Strategic Architecture of ProjectPilot: Streamlining Project Management in Engineering Education

Anuj Dighe[1], Mayur Kumawat[2], Chaitali Dhaije[3], Om Waghmare[4], Prof. Archana Mate[5]
[1, 2, 3, 4] *Students, Department of Computer Engineering, Dilkap Research Institute of Engineering and Management Studies, Neral 5 Professor, Department of Computer Engineering, Dilkap Research Institute of Engineering and Management Studies, Neral*

***Abstract*— In engineering education, mini-projects are essential for applying theoretical knowledge; however, manual management of these projects—group formation, guide allotment, and progress tracking—becomes increasingly complex as institutions scale. This paper presents ProjectPilot, a specialized Project Management System (MPMS) designed to digitize the academic project lifecycle. Unlike legacy systems that focus solely on allocation, ProjectPilot utilizes a three-tier role-based architecture (Student, Guide, and Branch Admin) to automate the "middle ground" of project execution. By integrating secure JWT-based authentication, forensic activity logging, and milestone-based progress tracking, the platform addresses systemic inefficiencies in communication and transparency, ultimately establishing a new standard for digital governance in engineering education.**

***Index Terms*— Academic Project Management, JWT Authentication, FProject Workflow, Role-Based Access Control (RBAC), Digital Education, Forensic Accountability.**

I. INTRODUCTION

The operational landscape of modern engineering education is undergoing a profound paradigm shift from static, paper-based administration to dynamic, data-driven governance. Within this context, "Projects" represent a critical crucible for student development. However, the administrative scaffolding supporting these projects has historically been fragile, fragmented, and prone to systemic inefficiency.

Recent work by Tembhurne et al. (2021), Bhatt et al. (2020), and Patil et al. (2022) underscores a clear evolution: from manual coordination toward purpose-built digital modules for allocation, dashboards, and visualization. Yet, these solutions remain compartmentalized—tackling single facets without end-to-end integration. To address these limitations, we propose ProjectPilot, a platform that unifies allocation, milestone tracking, evaluation, and secure document exchange under a single, role-driven architecture. ProjectPilot integrates automated milestone alerts, algorithmic guide suggestions, and encrypted file sharing to set a new standard for secure, transparent mini-project oversight.

II. LITERATURE REVIEW: ANALYSIS AND GAP IDENTIFICATION

A. Current State of Academic Project Management

The surveyed literature collectively depicts a field in transition. Tembhurne et al. [1] introduced unified portals to automate assignment, while Bhatt et al. [2] added role-specific dashboards. Patil et al. [3] advanced visualization via Gantt charts. Overall, while these systems demonstrate movement toward digital tooling, they remain fragmented, with security and end-to-end workflow integration largely unaddressed.

B. Specific Technical Challenges Identified

Through a deconstruction of existing literature, several critical challenges were isolated:

- Fragmented Processes: Allocation, tracking, and evaluation are often handled in separate tools, causing data silos.
- Manual Milestone Management: Prior dashboards required manual status updates, leading to visibility gaps and potential delays.
- Limited Progress Visualization: While Gantt charts improved timelines, they lacked real-time status updates tied to deliverable verification.

- Inadequate Security: Most legacy systems lack robust authentication (e.g., JWT), exposing student/faculty data to unauthorized access.

C. Individual Work Analysis

1. S. Tembhurne et al. (2021) [1]: Digitized the "market" of guide selection via centralized portals.
2. R. Bhatt et al. (2020) [2]: Introduced role-specific dashboards to simplify documentation.
3. A. Patil et al. (2022) [3]: Focused on temporal logic through reminder modules and dependency-based progress flows.
4. P. Sharma et al. (2021) [4]: Established the baseline for secure file sharing using JWT tokens.

Table I: Summary of Methodology Enhancements

Prior Methodology	ProjectPilot Enhancement	Impact
Centralized Allocation [1]	Algorithmic suggestions + manual override	Fairer, data-driven guide assignments
Role-based Dashboards [2]	JWT-enforced RBAC (Student/Guide/ Admin)	Strong security and role-tailored UX
Gantt-chart Tracking [3]	Automated milestone-triggered notifications	Proactive deadline management
JWT Authentication [4]	Forensic logging + Activity Auditing	Accountability and Defense-in-depth

III. ADOPTION AND IMPROVEMENT OF PRIOR METHODOLOGIES

ProjectPilot synthesizes and advances these methodologies to deliver a cohesive, automated platform.

1. Project Allocation → Automated Allotment with Decision Support ProjectPilot retains Tembhurne's [1] centralized portal concept but adds Algorithmic Guide Suggestions. By leveraging guide expertise and workload, the system recommends optimal pairings while allowing Branch Admins a "Manual Override" for administrative flexibility.

2. Dashboards → Granular JWT-Enforced RBAC While Bhatt et al. [2] used tailored views, ProjectPilot formalizes this into a strict Role-Based Access Control (RBAC) model. Permissions are enforced at every API endpoint via JWT claims, ensuring Students, Guides, and Admins interact only with data pertinent to their role, significantly reducing data leakage risk.

3. Static Visualization → Proactive Milestone Notifications Patil's [4] Gantt-chart tracking is evolved into First-Class Milestone Entities. Each milestone has dependencies and deliverables. The system triggers automated email/in-app notifications for upcoming and overdue tasks, shifting from passive charting to proactive deadline management.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

ProjectPilot utilizes the MERN stack (MongoDB, Express, React, Node.js) to provide a responsive, high-concurrency environment.

A. Tripartite Role Structure

1. Student: Primary generator of data; manages team formation and continuous development.
2. Guide: Gatekeeper of quality; manages groups via a centralized analytical dashboard.
3. Branch Admin: System architect; defines guide pools, sets global term dates, and ensures compliance.

B. Security and Forensic Accountability By strictly enforcing server-side time for all milestone locks, ProjectPilot removes ambiguity. Every action—a Guide approving a topic or a Student uploading a file—is recorded in Secure Activity Logs. This creates an irrefutable audit trail that protects stakeholders from disputes.

V. SYSTEM REQUIREMENTS

Component	Minimum Requirements
Processor	Intel Core i3 or above
RAM	4 GB Minimum (8 GB Recommended)

Storage	100 GB (For local development and document storage)
Display	1080p Screen (Optimized for dashboards)
Authentication	JWT (JSON Web Token) with signed payload

VI. CONCLUSION

ProjectPilot represents a synthesis of academic inquiry and modern engineering. It distinguishes itself by closing the "execution gap" between allocation and evaluation. By building on the foundations of Tembhurne, Bhatt, Patil, and Sharma, the platform establishes a new standard for the digital governance of engineering education—ensuring control for admins, clarity for guides, and a structured path to success for students.

REFERENCES

[1] S. Tembhurne et al., "Online Project Allocation and Evaluation System," *IJERT*, vol. 10, no. 5, 2021.
[2] R. Bhatt and V. Joshi, "Project Management System for Students and Faculties," *JETIR*, vol. 7, no. 6, 2020.
[3] A. Patil and K. Deshmukh, "Academic Project Monitoring System," *IJESC*, vol. 12, no. 4, 2022.
[4] P. Sharma and A. Kumar, "Secure File Sharing and Role Authentication using JWT," *IRJET*, vol. 8, no. 12, 2021.
[5] ProjectPilot MiniProject Manager | PDF | Mac Os | Java Script. https://www.scribd.com /document /894914793/ProjectPilot-MiniProject-Manager