Design And Implementation of Mentor Connect: An AI-Driven Student–Teacher Mentoring Platform

Prof. Shyamsundar Magar[1], Akhare Jayesh[2], Kakade Sujit[3], Navathar Rohit[4]
[1,2,3,4]*Department of Information Technology, Trinity College of Engineering and Research*

***Abstract*—Mentoring is essential for academic achievement and professional growth; however, conventional mentoring systems are often limited by scalability issues, inefficient data management, and insufficient personalization. This paper presents the design and development of Mentor Connect, an AI-enabled student–teacher mentoring platform that integrates mentoring activities with institutional management functions. The system follows the Agile Software Development Life Cycle (SDLC), supporting iterative development and adaptability to changing requirements. A layered client–server architecture is implemented using Flutter and Firebase to ensure scalability and maintainability. Object-oriented design principles are applied to achieve structured data modeling and modular implementation. Core functionalities include role-based authentication, attendance tracking, leave and gate pass management, complaint handling, and analytics-driven reporting. In addition, AI-assisted mentoring features provide personalized feedback and support informed decision-making.**

***Index Terms*—Mentor Connect, AI-Based Mentoring, Agile SDLC, Academic Management System, Object-Oriented Design, Role-Based Access, Flutter**

I. INTRODUCTION

Effective mentoring significantly contributes to improving students' academic performance, skill development, and over-all growth within educational institutions. However, traditional mentoring approaches are often limited by scalability constraints, inefficient administrative processes, and a lack of personalization. These limitations increase the workload on faculty members and reduce their ability to provide consistent and meaningful guidance to each student, thereby impacting the effectiveness of mentoring outcomes.

With the advancement of digital technologies, educational institutions are increasingly adopting centralized and data-driven approaches to modernize mentoring practices. In particular, the integration of Artificial Intelligence (AI) offers new opportunities to improve mentoring efficiency. Rather than replacing human mentors, AI serves as a supportive component within a teacher–AI Collaboration (TAC) framework. MentorConnect is proposed as an integrated platform that combines academic mentoring with institutional management functionalities. The system includes intelligent features for mentor–mentee allocation, structured goal tracking, and continuous performance evaluation, along with secure communication mechanisms. By consolidating these capabilities into a scalable solution, MentorConnect aims to enhance mentoring effectiveness while streamlining administrative processes in educational environments.

II. PROBLEM STATEMENT

Traditional mentoring and campus management systems are limited in terms of scalability, accessibility, and personalization. Manual and paper-based processes for tasks such as attendance tracking and reporting are time-consuming and susceptible to errors. This work addresses these challenges by proposing a comprehensive AI-assisted platform that modernizes campus management and provides a scalable, personalized, and data-driven framework for student-teacher mentoring.

III. RELATED WORK

Teacher–AI Collaboration (TAC):

Prior studies on the evolution of Teacher–AI Collaboration demonstrate that AI can effectively function as a supportive partner in mentoring environments. These studies highlight that AI-based feedback mechanisms can significantly reduce mentor

workload while still enabling personalized interaction with students.

Digital Mentoring Systems:
An integrative review of mentoring software in education highlights key issues such as limited scalability, high administrative effort, and lack of structured communication. The findings suggest that effective mentoring platforms should incorporate features such as goal tracking, progress monitoring, analytics, and centralized communication to enhance mentoring outcomes and faculty satisfaction.

Intelligent Online Mentoring Platforms:
Existing solutions, including Intelligent Online Mentoring Systems (OMS), demonstrate the effectiveness of client–server architectures in supporting real-time interaction between mentors and students. While these systems provide essential functionalities such as user management, communication modules, and performance tracking, they often lack advanced AI-driven personalization.

Mobile-Based Academic Monitoring Applications:
Re-search on mobile applications for student performance monitoring indicates that real-time access to attendance, academic records, and reports improves parental involvement and administrative efficiency. These studies emphasize the significance of mobile-first approaches in modern educational systems.

Design and Architecture of Mentoring Applications:
Research focusing on the design of mentoring applications highlights the importance of user-centric interfaces, modular system architecture, and scalable client–server models for efficient implementation.

IV. LITERATURE REVIEW

Recent research on mentoring platforms and AI-assisted educational systems has focused on overcoming key challenges such as scalability, efficiency, and personalization within aca-demic environments. Pesina (2025) emphasizes that mentoring software contributes significantly to teacher development by supporting structured goal setting, continuous progress monitoring, and analytics-based evaluation [1].

Yoshida et al. (2025) proposed the concept of Teacher–AI Collaboration (TAC), in which Artificial Intelligence operates as a supportive partner in the mentoring process. Their study demonstrates that AI-assisted systems can effectively reduce mentor workload while delivering structured and less stressful feedback to students [2].

U. P. et al. (2022) developed an Intelligent Ubiquitous Online Mentoring System based on a client–server architecture, facilitating efficient communication and centralized management of mentoring activities [3]. In a similar direction, Giraldo et al. (2020) highlight the importance of user-centric design and scalable architecture in mentoring applications, emphasizing the benefits of modular and mobile-first approaches [4].

Amran et al. (2023) demonstrates that mobile-based aca-demic monitoring systems enhance real-time tracking of student performance and improve communication among stake-holders, including parents, faculty, and administrators [5].

V. OBJECTIVES

The key objectives of the MentorConnect project are as follows:

- Develop a Multi-Role Integrated Platform:
To design and implement a secure and unified application that supports multiple stakeholders, including students, faculty (GFM), parents, and administrators, with role-based dashboards and controlled access.

- Integrate AI-Assisted Mentoring:
To incorporate Artificial Intelligence techniques that assist mentors in delivering structured feedback, analyzing student performance, and enabling personalized mentoring through the Teacher–AI Collaboration (TAC) framework.

- Support Real-Time Monitoring and Analytics:
To pro-vide functionalities for continuous tracking of attendance, academic progress, and student activities, along with analytical tools for data-driven decision-making and early detection of at-risk students.

- Automate Academic and Administrative Processes:

To streamline operations such as leave management, gate pass requests, complaint handling, and report generation, thereby reducing manual workload and improving efficiency.

- Design a Scalable and Modular Architecture:

To implement a layered and modular system architecture that facilitates future enhancements, ease of maintenance, and integration of additional features.

VI. PROPOSED SYSTEM AND ARCHITECTURE

MentorConnect is designed as a scalable, modular, and intelligent academic mentoring platform that integrates student management with AI-assisted mentoring capabilities. The system adopts a three-tier architecture, ensuring a clear separation between the presentation, application, and data layers. This architectural approach improves maintainability, scalability, and overall system performance while supporting multiple user roles and real-time interactions. The platform utilizes the Flutter framework to provide a unified cross-platform interface, enabling seamless deployment across mobile and web platforms.

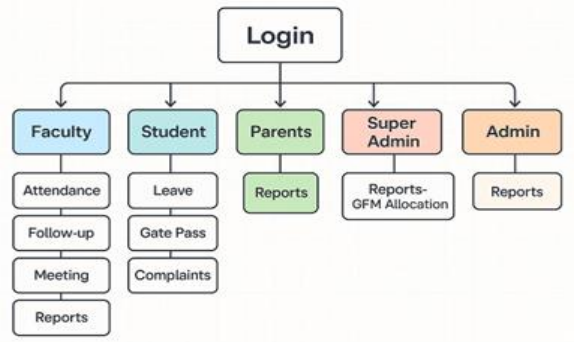

Fig. 1. MentorConnect: Student–Teacher Mentoring Flowchart

A. System Modules and Functionality

The system is designed to accommodate multiple stake-holders within an academic institution, each with specific functionalities and access privileges. The primary user roles include:

- Faculty (GFM):

Faculty members serve as mentors responsible for monitoring student performance, recording attendance, scheduling mentoring sessions, and generating reports. The integration of AI-based assistance enables efficient analysis of student data and supports the generation of structured feedback, thereby reducing manual workload and enhancing mentoring quality.

- Student:

Students use the platform to access attendance records, apply for leave and gate passes, submit complaints or suggestions, and monitor their academic progress. They can also participate in goal-setting activities and receive AI-assisted feedback, promoting continuous learning and engagement.

- Parents:

Parents are provided with controlled access to track their child's academic performance, attendance, and reports. This feature improves transparency and strengthens communication between the institution and parents.

- Admin / Super Admin:

Administrative users are re-sponsible for managing system operations, including user registration, role assignment, GFM allocation, and report generation. They also utilize analytics dashboards to support institutional decision-making and performance evaluation.

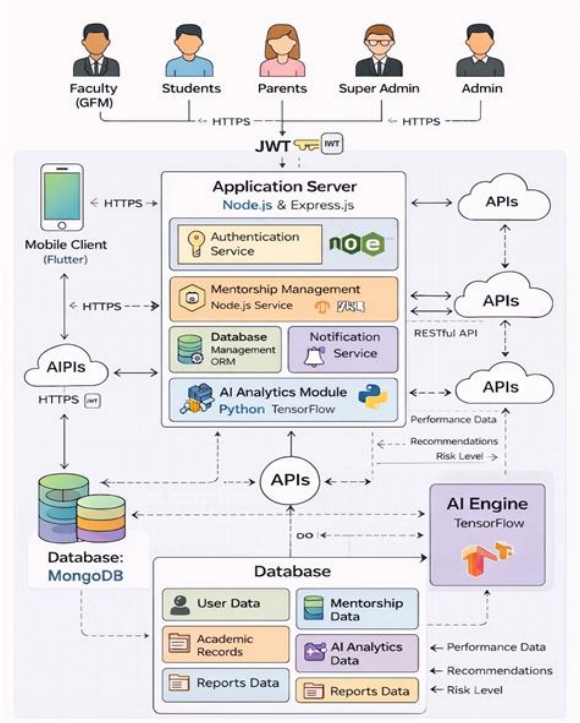

Fig. 2. MentorConnect: System Architecture

B. Technical Architecture

The technical design of MentorConnect is divided into three main layers:

1) Frontend Client (Presentation Layer):

The frontend is developed using the Flutter framework, offering a responsive and user-friendly interface for all stakeholders. It provides role-based dashboards, data entry forms (for attendance, leave, and complaints), and visualization components for reports and analytics.

2) Backend Server (Application Layer):

The backend layer manages business logic, authentication, authorization, and data processing. It implements role-based access control (RBAC) to ensure secure system access. Additionally, this layer integrates AI components for recommendation, feedback generation, and performance analysis.

3) Cloud Database (Data Layer):

The system uses MongoDB Atlas, a cloud-based NoSQL database, to store both structured and unstructured data, including user profiles, attendance records, leave requests, complaints, and reports. The database ensures scalability, high avail-ability, and efficient data retrieval, while maintaining data integrity through proper schema design and validation.

VII. KEY FEATURES OF MENTORCONNECT

MentorConnect incorporates a comprehensive set of features derived from existing research in digital mentoring systems, educational technology, and AI-assisted platforms. These features are designed to improve mentoring efficiency, enhance student engagement, and streamline academic administration through automation and intelligent decision support.

Fig. 3. UML Diagram Representing System Components and Interactions

- Intelligent Mentor–Mentee Matching:

The system employs algorithm-based matching techniques to assign mentors to students using multiple criteria such as aca-demic performance, subject expertise, interests, and career objectives. This approach minimizes manual allocation efforts and enhances mentoring effectiveness by ensuring better compatibility and personalized support.

- Goal Setting and Progress Tracking:

MentorConnect allows mentors and students to collaboratively define clear and measurable goals within a specified timeframe. The system continuously monitors progress through dash-boards and performance indicators, enabling users to evaluate achievements, identify gaps, and take necessary corrective actions.

- AI-Assisted Feedback and Recommendation:

The plat-form integrates AI components to assist mentors in generating structured feedback based on student performance, attendance, and activity data. It also provides personalized recommendations to support continuous improvement while reducing the manual workload on mentors.

- Real-Time Communication and Interaction:

The system includes communication features such as messaging and notifications, facilitating seamless interaction among students, faculty, and parents. This ensures timely information exchange, efficient issue resolution, and sustained engagement throughout the mentoring process.

VIII. MENTORING WORKFLOW

The mentoring process in MentorConnect is structured as a continuous and iterative cycle that facilitates organized interaction between mentors and students. This workflow supports systematic goal management, ongoing performance assessment, and timely feedback, aligning with modern data-driven mentoring approaches. The cyclical design enables continuous refinement and adaptive learning, thereby improving the overall effectiveness and personalization of the mentoring process.

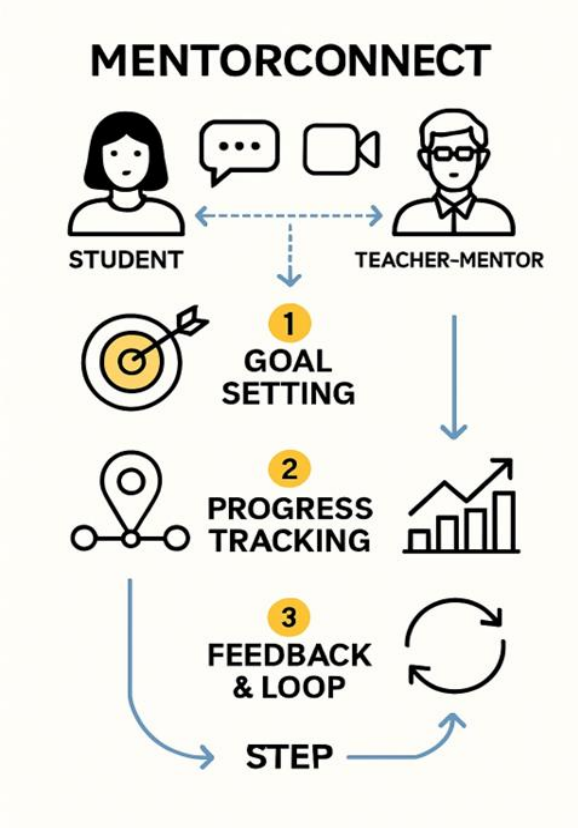

Fig. 4. MentorConnect: Mentoring Workflow Cycle

1) Goal Setting:
The process begins with collaborative goal formulation between the mentor and the student. Goals are defined based on academic performance, career objectives, and individual improvement areas. These goals are structured, measurable, and time-bound, and are recorded within the system for continuous monitoring. This step ensures proper alignment between mentor expectations and student objectives.

2) Progress Tracking and Monitoring:
After goal definition, the system continuously evaluates student progress using indicators such as attendance, academic performance, task completion, and participation in mentoring activities. Real-time dashboards and analytics provide insights to both mentors and students, enabling early identification of performance gaps and facilitating timely intervention.

3) Feedback and Evaluation:
Based on the collected data, mentors provide structured and constructive feedback to students. The system also incorporates AI-assisted mechanisms that analyze performance trends and recommend improvement strategies. This approach ensures consistency, objectivity, and scalability in the mentoring process.

4) Iteration and Continuous Improvement:
Following feedback, students work on improving their performance, and the mentoring cycle is repeated with updated or refined goals. This iterative process promotes continuous learning, enhances engagement, and encourages accountability, contributing to long-term academic and professional development.

IX. METHODOLOGY

The proposed MentorConnect system is developed using a structured and iterative methodology based on Agile Software Development principles combined with a three-tier architectural model. This approach ensures flexibility, continuous improvement, and efficient handling of evolving requirements in an academic mentoring environment. The methodology also aligns with the concept of Teacher–AI Collaboration (TAC), enabling the gradual integration of Artificial Intelligence as a supportive and intelligent component within the mentoring process. The use of a unified Flutter-based framework facilitates rapid development, cross-platform compatibility, and streamlined system integration.

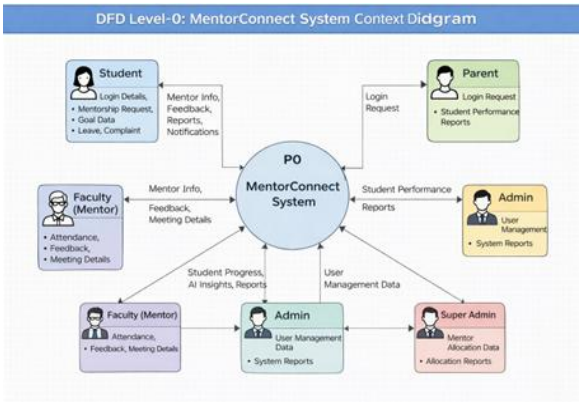

Fig. 5. Data Flow Diagram (Level-0)

- Requirement Analysis and System Design:
The initial phase involves comprehensive requirement gathering from stakeholders, including students, faculty, parents, and administrators. Functional and non-functional requirements are formally documented in the Software Requirements Specification (SRS).

- Iterative Development Using Agile Sprints:

The system is developed incrementally through multiple Agile sprints, where each iteration focuses on implementing specific modules such as authentication, attendance management, leave processing, complaint handling, and re-porting. Continuous integration and stakeholder feedback ensure alignment with user requirements.

- AI Module Integration and Enhancement:

The Artificial Intelligence component is incorporated in a phased manner, initially supporting basic functionalities such as recommendation systems and automated feedback generation. More advanced capabilities, including performance analysis and personalized mentoring suggestions, are integrated in later stages.

- Testing, Validation, and Refinement:

Comprehensive testing is conducted at multiple levels, including unit testing, integration testing, system testing, and user acceptance testing. Each module is validated against predefined requirements to ensure correctness, reliability, and performance.

- Deployment and Maintenance:

The final system is deployed on a cloud-based infrastructure with secure API-based communication between frontend and backend components. The deployment supports real-time data access for all stakeholders. Post-deployment, the system is continuously monitored and maintained to ensure stability, security, and scalability.

X. BENEFITS AND CHALLENGES

A. Benefits

- Reduced Mentor Workload:

The integration of AI-assisted mentoring reduces the administrative burden on faculty by automating repetitive tasks such as performance analysis, report generation, and feedback preparation. This enables mentors to focus on meaningful student interactions and personalized guidance, thereby improving overall mentoring efficiency.

- Improved Teacher Retention and Professional Growth:

By offering structured mentoring support and reducing workload-related stress, the system enhances job satisfaction among faculty members. Access to analytics and mentoring insights also facilitates continuous professional development, which can contribute to lower teacher attrition rates.

- Enhanced Personalization in Mentoring:

The use of intelligent algorithms enables personalized mentor–mentee matching and customized feedback based on individual student performance, behavior, and goals. This results in more effective mentoring relationships and improved student outcomes.

B. Challenges

- Balancing Technology with Human Interaction:

A key challenge is ensuring that the system supports, rather than replaces, the human aspect of mentoring. Maintaining an appropriate balance between automated assistance and personal interaction is essential to preserve the quality of mentor–student relationships.

- Data Privacy and Security Concerns:

As the system processes sensitive academic and personal data, ensuring data privacy and compliance with institutional and legal standards is critical. Robust security mechanisms, including encryption, authentication, and role-based access control, must be implemented.

- Accuracy and Reliability of AI Feedback:

AI-generated feedback may initially lack contextual understanding or precision, which can affect its effectiveness. Continuous training, validation, and refinement of AI models are necessary to improve accuracy and reliability.

XI. CONCLUSION AND FUTURE WORK

This paper has presented the design and implementation of MentorConnect, an AI-driven student–teacher mentoring platform that addresses the limitations of traditional mentoring systems by integrating a scalable three-tier architecture with features such as role-based access control, real-time analytics, and AI-assisted feedback.

The incorporation of Teacher–AI Collaboration (TAC) enables Artificial Intelligence to function as a supportive assistant to mentors while maintaining a

balance between automation and human interaction. This approach facilitates structured goal setting, continuous progress monitoring, and data-driven decision-making, thereby improving the effective-ness and scalability of the mentoring process.

Future work will focus on further enhancing the AI module through predictive analytics and machine learning techniques to improve personalization and feedback accuracy. Addition-ally, the system can be extended with advanced communication features and integration with learning management systems.

ACKNOWLEDGMENT

The authors would like to thank their project guide, Prof. Shyamsundar Magar, and the Department of Information Technology for their guidance and support.

REFERENCES

[1] R. Pesina, "Mentoring Software in Education and Its Impact on Teacher Development: An Integrative Literature Review," *Int. J. Integr. Technol. Educ. (IJITE)*, vol. 14, no. 1, 2025.

[2] H. Yoshida, Y. Onishi, and M. Arimoto, "Phased Evolution of Teacher–AI Collaboration for Effective Mentoring in Teacher Education," in Proc. 13th Int. Conf. Information and Education Technology (ICIET), 2025.

[3] U. P. et al., "An Intelligent Ubiquitous Online Application for Mentoring System," Int. J. Multidiscip. Res. Anal., vol. 5, no. 6, pp. 1431–1436, 2022.

[4] H. T. Giraldo, E. M. T. Sánchez, and F. J. G. Peñalvo, "Developing a Design Phase for a Mentoring Mobile App," in Proc. Int. Conf. Technological Ecosystems for Enhancing Multiculturality (TEEM), 2020, pp. 1–10.

[5] A. N. Amran *et al.*, "Enhancing Secondary Education: A Mobile Application for Student Performance Monitoring and Curriculum Management," *Asian J. Prof. Bus. Stud.*, vol. 4, no. 2, 2023.