A

Project Report

On

SMART COLLEGE SYSTEM SOFTWARE USING DATA SCIENCE

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Abhinav Verma

(2000040100003)

Aditya Kumar Singh

(2000040100010)

Under the guidance of

Er. Alok Singh Jadaun

COMPUTER SCIENCE & ENGINEERING

RAJA BALWANT SINGH ENGINEERING TECHNICAL CAMPUS

BICHPURI, AGRA

Affiliated to Dr. A.P.J. Abdul Kalam Technical University (Formerly known as UPTU), Lucknow

DECLARATION

We declare that the project work presented in this report entitled "**SMART COLLEGE SYSTEM SOFTWARE USING DATA SCIENCE**", under the guidance of "**ER. ALOK SINGH JADAUN**" submitted to the Computer Science and Engineering Department, Raja Balwant Singh Engineering Technical Campus, Agra, affiliated to Dr.A.P.J. Abdul Kalam Technical University (Formerly Known as U.P.T.U), Lucknow" in the academic session 2023-2024 for the award of the Bachelor of Technology degree in ***Computer Science& Engineering,*** in our original work. We have not plagiarized or submitted the same work for award of any other degree.

May, 2024	Abhinav Verma (2000040100003)
Agra	Aditya Kumar Singh (2000040100010)

CERTIFICATE

This is to certify that the Project entitled "**SMART COLLEGE SYSTEM SOFTWARE USING DATA SCIENCE**" has been submitted by **Abhinav Verma(2000040100003), Aditya Kumar Singh(2000040100010), VIII Sem.,** in partial fulfillment of the degree of Bachelor of Technology in Computer Science & Engineering of, Raja Balwant Singh Engineering Technical Campus, affiliated to Dr. A.P.J. Abdul Kalam Technical University (Formerly Known as U.P.T.U.), Lucknow" in the academic session 2023-24.

May, 2024

Agra

(Prof. Brajesh Kumar Singh)	(Er. Alok Singh Jadaun)
HOD, CSE	Assistant Professor, CSE Dept.

ACKNOWLEDGEMENT

We would like to acknowledge the contributions of numerous individuals who have supported us in the completion of this report. Without their kind assistance, this project would not have been possible. We are deeply grateful to Dr. Brajesh Kumar Singh (HOD) and Er. Alok Singh Jadaun (Project Guide) for their invaluable guidance, supervision, and provision of essential information. Their constant support has been instrumental in completing this project successfully. We would also like to express our heartfelt thanks to Dr. Brajesh Singh Kushwah, the Director of our esteemed institution, for his unwavering efforts in upholding its character and reputation. We are also thankful to our parents and faculty members for their cooperation and encouragement throughout the project. Additionally, we would like to acknowledge the contributions of our colleagues who were involved in the development of this project and otherindividuals who have assisted with their skills and abilities.

Abhinav Verma (2000040100003)

Aditya Kumar Singh (2000040100010)

ABSTRACT

In the contemporary educational landscape, the efficient management of academic institutions has become increasingly complex, demanding streamlined processes and robust technological solutions. Smart College System Software emerges as a pivotal tool in addressing these challenges, offering comprehensive solutions for various administrative tasks. This abstract delves into the significance, and functionalities in modern educational settings.

Smart College System Software encompasses a suite of integrated applications designed to automate and streamline administrative processes within educational institutions. From admissions and enrollment management to academic scheduling, grading, and financial management, these software systems centralize data, optimize workflows, and enhance communication among stakeholders. Moreover, they often include modules for student information management, faculty administration, library management, and alumni relations, fostering an interconnected ecosystem conducive to academic excellence and institutional efficiency.

The implementation of Smart College System Software yields numerous benefits for educational institutions. By automating routine administrative tasks, it frees up time and resources for strategic initiatives and academic pursuits. Enhanced data accuracy and accessibility promote informed decision-making and improve institutional performance metrics. Moreover, the centralized nature of these systems fosters collaboration and coordination across departments, fostering a cohesive academic environment. Students benefit from streamlined processes, timely access to information, and improved communication channels, enhancing their overall educational experience.

Table of Content

List of Figures

CHAPTER 1

1. INTRODUCTION, OBJECTIVE & SCOPE

1.1 Introduction

Smart College System Software is a software-based application that is the new technical way to manage all department-related jobs. Smart Collage system software is helpful for students as well as the colleges. In the existing system, all the activities are done manually.It is very costly and time-consuming. In our proposed system, students can view results by using SMTP. The data will be stored on the college server. To store the data MySQL server will be used. The Admin, Faculty, or the student should be a registered user. The faculty can log into their college account through the software itself and update the academic results like internal exam marks obtained by the students. In this system students have easy access to viewing the marks, The application will check user authentications. Students are not permitted to manipulate any data. The proposed work has two modules: A. Student B. Teacher C. Admin. In the student module, students need to register their university registration number, college registration number, and student name. Admin module maintains the student's marks on internal college exams. Student's attendance is also monitored by the software. The design and implementation of the system is to provide service in institutes and colleges. The system is to provide a comprehensive student information system and the user interface is to replace the current paper records. College Staff uploads attendance, and results, and shares subject notes and college notifications through a secure, online interface using Android devices. All data will be stored on the college server and validated on the server before actual record alteration occurs. The system plans for a student interface, allowing students to access tips and tricks provided by their seniors. All data is stored securely on MySQL servers managed by the college Administrator. This will decrease the paperwork and time needed to access student records. Previously, colleges relied heavily on paper records for this initiative which had its disadvantages. This system provides a simple interface for the maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. Achieving this objective is difficult to use a manual system as the information is scattered, can be redundant and collecting relevant information may be very time-consuming.

This feedback can be reviewed by the admin or management committee of the institute.

As technology becomes more and more integrated into our daily lives in the digital age, educational institutions are looking more and more for clever solutions to improve their operations and academic programs. One such remedy is the Smart College System Software, an all-inclusive platform created to completely transform the way universities oversee their academic, administrative, and student-related operations. This introduction gives a general overview of the Smart College System Software, emphasizing its importance, salient characteristics, and possible influence on the state of education today.

Smart College System Software represents a paradigm shift in college administration, offering a centralized platform that integrates various modules to streamline processes and improve efficiency. From admissions and enrollment management to academic scheduling, grading, and financial administration, this software provides a one-stop solution for the diverse needs of educational institutions. By leveraging advanced technologies such as cloud computing, data analytics, and mobile connectivity, Smart College System Software empowers colleges to enhance productivity, transparency, and accountability across all levels of operation. In conclusion, Smart College System Software represents a transformative solution for modernizing college administration and delivering a seamless educational experience. By harnessing the power of technology, colleges can overcome the challenges of traditional administrative practices and embrace a future-oriented approach to academic management. As educational institutions continue to adapt to the evolving needs of students and stakeholders, Smart College System Software emerges as a cornerstone of innovation, efficiency, and excellence in higher education. Smart College System Software is not merely a tool for administrative convenience; it is a catalyst for institutional transformation. By streamlining operations and fostering data-driven decision-making, it empowers colleges to adapt to changing market dynamics, regulatory requirements, and student expectations with agility andresilience.

The modular nature of Smart College System Software ensures scalability and customization to meet the unique needs of each educational institution. Whether managing a small liberal arts college or a large research university, administrators can tailor the software to align with their specific workflows, policies, and objectives, maximizing its effectiveness and utility. Beyond operational efficiency, Smart College System Software contributes to a culture of innovation and collaboration within academic communities. By facilitating communication and knowledge sharing among students, faculty, and administrators, it cultivates an environment conducive to creativity, experimentation, and continuous improvement.

The implementation of Smart College System Software is not without challenges, including initial investment costs, data migration complexities, and user adoption barriers. However, the long-term benefits far outweigh these challenges, as evidenced by increased productivity, cost savings, and stakeholder satisfaction experienced by institutions that embrace this technology-driven approach to college management.

As the educational landscape continues to evolve in response to technological advancements and changing societal needs, Smart College System Software will play an increasingly pivotal role in shaping the future of higher education. Empowering colleges to innovate, collaborate, and adapt in a rapidly changing environment, ensures that they remain relevant, resilient, and responsive to the needs of students and society at large.

1.2 Objective

College System software, also known as campus management software or college system software, is designed to streamline and automate various administrative and academic processes within educational institutions. The primary objectives of using college management software include:

1) Efficient Administrative Operations:

   Automating administrative tasks such as admissions, student registration, fee collection, and payroll management to reduce paperwork and minimize manual errors. Streamlining the management of resources, facilities, and infrastructure within the college.

2) Student Information Management:

   Maintaining a centralized database of student information, including personal details, academic records, attendance, and assessment results.

3) Academic Planning and Scheduling:

   Creating and managing class schedules, timetables, and exam schedules to optimize resource allocation and ensure a balanced academic calendar. Allowing students and faculty to access their schedules online.

4) Communication and Collaboration:

   Facilitating communication between students, faculty, and parents through a centralized platform, including email notifications, announcements, and messaging. Enabling collaborative tools for teachers and students, such as discussion forums, assignment submissions, and online grading.

5) Financial Management:

   Managing the financial aspects of the institution, including fee collection, expense tracking, and budget management. Generating financial reports and statements for auditing and analysis.

6) Admissions and Enrolments:

Simplifying the admissions process by offering online application forms and documentsubmission. Automating the evaluation and selection of candidates and generating admission offer letters.

7) Attendance and Performance Tracking:

Tracking student attendance and generating reports for monitoring and intervention. Recording and analyzing student performance to identify areas for improvement.

8) Library Management:

Automating library functions, such as cataloging books, managing checkouts and returns, and tracking overdue items.

Providing online access to the library catalog and resources.

9) Security and Access Control:

Ensuring data security and privacy through user access controls and data encryption. Monitoring and tracking access to sensitive information.

10) Reporting and Analytics:

Generating various reports and analytics to help administrators make data-driven decisions for improving the institution's performance. Monitoring key performance indicators (KPIs) and trends in student outcomes and administrative processes.

11) Alumni Management:

Maintaining a database of alumni and their contact information for networking and engagement. Managing alumni events and donations.

Overall, the primary objectives of college management software are to enhance the efficiency, transparency, and effectiveness of educational institutions in managing their administrative and academic processes, ultimately leading to a better learning environment for students and improved administrative control .

1.3 Scope

The scope of college management software is broad and encompasses various aspects of educational institution management. Its scope includes addressing the needs of colleges, universities, schools, and other educational organizations. Here are some key areas within the scope of college system software:

1) Administrative Functions:

   Admissions and Enrollment: Managing the admission process, application tracking, and enrollment of students.

   Fee and Financial Management: Handling fee collection, financial transactions, and budget management.

   Resource and Facility Management: Managing classrooms, labs, libraries, and other facilities.

   Human Resource Management: Managing staff and faculty records, payroll, and attendance.

2) Academic Management:

   Course and Curriculum Management: Creating, scheduling, and updatingcourses and curricula.

   Student Information Management: Maintaining student records, including personal details, academic progress, and attendance.

   Examination and Grading: Scheduling exams, recording grades, and generatingtranscripts and report cards.

   Timetabling: Creating and managing class schedules, exam schedules, and faculty timetables.

3) Library and Resource Management:

Library Automation: Managing library resources, including book cataloging, checkouts, and returns.

4) Attendance and Performance Tracking:

   Monitoring and recording student attendance. Tracking academic performance andgenerating progress reports.

5) Reporting and Analytics:

   Generating reports and analytics on various aspects of the institution, such asacademic performance, administrative processes, and financial data.

By encompassing these additional aspects, Smart College System Software offers a comprehensive solution for managing all aspects of college operations, promoting institutional effectiveness, student success, and academic excellence.

The scope of Smart College System Software continues to expand as technology evolves and institutions seek more efficient ways to manage their operations. With the increasing importance of data analytics and digital transformation in education, college management software is expected to play an even more significant role in the future, enhancing administrative and academic processes and improving the overall educational experience for students, faculty, and staff.

CHAPTER 2

1. REVIEW OF LITERATURE

Zhi-gang YUE, You-Wei JIN, et al.[1], proposed the method of management informatization in higher education. Based on a comprehensive investigation andanalysis of student management in higher education, they establish the models of college students' management informatization by adopting advanced information technology and constructing the student management informatization platform. Moreover,they analyze the characteristics of informatization management in higher education and elaborate on the methods to solve the difficulties confronting the student’s management of higher education. Finally, the key methods and technology to carry out the informatization management platform are presented.

Katsaros, Gregory, et al.[2], explained a holistic view of information management in cloud environments. Cloud computing gradually impacted the application computing environment, introducing new roles and responsibilities, as well as new systematic and business models. The amount of data available and the need to handle it properly grew in tandem with the increase in service providers' capabilities and potential. To that end, they present in this article an administration service architecture as well as the data model upon which the result is based. The concept was created to allow a storage service to be offered by a Cloud computing platform, but it is adaptive enough to work in a variety of Cloud scenarios. Data utilization of resources and cloud services is difficult due to a lack of standards and the infrastructure's flexibility. In this paper, they established a unified management strategy for a storage Cloud, but it can be used by any sort of Cloud provider. By analyzing the information flows of the Cloud storage architecture, they were able to capture the management requirements and construct the essential models. The ability to connect information from one model to another and the consistency of the models are critical to the Cloud system's effective management. In this scenario, rather than various information structures, the complete state of the Cloud service platform can be successfully recorded as a single monolithic model. They can use this feature to save "screenshots" of the system in case they need to relocate or re-start it in the future. The next step in this project is to investigate ontology specifications and the possibility of transforming this unified model into Cloud ontologies, which might serve as the administrative core of a Cloud infrastructure.

Yang, et al.[3], explained an efficient and secure dynamic auditing protocol for data storage in Cloud Computing. Data owners store their data on cloud storage, which users (data users) can gain entry through cloud computing technology. However, data exporting creates new security issues, necessitating the usage of a 3rd party monitoring service to ensure the integrity of data in the cloud. Because data in the cloud can be dynamically updated, several established faraway integrity-checking methodologies can only be applied to passive archive material and thus cannot be used to audit the service. An effective and secure dynamic monitoring mechanism is required to convince data owners that their data is properly stored in the cloud. In this research, a structured and privacy-preserving monitoring methodology for cloud storage systems was offered after designing an auditing framework. The arbitrary oracle model then extends our monitoring protocol to include information interactive operational activities, which is both safe and reliable. They enhanced their monitoring system to support batch auditing for different owners and different clouds without the need for a trustworthy organizer they suggested monitoring protocols are safe and reliable, according to the analysis and simulation results, and they lower the auditor's computation costs. In this article, we present a dynamic monitoring system that is both safe and reliable by design. Rather than employing the mask technique, To protect data privacy from auditors, it employs a mix of encryption and the bi-linearity property of bi-linear pairing. As a result, no additional organizer is required for our multi-cloud batch auditing methodology. Batch auditing for many owners is also possible with their accounting protocol. Furthermore, by transferring the auditing computational loads from the auditor to the server, the monitoring scheme causes reduced calculation and communication costs for the monitor, which considerably improves accounting performance and may be used inbig cloud storage systems.

S.R. Bharam Ago Udar et al.[4], proposed a web-based student information management. Student Information Management System (SIMS) provides a simple interface for the maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. The creation and management of date information regarding a student’s academic career is critically important in the university as well as colleges. Student information system deals with all kinds of student details, academic-related reports, college details, course details, curriculum, batch details, placement details, and other resource-related details too.

Andreas Papadakis, et al.[5], proposed a system for profiling students' progress in engineering disciplines is presented and built. This method allows students' status and advances in the field of communication systems to be evaluated and measured systematically. The framework is based on current engineering simulation tools that are commonly used in higher education. For the system's implementation and early piloting, widely used tools were employed as both the educational setting and the persistence service "context broker". Additionally, generic parsers were used for event processing. Performance, motivation, and competence were shown to have strong correlations. Focusing on the skills element, it was discovered that common programming language syntactical errors were not prevented. The integration of the outputs of numerous functions presented particular challenges. The algorithmic element is inaccessible to those who lack basic programming abilities. There was also a reluctance to seek assistance on behalf of the pupils. The qualitative findings were beneficial to both teachers and students in terms of successful “teaching and understanding of the course”, resulting in the improvement of guidelines and reading approaches. In terms of educational professionals, it was made feasible to ask useful questions regarding the time allocation of the entire process and individual questions, as well as the differential reaction latency to critical questions. Benchmarking against peers or course standards is possible thanks to the comparison functionality based on profiles.

Goyal, et al.[6], proposed an effective algorithmic approach for cost optimization in cloud-based data center cloud computing which provides consumers all around the world with cost-effective IT services. It can handle a wide range of user, scientific, commercial, and corporate applications. Cloud Computing is based on the premise that the complete system may be managed and operated using only an HTTP client. A web-based client is all that is required to work with Cloud Systems and all of its uses, including office apps, corporate components, and personal data systems. It is compatible with both past and present systems. An open-sourced framework that may be used to host a wide range of online apps. Everything inside a Cloud OS can be accessed and acquired from anywhere within it, according to its new definition. To gain access to the computer, the user simply needs to authenticate onto the Cloud Operating Database server using a conventional web browser. The desktop will contain all of the user's documents, programs, films, audio, and other media. You may upload and work with your data from any location using cloud operating systems. It covers practically all programs made by Cloud developers and Cloud providers, such as Word Processors, PDF Readers, Address Books, and many others. Cloud storage makes data capture and archiving more user-friendly and timely for internet users, which is the cornerstone of all types of cloud applications. However, a thorough examination of how to optimize cloud storage to increase data access and storage performance isrequired.

Saakshi Narula, et al.[7], proposed an overview of digital forensics on the topic of cloud computing security. After doing a security analysis, we illustrated how "AWS (Amazon Web Service)" cloud computing works. AWS is the most reputable cloud services provider, providing not only top-notch cloud privacy but also top-notch cloud services. Cloud computing is founded on the concept of "on-demand services", which means they can access cloud services whenever they need them and adjust them up or down as appropriate. Cloud security is the major concern in this paradigm. "Security around data, access, and privacy protection" is a major concern in cloud computing. Cloud computing should be safe and secure, and dangers should be minimized. According to a cloud computing analysis, security should be the primary function rather than a secondary function. In termsof cloud computing, AWS has a stellar track record.

Rafiqul Zaman Khan, et al.[8], proposed the study which is based on the necessity for cloud computing, cloud computing architecture, cloud computing kinds, cloud computing services, and cloud computing offerings. Purchasing merely computers for an organization's employees is insufficient because new applications may necessitate new software, which can be highly expensive, or the software licensing may not allow more than one user to install. Furthermore, the most crucial point is that the cost of maintaining both hardware and software is always far higher than the cost of installing the same. As a result, the IT personnel responsible for managing an organization's IT infrastructure are a significant financial and psychological strain. Cloud computing refers to the on-demand delivery of computer resources such as computation, storage, and software. Cloud computing services for storage, database, computing, and applications are provided by top service providers such as SalesForce, Google, SUN, Amazon, IBM, Oracle, and others. The clients do not require any costs for computing infrastructure construction or maintenance of any hardware or software, which relieves them of significant financial weight and mental stress. Clients need only link their computers or networks to cloud computing servers and pay for services using simple payment methods such as pay-per-use or subscription.

Siddhant Gok ule et al.[9], proposed an application to enhance the admission process and communication between members of an institute. With the development of the internet, internet information broadcasting and information-sharing technologies have made it possible for us to access information from all around the world with ease. In today's world,the use of the internet has eliminated the need for physical presence of people in all aspects. The main objective is to take advantage of fast growing popularity of Android devices by developing an Android application that provides people with a tool to fill in the application form for an Institution and also provides intra-college communication. The project provides the Design and Development of an Software that would allow us to submit admission forms to an institution and use services to allow communication within the institution.

Mehdi, et al.[10], proposed the lesson will begin with an overview of cloud computing environments, cloud services design, the need for mobile or native cloud computing in the app industry to deal with new mobile development, internet tools, application creating tools, and the rationale for moving apps to cloud computing services. The course will cover the basics, objectives, and typical topologies of mobile cloud computing systems, as well as provide an introduction to generic mobile cloud services for app developers and marketers. This lesson will go over some of the most common issues and expenses, as well as the role of mobile cloud computing infrastructure in the field of app design and how the app industry may make a low-cost transition to cloud computing systems. The lesson will go over privacy and security concerns. It will outline the services provided by key mobile cloud suppliers to demonstrate how mobile cloud traders can help Android application firms. We'll look at key cloud providers like "Microsoft Windows Azure, Amazon Web Services, and Google Cloud Platform". Finally, the lesson will go over some of the best techniques in the industry, as well as some future development possibilities.

Lalit Mohan Joshi et al.[11], proposed the development of an Online Intranet College System Software (CSS) that is of importance to either an educational institution or a college. The system is an Intranet-based application that can be accessed throughout the institution or a specified department. This system may be used for monitoring attendance for the college. Students as well as staff logging in may also access or search any of the college information. Attendance of the staff and students as well as marks of the students will be updated by staff. This system (C.M.S) is being developed for an engineering college to maintain and facilitate easy access to information. For this, the users must be registered with the system after which they can access as well as modify data as per the permissions given to them. CMS is an intranet-based application that aims at providing information to all levels of management within an organization. This systemcan be used as a knowledge/information management system for the college. A given student/staff (technical/Non-technical) can access the system to either upload or download some information from the database.

Nitin Naik et al.[12], proposed that data analysis has grown more important in the day-to-day operations of businesses. Every organization collects a massive amount of data daily. They then build their present plans based on the information they've gathered. The majority of small and medium firms, on the other hand, have faced two major hurdles in the area of data analytics: the need for a variety of expensive analytical tools and IT infrastructure, as well as their workers' IT talents. One of the most effective solutions for them would be the cloud's cost-effective and on-demand IT equipment and application assets. The Cloud System of Google is among the biggest and most advanced cloud systems, with a broad array of services, including those that are free, such as GoogleDrive. This article illustrates how to create a system of systems using Google Cloud System & SAML/OpenID Connect most simply and cost-effectively feasible. The full suite of Google Drive products, including Google Sheets, Google Maps, etc., can then be securely connected to the organization's system using the common SAML/OpenID Connect framework. Data analysis can then be performed using the full suite of Google Drive products, including Google Sheets, and Google Maps. Not only is this system of systems the least costly and consumer-friendly alternative, but it can also be utilized by anybody, at any time, from any location. The experimental simulation also demonstrates how simple it is to use these Google Drive capabilities to implement the proposed data analysis approach.

Anand Desai, et al.[13], proposed how data mining can be used in an educational environment. This study can be utilized to help schools classify students' student achievements as well as their perseverance as determined by a toughness assessment so that they can tailor their instruction to different groups of kids. For the required students, remedial lessons or extra examinations might be scheduled based on this classification. Students can also use the application to track their progress from semester to semester. The paper tries to classify pupils based on a variety of factors, including their academic records, such as marks in 10th and 12th grades, "an aptitude test, a grit test, and the CGPA score". The K-means algorithm gives the results of all the different tests, and the categorizationis based on these parameters. This article demonstrated how students can be categorized based on their background, talents, and achievements in both technical and non-technical areas, in addition to their academic performance. This study aims to identify students who require extra attention to reduce failure rates and take appropriate action at the correct moment.

Mar Bukh, et al.[14], proposed the goal to quantify and manage the underlying systematic performance/risk tradeoff in the "cloud computing paradigm". The cloud is regarded as a complex system, and systemic hazards are associated with the likelihood of a framework phase transformation to an undesirable permanent state. The findings point to a change in cloud infrastructure development and management paradigms away from maximizing financial prosperity and toward managing and optimizing the underlying condition benefit/risk tradeoffs, which can be done using mean-field and fluid approximations. Economic factors are thought to intensify this tradeoff by pushing "Cloud service providers" closer to the operating regime's limit, increasing the risk of overload when the system's capacity is insufficient to fulfill exogenous demand. According to this article, the financial effects of adaptive sharing of resources are inexorably tied to the threat of systemic overflow, which might occur gradually or abruptly. Instead of the standard economic efficiency maximization paradigm, optimization problems, which manage the systemic risk of overload, should be adopted.

Charlotte Kotas, et al.[15], proposed the compute-oriented instances from the Amazon Web Services and Microsoft Azure cloud platforms, and compared them against a variety of high-performance computing benchmarks. These comparisons show that the most cost-effective approach is determined by the application to be run. The cheapest cloud platform for a specific use case is determined by the application's computation and communication habits. The AWS c4.8xlarge was cheaper in terms of raw computation at the time the tests were done, but Azure's H16r offered cheaper bandwidth, according to this study. As a result, communication-intensive applications may benefit from the Azure H16r's quicker network and greater RAM, resulting in a more cost-effective solution overall. Given that cloud providers are always updating their products, testing any given application on the anticipated system is the best approach to determine how it will operate in the current cloud environment.

Fortunato, et al.[16]: explained the mobile market, mobile apps have had tremendous success. Many organizations were interested in having their own optimized mobile apps for all main mobile operating systems, therefore this prospect drew a lot of attention. However, when designed natively for each mobile platform, these advances are costly. New advancements in web technologies have enabled more features and capabilities that were previously only available in natively designed apps. This opened. up new opportunities for focusing all development efforts only on web apps, or apps that run in web browsers. The goal of this article is to learn about the evolutions, capabilities, and constraints of designing a web app that can operate on any device. This paper introduces Google's new Progressive Online App concept, which aims to standardize all web developments. When compared to designing the same solution for each separate mobile platform, the key advantages of developing the apps centralized as a Progressive Web App will be discussed. The current state of web technologies will also be discussed, as well as which cases Progressive Web Apps offer a strong alternative to mobile native apps.

Behl, et al.[17], proposed the concept of Progressive Web Apps (PWAs) which come out as a viable substitute for conventional mobile app development. PWAS incorporates the best characteristics of both mobile and web apps, as well as capabilities like offline application loading, pushed alerts, and backdrop synchronization, all of which help to boost user engagement. The paper presents a review of how to create modern applications with PWAs.The paper talks at length about the advantages of progressive web apps over native apps. The architectural design used to construct progressive web apps is also discussed in the paper to optimize the app's loading speed. The API supplied by service workers has also been thoroughly examined, including pushed alerts and backstory synchronization for a mobile app-like experience. A case study was conducted to demonstrate the operation of numerous progressive features. In the app development market, Progressive Web Applications have a bright future. According to certain research, over 80% of users only utilize a small handful of apps daily. As a result, memory is squandered since another program that the customer rarely uses consumes a large amount of storage. There's noneed to download or keep anything to in storage while creating a progressive web application.As a result, PWAS will save money by lowering the amount of memory required.

CAI Chang-an, WANG Qi et al.[18], proposed the method of developing a College Automation System which will manage the working of college management activity using a single platform. This system has an easy interface and strong data management. They have used Bootstrap which increases the responsiveness of the system. The objective of this system is to reduce the paperwork and manual processing. This System aims to reduce the workload and to save significant staff time. Their System provides automate admissions no manual processing is required. This paper describes automating the existing manual system. This is a paperless work. It can be monitored and controlled remotely. It reduces the manpower required. It provides accurate information this gathered information can be saved and can be accessed at any time. The data which is stored in their repository Helps in making decisions for the management providing accurate results. The storage the facility will relieve the job of the operator. Thus the system developed will be helpful to the administrator by easing tasks and providing accurate results.

Achal Agrawal, et al.[19], explained the advantages of adopting Flutter over alternative app development platforms are discussed in this paper. It covers how different platforms work and what role they play in application development. Flutter is an open SDK-source and is used for building high-performing, better-performing, and stable iOS and Android mobile apps. Flutter iOS has a higher top CPU performance than native iOS, while Native Android has a higher CPU performance than Flutter Android. The application produced in Flutter was much smaller in code and took a lot less time to construct than native Android and iOS apps. The preliminary findings in this paper indicate that Flutter has a modest advantage over native application development platforms, but more conclusive experiments must be conducted before a definite conclusion can be reached. Flutter and native appear to differ little in appearance to the majority of users. To a certain extent, it can imitate native appearances. Although Flutter may not be able to compete with native for developing applications at present, the results indicate that ithas a bright future.

Dhananjay Sinh et al.[20], proposed to look at the features, functionality, and enormous prospects of progressive web apps as a mobile development link connecting mobile apps and the web. Progressive Online Apps (POAs) are cross-platform web apps that leverage common web technologies to simulate the customer experience of a native app. The characteristics of a PWA combine the best of both online and native apps. Application shell, service worker, and manifest are all necessary components for developing progressive web apps. PWAs make use of important developments in modern web browsers, web Application Programming Interfaces, and front-end technologies to give mobile and desktop consumers outstanding app experiences. Developers may use PWA to create better experiences and, as a result, superior apps. PWAS has been discovered to offer a lot of potential as a web-mobile app development unifying force. Early adopters will have a significant edge in this sector due to smart application design and development. Based on feature comparison and research, this study tried to provide insight into the future benefits PWA could provide.

Label and the need to handle it properly grew in tandem with the increase of service providers' capabilities and potentials. To that end, they present in this article an administration service architecture as well as the data model upon which the result is based. The concept was created to allow a storage service to be offered by a Cloud computing platform, but it is adaptive enough to work in a variety of Cloud scenarios. Data utilization of resources and cloud services is difficult due to a lack of standards and the infrastructure's flexibility. In this paper, we established a unified management strategy for a storage Cloud, but it can be used by any sort of Cloud provider. By analyzing the information flows of the Cloud storage architecture, they were able to capture the management requirements and construct the essential models. The ability to connect information from one model to another and the consistency of the models are critical to the Cloud system's effective management. In this scenario, rather than various information structures, the complete state of the Cloud service platform can be successfully recorded as a single monolithic model. They can use this feature to save "screenshots" of the system in case they need to relocate or re-start it in the future. The next step in this project is to investigate ontology specifications and the possibility of transforming this unified model into Cloud ontologies, which might serve as the administrative core of a Cloud infrastructure.

CHAPTER 3

1. PROPOSED METHODOLOGY

3.1 Materials & Methods to be Used (Technical Details)

Our proposed “Smart College System Software using Data Science” is composed of various layers. The data analytics in Smart College System Software is performed at layers, based on different architectures. These layers include source systems and data collection, data loading and processing, and results utilization as depicted. The description of these layers along with the proposed architecture is provided in the following subsections.
By incorporating these technical details into the development and implementation process, Smart College System Software can deliver a robust, secure, and user-friendly solution that meets the diverse needs of educational institutions and stakeholders.

3.1.1 Project Category (Data Science)

The project “Smart College System Software Using Data Science” is being developed using Python Programming and Data Science techniques. Python's Tk inter-library for building graphical user interfaces (GUIs). Python's versatility makes it a popular choice for data science projects. Its rich ecosystem of libraries and frameworks, combined with its simplicity and readability, enables data scientists to perform a wide range of tasks efficiently. Tk inter is Python's standard GUI (Graphical User Interface) toolkit, providing a simple and intuitive way to create desktop applications with graphical interfaces. It is included with most Python installations, making it readily available for developers to use without any additional installations or dependencies.

3.1.2 Techniques to be Used

Developing a Smart College System Software involves using a variety of materials, tools, and methods. Here are the technical details related to materials and methods used in creating and implementing Smart College System Software :

1. **Tools**
   - **Spyder (Anaconda 3)**

     Spyder is a free and open-source scientific environment for Python.

Combining advanced analysis, debugging, editing, and profiling with data exploration. Spyder features a multi-language editor pane to create, open, and modify source files. Spyder, bundled with Anaconda, is a dynamic duo in the realm of Python-based data science. Spyder serves as the integrated development environment (IDE), offering a seamless platform for coding, debugging, and exploration. With its user-friendly interface and powerful features tailored for scientific computing, Spyder streamlines the data analysis workflow. Anaconda 3 complements Spyder by providing a robust Python distribution tailored for scientific computing and data analysis. Bundled with a plethora of pre-installed libraries and tools, Anaconda 3 ensures that users have everything they need right out of the box, simplifying package management and deployment. Together, Spyder and Anaconda 3 form a synergistic partnership, empowering data scientists, researchers, and developers to tackle complex data analysis tasks with ease and efficiency. Whether it's exploring datasets, prototyping machine learning models, or visualizing results, Spyder (Anaconda 3) stands as a cornerstone in the toolkit of modern data-driven professionals.

- **SMTP**

  SMTP (Simple Mail Transfer Protocol) is a TCP/IP protocol used in sending and receiving email. SMTP is used most commonly by email clients, including Gmail, Outlook, Apple Mail, and Yahoo Mail. SMTP cansend and receive email, but email clients typically use a program with SMTP for sending email. SMTP is a text-based protocol, making it easy for different mail servers and email clients to communicate with each other regardless of the underlying operating system or hardware. While SMTP handles the sending aspect of email, other protocols like POP3 and IMAP are used for retrieving emails from mail servers. Despite its simplicity, SMTP plays a crucial role in the global network of communication, enabling billions of emails to be sent and received every day, facilitating everything from personal correspondence to business transactions and beyond.

- **Data Science**

  Data Science is the study of data to extract meaningful insights for business. It is a multidisciplinary approach that combines principles and practices from the fields of mathematics, statistics, artificial intelligence, and computer engineering to analyze large amounts of data. Data science is a multidisciplinary field that utilizes scientific methods, algorithms, and systems to extract insights and knowledge from structured and unstructured data. It combines elements of statistics, mathematics, computer science, and domain expertise to uncover patterns, trends, and correlations that can inform decision-making and drive innovation across various industries. At its core, data science involves the collection, cleaning, processing, and analysis of large volumes of data to extract meaningful insights and solve complex problems. This process often encompasses data wrangling, exploratory data analysis, statistical modeling, machine learning, and data visualization techniques.

  Data scientists leverage a diverse set of tools and programming languages such as Python, R, SQL, and libraries like TensorFlow and sci-kit-learn to manipulate data and build predictive models. Additionally, they often work with big data technologies such as Hadoop, Spark, and cloud computing platforms to handle large-scale datasets efficiently.

  The applications of data science are wide-ranging and span across industries such as healthcare, finance, marketing, retail, and beyond. From predicting customer behavior and optimizing business processes to advancing medical research and improving public services, data science has the potential to revolutionize how we understand and interact with the world around us. As data continues to grow in volume and complexity, the role of data scientists becomes increasingly vital in harnessing its potential to drive meaningful impact and innovation.

2. **Programming Language**

- **Python**

  Python is a widely used high-level, interpreted, and object-oriented programming language that supports multiple programming paradigms, including procedural, object-oriented, and functional programming. It was created by Guido van Rossum between 1985 and 1990 and is dynamically-typed and garbage-collected. Python's source code is available under the GNU General Public License (GPL) and its design philosophy is centered around code readability through the use of significant indentation. Python is an excellent choice for developing smart college system software due to its versatility, simplicity, and a robust ecosystem of libraries and frameworks. Here are some reasons why Python is well-suited for this task its clean and readable syntax makes it easy for developers of varying skill levels to understand and write code. This is advantageous for college system software development, as it allows for faster prototyping and iteration. Python boasts a vast ecosystem of libraries and frameworks that cover a wide range of functionalities relevant to college system software development. Libraries like Django and Flask are popular choices for building web applications, while libraries such as Pandas and NumPy are well-suited for data manipulation and analysis, which can be useful for managing student records and academic data. It seamlesslyintegrates with other technologies and platforms, allowing for easy integration with existing systems and databases commonly used in colleges and universities. This facilitates the development of cohesive and interconnected college system software that can interface with various data sources and services. Python has a large and active community of developers who contribute to its development and provide support through online forums, tutorials, and documentation. This vibrant community ensures that developers have access to resources and assistance when building and maintaining college system software.

  Scalability and Performance: While Python is often praised for its ease of use, it also offers scalability and performance when needed. By leveraging techniques such as asynchronous programming and optimization strategies, Python can handle large-scale college system software with efficiency and reliability. Overall, Python provides the necessary tools and capabilities to develop a smart college system software that addresses the unique needs and challenges of modern educational institutions. Its flexibility, robustness, and community support make it an ideal choice for building innovative solutions that enhance the efficiency and effectiveness of college operations. Python is more than just a programming language; it's a versatile tool that empowers developers, scientists, educators, and enthusiasts alike. With its clean syntax and readability, Python fosters rapid development and encourages collaboration. From web development to data analysis, and artificial intelligence to automation, Python's rich ecosystem of libraries and frameworks equips users with the tools they need to tackle diverse challenges. Whether you're a beginner taking your first steps in programming or a seasoned developer pushing the boundaries of innovation, Python welcomes you with open arms, inviting you to explore its endless possibilities and unleash your creativity. One of Python's standout features is its emphasis on readability and maintainability. Its code is often described as "executable pseudo-code," making it easy to understand and maintain even in complex projects. This readability promotes collaboration among developers and facilitates the sharing of knowledge within the community.

Python's versatility shines through in its ability to be used in various domains, including web development, scientific computing, data analysis, machine learning, artificial intelligence, and more. Its extensive standard library provides a wide range of modules for tasks ranging from working with files and networking to handling data structures and implementing algorithms.

3. Libraries Used

- **NumPy**

  NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

- **Pandas**

  Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

- **Matplotlib**

  Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication-quality plots. Make interactive figures that can zoom, pan, and update.

- **SciPy**

  SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimizations, stats, and signal processing.

- **Seaborn**

  Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

4. **Databases Used**

- **MySQL**

  MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold vast amounts of information in a corporate network.

- **CSV file**

  A CSV is a comma-separated values file, which allows data to be saved in a tabular format. CSVs look like a garden-variety spreadsheet but with a .csv extension. CSV files can be used with almost any spreadsheet program,such as Microsoft Excel or Google Spreadsheets.

5. **Applications Used**

- **GUI**

  A graphical user interface (GUI) is a digital interface in which a user interacts with graphical components such as icons, buttons, and menus. In a GUI, the visuals displayed in the user interface convey information relevant to the user, as well as actions that they can take. GUIs enhance user experience by offering a familiar and intuitive interface, reducing the learning curve for new users and increasing productivity for experienced ones. They enable users to interact with software through direct manipulation, allowing actions to be performed with simple gestures like clicking, dragging, and typing. GUIs are not only user-friendly but also visually appealing, with customizable layouts, colors, and graphics that can be tailored to suit different preferences and branding requirements. This aesthetic appeal contributes to a positive user perception and fosters engagement with the software.

3.1.3 Parallel Techniques Available

Parallel Computing techniques can be used in Smart College System Software to enhance its performance, scalability, and responsiveness. Here are several parallel techniques that can be employed :

1. **Applications Used**
   - **Flask**

     Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Flask's minimalist approach doesn't sacrifice power. Its simplicity allows developers to focus on crafting clean, efficient code without unnecessary complexity. By embracing the "do-it-yourself" philosophy, Flask provides a solid foundation while still allowing developers to customize and extend functionality as needed. With its built-in development server and debugger, Flask streamlines the development process, making it easy to test and iterate on applications. Beyond its core features, Flask boasts a vibrant ecosystem of extensions and plugins, offering additional functionality for tasks such as authentication, database integration, and RESTful APIs. This rich ecosystem, coupled with Flask's active community and extensive documentation, ensures that developers have the resources they need to tackle any project.
   - **Django**

     Django is a Python-based web application framework that is free and open source. A framework is simply a collection of modules that facilitates development. They’re together and allow you to build apps or websites from scratch rather than starting from scratch. Django, the high-level Python web framework, is a powerhouse in the world of web development, renowned for its "batteries-included" philosophy that prioritizes simplicity and productivity. With Django, developers can quickly build secure, scalable web applications while adhering to the best practices and conventions. At its core, Django emphasizes the principle of "don't repeat yourself" (DRY) by providing a clean and pragmatic design that promotes code reuse and maintainability. Its robust set of built-in features, including an ORM (Object-Relational Mapping) for database interactions, automatic admin interface generation, and built-in security features like protection against common vulnerabilities such as SQL injection and cross-site scripting (XSS), accelerates development without sacrificing quality. Django's adherence to the MVC (Model-View-Controller) architectural pattern, or more accurately, the MTV (Model-Template-View) pattern, promotes clean separation of concerns, making it easier to manage complex applications. Its powerful templating engine allows for dynamic content generation, while its comprehensive documentation and vibrant community support ensure that developers have the resources they need to succeed. Furthermore, Django's extensibility through a rich ecosystem of reusable apps and packages allows developers to add functionality to their projects with ease. Whether it's integrating third-party libraries, implementing RESTful APIs, or deploying applications to production servers, Django provides robust solutions every step of the way.

In summary, Django empowers developers to build high-quality web applications efficiently, thanks to its pragmatic design, comprehensive feature set, and thriving community. Whether you're a beginner or an experienced developer, Django remains a top choice for projects of all sizes, from small prototypes to enterprise-level applications.

2. Databases Used

- **File System**

  A file system stores and organizes data and can be thought of as a type of index for all the data contained in a storage device. These devices can include hard drives, optical drives, and flash drives. The file system, often referred to as the backbone of a computer's storage architecture, organizes and manages data in a hierarchical structure, enabling users and software to store, retrieve, and manipulate files efficiently. At its core, the file system provides a method for organizing data into directories (or folders) and files, with each file typically containing data or metadata related to user-created content, system configurations, or application settings. One of the fundamental concepts of a file system is the directory hierarchy, which allows for logical organization and categorization of files. Users can create directories within directories, forming a tree-like structure that facilitates easy navigation and management of files. Additionally, file systems often support various attributes and permissions, allowing users to control access to files and directories based on security requirements. File systems also manage storage allocation and retrieval, determining how data is stored on physical storage devices such as hard drives or solid-state drives (SSDs). This includes managing disk space allocation, tracking file locations, and optimizing read and write operations for efficiency. As technology evolves, so too do file systems, with advancements such as journaling, encryption, and support for larger storage capacities continually being developed to meet the growing demands of modern computing. Overall, the file system serves as a critical component of computer systems, providing the foundation for data storage, organization, and management in both personal and enterprise environments.

3.2 Hardware and Software Resource Requirements and their Specifications

1. **Hardware Requirements**
    - 8 GB RAM (min)
    - SSD (Solid State Drives). Ensuring fast data access
    - Intel i5 or Ryzen 5 processor or similar
    - Network Connectivity
2. **Software Requirements**
    - Spyder (anaconda 3)
    - MySQL
    - SMTP (E-Mail)

3.3 Proposed Algorithm

1. **Define Requirements :**
    - Identify the key features and functionalities required for the smart college system.
    - Specify the data sources, including student information, courses, grades, attendance, etc.
2. **Data Collection :**
    - Gather relevant data from various sources such as databases, spreadsheets, or external APIs.
    - Preprocess and clean the data to handle missing values, outliers, and inconsistencies.
3. **Data Exploration and Visualization :**
    - Use libraries like Pandas, Matplotlib, and Seaborn to explore and visualize the data.
    - Generate descriptive statistics and visualizations to gain insights into the college data.
4. **Feature Engineering :**
    - Identify and create relevant features that can be used for analysis and predictions.
    - Consider features such as student attendance, previous grades, extracurricular activities, etc.
5. **Predictive Analytics (Machine Learning) :**
    - Choose a machine learning algorithm based on the tasks you want to accomplish (e.g., predicting student performance, recommending courses).
    - Split the data into training and testing sets.
    - Train the machine learning models using the training data.
    - Evaluate the model's performance using the testing data.
    - Implement models for tasks like predicting grades, identifying at-risk students, or suggesting courses.
6. **User Interface (GUI) Development :**

- Use a GUI library like Tk inter, and PY-Qt, to create an interactive user interface.
- Design screens for features such as student information, course registration, grades, and analytics.
- Integrate input forms and buttons for user interactions.

7. **Database Integration :**
   - Connect the software to a database (SQLite, MySQL, etc.) to store and retrieve data.
   - Implement functionalities to update student records, grades, and other relevant information.
8. **Real-Time Analytics :**
   - Implement real-time analytics features to monitor student performance and provide timely insights.
   - Use streaming data or scheduled tasks to update analytics dashboards.
9. **Security and Authentication :**
   - Implement user authentication to ensure secure access to the system.
   - Define user roles and permissions to control data access.
10. **Testing ;**
    - Conduct thorough testing to identify and fix any bugs or issues.
    - Perform user acceptance testing to ensure the software meets the specified requirements.
11. **Deployment :**
    - Deploy the smart college system software to a server or cloud platform for accessibility.
    - Ensure proper documentation for future maintenance and updates.
12. **Maintenance and Updates :**
    - Regularly update the system based on use feedback and changing requirements.
    - Monitor and maintain the system to ensure smooth operations.

3.4 System Architecture, Flow Chart, State transition Diagram, Data Flow Diagram

a) System Architecture

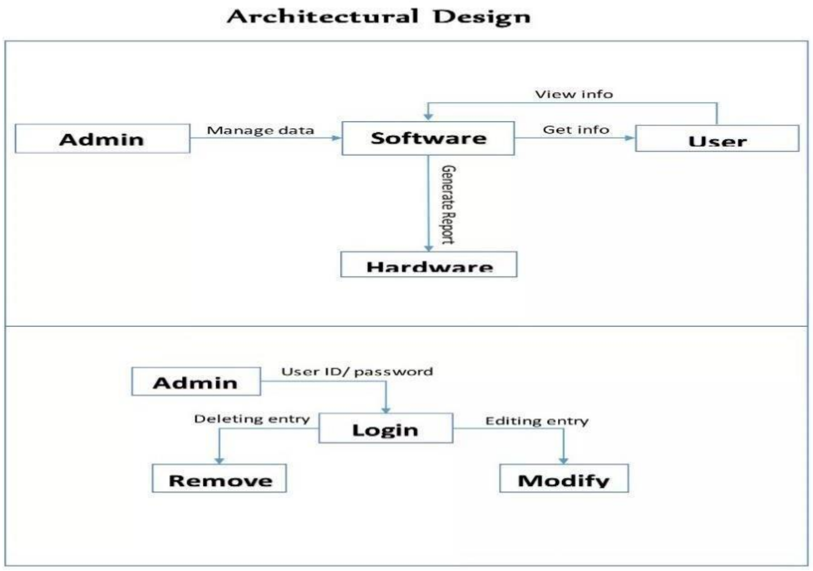


Fig. 3.4.1

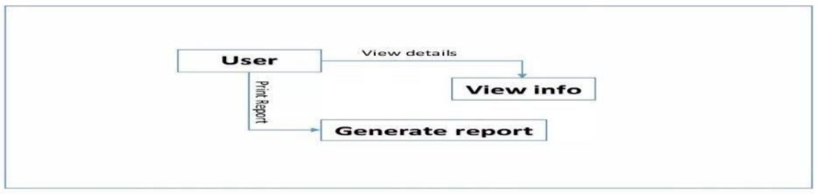


Fig. 3.4.2

b) **Flow Chart**

- **Admin**

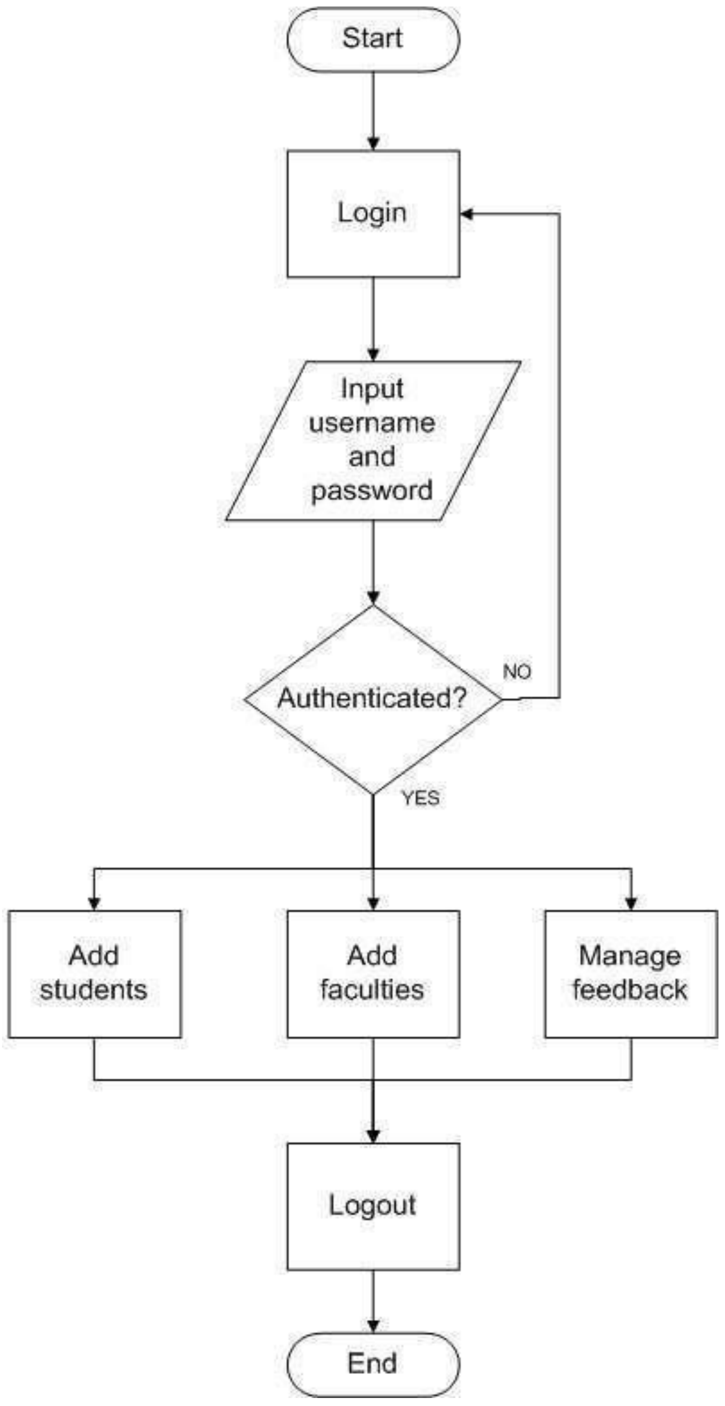


Fig. 3.4.3

- **Student**

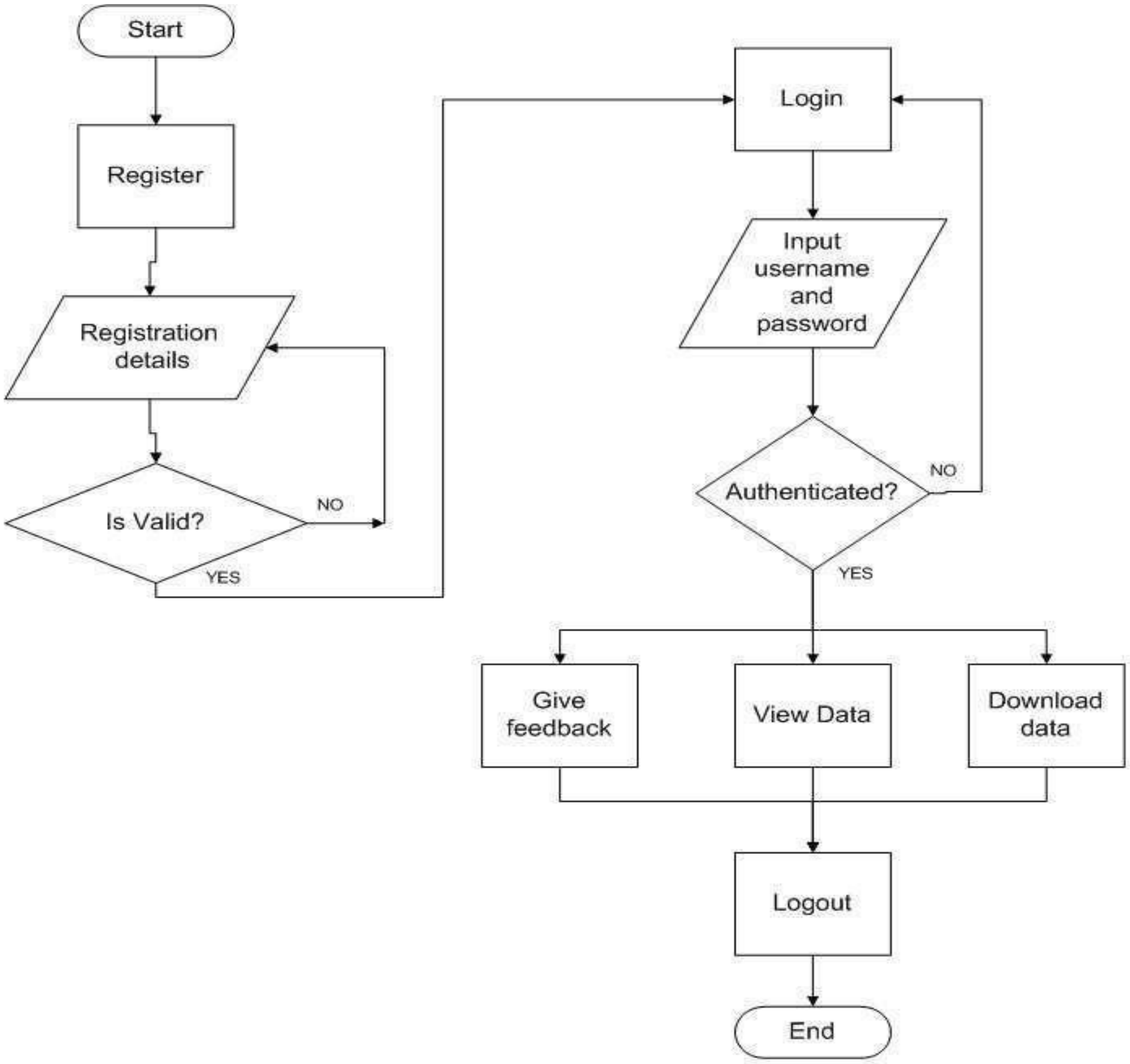
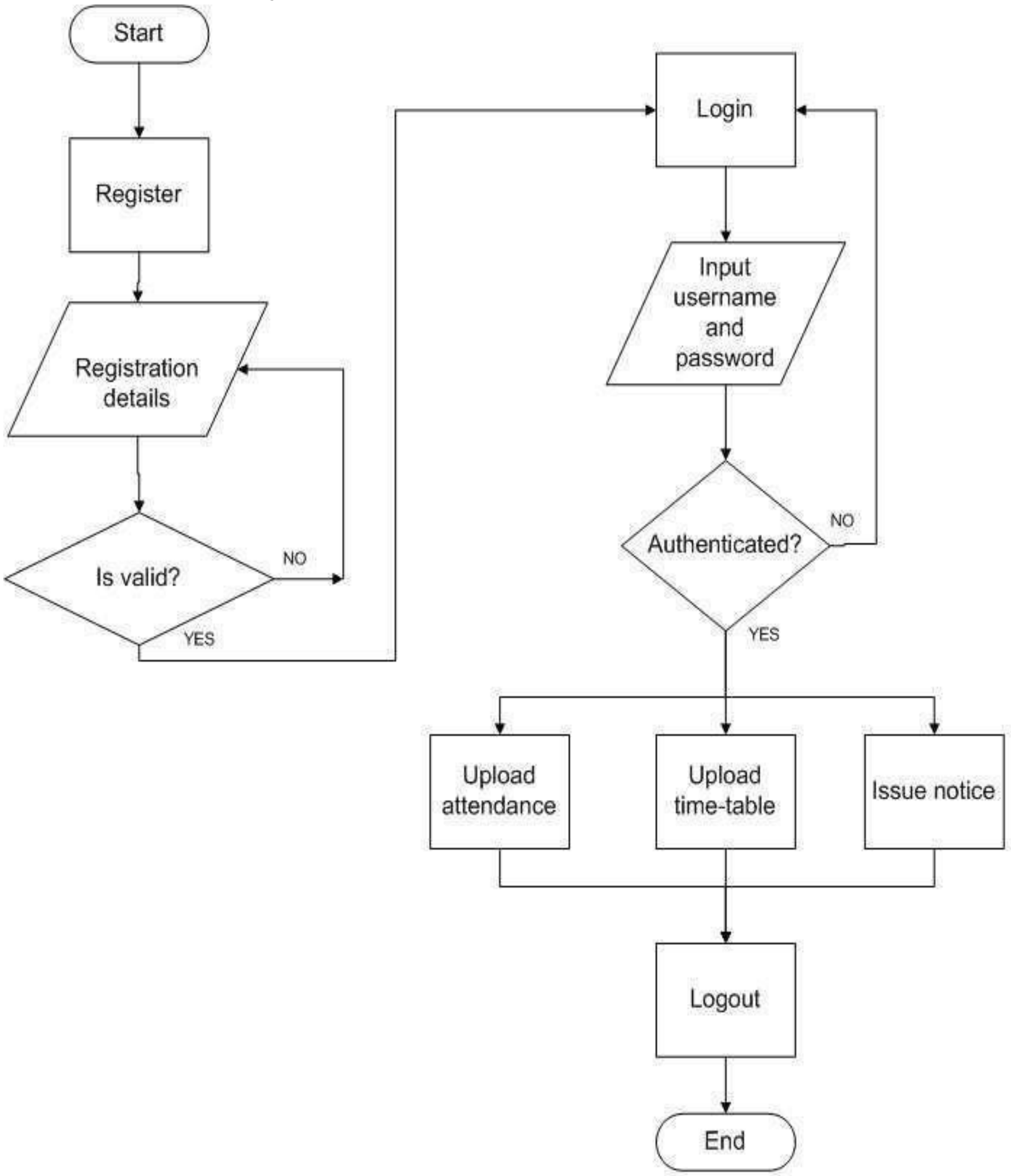


Fig. 3.4.4

- **Faculty**



` **Fig. 3.4.5**

c) **State transition Diagram**

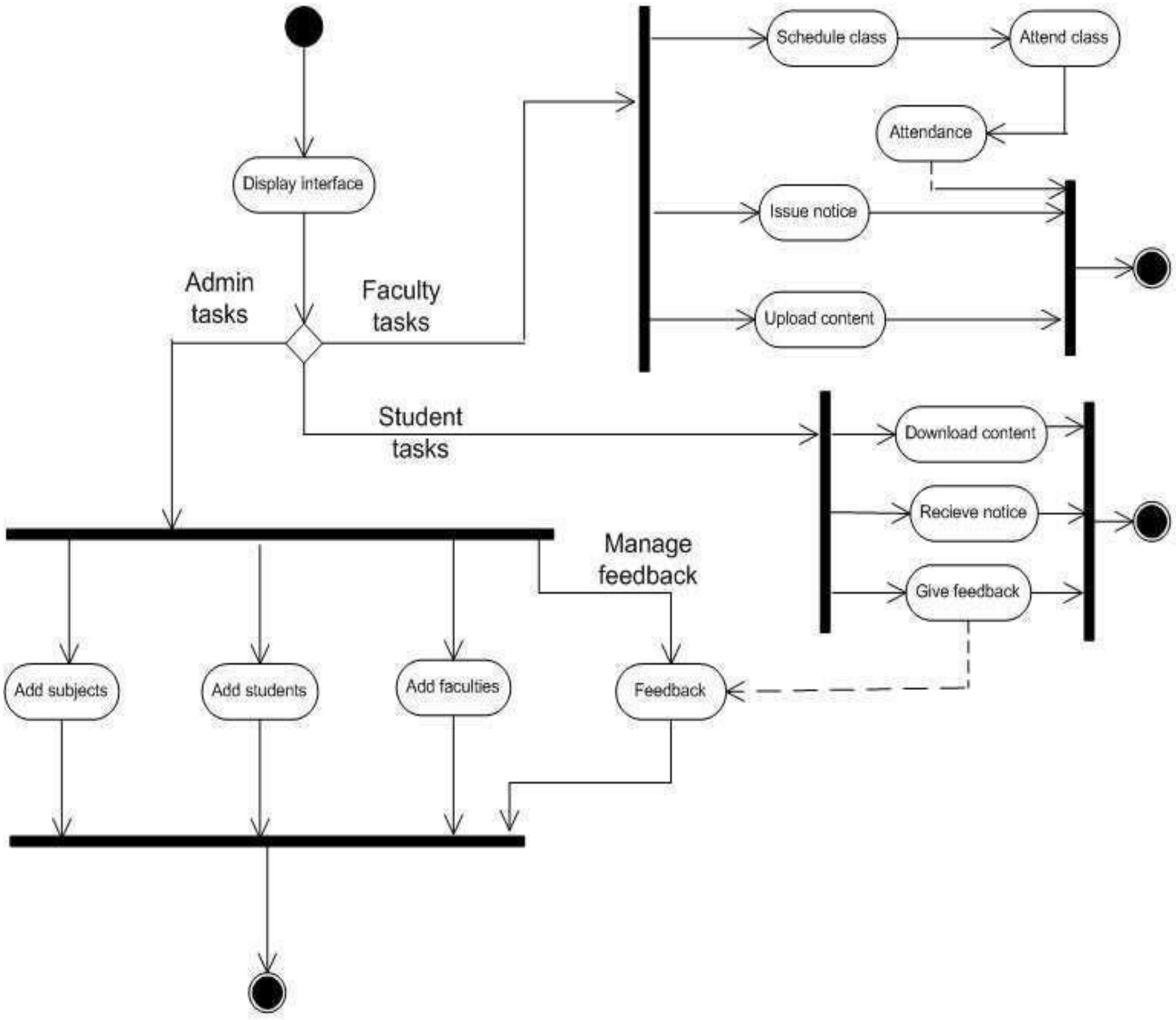
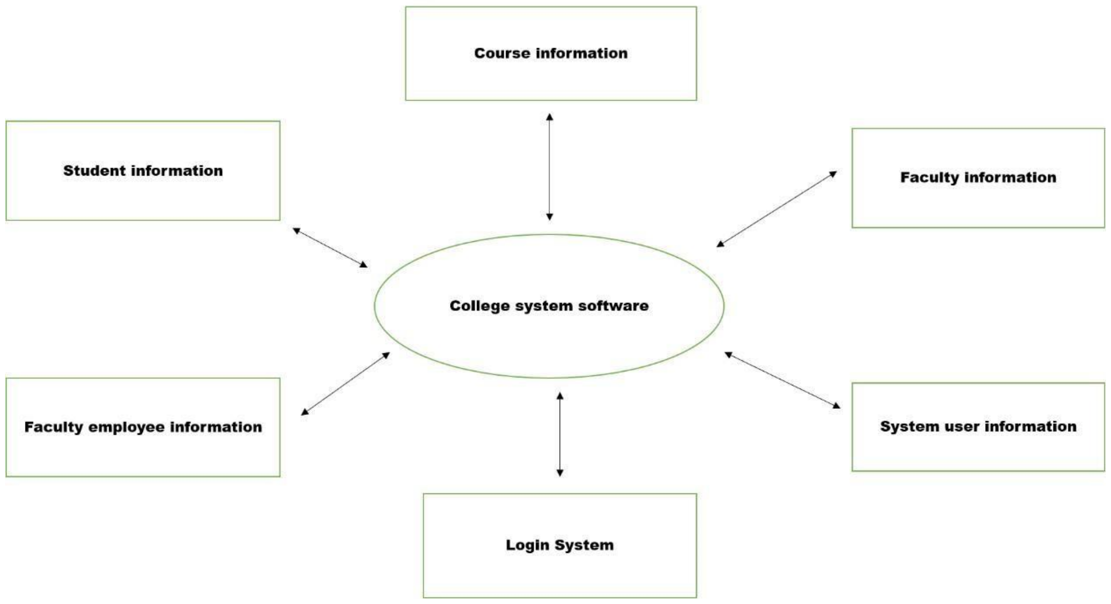


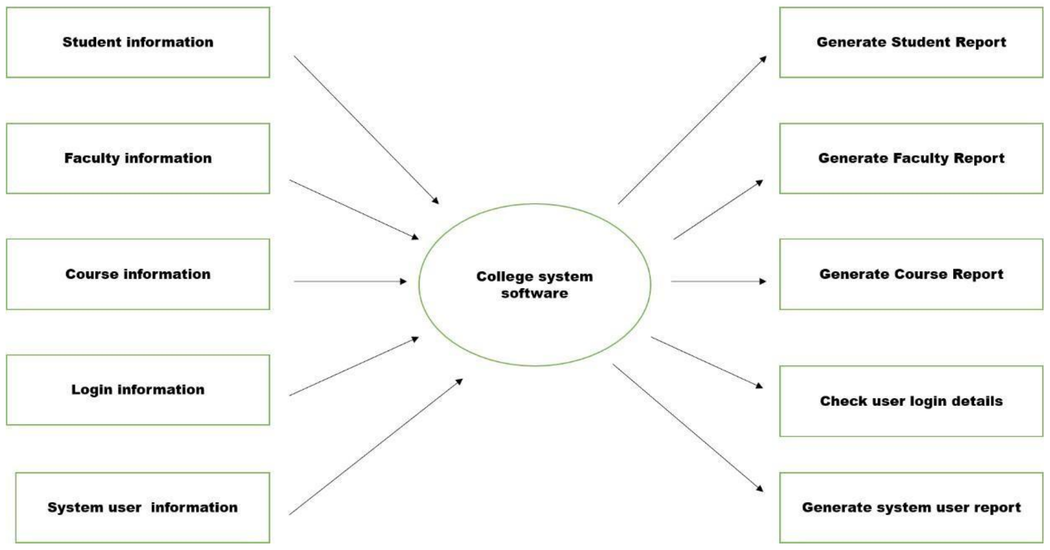
Fig. 3.4.6

d) Data Flow Diagram

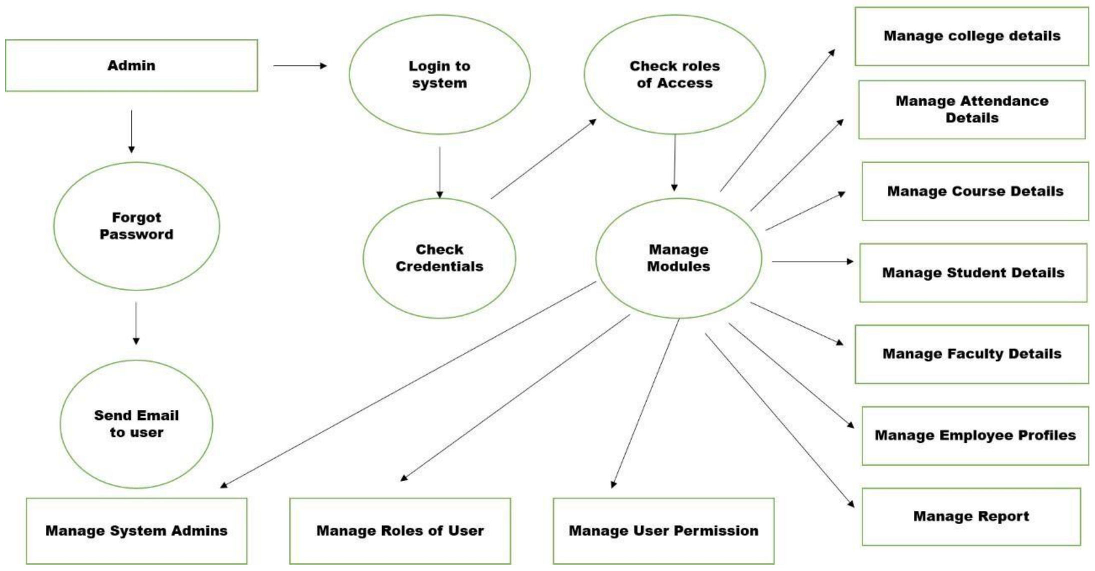
1) Zero-level DFD – Smart College System Software



2) First level DFD – Smart College System Software



3) Second level DFD – Smart College System Software



CHAPTER 4

4. TESTING TECHNOLOGIES AND SECURITY MECHANISMS

4.1 Testing Technologies

- ❖ **Unit testing :**
  - Use Python’s built-in ‘**unit test**’ or a third party like ‘**Pytest**’ to write unittests for individual functions and modules.
  - Ensure that functions behave as expected, especially critical functions handling sensitive data.
- ❖ **Integration testing :**
  - Test the interactions between different components of your system.
  - Verify that modules work together seamlessly.
- ❖ **Functional testing :**
  - Test the overall functionality of the smart college system.
  - Check if user interfaces, database interactions, and other features work as intended.
- ❖ **Security testing :**
  - Perform security testing to identify vulnerabilities.
  - Use tools like ‘**locust**’ or ‘**JMeter**’ to simulate concurrent users and measure response times.
- ❖ **User Acceptance Testing (UAT) :**
  - Involve end-users to validate that the system meets their requirements.
  - Identify and address any usability issues.

4.2 Security Mechanisms

- ❖ **Data Encryption :**
  - Encrypt sensitive data before storing it in the database.
  - Use libraries like ‘**cryptography**’ in Python for encryption.
- ❖ **Secure Password Handling :**

- Hash passwords before storing them using a strong hashing algorithm (e.g., b crypt).
- Implement salting to enhance password security.

❖ **Access Control :**

- Implement role-based access control (RBAC) to restrict access based on user roles.
- Regularly review and update access permissions.

❖ **Secure Database Queries :**

- Use parameterized queries or an Object-Relational Mapping (ORM) library (e.g., SQL Alchemy).

❖ **Session Management :**

- Implement secure session management to prevent session hijacking.
- Use secure, random session tokens.

❖ **Input Validation :**

- Validate and sanitize user inputs to prevent malicious input.
- Use server-side validation along with client-side validation.

❖ **Logging and Monitoring :**

- Implement comprehensive logging for security-related events.
- Set up monitoring systems to detect and respond to unusual activities.

❖ **Regular Updates and Patching :**

- Keep all software components (Python, MySQL, web server, etc.) up to date with the latest security patches.

❖ **Two-Factor Authentication (2FA) :**

- Implement 2FA for an extra layer of authentication.

❖ **Network Security :**

- Secure communication channels using HTTP.
- Implement firewalls and other network security measures.

CHAPTER 5

5. LIMITATIONS & DELIMITATIONS

5.1 *Limitations*

- **Scalability Issues :**
  - As the system grows in terms of users and data, scalability might become a concern. MySQL may face limitations in handling a massive amount of data and concurrent users. Considerations for database sharding or migration to a more scalable database solution might be necessary.
- **Security Concerns :**
  - While MySQL is a robust database system, the security of your application depends on how well you implement security measures. Ensure proper encryptions, secure coding practices, and user authentications to mitigate security risks.
- **Complexity of Features :**
  - Implementing advanced features such as machine learning algorithms, real-time analytics, or complex data processing may be challenging within the limitations of MySQL and may require integration with other technologies.
- **User Interface (UI) Limitations :**
  - Python’s native GUI libraries (e.g., Tk inter) might have limitations in creating modern and highly interactive user interfaces. Consider using additional front-end frameworks like React or Angular for a more sophisticated UI.
- **Dependency on Internet Connectivity :**
  - If the system heavily relies on internet connectivity, it may become less reliable in offline scenarios. Ensure that the application has appropriate error handling and offline capabilities where necessary.
- **Compatibility with Other Systems :**

- Integration with other systems or third-party services may face challenges due to compatibility issues. Ensure that the systems you integrate with haveappropriate APIs or connectors available.

5.2 *Delimitations*

- ❖ **Platform Dependency :**
  - If the system is developed specifically for Python, it may be limited to certain operating systems. Ensure that the application is tested and compatible with the intended platforms.
- ❖ **Limited Mobile Compatibility :**
  - If the application is designed primarily for desktop use, mobile compatibility might be limited. Developing a responsive design or a separate mobile application could be considered to address this limitation.
- ❖ **Data Privacy and Compliance :**
  - Adhering to data privacy regulations (such as GDPR) and other compliance standards is crucial. Ensure that the system complies with legal requirements relevant to the jurisdiction in which it operates.
- ❖ **Resource Constraints :**
  - The software might have limitations based on the hardware resources available. This includes considerations for server capacity, processing power, and memory, especially when dealing with large datasets.
- ❖ **Customization Limitations :**
  - Users might have limited ability to customize the software according to their specific needs. Providing a flexible and extensible architecture can help overcome this limitation to some extent.
- ❖ **Learning Curve for Users :**
  - The software might have a learning curve for users unfamiliar with Python or MySQL. Providing sufficient documentation and user training can mitigate this delimitation.

CHAPTER 6

CONCLUSION

In concluding a smart college system software that leverages data science and machine learning technologies, the integration of Python and MySQL with these advanced capabilities brings transformative and intelligent dimensions to the college system. Here arekey points for a conclusion :

- ❖ **Enhanced Decision-Making :**
  - The incorporation of data science allows for the analysis of large datasets, enabling administrators to make data-driven decisions. Machine Learning models can provide valuable insights into student performance, resource allocation, and overall academic trends.
- ❖ **Predictive Analytics for Student Success :**
  - Machine learning algorithms can be employed to predict student success factors, helping educators and administrators identify at-risk students early on. This proactive approach allows for timely interventions and personalizedsupport to enhance student outcomes.
- ❖ **Optimized Resource Allocation :**
  - Through data analysis, the system can optimize resource allocation based on historical data and predictive models. This includes faculty workload management, classroom utilization, and budget planning, leading to more efficient use of resources.
- ❖ **Personalized Learning Paths :**
  - Machine Learning algorithms can tailor educational content and learning paths based on individual student progress and preferences. This personalization fosters a more engaging and effective learning experience for students.
- ❖ **Continuous Improvement with Feedback Loops :**

- By implementing machine learning models that learn from user interactions and feedback, the system can continuously improve its recommendations, adapt to changing trends, and enhance overall user satisfaction.

❖ **Challenges in Implementation :**

- Despite the potential benefits, the implementation of data science and machine learning technologies comes with challenges. These include the need for high-quality, labeled datasets, ongoing model training and validation, and addressing biases in algorithms.

❖ **Ethical Considerations :**

- The use of machine learning in educational settings requires careful consideration of ethical implications. Ensuring fairness, transparency, and accountability in algorithmic decision-making is essential to build trust among users.

❖ **Integration with Existing Systems :**

- Integrating data science and machine learning features seamlessly with the existing Python and MySQL infrastructure may require careful planning and development. Compatibility, data consistency, and system performance should be prioritized.

❖ **Continuous Adaptation :**

- The field of data science and machine learning is dynamic, with advancements and new techniques emerging regularly. The smart college system should be designed with flexibility to incorporate future developments and ensure its relevance over time.

In conclusion, the integration of data science and machine learning technologies into a smart college system powered by Python and MySQL holds immense potential for revolutionizing education management. While challenges and ethical considerations exist, the benefits in terms of informed decision-making, personalized learning experiences, and resource optimization position the system as a powerful tool for enhancing the overall educational ecosystem.

CHAPTER 7

BIBLIOGRAPHY

7.1 References

[1] Zhi-gang YUE, You-Wei JIN “The development and design of the student management system based on the network environment-20 January 2011”.

[2] Katsaros "A Holistic View of Information Management in Cloud Environments-02 August 2012".

[3] Yang "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing-24 September 2012".

[4] S.R. Bharam Ago Udar, Geeta R.B., S.G. To tad “Web-Based Student Information Management - 6 June 2023”.

[5] Andreas Papadakis “Profiling Students' Performance and Measuring their Progress in the area of Multimedia Communications-3-5 April 2014”.

[6] Goyal “An effective algorithmic approach for cost optimization in Cloud-based data center-03 April 2014".

[7] Saakshi Narula "Cloud Computing Security: Amazon Web Service-06 April 2015".

[8] Rafiqul Zaman Khan "A Study of Cloud Computing-April 2015".

[9] Siddhant Gok Ule, Rohit Jadhav, Murugan Ayani Mane, Sanchita ChShajed “An Application to Enhance the Admission Process and Communication between Members of an Institute-April 2015".

[10] Mehdi “IEEE International Conference on Mobile Cloud Computing-25 June 2015”.

[11] Lalit Mohan Joshi “This document is the Software Requirement Specification(SRS) for the College System Software for College (CSS) project -11 July 2015”.

[12] Nitin Naik "Connecting Google Cloud system with organizational systems for SS effortless data analysis-24 November 2016".

[13] Anand Desai, “Student Profiling to improve teaching and learning: A Data Mining Approach-19 January 2017".

[14] Mar Bukh “Systemic Risks in the Cloud Computing Model: Complex Systems Perspective-19 January 2017".

[15] Charlotte Kotas "A Comparison of Amazon Web Services and Microsoft Azure Cloud Platforms for High-Performance Computing-29 March 2018".

[16] Fortunato "Progressive web apps: An alternative to the native mobile Apps-28 June 2018".

[17] Behl “Architectural Pattern of Progressive Web and Background Synchronization-23 August 2018".

[18] CAI Chang-an, WANG Qi “Design and implementation of student information management system based on B/S model-7 July 2020”.

[19] Achal Agrawal “Comparison of Flutter with Other Development Platforms-02 February 2021".

[20] Dhananjay Sinh “A Study on Progressive Web Apps as a Unifier for NativeApps and the Web-05 May 2021".

7.2 Snapshot

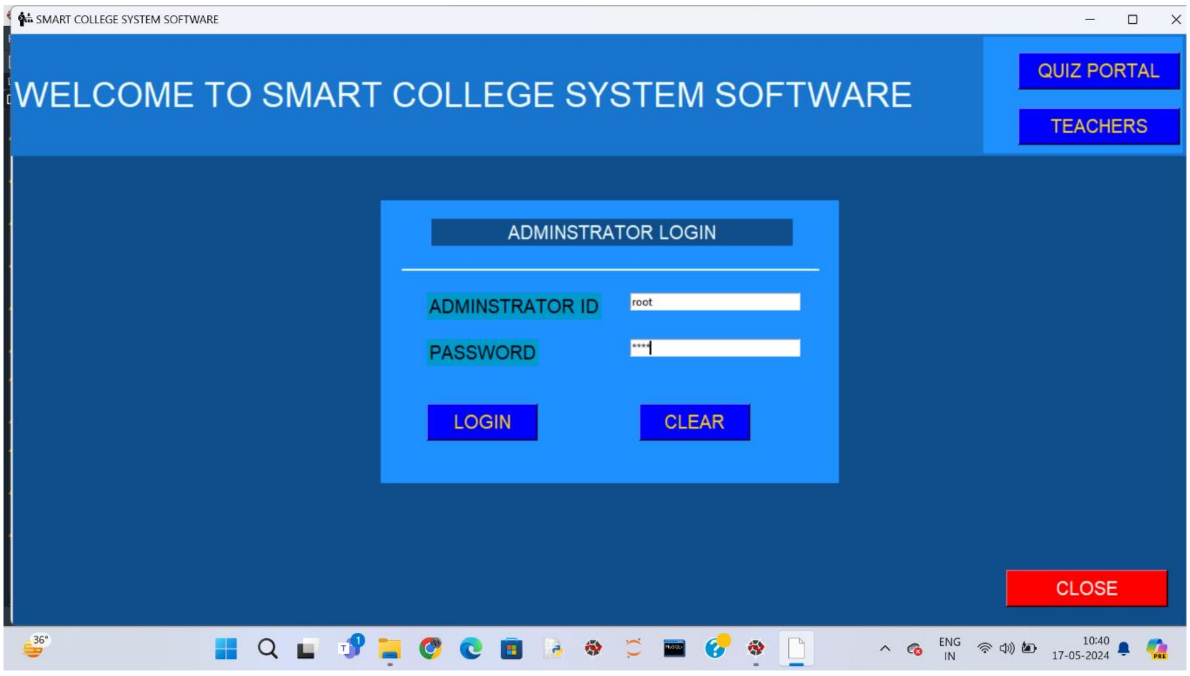


Fig 7.2.1 ADMINISTRATOR LOGIN

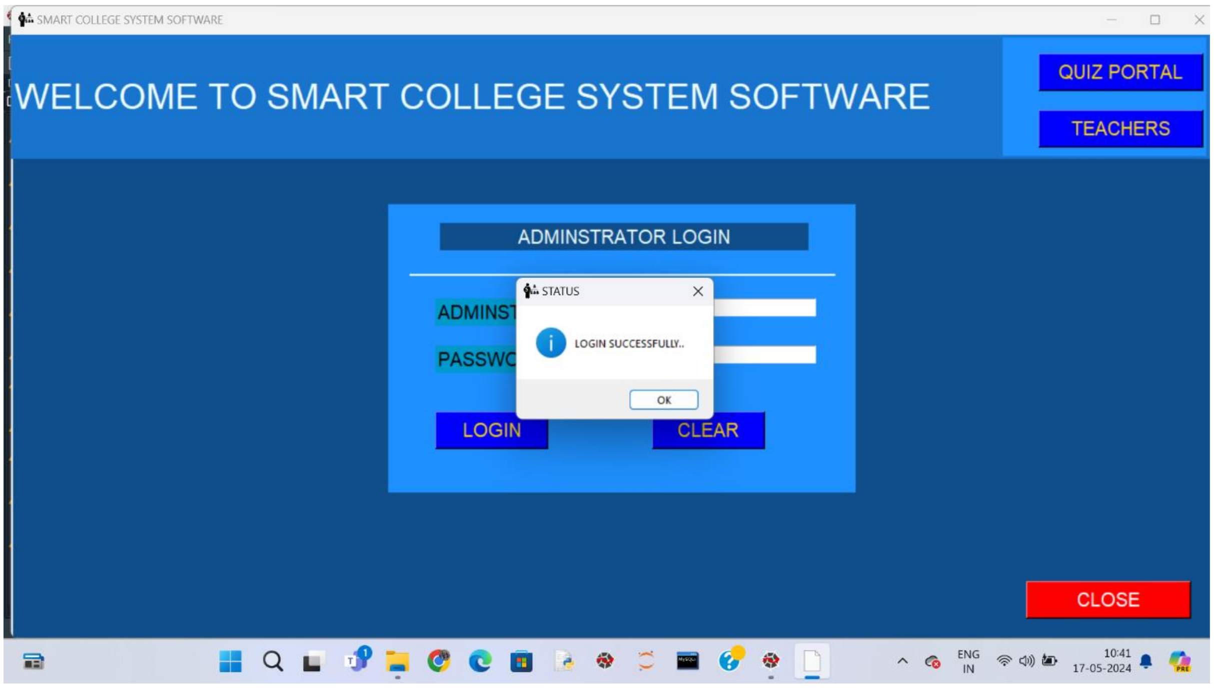


Fig 7.2.2 LOGIN SUCCESSFUL

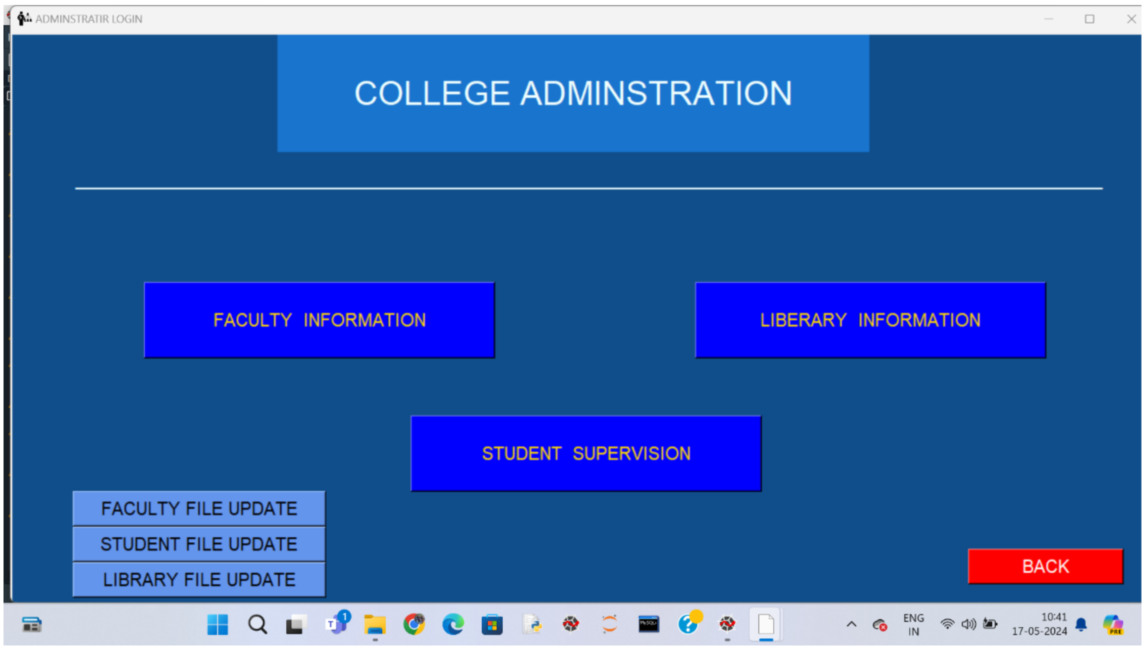
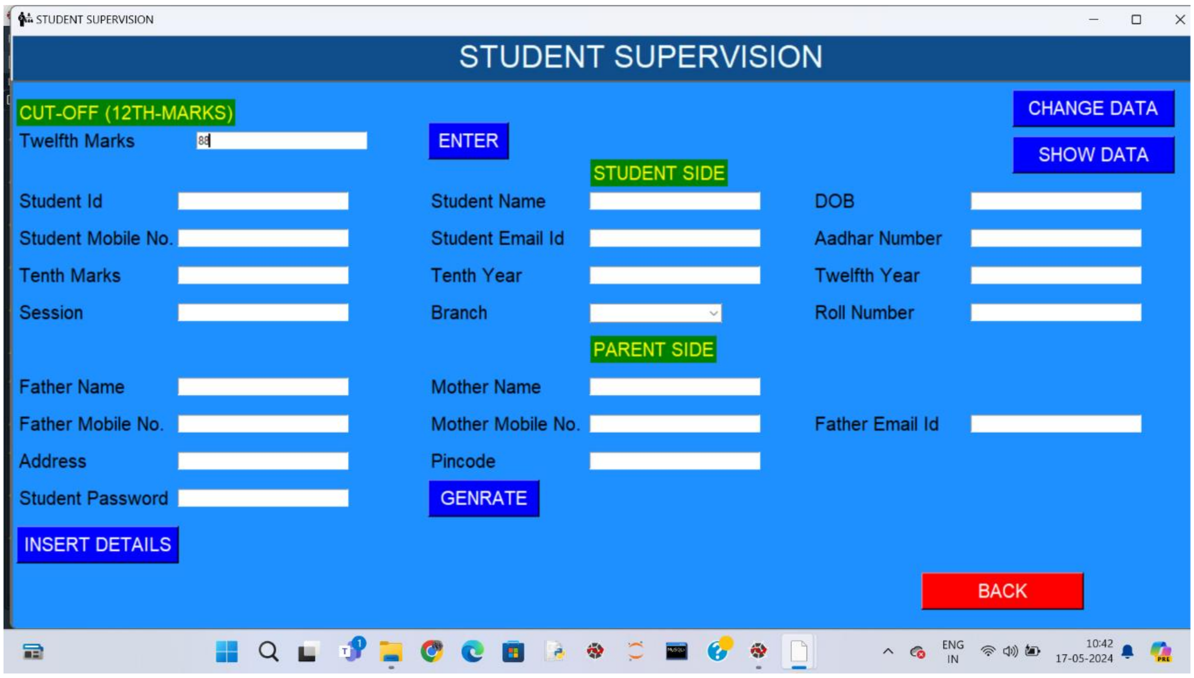
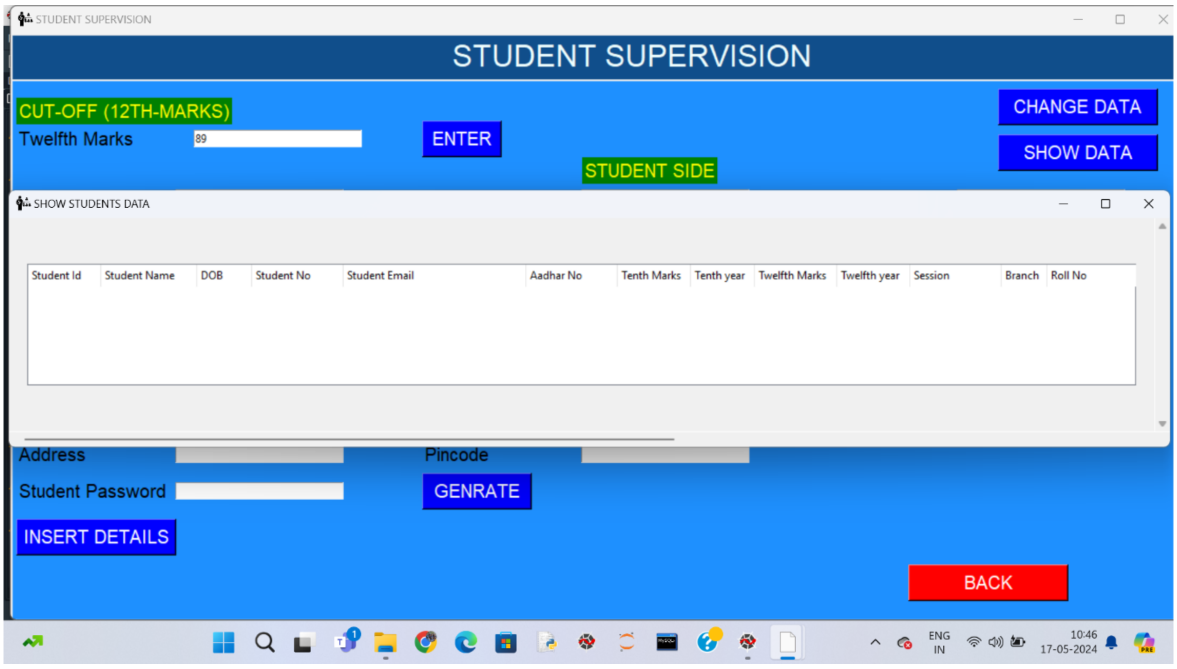


Fig 7.2.3 COLLEGE ADMINISTRATION



Twelfth Marks	88				
Student Id		Student Name		DOB	
Student Mobile No.		Student Email Id		Aadhar Number	
Tenth Marks		Tenth Year		Twelfth Year	
Session		Branch		Roll Number	
Father Name		Mother Name			
Father Mobile No.		Mother Mobile No.		Father Email Id	
Address		Pincode			
Student Password					

Fig 7.2.4 STUDENT SUPERVISION


STUDENT SUPERVISION

STUDENT SUPERVISION

CUT-OFF (12TH-MARKS)

Twelfth Marks 89

ENTER

STUDENT SIDE

CHANGE DATA

SHOW DATA

SHOW STUDENTS DATA

Student Id	Student Name	DOB	Student No	Student Email	Aadhar No	Tenth Marks	Tenth year	Twelfth Marks	Twelfth year	Session	Branch	Roll No

Address

Pincode

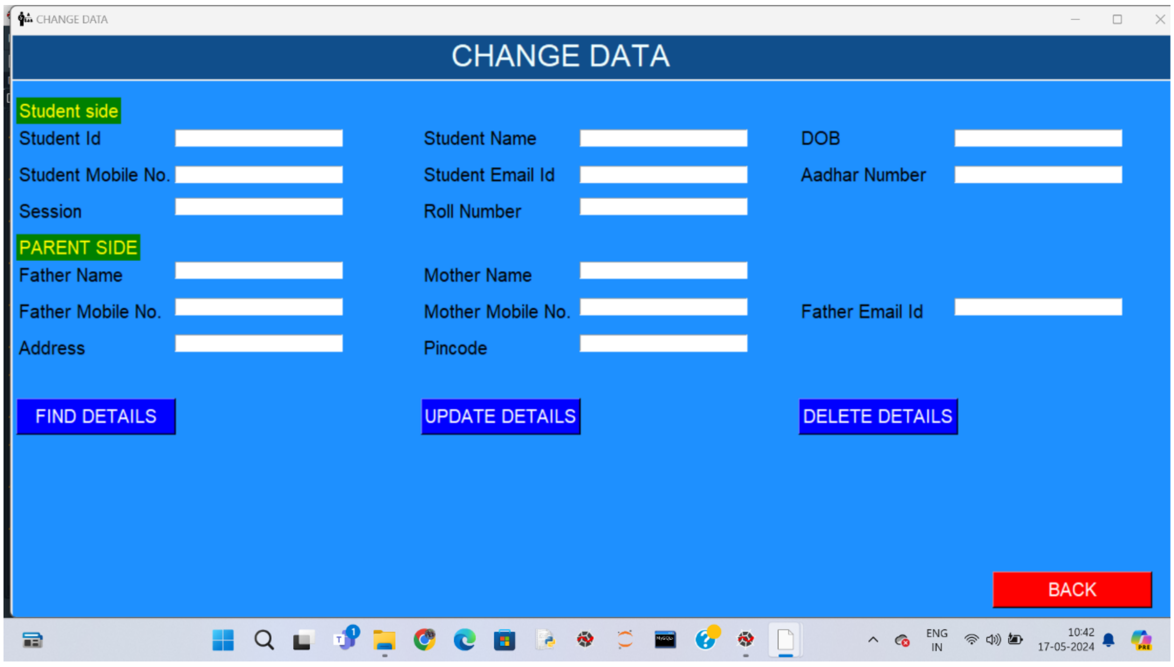
Student Password

GENRATE

INSERT DETAILS

BACK

Fig 7.2.5 STUDENT SUPERVISION DATA


CHANGE DATA

CHANGE DATA

Student side

Student Id

Student Name

DOB

Student Mobile No.

Student Email Id

Aadhar Number

Session

Roll Number

PARENT SIDE

Father Name

Mother Name

Father Mobile No.

Mother Mobile No.

Father Email Id

Address

Pincode

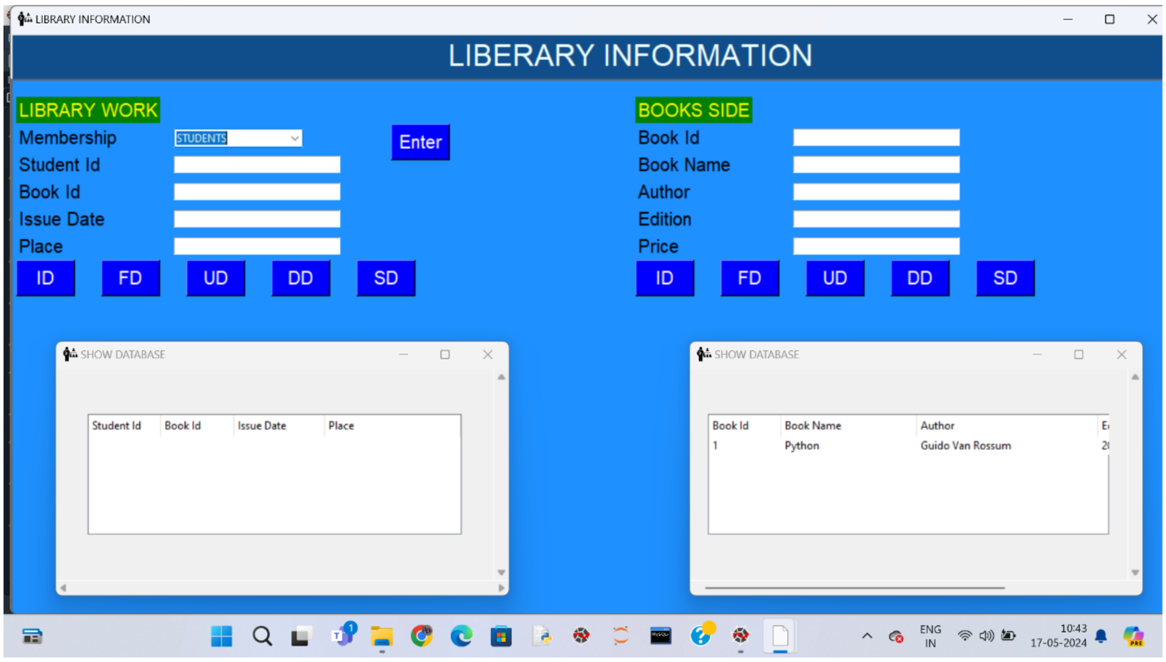
FIND DETAILS

UPDATE DETAILS

DELETE DETAILS

BACK

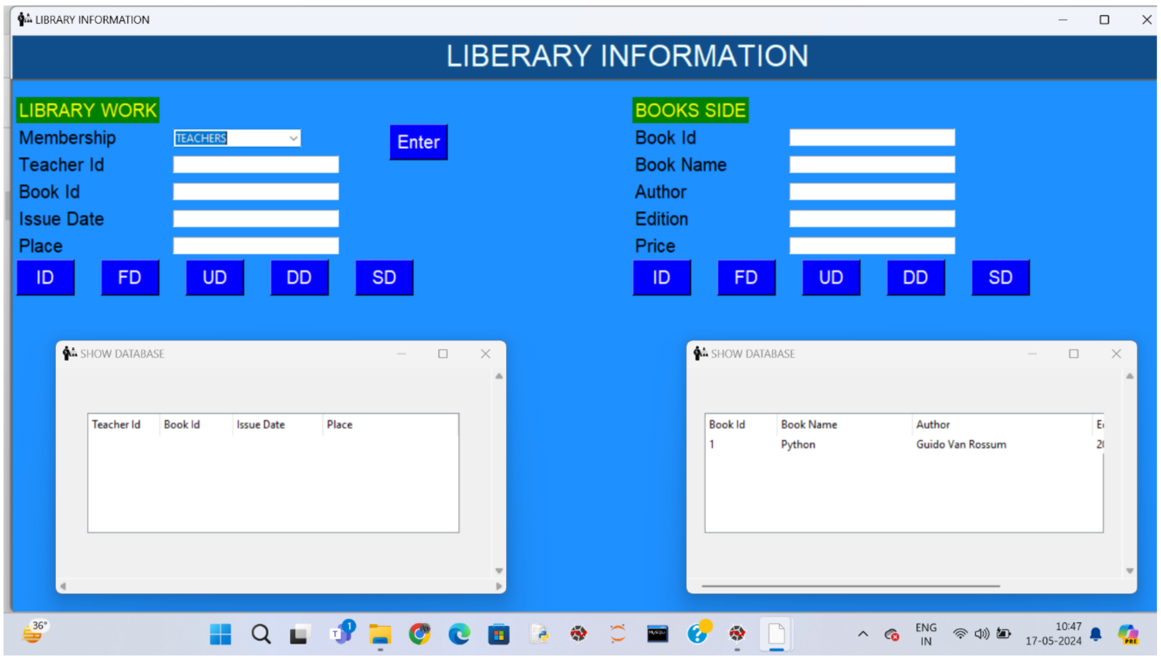
Fig 7.2.6 STUDENT CHANGE DATA



Student Id	Book Id	Issue Date	Place

Book Id	Book Name	Author	E
1	Python	Guido Van Rossum	2

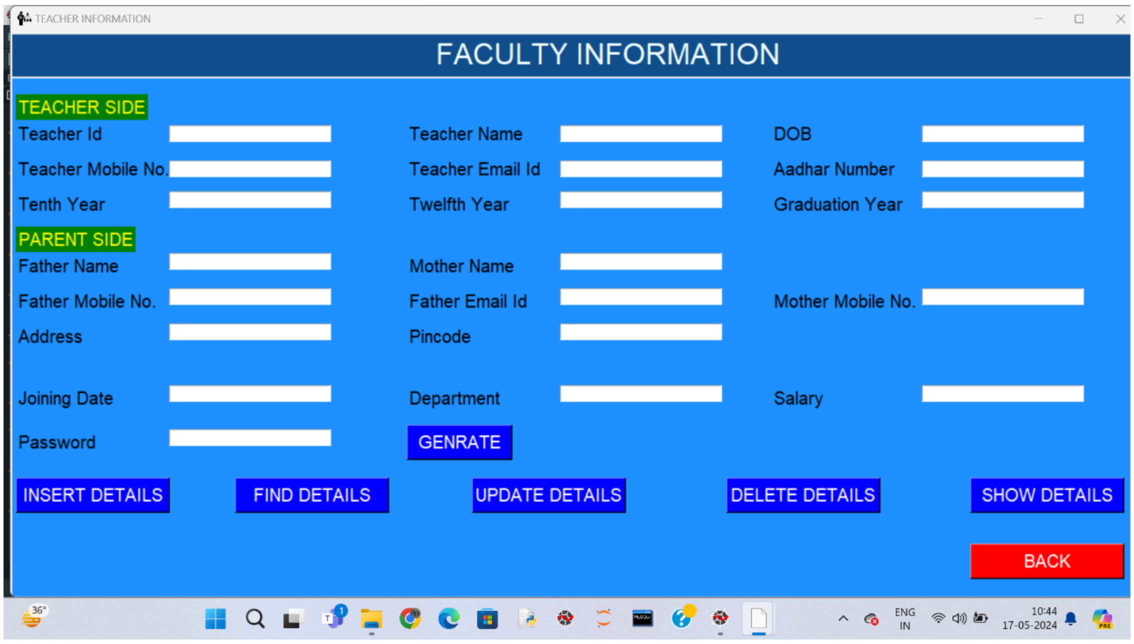
Fig 7.2.7 LIBRARY INFORMATION FOR STUDENT



Teacher Id	Book Id	Issue Date	Place

Book Id	Book Name	Author	E
1	Python	Guido Van Rossum	2

Fig 7.2.8 LIBRARY INFORMATION FOR TEACHER



TEACHER INFORMATION

FACULTY INFORMATION

TEACHER SIDE

Teacher Id		Teacher Name		DOB	
Teacher Mobile No.		Teacher Email Id		Aadhar Number	
Tenth Year		Twelfth Year		Graduation Year	

PARENT SIDE

Father Name		Mother Name			
Father Mobile No.		Father Email Id		Mother Mobile No.	
Address		Pincode			
Joining Date		Department		Salary	
Password		GENRATE			

Fig 7.2.9 FACULTY INFORMATION

7.3 APPENDICES

CODE:

```
import tkinter
from tkinter import *
from tkinter import messagebox
from tkinter import ttk
import pymysql
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
import random
import csv

screen=tkinter.Tk()
screen.geometry("1380x640+0+0")
screen.title("SMART COLLEGE SYSTEM SOFTWARE")
screen.iconbitmap('LOGO.ico')

def LOGIN():  # for Login the administrator
      adminstrator=A1.get()
     password=B1.get()

                 if adminstrator == "root" and password == "root" :
                 messagebox.showinfo("STATUS","LOGIN SUCCESSFULLY..")
 t=Toplevel(screen)
 t.geometry("1380x640+0+0")
 t.title("ADMINSTRATIR LOGIN")
 t.iconbitmap('LOGO.ico')

 def STUDENT_SUPERVISION(): # for Student Supervision Screen
     ss=Toplevel(t)
     ss.geometry("1380x640+0+0")
     ss.title("STUDENT SUPERVISION")
     ss.iconbitmap('LOGO.ico')

     def enter1():

         Twelfth=twelfth_marks_entry.get()


         if Twelfth >= '85':

             def genrate():

                 lower="abcdefghijklmnopqrstuvwxyz"
                 upper="ABCDEFGHIJKLMNOPQRSTUVWXTZ"
                 numbers="0123456789"
                 symbols="!@#$%^&*()_+=?><][" 
                 all_chars= lower + upper + numbers + symbols
```

```
            length=10

            password1= ''.join(random.sample(all_chars,length))

            print1=Label(SS_Home1,text=password1,font=25)
            print1.place(x=650,y=435)

        def ID():


db=pymysql.connect(host='localhost',user='root',password='root',database='scss')
            cur=db.cursor()
            a1=int(id_entry.get())
            a2=str(name_entry.get())
            a3=int(dob_entry.get())
            a4=int(mobile_no_entry.get())
            a5=str(email_id_entry.get())
            a6=int(aadhar_no_entry.get())
            a7=int(tenth_marks_entry.get())
            a8=int(tenth_year_entry.get())
            a9=int(twelfth_marks_entry.get())
            a10=int(twelfth_year_entry.get())
            a11=int(session_entry.get())
            a12=str(branch1.get())
            a13=int(roll_no_entry.get())
            a14=str(father_name_entry.get())
            a15=str(mother_name_entry.get())
            a16=int(father_mobile_entry.get())
            a17=str(father_email_entry.get())
            a18=int(mother_mobile_entry.get())
            a19=str(Address_entry.get())
            a20=int(Pincode_entry.get())
            a21=str(s_password_entry.get())
                sql="insert                    into                    students
values('%d','%s','%d','%d','%s','%d','%d','%d','%d','%d','%d','%s','%d','%s','%s','%d','%
s','%d','%s','%d','%s')"%(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,
a18,a19,a20,a21)
            cur.execute(sql)
                sql="select
Student_Id,Student_Name,DOB,Student_Mobile_No,Aadhar_No,Session,Roll_No,Fa
ther_Name,Mother_Name,Father_Mobile_No,Mother_Mobile_No,Father_Email_Id,
Address,Pincode    from    students    where    Student_Email_Id='%s'and
Student_Id='%d'"%(a5,a1)
            cur.execute(sql)
            data=cur.fetchone()
            r=list(data)
            a={}
            a['Student Id']=r[0]
            a['Student Name']=r[1]
            a['DOB']=r[2]
            a['Student Mobile Number']=r[3]
            a['Aadhar Number']=r[4]
            a['Session']=r[5]
            a['Roll Number']=r[6]
```

```
a['Father Name']=r[7]
a['Mother Name']=r[8]
a['Father Mobile Number']=r[9]
a['Mother Mobile Number']=r[10]
a['Father Email Id']=r[11]
a['Address']=r[12]
a['Pincode']=r[13]
def send_email(sender_email, password, recipient_email, subject, message):
    try:
        # Connect to SMTP server
        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.starttls()
        server.login(sender_email, password)

        # Create MIMEText object for the email body
        msg = MIMEMultipart()
        msg['From'] = sender_email
        msg['To'] = recipient_email
        msg['Subject'] = subject

        # Convert dictionary to a formatted string
                        message_text = "\n".join([f"{key}: {value}" for key, value
in message.items()])
        msg.attach(MIMEText(message_text, 'plain'))

        # Send email
        server.sendmail(sender_email, recipient_email, msg.as_string())

        print("Email sent successfully!")
        server.quit()

    except Exception as e:
        print(f"Failed to send email. Error: {e}")

data= a

# Email configuration
sender_email = "ayushverma63210@gmail.com"
password = "neel ipmt hhtw undg"
recipient_email = "%s"%(a5)
                subject = "CONGRATULATIONS..! Your Application Has
Been Registered Successfully."
message =  data # Dictionary data to be sent in email body

# Call function to send email
send_email(sender_email, password, recipient_email, subject, message)

#print(a)
db.commit()
db.close()
messagebox.showinfo('STATUS','Data Saved and Emailed')
def FACULTY_INFORMATION(): # for Faculty Information Screen
 fi=Toplevel(t)
 fi.geometry("1380x640+0+0")
```

```
             fi.title("TEACHER INFORMATION")
            fi.iconbitmap('LOGO.ico')
            def t_genrate():
              lower="abcdefghijklmnopqrstuvwxyz"
              upper="ABCDEFGHIJKLMNOPQRSTUVWXTZ"
            numbers="0123456789"
             symbols="!@#$%^&*()_+=?><][["
            all_chars= lower + upper + numbers + symbols
            length=10
            password1= ''.join(random.sample(all_chars,length))
            print2=Label(FI_Home1,text=password1,font=25)
            print2.place(x=640,y=400)
            def t_close_back():
            fi.destroy()

def t_ID():
   db=pymysql.connect(host='localhost',user='root',password='root',database='scss')
   cur=db.cursor()
   t1=int(teacher_id_entry.get())
   t2=str(teacher_name_entry.get())
   t3=int(t_dob_entry.get())
   t4=int(teacher_mobile_no_entry.get())
   t5=str(t_email_id_entry.get())
   t6=int(t_aadhar_no_entry.get())
   t7=int(t_tenth_entry.get())
   t8=int(t_twelfth_entry.get())
   t9=int(t_graduation_entry.get())
   t10=str(t_father_name_entry.get())
   t11=str(t_mother_name_entry.get())
   t12=int(t_father_mobile_entry.get())
   t13=str(t_father_email_entry.get())
   t14=int(t_mother_mobile_entry.get())
   t15=str(t_Address_entry.get())
   t16=int(t_Pincode_entry.get())
   t17=int(joining_entry.get())
   t18=str(department_entry.get())
   t19=int(salary_entry.get())
   t20=str(t_Password_entry.get())
         sql="insert                    into                    teachers
        values('%d','%s','%d','%d','%s','%d','%d','%d','%d','%s','%s','%d','%s','%d','%s','
        %d','%d','%s','%d','%s')"%(t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t
        17,t18,t19,t20)
         cur.execute(sql)
   #sql="select
Teacher_Id,Teacher_Name,DOB,Teacher_Mobile_No,Aadhar_No,Tenth_Year,Twelf
th_Year,Graduation_Year,Father_Name,Mother_Name,Father_Mobile_No,Father_E
mail_Id,Mother_Mobile_No,Address,Pincode,Joining_Date,Department,Salary   from
teachers where Teacher_Email_Id='%s'and Teacher_Id='%d'"%(t5,t1)
   #cur.execute(sql)
   db.commit()
   db.close()
   messagebox.showinfo('STATUS','Data Saved')
   fi.destroy()
def LIBERARY_INFORMATION():  # for Liberary Information Screen
```

```
li=Toplevel(t)
li.geometry("1380x640+0+0")
li.title("LIBRARY INFORMATION")
li.iconbitmap('LOGO.ico')

def enter():
   Membership=membership_entry.get()
 if Membership == 'STUDENTS':
def S_ID():

db=pymysql.connect(host='localhost',user='root',password='root',database='scss')
         cur=db.cursor()
         s1=int(student_id_entry.get())
         s2=str(book_id_entry.get())
         s3=int(issue_date_entry.get())
         s4=str(place_entry.get())
         sql="insert into slibrary values('%d','%s','%d','%s')"%(s1,s2,s3,s4)
         cur.execute(sql)
         db.commit()
         db.close()
         messagebox.showinfo('STATUS','Data Saved')
 def S_FD():
      db=pymysql.connect(host='localhost',user='root',password='root',database='scss')
         cur=db.cursor()
         s9=int(student_id_entry.get())
                  sql="select    Book_Id,Issue_Date,Place    from    slibrary    where
Student_Id='%d'"%(s9)
         cur.execute(sql)
         data=cur.fetchall()
         if len(data)==0:
            messagebox.showinfo('STATUS',"No records found")
         else:
            for res in data:
               book_id_entry.insert(0,str(res[0]))
               issue_date_entry.insert(0,int(res[1]))
               place_entry.insert(0,str(res[2]))
         db.close()
def S_UD():
      db=pymysql.connect(host='localhost',user='root',password='root',database='scss')
         cur=db.cursor()
         s5=int(student_id_entry.get())
         s6=str(book_id_entry.get())
         s7=int(issue_date_entry.get())
         s8=str(place_entry.get())
                  sql=("update  slibrary  set  Book_Id='%s',Issue_Date='%d',Place='%s'
where Student_Id='%d'")%(s6,s7,s8,s5)
         cur.execute(sql)
         db.commit()
         db.close()
         messagebox.showinfo('STATUS','Data Updated')
def S_DD():
      db=pymysql.connect(host='localhost',user='root',password='root',database='scss')
         cur=db.cursor()
         s10=int(student_id_entry.get())
```

```
            sql="delete from slibrary where Student_Id='%d'"%(s10)
            cur.execute(sql)
            db.commit()
            db.close()
            messagebox.showinfo("STATUS",'DELETED')
def ScreenClear():
 A1.delete(0,100)
 B1.delete(0,100)
def ScreenClose():
  screen.destroy()
Home=Canvas(screen,width=1380,height=640,bg='#104E8B')
Home.place(x=0,y=0)
Welcome=Label(screen,text="WELCOME   TO   SMART   COLLEGE   SYSTEM
SOFTWARE",anchor="w",bg="#1874CD",fg="#F0FFFF",width=100,height=3,font=
('Aeries',28))
            Welcome.pack()
Screenclose=Button(screen,text="CLOSE",width=15,font=10,bg='Red',fg='white',co
mmand=ScreenClose)Screenclose.place(x=1077,y=580)
SideLabel=Label(screen,bg="#1E90FF",width=30,height=8)
SideLabel.place(x=1052,y=3)
QuizPortal=Button(screen,text="QUIZ
PORTAL",width=15,font=10,bg='blue',fg='#FFD700')
QuizPortal.place(x=1090,y=20)
Attendance=Button(screen,text="TEACHERS",width=15,font=10,bg='blue',fg='#FFD
700')
Attendance.place(x=1090,y=80)
MidLabel=Label(screen,bg='#1E90FF',width=70,height=20)
MidLabel.place(x=400,y=180)
LoginName=Label(screen,text="ADMINSTRATOR
LOGIN",width=35,bg="#104E8B",fg="#F0FFFF",font=50)
LoginName.place(x=455,y=200)
            Line=Label(screen,text="___________________________________________",
bg="#1E90FF",fg="#F0FFFF",font=50)
Line.place(x=420,y=230)
A=Label(screen,text="ADMINSTRATOR ID",bg="#009ACD",font=5)
A.place(x=450,y=280)
A1=Entry(screen,width=30)
A1.place(x=670,y=280)
B=Label(screen,text="PASSWORD",bg="#009ACD",font=5)
B.place(x=450,y=330)
B1=Entry(screen,width=30,show='*')
B1.place(x=670,y=330)
C=Button(screen,text="LOGIN",width=10,font=10,bg='blue',fg='#FFD700',command
=LOGIN)
C.place(x=450,y=400)
C1=Button(screen,text="CLEAR",width=10,font=10,bg='blue',fg='#FFD700',comma
nd=ScreenClear)
C1.place(x=680,y=400)
screen.mainloop()
```

7.1 Curriculum Vitae

Dr. Brajesh Kumar Singh

Dr. Brajesh Kumar Singh was born in District Agra (U.P.) in 1978. He completed his doctorate in Computer Science and Engineering from Motilal Nehru National Institute of Technology, Allahabad (U.P.) in the year 2014. He joined as a Lecturer./ Asstt. Prof. at R.B.S. Engineering Technical Campus, Bich puri, Agra in the Year 2001. In the year 2007, he was appointed as Reader/ Assoc. Prof. in the same organization. In December 2017, he took over charge as Head of The Department of Computer Science and Engineering. In Oct 2018, he was promoted to the post of Professor. He has guided more than 50 B. Tech. and 9 M. Tech. projects of National and international repute. He is supervising 2 Ph.D. candidates. He has 50 publications to his credit in national and international journals and proceedings of high repute with a large number of citations of his research manuscripts. Dr. Singh has delivered several invited talks/ keynote addresses and chaired sessions at national and international conferences of high repute in India and abroad. He is having collaborative training programs/workshops with IIT Bombay. He significantly contributed to enhancing the research standards in the department of CSE. He is receives IBM’s best project awards. Dr. Singh has organized successfully more than 45 International and National Conferences/Seminars/Workshops as an organizing secretary/ member of the International Program Committee in India and abroad.

Academic Qualification

Ph. D. in CSE

Designation with Department

Professor & Head

Computer Sci. & Engineering

Specialization

Computer Science and Engineering

Experience - 19 Years and 6 Months

Er. Alok Singh Jadaun

Er. Alok Singh Jadaun is currently serving as Assistant Professor of Department of the Post Graduate Department of Computer Science & Engineering of the Raja Balwant Singh Engineering Technical Campus, Bich Puri, Agra. He obtained his B. Tech degree in Computer Science and Engineering from U.P.T.U with First Division in 2009. He obtained a Master of Technology (M. Tech) degree from Bhagwant University, Ajmerin Computer Science & Engineering with First Division in 2014. He has nine years and seven months of experience. He is presently engaged in research and development activities in the areas of Data Structure, Cryptography and Network Security, Computer Networks, and Distributed Systems.

Academic Qualification - B. Tech, M. Tech.
Designation with Department - Assistant Professor Computer Sci. & Engineering
Contact No – 9639125689, 7906813603
Email - alok.singh3131@gmail.com

Specialization - Data Structure, Computer Networks, Cryptography and Network Security, Software Engineering Object Oriented Programming

Experience - 9 Years 7 Months

Awards and Recognitions

Present Area of work - Data Structure, Distributed Systems, Computer Network, Data Structure and Software Engineering, IoT.

ABHINAV VERMA

Abhinav is a final-year student of Computer Science & Engineering at Raja Balwant Singh Engineering Technical Campus, Agra. He has completed his High School and Intermediate from CBSE board with 65% and 66%. He is skilled in Python, Data Science, Machine Learning, GUI (Tk inter), Django, Flask, MySQL, Mongo DB, Power BI, Tableau, Statistical Analysis, and SDLC.

Aditya Kumar Singh

Aditya is a final-year student of Computer Science & Engineering at Raja Balwant Singh Engineering Technical Campus, Agra. He has completed his High School and Intermediate from CBSE board with 77.6% and 74.6%. He is skilled in Python, C/C++, MySQL, HTML, Software Engineering, Data Structure & Algorithms.