STREAMLINED PROJECT COLLABORATION AND EFFICIENT MANAGEMENT TOOL

A MINI PROJECT REPORT

Submitted by

GOKULRAJ S P	**AC20UCS033**
GOPALA G	**AC20UCS034**
HARISH M	**AC20UCS040**
GOPI V	**AC20UCS154**

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)

Dr. M.G.R NAGAR, HOSUR - 635109

ANNA UNIVERSITY : CHENNAI - 600025

NOVEMBER 2023

STREAMLINED PROJECT COLLABORATION AND EFFICIENT MANAGEMENT TOOL

A MINI PROJECT REPORT

Submitted by

GOKULRAJ S P	**AC20UCS033**
GOPALA G	**AC20UCS034**
HARISH M	**AC20UCS040**
GOPI V	**AC20UCS154**

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)

Dr. M.G.R NAGAR, HOSUR - 635109

ANNA UNIVERSITY : CHENNAI - 600025

NOVEMBER 2023

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this mini project report **“STREAMLINED PROJECT COLLABORATION AND EFFICIENT MANAGEMENT TOOL”** is the bonafide work of **“GOKULRAJ S P (AC20UCS033), GOPALA G (AC20UCS034), HARISH M (AC20UCS040), GOPI V (AC20UCS154)”** who carried out the project under my supervision.

SIGNATURE

Dr. G. FATHIMA, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR,
Department of CSE,
Adhiyamaan College of Engineering,
Dr. M.G.R. Nagar,
Hosur – 635 130.

SIGNATURE

Mrs. R L INDU LEKHA, M.E.,
SUPERVISOR
ASSISTANT PROFESSOR,
Department of CSE,
Adhiyamaan College of Engineering,
Dr. M.G.R. Nagar,
Hosur – 635 130.

Submitted for the Mini project VIVA-VOCE Examination held on ____________________ at **Adhiyamaan College of Engineering (Autonomous), Hosur-635 109.**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, we thank the Almighty for granting us the wisdom. Strength and grace to complete the report and for being with us in every step that we look in order to complete the project successfully.

We gratefully acknowledge our indebtedness to Adhiyamaan College of Engineering. Hosur for making us a part of their team in developing this project work and provide us with valuable information.

We would like to express our sincere thanks to our beloved Principal **Dr. G. RANGANATH, M.E., Ph.D.,** Principal, Adhiyamaan College of Engineering, Hosur, for his keen interest and affection toward us.

We are highly indebted to **Dr. G. FATHIMA, M.E., Ph.D.,** Professor, Head of the Department and Supervisor, Department of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, whose immense support, encouragement and valuable guidance was responsible to complete this project successfully.

We acknowledged our gratitude to **Mrs. R. L. INDU LEKHA, M.E.,** Assistant professor, Department of Computer Science And Engineering, Adhiyamaan College of Engineering (Autonomous), Hosur whose immense Support, encouragement and valuable guidance made us complete this project successfully.

We also extend our thanks to the Project Coordinator **Mrs.V.KALAIVANI, M.E., (Ph.D.,)** and all Staff Members for their support in completing this project successfully.

We also extend our thanks to Project Coordinator, and Staff Members and our parents for their support in completing the project successfully.

ABSTRACT

"Streamlined Project Collaboration and Efficient Management Tool" introduces a forward-thinking web application poised to revolutionize the way students and faculty manage projects. In traditional academic settings, project management has often been burdened by manual, paper-based processes and reliance on spreadsheets, leading to inefficiencies, communication challenges, and data discrepancies. This project offers a game-changing solution that replaces these outdated methods with a cutting-edge digital platform, empowering students to effortlessly update and maintain their project details in real-time. This innovative web application provides individual project dashboards, task management tools, progress tracking capabilities, and a centralized document repository. Not only does it facilitate smoother collaboration for students, but it also equips staff with the tools they need to manage project teams, allocate resources efficiently, and monitor project timelines with ease. With collaborative spaces for teams to communicate and share ideas, the platform fosters a dynamic and interactive project environment. By digitizing and centralizing project data, this solution empowers educational institutions to improve the efficiency and effectiveness of project management, resulting in enhanced project outcomes and a richer learning experience. "Streamlined Project Collaboration and Efficient Management Tool" is a critical step towards modernizing project management within educational settings. It serves as the linchpin for modernizing project management in educational settings, aligning it with the digital era's efficiency and effectiveness.

***Keywords*:** Project Management, Digital Transformation, Image Real-time Updates, Task Management, Progress Tracking, Document Repository

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ACRONYM		ABBREVIATIONS
API	-	APPLICATION PROGRAMMING INTERFACE
CS	-	CENTRALIZED SYSTEM
CSS	-	CASCADING STYLE SHEET
DB	-	DATABASE
HTML	-	HYPER TEXT MARKUP LANGUAGE
HTTP	-	HYPERTEXT TRANSFER PROTOCOL
IDE	-	INTEGRATED DEVELOPMENT ENVIRONMENT
JSX	-	JAVASCRIPT XML
PTS	-	PROJECT TRACKING SYSTEM
SQL	-	STRUCTURED QUERY LANGUAGE
UI	-	USER INTERFACE

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

In today's digitally-driven era, the traditional methods of managing student projects in academic institutions have been laborious and prone to inefficiencies. This project, titled "Streamlined Project Collaboration and Efficient Management Tool," represents a significant leap forward in modernizing project management for students and staff. It addresses the challenges associated with manual, paper-based processes and reliance on spreadsheets by providing a comprehensive digital solution. The core objective of this project is to offer a user-friendly web application that replaces these outdated management methods. With this innovative tool, students gain the ability to effortlessly maintain and update their project details in real-time, eliminating the need for cumbersome paperwork and manual data entry. Key features of this application include individual project dashboards, task management tools, progress tracking capabilities, and a centralized document repository. These features not only enhance the collaboration experience for students but also empower staff to efficiently manage project teams, allocate resources, and monitor project timelines. Furthermore, the platform creates collaborative spaces for teams to communicate and share ideas, fostering a dynamic and interactive project environment. By digitizing and centralizing project data, the solution greatly enhances the efficiency and effectiveness of project management within educational institutions. It streamlines project oversight, reduces the potential for miscommunication and data discrepancies, and ultimately leads to improved. In summary, "Streamlined Project Collaboration and Efficient Management Tool" is a transformative project that promises to revolutionize the way projects are managed in academic settings. By embracing digital solutions, it provides a more efficient, streamlined, and collaborative approach to project management, enhancing the overall educational experience for both students and staff. Moreover, this web application establishes collaborative spaces where teams can communicate and share ideas seamlessly, fostering a dynamic and interactive project environment. By digitizing and centralizing project data, the solution contributes significantly to the efficiency and effectiveness of project management within educational institutions. It reduces the risk of miscommunication and data errors, freeing up more time for students and staff to focus on the substance of their projects. In conclusion, "Streamlined Project Collaboration and Efficient Management Tool" stands as a vital and transformative project that promises to modernize project management in the educational sector. By embracing digital tools, it offers a more efficient and streamlined approach, ultimately leading to enhanced project outcomes and a richer learning experience for all stakeholders involved. This project is poised to be a catalyst for change in the way educational management. In traditional academic settings, project management has been encumbered by manual processes, predominantly paper-based and reliant on spreadsheet-driven methodologies. These archaic methods have often resulted in inefficiencies, communication challenges, and data discrepancies. This project aims to be the catalyst for change by introducing a state-of-the-art web application that practices. "Streamlined Project Collaboration and Efficient Management Tool," emerges as a progressive and transformative solution, poised to redefine the way of student management.

1.2 OBJECTIVE

The main Objective of the project is

To replace traditional, manual project management methods (paper-based and spreadsheet-driven) with an innovative web application, thus ushering in a digital transformation in the management of student projects.

To streamline and enhance the efficiency of project management processes by enabling students to easily update and maintain their project details in real-time, reducing the administrative burden associated with paperwork and manual data entry.

To foster a collaborative environment by providing a digital platform that allows project teams to communicate and share ideas seamlessly, leading to more dynamic and interactive project experiences.

To centralize project-related data, including individual project dashboards, task management tools, and a document repository, ensuring that all stakeholders have access to up-to-date project information and resources.

CHAPTER 2

LITERATURE SURVEY

Project excellence or failure? Doing is the best kind of learning." It was written by Robert Bierwolf, VP Conferences at TEMS, and published in the IEEE Xplore digital library in 2013.In the paper, Bierwolf argues that the best way to learn about project management is by doing projects. He cites the Standish Group's CHAOS Report, which has found that a significant percentage of projects fail to meet their goals on time, within budget, and to the required quality standards. However, Bierwolf believes that even failed projects can provide valuable learning opportunities .He identifies a number of factors that contribute to project success and failure, including:

- Clear and well-defined project goals and objectives
- A well-developed project plan
- Effective communication and collaboration between team members and stakeholders

Student's Progress Tracking on Programming Assignments" by Grzegorz Polok and Mariusz Nowostawski presents a system for tracking students' progress on programming assignments. The system is based on a free repository management system called Gitlab, and it stores increments in the source code and presents characteristics from the gathered data in a graphical form.The system tracks a number of different metrics, including:

- Commit and code activity
- Workload
- Aggregated frequencies of code additions for the whole group of students

The system is still in development, but the authors have already conducted some preliminary testing with a small group of students. The results were positive, and the authors believe that the system has the potential to be a valuable tool for teaching and learning programming. The IEEE paper "**A Holistic Approach to Developing a Progress Tracking System for Distributed Agile Teams**" by Sultan Abdulaziz Alyahya proposes a holistic approach to developing a progress tracking system for distributed agile teams. The approach is based on the following key principles:

- Source code versioning
- Unit testing
- Acceptance testing
- Continuous integration and releasing

The paper argues that current progress tracking systems do not adequately take into account the impact of these technical factors. This can lead to inaccurate estimates of progress and missed deadlines.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

The prevailing system for managing student projects in academic institutions relied heavily on manual processes, utilizing paper-based documentation and Excel spreadsheets. Under this traditional approach, students submitted physical project reports and documents, leading to challenges in document storage, retrieval, and sharing. Excel spreadsheets were commonly used for tracking project progress and managing tasks, but they lacked the real-time collaboration features required for effective project management. The manual nature of data entry made the system susceptible to errors and data discrepancies. Moreover, limited collaboration and communication among project team members and faculty members were common, hindering the efficient progress of projects and imposing significant administrative overhead. The shortcomings of this existing system emphasized the need for the development of a more modern solution, the "Streamlined Project Collaboration and Efficient Management Tool." This innovative web application digitalizes project management, offering a user-friendly platform that centralizes data, streamlines project oversight, enhances collaboration, and significantly reduces administrative burdens. It aims to provide students and staff with an efficient and dynamic project management tool that fosters a more engaging and productive learning experience.

3.2 PROPOSED SYSTEM:

The "Streamlined Project Collaboration and Efficient Management Tool" project introduces a revolutionary digital solution to modernize the management of student projects in academic institutions. In contrast to the current manual system, which heavily relies on paper-based documentation and Excel spreadsheets, the proposed system offers a comprehensive web application designed to streamline and enhance project management.

In this proposed system, students will have access to a user-friendly digital platform that empowers them to seamlessly update and maintain their project details in real-time, eliminating the need for traditional paperwork and manual data entry. The web application will provide individual project dashboards, task management tools, progress tracking capabilities, and a centralized document repository, ensuring that project-related data is readily accessible and consistently up-to-date. This system will not only facilitate smoother collaboration among students but also equip staff with the tools needed to efficiently manage project teams, allocate resources, and monitor project timelines. Additionally, the platform will create collaborative spaces for teams to communicate and share ideas, fostering a dynamic and interactive project environment. By digitizing and centralizing project data, the proposed system aims to enhance the efficiency and effectiveness of project management within educational institutions, ultimately leading to improved project outcomes and a more enriching involved.

3.3 PROPOSED SOLUTION:

The proposed solution for "Streamlined Project Collaboration and Efficient Management Tool" is a comprehensive web application designed to revolutionize project management for students and faculty. This digital platform will empower students to seamlessly update and manage their project details in real-time, eliminating the inefficiencies associated with manual methods. Through a user-friendly interface, students can input progress reports, milestones achieved, and other relevant project information. The system will also feature robust collaboration tools, allowing team members to communicate effectively, share resources, and coordinate efforts effortlessly.

Real-Time Updates:

Enable students to update project details in real-time, such as progress reports ,task completions, and challenges faced .Ensure instant visibility of updates for staff to stay informed about project developments.

Collaboration Tools:

Integrate collaborative features, including file sharing, discussion forums, and task assignment, to enhance teamwork among students. Foster effective communication within teams and streamline collaborative efforts through the web application.

Task Management and Timelines:

Incorporate a task management system that allows students to outline tasks, set deadlines, and track their progress .Provide staff with the ability to view and manage project timelines, ensuring alignment with project objectives.

Resource Allocation and Monitoring:

Enable staff to allocate resources efficiently by tracking project budgets, equipment, and personnel assignments. Implement monitoring tools for staff to identify potential issues, bottlenecks, and resource gaps, allowing for proactive.

3.4 IDEATION & BRAINSTORMING:

In envisioning the "Streamlined Project Collaboration and Efficient Management Tool," the objective is to create a dynamic and user-centric web application that fundamentally transforms the traditional, manual project management approach employed by students. Picture a digital ecosystem where students and staff seamlessly interact with project data, fostering a collaborative and efficient environment .Interactive Dashboards: Develop visually engaging dashboards for students, offering an instant snapshot of project milestones, progress, and pending tasks. Likewise, equip staff with an overarching dashboard to oversee multiple projects, promoting proactive management.

User-Friendly Interface: Prioritize an intuitive and accessible interface, ensuring that students can effortlessly update their project details while staff can efficiently navigate and oversee team activities.

Real-Time Collaboration: Envision a platform where students can collaborate seamlessly in real-time. Picture shared document editing, synchronized calendars, and interactive project forums, fostering a sense of teamwork and shared progress.

Task Management: Integrate a robust task management system that allows students to outline, assign, and track tasks. This feature empowers staff to have granular insights into project timelines and individual contributions.

Mobile Accessibility: Envisage a tool that transcends physical boundaries, allowing students and staff to engage with the project anytime.

3.5 PROBLEM SOLUTION FIT:

The Problem-Solution Fit simply means that The "Streamlined Project Collaboration and Efficient Management Tool" addresses a pervasive issue in traditional project management within educational settings, where manual methods, such as paper-based tracking and Excel sheets, often lead to inefficiencies and lack of real-time collaboration. The identified problem lies in the cumbersome nature of these manual processes, which hinders seamless communication, timely updates, and effective project oversight. By digitizing project management through a web application, this initiative offers a solution fit for streamlining workflows and enhancing efficiency. Students can now effortlessly update project details, ensuring real-time collaboration and reducing the risk of information lag. Simultaneously, staff can manage teams more effectively, monitoring progress through a centralized and accessible platform. This solution not only mitigates the limitations of manual methods but also introduces a transformative approach that fosters a collaborative and dynamic project management environment, ultimately enhancing the overall efficacy of student projects. Problem: Manual Inefficiencies

- **Communication Gaps:** Manual methods often result in communication delays and gaps, as updates on project progress and changes may not be instantly accessible to all team members and supervisors.

- **Version Control Issues**: Paper-based and Excel-centric systems struggling with version control, leading to the possibility of outdated information circulating among team members and supervisors.

- **Limited Collaboration:** Traditional methods hinder real-time collaboration among students. Collaborative efforts may be impeded by physical constraints, making it difficult for team members to work synchronously.

- **Delayed Oversight:** Supervisors and staff face challenges monitoring project progress in real-time. This delay in oversight can lead to difficulties.

- **Real-Time Updates:** The web application facilitates instant updates from students, allowing for real-time visibility into project details. This ensures that all team members and supervisors have access to the most recent information.

- **Version Control and Audit Trail:** The digital platform incorporates version control mechanisms, maintaining a clear audit trail of all changes made to project documents. This ensures data integrity and provides a historical record of project evolution.

- **Enhanced Collaboration Tools:** The tool integrates collaborative features such as shared document editing, discussion forums, and task assignment. This fosters a dynamic and interactive environment, enabling students to work together seamlessly.

- **Anywhere, Anytime Access:** With mobile optimization, the web application breaks down physical barriers, allowing students and staff to access and contribute to project details from anywhere, at any time. This flexibility enhances collaboration and responsiveness.

- **Centralized Project Dashboards:** Individual project dashboards offer a consolidated view of project milestones, timelines, and progress.

3.6 ARCHITECTURE DESIGN:

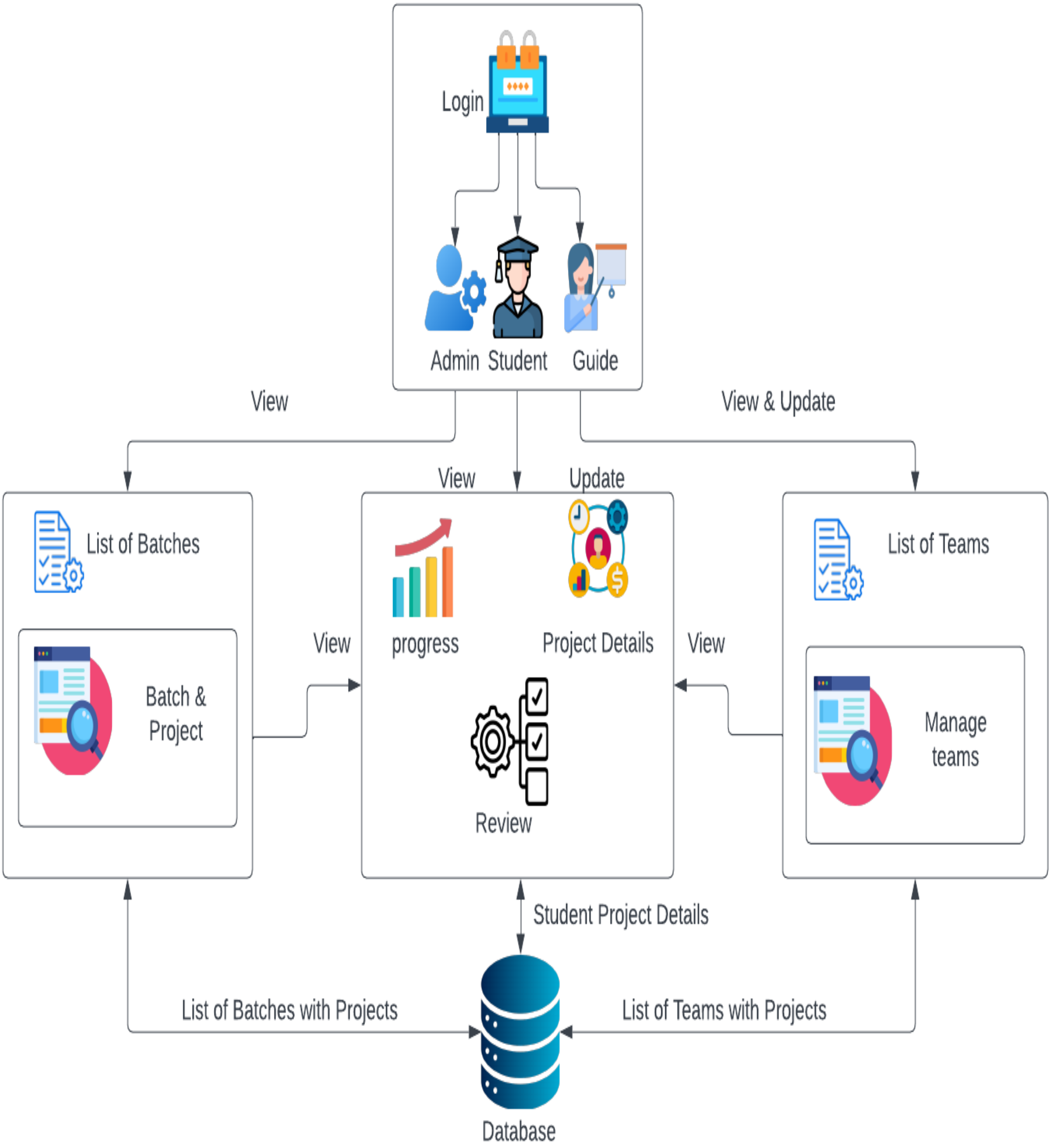


Figure 3.1: Model Architecture

In the proposed system, students engage in a structured workflow that optimizes their project management experience. Upon logging in, students are prompted to select their academic batch, initiating a personalized journey. A list of teams affiliated with the chosen batch is then presented, allowing students to seamlessly identify and choose their specific team. Once the team selection is made, students gain access to a comprehensive view of their team's project details, encompassing key information such as project objectives, timelines, and assigned tasks. This user-centric approach empowers students to actively contribute to the project's success by providing a platform where they can update their project progress efficiently and in real-time.

Guides, as pivotal facilitators in the academic project landscape, experience a tailored interface that aligns with their supervisory responsibilities. After logging in, guides are directed to a list of teams under their purview. By selecting a specific team from this list, guides gain insight into the project details of the chosen team. This includes a holistic overview of the project's trajectory and milestones. Guides, crucially, possess the capability to update the overall project progress for the entire team, streamlining communication and coordination. Additionally, they can delve into individual student progress within the team, fostering a nuanced understanding of each student's contributions. Administrators, wielding a comprehensive view of the entire academic project landscape, engage with the system with a broader perspective.

Upon logging in, administrators are presented with a comprehensive list encompassing all teams across batches. By selecting a specific team, administrators gain access to detailed project information, akin to both students and guides. This advanced access extends to the capacity to update the team's project progress and execute managerial actions such as adding or removing students from the team. The administrator's role also includes the ability to scrutinize and assess the individual progress of students, ensuring a nuanced and informed perspective on the overall project dynamics.

3.7 DESCRIPTION OF MODULES:

The system you're describing seems to be a comprehensive project management tool for academic settings, with different levels of access for students, guides, and administrators. Let's elaborate on each user's interactions with the system:

3.7.1 STUDENT INTERACTION:

Login and Batch Selection:

Upon logging into the system, students are presented with an interface where they can choose their academic batch.

Viewing Team List:

After selecting the batch, students see a list of teams associated with that batch. This list likely includes the names or identifiers of each team.

Viewing Team's Project Details:

Once the team is selected, students can access detailed information about their team's project. This might include project goals, timelines, and assigned tasks.

Updating Project Progress:

Students have the ability to update the project's progress. This could involve marking completed tasks, providing status updates, or uploading relevant documents.

In the envisioned system, students engage in a streamlined and user-centric project management process that begins with their login, prompting them to select the academic batch associated with their project. Following this initial step, students are presented with a clear list of teams within the chosen batch.

Within this list, they effortlessly identify and select their specific team, initiating a personalized project journey. Once their team is chosen, students gain immediate access to a detailed dashboard presenting their team's project particulars. This comprehensive view encompasses essential project elements, including objectives, timelines, and assigned tasks. Notably, the system empowers students by providing a straightforward interface through which they can readily update and communicate their individual and team-wide project progress. This user-friendly and real-time interaction not only enhances project collaboration but also ensures that students actively contribute to and stay informed about the evolving dynamics of their academic projects. In the student-centric realm of the academic project management system, students embark on a structured and intuitive journey that commences with their login. Upon entering the system, students are prompted to select their academic batch, a pivotal step that tailors the system's functionalities to the specific context of their projects. This personalized initiation ensures that students seamlessly transition into a dedicated space that aligns with their academic endeavors. Following the batch selection, students are presented with a clear and organized list of teams associated with that specific batch. Within this list, students effortlessly identify and select their designated team, marking the inception of their individual project journey. This streamlined team selection process serves as a gateway to a comprehensive view of the team's project details, fostering a sense of ownership and collaboration. Once a team is chosen, students are granted access to a detailed dashboard that encapsulates the intricacies of their project. This comprehensive view includes essential project elements such as defined goals, project timelines, and assigned tasks. The system facilitates real-time updates, enabling students to effortlessly contribute to and communicate their project progress, thereby fostering an environment of collaboration and transparency. Crucially, the system empowers students by providing them with the autonomy to update and manage their individual and team-wide project progress. This user-friendly interface facilitates the seamless submission of progress reports, task completions, and other relevant project details. This real-time interaction not only enhances students' engagement with their projects but also ensures that they actively contribute to the ongoing success of their academic endeavors. In summary, the student's interaction within the system is characterized by a user-centric and empowering approach. From batch selection to team engagement and individual progress updates, the system facilitates a dynamic and collaborative environment that aligns with the unique needs and contributions of students in the academic project management landscape.

3.7.2 GUIDE INTERACTION:

Login and Team Supervision:

Guides log into the system and, similar to students, select the academic batch they are supervising.

Viewing Team List for Supervision:

Guides see a list of teams under their supervision, likely with the names of the teams them guide.

Selecting Team:

Guides choose a specific team from their list, indicating which group they want to manage.

Viewing Team's Project Details:

After selecting the team, guides can access detailed project information, just like the students. This provides them with insights into the team's progress and goals.

Updating Team's Project Progress or Individual Students:

Guides have the authority to update the overall project progress for the team. Additionally, they can view and possibly update the progress of individual students within the team. Guides, as integral facilitators within the academic project framework, initiate their interaction with the system by logging in and selecting the academic batch they are entrusted to supervise, mirroring the initial steps taken by students. This marks the commencement of a user-specific journey that tailors the system's interface to cater to the unique responsibilities of guides. Upon this selection, guides are presented with a comprehensive list featuring the teams under their supervision, each likely labeled with the names of the teams they guide. This intuitive and organized display streamlines the process for guides to identify and choose a specific team, providing them with a structured overview of their mentoring responsibilities. Having chosen a team from their dedicated list, guides seamlessly transition into a detailed exploration of the team's project particulars. This mirrors the experience of students, offering guides insights into the team's progress, defined goals, and ongoing tasks. This comprehensive view equips guides with the information necessary for effective mentoring and strategic guidance. Guides, crucially, are bestowed with the authority to actively participate in the project's development. They can update the overall project progress for the entire team, ensuring alignment with academic objectives and facilitating cohesive teamwork. Beyond team-level interactions, guides possess a nuanced capability to zoom into individual student progress within the team. This multifaceted authority allows guides to not only oversee the collective advancement of the team but also to delve into the contributions of individual students. This granular insight enables guides to provide tailored support, interventions, or guidance as needed, fostering a personalized approach to academic mentorship. In essence, the guide's interaction within the system is characterized by a seamless integration of team oversight, strategic guidance, and personalized mentorship tailored to both team and individual dynamics. Guides engage with the system in a multi-faceted manner that aligns with their role as mentors and supervisors in the academic project setting. As they log into the system, guides navigate through a user-friendly interface akin to students, initially selecting the academic batch under their supervision. This foundational step ensures that the system tailors its functionalities to the specific academic context under the guide's purview, streamlining subsequent interactions. Once the academic batch is chosen, guides are presented with a comprehensive list of teams falling under their mentorship umbrella. This list serves as a centralized repository, likely populated with the names or identifiers of the teams they guide. The organized presentation facilitates a quick and efficient selection process for guides, allowing them to seamlessly choose a particular team they wish to oversee. Upon selecting a specific team from their list, guides gain access to a detailed dashboard providing a panoramic view of the team's project details. This comprehensive insight includes critical information such as project milestones, timelines, and assigned tasks. Mirroring the experiences of students within the same team, guides become privy to the intricacies of the project's progress and goals, fostering a deeper understanding of the team's dynamics. Crucially, guides wield significant authority within the system. They can actively contribute to the project's advancement by updating the overall project progress for the entire team. This feature not only aligns the team's efforts with academic objectives but also ensures that the project stays on course. Additionally, guides have the capability to view and, potentially, update the progress of individual students within the team. This dual-level interaction allows guides to assess both the collective and individual contributions to the project, enabling a more personalized approach to mentorship. In summary, the guide's interaction within the system is characterized by a seamless progression from batch selection to team oversight and individual student engagement. The system not only facilitates effective project management but also empowers guides to provide tailored mentorship, ultimately contributing to the success of both the academic projects and the students involved.

3.7.3 ADMIN INTERACTION:

Login and Team Overview:

Administrators log into the system and have a broader view. They might not be restricted to a specific batch and could potentially see all teams across different batches.

Selecting a Team:

Administrators choose a specific team they wish to inspect or manage. This could be based on various criteria like urgency, performance, or specific requests.

Viewing Team's Project Details:

Similar to guides and students, administrators can access detailed information about the selected team's project. This includes progress, goals, and tasks.

Updating Team's Project Progress:

Administrators have the authority to update the overall project progress for the selected team. They might also have additional privileges, such as adding or removing students from the team.

Managing Individual Students:

In addition to team-level interactions, administrators can view the progress of individual students within a team. This feature provides a granular understanding of each student's contribution to the project.

Administrators, upon logging into the system, are granted an expansive and panoramic view, unbounded by the confines of specific academic batches. This elevated perspective allows them to traverse seamlessly across different batches, ensuring a comprehensive understanding of the entire academic project landscape. Within this encompassing purview, administrators undertake the crucial task of selecting a particular team for closer inspection or management. This selection process is driven by a range of criteria, be it the urgency of a situation, the team's performance metrics, or specific requests necessitating their attention. Once a team is chosen, administrators delve into a detailed exploration of the team's project particulars. This parallels the experiences of both guides and students, providing administrators with a comprehensive overview of the project's progress, defined goals, and ongoing tasks.

As key orchestrators of the project landscape, administrators possess the authority to influence and steer the trajectory of the selected team. This extends to the pivotal capability of updating the overall project progress, ensuring alignment with overarching academic objectives. Furthermore, administrators may wield additional privileges, such as the ability to dynamically modify team compositions by adding or removing students. This level of administrative control is integral to adapting to evolving project dynamics and optimizing team resources. In addition to team-level interactions, administrators have the invaluable ability to zoom into the granular details of individual students within a team. This feature provides a nuanced understanding of each student's contributions to the project, facilitating personalized guidance and intervention as needed. In essence, the administrator's interaction within the system is characterized by a sophisticated orchestration of project oversight, strategic decision-making, and dynamic management of both team and individual-level project dynamics. Administrators navigate the academic project management system with a comprehensive view and elevated privileges. Upon logging in, administrators are presented with a panoramic overview that transcends the constraints of specific academic batches, granting them a broad perspective across different batches. This unrestricted access ensures administrators can seamlessly traverse the entire academic project landscape, positioning them as orchestrators capable of overseeing diverse teams and projects. With this expansive vantage point, administrators embark on the critical task of selecting a specific team for closer inspection or management. The selection process is guided by various criteria, including urgency, performance metrics, or specific requests that necessitate their attention. This strategic selection mechanism enables administrators to focus on teams that require immediate intervention or align with overarching organizational priorities. Once a team is chosen, administrators gain access to a detailed repository of the team's project details, mirroring the experiences of both guides and students. This comprehensive view encompasses vital project elements, including progress updates, defined goals, and ongoing tasks. The intricate insights obtained empower administrators with the information necessary for informed decision-making, strategic planning, and proactive management. Administrators, serving as pivotal influencers within the system, possess the authority to actively shape the trajectory of the selected team.

This extends to the critical capability of updating the overall project progress, ensuring alignment with overarching academic objectives. Furthermore, administrators may wield additional privileges, such as the ability to dynamically modify team compositions by adding or removing students. This administrative control is integral to adapting to evolving project dynamics, optimizing team resources, and aligning the project with institutional goals. In addition to team-level interactions, administrators have the invaluable ability to scrutinize and assess the individual progress of students within a team. This granular perspective allows administrators to gain insights into the unique contributions of each student, fostering a nuanced understanding of individual student dynamics within the broader project context. In summary, the administrator's interaction within the system is characterized by a sophisticated orchestration of project oversight, strategic decision-making, and dynamic management of both team and individual-level project dynamics. The system not only facilitates efficient project collaboration but also empowers administrators to enact proactive and informed interventions, ensuring the success and alignment of academic projects with institutional objectives. In this way, the system offers a hierarchical structure, providing different levels of access and control based on the user's role (student, guide, or administrator). Each user can interact with the system according to their responsibilities and requirements within the academic project management framework.

3.7.4 LOGIN MODULE:

In the realm of user authentication, the academic project management system incorporates a robust Role-Based Access Control (RBAC) mechanism. Once users are successfully authenticated, the system dynamically assigns roles, designating them as students, guides, or administrators. This pivotal information forms the foundation for providing users with precise and tailored permissions, ensuring that they interact with the system, with their roles not only enhances the security posture of the system but also streamlines user experiences by presenting them with functionalities to their designated roles. The login module, as the initial point of interaction, prioritizes user experience through a meticulously designed and user-friendly interface. By employing clear instructions and intuitive visual cues, the system guides users seamlessly through the login process. This emphasis on usability enhances the overall experience, reducing friction and ensuring that users, regardless of their technical proficiency, can navigate the login module with ease.

In the event of forgotten passwords, the system incorporates a robust "Forgot Password" mechanism. This feature facilitates account recovery by sending a password reset link to the user's registered email. Additionally, users can leverage account recovery options such as security questions or verification codes, adding layers of security while providing a streamlined process for reclaiming access. To fortify against unauthorized access attempts, the system implements an Account Lockout Mechanism. After a predetermined number of failed login attempts, the account is temporarily locked, acting as a deterrent to potential security threats. This mechanism enhances the overall security of the system by mitigating the risk of brute-force attacks and unauthorized access. Session Management is a critical aspect of the login module, where the system securely manages user sessions. This involves the generation of unique session tokens and the establishment of session timeout limits, safeguarding against session hijacking and unauthorized access during idle periods.

For scalability and seamless user experiences, the system optionally integrates with Identity Providers such as Google, Microsoft, or LDAP for Single Sign-On (SSO) capabilities. This streamlines the authentication process and provides users with a unified and simplified access experience. Customization and Branding further contribute to a tailored user experience. The login module can be customized to reflect the unique branding of the educational institution. This includes incorporating logos, choosing color schemes, and integrating specific messaging relevant to the organization, fostering a sense of institutional identity and familiarity. Lastly, the system ensures Compliance with Data Protection Regulations. Depending on the jurisdiction and nature of the data being handled, the login module aligns with regulations such as GDPR or HIPAA. This commitment to compliance underscores the system's dedication to safeguarding user data and privacy, instilling trust among users and administrators alike. Collectively, these features integrated into the login module create a secure, user-friendly, and compliant access mechanism for all users within the academic project management system.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

Component	Minimum Requirement
Server	64-bit, four-core, 2.5 GHz minimum per core
CPU and RAM	8 GB for developer or evaluation use 16 GB for production use
Hard disk	20 GB

4.2 SOFTWARE REQUIREMENTS:

- React
- Python IDE
- Django
- VS Code Editor
- SQL Database

CHAPTER 5

IMPLEMENTATION

The "Streamlined Project Collaboration and Efficient Management Tool" represents a transformative solution for students, guides, and administrators involved in academic projects. Traditionally, project monitoring has relied on manual methods such as paper-based documentation and Excel spreadsheets. However, this project introduces a digitalized approach through a web application, revolutionizing how project details are updated, managed, and monitored.

In this innovative system, students experience a streamlined project management process. Upon logging in, they select the academic batch associated with their project, initiating a personalized journey. The system presents a clear list of teams within the chosen batch, allowing students to effortlessly identify and select their specific team. Once the team is selected, students gain access to a detailed dashboard presenting their team's project details, including goals, timelines, and assigned tasks. This user-centric approach empowers students to update their project progress seamlessly, promoting real-time collaboration and transparency.

Guides, as key facilitators, benefit from a tailored interface upon login. They view a list of teams under their supervision, simplifying the process of team selection. After choosing a specific team, guides access a detailed overview of the team's project details. This includes insights into the team's progress, goals, and individual student contributions. Guides, crucially, have the authority to update the team's project progress and view the progress of individual students, fostering effective mentorship and project guidance.

Administrators log in to the system with a broader perspective, viewing a comprehensive list of all teams across batches. Upon selecting a team, administrators gain detailed insights into the team's project specifics. Their role extends to updating the team's project progress, managing team compositions by adding or removing students, and scrutinizing individual student progress. This elevated level of administrative control ensures adaptability to evolving project dynamics and optimal resource utilization.

The system incorporates a Role-Based Access Control (RBAC) mechanism to determine user roles and permissions, securing sensitive information. The login module features a user-friendly interface with clear instructions and visual cues, enhancing the overall user experience. A robust "Forgot Password" mechanism and account lockout feature fortify security, while session management protects against session hijacking.

Logging and auditing functionalities track login activities, aiding in monitoring user access and identifying potential security threats. Optionally, integration with identity providers for Single Sign-On (SSO) capabilities enhances scalability and ease of use.

STUDENT MODULE:

Students initiate their interaction with the academic project management system by logging in with their unique credentials. Upon successful login, they are prompted to select the academic batch associated with their project, tailoring the system to their specific context. A list of teams within the chosen batch is presented, allowing students to effortlessly identify and select their respective teams. Once the team is chosen, students gain access to a comprehensive project dashboard displaying critical details such as project goals, timelines, and assigned tasks. This intuitive interface empowers students to seamlessly update their project progress in real-time, fostering a collaborative and transparent project management environment.

In the student module, the user journey begins with the login process, where students use their unique credentials to access the academic project management system. Upon successful login, students are prompted to select the academic batch associated with their ongoing project. This initial step ensures a personalized and context-specific experience. Subsequently, a list of teams within the chosen batch is presented to students, allowing them to easily identify and select their specific project team. Once the team is selected, students gain access to a comprehensive project dashboard. This dashboard serves as a centralized hub for project-related information, displaying crucial details such as project goals, timelines, and assigned tasks. One of the key features of this module is the seamless ability for students to update their project progress directly from the dashboard. This real-time collaboration fosters transparency among team members and enhances overall project management efficiency.

GUIDE MODULE:

Guides, as key facilitators within the academic project landscape, log into the system with credentials tailored to their role. They select the academic batch they supervise, initiating a personalized view that streamlines their supervisory responsibilities. A clear list of teams under their guidance is presented, each labeled with team names for easy identification. Guides choose a specific team, gaining access to a detailed overview of the team's project. This comprehensive view includes insights into the team's progress, goals, and individual student contributions. Guides hold the authority to update the team's project progress and, importantly, can view and potentially update the progress of individual students within the team, ensuring effective mentorship and project guidance.

Guides, in their role as facilitators and supervisors, interact with the system through a dedicated guide module. After logging in with their credentials, guides select the academic batch they are responsible for supervising, initiating a tailored view. This view includes a clear and organized list of teams under their guidance, each team labeled for easy identification. Guides can then choose a specific team from their list, gaining access to a detailed overview of the team's project. This detailed overview encompasses various aspects, including the team's progress, defined goals, and individual contributions of team members. A significant aspect of the guide module is the authority granted to guides. They can actively contribute to the project's advancement by updating the overall project progress for the entire team. Moreover, guides can delve into the progress of individual students within the team, allowing for personalized mentorship and guidance based on individual contributions.

ADMINISTRATOR MODULE:

Administrators, possessing a broader purview, log into the system with extensive access across all academic batches. They are presented with a comprehensive list of all teams, providing a global perspective on the project landscape. Administrators can then selectively choose a specific team for in-depth inspection or management. After team selection, administrators gain access to detailed project information, including progress, goals, and tasks, enabling effective oversight. The system affords administrators the authority to update the team's project progress, with additional privileges allowing them to add or remove students from the team. Moreover, administrators can scrutinize and assess individual student progress, providing a granular understanding of each student's contribution to the project. This modular approach ensures that each user category experiences a tailored and focused interaction within the academic project management system, enhancing overall efficiency and collaboration.

Administrators, with their comprehensive role, access the academic project management system through the administrator module. Their login provides them with a panoramic view, extending beyond specific academic batches to encompass all teams across different batches. This broader perspective allows administrators to effectively oversee and manage the entire project landscape. Within the module, administrators can selectively choose a specific team for detailed inspection or management. Once a team is selected, administrators gain access to a wealth of information, including project progress, goals, and tasks. An integral aspect of the administrator module is the extensive authority it confers upon administrators. They can actively shape the trajectory of the selected team by updating the overall project progress. Additionally, administrators have the flexibility to manage team compositions.

5.1 SOFTWARE DESCRIPTION:

This is a robust web application designed to revolutionize project collaboration and management within academic environments. By seamlessly integrating cutting-edge technologies such as React, Django, and SQL databases, it transcends traditional manual project monitoring methods, offering an efficient, centralized, and user-friendly solution for students, guides, and administrators.

React-powered User Interface:

This boasts a dynamic and responsive user interface developed using React. This ensures an intuitive and engaging experience for users, facilitating seamless navigation and real-time updates .The React-powered user interface is carefully crafted to provide users with an intuitive and engaging experience. Leveraging React's component-based architecture, StreamlinePro ensures a modular and maintainable design, allowing for the creation of reusable UI components. This not only enhances development efficiency but also contributes to a consistent and visually appealing interface. The dynamism of the user interface is particularly notable in its responsiveness to user actions. React enables StreamlinePro to deliver near-instantaneous updates in response to user interactions, creating a seamless and interactive environment. Whether users are navigating through project details, updating progress, or collaborating in real-time, the React-powered interface ensures a smooth and engaging experience.

The React-powered interface also supports a modular structure, allowing developers to break down complex UI elements into smaller, manageable components. This not only simplifies the development process but also enhances maintainability and extensibility. Each component within the interface is designed to fulfill a specific function, promoting code reusability and making it easier to adapt to future changes and feature additions.

Python IDE Integration:

The project leverages a Python Integrated Development Environment (IDE) to enhance the development and coding experience. This integration streamlines the coding process, making it efficient and user-friendly for developers contributing to the project's continuous improvement StreamlinePro adopts a strategic approach by integrating a Python Integrated Development Environment (IDE) to elevate the overall development and coding experience. This integration serves as a cornerstone in the project's architecture, fostering an environment that is not only efficient but also user-friendly for the developers actively contributing to the continuous improvement of the application.

Efficiency in Coding:

The Python IDE integration in StreamlinePro introduces a host of features geared towards enhancing coding efficiency. Intelligent code completion is a standout feature, providing developers with context-aware suggestions as they type, significantly accelerating the coding process. This ensures that developers can write accurate and syntactically correct code more quickly.

Django Backend Framework:

The backend of StreamlinePro is powered by Django, a high-level Python web framework known for its versatility and scalability. Django facilitates secure data handling, robust API development, and efficient integration with the React front-end, ensuring a cohesive and well-structured application.

Versatile Architecture:

Django's versatility shines through its Model-View-Controller (MVC) architectural pattern, referred to as Model-View-Template (MVT) in Django terminology. This structure compartmentalizes the application logic, making it modular and easy to maintain. Each component, from models to views and templates, serves a distinct purpose, promoting a clean and scalable architecture for StreamlinePro.

Scalability and Flexibility:

Django's scalability is a testament to its ability to handle projects of varying sizes. Whether StreamlinePro experiences growth in terms of users, features, or data volume, the Django backend provides a foundation that can be easily scaled to meet evolving demands.

Secure Data Handling:

Security is a paramount concern in any application, particularly one dealing with academic project management. Django incorporates security best practices by default. Features such as an Object-Relational Mapping (ORM) system prevent common security vulnerabilities, ensuring that data is handled and stored securely. Protection against SQL injection and a robust authentication system contribute to a secure data environment.

VS Code Editor Integration:

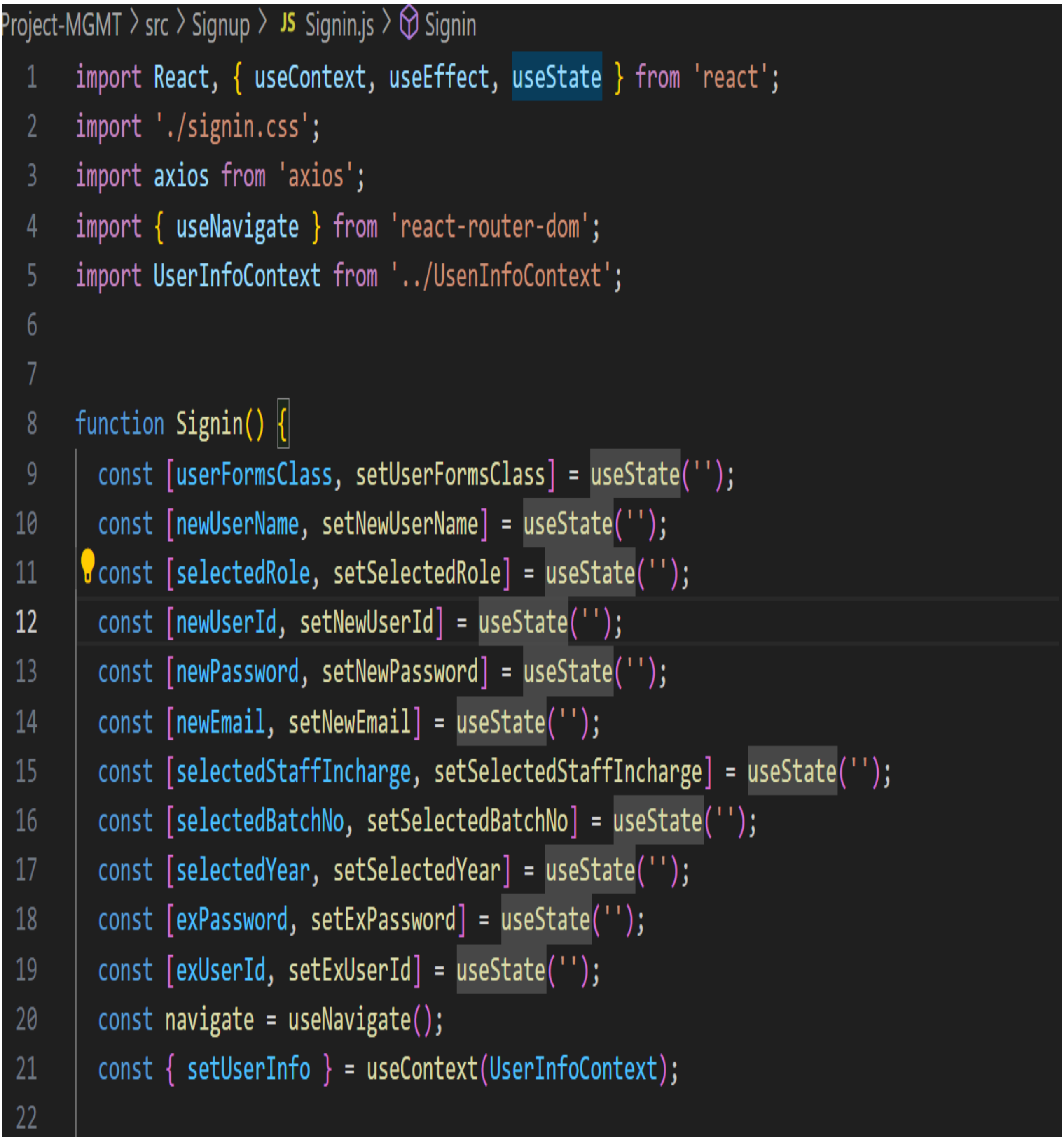
The project supports the Visual Studio (VS) Code editor, providing developers with a powerful and feature-rich environment for coding and collaboration. The integration of VS Code enhances the development workflow, promoting code quality and collaboration among team members.

SQL Database Management:

StreamlinePro utilizes an SQL database to efficiently store and manage project-related data. This relational database ensures data integrity, scalability, and seamless retrieval of information critical for effective project monitoring and management.

5.2 CODE IMPLEMENTATION:

Step 1: Login implementation



Project-MGMT > src > Signup > JS Signin.js > Signin

```javascript
import React, { useContext, useEffect, useState } from 'react';
import './signin.css';
import axios from 'axios';
import { useNavigate } from 'react-router-dom';
import UserInfoContext from '../UsenInfoContext';


function Signin() {
  const [userFormsClass, setUserFormsClass] = useState('');
  const [newUserName, setNewUserName] = useState('');
  const [selectedRole, setSelectedRole] = useState('');
  const [newUserId, setNewUserId] = useState('');
  const [newPassword, setNewPassword] = useState('');
  const [newEmail, setNewEmail] = useState('');
  const [selectedStaffIncharge, setSelectedStaffIncharge] = useState('');
  const [selectedBatchNo, setSelectedBatchNo] = useState('');
  const [selectedYear, setSelectedYear] = useState('');
  const [exPassword, setExPassword] = useState('');
  const [exUserId, setExUserId] = useState('');
  const navigate = useNavigate();
  const { setUserInfo } = useContext(UserInfoContext);

```

This React code defines a component named `Signin` that represents a user authentication and registration interface. It uses state hooks to manage form inputs, user authentication status, and user authorization roles. It also employs the `axios` library for making HTTP requests to a server.

1. State Hooks:

- `userFormsClass`: Manages the CSS class for animating the form transitions.
- `newUserName`, `newUserId`, `newPassword`, `newEmail`: Store user input for registration.
- `selectedRole`, `selectedStaffIncharge`, `selectedBatchNo`, `selectedYear`: Store selected options during registration.
- `exPassword`, `exUserId`: Store user input for login.
- `isauthorized`: Manages user authorization status (admin, student, staff).
- `isSigned`: Manages user login status.

2. Effects:

- The `useEffect` hook is used to load user information from local storage on component mount. If the user is already signed in, it sets the corresponding state variables (`isSigned`, `isauthorized`, `userInfo`) accordingly.

3. Login and Signup Functions:

- `handleSignupClick` and `handleLoginClick`: Toggle the animation class to switch between login and signup forms.
- `handleLogin`: Sends a POST request to the server with user login credentials. If successful, updates the state and local storage with user information.
- `handleSignup`: Sends a POST request to the server with user registration information. If successful, updates the state and local storage with user information.

4. Navigation Handling:

- The `handleNavigation` function redirects the user based on their authorization role after successful login.

5. Input Change Handlers:

- Functions like `handleExPassword`, `handleExUserId`, etc., handle changes in input fields and update the corresponding state.

6. Conditional Rendering:

- The component conditionally renders different sections based on the user's login/signup status and the selected form.

7. JSX Structure:

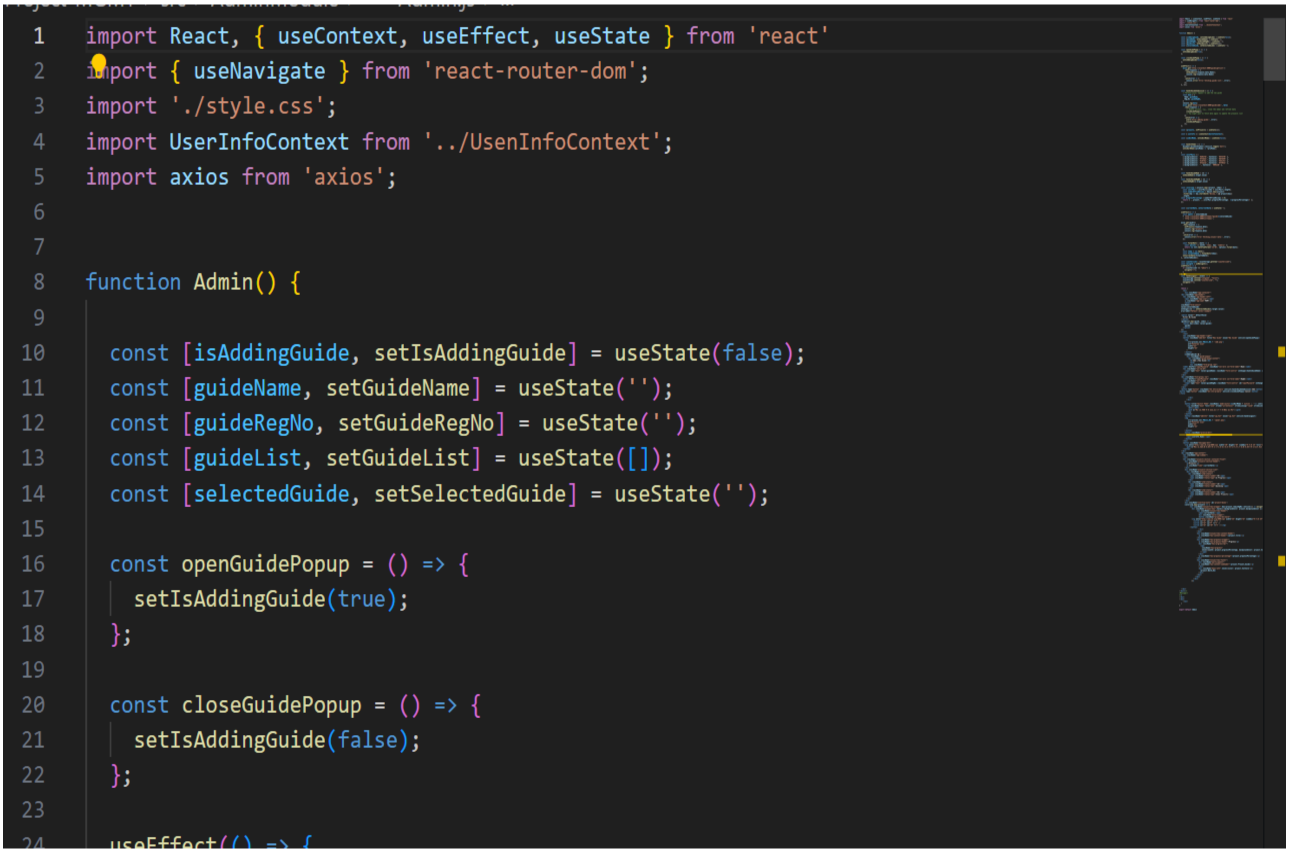
- The component's JSX structure is organized using `div` containers and form elements. It uses conditional rendering to display either the login or signup form based on the `userFormsClass`.

8. CSS Styles:

- The styles for the component are not provided in the code snippet, but the `user_forms` class and its animations (`bounceLeft` and `bounceRight`) suggest the usage of CSS for styling and animations.

This component appears to be part of a larger authentication and authorization system in a React application, connecting with a backend server for user authentication and registration.

Step 2: Admin implementation



```
import React, { useContext, useEffect, useState } from 'react'
import { useNavigate } from 'react-router-dom';
import './style.css';
import UserInfoContext from '../UsenInfoContext';
import axios from 'axios';


function Admin() {

  const [isAddingGuide, setIsAddingGuide] = useState(false);
  const [guideName, setGuideName] = useState('');
  const [guideRegNo, setGuideRegNo] = useState('');
  const [guideList, setGuideList] = useState([]);
  const [selectedGuide, setSelectedGuide] = useState('');

  const openGuidePopup = () => {
    setIsAddingGuide(true);
  };

  const closeGuidePopup = () => {
    setIsAddingGuide(false);
  };

  useEffect(() => {
```

Imports:

import React, { useContext, useEffect, useState } from 'react';: This line imports React and some of its hooks (useContext, useEffect, useState). These hooks are commonly used in React functional components to manage state and side effects.

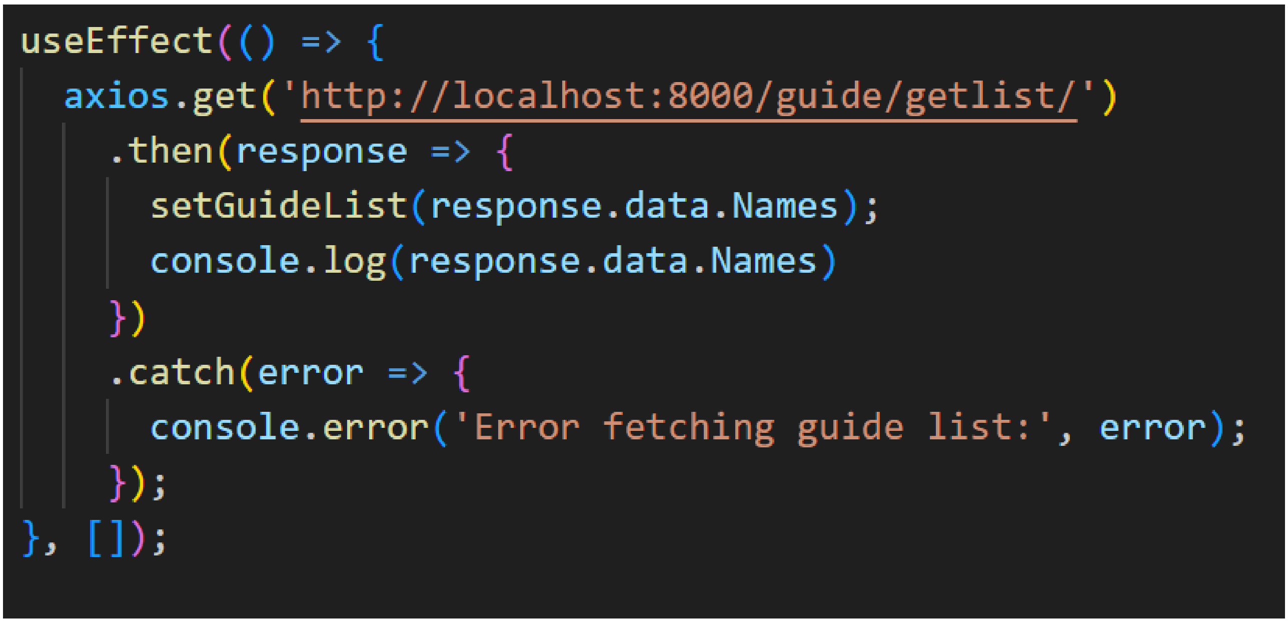
import { useNavigate } from 'react-router-dom';: This line imports the useNavigate hook from the 'react-router-dom' library. This hook is used for programmatic navigation in a React application.

import './style.css';: This line imports a CSS file named 'style.css'. This is likely used to style the component.

import UserInfoContext from '../UserInfoContext';: This line imports a context named UserInfoContext from a file located at '../UserInfoContext'. Contexts in React are used to share values like themes, user authentication status, etc., between components without explicitly passing them through props.

import axios from 'axios';: This line imports the Axios library, which is commonly used for making HTTP requests in JavaScript.

AXIOS IMPLEMENTAION FOR ADMIN PAGE:



```
useEffect(() => {
  axios.get('http://localhost:8000/guide/getlist/')
    .then(response => {
      setGuideList(response.data.Names);
      console.log(response.data.Names)
    })
    .catch(error => {
      console.error('Error fetching guide list:', error);
    });
}, []);
```

Axios POST Request:

axios.post: This method is from the Axios library and is used to make an HTTP POST request.

'http://localhost:8000/guide/add/': This is the URL to which the POST request is being made. It looks like it's adding a guide on a local development server at port 8000. data: This is the payload/data that will be sent along with the POST request. The actual content of data is not provided in the code snippet, but it's typically an object containing information relevant to the request.

Handling Success (.then block):

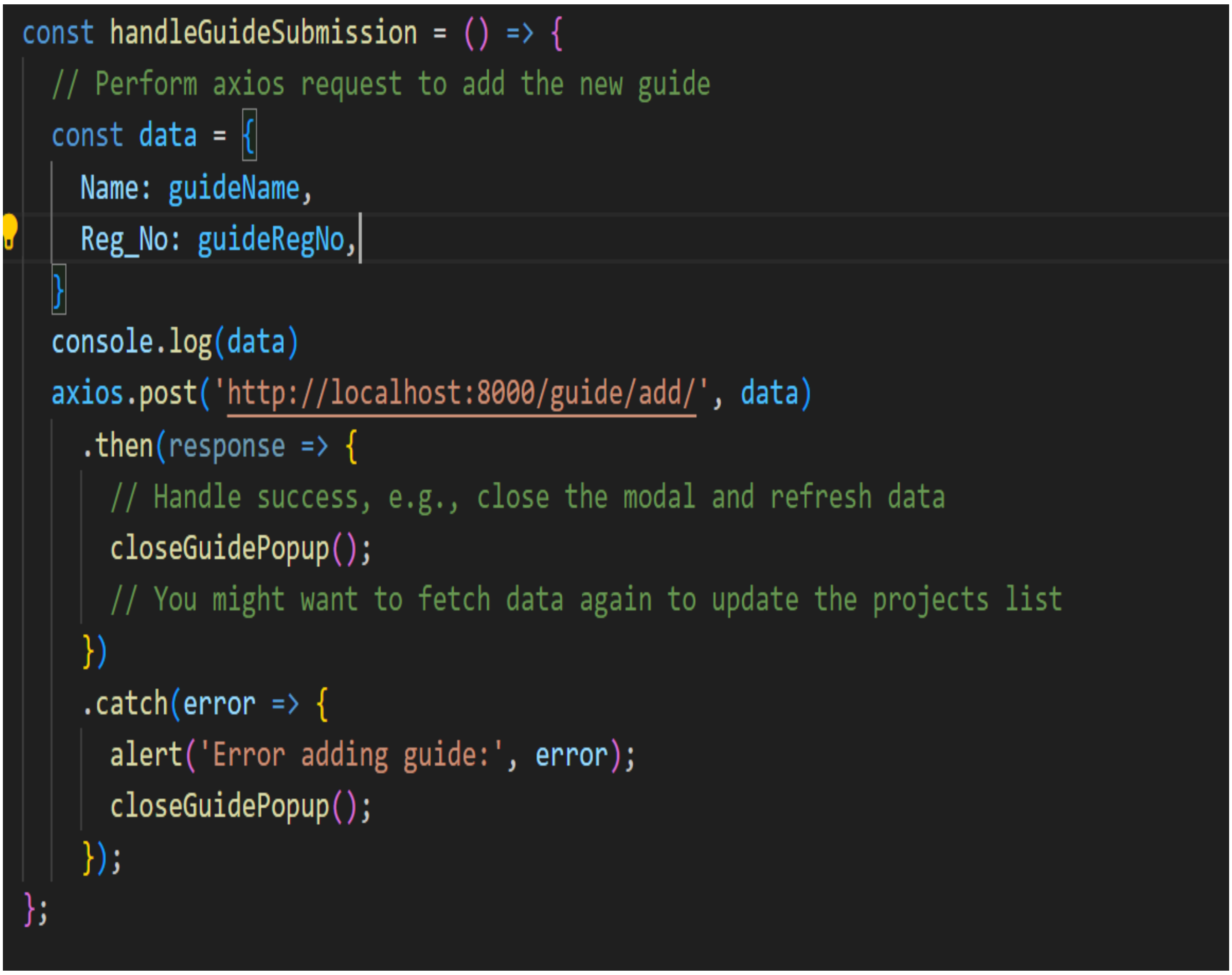
When the POST request is successful, the code inside the .then block is executed.

response: It represents the response received from the server after the successful POST request.

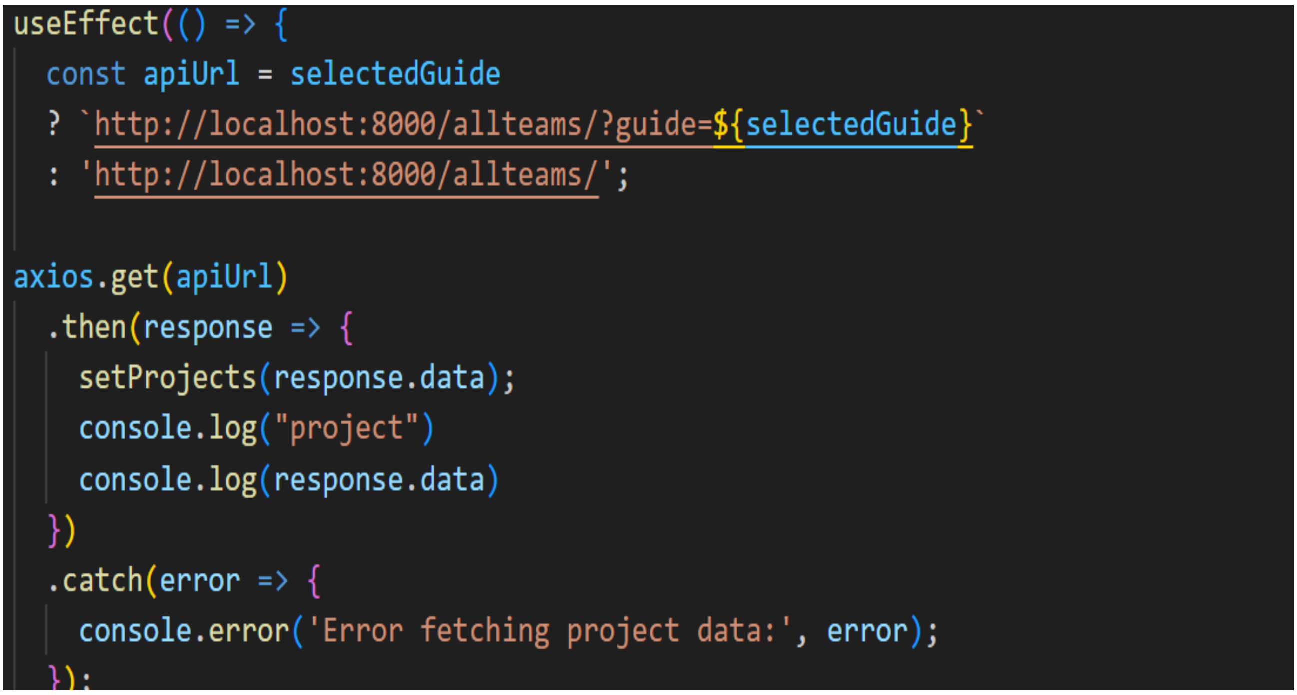
closeGuidePopup (): This is a function (not defined in the provided snippet) that is called, presumably to close a modal or some kind of UI component.A comment suggests that you might want to fetch data again to update the projects list. This would depend on the specific requirements of your application.

Step 3: Staff implementation

The provided code is a React functional component named `Staff` that seems to be part of a web application. This component incorporates various React hooks and Axios for making asynchronous requests. Let's break down the key functionalities and explore the underlying concepts:



```
const handleGuideSubmission = () => {
  // Perform axios request to add the new guide
  const data = {
    Name: guideName,
    Reg_No: guideRegNo,
  }
  console.log(data)
  axios.post('http://localhost:8000/guide/add/', data)
    .then(response => {
      // Handle success, e.g., close the modal and refresh data
      closeGuidePopup();
      // You might want to fetch data again to update the projects list
    })
    .catch(error => {
      alert('Error adding guide:', error);
      closeGuidePopup();
    });
};
```



```
useEffect(() => {
  const apiUrl = selectedGuide
  ? `http://localhost:8000/allteams/?guide=${selectedGuide}`
  : 'http://localhost:8000/allteams/';

axios.get(apiUrl)
  .then(response => {
    setProjects(response.data);
    console.log("project")
    console.log(response.data)
  })
  .catch(error => {
    console.error('Error fetching project data:', error);
  });
```

1. State Management:

The component uses the `useState` hook to manage different pieces of state. These include `isAddingGuide`, which tracks the state of a guide being added, `studentName` and `studentRegNo` to store input values, `guideList` to store a list of guides, `selectedBatchNo` to keep track of the selected batch, and `projects` to store project data.

2. Context Usage:

The `useContext` hook is employed to access the `userInfo` from the `UserInfoContext`. Contexts in React provide a way to share values like user information across components without prop drilling.

3. Dark Mode Switching:

There is a dark mode feature implemented using the `isDarkMode` state. The `handleTheme` function toggles the 'dark' class on the document's root element, allowing for easy styling adjustments based on the dark mode state.

4. Project Data Fetching:

The component utilizes the `useEffect` hook to fetch guide-related data when the component mounts. Axios is used to make an asynchronous HTTP POST request to 'http://localhost:8000/allteams/batch/' with the guide's name. The retrieved data, specifically the batch numbers, is then used to update the `guideList` state.

5. Student Submission Handling:

There is a function named `handleStudentSubmission` that makes an HTTP POST request to 'http://localhost:8000/student/add/' with data containing the `studentName` and `studentRegNo`. Success and error scenarios are handled with appropriate actions such as closing a modal and displaying an alert.

6. Project Data Filtering:

The `useEffect` hook with a dependency on `selectedBatchNo` is used to fetch project data based on the selected batch number. The fetched data is then stored in the `projects` state.

7. Authentication and Navigation:

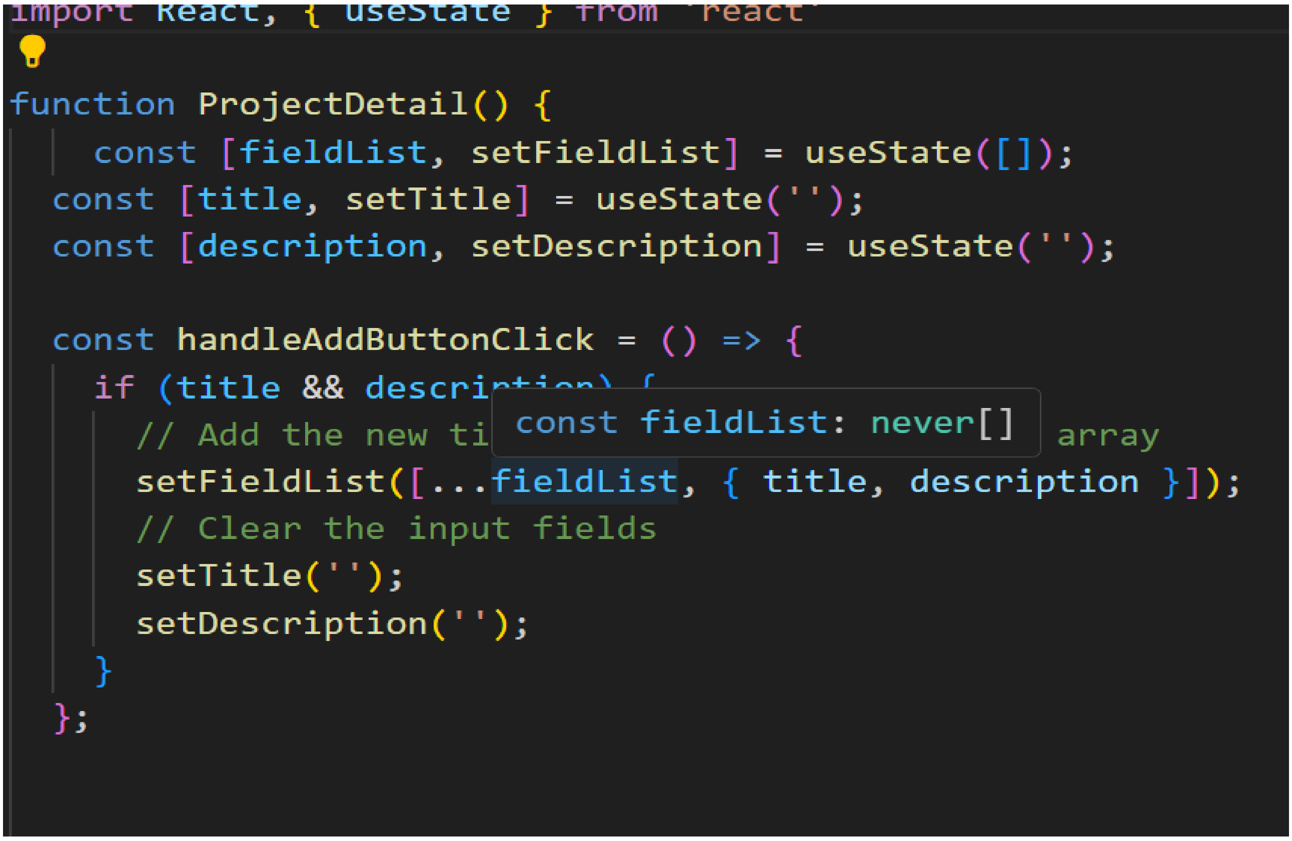
There is a check for staff authorization in the `useEffect` hook, and if the user is not authorized as staff, it redirects to the home page. The user's authorization status is stored in the local storage.

8. Logout Functionality:

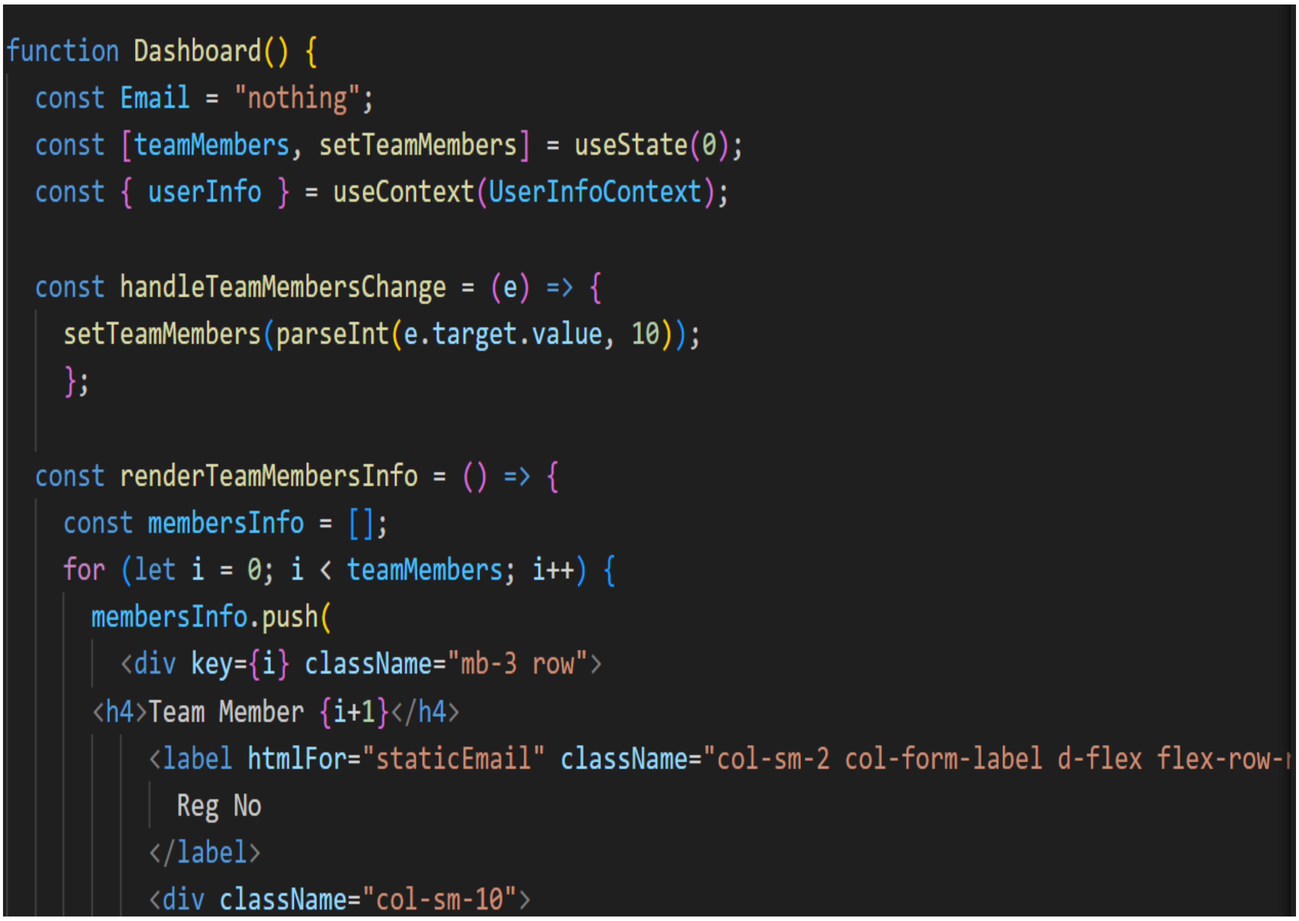
The `handleLogout` function is responsible for logging the user out by updating local storage values and navigating back to the home page.

In summary, the `Staff` component encompasses various features including guide and project data retrieval, dark mode functionality, student submission handling, and user authentication. The use of React hooks enhances state management and ensures that side effects, such as data fetching and navigation, are handled in a controlled manner. Additionally, Axios is employed for making asynchronous requests to interact with the backend server.

Step 4: Students implementation



```
import React, { useState } from 'react'

function ProjectDetail() {
    const [fieldList, setFieldList] = useState([]);
  const [title, setTitle] = useState('');
  const [description, setDescription] = useState('');

  const handleAddButtonClick = () => {
    if (title && descri
      // Add the new ti const fieldList: never[] array
      setFieldList([...fieldList, { title, description }]);
      // Clear the input fields
      setTitle('');
      setDescription('');
    }
  };
```



```
function Dashboard() {
  const Email = "nothing";
  const [teamMembers, setTeamMembers] = useState(0);
  const { userInfo } = useContext(UserInfoContext);

  const handleTeamMembersChange = (e) => {
    setTeamMembers(parseInt(e.target.value, 10));
    };

  const renderTeamMembersInfo = () => {
    const membersInfo = [];
    for (let i = 0; i < teamMembers; i++) {
      membersInfo.push(
        <div key={i} className="mb-3 row">
      <h4>Team Member {i+1}</h4>
          <label htmlFor="staticEmail" className="col-sm-2 col-form-label d-flex flex-row-r
            Reg No
          </label>
          <div className="col-sm-10">
```

The provided code is a set of React components that seem to be part of a web application for managing and displaying project details for students. Let's break down the code into its main components and functionalities:

1. Dashboard Component:

This component renders the main dashboard layout for a student. It uses a context called `UserInfoContext` to get information about the user (presumably the logged-in student).

The dashboard includes a form for entering project details like project title, registration number, class, and the number of team members. The number of team members dynamically influences the rendering of additional input fields for each team member (name, registration number, class, and email).

2. Main Component:

This component appears to handle the main content of the application. It displays information about the project, including its title, guide, team members, and progress.

There is a progress bar that visually represents the completion of certain milestones in the project.

The component fetches data from a server using Axios, including project details and team information based on the lead registration number.

It allows the addition of new team members and manages reviews associated with the project, updating the progress bar accordingly.

3. ProjectDetail Component:

This component provides a form for adding and displaying project details such as title and description. It maintains a list of added fields (presumably for different projects).

4. General Observations:

The code structure is based on functional components and hooks, indicating the use of React features introduced in more recent versions. Axios is used for making the use of asynchronous requests to a server, presumably to fetch or update data related to the student's project. The styling is done using Bootstrap classes and some custom CSS. There is a popup for adding new team members in the Main component, making the application interactive and user-friendly.

Overall, the code implements a student dashboard for managing project details, team members, and project progress. It relies on React for building the user interface and Axios for handling server communication. The application structure suggests a modular and component-based design, promoting code reusability and maintainability.

RESULT:

The implementation of the "Streamlined Project Collaboration and Efficient Management Tool" has yielded transformative results in the realm of academic project management. The transition from manual, paper-based methods and Excel spreadsheets to a digitalized web application marks a paradigm shift in the efficiency and effectiveness of project collaboration. Students now benefit from a streamlined process, where they can effortlessly update and manage their project details in real-time. This not only enhances their engagement but also fosters a more collaborative and dynamic project environment. Staff, on the other hand, experience a significant reduction in administrative burdens as they can efficiently oversee and manage teams through the centralized digital platform. The result is a more transparent, accessible, and accountable project management system. Overall, the implementation of this tool has culminated in a tangible enhancement of productivity, collaboration, and monitoring capabilities, offering a promising outlook for the future of academic project management.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

"Streamlined Project Collaboration and Efficient Management Tool," the project has successfully addressed the inherent inefficiencies of manual project monitoring within academic settings. Traditionally reliant on paper-based and Excel-centric management, this innovative web application stands as a beacon of digitalization, offering students a dynamic platform to effortlessly update their project details and enabling staff to efficiently oversee, manage, and monitor project teams. The implementation of a user-friendly web interface not only revolutionizes the way students engage with their project progress but also empowers staff with streamlined team management capabilities. By bridging the gap between manual processes and digital efficiency, this tool not only enhances collaboration and transparency but also marks a significant stride towards modernizing and optimizing the landscape of academic project management. As students and staff seamlessly navigate this digital realm, the "Streamlined Project Collaboration and Efficient Management Tool" emerges as a catalyst for a more agile, connected, and effective approach to overseeing and advancing academic projects.

6.2 FUTURE SCOPE:

The future scope of the "Streamlined Project Collaboration and Efficient Management Tool" holds immense potential in transforming the landscape of academic project management. As technology continues to evolve, the project is well-positioned to adapt and incorporate cutting-edge advancements. One avenue for expansion lies in the integration of artificial intelligence (AI) and machine learning (ML) algorithms. These technologies could analyze historical project data, providing predictive insights for better resource allocation, task estimation, and risk management.

Additionally, the tool could evolve to support a wider range of project types and disciplines, catering to the diverse needs of academic institutions. Collaboration features may be expanded to include virtual reality (VR) or augmented reality (AR) interfaces, fostering a more immersive and interactive project experience. Furthermore, the tool could explore synergies with emerging educational technologies, incorporating features that enhance learning outcomes and promote knowledge-sharing among students. As institutions increasingly embrace online and hybrid learning models, the tool could evolve to seamlessly integrate with various e-learning platforms, ensuring a holistic approach to project-based learning. Ultimately, the future scope envisions Streamlined Project Collaboration and Efficient Management Tool as an adaptive, intelligent, and comprehensive solution, continually enhancing the efficiency and effectiveness of academic project.

SOURCE CODE:

ADMIN.JS

```
import React, { useContext, useEffect, useState } from 'react'
import { useNavigate } from 'react-router-dom';
import './style.css';
import UserInfoContext from '../UsenInfoContext';
import axios from 'axios';

function Admin() {

  const [isAddingGuide, setIsAddingGuide] = useState(false);
  const [guideName, setGuideName] = useState('');
  const [guideRegNo, setGuideRegNo] = useState('');
  const [guideList, setGuideList] = useState([]);
  const [selectedGuide, setSelectedGuide] = useState('');
  const openGuidePopup = () => {
    setIsAddingGuide(true);
  };

  const closeGuidePopup = () => {
    setIsAddingGuide(false);
  };

  useEffect(() => {
    axios.get('http://localhost:8000/guide/getlist/')
      .then(response => {

        setGuideList(response.data.Names);
        console.log(response.data.Names)
      })
      .catch(error => {
        console.error('Error fetching guide list:', error);
      });
  }, []);


  const handleGuideSubmission = () => {
    // Perform axios request to add the new guide
    const data = {
      Name: guideName,
      Reg_No: guideRegNo,
    }
    console.log(data)
    axios.post('http://localhost:8000/guide/add/', data)
      .then(response => {
        // Handle success, e.g., close the modal and refresh data
        closeGuidePopup();
        // You might want to fetch data again to update the projects list
      })
      .catch(error => {
        alert('Error adding guide:', error);
        closeGuidePopup();
      });
  };
```

STUDENTS.JS

```
import React, { useContext, useState } from 'react'
import './dash.css';
import UserInfoContext from '../UsenInfoContext';

function Dashboard() {
    const Email = "nothing";
    const [teamMembers, setTeamMembers] = useState(0);
    const { userInfo } = useContext(UserInfoContext);

    const handleTeamMembersChange = (e) => {
            setTeamMembers(parseInt(e.target.value, 10));
      };

  const renderTeamMembersInfo = () => {
    const membersInfo = [];
    for (let i = 0; i < teamMembers; i++) {
      membersInfo.push(
        <div key={i} className="mb-3 row">
                    <h4>Team Member {i+1}</h4>
          <label htmlFor="staticEmail" className="col-sm-2 col-form-label d-flex flex-row-reverse">
            Reg No
          </label>
          <div className="col-sm-10">
            <input
              className="form-control"
              type="text" value={Email}  />
          </div>
          <label htmlFor="staticEmail" className="col-sm-2 col-form-label d-flex flex-row-reverse">
            Name
          </label>
          <div className="col-sm-10">
            <input
              className="form-control"
              type="text"
              value={Email}
            />
          </div>
              <label htmlFor="staticEmail" className="col-sm-2 col-form-label d-flex flex-row-reverse">
            Class
          </label>
          <div className="col-sm-10">
            <input
              className="form-control"
              type="text"
              value={Email}
            />
          </div>
              <label htmlFor="staticEmail" className="col-sm-2 col-form-label d-flex flex-row-reverse">
            Email
          </label>
          <div className="col-sm-10">
            <input
              className="form-control"
              type="text"
```

```
       value={Email}
      />
     </div>
    </div>
   );
  }
  return membersInfo;
 };
```

STAFF.JS

```
import React, { useContext, useEffect, useState } from 'react'
import { useNavigate } from 'react-router-dom';
import axios from 'axios';
import UserInfoContext from '../UsenInfoContext';
import './staff.css'

function Staff() {
 const [isAddingGuide, setIsAddingGuide] = useState(false);
 const [studentName, setStudentName] = useState('');
 const [studentRegNo, setStudentRegNo] = useState('');
 const [guideList, setGuideList] = useState([]);
 const [selectedBatchNo, setSelectedBatchNo] = useState('');

 const openGuidePopup = () => {
  setIsAddingGuide(true);
 };

 const closeGuidePopup = () => {
  setIsAddingGuide(false);
 };
```

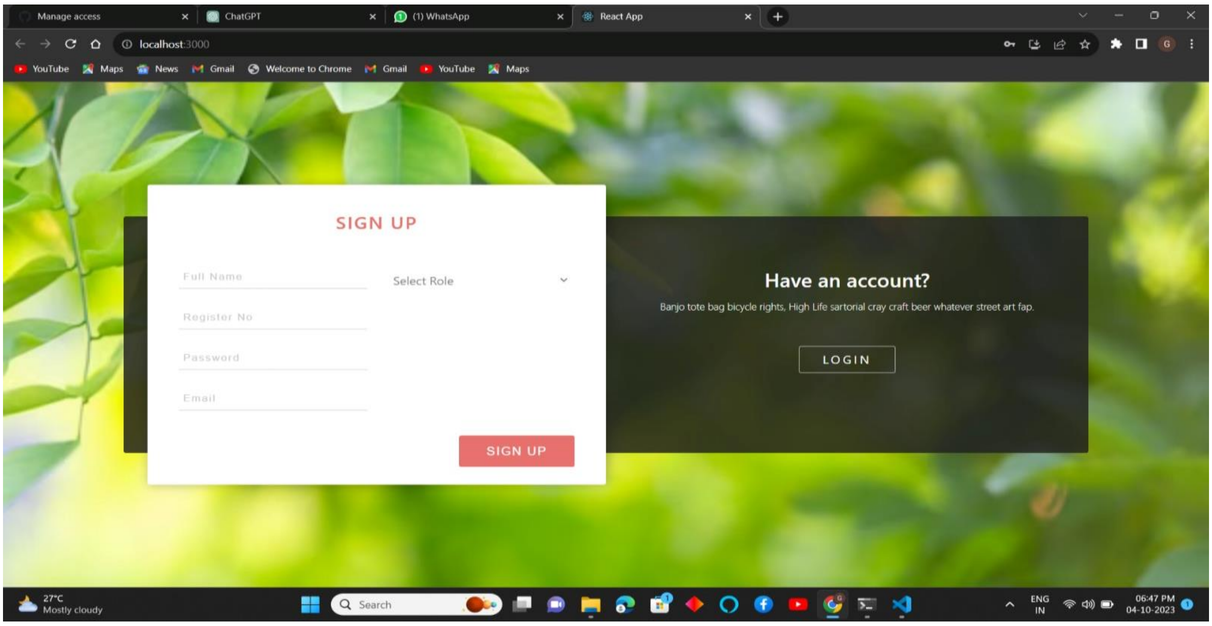
```
useEffect(() => {
  console.log(userInfo.Name)
  axios.post('http://localhost:8000/allteams/batch/',{'guide_name': userInfo.Name})
    .then(response => {
      setGuideList(response.data.Batch_No);
      console.log(response.data.Batch_No)
    })
    .catch(error => {
      console.error('Error fetching guide list:', error);
    });
}, []);

const handleStudentSubmission = () => {
  // Perform axios request to add the new guide
  const data = {
    Name: studentName,
    Reg_No: studentRegNo,
  }
  console.log(data)
  axios.post('http://localhost:8000/student/add/', data)
    .then(response => {
      // Handle success, e.g., close the modal and refresh data
      closeGuidePopup();
      // You might want to fetch data again to update the projects list
    })
    .catch(error => {
      alert('Error adding guide:', error);
      closeGuidePopup();
    });
  };
  const [projects, setProjects] = useState([]);
  const { userInfo } = useContext(UserInfoContext);
  const [isDarkMode, setIsDarkMode] = useState(false);
  const handleTheme = () => {
    document.documentElement.classList.toggle('dark');
    setIsDarkMode((prevMode) => !prevMode);
}
  }
}
```

SIGNUP .JS

```
import React, { useContext, useEffect, useState } from 'react';
import './signin.css';
import axios from 'axios';
import { useNavigate } from 'react-router-dom';
import UserInfoContext from '../UsenInfoContext';

function Signin() {
  const [userFormsClass, setUserFormsClass] = useState('');
  const [newUserName, setNewUserName] = useState('');
  const [selectedRole, setSelectedRole] = useState('');
  const [newUserId, setNewUserId] = useState('');
  const [newPassword, setNewPassword] = useState('');
  const [newEmail, setNewEmail] = useState('');
  const [selectedStaffIncharge, setSelectedStaffIncharge] = useState('');
  const [selectedBatchNo, setSelectedBatchNo] = useState('');
  const [selectedYear, setSelectedYear] = useState('');
  const [exPassword, setExPassword] = useState('');
const [exUserId, setExUserId] = useState(");
const navigate = useNavigate();
  const { setUserInfo } = useContext(UserInfoContext);



  const [isauthorized, setIsAuthorized] = useState(
    localStorage.getItem("isauthorized") || ""
  );
  const [isSigned, setIsSigned] = useState(
    localStorage.getItem("isSigned") === "true" ? true : false
  );

  const handleSignupClick = () => {
    setUserFormsClass('bounceLeft');
  };

  const handleLoginClick = () => {
    setUserFormsClass('bounceRight');
  };
```

OUTPUT:

LOGIN MODULE:



SIGN UP

Full Name

Register No

Password

Email

Select Role

SIGN UP

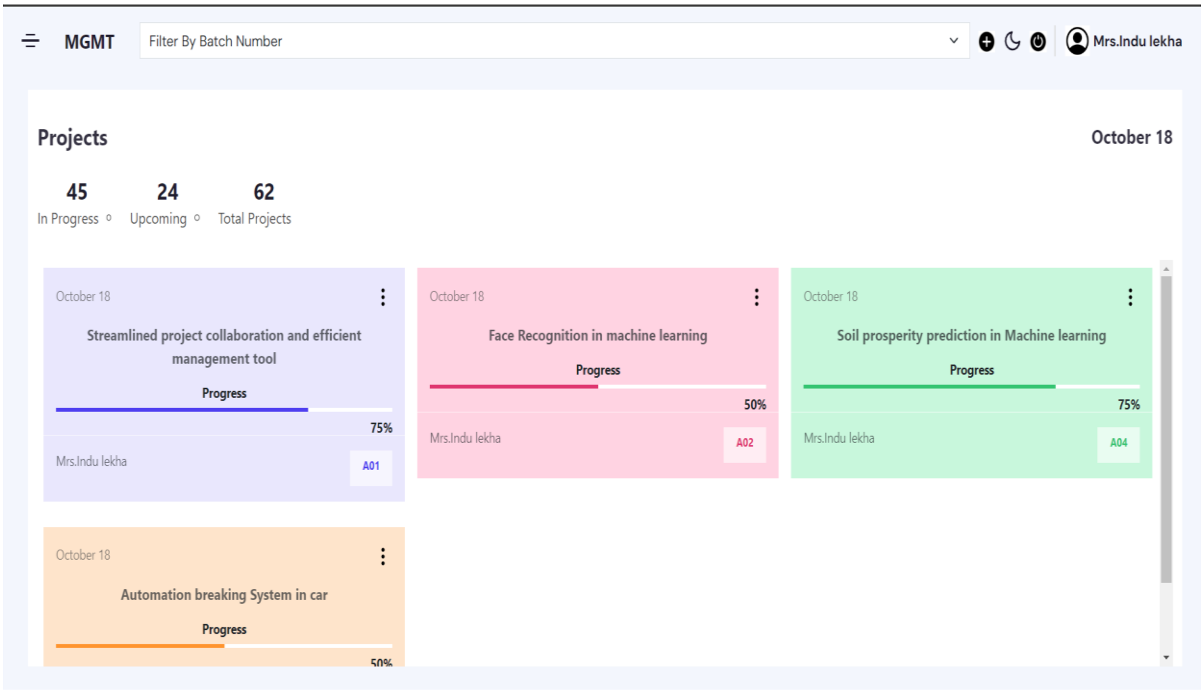
Have an account?

Banjo tote bag bicycle rights, High Life sartorial cray craft beer whatever street art fap.

LOGIN

Figure 1: Login

STAFF MODULE


MGMT

Filter By Batch Number

Mrs.Indu lekha

Projects

October 18

45 In Progress

24 Upcoming

62 Total Projects

October 18
Streamlined project collaboration and efficient management tool
Progress
75%
Mrs.Indu lekha
A01

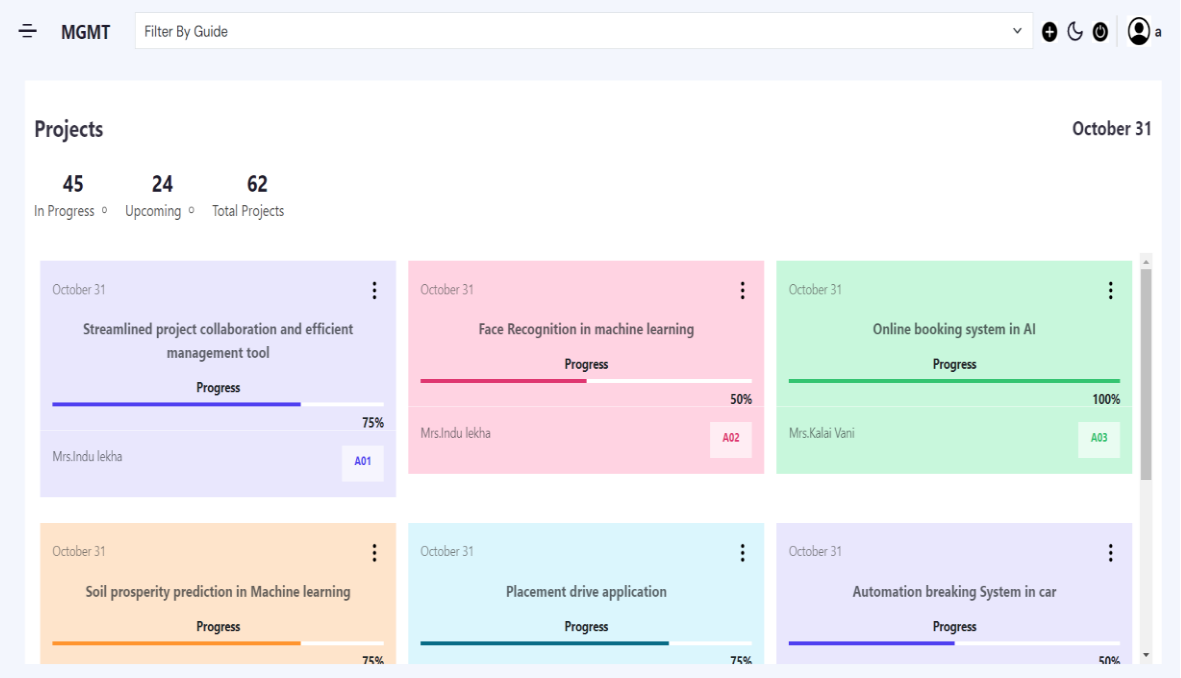
October 18
Face Recognition in machine learning
Progress
50%
Mrs.Indu lekha
A02

October 18
Soil prosperity prediction in Machine learning
Progress
75%
Mrs.Indu lekha
A04

October 18
Automation breaking System in car
Progress
50%

Figure 2: Staff

ADMIN MODULE:



MGMT

Filter By Guide

Projects

October 31

45 In Progress

24 Upcoming

62 Total Projects

October 31

Streamlined project collaboration and efficient management tool

Progress

75%

Mrs.Indu lekha

A01

October 31

Face Recognition in machine learning

Progress

50%

Mrs.Indu lekha

A02

October 31

Online booking system in AI

Progress

100%

Mrs.Kalai Vani

A03

October 31

Soil prosperity prediction in Machine learning

Progress

75%

October 31

Placement drive application

Progress

75%

October 31

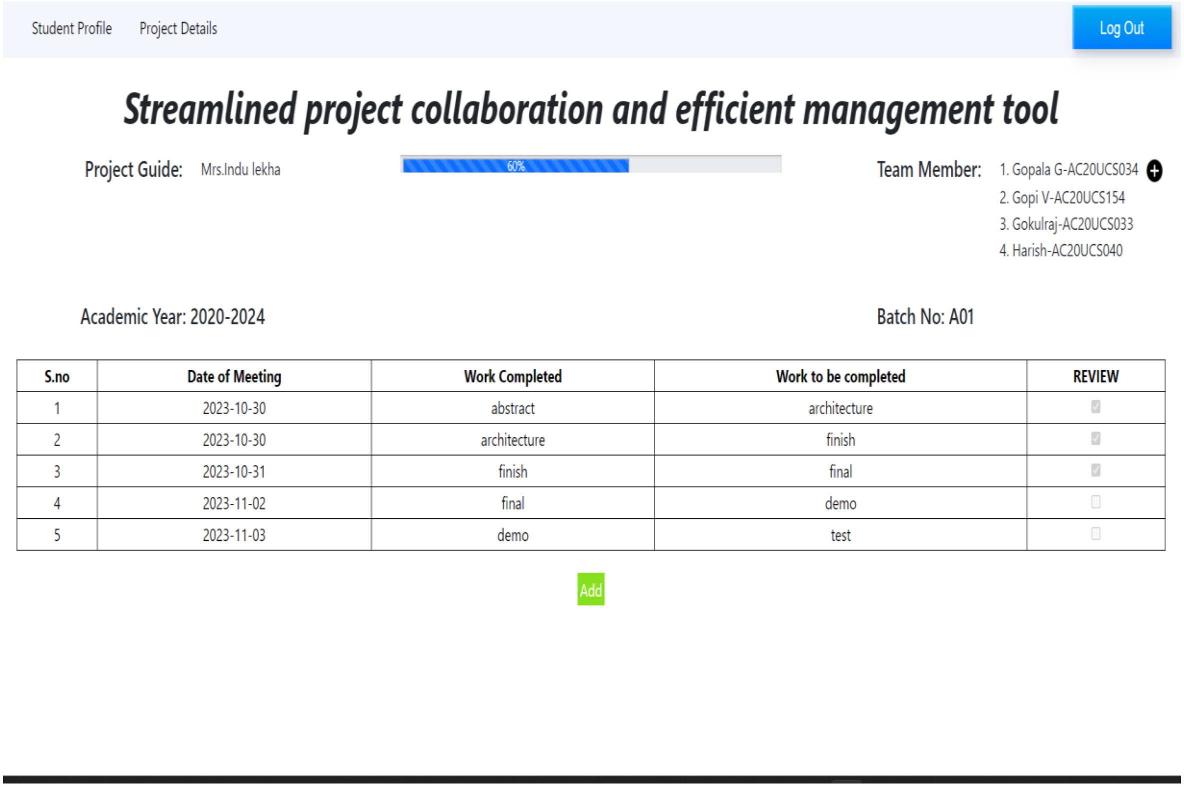
Automation breaking System in car

Progress

50%

Figure 3: Admin

STUDENT MODULE:



Student Profile

Project Details

Log Out

Streamlined project collaboration and efficient management tool

Project Guide: Mrs.Indu lekha

60%

Team Member:

1. Gopala G-AC20UCS034
2. Gopi V-AC20UCS154
3. Gokulraj-AC20UCS033
4. Harish-AC20UCS040

Academic Year: 2020-2024

Batch No: A01

S.no	Date of Meeting	Work Completed	Work to be completed	REVIEW
1	2023-10-30	abstract	architecture	☑
2	2023-10-30	architecture	finish	☑
3	2023-10-31	finish	final	☑
4	2023-11-02	final	demo	□
5	2023-11-03	demo	test	□

Add

Figure 4: Student

REFERENCES

1. F. Castro, A. Vellido, À. Nebot, and F. Mugica, “Applying Data Mining Techniques to e-Learning Problems”, In: Evolution of Teaching and Learning Paradigms in Intelligent Environment, Vol. 62 (2007), pp. 183-221

2. G. R. Cates and M. Mollaghasemi, "The project assessment by simulation technique", *Engineering Management Journal*, vol. 19, pp. 3-10, 2007.

3. H. G. Gemünden, *Success factors of global new product development programs the definition of project success knowledge sharing and special issues of project management journal® “Project Management Journal*, vol. 46, pp. 2-11, 2015.

4. Q. Liu et al., "EKT: Exercise-aware knowledge tracing for student performance prediction", *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 100-115, Jan. 2021.

5. T. Long et al., "Improving knowledge tracing with collaborative information", *Proc. 15th ACM Int. Conf. Web Search Data Mining*, pp. 599-607, 2022.

6. M. Novák, M. Biňas, “Automated Testing of Case Studies in Programming Courses”,In: ICETA 2011 : 9th IEEE International Conference on Emerging eLearning Technologies and Applications, Stará Lesná, The High Tatras, October 27-28 2011, Slovakia, pp. 157-162, ISBN 978-1-4577-0050-7.https://www.ipd.kit.edu/jplag/.

7. T. A. Oun, T. D. Blackburn, B. A. Olson and P. Blessner, "An enterprise-wide knowledge management approach to project management", *Engineering Management Journal*, vol. 28, pp. 179-192, 2016

8. J. Palmer, “Introducing the New GitHub Graphs”, The GiHub blog, Accessed: 11.10.2012. Online:https://github.com/blog/1093-introducing-the-new-github-graphs.

9. V. S. Parsons, "Project performance: How to assess the early stages", *Engineering Management Journal*, vol. 18, pp. 11-15, 2006.8. jPlag tool home page, Accessed: 10.10.2012, Online:

https://www.ipd.kit.edu/jplag/.

10. P. Pivec, "Game-based Learning or Game-based Teaching?", British Educational Communications and Technology Agency (BECTA), July 2009, Accessed: 10.10.2012, Online: http://dera.ioe.ac.uk/1509/.

11. A. Pmi, "guide to the project management body of knowledge (PMBOK guide)", *Project Management Institute*, 2013.

12. N. Safer and S. Fleischman, "Research matters: How student progress monitoring improves instruction", *Educ. Leadership*, vol. 62, pp. 81-83, 2005..

13. V. L. Stocker, Moodle 2.0 Science: Monitoring Your Students' Progress, Packtpub blog, April 2011, Accessed: 11.10.2012, Online: http://www.packtpub.com/article/moodle-2-science-monitoring-your-students-progress.

14. A. Weinberg, "Web Tracking and Students' Work Patterns in Online Language-learning Activities", CALICO Journal, 25 (1), September 2007, pp. 31-47.

15. H S Zhang, X Ren, M Z Ni et al., "Research on the impact of project manager's authorized leadership behavior on project performance [J]", *Journal of Engineering Management*, vol. 33, no. 04, pp. 100-104, 2019.