INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 10 ISSUE: 03 | MARCH - 2026 SJIF RATING: 8.659 ISSN: 2582-3930

# PlanAIr-AI-Based Smart Productivity and Task Planning Web Application

**Girsh Dayaghan Sakpal**
Department of Artificial Intelligence and Data Science
Rizvi College of Engineering, Bandra
Mumbai, India
girishdsakpal@gmail.com

**Pilwalkar Aryan Prakash**
Department of Artificial Intelligence and Data Science
Rizvi College of Engineering, Bandra
Mumbai, India
aryanpilwalkar@gmail.com

**Kaif Reshamwala**
Department of Artificial Intelligence and Data Science
Rizvi College of Engineering, Bandra
Mumbai, India
kaifreshamwala@eng.rizvi.edu.in

Guide

**Prof. Ramkumar Maurya**
Department of Artificial Intelligence and Data Science
Rizvi College of Engineering, Bandra
Mumbai, India
ram@eng.rizvi.edu.in

Guide

**Prof. Junaid Mandviwala**
Department of Artificial Intelligence and Data Science
Rizvi College of Engineering, Bandra
Mumbai, India
junaid@eng.rizvi.edu.in

***Abstract*** - PlanAIr is a web-based smart productivity and task planning application designed to bridge the gap between simple task listing and intelligent, personalized scheduling. By integrating scientific frameworks like the Eisenhower Matrix for prioritization and custom rule-based algorithms for time-blocking, the system automates the creation of a realistic daily work plan. A unique feature of PlanAIr is its emotional awareness, where daily mood check-ins dynamically adjust session lengths and workload intensity to match the user's cognitive state. Developed using Python, Flask, and SQLite, the application includes a comprehensive insights dashboard for tracking productivity trends. Results demonstrate that rule-based AI can effectively manage complex scheduling constraints, providing users with a robust tool for structured life planning.

***Keywords -*** AI Scheduling, Productivity, Eisenhower Matrix, Task Planning, Flask, Web Application, Emotional Awareness, Time Management.

## I. INTRODUCTION

In today's fast-paced world, effective time management is a critical yet elusive skill. While digital to-do lists are ubiquitous, most offer little more than static checkboxes with due dates, failing to account for a user's actual capacity or emotional state. Users frequently struggle with "decision fatigue" the inability to determine which task to tackle first often resulting in the neglect of important, non-urgent long-term goals [1].

PlanAIr was developed to solve these systemic planning failures. By combining the Eisenhower Matrix with intelligent time-blocking, the application transforms a raw list of tasks into a structured, executable schedule [1],[2]. The core objective is to provide a "human-centric" tool that adapts to the person, rather than demanding the person adapt to a rigid tool.

## II. KEY CONCEPTS

1. The Eisenhower Matrix Framework:

The core prioritization engine of PlanAIr is based on the Urgent-Important Matrix popularized by Stephen Covey [1]. The system categorizes every user task into one of four distinct quadrants based on urgency and importance scores ranging from 1 to 4:

- Quadrant 1 (Do Now): Urgent and important tasks requiring immediate attention, such as deadlines or crises.
- Quadrant 2 (Schedule): Important but not urgent tasks that contribute to long-term goals and personal development.
- Quadrant 3 (Delegate): Urgent but not important tasks, often consisting of interruptions.
- Quadrant 4 (Avoid): Tasks that are neither urgent nor important and should be minimized. PlanAIr automates this classification to guide users toward high-value activities typically found in Quadrant 2.

Diagram:


Fig 2.1 Eisenhower Matrix

2. Automated Time Blocking & Cognitive Load Management:

PlanAIr employs time blocking to reduce decision fatigue by reserving specific time slots for dedicated focus [2]. The system divides available free time into discrete sessions capped at a maximum of 120 minutes to prevent cognitive fatigue. To ensure recovery, 15-minute break buffers are automatically inserted between sessions, avoiding back-to-back scheduling. This structured distribution ensures complex tasks are handled in manageable chunks without diminishing returns [2].

3. Emotional Awareness in Productivity:
The application integrates behavioral science by acknowledging that emotional states directly influence a user's cognitive performance [3]. Daily mood check-ins allow the system to dynamically adjust workload intensity to match the user's mental energy. Low mood reports (scores 1 or 2) trigger shorter sessions and deprioritize non-essential tasks for that day. This human-centric approach ensures the planning tool adapts to the person rather than demanding rigid adherence.

4. Intelligent Priority & Slot-Finding Logic:
A dual-layer algorithm determines the work sequence, utilizing a deadline-first sorting method to protect time-critical tasks. Secondary priority is computed using Eisenhower quadrant weights, granting "Do Now" tasks a base score of 100. The slot-finder merges overlapping busy hours and clips the schedule to the current time for mid-day runs. This logic prevents non-deadline tasks from consuming capacity needed for urgent, high-priority commitments.

5. Multi-Factor Productivity Scoring:
To provide granular insights, PlanAIr moves beyond binary completion by using a weighted productivity formula. The score is composed of session completion (40%), important task focus (30%), and adherence to the original schedule (20%). A final 10% is awarded as a streak bonus to incentivize consistent daily engagement with the platform. These metrics are capped at 100 and stored to visualize long-term productivity trends in the insight dashboard.

## III. METHODOLOGY

1. Development Setup:

1.1 Technologies Used: The application utilizes Python and Flask for backend development, enabling the implementation of complex scheduling logic and a modular REST API [4],[7]. SQLite with Flask-SQLAlchemy serves as the relational database engine for data persistence and relationship management [5],[8]. The frontend is constructed using HTML, CSS, and JavaScript, with Chart.js integrated for rendering interactive productivity analytics [6].
1.2 System Architecture: PlanAIr follows a Model-View-Controller (MVC) architecture organized through Flask Blueprints. The system is divided into six primary layers: the database layer (SQLite), the model layer (SQLAlchemy classes), the business logic layer (scheduler and engines), the route layer (Blueprints), the template layer (Jinja2), and the presentation layer (CSS/JS).
1.3 Database Design: The application employs a relational schema featuring seven tables with user_id foreign keys to ensure data isolation. Key tables include users for authentication, busy_hours for unavailable time blocks, tasks for quadrant definitions, and schedule_sessions for tracking work blocks

2. Procedures Adopted:

The development of the PlanAIr platform followed these core procedural steps:

2.1 Task Prioritization: User tasks are categorized using the Eisenhower Matrix, where urgency and importance scores (1-4) determine the task's quadrant and base priority.
2.2 Deadline-First Sorting: The system implements a primary sort by deadline proximity, ensuring time-critical tasks are prioritized for available slots.
2.3 Availability Mapping: The algorithm identifies free time by merging user-defined busy hours and clipping the schedule to the current time to prevent past-time scheduling.
2.4 Session Generation: The scheduler iterates through a 14-day window, creating ScheduleSession records that respect a 120-minute maximum length and 15-minute break buffers
2.5 Mood-Aware Adjustment: Upon a mood update, the system automatically triggers a background regeneration to reduce workload intensity or session duration based on the reported emotional state.
2.6 Progress Tracking: Users log actual work hours via a session modal, which updates the task's completed_hours and triggers auto-completion when estimates are met.
2.7 Analytics & Suggestions: The system calculates daily productivity scores and generates rule-based suggestions to flag overdue tasks or schedule overloads.

3. Algorithms and Models:

3.1 Priority Scoring Algorithm:
The numeric score is determined by the Eisenhower quadrant: Do Now (100 pts), Schedule (75 pts), Delegate (50 pts), and Avoid (25 pts). An additional urgency boost of up to 50 points is applied for tasks due within seven days.
3.2 Scheduling Logic: The core engine follows a rule-based sequence:

1. Clear all future uncompleted sessions.
2. Sort pending tasks by the priority algorithm.
3. For each task, find the first valid time slot that fits within available minutes and respects mood-based session caps.
4. Commit sessions and insert mandatory 15-minute recovery buffers.

3.3 The daily score is a weighted calculation across four performance metrics:

$$\text{Score} = (0.4 * S) + (0.3 * I) + (0.2 * A) + (0.1 * B)$$

Where S is session completion, I is important task completion, A is schedule adherence, and B is the streak bonus [1],[2].
3.4 Rule-Based Suggestion Engine:
The engine evaluates six distinct rules, including an Overload Rule (triggered at >6 hours scheduled) and a Neglect Guard (identifying important tasks with no progress for 3 days).

Diagram:


Fig 3.1 Procedure Adopted


Fig 3.2 Eisenhower matrix

## IV. RESULTS

4.1 Usability and User Experience:
The initial testing of PlanAIr was conducted through an interactive web platform that prioritizes a clean, modern interface. The application includes a fully integrated light and dark mode toggle, where CSS variable overrides ensure that all elements including charts, matrix cells, and suggestion cards adapt to the selected theme.
Testing across Chrome, Edge, and Firefox on Windows confirmed that Chart.js components successfully detected theme attributes to adjust grid line and text colours [6]. Furthermore, the mobile responsive design was verified at viewport widths of 375px, 768px, and 1440px, with navigation menus and grid layouts collapsing correctly into single-column formats for smaller screens.
4.2 Functionality and Performance
PlanAIr successfully implemented all major features across six development phases. The core scheduling algorithm was stress-tested with task sets ranging from one to eight concurrent tasks.
Key performance observations include:

- Deadline Priority: Time-critical tasks were consistently scheduled before flexible tasks, preventing urgent work from being displaced.
- Buffer Accuracy: The 15-minute break buffer was correctly applied between consecutive sessions, and the current-time clipping mechanism successfully prevented the system from scheduling sessions in the past.
- Mood-Aware Scaling: At mood level 1 ("Rough"), the system successfully reduced session lengths to 45 minutes, while at levels 4 and 5 ("Good/Great"), sessions were extended to 150 minutes for deep work.
- Progress Tracking: Session logging was verified with a 38-hour task, where the progress bar correctly incremented from 0% to 5.3% after the first session and automatically marked the task as "done" once estimated hours were reached.

4.3 Data-Driven Insights
The platform provides a comprehensive Insights Dashboard that renders five distinct chart types to facilitate informed productivity decisions:

- Line Charts: Visualize weekly productivity and mood trends over time.
- Stacked Bar Charts: Display task completion data for objective performance review.
- Horizontal Bar Charts: Analyze productive hours to identify peak performance windows.
- Doughnut Charts: Contrast workload distribution across the four Eisenhower quadrants.

Diagram:


Fig 4.1 PlanAIr User Interface

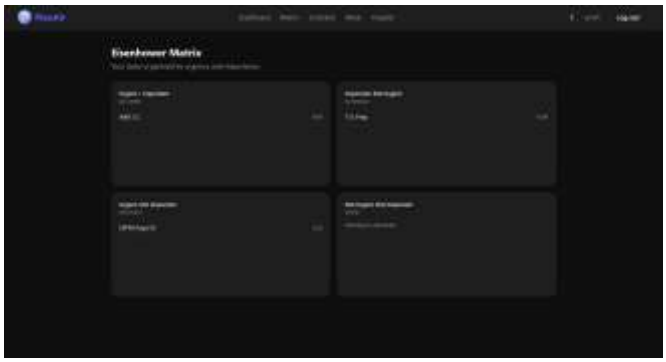


Fig 4.2 EisenHower Matrix

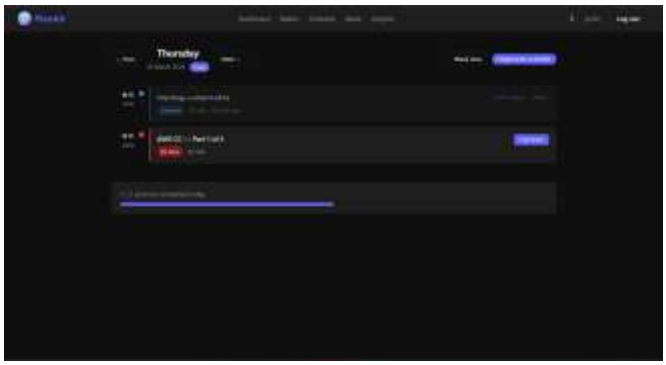


Fig 4.3 Today's Schedule

Fig 4.4 Weekly Schedule


Fig 4.5 Insights

Fig 4.6 Landing Page

## V. CONCLUSION

5.1 Effectiveness of PlanAIr Platform

The PlanAIr platform has successfully achieved its objective of creating a complete, intelligent personal productivity system that surpasses the capabilities of conventional to-do applications. By demonstrating that rule-based AI scheduling can produce genuinely intelligent behavior without the complexity of deep learning, the system provides a realistic solution for modern time management. The integration of the Eisenhower Matrix ensures a scientifically grounded framework for prioritization that users can understand and trust [1]. Furthermore, the deadline-aware scheduling algorithm establishes a critical correctness property by ensuring that time-sensitive work is never displaced by lower-priority tasks.

5.2 User Experience and Accessibility

PlanAIr offers an intuitive, responsive web interface designed for ease of use across both desktop and mobile platforms. The inclusion of a dedicated Insights Dashboard with five distinct chart types provides users with granular visibility into their productivity trends and completion patterns. A key differentiator in accessibility is the mood-aware scheduling system, which acknowledges the human reality that productivity is not constant. This adaptive approach empowers users by adjusting the workload to fit their mental energy, rather than demanding the user adapt to a rigid tool.

5.3 Technological Efficiency

Leveraging the power of the Flask framework and SQLAlchemy ORM, PlanAIr provides a robust backend capable of managing complex relational data and scheduling logic efficiently [4],[5]. The use of Flask Blueprints ensures a modular architecture, allowing for clean separation of concerns and easy future updates. The implementation of session logging and progress tracking represents a significant technical improvement over simple checkbox-based completion, providing precise data for the productivity scoring engine.

5.4 Future Improvements

Future development iterations will focus on expanding the application's scalability and intelligence. Key enhancements include:

- Database Migration: Transitioning to a persistent cloud database such as PostgreSQL to eliminate the storage limitations of SQLite.
- Cross-Platform Accessibility: Developing a mobile application version using React Native to extend the tool's reach.
- Advanced AI Integration: Incorporating machine learning-based scheduling to adapt session lengths and timing based on historical completion patterns.

## REFERENCES

[1] S. R. Covey, The 7 Habits of Highly Effective People. New York: Free Press, 1989.

[2] C. Newport, Deep Work: Rules for Focused Success in a Distracted World. New York: Grand Central Publishing, 2016.

[3] A. M. Isen, K. A. Daubman, and G. P. Nowicki, "Positive affect facilitates creative problem solving," Journal of Personality and Social Psychology, vol. 52, no. 6, pp. 1122–1131, 1987.

[4] A. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.

[5] M. Bayer, "SQLAlchemy: The Architecture of Open Source Applications," 2012. [Online]. Available: https://www.aosabook.org/en/sqlalchemy.html.

[6] Chartjs.org, "Chart.js Documentation," 2024. [Online]. Available: https://www.chartjs.org/docs/latest/.

[7] Flask Documentation, "Flask — A Python Microframework," 2024. [Online]. Available: https://flask.palletsprojects.com/.

[8] SQLAlchemy Documentation, "SQLAlchemy 2.0 Documentation," 2024. [Online]. Available: https://docs.sqlalchemy.org/.

**Authors Profile**

***Mr. Girish Dayaghan Sakpal*** is a third-year undergraduate student in Artificial Intelligence and Data Science at Rizvi College of Engineering, University of Mumbai. His interests include programming, artificial intelligence, and data science, with a focus on real-world applications. He has developed projects such as a facial recognition-based attendance system and a student-centric notes web platform. He is currently working on predictive modeling and intelligent systems to enhance his expertise and contribute to practical, technology-driven solutions.

***Mr. Aryan Prakash Pilwalkar*** is a third-year Bachelor of Technology student at Rizvi College of Engineering, Mumbai University, specializing in Artificial Intelligence and Data Science. Deeply focused on the practical applications of data science, he contributed significantly to EarnIt, a machine learning-based salary prediction system that utilizes advanced regression models and dynamic skill-based adjustments to provide accurate career analytics. His technical portfolio also includes an automated facial recognition attendance system centered on ethical computer vision and a digital repository for academic resources designed to enhance peer learning. Through these projects, he combines a strong foundation in predictive modeling with a dedication to building innovative, data-driven solutions.

***Mr. Kaif Reshamwala*** is a third-year Bachelor of Engineering student in Artificial Intelligence and Data Science at Rizvi College of Engineering, Mumbai University. His areas of interest include artificial intelligence, machine learning, web development, and user interface design. He is currently working on EarnIT, a salary prediction system using machine learning. With strong problem-solving and programming skills, he aims to develop innovative and user-friendly technological solutions.