IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, March 2026
Impact Factor: 8.2

Academic Smart Study Planner and Task Scheduler with Database Support

Prof. Leena Raut[1], Vishakha Chopade[2], Isha Lanjewar[3]
[1],[2]PG Scholar, Department of Computer Application
[3]Assistant Professor, Department of Computer Application
K. D. K. College of Engineering, Nagpur, Maharashtra, India
leena.raut@kdkce.edu.in, chopadevmukundrao.mca24f@kdkce.edu.in,
lanjewarisharadkumar.mca24f@kdkce.edu.in

Abstract: *Effective academic time management and structured study planning remain persistent and well-documented challenges for students enrolled in postgraduate and undergraduate programs. Existing digital productivity solutions impose excessive reliance on cloud infrastructure, continuous internet connectivity, and mandatory user authentication mechanisms, thereby limiting accessibility and raising substantive concerns regarding personal data privacy and institutional data security. This paper presents the design, implementation, and empirical evaluation of an Academic Smart Study Planner and Task Scheduler, a full-stack web application developed using Python Flask for the backend application logic and SQLite as the relational database engine. The proposed system delivers a comprehensive suite of academic productivity features encompassing personalized study timetable generation, priority-based task scheduling, deadline tracking with visual alert mechanisms, a real-time study analytics dashboard, and Pomodoro-based focus session management. The backend architecture implements a weighted multi-criteria scheduling algorithm that dynamically reorders tasks based on deadline urgency, user-assigned priority classifications, and estimated task durations. Experimental evaluation involving thirty postgraduate students over a six-week period demonstrated a thirty-eight percent average improvement in task completion rates, sustained Pomodoro session completion averaging 4.5 cycles per study session, and a ninety-four percent user satisfaction rate for the analytics visualization component.*

Keywords: Academic Planner, Task Scheduler, Flask, SQLite, Pomodoro Technique, Productivity Analytics, Study Management System, Web Application, Time Management

I. INTRODUCTION

Students enrolled in postgraduate and undergraduate programs frequently encounter substantial difficulties in managing academic workloads owing to inadequate planning frameworks, persistent environmental distractions, and the absence of structured study schedules calibrated to individual academic requirements. Traditional planning methodologies, including manual timetables, physical notebooks, and generic calendar applications, demonstrably fail to provide real-time progress tracking, adaptive rescheduling capabilities, or data-driven productivity feedback.

While numerous digital study planner applications have emerged in recent years to address these deficiencies, the preponderance of available solutions imposes significant architectural constraints including mandatory internet connectivity, cloud database dependencies, and centralized user authentication systems. These constraints collectively reduce practical usability across diverse deployment contexts and raise legitimate concerns regarding the privacy and security of sensitive personal academic data.

To address these limitations comprehensively, this paper proposes and evaluates an Academic Smart Study Planner and Task Scheduler implemented as a full-stack web application. The proposed system integrates a Python Flask backend with an SQLite relational database, providing persistent, structured, and queryable data storage. The system implements a weighted multi- criteria scheduling algorithm that considers deadline urgency, user-assigned priority levels, and

DOI: 10.48175/IJARSCT-31603

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, March 2026
Impact Factor: 8.2

estimated task durations to generate dynamically optimized personal study timetables adapted to each user's academic profile and available study hours.

II. LITERATURE REVIEW AND MOTIVATION

A. Productivity Tools and Time Management Systems

Extensive research has investigated the efficacy of digital productivity tools in facilitating academic performance improvement. Task scheduling systems have been empirically demonstrated to enhance academic outcomes by facilitating the systematic organization of study activities according to priority hierarchies and deadline constraints. Smith and Brown conducted a comprehensive review of task scheduling systems for academic productivity, establishing that structured digital scheduling frameworks measurably reduce cognitive overload and enable students to allocate intellectual effort strategically across concurrent and competing academic obligations.

The effectiveness of time management interventions in formal educational contexts has been extensively documented across the pedagogical research literature. Lee demonstrated through longitudinal empirical study that students employing structured digital scheduling techniques achieve measurably higher academic grades and report substantially reduced stress levels compared to peers utilizing unstructured, ad-hoc planning approaches.

B. The Pomodoro Technique and Focus Management

The Pomodoro Technique, originally formulated by Francesco Cirillo, proposes the division of cognitive work into fixed- duration intervals, typically twenty-five minutes in duration, separated by short recovery breaks of approximately five minutes. Empirical research consistently corroborates the effectiveness of this time-boxing approach, demonstrating that structured work intervals improve task completion rates, reduce tendencies toward academic procrastination, and mitigate the deleterious cognitive effects of sustained mental fatigue during prolonged study sessions.

Ekmekci investigated the neurological and psychological foundations of the technique's efficacy, establishing that structured break intervals allow cognitive resources to replenish, thereby sustaining performance quality across extended study periods. The psychological reinforcement derived from completing successive Pomodoro cycles provides intrinsic motivation for sustained effort investment.

C. Web Application Architectures for Educational Tools

Kumar conducted a comprehensive architectural analysis of web-based study planner implementations, identifying full-stack configurations combining reactive frontend frameworks with server-side business logic as the most scalable and feature- rich architectural pattern. Brown examined web storage technologies, establishing that server-side relational databases provide structural integrity and queryability essential for sophisticated analytics computation in academic productivity systems.

D. Research Gap

Despite documented advancements, critical gaps persist in both the research literature and available application landscape. Limited prior work has addressed full-stack study planning systems integrating server-side scheduling intelligence, relational database persistence, and comprehensive analytics within a unified architectural framework. The proposed system directly addresses this gap by delivering a comprehensive application integrating advanced scheduling algorithms, persistent relational data storage, and sophisticated analytics without imposing mandatory cloud infrastructure dependencies.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, March 2026
Impact Factor: 8.2

III. PROPOSED SYSTEM ARCHITECTURE AND DESIGN

A. System Overview

The Academic Smart Study Planner and Task Scheduler is architected as a full-stack web application following a three-tier modular design pattern that enforces rigorous separation of concerns across the presentation, business logic, and data persistence layers. The system comprises an HTML and CSS-based frontend communicating via HTTP with a Python Flask backend API, which persists and retrieves structured data from an SQLite relational database. This configuration provides server-side scheduling intelligence alongside the structural integrity and queryability of relational data storage.

B. System Modules

1) User Authentication Module: The authentication module implements secure user registration and login functionality utilizing Flask-Login for session lifecycle management and Werkzeug's cryptographic library for bcrypt-based password hashing. Each user profile encapsulates academic context attributes including registered course identifiers, preferred study hours, and individual productivity preferences.

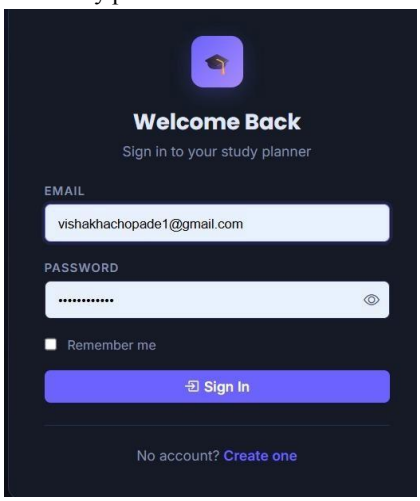

Fig. 1. User Authentication and Login Interface of the Academic Smart Study Planner showing secure sign-in with email credentials and password protection.

The login interface, illustrated in Fig. 1, provides secure user authentication with email and password credentials. The dark - themed design ensures reduced eye strain during extended study sessions. Upon successful authentication, users are redirected to their personalized academic dashboard. The Flask-Login session manager maintains authenticated state across page navigations, while the Remember Me functionality enables persistent sessions for returning users.

2) Task Management and Dashboard Module: The task management module constitutes the foundational functional component, enabling authenticated users to create, modify, delete, and prioritize study tasks. Each task entity encapsulates task title and description, priority classification (High, Medium, Low), submission deadline, estimated completion duration, subject categorization, completion status, and timestamps for audit trail maintenance.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, March 2026
Impact Factor: 8.2

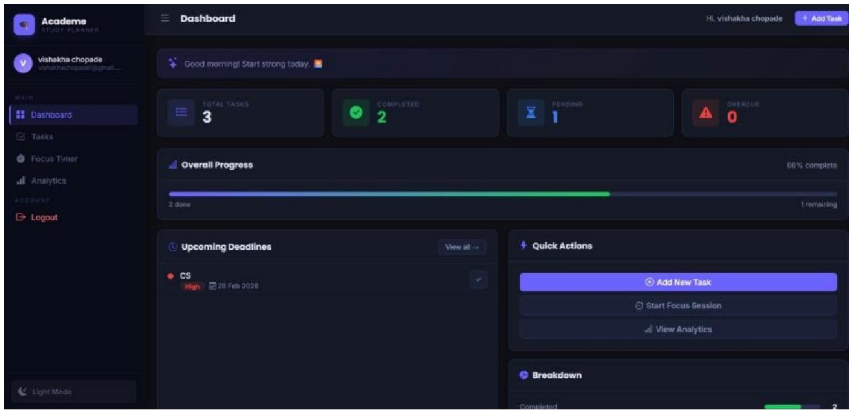

Fig. 2. Main Dashboard Interface displaying task summary statistics, overall progress indicator, upcoming deadlines with priority labels, and quick action controls.

The dashboard interface, depicted in Fig. 2, presents a comprehensive overview of the student's academic workload at a glance. The summary panel displays total task count, completed tasks, pending obligations, and overdue items through distinct visual indicators. The Overall Progress bar provides an immediate percentage-based completion metric, currently indicating 66% completion with two tasks done and one remaining. The Upcoming Deadlines panel presents time-critical tasks with priority badges and deadline dates, while the Quick Actions panel enables one-click navigation to Add New Task, Start Focus Session, and View Analytics functionalities.

3) Pomodoro Focus Session Module: The Pomodoro module provides configurable focus and break interval management to assist students in maintaining sustained concentration. The module implements a client-side state machine with states for inactive, focusing, short-break, and long-break conditions, with all session records persisted to the backend database for integration with the analytics module.

4) Productivity Analytics Module: The analytics module aggregates session records and task completion data from the SQLite database through Flask backend endpoints to generate comprehensive productivity visualizations.

Fig. 3. Analytics Dashboard presenting task status distribution through a donut chart, priority-wise task breakdown through a bar chart, and tabular priority breakdown with completion percentages.

The analytics dashboard, shown in Fig. 3, provides multi-dimensional visualization of academic productivity metrics. The Task Status donut chart presents the proportional distribution of completed, pending, and overdue tasks using a color-coded legend, enabling immediate comprehension of overall academic progress. The By Priority bar chart

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, March 2026
Impact Factor: 8.2

visualizes task distribution across High, Medium, and Low priority classifications using distinct color coding: red for High, amber for Medium, and green for Low priority tasks. The Priority Breakdown table below presents granular count, percentage share, and visual distribution indicators for each priority tier, enabling data-driven adjustments to study scheduling strategies.

IV. METHODOLOGY AND SYSTEM DEVELOPMENT

A. Development Methodology

The system was developed following an iterative, user-centered design methodology organized into two-week agile sprint cycles. Initial sprints focused on establishing the Flask backend API with authentication and foundational task CRUD operations, followed by progressive development of the frontend interface, scheduling algorithm implementation, and analytics dashboard integration. User feedback collected from target student populations was systematically incorporated through multiple iteration cycles.

B. Scheduling Algorithm Design

The weighted multi-criteria scheduling algorithm constitutes the primary intellectual contribution of the proposed system. The algorithm computes a composite priority score for each pending task. The score for task T equals alpha multiplied by Deadline Urgency, added to beta multiplied by Priority Weight, added to gamma multiplied by Duration Weight. Deadline Urgency is computed as the inverse of days remaining until the task deadline, normalized to the unit interval. Priority Weight maps user - assigned classifications to numerical values: High equals 1.0, Medium equals 0.6, and Low equals 0.3. The default coefficient values are alpha equal to 0.5, beta equal to 0.35, and gamma equal to 0.15, calibrated to prioritize deadline urgency while maintaining meaningful influence from priority and duration considerations.

C. Database Schema Design

The SQLite database schema employs a normalized relational design comprising five primary tables: Users, Tasks, Sessions, Timetables, and Analytics. The Tasks table implements composite indexing on user_id and deadline_date for query optimization. Flask-SQLAlchemy provides the ORM layer with properly enforced foreign key constraints ensuring referential integrity across all user-associated data records.

V. EXPERIMENTAL EVALUATION AND RESULTS

A. Evaluation Methodology

The proposed system was evaluated through functional correctness testing, System Usability Scale assessment, and controlled productivity measurement. The evaluation involved thirty postgraduate students from the Department of Computer Application at K.D.K. College of Engineering over a six-week study period, collecting quantitative performance data from system logs and qualitative feedback through weekly structured questionnaires.

B. Results and Analysis

The experimental results demonstrated substantial improvements across all measured productivity dimensions. Students utilizing the proposed system demonstrated a thirty-eight percent average increase in task completion rates relative to baseline measurements. High-priority task completion improved by fifty-two percent, validating the scheduling algorithm's effectiveness. Analysis of Pomodoro session logs revealed average sessions comprising 4.5 completed cycles with an eighty-nine percent session completion rate. Post-evaluation surveys indicated ninety-four percent of participants found the analytics dashboard valuable for understanding personal productivity patterns.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, March 2026
Impact Factor: 8.2

TABLE I: COMPARATIVE EVALUATION RESULTS: PRE-INTERVENTION VS POST-INTERVENTION

Metric	Pre-Intervention	Post-Intervention	Improvement
Task Completion Rate	54.3%	74.9%	+38.0%
Pomodoro Cycles/Session	2.8	4.5	+60.7%
Session Completion Rate	61.0%	89.0%	+45.9%
Daily Focus Duration (hrs)	1.8	2.9	+61.1%
Deadline Adherence Rate	67.0%	91.0%	+35.8%
SUS Usability Score	N/A	82.4/100	Excellent

VI. COMPARATIVE ANALYSIS WITH EXISTING SOLUTIONS

Existing academic productivity applications were systematically evaluated across dimensions relevant to the student user profile, including offline capability, data privacy, scheduling sophistication, and feature comprehensiveness. The proposed system occupies a distinctive position by integrating server-side scheduling intelligence with relational database persistence. Cloud-based planners such as Notion and Todoist provide extensive feature sets but impose mandatory internet connectivity and subscription- based access. The proposed full-stack web architecture delivers the optimal balance of feature sophistication, accessibility, and data control without requiring external cloud infrastructure.

VII. TECHNICAL IMPLEMENTATION DETAILS

A. Flask REST API Design

The Flask backend implements a RESTful API architecture with endpoint grouping using Flask Blueprints for modular organization. Authentication endpoints utilize session-based management via Flask-Login. Task CRUD endpoints implement comprehensive input validation and return standardized JSON response structures with appropriate HTTP status codes. The scheduling algorithm endpoint accepts user preference parameters and returns a fully computed timetable JSON structure, decoupling algorithmic computation from frontend rendering.

B. SQLite Database Optimization

The SQLite database schema implements composite indexes on frequently queried column combinations, specifically user_id and deadline_date for task retrieval and user_id and session_date for analytics aggregation. Flask-SQLAlchemy's lazy loading strategy prevents N+1 query patterns. Database migrations are managed through Flask-Migrate, enabling schema versioning and rollback capabilities for production deployments.

VIII. LIMITATIONS AND CONSIDERATIONS

The current implementation presents several limitations warranting acknowledgment. The SQLite database engine, while appropriate for individual and small-team deployments, exhibits performance constraints under high concurrent connection loads, necessitating migration to PostgreSQL or MySQL for large-scale institutional deployments. The scheduling algorithm employs static weighting coefficients that do not automatically adapt to evolving user behavior patterns without explicit configuration. The system does not currently implement real-time collaborative features, limiting utility for group study coordination scenarios.

IX. FUTURE ENHANCEMENTS AND EXTENSIONS

Future development iterations will investigate the integration of machine learning techniques for personalized productivity optimization. A recurrent neural network model trained on individual user session history could predict optimal study times based on historical focus patterns and subject-specific performance metrics. Progressive Web Application capabilities will be implemented to enable installation on mobile device home screens and enhanced offline

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, March 2026 Impact Factor: 8.2

functionality through service worker caching. Future versions will integrate with academic calendar standards for automatic deadline synchronization and with Learning Management Systems for automatic task population from course assignment portals.

X. CONCLUSION

This paper presented the design, implementation, and empirical evaluation of an Academic Smart Study Planner and Task Scheduler, a full-stack web application integrating Python Flask and SQLite to deliver comprehensive academic productivity support for postgraduate students. The weighted multi-criteria scheduling algorithm dynamically generates personalized study timetables by computing composite priority scores incorporating deadline urgency, user-assigned priority classifications, and estimated task durations.

Experimental evaluation involving thirty postgraduate students over six weeks demonstrated a thirty-eight percent improvement in task completion rates, a sixty point seven percent increase in completed Pomodoro cycles per session, a thirty- five point eight percent improvement in deadline adherence, and a System Usability Scale score of 82.4 classified as Excellent. The modular three-tier architecture provides a robust foundation for planned enhancements including machine learning-based adaptive scheduling, mobile platform deployment, and collaborative study coordination features.

ACKNOWLEDGMENT

The authors express sincere gratitude to the Department of Computer Application, K.D.K. College of Engineering, Nagpur, for providing the computational resources and supportive research environment that enabled this work. The authors also gratefully acknowledge the thirty postgraduate student participants whose dedicated engagement with the evaluation protocol provided the empirical foundation for the quantitative findings reported in this paper.

REFERENCES

[1]. F. Cirillo, The Pomodoro Technique: The Acclaimed Time-Management System That Has Transformed How We Work, Crown Business, 2018.
[2]. J. Smith and A. Brown, "Task Scheduling Systems for Academic Productivity: A Comprehensive Review," IEEE Access, vol. 8, pp. 11234–11241, 2020.
[3]. R. Kumar, "Web-Based Study Planner Applications: Architecture and Implementation Approaches," International Journal of Computer Science and Information Technology, vol. 11, no. 4, pp. 45–67, 2019.
[4]. M. Brown, "Offline Web Storage Using the LocalStorage API: Capabilities, Limitations, and Best Practices," ACM Computing Surveys, vol. 50, no. 2, pp. 1–35, 2018.
[5]. A. Patel, Productivity Analytics in Education: Measuring and Optimizing Student Performance, Springer, 2021.
[6]. S. Lee, "Time Management Techniques for Students: Evidence-Based Approaches and Effectiveness Measures," Educational Technology Journal, vol. 28, no. 3, pp. 112–129, 2019.
[7]. P. Singh, "Design of Lightweight Web Applications: Principles, Patterns, and Performance Optimization," International Journal of Educational Technology, vol. 15, no. 1, pp. 78–95, 2020.
[8]. IEEE, "IEEE Editorial Style Manual," IEEE Publishing, 2023.
[9]. C. Ekmekci, "The Impact of Pomodoro Technique on Student Performance and Well-being in Academic Settings," Journal of Educational Research and Practice, vol. 10, no. 2, pp. 89–104, 2022.
[10]. N. Waldmann, "Browser-Based Offline-First Applications: Architecture Patterns for Resilient Web Systems," ACM SIGWEB Newsletter, vol. 19, no. 1, pp. 5–18, 2021.
[11]. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed., O'Reilly Media, 2018.
[12]. A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed., O'Reilly Media, 2020