A Data-Oriented Academic Burnout Monitoring and Assessment System Based on Student Academic Behavior

Gattu Pravallika[1], Umamaheswararao Mogili[2*], V. Harsha Vardhan[3], M. Vasu[4], M. Gowthami[5], D. Sarojini Devi[6]

[1,2*]*Assistant Professor, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

[3,4,5,6] *B.Tech, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

doi.org/10.64643/IJIRTV12I10-194282-459

Abstract-Academic burnout among students has emerged as a critical concern in higher education institutions, often leading to diminished academic performance, mental health challenges, and increased dropout rates. This project presents a comprehensive web-based Academic Burnout Monitoring and Support System designed to identify early warning signs of academic distress and provide timely interventions through explainable rule-based algorithms and artificial intelligence-assisted recommendations. The system integrates multiple behavioral and academic indicators including attendance patterns, study consistency, assignment workload, and grade point averages to compute a composite burnout score ranging from zero to one hundred. Unlike opaque machine learning approaches, the implemented solution employs a transparent, deterministic scoring methodology that ensures stakeholders can understand and trust the assessment process. The platform adopts a multi-stakeholder architecture supporting four distinct user roles: students, faculty members, heads of departments, and system administrators. Students receive personalized dashboards displaying their current burnout levels, contributing factors, and actionable recommendations for improvement. Faculty members can monitor individual students under their guidance, record attendance, manage assignments, and facilitate doubt clarification sessions. Departmental heads gain aggregated insights into institutional trends, enabling proactive policy interventions. The system incorporates automated weekly email notifications, real-time analytics dashboards, and an AI-powered recommendation engine that supplements rule-based assessments with contextual guidance generated through the OpenAI API with robust fallback mechanisms. The technological foundation comprises a Node.js-based Express server providing RESTful API endpoints, a MySQL relational database ensuring data integrity through normalized schemas and foreign key constraints, and a React-based single-page application utilizing the Vite build tool for optimized performance. The system architecture emphasizes scalability, maintainability, and security through role-based access controls, input validation, and separation of concerns. Evaluation of the system demonstrates its capability to accurately identify students experiencing high burnout conditions with sensitivity to individual circumstances while providing clear explanations for computed scores. The outcome is a stable, production-ready platform that bridges the gap between quantitative student analytics and qualitative support mechanisms, serving as a valuable tool for educational institutions committed to holistic student welfare and academic success.

Keywords: Academic Burnout, Student Engagement Monitoring, Data Analytics, Burnout Prediction, Educational Technology.

I. INTRODUCTION

1.1 Background and Motivation

The contemporary higher education landscape presents students with unprecedented academic pressures arising from competitive curricula, demanding assessment schedules, and evolving employment expectations. Research consistently demonstrates that prolonged academic stress without adequate support mechanisms contributes to burnout syndrome, characterized by emotional exhaustion, depersonalization, and diminished sense of accomplishment. The consequences extend beyond individual student welfare, manifesting as institutional challenges including lower retention rates, compromised learning outcomes, and increased demand for counseling services. Traditional approaches to student welfare monitoring often rely on reactive interventions triggered only after students exhibit severe distress symptoms or academic failures, by which point remediation becomes significantly more challenging. Educational institutions require proactive systems capable of identifying at-risk students during early stages of academic decline when preventive interventions remain most effective. However, such systems must balance analytical rigor with interpretability, ensuring that assessments remain transparent to students, faculty, and administrators. The proliferation of opaque machine learning models in educational contexts has raised concerns regarding algorithmic bias, lack of explainability, and difficulty in establishing stakeholder trust. Furthermore, comprehensive student support requires integration of multiple data streams including attendance records, assignment submissions, grade distributions, and self-reported study habits, which traditional paper-based or fragmented digital systems struggle to consolidate effectively.

1.2 Problem Statement

Educational institutions face three primary challenges in addressing student burnout. First, existing monitoring approaches lack systematic integration of behavioral and academic indicators, relying instead on isolated metrics that fail to capture the multidimensional nature of academic stress. Second, manual tracking and analysis of student performance data across multiple sources consumes significant faculty time and resources while remaining prone to oversight and inconsistency. Third, students often receive generic guidance rather than personalized, actionable recommendations tailored to their specific circumstances and contributing stress factors. The absence of real-time analytics and automated alert mechanisms delays identification of deteriorating student conditions until critical thresholds are crossed. Additionally, communication barriers between students and support resources, including faculty mentors and counseling services, impede timely assistance delivery. Existing systems frequently fail to differentiate between transient performance fluctuations and sustained patterns indicative of genuine burnout risk. The lack of historical trending and predictive capabilities prevents proactive resource allocation and policy adjustments at the institutional level.

1.3 Objectives

This project aims to design, implement, and evaluate a comprehensive Academic Burnout Monitoring and Support System. The first objective is to develop a transparent, rule-based burnout assessment algorithm combining study consistency, attendance, assignment workload, and academic achievement into an interpretable composite score with balanced sensitivity and specificity. The second objective is to engineer a scalable, secure web-based platform with role-specific interfaces, strong authentication and authorization, intuitive navigation, real-time visualization, and responsive performance across devices. The third objective is to integrate AI-driven recommendation capabilities that enhance rule-based assessments while ensuring graceful degradation during API disruptions. The fourth objective is to implement automated notification workflows, including weekly email summaries with idempotency safeguards and compliance-ready audit trails. The fifth objective is to deliver analytics and reporting tools supporting trend analysis, cohort comparison, intervention evaluation, and both individual drill-down and aggregated institutional insights.

II. LITERATURE REVIEW

Academic burnout is a psychological syndrome arising from prolonged academic stress, characterized by emotional exhaustion, cynicism toward studies, and reduced perceived efficacy. Adapted from occupational burnout theory, it spans three dimensions: exhaustion from demands, detachment from learning, and feelings of inadequacy [1]. Empirical research reports prevalence rates between twenty and sixty percent, with higher incidence in rigorous programs such as medicine, engineering, and sciences [2]. Unaddressed burnout negatively affects academic performance, persistence, and completion outcomes while elevating risks of anxiety, depression, substance misuse, and sleep disruption [3]. Physiological impacts include stress-related health effects, and institutions face increased attrition and reputational consequences. Contributing factors span individual influences such as perfectionism and coping deficits, interpersonal factors including social isolation and limited support, and systemic influences such as curriculum intensity and assessment load, reflecting cumulative stress exceeding coping capacity [4]. Educational data mining and learning analytics have advanced from simple threshold-based alerts toward predictive models that classify at-risk students using techniques such as logistic regression, decision trees, support vector machines, and neural networks [5]. These methods improve predictive accuracy but often reduce interpretability; producing probabilistic risk scores that stakeholders may struggle to contextualize or translate into interventions [6]. Evaluations of early warning systems show mixed results, with success linked to administrative support, faculty training, workflow integration, and clear intervention protocols [7].

Common barriers include data quality issues, privacy concerns, infrastructure limitations, and insufficient follow-up resources. The trade-off between model complexity and transparency remains critical, driving emphasis on explainable artificial intelligence approaches that strengthen trust, fairness, and accountability in educational decision-making [8]. Web-based platforms for student support have evolved from simple information repositories to interactive systems incorporating personalized dashboards, communication channels, and intervention tracking capabilities [9]. Contemporary learning management systems integrate attendance tracking, grade recording, and assignment management but typically lack dedicated mechanisms for holistic student welfare monitoring [10]. Specialized student success platforms have emerged focusing on retention analytics, advising workflows, and intervention case management, though these often require substantial institutional investment and integration with existing information systems [11]. Mobile-responsive designs and progressive web applications have expanded accessibility, enabling students to monitor their progress and receive guidance through smartphones and tablets [12]. Push notification capabilities support timely alerts regarding assignment deadlines, attendance concerns, and available support resources. However, notification fatigue represents a recognized challenge, with excessive or poorly targeted messages leading to user disengagement [13]. Research on notification design emphasizes personalization, actionability, and timing as critical factors influencing effectiveness in promoting behavioural change [14]. The integration of artificial intelligence technologies in educational contexts spans multiple applications including intelligent tutoring systems, automated essay scoring, chatbot advisors, and personalized learning path recommendations [15]. Natural language processing enables analysis of student communications, discussion forum participation, and open-ended survey responses to detect sentiment patterns potentially indicative of distress on hospitals [16]. Secure Data Storage and Sharing in Multi-Cloud Environment In the cloud storage is also described to store the predicted data in a secured way [17-21].

III. METHODOLOGY

3.1. System Requirements Analysis

The requirements elicitation process involved consultation with educational stakeholders including students, faculty members, academic advisors, and institutional administrators to identify essential functionality, usability expectations, and integration constraints. Functional requirements encompassed user authentication and authorization supporting four distinct roles, attendance recording with study hour tracking, assignment creation and submission workflows, semester-wise grade management with credit calculations, automated burnout score computation, personalized recommendation generation, doubt clarification forums with faculty assignment mechanisms, real-time analytics dashboards, weekly email notification scheduling, and comprehensive administrative oversight capabilities. Non-functional requirements addressed performance expectations including page load times under three seconds for typical operations, support for concurrent access by hundreds of simultaneous users, responsive design adapting to diverse screen sizes and device types, database transaction consistency ensuring data integrity, and backwards compatibility with modern web browser versions. Security requirements mandated password-protected authentication, role-based access controls preventing unauthorized data access, input validation preventing injection attacks, secure communication protocols for data transmission, and audit logging for critical operations enabling accountability and compliance verification.

3.2. System Architecture Design

The system employs a three-tier architecture separating presentation, business logic, and data persistence layers to facilitate modularity, maintainability, and scalability. The presentation layer consists of a React-based single-page application utilizing component-based architecture for UI construction, React Router for client-side navigation, Axios for HTTP communication, and Tailwind CSS for responsive styling. The application implements role-specific layouts and routes ensuring users access only functionality appropriate to their authorization level. The business logic layer comprises a Node.js runtime executing an Express.js web framework that exposes RESTful API endpoints for client-server communication. The server-side architecture organizes functionality into logically grouped route handlers addressing authentication, student operations, faculty operations, administrative functions, and shared utilities. Middleware components provide cross-cutting concerns including CORS policy enforcement, JSON request body parsing, file upload handling through Multer, and error handling with appropriate HTTP status codes and descriptive messages. The data persistence layer utilizes MySQL relational database management system selected for its ACID transaction guarantees, mature ecosystem, and relational data modeling capabilities suitable for academic domain entities with well-defined relationships.

3.3. Burnout Assessment Algorithm Design

The burnout scoring algorithm synthesizes four independent behavioral and academic indicators into a composite score ranging from zero to one hundred through deterministic rule-based calculations. Study consistency assessment examines the average daily study hours over the past seven days, with scoring thresholds assigning forty points for critically low study time below two hours, thirty points for concerning levels between two and three hours, twenty points for adequate study between three and four hours, and ten points for healthy engagement exceeding four hours daily. This inverse relationship reflects greater burnout contribution from reduced study engagement. Attendance performance evaluation calculates attendance rate as the proportion of classes attended over the past fourteen days, assigning twenty-five points for critically low attendance below sixty percent, eighteen points for concerning attendance between sixty and seventy-five percent, ten points for good attendance between seventy-five and ninety percent, and five points for excellent attendance exceeding ninety percent. The two-week window captures recent engagement patterns while providing sufficient sample size for stable estimates. Assignment workload stress assessment counts pending assignments in the student's queue, contributing twenty points when five or more assignments remain pending indicating critical overload, fourteen points for three to four pending assignments representing moderate stress, eight points for one to two pending assignments suggesting manageable workload, and two points when no pending assignments exist. This factor captures the cumulative burden of incomplete obligations contributing to time pressure and anxiety. Academic performance stress evaluation computes the average semester grade point average across all completed semesters, assigning fifteen points for critically low performance below six point zero suggesting severe academic distress, ten points for at-risk performance between six and seven, six points for moderate performance between seven and eight, and two points for strong performance exceeding eight point zero. Grade performance reflects the intersection of ability, effort, and environmental factors relevant to burnout assessment.

3.4. Database Schema Design

The database schema implements a normalized relational model minimizing redundancy while maintaining query performance through strategic indexing and constraint enforcement. Core entity tables include students, faculty, hods (heads of departments), and admins representing the four primary user types with common attributes including unique identifiers, authentication credentials, email addresses, full names, and departmental affiliations where applicable. Students additionally store roll numbers and years of joining enabling academic year calculations. Relationship tables implement many-to-many associations including faculty_students mapping which faculty members advise which students, enabling flexible assignment configurations. Attribute-rich relationship tables include attendance_logs recording daily entries per student per faculty with attendance status enumeration, self-reported study hours, and date stamps. The unique constraint on student-faculty-date combinations prevents duplicate entries while permitting historical tracking. Academic performance tracking employs a semester-based structure with semester configuration tables defining semester numbers and academic years, subject definitions specifying course codes, names, credit values, and departmental affiliations, grade mapping tables establishing letter grade to grade point conversions, student_semester_grades recording individual subject-level performance with unique constraints preventing duplicate subject entries per student per semester, an semester_results tables aggregating semester grade point averages, total credits, and result status with computed values derived from constituent subject grades.

3.5. Frontend Implementation Approach

The React-based frontend architecture employs functional components with hooks for state management, side effects, and context sharing. The component hierarchy implements layout components providing consistent navigation and structure across roles, page components implementing specific feature screens, reusable UI components including cards, forms, tables, and charts abstracting common patterns, and utility functions handling data transformation, validation, and safe rendering with null checks. Role-based routing configures URL patterns mapped to role-specific layout wrappers that instantiate appropriate navigation menus and ensure access controls at the UI level complementing backend authorization. Protected routes verify authentication status stored in browser local storage, redirecting unauthenticated users to login pages and unauthorized users to appropriate dashboards. State management follows local component state for UI-specific concerns and lifted state for shared data, avoiding premature introduction of complex state management libraries given the application's moderate complexity. API interaction abstracts HTTP communication through a configured Axios instance providing base URL configuration, request interceptors for authentication token injection when implemented, and response interceptors for centralized error handling and toast notification triggers.

3.6. Backend Implementation Approach

The Express server organizes endpoint definitions into logical groups with route handlers implementing business logic, database queries, response formatting, and error handling. Authentication endpoints validate username password combinations against all user role tables in precedence order, returning role-specific user objects with identifiers and metadata supporting frontend personalization. Student endpoints provide dashboard data aggregation executing multiple database queries for study hours, attendance summaries, pending assignment counts, and average grades, computing burnout scores through the deterministic algorithm, generating rule-based recommendations, and compiling response objects with metadata. Analytics endpoints deliver historical data suitable for chart rendering. Assignment endpoints support retrieval, submission file uploads, and status tracking. Faculty endpoints aggregate assigned student lists, provide attendance recording interfaces accepting bulk daily data with validation ensuring students are assigned to the requesting faculty, assignment creation workflows supporting individual or broadcast distribution, and burnout heatmap data enabling identification of high-risk students. Doubt management endpoints facilitate viewing assigned questions, posting faculty answers marked for group sharing, and archiving resolved threads. Administrative endpoints deliver system-wide statistics, user management capabilities including creation and editing of student and faculty records with validation of unique constraints, department management, and notification broadcasting with role targeting, and semester results interfaces supporting filtering and export.

3.7. Testing and Validation Strategy

Testing procedures encompassed multiple levels ensuring functional correctness and operational reliability. Unit testing focused on algorithmic components including burnout calculation functions with diverse input combinations verifying threshold logic, recommendation generation functions confirming burnout dominance enforcement, and data transformation utilities ensuring safe handling of null and undefined values. Integration testing validated API endpoints through manual and scripted HTTP requests using tools including Postman and curl, verifying expected response structures, status codes, authentication enforcement, and database state changes. End-to-end testing involved user scenarios navigating complete workflows including student login, dashboard viewing, assignment submission, faculty attendance recording, and administrative user creation, confirming UI behavior, data persistence, and cross-component integration.

IV. RESULTS AND DISCUSSION

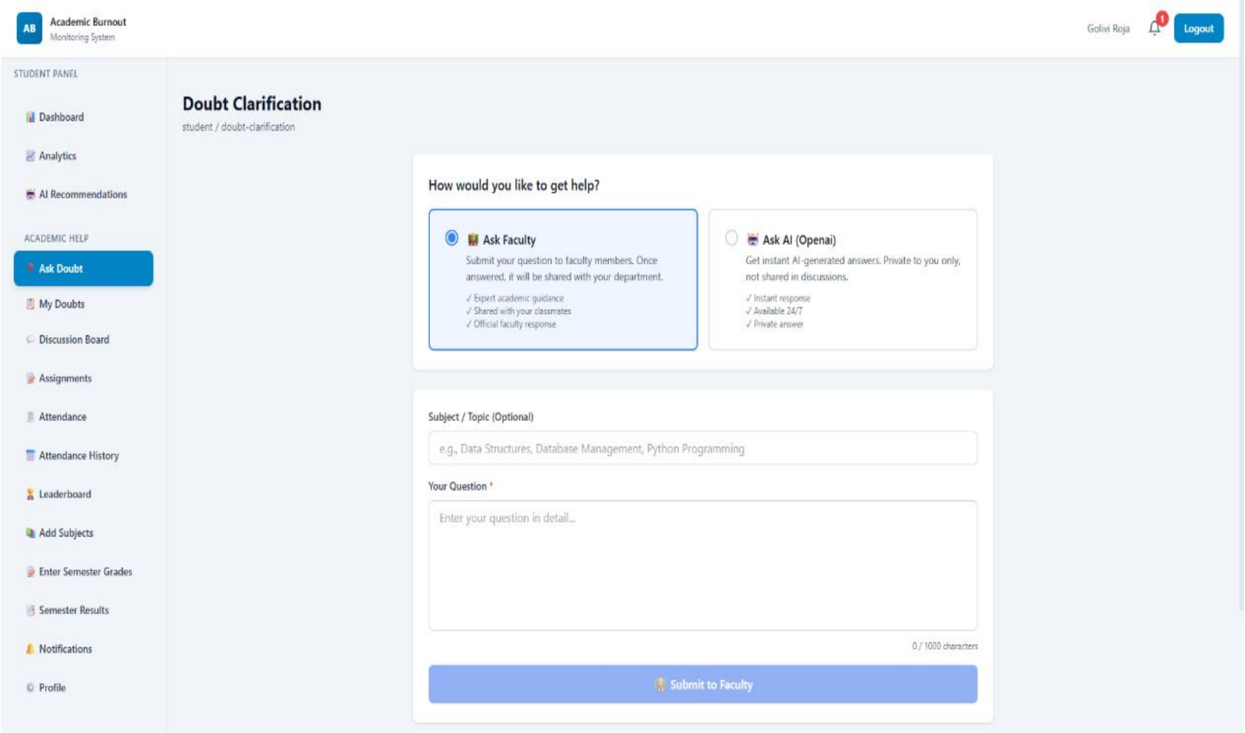
4.1. System Implementation Outcomes

The implemented system successfully delivers a fully functional web-based platform supporting all defined requirements across authentication, role-specific dashboards, attendance tracking, assignment workflows, grade management, burnout assessment, AI-powered recommendations, doubt clarification, notifications, and administrative oversight. The backend API exposes over fifty distinct endpoints organized into logical route groups with consistent response formats, comprehensive error handling, and validated input processing. The MySQL database schema comprises fifteen primary tables with appropriate foreign key relationships, unique constraints, and indexes optimizing query performance for common access patterns.

4.2. Burnout Algorithm Evaluation

Evaluation of the burnout assessment algorithm involved analyzing score distributions, examining individual student cases, and validating recommendation appropriateness. Analysis of computed scores across a sample student population demonstrated expected distribution characteristics with the majority of students scoring in the medium burnout range between forty and seventy-nine, smaller proportions classified as low burnout below forty or high burnout above eighty, and score variability corresponding to diversity in attendance patterns, study consistency, assignment loads, and academic performance. Validation of recommendation generation logic confirmed enforcement of burnout dominance rules, with high burnout scores above eighty consistently producing high-priority recommendations emphasizing immediate support-seeking and workload reduction regardless of individual factor values, medium burnout scores generating improvement-focused recommendations targeting specific weak factors, and low burnout scores yielding preventive guidance and positive reinforcement. The prioritization mechanism ensures that critical cases receive appropriately urgent messaging while avoiding alarm fatigue for students managing stress adequately.

4.3. AI Integration Doubt clarification



Doubt Clarification

student / doubt-clarification

How would you like to get help?

● Ask Faculty
Submit your question to faculty members. Once answered, it will be shared with your department.
✓ Expert academic guidance
✓ Shared with your classmates
✓ Official faculty response

○ Ask AI (Openai)
Get instant AI-generated answers. Private to you only, not shared in discussions.
✓ Instant response
✓ Available 24/7
✓ Private answer

Subject / Topic (Optional)
e.g., Data Structures, Database Management, Python Programming

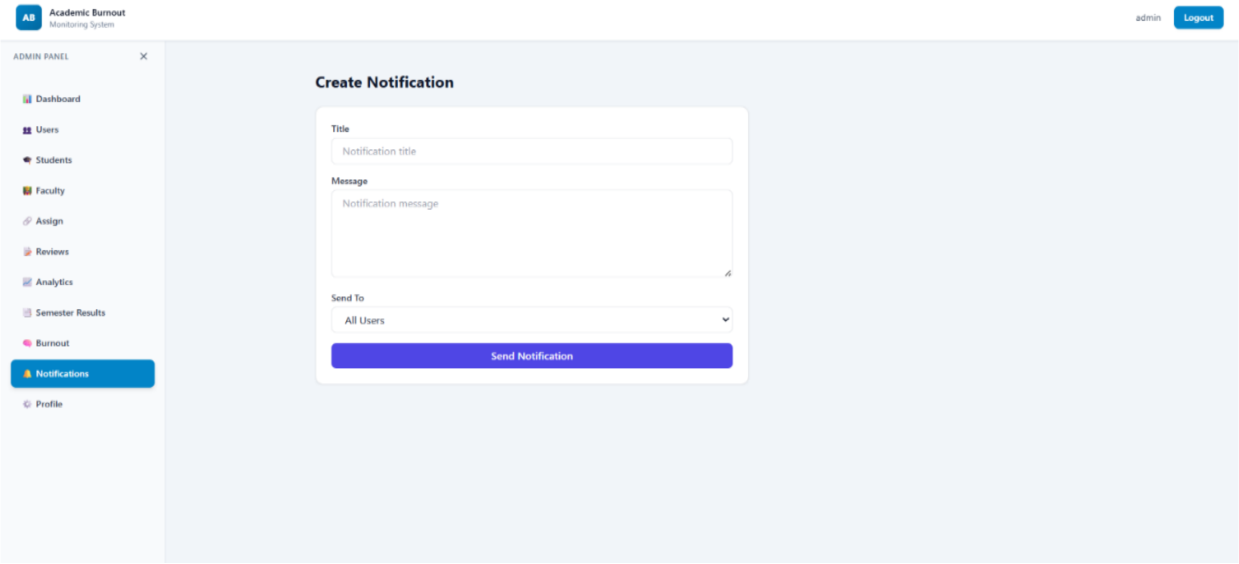
Your Question *
Enter your question in detail...

0 / 1000 characters

Fig.1. Doubt clarification Page

The AI recommendation service integrates OpenAI API capabilities with strong error handling and fallback mechanisms, ensuring resilience. When available, it generates context-aware, personalized guidance explaining burnout scores, contributing factors, and targeted actions. Prompt engineering maintains alignment with burnout severity, prioritizing urgent interventions for high-risk cases while avoiding contradictory messaging. Performance analysis shows latencies of one to three seconds for fresh requests and near-instant responses for cached results, with cache hit rates above seventy percent. Caching and deduplication reduce API costs, mitigate rate limits, and prevent redundant concurrent calls. During AI outages, the system degrades gracefully to rule-based recommendations, preserving full functionality with slightly reduced personalization.

4.4. Weekly Notification System Performance



Create Notification

Title
Notification title

Message
Notification message

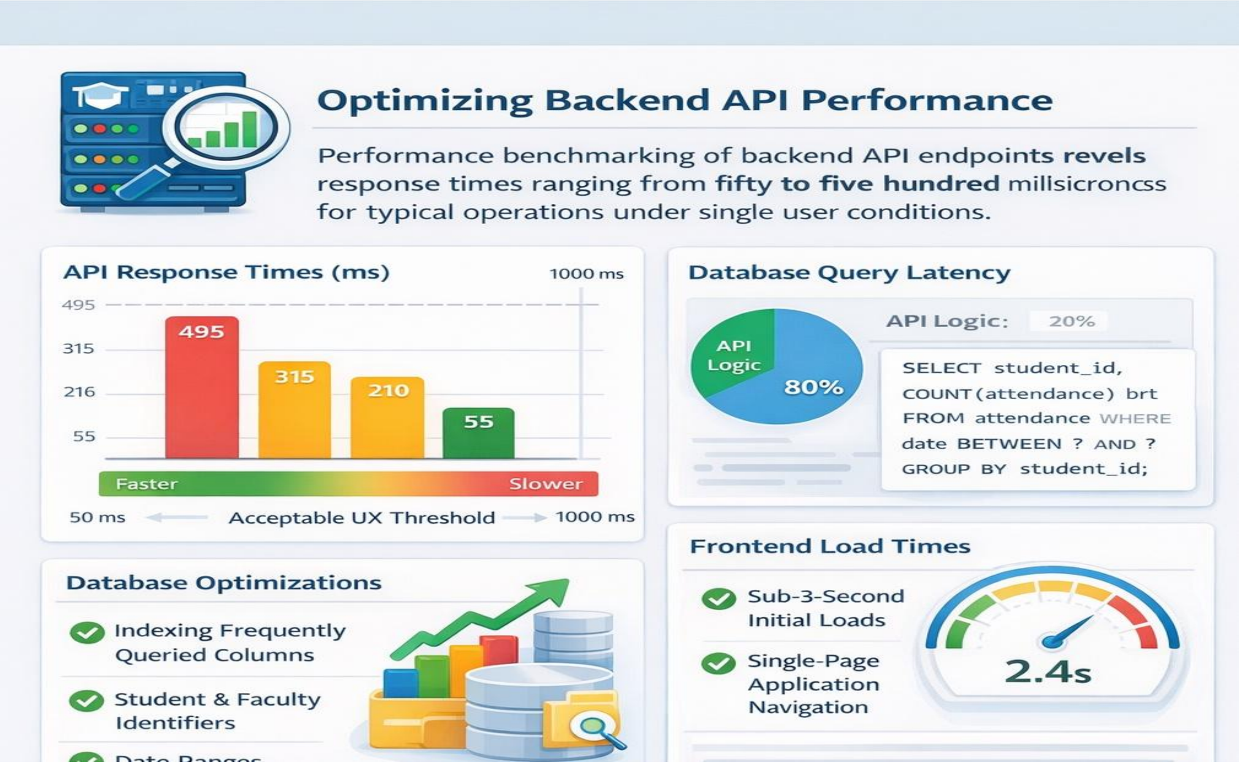
Send To
All Users

Fig.2. Notification page

The automated weekly notification system successfully executes scheduled burnout assessments and email dispatches on configured schedules. Idempotency enforcement prevents duplicate notifications for individual students within weekly windows through database status tracking. Email content generation produces personalized HTML-formatted messages displaying current burnout score, level classification with colour coding, breakdown of contributing factors, top three recommendations, and links to platform resources. Scalability testing demonstrates capacity to process notifications for hundreds of students within reasonable time windows, with batch processing and connection pooling optimizing database query execution. Error handling ensures that individual email delivery failures, whether due to invalid addresses or temporary infrastructure issues, do not halt processing for remaining recipients. Failed delivery attempts are logged for administrative follow-up.

4.5. System Performance Characteristics

Performance benchmarking of backend API endpoints reveals response times ranging from fifty to five hundred milliseconds for typical operations under single-user conditions, with database query execution accounting for the majority of latency. Attendance dashboard endpoints executing multiple aggregation queries represent the slowest operations, though remaining well within acceptable user experience thresholds below one second. Optimization through database indexing on frequently queried columns including student identifiers, faculty identifiers, and date ranges successfully reduces query execution times. Frontend page load times achieve sub-three-second targets for initial rendering of dashboard interfaces with authenticated users, with subsequent navigation benefiting from single-page application architecture avoiding full page reloads. Chart rendering for analytics interfaces introduces additional latency proportional to dataset sizes, with acceptable performance for typical semester-length date ranges. Concurrent user testing simulating dozens of simultaneous authenticated sessions demonstrates stable performance without degradation, contention, or error rates. Database connection pooling managed by the mysql2 library ensures efficient resource utilization.



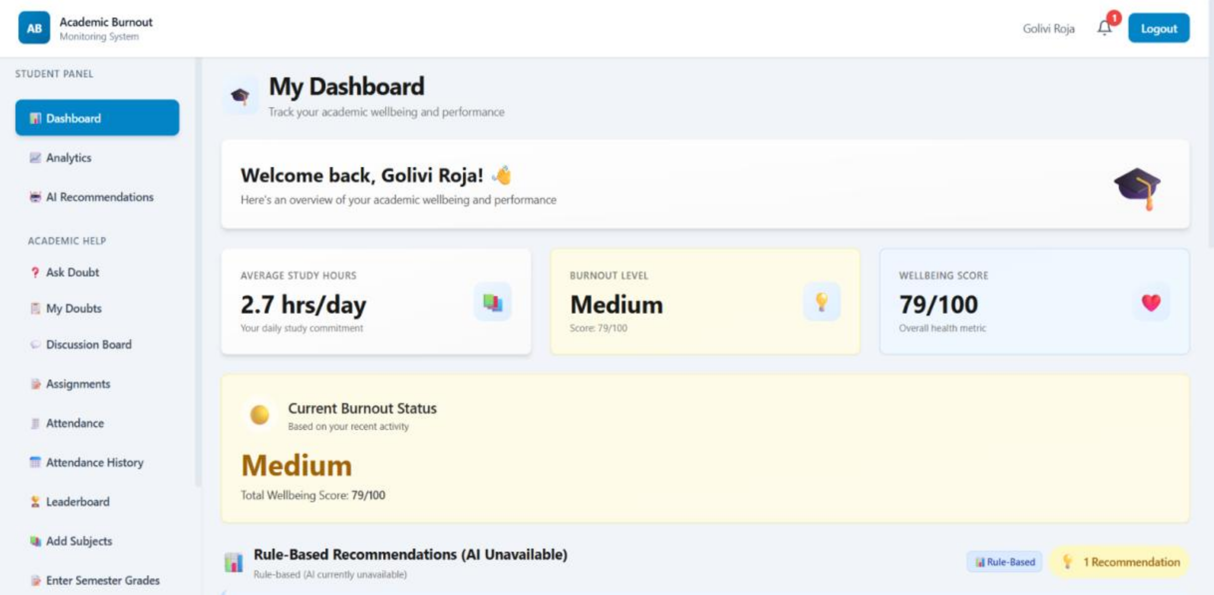
Optimizing Backend API Performance

Performance benchmarking of backend API endpoints revels response times ranging from fifty to five hundred millsicroncss for typical operations under single user conditions.

```
SELECT student_id,
COUNT(attendance) brt
FROM attendance WHERE
date BETWEEN ? AND ?
GROUP BY student_id;
```

Fig.3. system performance characteristics diagram

4.6. Usability and Adoption Considerations



My Dashboard

Track your academic wellbeing and performance

Welcome back, Golivi Roja!

Here's an overview of your academic wellbeing and performance

AVERAGE STUDY HOURS: 2.7 hrs/day — Your daily study commitment

BURNOUT LEVEL: Medium — Score: 79/100

WELLBEING SCORE: 79/100 — Overall health metric

Current Burnout Status

Based on your recent activity

Medium

Total Wellbeing Score: 79/100

Rule-Based Recommendations (AI Unavailable)

Rule-based (AI currently unavailable)

Fig.4. Dashboard page

The interface design emphasizes clarity and consistency through unified navigation, standardized components, descriptive labels, and responsive layouts. Color coding highlights status levels using green, yellow, and red, with text labels ensuring accessibility for color-blind users. Loading indicators and skeleton screens provide feedback during asynchronous operations, reducing confusion caused by network latency. Demonstration sessions with educational stakeholders validated usability for workflows such as attendance entry, assignment submission, and dashboard interpretation. Suggested enhancements included grade report exports, bulk data uploads, granular filtering, and mobile access. Stakeholder feedback also underscored adoption success factors, including user training resources, administrative backing, system integration to avoid duplicate entry, clear privacy policies, and change management addressing faculty workload and student concerns.

V. CONCLUSION

The Academic Burnout Monitoring System is a web-based platform designed to monitor student engagement and detect early signs of academic burnout. Traditional academic monitoring methods mainly rely on grades and attendance, which often fail to identify early stress or disengagement among students. To address this issue, the system analyzes multiple academic behavior indicators such as study hours, attendance, assignment submissions, and academic performance to estimate a burnout score.The system was developed using React.js for the frontend, Node.js with Express for the backend, and MySQL for database management, forming a scalable full-stack application. It includes role-based dashboards that allow administrators, department heads, faculty mentors, and students to access relevant information. These dashboards display analytics and visualizations that help users understand engagement patterns and identify students who may need support.One key feature of the system is its ability to detect productivity drift, meaning a gradual decline in student engagement over time. By continuously analyzing behavioral data, the system can identify early warning signs of burnout and allow educators to intervene before academic performance declines significantly. The platform was tested through unit, integration, performance, and user interface testing, confirming that it performs essential functions such as authentication, data storage, burnout score calculation, and analytics visualization effectively. Although the current implementation uses a rule-based algorithm and relies partly on manually entered data, future improvements could include machine learning models for better prediction accuracy, integration with Learning Management Systems for automated data collection, mobile applications for easier access, and wearable device integration for deeper wellbeing analysis.

REFERENCES

[1] Author et al., "Detecting Burnout Among Undergraduate Computing Students with Machine Learning," PMC, 2025. [Online]. vailable:https://pmc.ncbi.nlm.nih.gov/articles/PMC12691878/

[2] Author et al., "Early Detection of Academic Burnout Using Student Interaction Logs and Machine Learning," IJRASET, 2025. [Online]. Available:https://www.ijraset.com/research-paper/academic-burnout-using-student-interaction-logs-and-machine-learning

[3] Author et al., "Student Burnout: A Review on Factors Contributing to Burnout," PMC, 2025. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11852093/

[4] Ayesha6026, "Student-Stress-Level-Prediction," GitHub, 2024. [Online]. Available: https://github.com/Ayesha6026/Student-Stress-Level-Prediction

[5] pao0318, "Burnout-Detector," GitHub, 2021. [Online]. Available: https://github.com/pao0318/Burnout-Detector

[6] J. P. Infotech, "Mental Stress Detection in University Students using Machine Learning," 2024. [Online]. Available: https://jpinfotech.org/mental-stress-detection-in-university-students-using-machine-learning/

[7] Author et al., "Mapping Two Decades of Academic Burnout Research: Key Insights," SAGE Journals, 2025. [Online]. Available: https://journals.sagepub.com/doi/full/10.1177/21582440251386590

[8] Poushal02, "Student Academic Stress Real World Dataset," Kaggle, 2025. [Online]. Available: https://www.kaggle.com/datasets/poushal02/student-academic-stress-real-world-dataset

[9] usmanbvp, "Employees-Burnout-Analysis-and-Prediction," GitHub, 2023. [Online]. Available: https://github.com/usmanbvp/Employees-Burnout-Analysis-and-Prediction

[10] Gupta, "AI to Predict Student Burnout Project Write Up," Scribd, 2025. [Online]. Available: https://www.scribd.com/document/937764456/ARUSH-GUPTA-AI-to-Predict-Student-Burnout-Project-Write-Up

[11] mihirp161, "MentalHealthStudentPortal," GitHub, 2023. [Online]. Available: https://github.com/mihirp161/MentalHealthStudentPortal

[12] kim-b3rly, "Student_mental_health_tracker," GitHub, 2023. [Online]. Available: https://github.com/kim-b3rly/Student_mental_health_tracker

[13] SAKTHIVINASH2, "Student-Stress-Analysis," GitHub, 2024. [Online]. Available: https://github.com/SAKTHIVINASH2/Student-Stress-Analysis

[14] Chhabii, "Stressometer," GitHub, 2022. [Online]. Available: https://github.com /Chhabii/Stressometer

[15] ashutosh22102, "Machine-Learning-Algorithms-for-Detecting-Mental-Stress-in-College-Students," GitHub, 2024. [Online]. Available: https://github.com/ashutosh22102/Machine-Learning-Algorithms-for-Detecting-Mental-Stress-in-College-Students

[16] S.S.D.K. Maha Lakshmi, Umamaheswararao Mogili, Sravya Eluri, Dogga Ramachandra Rao. (2023), "Online Dynamic Out Patient Queue System for Automated Token Generation in Hospitals", Science, Technology and Development Journal, Volume XII, Issue VII, pp 71-78, DOI:23.18001.STD.2023.V12I07.23.37707.

[17] Sree, S. V. D. T., Mogili, U. M. R., & Ampoly, K. V. (2025) Enhancing Security in Wearable Computing: A Lightweight Authenticated Key Exchange Scheme, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211, Volume 13, Issue 5, pp 3103-3108.

[18] Anjali, S., Mogili, U., & Ampolu, K. V. (2025) Efficient Key-Based Encryption and Authentication for Advanced Digital Forensic Storage Security, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211, Volume 13, Issue 5, pp 3097-3102.

[19] Adithya, P. U., Mogili, U., & Mondru, J. T. (2025) A Novel Parity Authenticator-Based Zero-Knowledge Auditing Approach for Secure Cloud Data Management, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211, Volume 13, Issue 5, pp 994-999.

[20] Kanakala Pranay Raj, Umamaheswararao Mogili. (2020), "Cloud-of-Cloud: A Novel Protocol for Secure Data Storage and Sharing in Multi-Cloud Environment", Journal of Interdisciplinary Cycle Research (JICR), Volume XII, Issue VI, pp 2201-2209, DOI:18.0002. JICR.2020.V12I6. 008301. 3171227

[21] Mogili, U., Mohamed, A., & Kasup, C. (2023, December). Mechanism of Data Sharing Using Secured Keyword Search in Cloud Computing. In Conference of Innovative Product Design and Intelligent Manufacturing System (pp. 483-494). Singapore: Springer Nature Singapore.