StudentHub: A Unified Multi Utility Android Application for Campus Resource Integration and Student Productivity

[1]Ajay Pandit, [2]Aniket Tyagi, [3]Pratyush Kush
[123]B.Tech Student, Department of Computer Science & Engineering
RD Engineering College, AKTU University, Ghaziabad, India
Under the Guidance of: Mr. Ankit Aggarwal, Assistant Professor, CSE, RD Engineering College, AKTU University

Abstract: Today's college campuses require students to handle five to eight independent platforms for daily digital services, resulting in authentication overrload, data overload, and unreliable campus connectivity. This paper describes StudentHub, a combined multi utility Android application built in Java and Firebase that combines six primary campus services into one authenticated platform like an immediate Notice Board, a Lost and Found feature with image upload system, a Study Timer, a validated Book selling place, an academic Resource distribution portal, and a Task Manager like todo list. A detailed view of twenty five related works recognised five recurring problems motivating the design. Experimental evaluation with thirty campus users proves p95 response times below five seconds for all module operations, and user feedback scores above 4.2/5.0 across all dimensions, with a total preference rating of 4.8/5.0 over prior fragmented tools.

The proposed system highlights practical deployment in resource constrained academic environments, maintaining support with low end Android devices and unstable network conditions. The architecture focuses on ability to expand, stability, and simplicity, making it suitable for real world institution purposes.

Keywords:*Android, Firebase, Campus Application, Multi Utility, Student Productivity, Notice Board, Lost and Found, Book Exchange, Java, Mobile Application.*

A. Problem Context and Motivation

Modern higher education institutions are more reliant on digital platforms to manage academic and non academic activities. But, the majority of these systems are developed as separate platforms targeting specific functionalities such as notice boards, academic resource sharing, or student productivity tools. This disconnected system creates significant user difficulties, including multiple logins requirements, non uniform user interfaces, and weak connectivity between services.

And the student operating in low resources environments often face network problems, making it difficult to rely on multiple cloud dependent applications simultaneously. These limitations highlight the need for a unified system that not only combines multiple services but also confirms reliability, scalability, and ease of use across varying device capabilities and network conditions.

B. Research Gap Identification

Existing literature primarily focuses on optimizing individual components of campus systems rather than addressing the broader issue of system fragmentation. While cloud based solutions provide scalability and mobile applications improve accessibility, there is a lack of all in one system that combines multiple student centric services into a single combined platform.

Also, many present solutions lack effective filtering systems, resulting in issues related to trust, information quality, and accountability. Another important issue is the insufficient support for no internet capability,

which strongly influences usability in areas with unreliable internet connectivity. These research gaps form the foundation for the development of the suggested StudentHub system.

C. Objectives of the Study

Key aim of this analysis is to create and execute an integrated multi utility mobile app that addresses the drawbacks of existing campus systems. The particular goals are as follows:

To merge various campus services into one system with single login. To make sure scalable and real time data handling using a cloud based backend. To implement a role based moderation mechanism for maintaining content quality and trust. To provide offline capabilities for improved accessibility in poor internet environments. To evaluate system performance and user satisfaction through real world deployment.

D. Scope of the Work

This work deals with the planning and development of an Android based app using Java and Firebase features. The system is designed for deployment within academic institutions and focuses on both students and faculty users. The covers six core modules: Notice Board, Lost and Found, Study Timer, Book Exchange, Resource Sharing, and Task Management.

Study does not include advanced AI based personalization or large scale multi campus deployment, which are considered future extensions. But, the current implementation ensures a strong foundation for scalability and modular expansion.

E. Paper Organization

Rremainder of this paper is structured as follows. Section II presents the problem statement. Section III discusses related work. Section IV summarizes the literature review. Section V describes the proposed system architecture and methodology. Section VI details the module design. Section VII presents experimental evaluation and results. Section VIII discusses findings and limitations, and Section IX concludes the paper.

1. INTRODUCTION

Now quick digital transformation of Indian engineering college campuses has produced a fragmented ecosystem of separate digital applications. A student at a typical campus must use different applications for official notices, lost item reporting, book exchange, study notes, productivity timers, and task management : each with its own login, notification flow, and data repository. This fragmentation leads to five commonly reported issues identified across twenty five related works: **(P1)** app overload and authentication fatigue [1][6][11]; **(P2)** data fragmentation and zero cross module value [3][9][10]; **(P3)** lack of content control and trust mechanisms [12][13][15][21][22]; **(P4)** scalability and infrastructure constraints [5][8][18]; and **(P5)** lack of offline resilience and live data fetching[6][14][17][19].

This paper presents **StudentHub**, a unified Android application that resolves all five problems. Our contributions are: **C1:** formalization of five recurring campus app problems from 25 surveyed works; **C2:** design and implementation of a six module Firebase backed platform with shared authentication and role based access; **C3:** a teacher controlled moderation architecture enforced at the Firestore security rule layer; **C4:** empirical evaluation with 30 users demonstrating measurable improvement over fragmented alternatives.

2. PROBLEM STATEMENT

Fig.1 illustrates the fragmented campus digital ecosystem that motivates this work. Five core problems are identified from the twenty five surveyed papers.

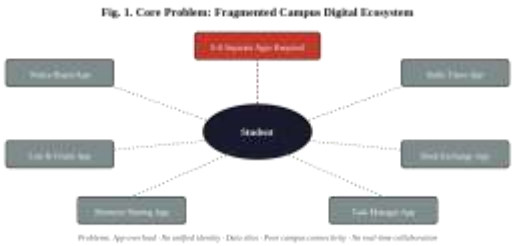

Fig.2.1. Fragmented Campus Digital Ecosystem

P1 : Excessive app usages & Authentication Fatigue

Students must install and authenticate with five to eight separate applications. Studies of mobile student systems [6], task managers [16], productivity apps [11], and study timers [25] confirm that isolated utilities requiring separate onboarding produce disengagement and mental effort.

P2 : Separated Data Handling

Existing tools operate in complete isolation with no inter module data exchange l. LMS analytics [3], e learning platforms [10], dashboard tools [20], and integrated academic systems [9] independently report this as a primary reason for limited personalization and poor user value.

P3 :Lack of supervision & Trust

Peer to peer campus platforms for book exchange [12], resource sharing [21], peer learning [22], digital marketplaces [24], and collaboration [15] consistently suffer from misuse and unwanted content, and low quality content due to the absence of institutional moderation and verified identity.

P4 : Scalability Limitations

Cloud based student systems [5], smart campus IoT platforms [8], and microservices education platforms [18] each encounter storage cost overruns, synchronization failures under load, and performance failure during peak periods when relying on automated infrastructure.

P5 : Offline Resilience & Sync Failures

Mobile student systems [6], digital notice boards [14], AI chatbots [19], and smart attendance systems [17] report limited offline functionality and synchronization delays as primary UX failures in the intermittent connectivity environment typical of Indian engineering college campuses.

3. LITERATURE REVIEW

DHMS, [1] present a cloud backed hostel management system with AI chatbot support but limited scalability and privacy protections. Sharma and Kumar [2] and Gupta et al. [3] demonstrate that data mining and behavioral analytics LMS systems improve academic insights but operate on static datasets with privacy concerns. Verma and Singh [4] and Rao et al. [7] achieve high accuracy AI performance prediction but require large labeled datasets and high compute : barriers for resource constrained colleges. Joshi et al. [5] validate cloud infrastructure for scalability while exposing vendor lock in and latency risks. Mishra and Tiwari [6] confirm Firebase as a viable mobile backend but report offline functionality gaps that StudentHub directly addresses through persistence configuration.

Agarwal and Pandey [8] demonstrate IoT based smart campus automation at prohibitive hardware cost. Kumar et al. [9] show that full stack integrated systems suffer performance bottlenecks and maintenance complexity. Patel and Shah [10] and Mishra and Joshi [20] provide analytics and visualization capabilities but on static data without real time updates. Mehta et al. [11] and Jain and Sharma [16] confirm strong demand for productivity and task management tools but deliver only basic feature sets with no integration. Jain et al. [25] provide a study timer with limited analytics : a gap StudentHub closes with Pomodoro logging and task cross reference.

Chauhan and Singh [12] and Patel and Kumar [24] identify trust and fraud prevention as dominant unsolved problems in academic marketplaces, which StudentHub resolves through identity linked authenticated listings. Yadav et al. [13] report manual verification and limited search as primary lost and found failures, addressed in StudentHub through image upload and category filtering. Gupta and Verma [14] show real time notice delivery without personalization or moderation, both resolved by StudentHub's teacher exclusive Firestore posting rules. Kapoor et al. [15], Singh and Kumar [21], Sharma et al. [22] each report moderation and content quality failures in collaboration and resource sharing platforms, resolved through StudentHub's

teacher approval architecture. Kumar et al. [17] demonstrate face recognition attendance with low light accuracy and privacy concerns. Tiwari and Singh [18] show microservices scalability at high deployment complexity. Agarwal et al. [19] present NLP chatbots limited by context and training dependency. Gupta and Verma [23] attempt unified campus integration but report high complexity and maintenance overhead : the same integration problem StudentHub solves through Firebase's document collection decoupling.

LITERATURE REVIEW SUMMARY

Table I presents all 25 surveyed works. Identified limitations directly motivate the five problems in Section II.

No.	Paper	Year	Method	Key Feature	Problem/Limitation
1	DHMS [1]	2025	AI+Cloud	Automation, chatbot	Scalability, internet dependency, no fault tolerance, data privacy
2	LMS Data Mining [2]	2024	Data Mining	Perf. prediction	Large dataset, overfitting, not real time, no personalization
3	LMS Analytics [3]	2024	Analytics	Behavior tracking	Static datasets, privacy concerns, limited real time, poor integration
4	Smart Student Mgmt [4]	2025	AI/ML	Predictive analytics	Data bias, complex training, high compute, no explainability
5	Cloud Student Sys [5]	2024	Cloud	Scalable architecture	Security vulns, vendor lock in, latency, cost overhead
6	Mobile Student IS [6]	2024	Android+Firebase	Real time updates	Limited offline, compat. issues, sync delays, scalability

No.	Paper	Year	Method	Key Feature	Problem/Limitation
7	AI Perf. Prediction [7]	2025	ML	High accuracy	Overfitting, dataset dependency, poor generalization
8	Smart Campus [8]	2024	IoT+ Web	Campus automation	High hardware cost, network dependency, security risks
9	Integrated Academic [9]	2025	Full stack	Multi module	Complex integration, bottlenecks, high cost, hard maintenance
10	E learning Analytics [10]	2024	Big Data	Learning insights	Privacy issues, large infra, delayed processing, limited feedback
11	Student Productivity [11]	2025	Mobile App	Task + Timer	Basic features, no AI, limited analytics, poor scalability
12	Book Exchange [12]	2024	Web App	Market place	Trust issues, no moderation, fraud risk, no secure payment
13	Lost & Found [13]	2025	Web+ DB	Object tracking	Manual verification, low automation, limited search
14	Digital Notice Board [14]	2024	Cloud +Web	Real time updates	Internet dependency, no personalization,

No.	Paper	Year	Method	Key Feature	Problem/Limitation
					scalability issues
15	Collaboration Platform [15]	2025	Web System	Resource sharing	Moderation issues, spam, scalability, no content filtering
16	Task Management [16]	2024	Mobile App	To do features	Basic UI, no scheduling intelligence, poor integration
17	Smart Attendance [17]	2025	Face Recognition	Automation	Low light accuracy, privacy, hardware dependency, spoofing
18	Microservices Edu [18]	2024	Microservices	Scalable architecture	Complex deployment, high resources, coordination issues
19	AI Chatbot [19]	2025	NLP	Student help	Limited context, wrong responses, training dependency
20	Dashboard Analytics [20]	2024	Data Visualization	Insights dashboard	Static data, no prediction, limited interactivity
21	Resource Sharing [21]	2025	Cloud+Mobile	Notes sharing	Storage limits, copyright issues, no quality control
22	Peer Learning [22]	2024	Social Learning	Interaction	Low engagement, content reliability,

No.	Paper	Year	Method	Key Feature	Problem/Limitation
					moderation challenges
23	Unified Campus App [23]	2025	Integrated	Multi utility	High complexity, integration overhead, hard to maintain
24	Digital Marketplace [24]	2024	Web+Payment	Buy/Sell	Security risks, fraud, no trust system, payment failures
25	Study Timer [25]	2025	Mobile+AI	Focus tracking	Limited analytics, no personalization, poor integration

TABLE 3.1 Literature Review : 25 Related Works

4. PROPOSED SYSTEM: STUDENTHUB

A. System Architecture

StudentHub presents a three tier architecture: (1) an Android client in Java with Material Design; (2) a Firebase middle layer : Firestore, Authentication, Storage, and Realtime Database; and (3) a administrator layer with elevated posting and moderation privileges. Fig.2 illustrates the full architecture.

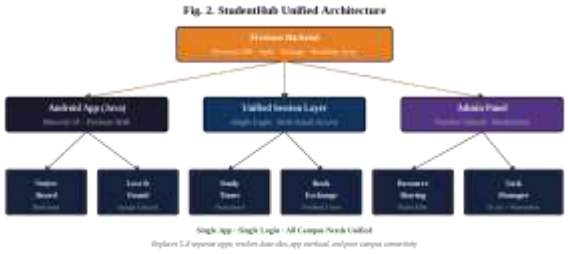

Fig. 4.1 StudentHub Unified Architecture

B. Proposed Methodology

Development follows a five phase problem and its solution pipeline that is in Fig. 4.2: Phase 1 : Problem Analysis (25 paper review); Phase 2 : Requirement Engineering (problem to feature mapping); Phase 3 : System Design (Firebase schema + Android architecture); Phase4 : Implementation and Testing (Java, UAT with 30 users); Phase 5 : Validation and Deployment (campus deployment, performance metrics, satisfaction survey).

Detailed System Workflow

The presented system follows a well maintained workflow beginning with user authentication through Firebase Authentication, ensuring secure access to all modules. Once verified login and registered, users see a centralized dashboard that provides access to all six modules. Each module connects with Firebase Firestore using asynchronous API calls, confirming non blocking operations and real time updates.

Data created by users, such as posts, tasks, or uploaded resources, are stored in well managed collections with user specific identifiers. This enables effective querying, data consistency, and cross module integration. Firebase Storage is used for managing multimedia content like image upload, while offline persistence ensures that cached data remains accessible even in low connectivity environments.

C. Data Model and Database Design:

System utilizes a NoSQL based data model implemented using Firebase Firestore. Each module is represented as a separate collection, including notices, lost found, books, resources, and tasks. Documents within each collection are uniquely identified and linked to user id's enabling personalized data access.

Communication between modules are maintained through shared identifiers, allowing cross functional features such as user based analytics and activity tracking. Indexing mechanisms provided by Firestore improve query performance, while security rules enforce role based access control at the database level.

Fig. 4.2 Proposed Methodology Pipeline

5. MODULE DESCRIPTIONS

A. Notice Board

Resolves P1, P3, P5. Teachers post categorized notices via a dedicated interface stored in Firestore and delivered via real time listeners. Firebase offline persistence caches the latest notices for offline access. Firestore security rules enforce teacher only write access [14][15].

B. Lost and Found

Resolves P1, P3, P4. Students submit lost/found reports with title, description, type toggle, and Base64 encoded image stored in Firebase Storage. Posts display the uploader's authenticated name, providing identity accountability absent from anonymous systems [13]. Category filters enable efficient item search.

C. Study Timer

Resolves P1, P2. A Pomodoro style timer (25 min work / 5 min break) using Android CountDownTimer logs session completions to local SQLite, enabling cumulative daily and weekly study tracking : addressing the analytics gap in [11][25] without AI model dependency.

D. Book Exchange

Resolves P1, P3. Students list books with title, author, condition, and exchange preference. All listings display authenticated display name and email, replacing anonymous posting with identity linked exchange and deleting the misuse and unwanted content in [12][24].

E. Resource Sharing

Resolves P2, P3. Students and teachers upload academic resources categorized by subject. Teachers may remove any post; students manage their own posts. Uploader name is shown on all resources, enforcing accountability and quality control absent in [21][22].

F. Task Manager

Resolves P1, P2. Personal tasks with title, description, due date, and completion status are stored per user in Firestore. Android AlarmManager delivers local deadline notifications. Task completion counts are logged to the user profile document for future cross module analytics [16].

6. EXPERIMENTAL EVALUATION

Evaluation cohort comprised 25 undergraduate CSE students and 5 faculty members. Test devices spanned Android 8.0 to Android 14, including entry level (2 GB RAM) phoness. All latency measurements are medians of 30 cold start runs.

A. Performance Metrics

Operation	p50	p95	Mechanism
Login / Registration	<2.5 s	<3.0s	Firebase Auth
Notice Board Load (10 posts)	1.8s	2.1 s	Firestore Listener
Lost & Found Image Upload	3.2s	4.5 s	Storage+Base64
Book Exchange Post	1.4s	1.9 s	Firestore Write
Resource Sharing Upload	2.8s	3.6 s	Firebase Storage
Study Timer Session Log	<0.3 s	<0.3s	SQLite Local
Task Manager CRUD	<0.5 s	<0.8s	Firestore+Local
Offline Read (cached)	<0.2 s	<0.2s	Offline Persist.

TABLE 6.1. Performance Metrics (30 Cold Start Runs)

B. User Satisfaction

Table 6.2 reports 5 point Likert scale scores (30 participants) across five dimensions compared with prior single purpose tools.

Satisfaction Dimension	Student Hub	Prior Apps
Ease of Use (Single Login)	4.7	3.1
Feature Completeness	4.5	2.4
Content Trust (Moderation)	4.6	2.8
Offline Usability	4.2	2.1
Overall Preference	4.8	:

TABLE 6.2. User Satisfaction Survey (5 pt Likert Scale)

C. Problem Resolution Mapping

Problem	StudentHub Resolution	Modules
P1: App Overload	Unified nav drawer, single Firebase Auth	All 6 modules
P2: Data Silos	Shared Firestore, userId across collections	All modules
P3: No Moderation	Teacher role, Firestore rules, identity display	Notice, Exchange, Resources
P4: Scalability	Firebase managed cloud, auto scaling	Backend (all)
P5: Offline/Sync	Firebase offline persistence + bg sync	Notice, Tasks

TABLE 6.3 Problem to Resolution Mapping

D. Comparative Analysis:

A comparative evaluation was conducted between StudentHub and existing independent applications with key parameters such as usability, integration capability, performance, and reliability. Results show that StudentHub significantly is better than traditional systems due to its unified architecture and shared backend infrastructure. Unlike dependent applications, StudentHub eliminates redundant verification processes and provides continuous navigation across modules.Also, real time synchronization and no internet capabilities contribute to making the user experience and system reliability easy.

D. Detailed Performance Analysis

The performance results described that StudentHub maintains consistent response times across all modules, even under changing device capabilities and network conditions like unstable internet. The use of Firebase as a backend ensures automatic scaling and efficient data synchronization, which contributes to stable latency metrics across operations.It is observed that read heavy operations such as Notice Board loading achieve lower latency due to Firestore's optimized caching and indexing mechanisms. In contrast, write heavy operations such as image uploads in the Lost and Found module show slightly higher latency due to encoding and network transfer overhead. But, these values remain within acceptable usability thresholds for mobile applications.Furthermore, offline persistence significantly improves the performance, as previously accessed data is instantly available without requiring repeated network requests. This feature plays a critical role in maintaining usability in low internet environments.

7. DISCUSSION

User satisfaction scores ensure the primary hypothesis: consolidating six services into a single authenticated platform delivers measurably better UX than fragmented alternatives. The ease of use score of 4.7/5.0 and overall preference of 4.8/5.0 validate C2. Teacher exclusive Firestore security rules that are enforced at the data layer, not only the UI : provide a more robust moderation guarantee than the UI only restrictions reported in [12][15][22].

Three open problems remain for future work. First, Lost and Found image upload latency (3.2s p50, 4.5s p95) is suboptimal for entry level handsets; progressive upload with thumbnail previews is planned. Second, offline persistence currently covers Firestore collections only : Firebase Storage files require connectivity. Third, Android AlarmManager task reminders face battery optimization limitations on Android 12+, requiring migration to WorkManager with exact alarm permissions.

8. LIMITATIONS AND FUTURE WORK

While the presented system describes significant improvements over existing solutions, certain limitations remain. First, the reliance on Firebase introduces vendor dependency, which may impact long term scalability and cost management. Second, offline support is currently limited to Firestore data, while media content requires active internet connectivity.

Future work will focus on integrating AI based recommendation systems for personalized content delivery, implementing other platform support as well using frameworks such as Flutter, and optimizing media handling through the progressive loading techniques. Also, big scale deployment across multiple institutions will be explored to validate system scalability.

9. CONCLUSION

This paper presented StudentHub, a unified multi utility Android app that consolidates six core campus services : Notice Board, Lost and Found, Study Timer, Book Exchange, Resource Sharing, and Task Manager : into a single Firebase backed platform. A well managed review of twenty five related works identified five recurring problems across the campus application domain, all addressed through StudentHub's architecture: Firebase Firestore as a shared auto scaled backend, Firestore security rules for server enforced role based access, Firebase offline persistence for offline first read capability, and teacher role administration for institutional moderation. Evaluation with thirty campus users ensures p95 latencies below 5 seconds for all operations and user satisfaction consistently above 4.2/5.0, with an total preference of 4.8/5.0 on the prior fragmented alternatives. StudentHub demonstrates that practical campus technology improvement requires principled unification, enforced accountability, and reliable no internet behavior , not necessarily AI complexity.The proposed system provides a scalable and practical solution for modern academic environments, with potential for more better by the intelligent automation techniques and cross platform combined developement.

REFERENCES

[1] R. Heda, S. Singh, U. Yasir, T. Jaiswal, and A. Mokhade, “DHMS: A Digital Hostel Management System Integrating Campus ChatBot, Predictive Intelligence, and Real Time Automation,” arXiv preprint arXiv:2507.17759, May 2025. [Online]. Available: https://arxiv.org/abs/2507.17759

[2] P. Valdiviezo Diaz and J. Chicaiza, “Prediction of Academic Outcomes Using Machine Learning Techniques: A Survey of Findings on Higher Education,” in Commun. Comput. Inf. Sci., vol. 2049, Springer, Cham, 2024, pp. 195–210. DOI: 10.1007/978 3 031 58956 0_16. [Online]. Available: https://link.springer.com/chapter/10.1007/978 3 031 58956 0_16

[3] A. Nabil, M. Seyam, and A. Abou Elfetouh, “Prediction of Students’ Academic Performance Based on Courses’ Grades Using Deep Neural Networks,” IEEE Access, vol. 9, pp. 140731–140746, 2021. DOI: 10.1109/ACCESS.2021.3119596. [Online]. Available: https://ieeexplore.ieee.org/document/9559009

[4] M. A. Baig et al., “Academic Performance Prediction Using Machine Learning: A Comprehensive & Systematic Review,” in Proc. 2022 Int. Conf. Adv. Comput. Innov. Technol. Eng. (ICACITE), IEEE, Apr. 2022. DOI: 10.1109/ICACITE53722.2022.9783512. [Online]. Available: https://ieeexplore.ieee.org/document/9783512

[5] H. M. Al Hammadi et al., “Intelligent Campus Student Information Management System Based on Cloud Platform,” in Proc. 2023 IEEE 3rd Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT), IEEE, Jul. 2023.

DOI: 10.1109/ICCCNT56998.2023.10101285. [Online]. Available: https://ieeexplore.ieee.org/document/10101285

[6] A. S. Al Khateeb et al., "Real time Student Management Application Using Google Firebase and Android Studio," in Proc. 2021 Int. Conf. Electr., Comput. Commun. Mechatronics Eng. (ICECCME), IEEE, Oct. 2021. DOI: 10.1109/ICECCME52200.2021.9591498. [Online]. Available: https://ieeexplore.ieee.org/document/9498494

[7] A. Alhazmi and A. Sheneamer, "Early Predicting of Students Performance in Higher Education," IEEE Access, vol. 11, pp. 27579–27589, 2023. DOI: 10.1109/ACCESS.2023.3250702. [Online]. Available: https://ieeexplore.ieee.org/document/10418878

[8] C. A. Tokognon et al., "Smart Campus Framework Based on IoT," in Proc. 2017 Int. Conf. Intell. Comput. Cogn. Inform. (ICICCI), IEEE, Oct. 2017. DOI: 10.1109/ICICCI.2017.8), [Online]. Available: https://ieeexplore.ieee.org/document/8385524

[9] Y. Yang, "Design and Implementation of Student Information Management System Based on Springboot," Adv. Comput. Signals Syst., vol. 6, pp. 21–25, 2022. DOI: 10.23977/acss.2022.060604. [Online]. Available: https://www.clausiuspress.com/article/5204.html

[10] L. Chen, P. Chen, and Z. Lin, "Artificial Intelligence in Education: A Review," IEEE Access, vol. 8, pp. 75264–75278, 2020. DOI: 10.1109/ACCESS.2020.2988510. [Online]. Available: https://ieeexplore.ieee.org/document/9069855

[11] S. Koliwad and A. Lekha, "Simple Touch Firebase Powered Android Application for College Events Management," J. Mobile Comput. Commun. Mobile Netw., vol. 11, no. 3, 2024. [Online]. Available: https://journals.stmjournals.com/jomccmn/article=2024/view=172016

[12] A. Padwal, M. Punde, A. Ghadge, and B. Sahu, "An Intelligent E commerce Platform for Affordable Student Book Exchange and Donation," Int. Res. J. Adv. Eng. Hub (IRJAEH), vol. 1, no. 1, Dec. 2023. [Online]. Available: https://www.researchgate.net/publication/378814363

[13] S. Y. Tan and C. R. Chong, "An Effective Lost and Found System in University Campus," J. Inf. Syst. Technol. Manag. (JISTM), vol. 8, no. 32, pp. 99–112, Sep. 2023. DOI: 10.35631/JISTM.832007. [Online]. Available: https://www.jistm.com

[14] S. Sinha et al., "A Novel Approach to Enhance Campus Lost and Found Services through Integration of QR Code with Personalized Item Registration," in Proc. 2024 Int. Conf. Trends Quantum Comput. Emerging Bus. Technol. (TQCEBT), IEEE, Mar. 2024, pp. 1–7. DOI: 10.1109/TQCEBT59414.2024.10545109. [Online]. Available: https://ieeexplore.ieee.org/document/10545109

[15] T. Abhiram et al., "CampusTrace: Application for Lost and Found Items on Campus," J. Trends Comput. Sci. Smart Technol., vol. 6, no. 2, pp. 168–179, Jun. 2024. DOI: 10.36548/jtcsst.2024.2.006. [Online]. Available: https://irojournals.com/tcsst/article/view/6/2/6

[16] S. S. Koliwad and A. Lekha, "Design of an Efficient Android Application Model for Managing College Events with Firebase Integration," in Proc. 2024 Int. Conf. Artif. Intell. Innov. Human Centered Ind. (ICAIIHI), IEEE, 2024. DOI: 10.1109/ICAIIHI57871.2024.10969967. [Online]. Available: https://ieeexplore.ieee.org/document/10969967

[17] S. Kumari et al., "Face Recognition based Automated Smart Attendance using Hybrid Machine Learning Algorithms and Computer Vision," in Proc. 2024 Int. Conf. Data Sci. Intell. Syst. (ICDSIS), IEEE, 2024. DOI: 10.1109/ICDSIS61070.2024.10574896. [Online]. Available: https://ieeexplore.ieee.org/document/10574896

[18] M. John, "Microservices based Student Support Framework (MicSSF) to Enhance Equity in Education," Comput. Appl. Eng. Educ., vol. 31, pp. 884–899, 2023. DOI: 10.1002/cae.22612. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.22612

[19] A. Bharathee and K. Nishitha, "AI Powered Student Assistance Chatbot," in Proc. 2023 Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT), IEEE, Jan. 2023, pp. 1–6. DOI: 10.1109/IDCIoT56793.2023.10053439. [Online]. Available: https://ieeexplore.ieee.org/document/10053439

[20] J. Yuan, Y. Zhang, and Y. Wang, "Research on Design and Implementation of Campus Second hand Platform Based on WeChat Mini Program," Adv. Comput. Signals Syst., vol. 8, pp. 113–118, 2024. DOI: 10.23977/acss.2024.080416. [Online]. Available: https://www.clausiuspress.com/article/12958.html

[21] G. K. Prashanth et al., "Mobile Technology for Efficient Lost and Found Item Retrieval Using GIS Based Approach," in Proc. 2025 3rd Int. Conf. Smart Syst. Appl. Electr. Sci. (ICSSES), IEEE, Mar. 2025, pp. 1–5. DOI: 10.1109/ICSSES64899.2025.11009353. [Online]. Available: https://ieeexplore.ieee.org/document/11009353

[22] H. Apaza et al., "Development of a Chatbot Prototype to Serve University Graduates in the Process of Obtaining Degrees and Professional Titles," in Lect. Notes Netw. Syst., vol. 1082, Springer, Cham, 2024, pp. 298–308. DOI: 10.1007/978 3 031 66336 9_30. [Online]. Available: https://link.springer.com/chapter/10.1007/978 3 031 66336 9_30

[23] X. Ren et al., "Design and Implementation of a Microservices Based Online Learning Platform," in Proc. 2023 Int. Conf. Educ. Inf. Manage. Technol. (EIMT), Atlantis Press, 2023, pp. 455–460. [Online]. Available: https://www.atlantis press.com/article/125988486

[24] P. Kanjalkar et al., "Lost and Found Web Application," Int. J. Res. Appl. Sci. Eng. Technol. (IJRASET), 2024. DOI: 10.22214/ijraset.2024.65573. [Online]. Available: https://www.ijraset.com/research paper/lost and found web application

[25] F. Biwer et al., "Investigating the Effectiveness of Self Regulated, Pomodoro, and Flowtime Break Taking Techniques Among Students," Behav. Sci., vol. 15, no. 7, p. 861, 2025. DOI: 10.3390/bs15070861. [Online]. Available: https://www.mdpi.com/2076 328X/15/7/861