costs can be challenging, especially if the system requires frequent updates or scaling as user demand grows.

2. Privacy and Security Concerns: Although the system includes encryption and role-based access, storing large volumes of sensitive information (e.g., student grades, attendance records) on a web-based platform poses inherent security risks. Compliance with data protection regulations, such as GDPR, requires robust security protocols, regular audits, and potentially costly cybersecurity measures to safeguard against unauthorized access or data breaches.

3. Dependency on Internet Connectivity: Since the system is web-based, it relies on a stable internet connection for users to access features and updates in real time. In areas with limited or unreliable internet connectivity, this dependency could hinder the system's effectiveness and user experience, especially for remote or rural campuses.

4. Complexity in Initial Training and User Adoption: Implementing a new digital system often requires training for users to become familiar with its features and functionalities. Some students and faculty may find it challenging to adapt to the new system, particularly if they are not comfortable with technology, which may initially affect user adoption and satisfaction.

2.4 UNIVERSITY RESEARCH PROJECT MANAGEMENT SYSTEM BASED ON CLOUD PLATFORM

Managing research projects within universities is a complex task due to the variety and volume of projects, which may include scientific research and teaching-related projects across different disciplines and funding levels. Traditionally, project management in universities has relied on manual processes, which are often inefficient, challenging to track, and resource-intensive. These methods face limitations in terms of data accessibility, transparency, and ease of collaboration, making it difficult for project managers, researchers, and evaluators to efficiently handle project information and assessments.

The introduction of cloud computing offers a promising solution to these challenges, providing scalable, distributed computing power and data storage accessible over the internet. With advancements in cloud technology, universities can leverage a cloud-based system to streamline research project management, allowing for real-time data access, improved collaboration, and centralized control over project documentation. This paper discusses the design and implementation of a cloud-based university research project management system, aimed at digitizing project workflows and enhancing project tracking and evaluation. By automating submission, review, and feedback processes, the system aims to increase efficiency, reduce manual workload, and improve the quality and accessibility of project information.

2.4.1 Methodology

The development of the university research project management system followed a structured methodology that combined cloud computing principles with web-based application design. The primary objective was to enable project managers, researchers, and reviewers to submit, track, and evaluate project-related information seamlessly. The methodology involved several key stages.

1. Requirement Analysis: The first stage involved analyzing the needs of universities in managing research projects. Interviews and surveys were conducted with project managers, faculty members, and evaluators to determine the main functional requirements, including project submission, review, progress monitoring, and results display.

2. System Architecture Design: The system was designed using a cloud-based, three-layered architecture:

3. Infrastructure Layer: This layer provides the basic resources, including servers and storage, supporting the application's backend.

4. Platform Layer: Hosts databases, application programming interfaces (APIs), and development environments, enabling the creation and deployment of custom applications.
5. Software Layer: Offers user-oriented services, enabling users to access the system from any location via a web browser
6. Module Development: The system was divided into several modules, each handling specific tasks:
7. Application and Review Module: Manages online project submissions and reviews, enabling administrators to monitor and approve research proposals.
8. Opening Management Module: Allows project leaders to submit opening reports and task outlines, which are reviewed by experts.
9. Progress Management Module: Tracks project milestones and progress, providing feedback and suggestions to researchers.
10. Completion Management Module: Handles final submissions and project evaluations, facilitating the acceptance of completed projects.

2.4.2 Advantages

1 Improved Efficiency and Productivity: By automating each stage of project management, from application to completion, the system minimizes manual paperwork and reduces the administrative burden on project managers and faculty. This streamlined approach accelerates the approval and evaluation processes, leading to faster project execution.

2 Real-Time Access and Collaboration: The cloud-based system allows users to access and manage research project data anytime, from any location, fostering collaboration between faculty, reviewers, and administrators. This real-time access is particularly beneficial for distributed project teams, as it ensures that all stakeholders can stay informed and engaged throughout the project lifecycle.

3 Enhanced Data Security and Integrity: By leveraging the cloud, the system ensures secure storage and backup of project data, reducing the risk of data loss due to hardware failures. Additionally, the platform supports role-based access control, which restricts data access to authorized users, safeguarding sensitive project information.

2.4.3 Disadvantages

1 .Initial Setup and Maintenance Costs: Implementing a cloud-based research project management system involves initial setup costs, including cloud service fees, software development, and integration with existing systems. While these costs may be justified over time through increased efficiency, they can be a barrier for institutions with limited budgets.

2 Dependence on Internet Connectivity: Since the system is cloud-based, a stable internet connection is necessary for users to access the platform. Any connectivity issues can hinder access, which could disrupt workflows, especially in areas with unreliable internet service.

3 Data Privacy and Security Concerns: Although the cloud platform offers secure data storage, some institutions may have concerns about storing sensitive research data on external servers. Ensuring compliance with data protection regulations, such as GDPR, requires robust security protocols, which may be complex and costly to implement.

4 User Training and Adoption Challenges: The transition from traditional, manual project management methods to a cloud-based system may require users to undergo training, which can be time-consuming. Users who are not comfortable with digital tools may face challenges in adopting the new system, potentially slowing down initial operations.

5 Potential System Downtime and Technical Issues: Like any web-based system, the URPMS is susceptible to technical problems, such as server downtime or software bugs. These issues can disrupt access to essential functions and may require IT support to resolve, impacting user productivity and project timelines.

2.5 RECOMMENDER SYSTEMS IN SMART CAMPUS DOMAIN: A SYSTEMATIC MAPPING

Recommender systems are essential in modern digital platforms for filtering vast amounts of data to provide users with relevant content based on their preferences. These systems are increasingly being applied in the educational sector, where they support smart campuses by delivering personalized recommendations for resources such as courses, books, research materials, and other educational content. The concept of a smart campus integrates Internet of Things (IoT) and data analytics to create responsive, interconnected university environments that optimize resource use, improve student experience, and enhance operational efficiency. Within this setting, recommender systems are pivotal in aiding students, faculty, and staff by personalizing the campus experience and facilitating access to educational and administrative resources that align with user needs.

This paper conducts a systematic mapping of existing research in recommender systems within smart campus environments. By examining studies from 2017 to 2022, the paper identifies key trends, algorithms, and types of filtering used in recommender systems tailored to smart campus needs. Through this mapping, the paper highlights the most common types of recommender algorithms and filtering methods, providing insights into current research gaps and opportunities. This overview aims to advance the understanding of how recommender systems can be best utilized in smart campuses, focusing on the unique demands and challenges posed by educational environments.

2.5.1 Methodologies

The study employs a systematic mapping approach, which involves reviewing and categorizing literature on recommender systems within smart campuses. The process began by defining specific research questions aimed at understanding trends, commonly used algorithms, popular filtering types, and the types of databases utilized for recommender systems in smart campus settings. Database Search and Keyword Selection: Using academic databases like Scopus, IEEE, and SpringerLink, the researchers searched for articles from 2017 to 2022 related to ”recommender systems,” ”smart campus,” and other keywords associated with filtering techniques like ”collaborative filtering” and ”content-based filtering.” A total of 611 articles were initially identified. Filtering Process: A rigorous filtering process narrowed down the articles to 55 by applying criteria based on relevance to recommender systems within smart campuses. The inclusion criteria required articles to focus on educational re-

sources, smart campus applications, or recommender algorithms.

Analysis and Categorization: Selected articles were analyzed to categorize the types of recommended items, the algorithms used, and the evaluation methods applied. The analysis also included a review of the countries contributing the most research in this domain, with China emerging as a leading contributor.

2.5.2 Advantages

1 Efficient Resource Utilization: Smart campuses leverage recommender systems to better manage and allocate resources. By analyzing user behavior and preferences, these systems can suggest underutilized facilities or resources, supporting more efficient campus operations.

2 Supports Diverse Learning Styles: Recommender systems can adapt to different learning needs and preferences, offering students materials suited to their individual learning pace and style. This adaptability promotes a more inclusive educational environment by accommodating various learning preferences.

3 Real-Time Insights and Analytics: These systems provide campus administrators with real-time data on user preferences, usage patterns, and engagement metrics, allowing for data-driven decision-making. Such insights can guide improvements in campus services and infrastructure.

4 Increased Research and Development Opportunities: By integrating recommender systems within smart campuses, universities can foster innovation in AI, machine learning, and data science, offering rich ground for further research and the development of advanced algorithms.

2.5.3 Disadvantages

1 High Implementation Costs: Developing and deploying recommender systems in a smart campus setting requires significant financial and technical resources, including investments in cloud infrastructure, data storage, and IoT integration. These high initial costs may be prohibitive for institutions with limited budgets.

2 Data Privacy and Security Concerns: Recommender systems in smart campuses gather and analyze a vast amount of personal and behavioral data, raising concerns about data security and privacy. Ensuring compliance with data protection regulations, like GDPR, requires robust security measures, which can be complex and costly.

3 Reliance on Internet Connectivity: Since these systems often operate on cloud platforms, they require a stable internet connection for optimal functionality. In cases of network outages or unreliable connectivity, users may face disruptions that limit access to personalized recommendations.

4 Complexity in User Training and Adaptation: The shift to smart campus technology, including recommender systems, may require training for students and faculty to navigate new digital tools. Resistance to adopting new technologies or lack of technical expertise can limit the system's effectiveness.

2.6 Design of smart campus management system based on internet of things technology

The rapid development of the Internet of Things (IoT) has paved the way for advanced applications in educational environments, particularly in the creation of smart campuses. A smart campus integrates IoT technology with traditional educational infrastructure to create an interactive, data-driven environment. The smart campus management system (SCMS) leverages IoT for real-time data collection, processing, and analysis, aiming to enhance campus management and streamline daily operations. This involves deploying IoT devices across campus facilities to monitor activities, gather data, and analyze behavior, supporting improved resource allocation and operational efficiency.

2.6.1 Methodology

The methodology for developing the smart campus management system (SCMS) utilizes a comprehensive approach, focusing on IoT-driven data collection, big data analysis, and a structured service design. The first stage involves the deployment of face recognition terminals across key areas on campus—such as libraries, dormitories, and academic buildings—to monitor entry and exit, gather real-time behavioral data, and enhance security. These terminals communicate over TCP/IP protocols, transmitting data to a central server, enabling continuous data collection without manual input.The next stage focuses on data processing and analysis using a big data platform built with Hadoop's HDFS and MapReduce, which enables efficient storage, processing, and analysis of large volumes of data. This platform allows the SCMS to aggregate and visualize data in real time, providing campus managers with insights into patterns such as student attendance, facility utilization, and movement trends. The modular architecture of the system supports this scalability, allowing integration with third-party systems and expansion to include additional data modules as campus needs evolve. The system architecture follows a modular design, enhancing scalability and flexibility. The interface is designed for ease of use, offering administrators, faculty, and staff access to specific data modules based on roles and permissions, thus providing a role-based access control that improves security and operational efficiency. During the development process, formative usability testing was conducted, with 36 testers evaluating the system on metrics such as ease of learning, interaction, fault tolerance, and overall user satisfaction. Feedback from these tests informed iterative improvements, enhancing the interface, data readability, and user experience. Finally, the system underwent real-world performance evaluation, with user satisfaction surveys and performance tests assessing factors such as response time, data

accuracy, and usability under varying conditions. This rigorous methodology ensures that the SCMS is both responsive and scalable, capable of meeting the dynamic demands of a smart campus environment while delivering secure, data-driven insights to support effective campus management.

2.6.2 Architecture

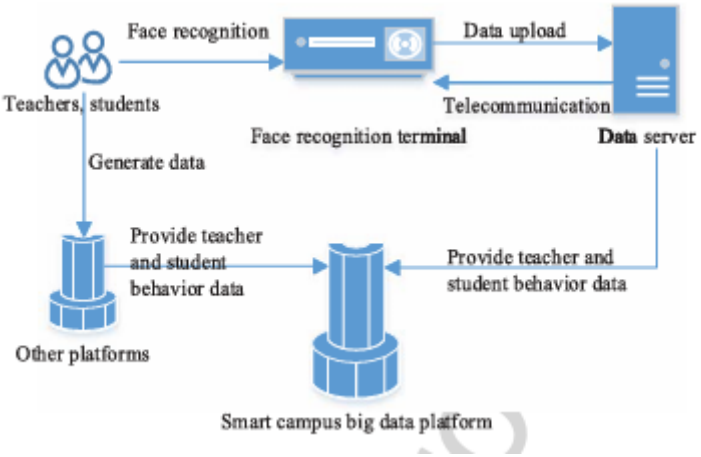

Figure 2.1: Architecture

2.6.3 Advantages

(a) Enhanced Campus Security and Access Control: By using face recognition for entry and exit monitoring, the SCMS ensures secure access to campus facilities, preventing unauthorized access and improving overall campus security. This system allows administrators to monitor access in real time, responding quickly to any irregularities.

(b) Improved Resource Allocation and Management: The data collected on facility usage and student behavior allows administrators to make informed decisions on resource allocation. For instance, understanding peak usage times for libraries or labs can help optimize staffing and operational schedules, ensuring efficient resource utilization.

2.6.4 Disadvantages

(a) High Initial Implementation and Maintenance Costs: Deploying IoT devices across a campus and maintaining a big data platform requires substantial financial investment. The initial setup, infrastructure upgrades, and ongoing maintenance costs may be challenging for institutions with limited budgets.

(b) Data Privacy and Security Risks: The SCMS collects extensive personal and behavioral data, raising concerns about data privacy and security. Ensuring compliance with data

protection regulations, like GDPR, requires strong security protocols and continuous monitoring, adding complexity and cost to system maintenance.

2.7 Investigation on Smart Campus Management Platform Based on Digital Twin

As educational institutions strive for greater efficiency and digital integration, smart campuses have emerged as a key area of focus, integrating advanced technologies to optimize campus operations and enhance student, faculty, and administrative experiences. In this context, Digital Twin technology provides a powerful tool to mirror physical assets, processes, and even student interactions within a digital platform, offering real-time visibility and control over campus environments. This technology supports three-dimensional visualization, simulation, and analysis, transforming how campuses manage resources and deliver services. By leveraging Digital Twin, educational institutions can achieve high levels of operational accuracy and data-driven insights, which can improve decision-making, enhance security, and support the sustainable management of campus facilities.The focus of this paper is on developing a Smart Campus Management Platform that utilizes Digital Twin technology to digitize various campus aspects—such as student management, facility monitoring, and academic resources—into a comprehensive management ecosystem. This platform aims to address common campus management challenges, such as real-time tracking of student attendance, communication delays, and facility management inefficiencies. The integration of big data analytics with Digital Twin enables the collection and analysis of vast amounts of campus data, allowing for better management practices and improved responsiveness to campus needs. Through this advanced platform, the paper proposes a model for modernizing educational administration, enhancing both the quality of education and the efficiency of daily operations.

2.7.1 Methodologies

The digital twin-based smart campus management platform incorporates several key methodologies to enhance campus operations. First, Digital Twin Integration creates a virtual model of the campus, utilizing real-time data from various components such as students, facilities, and resources. By leveraging sensors and IoT devices, the platform continuously gathers data to form a digital replica of physical assets, which enables effective monitoring, simulation, and analysis of campus activities. Complementing this is the Big Data Analysis methodology, which processes large volumes of data from campus operations, including student attendance and resource usage. This analysis identifies trends and patterns, supporting data-driven decision-making across departments and enhancing operational efficiency. The platform is organized through a Modular Management System that divides functionality into

dedicated modules, including student management, teacher management, facility management, and security management. This modular structure allows each department to focus on specific administrative functions while ensuring seamless integration across the platform. To guarantee the platform's efficiency, a Testing and Performance Evaluation methodology assesses stability, response time, throughput, and security, especially under high-demand scenarios like concurrent user sessions. Performance indicators such as system response time and throughput provide insights for further optimization. Finally, the platform requires Continuous Maintenance and upgradation to stay aligned with evolving campus needs and technological advancements. This approach ensures ongoing improvements in system security, data management, and functional enhancements, allowing the platform to adapt to dynamic educational environments. Together, these methodologies create a comprehensive and efficient smart campus management platform that leverages digital technology and data analytics to support high-quality, streamlined campus operations.

2.7.2 Advantages

(a) Enhanced Efficiency: By using digital twin technology, campuses can simulate and monitor real-time activities, leading to streamlined operations. This boosts efficiency in resource allocation, maintenance, and campus facility management, allowing administrators to make quick, informed decisions.

(b) Data-Driven Decision Making: With big data analysis, the platform provides insights into patterns in student attendance, resource usage, and academic performance. This helps optimize processes and improves campus planning, enhancing the overall student experience.

(c) Improved Security and Safety: The platform's real-time monitoring and predictive analysis capabilities strengthen campus security. It can quickly identify anomalies or potential security risks, improving emergency response and student safety.

(d) Modular and Scalable Design: The modular architecture allows for easy customization and expansion. New modules can be added as campus needs evolve, making the system highly adaptable and future-proof.

2.7.3 Disadvantages

(a) High Implementation Costs: The integration of digital twin technology, IoT devices, and big data analytics involves significant upfront costs, including the purchase of hardware, software, and the installation of sensors.

(b) Complex Maintenance and Upgrading Needs: The platform requires regular maintenance, data backup, and security upgrades. The complexity of managing these processes could demand specialized personnel and resources, leading to ongoing costs.

(c) Potential Technical Issues: The system's heavy reliance on technology and connectivity may lead to challenges like data loss, technical glitches, or cyberattacks, which could disrupt campus operations and compromise data integrity.

(d) User Resistance and Training Requirements: Implementing this advanced system may face resistance from users unfamiliar with digital platforms. Comprehensive training is necessary for students, faculty, and staff, which requires time and resources to ensure smooth adaptation.

2.8 College Management System

The introduction of the paper titled ”College Management System” presents the project as a comprehensive Intranet-based application designed to enhance efficiency in managing and accessing information within a college or university. The system is intended to provide centralized management of various institutional data, making it easier for students, staff, and administrators to retrieve, update, and monitor relevant information based on specific permissions and roles. The College Management System (CMS) is developed with PHP Laravel, HTML, and CSS, enabling a robust web-based platform that integrates seamlessly with the institution's infrastructure.The CMS is structured to handle numerous aspects of college life, from maintaining detailed student and faculty profiles to managing notices, event updates, course materials, fee payments, and academic records. The platform operates on a secure MySQL database, overseen by a college administrator who can control access and make modifications as necessary, ensuring data integrity and privacy. Each user role—whether student, faculty, or administrator—has a unique home page with customized access levels, allowing them to interact with the system's features pertinent to their needs.The CMS aims to reduce manual processes, streamline data access, and enhance information sharing across departments, ultimately reducing the workload on administrative staff and improving response times for data retrieval. Additionally, the platform provides features such as leave applications, exam schedules, and announcements, creating a centralized communication hub. By facilitating these interactions digitally, the CMS not only improves operational efficiency but also promotes a paperless, eco-friendly approach to campus management, aligning with modern technological and sustainability goals. This system addresses the evolving needs of educational institutions, providing a scalable and adaptable solution that enhances user experience and supports the institution's academic and administrative functions,

2.8.1 Methodology

The College Management System (CMS) employs several methodologies to improve information access and management within a college. First, it uses a User Role-Based Access and Authentication system, where unique roles are assigned to different users, such as students, faculty, and administrators. Each user has a designated login ID and password, granting access only to information relevant to their role. This role-based access control enhances data security and limits unauthorized access. The system relies on Centralized Data Management, which integrates and securely stores all institutional data, including student profiles, teacher profiles, notices, fees, and academic materials. Utilizing MySQL, the CMS centralizes data

to allow quick and efficient retrieval for routine college activities, all managed by the college administrator. In addition, the CMS includes a Digital Communication and Notice System, allowing administrators and faculty to post updates, schedules, and announcements in real time. This paperless communication method ensures timely information flow, supporting smooth coordination across the college. Another core methodology is Automated Processes and Records Management, which streamlines tasks like leave applications, fee payments, and exam scheduling. By automating these processes, the CMS reduces manual effort, increases accuracy, and enables quick retrieval of student records, enhancing administrative efficiency. Lastly, the system's Responsive Web Interface, developed with PHP Laravel, HTML, and CSS, ensures accessibility across devices, including Android mobiles. This flexibility allows students and faculty to access the platform from various locations, improving engagement and usability. Together, these methodologies create an efficient, secure, and user-friendly CMS that addresses administrative challenges while fostering a paperless environment.

2.8.2 System Architecture

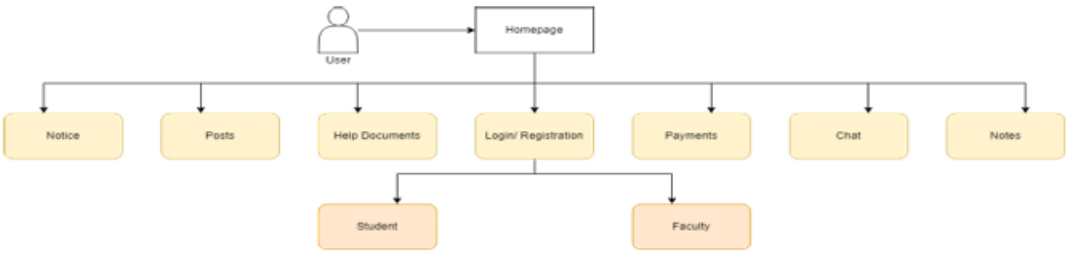

Figure 2.2: Architecture

2.8.3 Advantages

(a) Centralized Data Access: The CMS centralizes all student, faculty, and administrative data, making it easy to access and manage information efficiently. This minimizes errors associated with manual data entry and retrieval.

(b) Improved Communication: he digital notice system enables real-time communication between administrators, faculty, and students, ensuring that everyone receives timely updates on events, schedules, and announcements, which improves coordination across the college.

(c) Enhanced Security and Role-Based Access Control: By assigning specific access permissions to users based on their roles, the CMS improves data security and ensures that sensitive information is only accessible to authorized personnel..

(d) Automation of Routine Tasks: Automated processes such as leave applications, fee payments, and record retrieval save time and reduce the administrative burden, allowing staff to focus on more critical tasks.

2.8.4 Disadvantages

(a) High Initial Setup Cost: Implementing the CMS, including purchasing hardware, software, and necessary security infrastructure, can be expensive, especially for smaller institutions with limited budgets.

(b) Dependency on Internet Connectivity: Since the CMS is an online platform, users require stable internet access to interact with it. Poor connectivity can hinder access, particularly for students in remote areas..

(c) Data Privacy Concerns: With extensive data collected and stored on the system, there is a risk of data breaches. Unauthorized access to sensitive information could compromise student and staff privacy, necessitating robust security measures..

(d) Ongoing Maintenance: The CMS requires continuous maintenance, including updates, backups, and security monitoring, to ensure smooth operation and data protection. This can add to the operational costs and necessitate dedicated IT support.

2.9 Student Information System: Investigating User Experience (UX)

The paper, ”Student Information System: Investigating User Experience (UX)”, highlights the crucial role of Student Information Systems (SIS) in enhancing productivity and managing academic services within higher education. SIS platforms streamline essential functions like course registration, grade tracking, and transcript generation, supporting both administrative efficiency and student performance. Emphasizing usability and user experience (UX), the study underscores that systems designed with a strong focus on positive UX significantly improve user satisfaction, engagement, and overall functionality. Despite the growing emphasis on UX in academia and industry, research gaps exist—particularly in Arab Gulf universities—where student perspectives on SIS usability and experiences are underexplored. Addressing this, the study examines student UX with the SIS implemented at Kuwait's Public Authority for Applied Education and Training (PAAET).

2.9.1 Methodology

The methodology of the paper ”Student Information System: Investigating User Experience (UX)” includes both qualitative and quantitative approaches to evaluate the user experience (UX) of the Student Information System (SIS) at Kuwait's Public Authority for Applied Education and Training (PAAET). The key components are:

Research Sample: The study surveyed 645 students from five PAAET colleges, with demographic data showing a higher representation of female students (81.4%).

Research Instruments:

Focus Group: A preliminary focus group with 16 students was conducted to validate and refine the UX questionnaire statements. Participants provided insights into the clarity and relevance of the questions, which were then adjusted based on their feedback. Questionnaire: The primary tool was a customized User Experience Questionnaire (UEQ), assessing six UX dimensions: attractiveness, efficiency, perspicuity, dependability, stimulation, and novelty. The questionnaire was initially piloted with 50 students to ensure reliability and clarity, with adjustments made as needed. Data Collection and Analysis: The final questionnaire was distributed online across the colleges. Data analysis was conducted using SPSS, with reliability confirmed through high internal consistency (Cronbach's alpha). The researchers also calculated correlations between each UX dimension and overall user satisfaction.

This combined focus group and questionnaire approach allowed the researchers to gather

in-depth feedback on SIS usability, highlighting both strengths and areas for potential improvement.

2.9.2 Advantages

(a) Improved Academic and Administrative Efficiency: The SIS allows students to manage academic.

(b) Enhanced User Experience (UX): Although the overall UX is slightly positive, the SIS scored well in perspicuity (ease of use), dependability (trustworthiness), and stimulation (motivation). This means that the system is relatively user-friendly, reliable, and engaging for students.

2.9.3 Disadvantages

(a) Limited Efficiency: The SIS scored low in efficiency, with some users reporting that system commands do not always execute quickly and that navigating the system can be time-consuming. This inefficiency impacts the overall productivity of users.

(b) Lack of Advanced Features: The system lacks intelligent, personalized features, such as predictive ”what-if” scenarios, course recommendations, or GPA enhancement suggestions. Students indicated that more interactive and advanced functionalities could improve their experience.

2.10 Smart Campus Management with Advanced Learning Management System

The introduction of the paper ”Smart Campus Management with Advanced Learning Management System” explores how IoT (Internet of Things) devices are driving innovation in educational environments by connecting various campus facilities into an integrated smart network. This approach leverages IoT sensors, AI algorithms, and cloud connectivity to streamline the management of classrooms, offices, laboratories, libraries, and dormitories, creating a connected ecosystem that supports real-time monitoring, resource optimization, and enhanced security. The smart campus is designed to foster a more efficient, eco-friendly environment by implementing automated systems like smart lighting, renewable energy sources, and advanced metering to monitor and reduce energy consumption.Furthermore, smart classrooms equipped with an advanced learning management system (LMS) facilitate interactive learning, provide automated attendance, and enable seamless communication between students and teachers, thereby enhancing the educational experience. Centralized data flows allow administrators, teachers, students, and parents to interact with campus resources more effectively, ensuring up-to-date information access and fostering collaboration. By integrating renewable energy management, automated infrastructure control, and real-time security measures, this smart campus model aims to meet the evolving needs of modern educational institutions while promoting sustainability and operational efficiency. The paper proposes a sophisticated framework to modernize campus infrastructure, making it adaptable, secure, and responsive to future technological advancements.

2.10.1 Methodologies

The development of a smart campus with an advanced Learning Management System (LMS) can be achieved through a combination of innovative methodologies that integrate IoT, AI, and renewable energy. By using an IoT-based monitoring and control system, sensors installed across campus can gather data on occupancy, temperature, and lighting, enabling efficient control of resources and reducing energy consumption. The LMS incorporates features like e-learning, automated attendance, and personalized study plans with AI, while virtual reality provides immersive learning experiences. Renewable energy integration through photovoltaic solar panels and smart meters supports sustainable resource management, reducing reliance on non-renewable energy. Data analytics on cloud-stored information from advanced metering infrastructure (AMI) allows for data-driven decisions in managing energy, water, and emissions. A comprehensive security system employs cameras, sensors, and AI to se-

cure campus perimeters and entry points with minimal human intervention. Finally, a smart parking and transportation management system optimizes parking availability and reduces congestion by using sensors and a real-time app for monitoring vehicle occupancy. Together, these methodologies create an efficient, adaptive, and sustainable campus environment.

2.10.2 Proposed System Model

2.10.3 Advantages

(a) Enhanced Resource Efficiency: IoT-enabled systems, such as automated lighting and HVAC controls, optimize resource usage, reducing waste and lowering energy costs.

(b) Sustainable Energy Management: Integrating renewable energy sources like solar panels promotes sustainability by decreasing dependence on non-renewable resources and cutting emissions.

(c) Increased Security and Safety : AI-driven security systems with cameras, sensors, and automated alerts enhance campus safety, allowing for rapid responses with minimal human intervention.

(d) Convenient Campus Management: : Automated attendance, smart parking, and streamlined resource management reduce manual administrative tasks, making campus operations smoother for students, faculty, and staff.

2.10.4 Disadvantages

(a) High Initial Costs: Setting up IoT devices, sensors, renewable energy sources, and AI infrastructure requires a significant upfront investment, which may be challenging for some institutions.

(b) Data Privacy and Security Concerns: Collecting and storing large amounts of data raises concerns around data security and privacy, especially when sensitive information about students and staff is involved.

(c) Maintenance and Technical Expertise Requirements: These advanced systems require regular maintenance and updates, necessitating a skilled technical team for upkeep and troubleshooting.

(d) User Adaptation and Training Needs: Introducing these advanced technologies may require students, staff, and faculty to undergo training, which can take time and effort to ensure effective usage.

2.11 Revising Technology Adoption Factors for IoT-Based Smart Campuses: A Systematic Review

The paper, titled Revising Technology Adoption Factors for IoT-Based Smart Campuses: A Systematic Review, explores how the Internet of Things (IoT) can transform educational institutions into smarter, more sustainable environments. The study underscores IoT's potential to surpass technologies like artificial intelligence (AI) and robotics in enhancing campus efficiency, connectivity, and convenience. Despite the high interest in IoT-enabled smart campuses, the research identifies a lack of comprehensive models guiding the adoption and integration of Through a systematic literature review, the authors aim to bridge this gap by examining various adoption factors and existing models, leading to the proposal of a framework tailored for smart campuses. The paper discusses key factors such as scalability, replicability, reliability, security, privacy, and trust, which are critical for the widespread adoption of IoT in campus environments. By categorizing these elements, the study offers a structured approach to developing IoT solutions that align with campus sustainability goals and operational requirements. IoT technologies specifically for educational settings.

2.11.1 Methodologies

The methodology used in this paper follows a systematic literature review (SLR) approach to evaluate existing research on IoT adoption in smart campuses. This structured approach allows for a comprehensive and unbiased synthesis of studies on technology adoption models, focusing on their applicability to smart campus settings. The SLR process began with a search across three major databases—ScienceDirect, IEEE Xplore, and Springer—using specific keywords related to IoT adoption and smart campus concepts. The authors applied rigorous inclusion and exclusion criteria to ensure relevance, focusing on peer-reviewed, high-quality articles published in English and excluding conference papers, non-accessible texts, and studies not centered on IoT adoption models.Once the articles were selected, the authors analyzed theoretical frameworks used in the studies, such as the Technology Acceptance Model (TAM), Unified Theory of Acceptance and Use of Technology (UTAUT), and Value-based Adoption Model (VAM), along with their corresponding factors, like ease of use, usefulness, security, and trust. By categorizing these variables, the study developed a taxonomy for IoT adoption factors and created a framework tailored to the needs of IoT-enabled smart campuses. This process also included thematic analysis to organize factors into broad categories—technology-specific, organizational-specific, environmental-specific, and end-user-specific factors. The methodology thus provided a systematic foundation for

identifying gaps in current research and forming a new model suited for smart campus environments.

2.11.2 Proposed System Model

2.11.3 Advantages

(a) Enhanced Efficiency and Convenience: IoT technology on smart campuses can improve the management of campus resources, enhancing user convenience and overall operational efficiency.

(b) Support for Sustainable Development: Smart campuses help universities reduce energy consumption, manage resources more sustainably, and meet environmental goals.

(c) Improved Decision-Making : The massive data collection from IoT devices supports data-driven decision-making processes, allowing administrators to address campus needs more effectively.

(d) Increased User Engagement: : Interactive technologies, like augmented reality (AR), improve user engagement and the learning experience, aligning with the demands of modern educational environments.

2.11.4 Disadvantages

(a) High Deployment Costs: Setting up IoT infrastructure on a campus is costly, covering devices, network equipment, and integration with existing systems.

(b) Data Privacy and Security Concerns: Collecting and storing large amounts of data raises concerns around data security and privacy, especially when sensitive information about students and staff is involved.

(c) MData Security and Privacy Concerns: IoT devices pose significant security risks, with concerns around data breaches, unauthorized access, and privacy for users. of Implementation: Integrating IoT solutions on a campus can be complex due to the variety of devices, network requirements, and infrastructure compatibility issues

2.12 The Making of Smart Campus: A Review and Conceptual Framework

The concept of a ”smart campus” is transforming higher education by integrating advanced digital technologies, such as the Internet of Things (IoT), big data, artificial intelligence (AI), and cloud computing, into campus environments. Similar to the development of smart cities, smart campuses offer a connected, data-driven infrastructure that aims to enhance learning, research, sustainability, and campus management. However, as institutions increasingly adopt these innovations, the definition and framework of a smart campus remain ambiguous, and comprehensive, practical applications of the concept are still limited. This study aims to address this gap by conducting a systematic literature review to understand the current state of smart campus research and practice, applying a structured approach to analyze the existing knowledge base. Using the PRISMA methodology, this paper classifies research findings into four major domains—society, economy, environment, and governance—that are aligned with central concepts of digital technology and big data. By examining existing research within these domains, the study seeks to develop a robust conceptual framework, highlight the gaps between theory and practice, and provide recommendations for future research and practical implementation in the development of smart campuses.

2.12.1 Methodologies

The paper employs a systematic literature review methodology following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) approach, which consists of four stages: identification, screening, eligibility, and inclusion. To capture relevant studies, the authors conducted an online search using the Scopus database with keywords such as “smart,” “intelligent,” “campus,” “university,” and “higher education institution,” focusing on publications from 2017 to 2022 to reflect recent technological advancements. Inclusion criteria were peer-reviewed journal articles in English that were accessible online and relevant to the concept of smart campuses, specifically addressing the four domains of society, economy, environment, and governance. Non-peer-reviewed sources, government reports, and documents in languages other than English were excluded. The PRISMA approach helped narrow an initial pool of 3,601 articles to 123 that met all criteria. For structured analysis, the articles were organized under the four domains commonly used in urban research and aligned with smart technology themes, which allowed for a thematic exploration of key trends, technological developments, and thematic focuses. Each selected study was manually analyzed for either conceptual contributions or practical applications (such as pro-

totypes), revealing a gap between conceptualization and real-world implementation across the domains. This method enabled the authors to systematically assess the existing landscape of smart campus research, identifying current gaps and suggesting directions for future studies.

2.12.2 Proposed System Model

2.12.3 Advantages

(a) Comprehensive Literature Review: The paper uses the PRISMA approach, ensuring a systematic review of relevant literature, which provides a solid foundation for understanding the current state of smart campus development. Framework: By categorizing research into four domains—society, economy, environment, and governance—the paper provides a clear, organized approach to studying smart campuses, highlighting key areas for development.

(b) Focus on Future Research: It identifies gaps in research and real-world applications, offering directions for future studies and development to address these limitations.

(c) Practical Implications: The framework includes actionable areas, such as energy management, data governance, and community engagement, making it valuable for universities and researchers planning smart campus initiatives.

2.12.4 Disadvantages

(a) Limited Empirical Data: The study is largely conceptual and relies on existing literature without new empirical data, which may limit the generalization of its findings.

(b) Underdeveloped Governance Domain: Governance, which is critical for effective implementation, is the least explored domain, and the paper notes a shortage of real-world applications in this area.

(c) Heavy Focus on Technology Over People:: The review emphasizes technological aspects like IoT, AI, and big data, while the human aspects (e.g., user experience) are relatively underrepresented.

(d) Potential Reviewer Bias: Since the review was conducted manually, there may be a risk of bias, and the paper itself acknowledges a lack of bibliometric tools for data analysis.

Chapter 3

PROPOSED SYSTEM

The proposed system aims to transform the existing macro-enabled Excel application, which is currently used for IQAC (Internal Quality Assurance Cell) operations related to NAAC (National Assessment and Accreditation Council) and NBA (National Board of Accreditation) accreditations, into a robust and scalable web application. By utilizing Node.js, the new web-based solution will streamline the management of accreditation data, improve the accuracy of reports, and ensure that educational institutions can seamlessly manage the various data-driven tasks required for accreditation purposes. This shift from Excel to a web application will also improve collaboration and data accessibility among team members, enhancing workflow efficiency.

The web application will include features such as automated report generation, real-time data synchronization, and a user-friendly interface that simplifies complex accreditation tasks. It will allow administrators to input data, track progress, and generate reports related to accreditation criteria without the limitations of Excel spreadsheets. Additionally, the system will incorporate user authentication protocols, ensuring that sensitive accreditation data is securely stored and only accessible to authorized personnel. The web-based system will be designed with scalability in mind, allowing for future enhancements and integrations with other systems such as Learning Management Systems (LMS) or Student Information Systems (SIS).

3.1 Process Overview

(a) **Project Planning**

a. Define Project Scope and Objectives: Clearly outline the specific goals and functionalities of the Campus View: IQAC Management System .

b. Identify Target Audience: Understand the needs and preferences of the target users to tailor the app's features accordingly.

c. Establish Project Timeline and Milestones: Create a detailed project timeline with milestones to track progress and ensure timely completion.

d. Assemble Project Team: Bring together a team with expertise in NFC technology, mobile app development, user experience design, and project management.

Requirements Gathering and Analysis

a. Gather User Requirements: Conduct user research and surveys to gather insights into user needs, expectations, and pain points.

b. Analyze User Feedback: Analyze the collected user data to identify common themes, prioritize features, and refine the app's design.

c. Define Functional and Non-functional Requirements: Clearly define the functional requirements (what the app should do) and non-functional requirements (performance, usability, security, etc.).

Design and Development.

a. User Interface (UI) and User Experience (UX) Design: The first step in developing the web application is to create an intuitive, user-friendly, and visually appealing design. The UI will be designed to ensure that the IQAC personnel can easily navigate through different sections, input data, and generate reports with minimal effort. The UX will focus on enhancing the overall user experience by making sure the application is responsive, consistent across devices, and easy to interact with. This includes considering color schemes, button placements, and workflow steps that minimize cognitive load, leading to an efficient user experience.

b. Responsive Design: The application will be developed with a mobile-first approach, ensuring that it is fully responsive across devices like desktops, tablets, and mobile phones. This will provide flexibility to users, allowing them to access the system from any device while maintaining the integrity and usability of the interface.

c. Interactive Elements: Interactive elements such as real-time data entry, dropdowns, and search functionalities will be incorporated to ensure smooth user interaction. The

system will offer a seamless flow from one task to the next, enabling IQAC administrators to complete processes quickly and without errors.

Software Development

a. Backend Development (Node.js): The backend will be developed using Node.js, a powerful, event-driven JavaScript runtime environment. This will provide a fast and scalable foundation for handling multiple users and complex tasks, such as data processing, user management, and report generation.

b. Frontend Development: The frontend will be built using React.js, ensuring a dynamic and responsive user interface. The application will also utilize modern HTML5, CSS3, and JavaScript features to create an interactive user experience, incorporating animations, transitions, and real-time updates.

c. Database Integration (MySQL/MongoDB): The application will be integrated with a MySQL or MongoDB database for efficient storage and retrieval of accreditation-related data. The backend will ensure that data is securely stored and easily accessible for generating reports, analyzing trends, and tracking compliance.

Testing and Quality Assurance

a. Unit Testing: Each individual component of the web application will undergo unit testing to ensure that it performs as expected. This will involve testing backend APIs, database queries, and individual functions to verify their correctness before integration.

b. Integration Testing: Integration testing will ensure that all components of the application, such as the frontend, backend, and database, work seamlessly together. This testing phase will focus on detecting issues that arise from interactions between different parts of the system.

c. User Acceptance Testing (UAT): A group of representative users will be engaged to conduct user acceptance testing. This phase will focus on validating the usability, functionality, and overall experience of the application to ensure that it meets the expectations of the IQAC personnel.

d. Performance Testing: The system will undergo performance testing to evaluate its response time, scalability, and overall stability under various load conditions. This ensures the application remains efficient even with a large number of concurrent users.

3.1.1 Process Flow Diagram

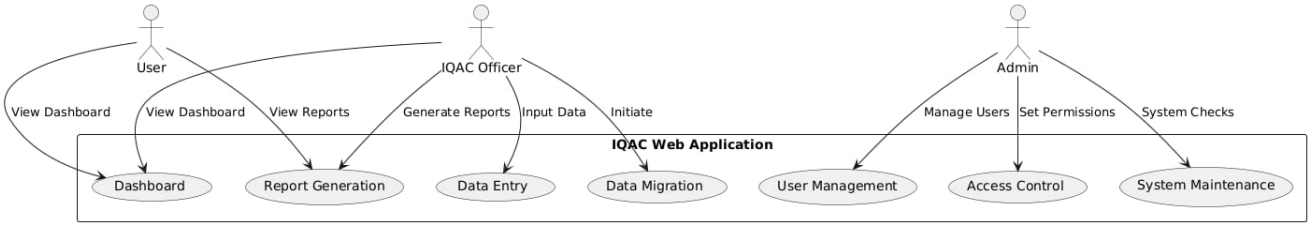

Figure 3.1: Process Flow Diagram

3.1.2 Architecture Diagram

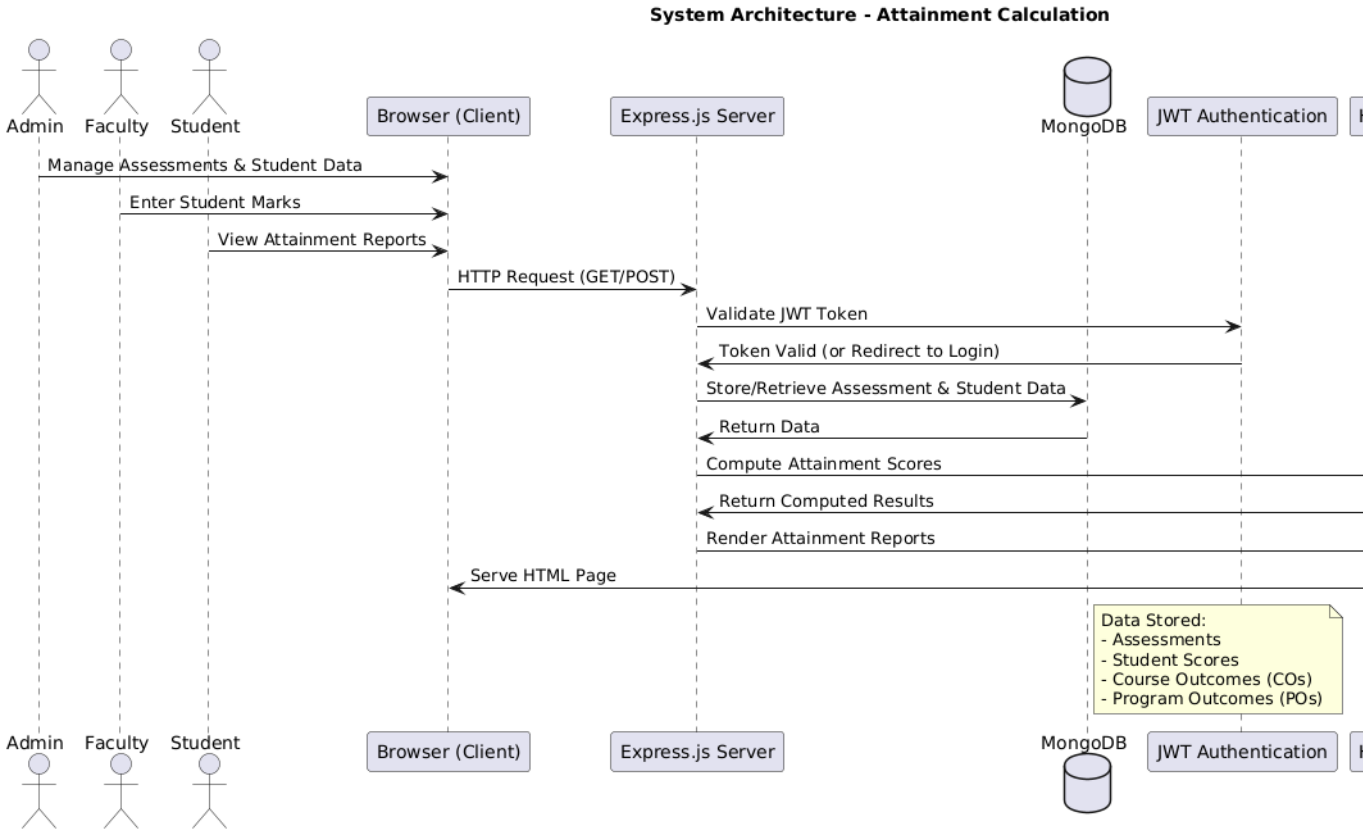

Figure 3.2: Architecture Diagram

3.1.3 Use case Diagram

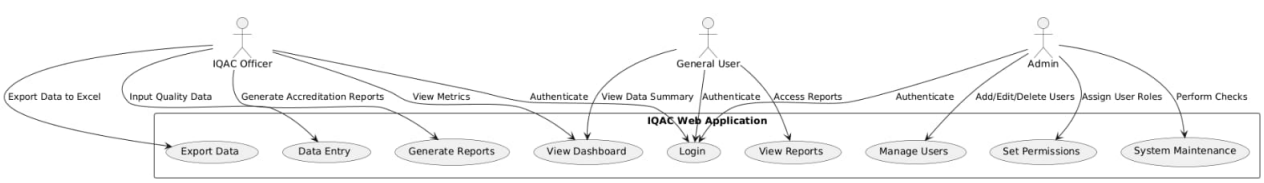

Figure 3.3: Use case diagram

3.1.4 ER diagram

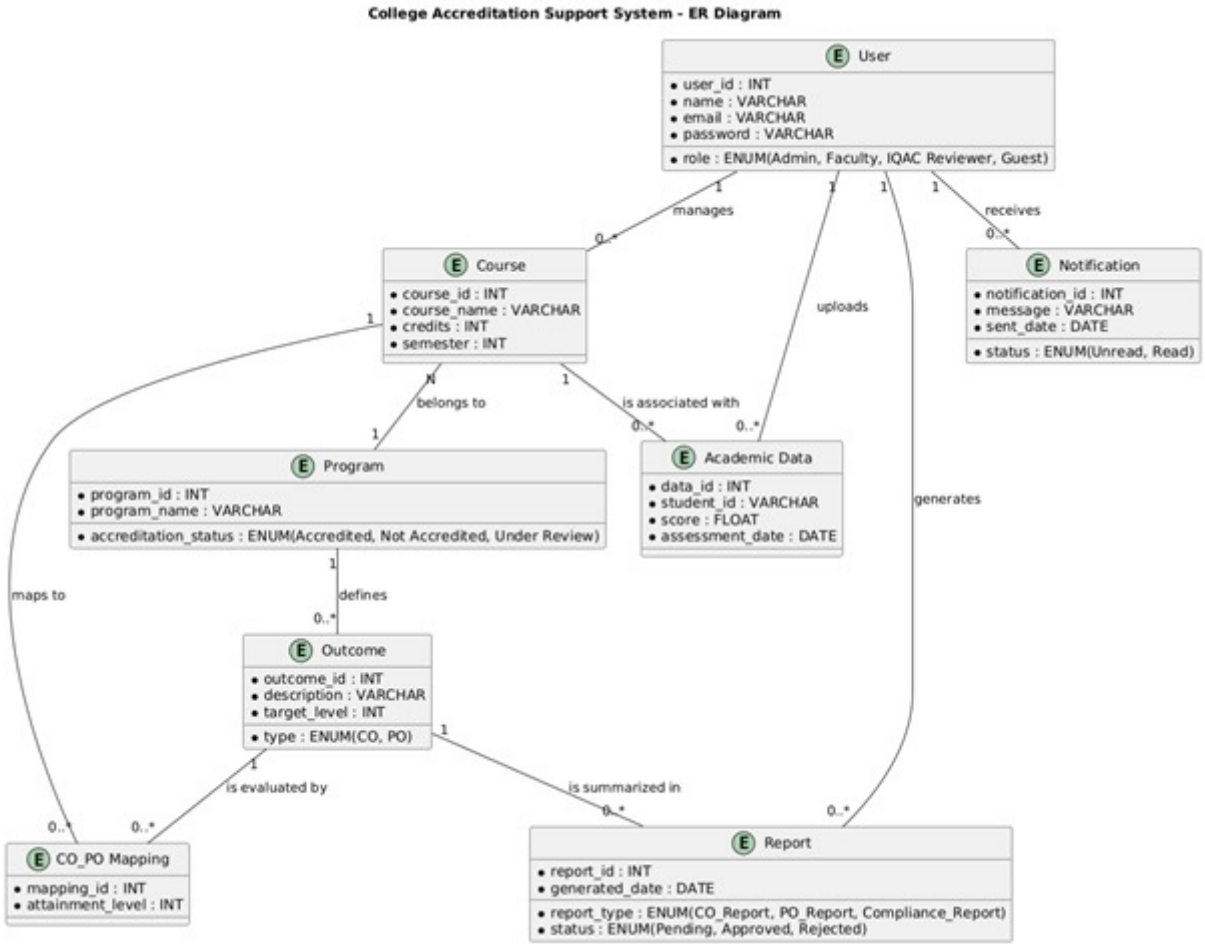

Figure 3.4: ER diagram

3.1.5 Class Diagram

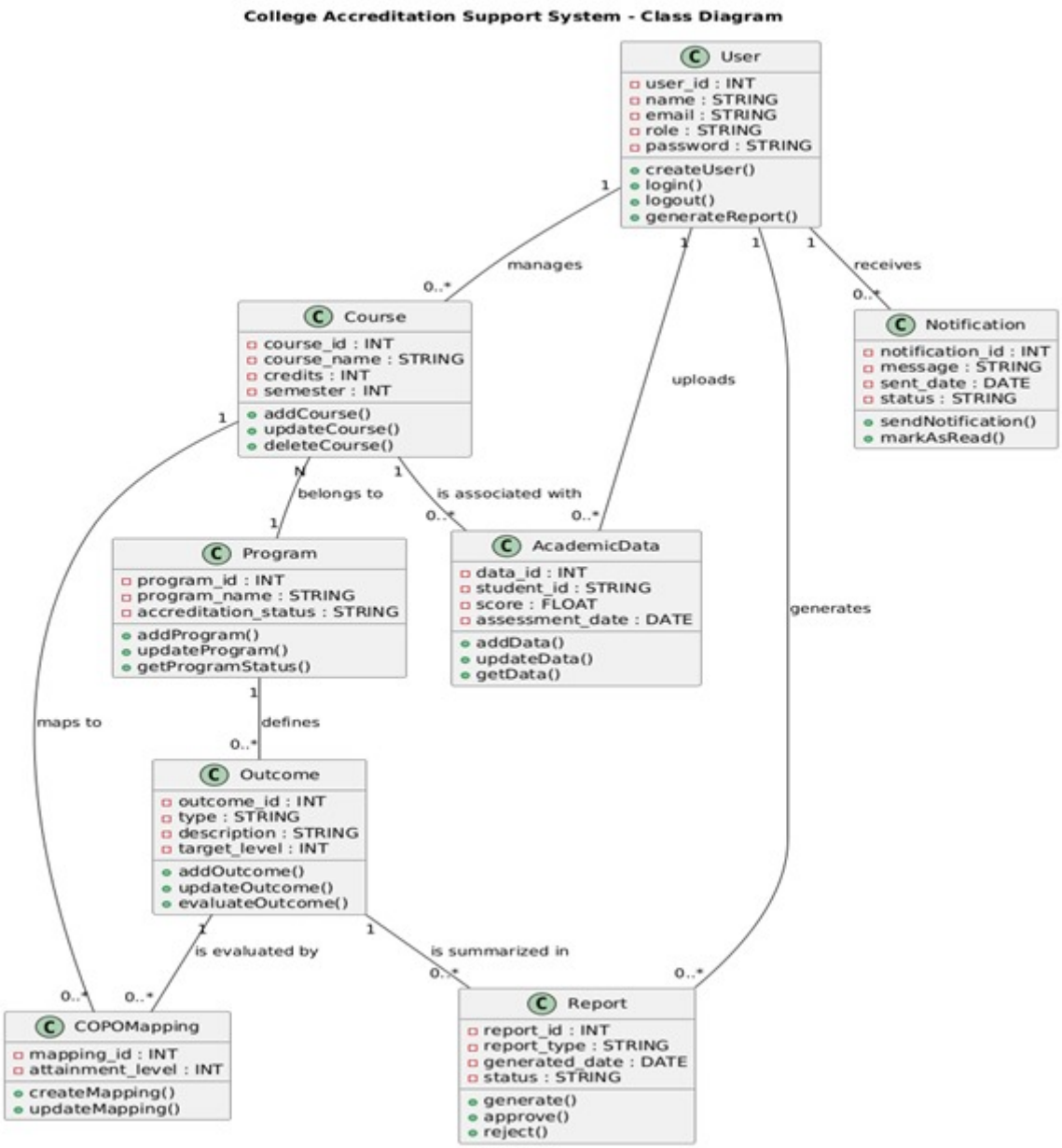

Figure 3.5: Class diagram

3.2 System Requirements

3.2.1 Hardware Requirements

The Campus View software requires a system with an Intel i5 processor or higher (or an AMD equivalent) to ensure smooth performance. A minimum of 8GB RAM is needed for basic operations, though 16GB is recommended for better efficiency. The application requires at least 256GB of SSD storage, but 512GB SSD is preferable for faster data access and processing.

3.2.2 Software Requirements

The application runs on Windows 10/11, macOS, or Linux (Ubuntu) operating systems.The software platforms used in this system are Express.Js,MongoDB,Node.Js and Version control is maintained through GitHub to track development changes.

3.2.3 GitHub

GitHub is a web-based platform for version control and collaboration, allowing developers to manage and track changes in their code efficiently. It is built on Git, a distributed version control system, enabling multiple developers to work on the same project without conflicts. GitHub provides features like repositories, where code is stored and managed,branches for parallel development, and pull requests for merging changes. It supports collaborative coding, issue tracking, and CI/CD pipelines for automated testing and deployment. With integration options for various development tools, GitHub is widely used in open-source projects, corporate software development, and academic research, making it a crucial tool for modern software engineering.

3.2.4 MongoDB

MongoDB is a popular open-source NoSQL database that uses a document-oriented data model. Instead of tables and rows like in traditional relational databases, MongoDB stores data in flexible, JSON-like documents with dynamic schemas, making it easy to store and retrieve complex data structures. It is designed for scalability and high performance, capable of handling large volumes of data and high throughput applications. MongoDB offers features such as automatic sharding for horizontal scaling, replica sets for high availability, and rich query capabilities, making it suitable for a wide range of use cases, including web and mobile applications, real-time analytics, and content management systems.

3.2.5 Node.js

Node.js is an open-source, server-side JavaScript runtime environment built on Chrome's V8 JavaScript engine. It allows developers to execute JavaScript code outside of a web browser, enabling them to build scalable, high-performance network applications. Node.js uses an event-driven, non-blocking I/O model, making it lightweight and efficient for handling concurrent connections and asynchronous operations. It has a vast ecosystem of packages available through npm (Node Package Manager), providing developers with a wide range of tools

and libraries for building various types of applications, including web servers, APIs, microservices, and real-time applications. Node.js is widely used in modern web development due to its speed, scalability, and versatility.

3.3 System Design

3.3.1 Module Design

Module 1: User Management

Description: Handles authentication, role-based access, and user profile management.

Technologies Used: Node.js, Express.js, JWT Authentication, MongoDB.

Functionality:

- Admin, Faculty, and Student roles.
- JWT-based authentication for secure access.
- CRUD operations on user profiles.

Module 2: Assessment Management

Description: Allows faculty to manage assessments and enter student marks.

Technologies Used: Node.js, MongoDB, Handlebars (HBS).

Functionality:

- Faculty can create/edit/delete assessments.
- Students' scores are stored and retrieved securely.

Module 3: Attainment Calculation

Description: Computes attainment scores for Course Outcomes (COs) and Program Outcomes (POs).

Technologies Used: Node.js, MongoDB, Attainment Calculator.

Functionality:

- Fetch student marks and course data.
- Compute attainment scores based on predefined formulas.
- Store and retrieve computed results.

Module 4: Reporting and Visualization

Description: Generates attainment reports and insights for accreditation.

Technologies Used: Handlebars (HBS), Express.js, MongoDB.

Functionality:

- Render attainment reports for Admin, Faculty, and Students.
- Support data visualization for insights.

3.3.2 Database Design

Database Used: MongoDB

Collections:

- **Users** (Stores Admin, Faculty, and Student data)
- **Assessments** (Stores test/exam details)
- **Scores** (Stores student marks)
- **AttainmentResults** (Stores computed attainment scores)
- **CourseOutcomes** (Stores CO-PO mapping data)

Schema Structure: Each collection contains structured documents with references for efficient querying.

3.3.3 Interface Design

Frontend: Browser-based interface using Handlebars (HBS).

3.4 Implementation Details

Implementing **Campus View** requires a well-structured approach, integrating both technical and administrative aspects to ensure seamless accreditation data management and student classification.

Firstly, for system architecture, the frontend is developed using **React.js** to provide an interactive and responsive user interface, while the backend is built using **Node.js with Express.js** to handle API requests efficiently. The database is managed using **MongoDB**, allowing structured storage and retrieval of student performance data, CO-PO Attainment records, and feedback analysis.

Secondly, regarding data security and access control, **Role-Based Access Control (RBAC)** is implemented to ensure restricted access to sensitive data. Authentication is secured using **JWT (JSON Web Token)**.

Thirdly, system deployment and maintenance involve hosting the backend on **AWS EC2** for scalable performance. A **Continuous Integration/Continuous Deployment (CI/CD)** pipeline ensures automated testing and seamless deployment of new features without disrupting ongoing operations.

Fourthly, the testing phase is crucial for ensuring system reliability. **Unit testing** is performed using **Jest and Mocha** to verify individual components, while **integration testing** ensures smooth interaction between the frontend, backend, and database. **Validation testing** is conducted to confirm the accuracy of CO-PO Attainment calculations and student classification algorithms. Additionally, **performance testing** using **Apache JMeter** ensures the system can handle high user loads effectively.

Finally, collaboration with institutional authorities plays a vital role in the successful implementation of **Campus View**. Working closely with faculty and accreditation committees helps in aligning the software with NAAC/NBA accreditation requirements. Regular feedback sessions ensure continuous improvements, making the system more efficient and user-friendly for academic institutions.

3.5 Testing

Testing is a crucial phase in the development of **Campus View**, ensuring that all components function correctly and meet the specified requirements. Various testing methodologies are employed to verify the accuracy, security, and performance of the system. The goal is to identify and resolve errors early, ensuring a smooth user experience and compliance with accreditation requirements.

3.5.1 Test Cases

System testing ensures that all individual modules function as expected and integrate seamlessly. Analysts define test conditions, generate test data, execute tests, and compare actual results with expected outcomes. Additionally, users are involved in testing to validate real-world usage scenarios. Testing is conducted experimentally to identify issues before deployment, ensuring reliability. A limited number of users interact with the system to detect unexpected behaviors, while independent testers verify software quality for unbiased results.

Parallel testing is conducted as a final validation step, running the new system alongside existing methods to ensure accuracy and reliability. The primary aim is to detect errors, validate design choices, and confirm system functionality before full implementation. Testing phases include various methodologies to guarantee system robustness.

3.5.2 Unit Testing

No.	Test Case	Input Data	Expected Result	Actual Result	Remark
1	Verify CO-PO mapping accuracy	Course-wise student performance data	CO-PO attainment is correctly calculated	Calculations matched expected values	Success
2	Verify student classification	Student scores in different courses	Students are categorized as weak or bright based on defined criteria	Classification was accurate based on thresholds	Success

3	Verify feedback analysis	Student/Faculty feedback data	Relevant insights are generated for curriculum improvement	Insights were displayed correctly	Success
4	Verify report generation	Request for NAAC/NBA report	Report is generated with correct attainment values	Report was generated with accurate data	Success
5	Verify data visualization	View attainment graphs	Charts display correct CO-PO trends	Graphs updated correctly with real-time data	Success
6	Verify data import/export	Upload student performance data	System processes data and allows export in required format	Data was successfully uploaded and exported	Success
7	Verify role-based access	Login as admin/faculty/student	Each user sees only relevant data and features	Users were restricted to their respective roles	Success

Unit testing involves testing individual components of the application to ensure they function correctly. Each module of **Campus View** is tested independently to verify expected output. This step includes validation checks to confirm data integrity and ensure compliance with accreditation standards. Unit testing is conducted continuously during development to identify and fix errors early, improving overall system stability.

3.5.3 Integration Testing

Integration testing verifies that different modules of the system interact correctly. Individual components, previously tested in isolation, are combined and tested as a group. This phase ensures smooth communication between various modules, such as student classification, feedback analysis, and CO-PO attainment calculations. Testing is conducted iteratively

to identify interface issues and improve overall system functionality.

3.5.4 Validation Testing

Validation testing confirms that the final software package meets the intended requirements. The system undergoes black-box testing to ensure all user functionalities work as expected. The outcome of validation testing determines whether the system meets accreditation needs or requires further improvements. Any deviations from the expected performance are documented and addressed before deployment.

3.5.5 User Acceptance Testing

User acceptance testing (UAT) is the final phase before deployment, where end users test the system in real-world conditions. Users validate the system's ability to handle actual data and workflows, ensuring alignment with accreditation and institutional requirements. The testing process involves running predefined scenarios, confirming that the system meets business needs. Upon successful completion, **Campus View** is approved for full-scale deployment in the institution.

Chapter 4

CONCLUSION

The conversion of a macro-enabled Excel application for IQAC operations into a web application using Node.js has successfully streamlined the processes essential for NAAC and NBA accreditations. By moving away from the constraints of desktop applications, this project has enabled centralized access, scalability, and improved efficiency in data management for IQAC operations. The web-based solution not only supports enhanced data security and backup but also simplifies collaboration and data-sharing among stakeholders.

This transformation has significantly reduced manual intervention, minimized the risk of errors, and improved the overall user experience, allowing institutions to meet accreditation requirements more efficiently. As a future scope, the web app can be further extended to include advanced data analytics and reporting tools, providing deeper insights into quality parameters and helping institutions make data-driven decisions.

References

[1] Kamburugamuve, S., Fox, G., Yasaweerasinghelage, R. (2018). A Survey on Serverless Computing. Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), 2561–2570.

[2] Bauer, M., Adams, N. (2019). Mastering Node.js: Building Web Applications with Express, MySQL, and JavaScript. Packt Publishing.

[3] Microsoft (2021). Guidelines for Data Migration from Excel to Web Applications. Microsoft Documentation

[4] Chaudhary, S., Gautam, A., Sharma, S. (2022). Data Security in Web Applications: A Comprehensive Study. International Journal of Computer Applications, 184(18), 25–31.

[5] National Assessment and Accreditation Council (NAAC). (2023). Manual for Self-Study Report (SSR) – Affiliated/Constituent Colleges. Retrieved from NAAC Website.

[6] Accreditation and Assessment for Educational Institutions: Guidelines and Practices. (2023). NBA Handbook for Accreditation Process. NBA Publications.

Appendix A

Jira Reports

Figure A.1: Burn up

Figure A.2: Burn down

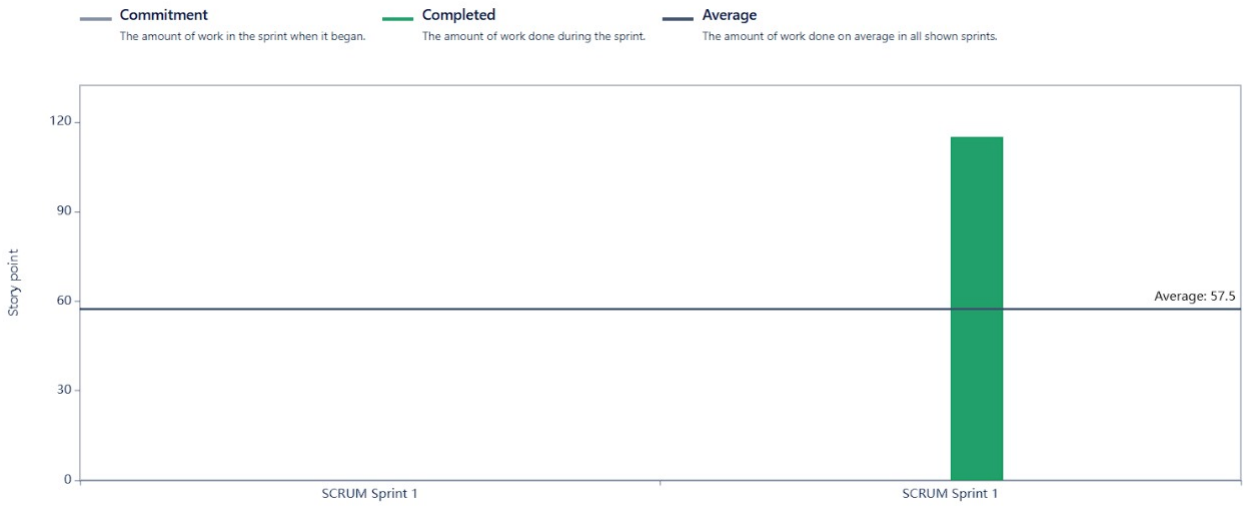

Figure A.3: Velocity

Appendix B

Sample Code

```
-----------------------------
Attaintement Calculation:
-----------------------------
const Assessment = require('../models/assessmentSchema');

async function calculateAssessment(assessmentId) {
    try {
        const assessment = await Assessment.findById(assessmentId);
        if (!assessment) {
            return { message: "Assessment not found!" };
        }

        let coScores = {};
        let coTotalMarks = {};

        // Iterate through tools and questions
        assessment.tools.forEach(tool => {
            tool.questions.forEach(question => {
                const coNumber = question.coNumber;
                const maxMarks = parseFloat(question.maxMarks) || 0;

                if (!coScores[coNumber]) {
                    coScores[coNumber] = 0;
                    coTotalMarks[coNumber] = 0;
                }
```

```
                coScores[coNumber] += maxMarks;
                coTotalMarks[coNumber] += 10;
                // Assuming CO attainment is scaled to 10
            });
        });

        // Compute CO Attainment Percentage
        let coAttainment = {};
        Object.keys(coScores).forEach(co => {
            coAttainment[co] = ((coScores[co] / coTotalMarks[co]) * 100)
            .toFixed(2) + "%";
        });

        return {
            assessmentId,
            coAttainment
        };

    } catch (error) {
        console.error("Error in assessment calculation:", error);
        return { message: "Internal Server Error", error: error.message };
    }
}

module.exports = calculateAssessment;
```

Attaintement Tool Generation

```
router.post('/submit',authenticateToken, upload.none(),
async (req, res) => {
```

```
try {
  if (!req.body.tool || !Array.isArray(req.body.tool) ||
  req.body.tool.length === 0) {
    return res.status(400).json(
    { message: "Invalid assessment tools data." });
  }

  // Utility function to ensure arrays are properly formatted
  const parseArray = (data) => {
    if (!data) return [];
    if (Array.isArray(data)) return data;
    return Object.values(data);
    // Convert object to array (Fixes multer issue)
  };

  const finalData = {
    branch: req.body.branch || "",
    subject: req.body.subject || "",
    semester: req.body.semester || "",
    courseCode: req.body.courseCode || "",
    batch: req.body.batch || "",
    numCOs: req.body.numCOs || 0,
    numStudents: req.body.numStudents || 0,
    numPOs: req.body.numPOs || 0,
    numPSOs: req.body.numPSOs || 0,
    universityExam: req.body.universityExam || "No",
    assessmentYear: req.body.assessmentYear || "Unknown",
    facultyName: req.body.facultyName || "Unknown",
    tools: req.body.tool.map((tool, index) => {
      const questions = parseArray(req.body.questions?.[index]);
      const coNumbers = parseArray(req.body.coNumber?.[index]);
      const bloomsTaxonomy = parseArray(req.body.bloomsTaxonomy?.[index]);
      const maxMarks = parseArray(req.body.maxMarks?.[index]);

      return {
        toolName: tool,
```

```
            questions: questions.map((q, i) => ({
              questionNumber: q || "N/A",
              coNumber: coNumbers[i] || "N/A",
              bloomsTaxonomy: bloomsTaxonomy[i] || "N/A",
              maxMarks: maxMarks[i] || "0",
            })),
          };
        }),
      };

      // Save to MongoDB
      try {
        const newAssessment = new Assessment(finalData);
        await newAssessment.save();

        console.log("Final Data Saved:", JSON.stringify(finalData, null, 2));
        res.status(201).json({ message: "Data successfully saved!" });
      } catch (error) {
        console.error("Error saving data:", error);
        res.status(500).json({ message: "Internal Server Error",
        error: error.message });
      }

    } catch (error) {
      console.error("Error saving data:", error);
      res.status(500).json({ message: "Internal Server Error",
      error: error.message });
    }
});
```

```
------------------------------
Add student data
------------------------------
```

```
router.post("/step2", authenticateToken,async function (req, res) {
```

```
try {
  const data = req.body;
  console.log(data);

  if (!data || !data.students) {
    return res.status(400).json({ error: "Invalid data format" });
  }

  // Convert students object into an array
  const students = Object.values(data.students).map((student) =>
  ({
    name: student.name,
    roll: student.roll,
    tools: Object.entries(student.tools).map(([toolName, toolData]) =>
    ({
      toolName: toolName,
      questions: toolData.questions.map((q) => ({
        questionNumber: q.questionId,
        coNumber: q.co,
        bloomsTaxonomy: q.blooms,
        maxMarks: q.maxMarks,
        marksObtained: q.marksObtained, // Store marks obtained
      })),
    })),
  }));

  // Create and save in AssessmentRecords collection
  const assessmentRecord = new AssessmentRecord({
    branch: data.branch,
    subject: data.subject,
    semester: data.semester,
    courseCode: data.courseCode.trim(),
    batch: data.batch,
    numCOs: data.numCOs || "N/A",
    numStudents: data.numStudents || students.length.toString(),
    // Auto count students
```

```
      numPOs: data.numPOs || "N/A",
      numPSOs: data.numPSOs || "N/A",
      universityExam: data.universityExam || "N/A",
      assessmentYear: data.assessmentYear || "N/A",
      facultyName: data.facultyName || "N/A",
      students: students, // Store student data
    });

    await assessmentRecord.save(); // Save to MongoDB

    res.status(201).json({ message:
    "Assessment record saved successfully!" });
  } catch (error) {
    console.error("Error saving data:", error);
    res.status(500).json({ error: "Internal server error" });
  }
});

-----------------------
Registration
-----------------------

router.post('/register', upload.none(), async (req, res) => {
  console.log(req.body);

  try {
    const { email, password } = req.body;

    // Validate input
    if (!email || !password) {
      return res.status(400).json({ message:
      'Username and password are required.' });
    }

    // Check if the username already exists
    const existingUser = await User.findOne({ email });
```

```
    if (existingUser) {
      return res.status(409).json({ message: 'Username already exists.' });
    }

    // Hash the password
    const hashedPassword = await bcrypt.hash(password, 10);

    // Create a new user
    const newUser = new User({ email, password: hashedPassword });
    await newUser.save();

    // Generate a JWT token for the new user
    const token = jwt.sign(
      { username: newUser.email }, // Payload
      process.env.JWT_SECRET, // Secret key
      { expiresIn: '1h' } // Token expiration time
    );

    res.status(201).json({ token, message:
    'User registered successfully!' });
  } catch (error) {
    console.error('Error registering user:', error);
    res.status(500).json({ message:
    'Internal Server Error', error: error.message });
  }
});

-----------------------
Login
-----------------------

router.post('/login', async (req, res) => {
  const { email, password } = req.body;

  try {
    // Find the user in the database
```

```
    const user = await User.findOne({ email });
    if (!user) {
      return res.status(401).json({ message:
      'Invalid email or password.' });
    }

    // Compare the provided password with the hashed password
    const isPasswordValid = await bcrypt.compare(password, user.password);
    if (!isPasswordValid) {
      return res.status(401).json({ message:
      'Invalid email or password.' });
    }

    // Generate a JWT token
    const token = jwt.sign(
      { email: user.email }, // Payload
      process.env.JWT_SECRET, // Secret key
      { expiresIn: '1h' } // Token expiration time
    );

    // Set the token in an HTTP-only cookie
    res.cookie('token', token, {
      httpOnly: true,
      secure: process.env.NODE_ENV === 'production',
      maxAge: 60 * 60 * 1000, // 1 hour
    });

    // Add the token to the response headers
    res.setHeader('Authorization', `Bearer ${token}`);

    res.redirect('/')
  } catch (error) {
    console.error('Error during login:', error);
    res.status(500).json({ message: 'Internal server error.' });
  }
});
```

Appendix C

Screenshots

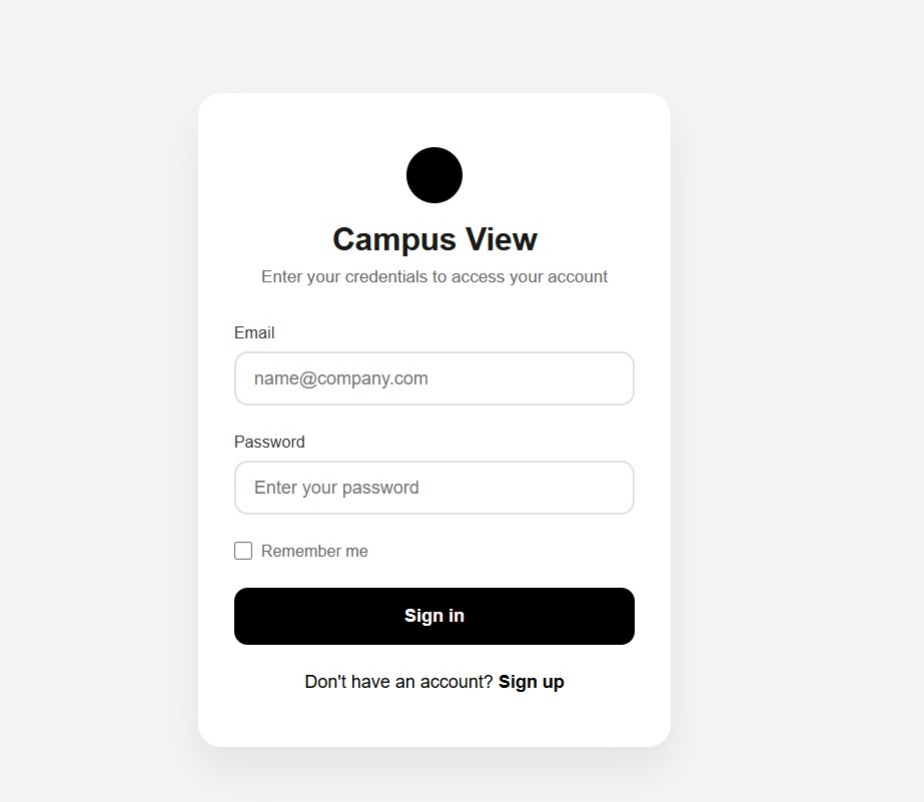

Figure C.1: Registration Page

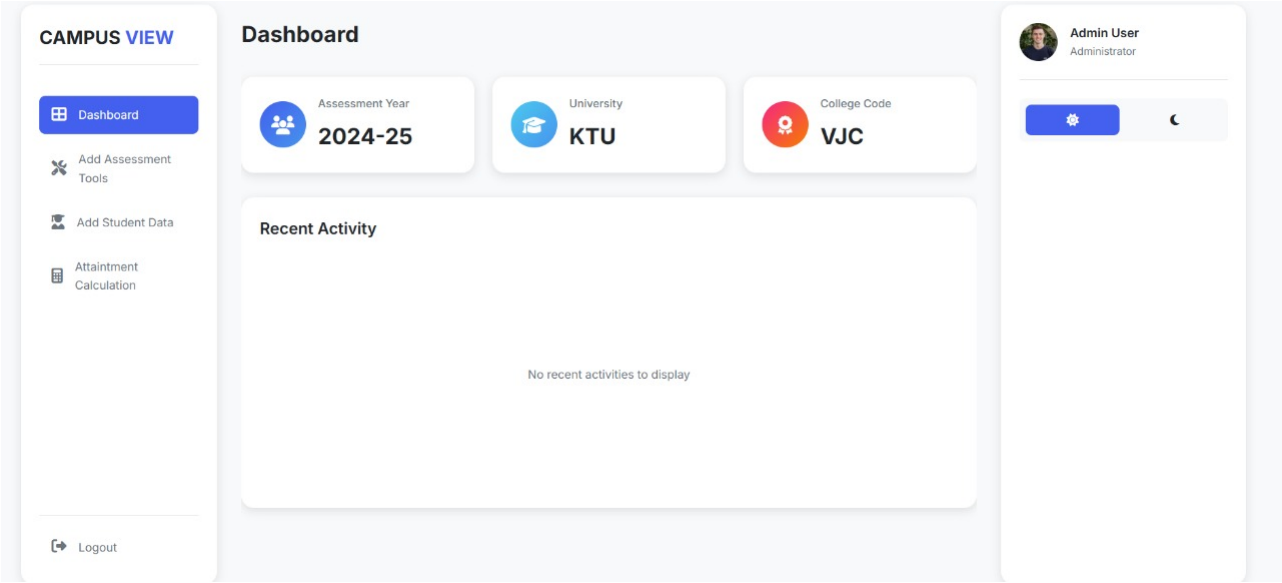

Figure C.2: Dashboard

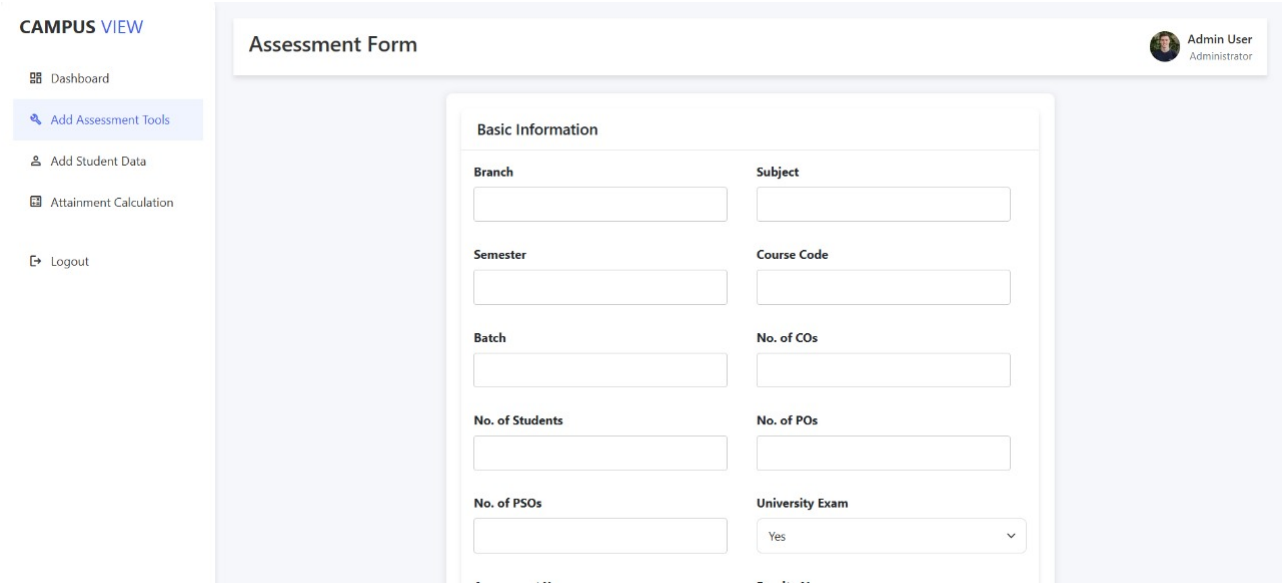

Figure C.3: Add assessment tools

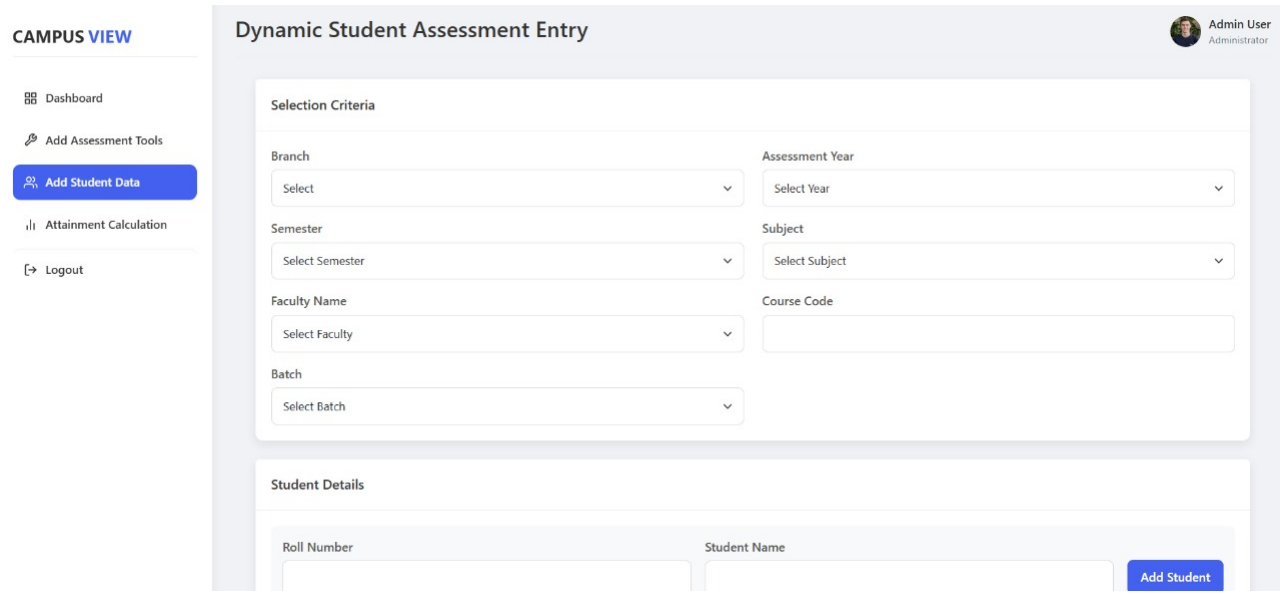

Figure C.4: Add student data

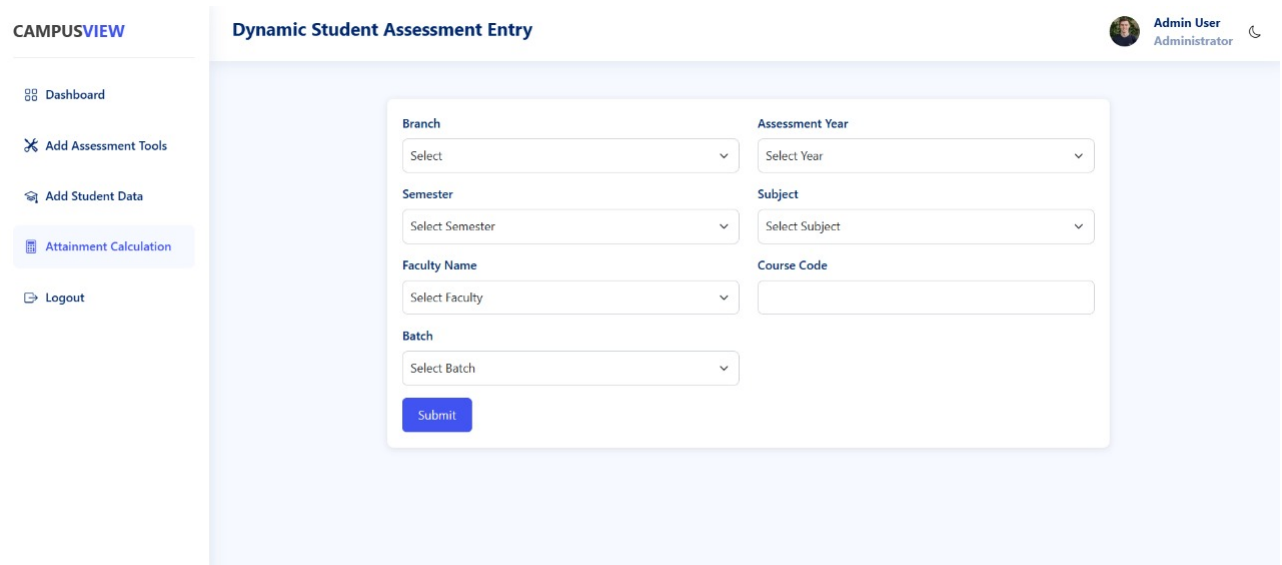

Figure C.5: Attainment calculation

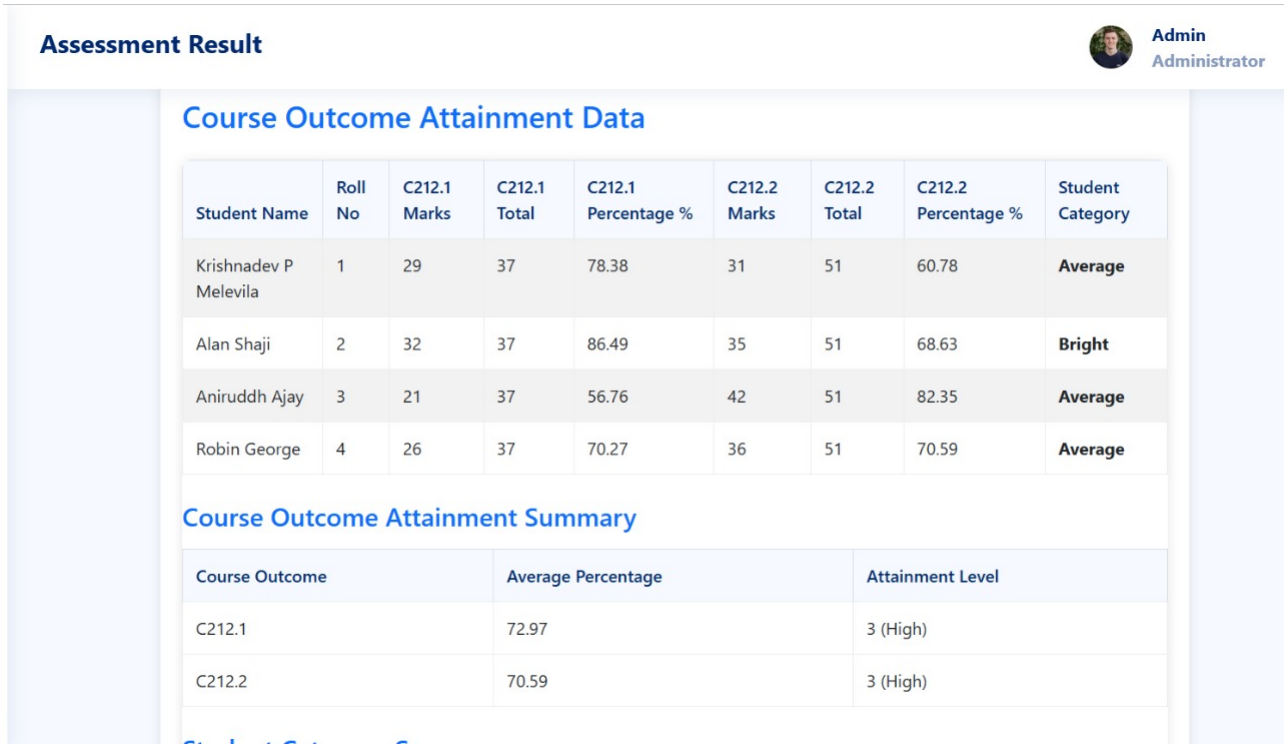

Figure C.6: Attainment summary

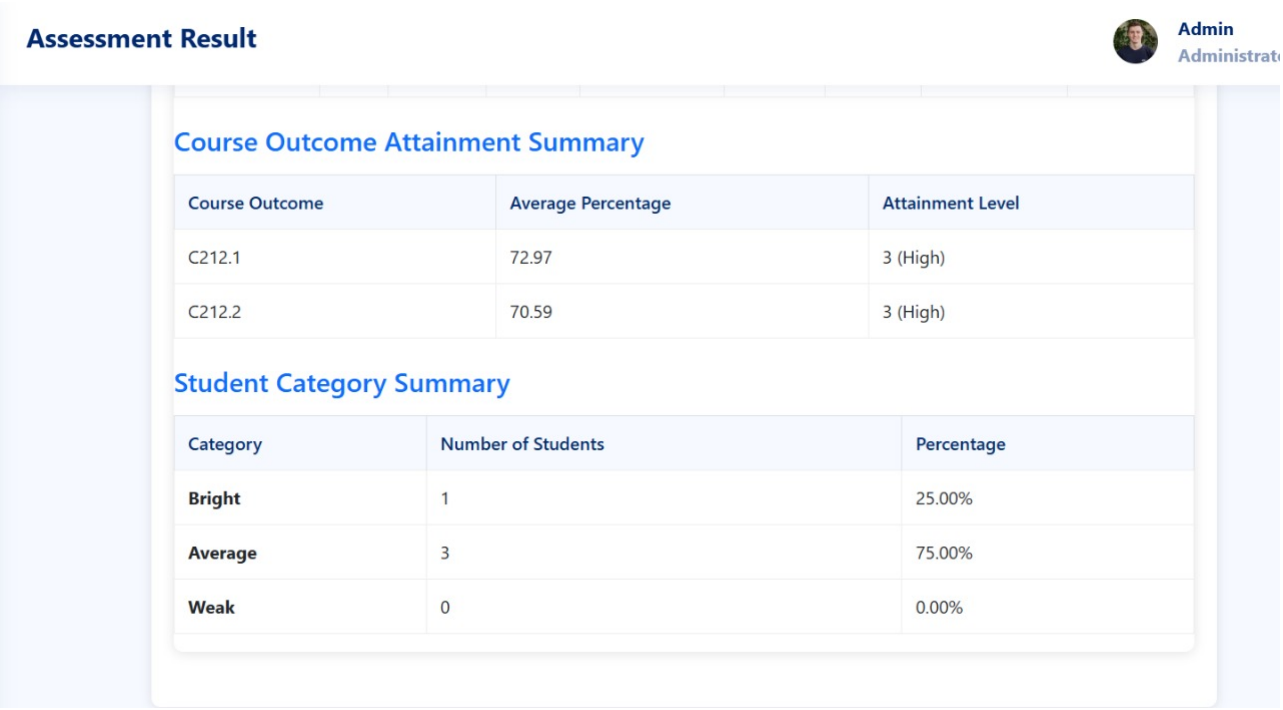

Figure C.7: Attainment summary