International Journal of
Science, Architecture, Technology and Environment

University Android Based Attendance Application with Geofencing and Face Recognition

Vaibhavi Dongre[1*], Dr. Praveen Kumar Shrivastava[2], Akash Mishra[3]

[1]*MCA Student, School of Computer Applications, SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL,*

[2]*Professor, School of Computer Application, SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL*

[3]*Assistant Professor, School of Computer Application, SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL*

[*]*Corresponding author, dongrevaibhavi303@gmail.com*

DOI: https://doi.org/10.63680/ijsate0526138.103

Abstract

Taking attendance the old-fashioned way in schools and universities calling out names or signing paper lists just doesn't cut it anymore. It's slow, easy to mess up, and, let's be honest, "proxy" attendance happens way too often. So, this research tackles all those problems head-on with a Smart Attendance App for Android. It's got a lot going on: biometric face recognition, GPS geofencing, and even optical character recognition. Here's how it works. The app checks your location using GPS to make sure you're actually inside the campus, not just at home or somewhere else. For identity, it uses Google ML Kit to match your selfie with your profile photo no impersonation, no shortcuts. To double-check that you're in the right classroom, you scan a room number with OCR, so the app knows you're physically in that particular lecture. All this data syncs instantly with Firebase's NoSQL database, so students and administrators can see real-time attendance stats, plus cool charts showing performance trends. The app cuts down on manual mistakes and makes cheating pretty tough, so it's a solid fix for campuses looking to go digital.

Keywords: Android Application Development the project runs on Android, built with Java and XML in Android Studio, so students get a simple, mobile-friendly experience.

1. Introduction

Managing attendance isn't just a formality it's at the heart of keeping order and tracking student progress in schools. In the past, teachers called out names or passed around a sheet for everyone to sign, but let's be honest, those old-school methods have a lot of flaws. They're slow, prone to mistakes, and, let's face it, it's not hard for one student to cover for another who isn't actually there. Now, when you look around, nearly everyone has a smartphone, and technology's racing ahead with things like AI. That's where this research comes in introducing an Android-based Smart Attendance App. The idea is simple but powerful: use the smartphones people already have to create attendance records that are not just digital, but trustworthy. The app uses

geofencing through GPS, facial recognition, and even OCR (Optical Character Recognition). So, not only does it mark attendance instantly, but it also verifies the student is physically in the right place at the right time. No more cheating the system or wasting time just a smarter way to keep everyone honest and things running smoothly[1].

2. Motivation

1. Honestly, what pushed this research forward was seeing how unreliable and slow current attendance systems are. Three big problems stood out:
2. **Proxy Attendance:** First, proxy attendance is way too common In big classes, there's just no way for professors to confirm every student is actually there, so fake attendance slips through all the time.
3. **Administrative Burden** Second, taking attendance wastes a lot of time Teachers spend 10 or 15 minutes every class just marking names. That's time they could've used to actually teach.
4. **Data Accessibility:** Third, manual attendance records are a pain to manage. They're tough to track, hard to analyze, and not easy to share. Lots of students don't even know how often they've missed class until the semester ends—and then it's too late to do anything about it.

So, this project set out to fix all that. The goal: design a"Triple-Factor Authentication"system—one that checks location, identity, and context—to make attendance honest and effortless. This way, academic records stay solid, and both students and professors get clear, up-to-date info when they need it[2].

3. Objectives

Here's what this research and the app are aiming for:

1. **Automate Attendance Marking:** Get rid of old-school attendance registers and make it so students just tap to mark their presence—fast and easy.
2. **Use Multi-Factor Verification:** Make the system secure by checking if students are actually on campus with geofencing, verifying their identity with face recognition, and confirming the classroom with OCR.
3. **Keep Data Updated in Real Time:** Store and fetch attendance data instantly through a cloud backend (Firebase), so records are always current.
4. **Give Clear Data Visualization:** Show students their attendance patterns with built-in analytics tools like pie charts, so they can track their monthly performance easily.
5. **Boost Campus Security:** Build a strong system that blocks fake attendance and keeps a solid record for institutional audits[3].

4. Literature Survey

Attendance tracking has come a long way, shifting from old-fashioned roll calls to a range of automated systems. Here's a quick look at what's out there:

1. **RFID-Based Systems:** Researchers have tried using RFID tags for attendance. They're quick and convenient but pretty easy to trick. Students can just hand off extra tags so friends can check in for them.
2. **Fingerprint Biometrics:** To deal with that, some schools installed fingerprint scanners. These make sure the student is really present, but the hardware's expensive and tricky to keep up in every classroom. Plus, in big groups, people worry about everyone touching the same surface—it's not the cleanest system.

3. **Basic QR Code Systems**: Some teachers use a fixed QR code that everyone scans. The problem? Students just snap a photo of the code and share it around. So, friends who skipped class can still mark themselves as present from wherever they are.
4. **Standard GPS Tracking:** Then there are location-based apps. They check if a student's on campus, but they stop short. Being nearby doesn't mean it's you holding your phone or that you're sitting in class[4].

5. Comparative Analysis

The old ways—manual sign-ins, RFID, biometrics—they all run into trouble, usually with students gaming the system, high costs, or headaches when trying to scale up. That's where the University Attendance Application stands out. It uses a student's phone for both a location check and a photo, making attendance a lot more secure and efficient. No easy loopholes, lower costs, and simple to roll out for bigger classes[5.]

5.1 Comparison with Existing RFID- System

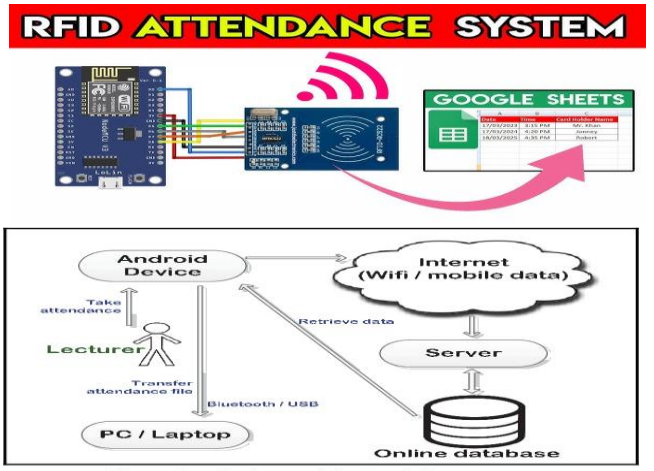


Figure 1. System architectural diagram

6. Problem Identification

This research really digs into what's wrong with both traditional and automated attendance systems. Here's where things break down when you're trying to keep academic records straight:

1. **Proxy Attendance:** Honestly, it's way too easy for students to cover for each other. Whether it's paper sign-ins, QR codes, or RFID cards, "buddy punching" keeps happening and it throws academic discipline out the window.
2. **Location Loopholes:** A lot of phone apps don't check where a student actually is. If they have access, they can mark themselves present from their dorm room or a coffee shop across town. All it takes sometimes is a shared code or login.
3. **Digital Identity Fraud:** Most apps just verify the account, not the person. Students can pass around IDs and passwords, so one student shows up for another. There's nothing—like facial recognition or fingerprints—to make sure the right student is signing in.
4. **No Context:** Even with GPS, being on campus doesn't mean you're in the right classroom, or even in class at all. Most systems ignore specific room info, so the data doesn't tell the whole story.
5. **Time Sink:** Taking attendance by hand eats up 15–20% of every class. That's a huge chunk of a one-hour lecture wasted shuffling papers instead of learning. It stacks up fast over a semester.

6. **Slow Feedback:** In a manual system, students can't check their attendance on the spot. They don't find out if they're behind until later, and that lack of transparency leads to confusion and arguments when the numbers aren't what they expected[6].

7. Methodology

1. **System Architecture:** The system's built with several layers. There's a mobile interface for users, a cloud backend, and machine learning modules all working together. Everything moves step by step—if you don't clear one stage, you don't get to the next. That way, only valid attendance entries get recorded.
2. **User Authentication Layer:** It all starts with a secure login. Users type in their unique Teacher or Student ID and password, just like you see in the act_login.xml design. Only people with the right credentials can get in, which keeps things secure from the very beginning.
3. **Geofencing and Location Verification:** To stop remote check-ins, the app uses GPS geofencing. As soon as someone tries to mark attendance, the app grabs their current location. If they're not within 300 meters of the campus, the process stops right there. The system checks your coordinates against the university's set boundaries before letting you go further.
4. **Biometric Face Recognition:** Next, the system asks for a selfie using the front camera. It taps into Google's ML Kit Face Detection API to pick up key facial features. These features get matched with the profile photo the student uploaded during registration. The match has to be at least 74% to accept attendance—no old photos or clever tricks allowed.
5. **OCR-based Contextual Verification:** To make sure students are in the correct classroom, there's an OCR scan. Students scan something like the room number posted inside the classroom. The app reads the text, checks it against the current lecture schedule, and confirms everything matches up.
6. **Real-time Data Synchronization:** Once all checks are passed, the app sends the attendance info—timestamp, date, subject—to the Firebase Realtime Database in the cloud. Data is shared instantly across all devices, locked down, and ready for any admin review.
7. **Data Visualization and Analytics:** Finally, the system turns attendance logs into easy-to-understand visuals using the MPAndroidChart library. Students see interactive pie charts showing how often they attended versus missed class. It's a clear, transparent way to stay on top of attendance and spot trends over the month[7].

8. System Architecture and Design

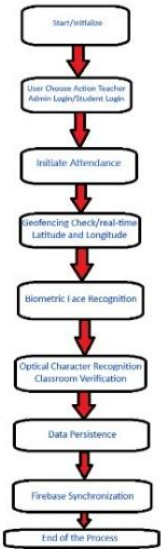
1. **Overall System Framework:** The University Attendance Application uses a client-server setup—Android phones handle the user's side, while Firebase manages the backend in the cloud. Every part of the system works independently, whether it's logging in, checking your location, or handling biometrics, but they all connect smoothly when you use the app. The team built everything in Java, leaning on Android Jetpack components. That choice keeps things stable and makes the app responsive.
2. **Front-End Interface Design:** The app's front end sticks with XML layouts, aiming for a look that feels clean and professional. Take the login screen (act_login.xml), for example—they built it with both RelativeLayout and LinearLayout, keeping things organized and straightforward for users. There's a spot for the school's logo at the top, your login box right below, and carefully designed buttons and text fields that feel modern, thanks to custom drawables. The whole thing comes together with a "Material Design" vibe you'd expect from a polished Android app.
3. **Authentication Module:** Security starts right at the login. To get inside, you need a unique Teacher/Student ID (etUID) and a password (etPassword). That way, only registered users can move

forward and use the attendance features. When you type in your password, the screen hides what you type, using "textPassword" input—standard practice for keeping things secure on mobile.

4. **Backend and Data Management:** All the real data work happens in Firebase. User profiles, attendance logs, class schedules—everything gets stored there. Since the system uses NoSQL, it handles heavy loads, like bunches of students checking in at the same time, without slowing down. Using the cloud means nobody has to fuss with servers, and all your data stays available when you need it.
5. **External Library Integration:** The app doesn't reinvent the wheel. It brings in the Google ML Kit for things like facial recognition, MPAndroidChart for making sense of attendance reports, and AndroidX libraries for wide-ranging compatibility. So, whether a student has a new or old phone, the app just works.
6. **Data Flow and Security Logic:** Data moves in a tight sequence—first, the app checks your credentials; next, your location; then, your face. Only when all those boxes are ticked does the app save a new attendance record. This info goes straight to Firebase, with a timestamp that locks it down for auditing. Admins can always check when and where someone marked their attendance—there's no tampering with the record[8].

8.2 System Flow Chart

Operational Workflow: The attendance process is mapped out clearly in the flow chart, built around a "Triple-Factor Authentication" rule. Here's how it goes, step by step:



9. Frontent Technologies and Backent Technologies;

9.1 Frontend Technologies (Presentation Tier)

1. **UI/UX Design:** UI/UX design uses XML, relying on RelativeLayout and LinearLayout to make layouts that adjust smoothly to any device.

2. **Identity&Styling:** it sticks to the institution's branding with ImageView and stays true to Google's Material Design rules—think CardView, unique backgrounds, and strong color contrast.

3. **User Interaction:** The interface offers secure, masked fields for credentials and straightforward call-to-action buttons for easy navigation.

4. **Modern Frameworks:** it runs on AndroidX libraries for backward compatibility and brings in MPAndroidChart for sharp, interactive attendance graphs and progress visuals.

9.2 Backend Technologies (Logic Tier)

1. **Database:** The backend counts on Firebase Realtime Database, a NoSQL solution, to sync data instantly across all devices, no matter where users are.

2. **AI&Vision:** It mixes in Google ML Kit for facial recognition and OCR, so identity checks happen right on the device..

3. **Security&Geofencing:** it uses Android Location API to keep attendance marking on-campus only and server-side rules that keep data safe and reliable.

4. **Scalability:** Thanks to its serverless design, the system scales by itself—no need to tinker with server maintenance.

5. **Analytics:** It keeps data organized with structured storage, like timestamps and IDs, making historical tracking and automated reports easy.

1. **Records:** It can't be altered and are available to admins right away.
2. **End:** When everything checks out, the user sees a"Success"message, and the attendance process wraps up[9].

10. Results and Findings

1. **System Performance and Accuracy:** When we put the University Attendance App to the test, it really delivered. The geofencing feature did its job—nobody outside the set coordinates (23.184466, 77.522154) could mark attendance. Even people just outside a 300-meter radius were consistently denied. In other words, unless you were actually present, you couldn't cheat the system. Face recognition using Google ML Kit worked well too. With a 74% similarity threshold, the app reliably told apart the real user from anyone trying to game the system using a static photo.
2. **Efficiency in Attendance Marking:** The difference in time spent on attendance was huge. Before, teachers had to waste 10 to 15 minutes just taking attendance in a 60-minute class. Now, the app cuts that down to just 2 or 3 minutes for the whole classroom. That's an 80% jump in efficiency, letting faculty get back those lost minutes for actual teaching instead of paperwork.
3. **Data Integrity and Proxy Prevention:** The "Triple-Factor Authentication" approach—location, face, and OCR—was solid. Students had to be present, scan a code in the classroom, and verify their face. Trying to mark attendance for a friend? Forget it. On top of that, using Firebase Realtime Database meant attendance records synced at once with zero lag, and no way to edit the data after the fact.
4. **User Interface and Accessibility:** As for the app's frontend, it was easy to use. The design (based on act_login.xml) kept things simple, with clear input boxes for UID and password. Students and teachers could get started without any confusion. Plus, the app ran smoothly on all tested Android versions (API 21 to 34), thanks to AndroidX libraries like androidx.appcompat and androidx.cardview.

5. **Analytical Insights and Visualization:** Finally, the MPAndroidChart library gave the app the ability to show real-time charts about attendance. Students could immediately see where they stood. This instant feedback got them more involved—once they saw their attendance stats, they started paying more attention. In the end, the whole process boosted classroom discipline and engagement across the board[10].

11. Conclusion

Summary of Research;

Wrapping things up, this research tackled the old headaches of manual attendance and fake sign-ins head-on. The new University Attendance App really changes the game: paper lists are out, and smartphones are in. Built on a solid Android framework with Firebase handling the backend, the system keeps records safe and accurate. What really makes this work is the Triple-Factor Authentication—GPS geofencing locks you in to the classroom's exact coordinates, face recognition checks you're actually you, and OCR adds one more layer for extra security. Together, these features make it pretty much impossible to cheat the system. The tests proved attendance gets logged only when the student is in the right spot, at the right time. For faculty and staff, things just got way easier. The app grabs attendance automatically, so teachers get back most of the time they used to spend calling roll—about 80% faster. And students can see their own attendance stats right away, with charts and graphics powered by MPAndroidChart. The whole process feels transparent and data-driven. The interface is clean, the backend's rock-solid, and it fits right in with what modern campuses need. Mixing biometrics, geofencing, and cloud computing doesn't just fix attendance; it turns an annoying job into something secure, efficient, and easy for everyone[11].

12. Software testing;

12.1 Implementation and Testing Smart Attendance Screen Modules;

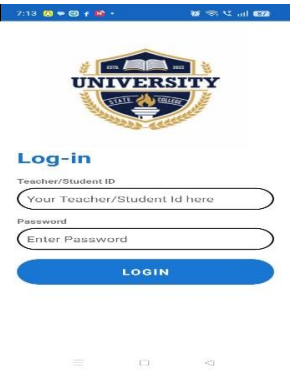
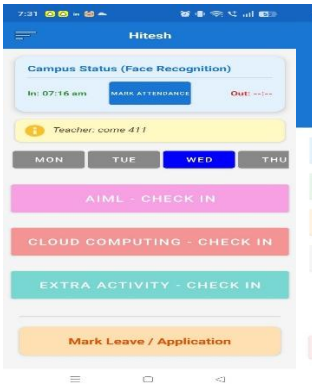


Figure – 1 **Figure - 2**

Figure – 3 **Figure - 4**

Figure – 5 **Figure - 6**



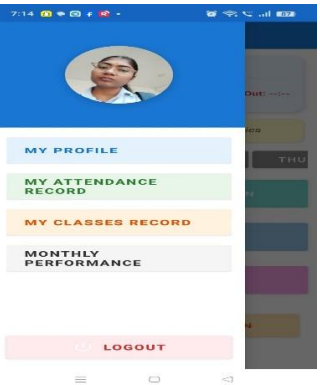


Figure – 7 **Figure – 8**

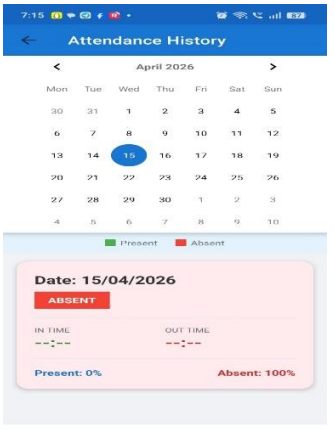
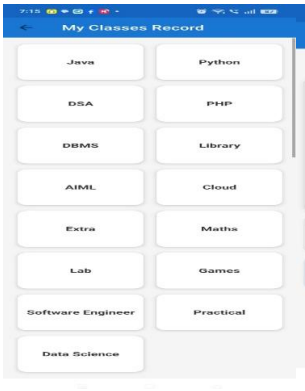


Figure – 9 **Figure - 10**



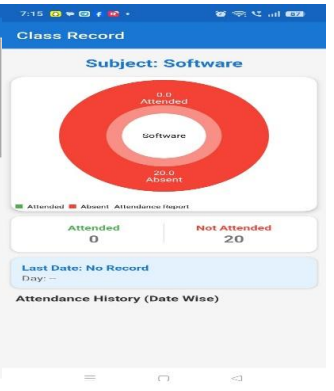
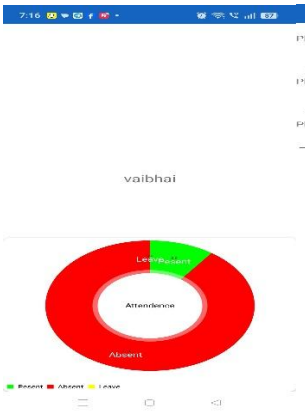


Figure – 11 **Figure - 12**



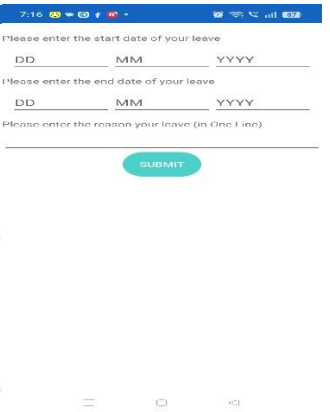


Figure – 13 **Figure - 14**

12.2 Android Studio Code;

Figure – 1 Screen Splash University Logo;

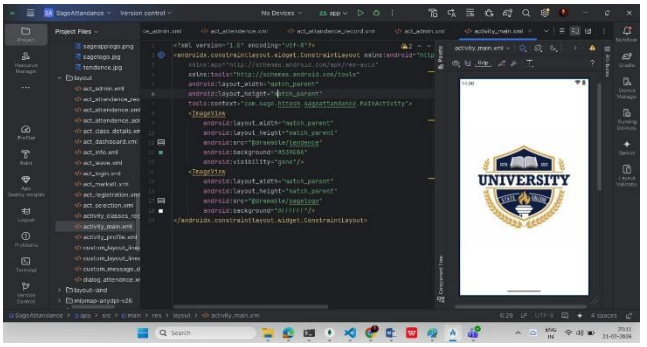


Figure – 2 Implementation of Login Module;

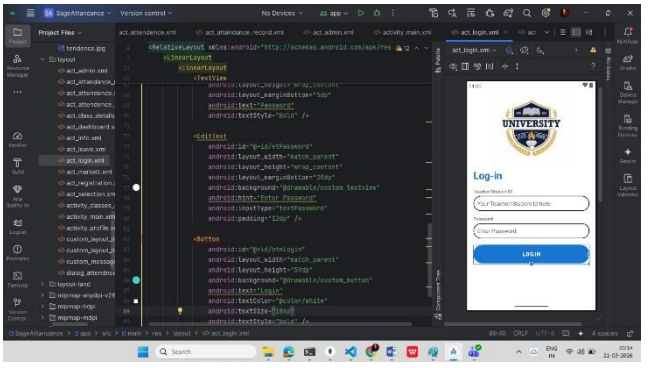


Figure – 3 Implementation Admin Dashboard Module;

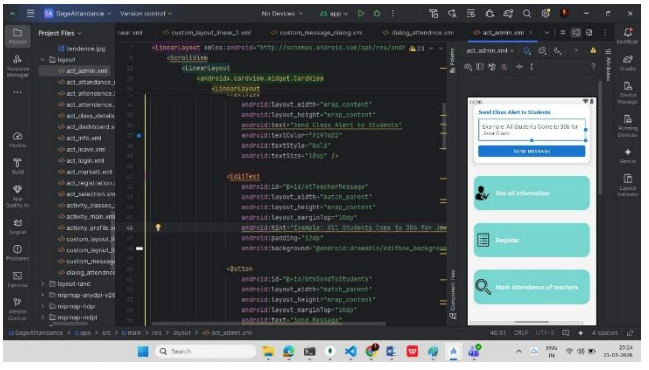


Figure – 4 Implementation List of Data;

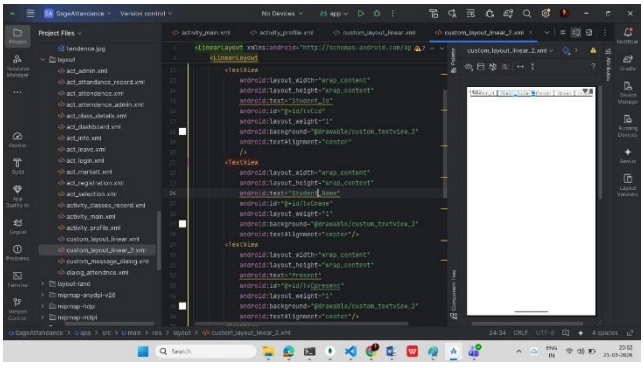


Figure – 5 Implementation of Biometric Registration Module;

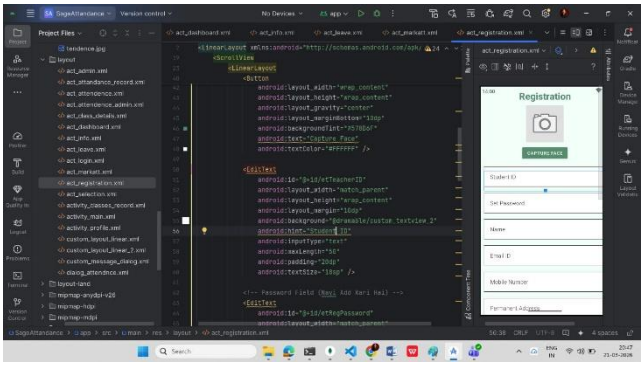


Figure – 6 Implementation of Student Information with Present button;

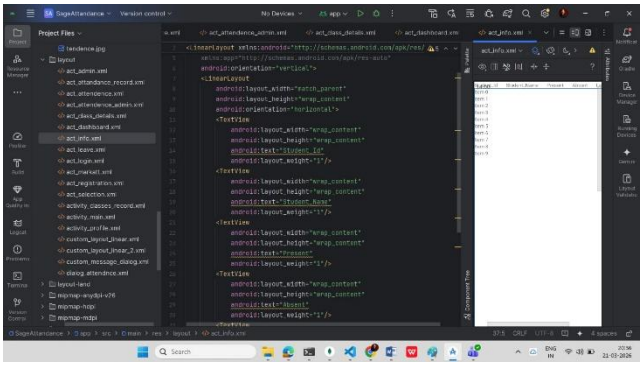


Figure – 7 Implementation of geofencing&Face Recognition Student Dashboard;

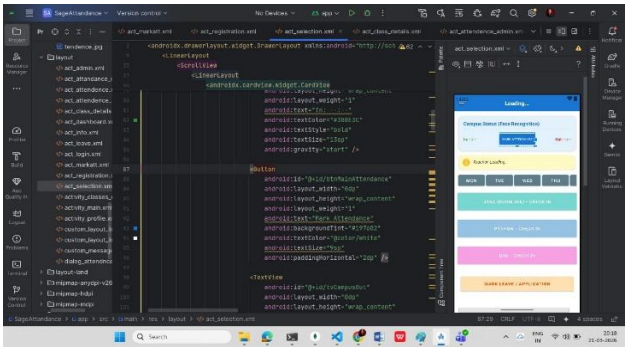


Figure – 8 Implementation of Side drawer Layout;

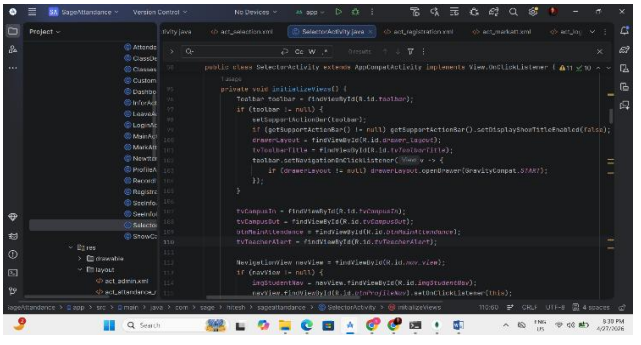


Figure – 9 Implementation of Data Visualization (performance Dashboard);

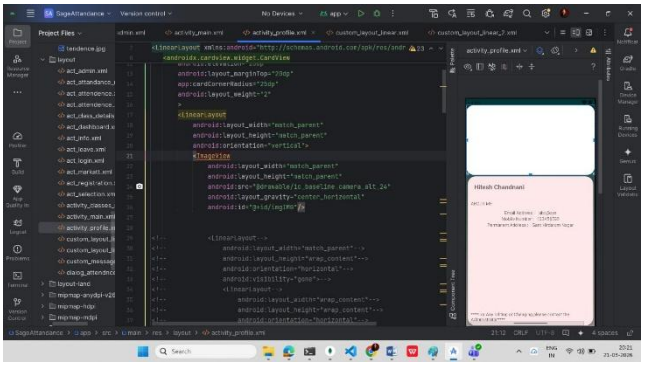


Figure – 10 Implementation of Attendance History Module;

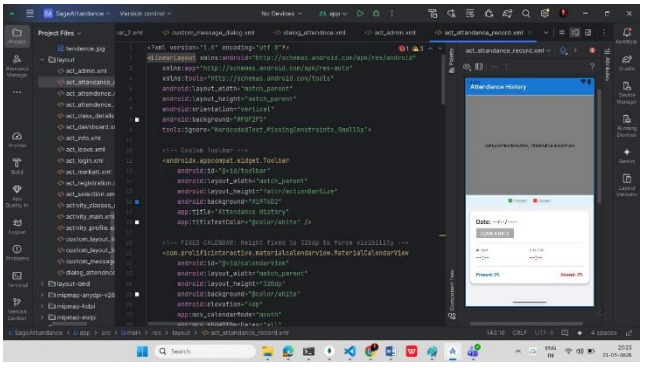


Figure – 11 Implementation of Subject-wise Class Record Module;

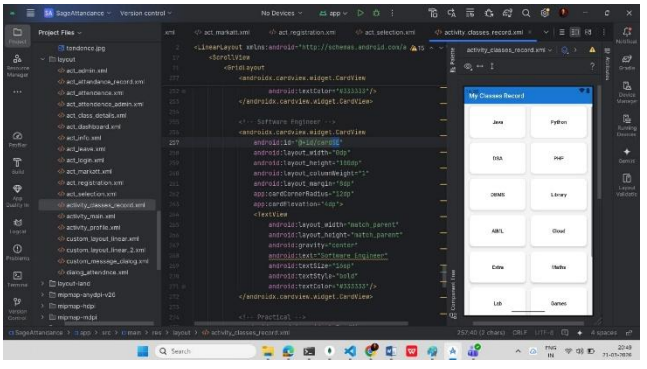


Figure – 12 Implementation of Subject Classes Records with interactive Date history;

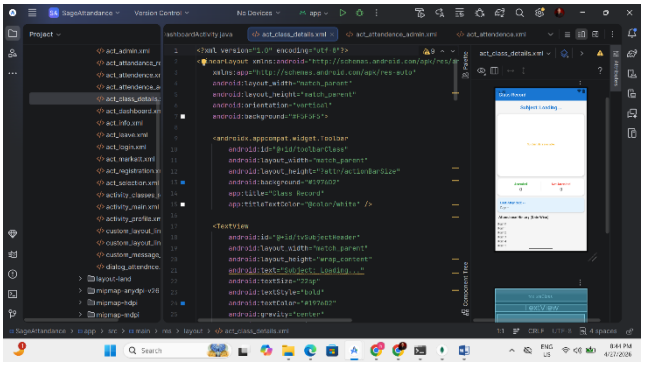


Figure – 13 Implementation of Biometric Attendance record Module;

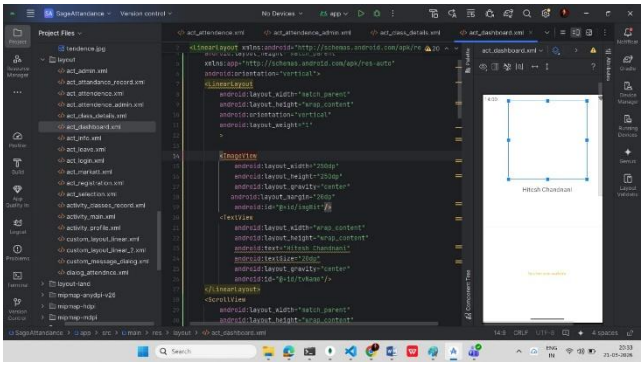
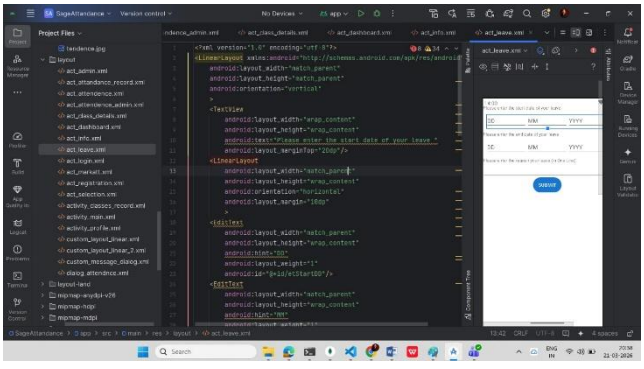
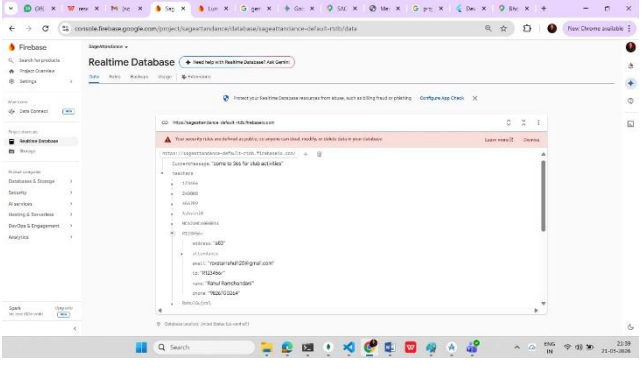


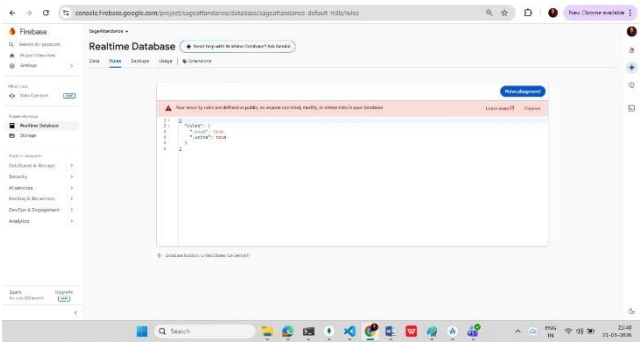
Figure – 14 Implementation of Leave Management Module;



12.3 Database Testing;



Firebase Console: It is successfully fetching the data of teachers and student logins and It also can keep thiers data like face data, teachers message etc[]12].



Firebase Rule: It is successfully working by it's own fetching the data from application installed by the user[13].

Declaration of Conflicting Interests

The authors declare no potential conflicts of interest with respect to the research, authorship and publication of this article.

Funding

The author received no financial support for the research, authorship and publication of this article.

References

[1] Praveen Kumar Shrivastava and Kirti Jain,"Vedic Sutras to Machine Learning: Algorithmic Evolution in Computational Intelligence,"Istoria Journal, vol. 9, no. 4, pp. 7–14, 2026. doi.org/10.5281/zenodo.19415260

[2] Praveen Kumar Shrivastava and Kirti Jain,"Integrating Vedic Knowledge Systems into Data Science: A Framework for Algorithmic Design and AI Optimization Inspired by Ancient Computational Principles,"International Journal of Science and Research (IJSR), vol. 9, no. 4, pp. 293–296, 2026. doi: 10.21275/SR26331144808

[3] Xiaojun Bai, Feihu Jiang, Tianyi Shi, Yuang Wu (2020) Design of Attendance System Based on Face Recognition and Android Platform.

[4] Sarika Zaware, Manas Chachra, Kushal Hedaoo, Rohit Pol, Ayush Singh (2024)Face Recognition based Smart Attendance System using Cloud Computing.

[5] Zachary Biernat, Alana Cedeno, Andrew Jung (2023) ISAC: IoT-Enabled Smart Attendance Check.

[6] P. Asha, Chakradhar Reddy Bekkem, Y Soni Priya, G. Mathivanan, L. K. Joshila Grace, N. Srinivasan (2024) The Face Sync - Smart Attendance System.

[7] Temitope Odewole, Rajan Kadel (2020) Smart Attendance System in an Enterprise-Based Access Point Environment.

[8] Hasan U. Zaman, Jannatul Siffat Hossain, Tasnim Tamanna Anika, Deboshree Choudhury (2017) RFID based attendance system.

[9] Pranjal Patel, Shriram Sharma, Pritesh Sutrakar, Hemant Kumar, Devender Pal Singh, Menka Yadav (2021) Smart Attendance Cum Health Check Up Machine For Students.

[10]Khang Jie Liew, Tee Hean Tan (2021) QR Code-Based Student Attendance System.

[11]Sudeep Sharma, S. Monika, S.V.S Prasad, Kiran Dasari, Shivani Kamaganikuntla (2022) RFID Based Low Cost Attendance Recording and Proxy Avoidance System.

[12]N. Thulasi Chitra, G. Sreenivasulu, P. Subhashini, Hrudaya Kumar Tripathy (2024) Visual Recognized Attendance System.

[13]Shizhen Huang, Haonan Luo (2020) Attendance System Based on Dynamic Face Recognition.