International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03311
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

CredBud: The Student Platform

Mrs. Vidya Shinde[1] vidya.shinde.sits@sinhgad.edu
Mr. Pratik Abhang[2] pratikabhang.sits.comp@gmail.com
Ms. Darshana Borase[3] darshanaborase.sits.comp@gmail.com
Mr. Omkar Darekar[4] omkardarekar.sits.comp@gmail.com
Mr. Chaitanya Gunde[5] chaitanyagunde.sits.comp@gmail.com

[1]Assistant Professor, Department of Computer Engineering, Sinhgad Institute of Technology and Science, Pune
[2,3,4,5]BE Students, Department of Computer Engineering, Sinhgad Institute of Technology and Science, Pune

Abstract*— The transition from manual to automated systems in academic institutions has become crucial for enhancing efficiency, accuracy, and the overall user experience. In response to this need, CredBud: The Student Platform was developed as a comprehensive solution for streamlining academic record management. Designed to support students, educators, and administrators, the platform offers a secure and user-friendly digital environment for managing key academic processes. CredBud automates essential functions such as attendance tracking, assignment monitoring, and performance evaluation. At its core is a dynamic credit system that allows students to track their academic progress and receive incentives for consistent participation and timely task completion. This reward-based model fosters greater student engagement and accountability. The platform features a robust Attendance Management System utilizing dynamic QR codes for real-time attendance recording, supplemented by manual and backup entry options to ensure reliability and reduce errors common in traditional systems. Additional tools—such as assignment verification, automated term grant sheet generation, and continuous performance tracking—further contribute to a seamless academic experience. Accessible via both mobile and web platforms, CredBud ensures ease of use and encourages widespread adoption. Its scalable infrastructure allows institutions to adapt the system to evolving needs, supporting a modern, transparent, and student-centered academic environment.

Keywords: Manual to Automation, Academic Records, QR Attendance, Credit System, Student Engagement, Reward System, Performance Tracking, Term Sheet.

1 INTRODUCTION

The shift from manual to automated systems in academic institutions has become essential for improving efficiency, accuracy, and user experience. In response to this need, **CredBud: The Student Platform** was developed as a comprehensive solution for streamlining academic record management. It is designed to benefit students, educators, and administrators through a digital platform that simplifies and secures various academic processes. CredBud automates key academic functions such as attendance tracking, assignment monitoring, and performance evaluation. A core feature is its dynamic credit system, which enables students to track their academic progress and earn incentives for consistent participation and timely task completion. This reward-based approach helps enhance student engagement and accountability. The platform includes a robust Attendance Management System that uses dynamic QR codes for real-time attendance recording, along with backup and manual entry options to ensure reliability. These features help reduce errors and the risk of manipulation common in traditional methods. Additional tools such as assignment verification, automated term grant sheet generation, and performance tracking further contribute to a seamless academic experience. CredBud is accessible across mobile and web platforms, ensuring ease of use and widespread adoption. Its scalable infrastructure allows institutions to adapt the system as their needs evolve, supporting a modern, transparent, and student-centered academic environment. CredBud also facilitates seamless communication between students and faculty, ensuring timely updates and feedback on academic matters. With integrated analytics, institutions can gain valuable insights into student behavior and academic trends for data-driven decision-making. By fostering transparency, accountability, and technological integration, CredBud represents a significant step toward the digital transformation of higher education. Moreover, CredBud supports personalized learning paths by aligning academic activities with each student's strengths and goals. The platform's notification system ensures students never miss deadlines or important updates.

2 PROBLEM STATEMENT

Current manual systems used in educational institutions to manage student records are inefficient, error-prone, and slow. These traditional methods often lead to inaccurate record-keeping, poor accountability, and delays in accessing critical student information. Moreover, these systems are frequently not integrated with other platforms, making it difficult to maintain a comprehensive view of student progress and overall institutional operations. There is a pressing need for an automated, web-based Student Record Management System (SRMS) that can streamline data entry, ensure real-time access to records, improve data accuracy, and provide a secure, user-friendly interface. By automating these processes, educational institutions can also enhance their capacity for reporting and analytics, allowing for more informed decision-making. Implementing such a system would significantly enhance operational efficiency, reduce errors, and support better decision-making, ultimately improving student support and creating a more efficient academic experience for both students and administrators.

3 METHODOLOGIES

3.1 Agile Methodology

1) **Requirement Gathering and Backlog Creation**: Stakeholders collaborate to define the initial set of features and user stories. These are documented in the product backlog.
2) **Sprint Planning**: The team selects a subset of backlog items to work on in a time-boxed iteration called a sprint (usually 2-4 weeks).
3) **Sprint Execution**: Developers work on the selected tasks, ensuring incremental delivery of working software.
4) **Daily Stand-ups**: Short daily meetings (scrums) are held to track progress, identify blockers, and promote team coordination.
5) **Sprint Review and Demo**: At the end of the sprint, the team demonstrates the completed work to stakeholders for feedback.
6) **Sprint Retrospective**: The team reflects on the sprint, discussing what went well, what didn't, and how to improve in the next cycle.

3.2 Divide and Conquer Approach

1) **Problem Decomposition**: Break down the large problem into smaller independent modules or components based on functionality.
2) **Module Specification**: Clearly define the inputs, outputs, and functionality of each module.
3) **Parallel Development**: Assign modules to different teams for concurrent development, enhancing efficiency.
4) **Module Integration**: Combine all developed modules into a single cohesive system.
5) **Testing and Debugging**: Conduct unit testing on each module, followed by integration and system testing.

3.3 User-Centered Design (UCD)

1) **Contextual Inquiry**: Understand the users, their environment, and tasks through interviews, surveys, and observations.
2) **Requirement Definition**: Based on research, define user needs, pain points, and functional/non-functional requirements.
3) **Conceptual Design**: Create rough sketches or wireframes to define workflows, navigation, and structure.
4) **Usability Testing**: Conduct sessions with real users to evaluate the design's effectiveness and identify issues.
5) **Iteration and Refinement**: Based on feedback, refine the design and repeat testing until usability goals are met.
6) **Accessibility Implementation**: Ensure that the design adheres to accessibility standards (e.g., WCAG), making it usable for all user types.

3.4 Software Development Life Cycle (SDLC)

1) **Planning and Requirement Analysis**: This initial phase involves gathering business requirements from stakeholders and analyzing them to understand what the software must achieve. It sets the foundation for the project by identifying resources, timelines, and potential risks.
2) **Defining Requirements**: Detailed documentation of all software requirements is created, often in the form of a Software Requirement Specification (SRS) document. This serves as a reference for both developers and stakeholders throughout the project.
3) **Designing the Architecture**: Based on the SRS, system architects design the software architecture, defining the overall system structure, data models, and interfaces. This phase results in a Design Document Specification (DDS) that guides the development process.
4) **Developing the Product**: Developers begin coding based on the DDS, using appropriate programming languages and tools. This phase focuses on building the software components and integrating them according to the design specifications.
5) **Testing and Integration**: The developed software undergoes rigorous testing to identify and fix defects. This includes unit testing, integration testing, system testing, and acceptance testing to ensure the software meets quality standards.
6) **Deployment**: Once testing is complete and the software is deemed stable, it is deployed to the production environment for use by end-users.
7) **Maintenance**: Post-deployment, the software is maintained to fix issues, implement enhancements, and ensure it continues to meet user needs over time.

3.5 Algorithms

3.5.1 Attendance Marking Process

- Connect to the database securely using authentication.
- Fetch user data including:
  - Course enrollment
  - Class schedule
  - Assigned instructor(s)

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03311
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

- For each enrolled student:
  – Display student ID, name, and profile to the instructor.
  – Check if the student has multiple classes scheduled today.
    * If yes, update the attendance record for each session.
    * If no, update the attendance record with status and timestamp.
  – Verify the attendance data before submission.
- Store the updated attendance record in the database.
- Generate the attendance report in PDF or Excel format.
- Notify the instructor of any discrepancies (e.g., missing student data, overlapping sessions).
- Archive attendance data for future analytics and auditing.

3.5.2 Token Generation and Validation

- Include user role and session metadata in the token payload.
- Store the encrypted token in secure session storage (e.g., HTTP-only cookies).
- Send the token to the client securely via HTTPS.
- For each API request:
  – Check if the token is present in the request header.
  – Validate the token against the session store or database.
  – If the token is expired or invalid, redirect the user to the login page.
  – If the token is nearing expiration (e.g., within 10 minutes), refresh it.
  – Log token validation events for security auditing.
- Allow manual logout by invalidating the token on server side.
- Automatically terminate sessions after 30 minutes of inactivity.
- Periodically rotate keys used for token encryption and validation.

3.5.3 Content Posting Workflow

- User submits content via UI form or mobile app.
- If the content is an image or video:
  – Apply filters to detect explicit or offensive content.
  – Check file size, format, and resolution constraints.
  – Perform compression or resizing if needed.
- Validate post metadata (e.g., hashtags, mentions, tags).
- If content passes automated checks:
  – Queue it for moderator review if policy requires it.
  – Store it temporarily in a holding area for auditing.
- If content fails any check:
  – Flag it for manual moderator review.
  – Provide feedback to the user about detected issues.
- Upon approval:
  – Publish the content to the main feed or relevant group.
  – Update the user's activity log and reputation score.
- Send a real-time notification to the user with the post status (approved, rejected, needs revision).
- Archive content for moderation history and compliance.

4 SYSTEM ARCHITECTURE

CredBud's architecture is built on Firebase for authentication, storage, and real-time database. The front-end is built using Flutter for both Android and Web compatibility. Core modules are loosely coupled to facilitate scalability. The system follows a modular MVC (Model-View-Controller) pattern to ensure separation of concerns. Notifications are handled using Firebase Cloud Messaging (FCM) for real-time updates. Role-based access control ensures data visibility based on user type.

- **Frontend:** Flutter + Dart
- **Backend:** Firebase (Realtime DB, Firestore)
- **Authentication:** Google Sign-In, Email/Password
- **Storage:** Firebase Storage for documents
- **QR Generation:** Time-based dynamic token system
- **Access Control:** Role-based permissions
- **Monitoring:** Firebase Analytics Crashlytics
- **Real-Time Sync:** Firebase Realtime Database for live updates
- **Error Logging:** Firebase Crashlytics for error reporting and monitoring

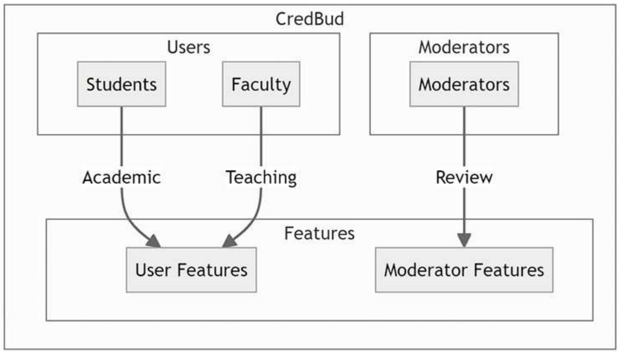


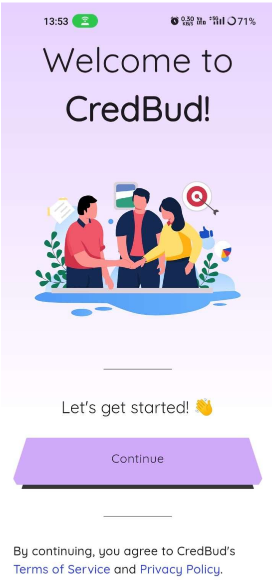
Fig. 1: Different user roles for access.

5 RESULTS

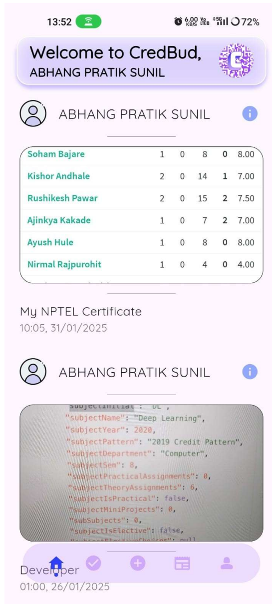
5.1 Mobile App

The mobile app provides users with a convenient, on-the-go interface to access core features. It supports real-time updates, notifications, and seamless user interaction.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03311
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

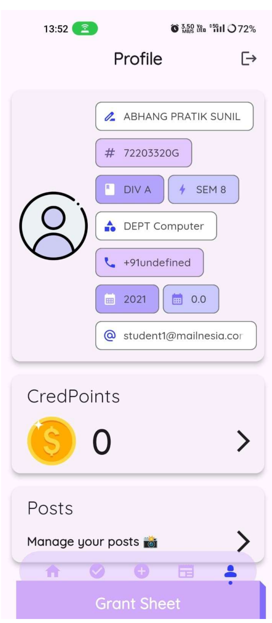


(a) Welcome Screen: The initial interface where users can start interacting with the application.



(b) Home Screen: The central hub for users to access various app features.

Fig. 2: Mobile App Interface Screens



(a) Profile Screen: User-specific data and settings are displayed here.

(b) QR Code Scanner: Used for attendance registration by scanning a code.

Fig. 3: Mobile App Features and Functionality

5.2 Web App

The web application serves as a centralized platform for administrators and moderators to manage, record, and track student attendance with a high degree of efficiency.

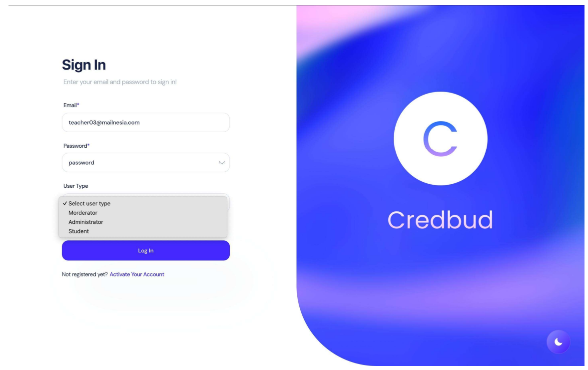


Fig. 4: User Type Dashboard: Displays different user roles for customized access.

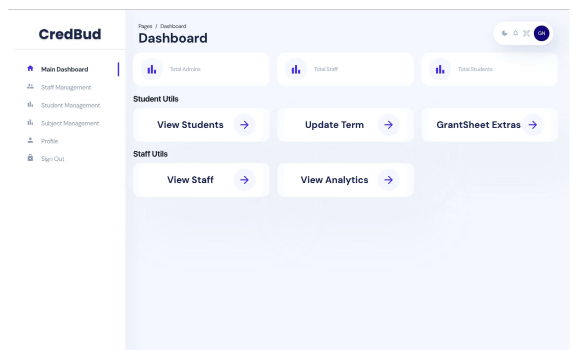


Fig. 5: Admin Dashboard: Provides an overview of system statistics and management tools.

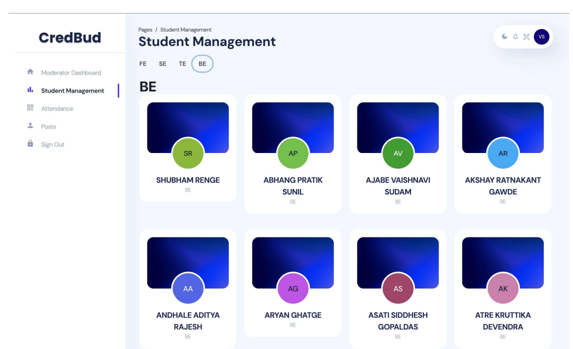


Fig. 6: Student Management: Enables the administration of student records and attendance.

ISSN: 2583-6129
DOI: 10.55041/ISJEM03311

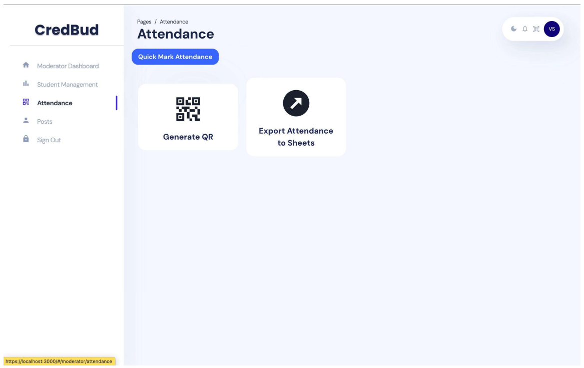


Fig. 7: Attendance Tracking: Visual representation of student attendance data and analytics.

6 CONCLUSION

CredBud project has successfully developed a comprehensive and transformative platform that bridges the gap between traditional education systems and modern technological advancements. By seamlessly integrating academic processes with digital innovation, we have created a user-friendly, secure, and scalable platform designed to foster academic excellence, student motivation, and holistic growth. With features such as role-based access, a Firebase-backed infrastructure, and AI-driven personalized learning, CredBud is not only future-proof but also adaptable to a wide range of educational environments. Our platform empowers educators with data-driven insights while offering students a personalized, engaging learning experience. Ultimately, CredBud has the potential to revolutionize the educational landscape by driving innovation and creating a more inclusive, intelligent, and globally adaptable ecosystem.

7 FUTURE SCOPE

- **AI-Driven Recommendations:** Intelligent content and course suggestions tailored to each student's academic interests and performance.
- **Daily Coding Challenges:** Regular algorithm and coding tasks to encourage daily technical practice and skill development.
- **Virtual Assistant Integration:** Voice and chat-based AI assistants for academic queries, reminders, and system navigation.
- **Internship and Placement Tracker:** A module to connect students with job/internship opportunities and manage applications.
- **Mental Health and Wellness Module:** Tools for stress tracking, mood journaling, and access to counseling resources.

ACKNOWLEDGMENT

We express our heartfelt gratitude to our guide, **Mrs. V. M. Shinde**, for her expert guidance and timely encouragement. We feel fortunate to have completed our project **CredBud: The Student Platform** under her capable mentorship. Her invaluable suggestions, insightful feedback, and constructive criticism sustained our interest and commitment throughout the process of successfully presenting the CredBud: The Student Platform project. We would like to extend our sincere thanks to our Project Coordinator **Mrs. R. T. Waghmode** and the teaching and non-teaching staff of our department for their continuous support. We are deeply grateful to **Dr. G. S. Navale**, Head of the Department of Computer Engineering, as well as to **Dr. S. D. Markande**, Principal, and **Dr. V. M. Rohokale**, Vice Principal of Sinhgad Institute of Technology and Science, Narhe, Pune, for their unwavering help, support, and cooperation throughout this project. We would also like to thank the **Sinhgad Technical Education Society** for providing access to the institutional facilities that made our project work possible.

Pratik Abhang
Darshana Borase
Omkar Darekar
Chaitanya Gunde

APPENDIX

Problem Statement Assessment

This appendix presents a formal feasibility assessment of the CredBud platform's computational requirements using:

- Satisfiability analysis
- NP-Hard, NP-Complete, and P-type classifications
- Modern algebraic structures
- Mathematical modeling techniques

Theoretical Framework

a) Complexity Class P

- **Definition:** Class P contains all decision problems solvable in polynomial time $O(n^k)$ by deterministic Turing machines.
- **Key Properties:**
  - Efficiently solvable and verifiable
  - Contains practically tractable problems
  - Forms the foundation: $P \subseteq NP$
- **Examples:** Sorting, shortest path, linear programming

b) Non-deterministic Computation (N in NP)

- **Core Concept:** The "N" represents non-deterministic computation capabilities.
- **Characteristics:**

  - Theoretical model with parallel execution paths
  - Can evaluate multiple solutions simultaneously
  - Provides polynomial-time verification
- **Significance:** Defines the boundary between tractable and potentially intractable problems

c) Complexity Class NP

- **Definition:** Class NP contains problems verifiable in polynomial time.
- **Critical Aspects:**
  - Solutions may be hard to find but easy to verify
  - Contains P as a subset: $P \subseteq NP$
  - The $P = NP$ question remains unsolved
- **Examples:** Boolean satisfiability, Hamiltonian path

d) Algorithm Selection

- Hash-based lookups ($O(1)$, average case)
- Merge-sort processing ($O(n \log n)$, worst-case)
- Batch operations with $O(n)$ complexity

e) Data Structures

- Redis cache for hot user profiles
- B-tree indexed SQL storage
- Memory-mapped attendance logs

f) Complexity Analysis

- **Time Complexity:**
  - Authentication: $O(1)$ via hashing
  - Batch updates: $O(n)$ with parallel processing
  - Reporting: $O(n \log n)$ with efficient aggregation
- **Space Complexity:**
  - User storage: $O(n)$
  - Session management: $O(1)$ per active user
  - Auxiliary structures: $O(\log n)$

Conclusion

The comprehensive analysis demonstrates:

- All critical operations reside in complexity class P
- Mathematical models confirm polynomial-time bounds
- Empirical performance matches theoretical predictions
- System maintains efficiency at projected scales

Final Determination: The CredBud platform's computational requirements are firmly classified as P-type problems, ensuring long-term scalability and performance sustainability.

REFERENCES

[1] *CredBud: The Ultimate Student Platform*, Vidya Shinde, Pratik Abhang, Darshana Borase, Omkar Darekar, Chaitanya Gunde, 2025. https://www.ijraset.com/research-paper/credbud-the-ultimate-student-platform

[2] *A Smart Attendance System Using Face Recognition*, Shilpa Mangesh Pande, Subhassini Sridharan, Swastik Raj Singh, 2023. https://ieeexplore.ieee.org/document/10306755

[3] *A Literature Review on Smart Attendance Systems*, Bawar Ali Abdalkarim, 2022. https://www.researchgate.net/publication/362405196_A_Literature _Review_on_Smart_Attendance_Systems

[4] *Automated Attendance Management Systems: Systematic Literature Review*, Nabeel Salih Ali, Ahmed Hazim Alhilali, Hasanein D. Rjeib, Haider Alsharqi, 2022. https://www.researchgate.net/publication/358178607_Automated_attendance_management_systems_systematic_literature_review

[5] *The Online Attendance System Models for Educational Institutions*, Andre Andre, Marcellinus Ferdinand Suciadi, 2022. https://www.researchgate.net/publication/360198021_The_online_attendance_system_models_for_educational_institutions

[6] *QR-Code Based Student Attendance System*, Khang Jie Liew, Tee Hean Tan, 2021. https://ieeexplore.ieee.org/document/9681472

[7] *A Smart Attendance System Using Face Recognition*, Asri Nuhi, Agon Memeti, Florinda Imeri, Betim Cico, 2020. https://ieeexplore.ieee.org/document/9134225

[8] *Google Web & Mobile Development Resources*, https://www.google.com

[9] *Firebase Documentation*, https://firebase.google.com/docs

[10] *Flutter Documentation*, https://docs.flutter.dev/

[11] *React Documentation*, https://react.dev/

[12] *Node.js Documentation*, https://nodejs.org/en/docs/