

Smart College Management System

Keerthana S J^{*1} and Prof. Sandeepa K M^{*2}

^{*1}Student, Department of Computer Applications

^{*2}Assistant Professor, Department of Computer Applications

Maharaja Institute of Technology Mysore, Mandya, Karnataka, India.

Abstract: *The Smart College Management System (SCMS) to solve a problem I kept noticing in how colleges of my institution's size handle everyday admin work: attendance registers, fee ledgers and notices are all kept separately, often on paper or in scattered spreadsheets, which makes them hard to track and easy to get wrong. My system brings these three tasks onto one platform with three logins — Admin, Faculty and Student — using CodeIgniter 4 (PHP) and MySQL. The part I am most satisfied with is the attendance module: before a faculty member can mark attendance, the system checks their device's GPS coordinates against a stored college location using the Haversine formula, so attendance can only be marked from within campus. I have also implemented fee collection with receipt generation and balance tracking, course/subject setup, and a simple notice board. This paper walks through how I designed and built the system, what worked, and what I would still improve.*

Keywords: *Smart College Management System*

1. INTRODUCTION

When I started scoping this project, I looked at how attendance and fees are actually handled day-to-day: attendance is marked on paper registers that can be signed on someone else's behalf, and fee payments are tracked in registers or spreadsheets that are easy to misplace or fall out of sync. None of this is a technology problem exactly — it is a coordination problem, and a small web application is enough to fix most of it if it is built around the way the college actually works, not around a generic feature list.

That is why I designed SCMS around three specific roles instead of one shared login. The Admin sets up the structural data — departments, courses, subjects, fee structures and the reference GPS location — because these things change rarely and should not be editable by faculty or students. Faculty use the system daily to mark attendance for the course and semester they are assigned to. Students use it to check their own attendance percentage, fee balance, timetable and any notices, without having to go and ask the office.

The module I spent the most time thinking through was attendance. A digital attendance system on its own does not stop someone from marking a class present while being nowhere near the college — it just moves the same problem from paper to a screen. To actually address that, I added a location check: the browser reads the faculty member's GPS position, and the server compares it against the college's stored coordinates using the Haversine great-circle distance formula before the attendance form is even shown. If the distance is more than the allowed radius, the form simply does not open.

The rest of this paper is organised as follows: Section II covers how I built the system and the tools I used; Section III explains how the modules and data flow fit together; Section IV discusses what the finished system does and where it still falls short; and Section V closes with my conclusions and what I would build next.

II. METHODOLOGY

I did not build this in one pass. I started by listing out, module by module, what each of the three roles needed to be able to do, then designed the database tables for that module, then wrote the controller logic, then the view, and tested that one piece before moving to the next. This meant the login and role-redirect flow was working before I touched attendance, and attendance was working before I started on fees.



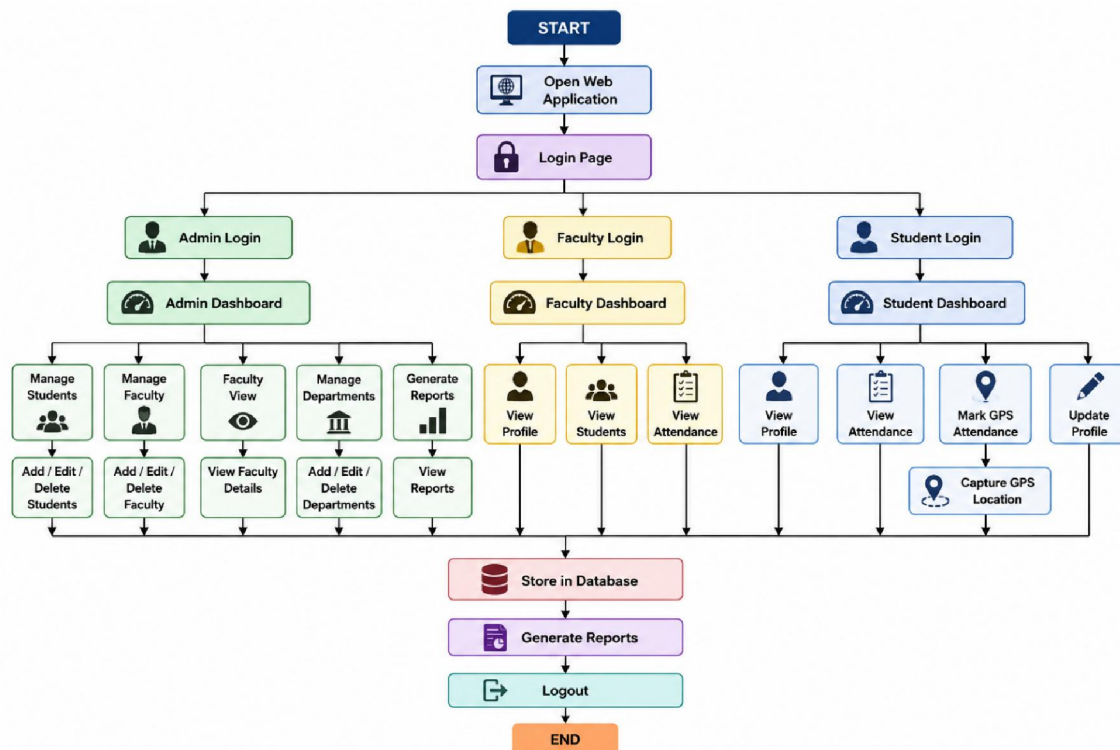
I chose CodeIgniter 4 mainly because its Model-View-Controller structure keeps things organised without adding much overhead — routes map cleanly to controller methods, and its Query Builder let me write parameterised database queries without hand-writing raw SQL everywhere. MySQL stores everything from static data like departments and courses to transactional data like attendance rows and fee payments. The frontend is plain HTML, CSS and JavaScript; the only notable client-side piece is the call to the browser's Geolocation API to get the faculty member's coordinates for the attendance check.

III. MODELING AND ANALYSIS

Table 1. Tools and usage

Component	Technology
Backend Framework	CodeIgniter 4 (PHP)
Database	MySQL
Frontend	HTML, CSS, JavaScript
Architecture	Model-View-Controller (MVC)
Location Verification	Haversine formula (browser Geolocation API)
Authentication	PHP password_hash() / session-based login

FLOW DIAGRAM



IV. SNAP SHOTS

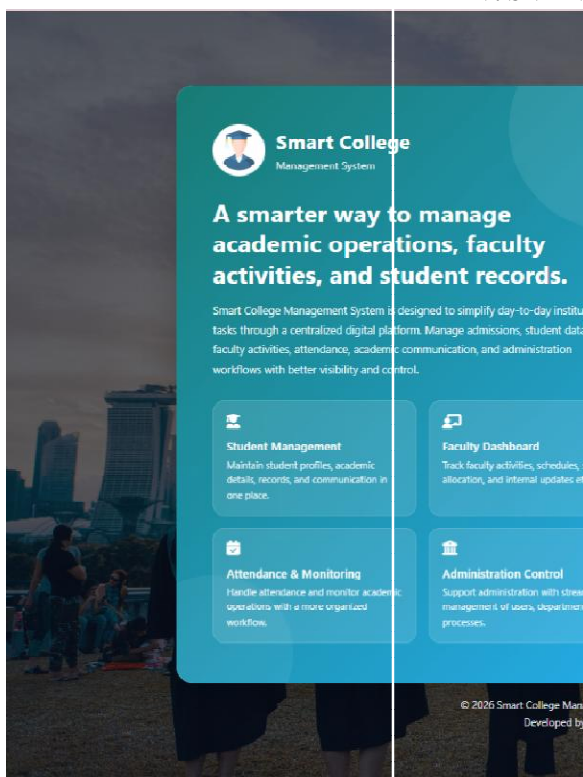


Figure 1: Login Page

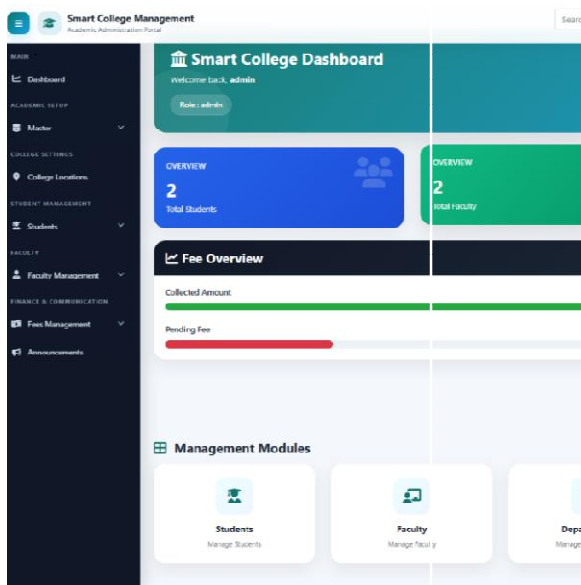


Figure 2: Admin Dashboard



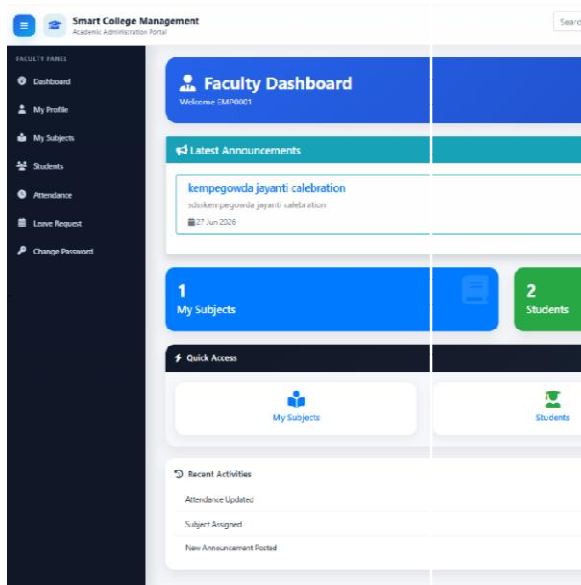


Figure 3: Faculty dashboard

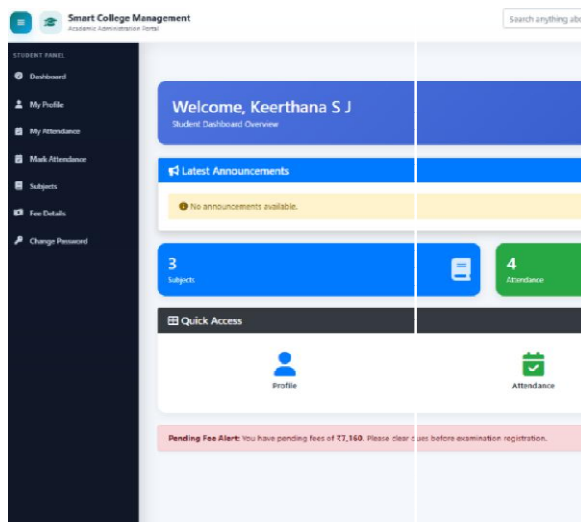


Figure 4: Student Dashboard



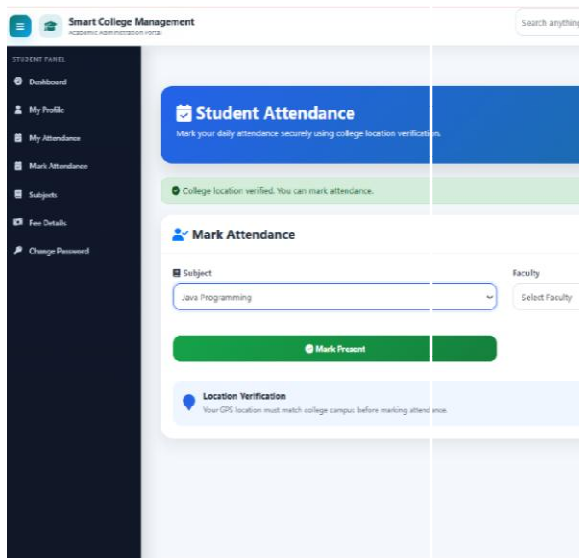


Figure 4: Student Attendance

V. RESULTS AND DISCUSSION

By the end of development, all three portals were working end to end. As Admin, I can set up departments, courses, subjects and fee structures, configure the attendance geofence, and pull up fee reports showing total collections, outstanding balances, and a monthly/payment-mode breakdown. As Faculty, I can only mark attendance if I am within the radius I configured around the college — which was the whole point of building the geofence in the first place. As Student, I can see my attendance history, fee status and notices in one place instead of asking around.

While testing, I confirmed the core flows behave the way I intended: duplicate email registration is rejected, mismatched passwords are caught, the geofence correctly opens or blocks the attendance form based on distance, and fee collection calculates the balance and receipt number correctly every time. One gap I found while reviewing my own code is that the `password_verify()` check in the login flow is currently commented out, so login works off a matching email alone at the moment — this is the first thing I plan to fix before treating the system as production-ready.

VI. CONCLUSION

Building SCMS showed me that a college's core day-to-day admin work — attendance, fees, course records and notices — can realistically be handled by one MVC-based web application with separate Admin, Faculty and Student logins, without needing a large team or specialised hardware. The geofenced attendance module was the part I am proudest of, since it directly targets a real weakness (proxy attendance) rather than just digitising the same paper process. The fee module taught me that keeping every payment as its own record, rather than overwriting a running total, makes the history far easier to audit later.

I consider the project to have met what I set out to do: a working application with clear user roles and a complete path from login to task completion for each of them. Going forward, I want to re-enable proper password verification, add email or SMS notification delivery, and extend the reports section to export PDF statements.

VII. ACKNOWLEDGEMENTS

I express my gratitude to the project guide for their invaluable guidance, and unwavering support throughout the project duration. I'm equally thankful to the faculties of MIT Mysore further insightful suggestion and motivation which makes well experience in my work. I extend my sincere appreciation to the management of the institute for providing the



necessary resources, that helps in achieving successful project. I'm deeply grateful to my classmates and friends for their coordination, assistance and encouragement during the process of development and I finally, acknowledged with gratitude the contribution of all those directly or indirectly supported in upbringing the project.

REFERENCES

- [1] G. Yuan, "Design and Evaluation of Intelligent Student Management System Based on Multi-Modal Data," *Systems and Soft Computing*, vol. 7, Article 200370, **2025**, doi: 10.1016/j.sasc.2025.200370.
- [2] W. A. Sajid, N. Mirzah, F. M. Mustafa and Y. Shabala, "Educational Institution Management Information System," in *Proceedings of the 35th Conference of Open Innovations Association (FRUCT)*, pp. 625–632, **2024**, doi: 10.23919/FRUCT61870.2024.10516390.
- [3] H. Huang and B. Li, "Design and Implementation of Student Management System of Integrated Programmable Device Programming System," *Scientific Reports*, vol. 14, Article 11873, **2024**, doi: 10.1038/s41598-024-62844-z.
- [4] T. Daim, D. Ozdemir Gungor, N. Basoglu, A. Yarga and H. VanDerSchaaf, "Exploring Student Information Management System Adoption Post Pandemic: Case of Turkish Higher Education," *Technology in Society*, vol. 77, Article 102557, **2024**, doi: 10.1016/j.techsoc.2024.102557.
- [5] K. Wahab, A. Ullah, Y. Wang and A. Majid, "Development of a Web-Based Student Portal System for University Students," in *Proceedings of the 2023 International Conference on Intelligent Education and Intelligent Research*, **2023**, doi: 10.1109/IEIR59294.2023.10391217.
- [6] C. Gürkut, A. Elçi and M. Nat, "An Enriched Decision-Making Satisfaction Model for Student Information Management Systems," *International Journal of Information Management Data Insights*, vol. 3, no. 2, Article 100195, **2023**, doi: 10.1016/j.jjime.2023.100195.

