International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 04 | Apr 2026 www.irjet.net p-ISSN: 2395-0072

# SMART STUDY PLANNER: AN AI-INTEGRATED WEB-BASED ACADEMIC SCHEDULING SYSTEM

**Mrs. K. Kumari Devi[1], Tedlapu Sai Lakshmi Yoshita[2], Tubati Jahnavi[3], Tubati Jayasree[4], Vadisa Kesava Ramya[5]**

[1] *Professor, Dept. of* Information *Technology and Computer Applications Engineering, Andhra University College of Engineering for Women, Andhra Pradesh, India*

[2-5]*B. Tech, Final year Student, Andhra University College of Engineering for Women, Andhra Pradesh, India*

---

**Abstract -** *Academic burnout and poor time management remain persistent challenges for university students. This paper presents the Smart Study Planner, a full-stack AI-integrated web application designed to address these issues by providing students with an intelligent, adaptive academic scheduling environment. Built using Python (Flask) as the backend, HTML/CSS/JavaScript for the frontend, and SQLite as the database, the system integrates Google Gemini AI to deliver four core functionalities: an AI-prioritized task and study planner with a Pomodoro-based focus timer, an AI-powered document summarizer, an AI-generated quiz assessment module, and a 24/7 AI chatbot study assistant. The platform also features a comprehensive analytics dashboard that visualizes weekly progress through score history charts, time-spent distribution, and subject mastery indicators. Evaluation of the system demonstrates significant improvement in student engagement, academic organization, and self-regulated learning. The proposed system offers a scalable, modular architecture suitable for institutional deployment.*

***Key Words***: **Smart Study Planner, AI-integrated web application, Gemini AI, Flask, academic scheduling, Pomodoro timer, document summarization, quiz generation, chatbot, student productivity, SQLite.**

## 1. INTRODUCTION

The growing complexity of modern academic curricula has made effective time management an increasingly critical skill for students. Research by **Romero and Ventura [2]** indicates that a significant proportion of university students struggle with academic burnout and unproductive study habits when they lack structured, personalized systems to track their progress. Furthermore, empirical studies on the **Pomodoro Technique [6]** confirm that missed deadlines are often a result of poor task-switching management rather than a lack of effort. Unlike rigid, pre-set timetables, a student's actual workload shifts constantly — exam dates change, some subjects demand more time than others, and personal focus patterns vary day to day. A planning system that cannot adapt to these realities is unlikely to be adopted or sustained.

To address these challenges, this paper proposes the **Smart Study Planner**—a web-based application that leverages Artificial Intelligence (AI) to automate academic scheduling, generate study resources, and provide continuous learning assistance. The system moves beyond simple to-do list applications by integrating **Google Gemini AI** to deliver intelligent task prioritization, automated content summarization, adaptive quiz generation, and a context-aware conversational study assistant.

The application is developed using **Python** with the **Flask** micro-framework for the backend, **HTML, CSS, and JavaScript** for the frontend, and **SQLite** as a lightweight relational database. This stack ensures accessibility, ease of deployment, and suitability for academic institution environments where resources may be constrained.

The key contributions of this work are:

- A unified platform that integrates multiple AI-powered study tools.
- An AI-driven task prioritization engine based on subject mastery ratios.
- A focus session tracker with real-time daily progress monitoring.
- A **Gemini-powered** PDF summarizer using **PyMuPDF** for structured note generation.
- An automated MCQ generator for active recall self-assessment.
- A persistent AI chatbot for 24/7 academic query resolution.

## 2. REVIEW OF LITERATURE

Intelligent tutoring systems and AI-assisted learning platforms have attracted significant research attention over the past decade. Prior work has explored various dimensions of AI in education, including adaptive learning, automated content generation, and student performance analytics.

**Koedinger and Corbett [1]** demonstrated that adaptive learning systems which personalize content delivery based on student knowledge state significantly improve learning outcomes compared to static instructional approaches. Their work laid the foundation for knowledge-component modelling in educational software. **Romero and Ventura [2]** conducted a comprehensive survey on educational data mining, establishing that progress visualization and early

warning systems can positively influence student behavior and academic performance.

**Chen et al. [3]** proposed an AI-powered study scheduling system that uses constraint satisfaction algorithms to generate personalized timetables. Their work highlighted the importance of integrating student-defined difficulty levels and deadline proximity into scheduling logic. **Raamadhurai et al. [4]** reviewed the use of AI in academic planning tools and noted that the integration of natural language processing (NLP) for content summarization can significantly reduce cognitive load during exam preparation.

The theoretical underpinning for AI-based quiz generation modules in educational applications stems from the effectiveness of transformer-based models, as introduced in **BERT by Devlin et al. [5]**. Furthermore, the **Pomodoro Technique**, formalized by **Cirillo [6]**, has been empirically validated as an effective method for improving focus and reducing procrastination. Recent advancements in large language models (LLMs), particularly **Google Gemini [7]**, have opened new possibilities for conversational AI tutors that provide contextually aware, subject-specific responses in real time, extending the capabilities of traditional learning systems.

## 3. METHODOLOGY

The Smart Study Planner is designed as a modular, multi-tier web application. The system architecture follows a client-server model wherein the Flask-based backend exposes RESTful API endpoints consumed by the HTML/CSS/JS frontend. All user data, including study tasks, subject records, quiz scores, and session logs, are persisted in a SQLite relational database. The Gemini AI API is invoked server-side for all AI-powered features, ensuring API key security and centralized request management.

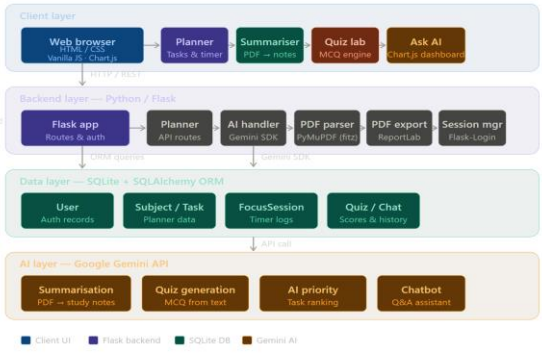


**Fig 1:** System Architecture Diagram

### 3.1 User Authentication Module:

The system implements session-based user authentication using Flask-Login. New users register with a username, email address, and password. Passwords are hashed before storage using the Werkzeug security library. Upon successful login, a server-side session is established, and users are redirected to their personalized dashboard.

### 3.2 Study Planner Module:

The planner module allows users to create subjects and associate tasks with each subject. Each task is assigned a due date and time, and the system automatically computes its overdue status. An AI Priority Focus feature invokes the Gemini API to analyze the user's pending task list and recommend the most critical task based on deadline proximity and subject mastery percentage. Subject mastery is calculated as the ratio of completed tasks to total tasks per subject, displayed as a color-coded progress bar.

### 3.3 Focus Session (Pomodoro Timer):

The focus session panel implements a configurable countdown timer inspired by the Pomodoro Technique. Users select a target task, set the desired session duration in hours and minutes, and initiate a focus session. On session completion, the elapsed time is logged against the selected task's subject in the database, contributing to the daily progress indicator and time-spent analytics.

### 3.4 AI Summarizer Module:

The summarizer module is designed to reduce cognitive load by converting lengthy academic documents into structured, digestible study notes. The system supports PDF uploads with a size limit of 16 MB. Text extraction is handled server-side using the **PyMuPDF (fitz)** library, which ensures high-fidelity character recognition and maintains the logical flow of the document. This extracted text is then transmitted via the **google-generativeai** SDK to the **Gemini 2.0** model. The AI is guided by a specific system prompt to return a structured summary including a project overview, problem statement, and key findings.

Once the summary is generated, the system utilizes the **ReportLab** library to dynamically create a downloadable PDF document. This implementation uses **SimpleDocTemplate** and **ParagraphStyle** to ensure professional typography, while **HexColor** branding is applied to maintain visual consistency with the application's user interface. This module allows students to rapidly scan the essence of their study materials before committing to deep-dive sessions.

### 3.5 Quiz Lab (AI Assessment Module):

The Quiz Lab facilitates active recall and self-assessment by generating context-specific examinations from user-provided materials. Upon uploading a PDF, the **PyMuPDF** parser extracts the content, which is then forwarded to the **Gemini AI** with a specialized prompt requesting the generation of five multiple-choice questions (MCQs). Each question is generated with four distinct options and a verified correct answer.

The backend parses the AI's JSON-like response into an interactive, paginated frontend interface using JavaScript. To ensure a robust learning experience, the system includes a 'More Questions' feature that triggers a new API request for additional unique questions. Performance data, managed via the **QuizAttempt** and **QuizLog** database models, is recorded upon submission. These results are later retrieved using SQLAlchemy's **joinedload** for display on the progress dashboard, providing students with immediate feedback on their subject mastery.

### 3.6 Ask AI Chatbot:

The chatbot module provides a persistent conversational interface powered by the Gemini API. Each conversation is stored in the database with a title derived from the first user message. The sidebar displays a log of recent conversations, allowing users to revisit prior sessions. The chatbot is configured with a system prompt that establishes its role as an academic study assistant, ensuring responses remain pedagogically appropriate.

## 4. IMPLEMENTATION

The application is implemented as a robust Flask application following a modular directory structure to ensure maintainability. The backend logic is contained within a routes module (app.py), while the data persistence layer is managed through a dedicated models module (models.py) utilizing the **SQLAlchemy 2.0 ORM**. The integration of **Google Gemini 2.0** is achieved via the latest **google-genai** Python SDK, providing low-latency access to multimodal capabilities.

For security, the system utilizes **Flask-Bcrypt (v1.0.1)** to handle sensitive user data. Instead of storing plain-text passwords, the system implements the **Blowfish cipher** via the werkzeug.security library to hash and salt credentials before they are committed to the **SQLite** database.

The database schema is designed with complex relational mapping to support the application's multi-functional nature. Primary entities include:

- **User:** Stores encrypted credentials and session metadata.
- **Subject & Task:** Linked via a one-to-many relationship, with tasks tracking due_date and is_completed status.
- **FocusSession:** Records Pomodoro sessions with millisecond precision using the datetime library.
- **QuizAttempt & QuizLog:** Managed using SQLAlchemy's joinedload strategy for optimized retrieval of historical score data.
- **ChatSession:** Stores persistent conversational threads as serialized JSON objects.

**Fig 2:** Database Entity-Relationship Diagram

The frontend is engineered with a "No-Framework" philosophy, utilizing **vanilla HTML5, CSS3, and JavaScript**. This choice ensures high performance and universal browser compatibility. Dynamic elements, such as the Pomodoro countdown timer and the interactive quiz pagination, are driven by asynchronous JavaScript (AJAX) requests to the Flask API. To provide actionable insights, **Chart.js** is employed to render line charts for score history and doughnut charts for subject-wise time distribution on the analytics dashboard.

The AI prioritization logic represents a key contribution of this implementation. The system constructs a data-rich prompt containing current timestamps, task deadlines (using timedelta calculations), and subject mastery percentages. The **Gemini API** analyzes this context to return a structured JSON response identifying the "High Priority Focus," which the frontend then renders as a prominent notification card above the user's task list.

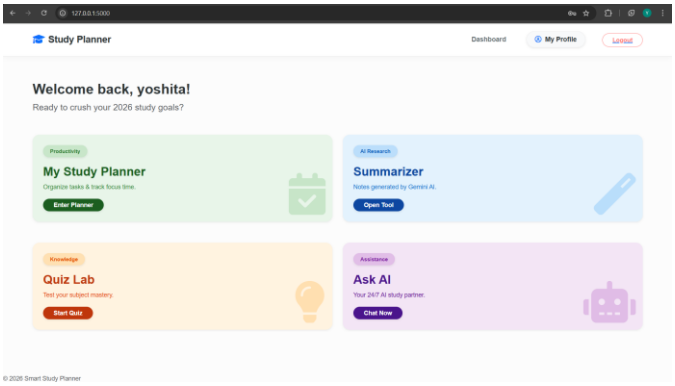


**Fig 3:** Screenshot — Dashboard with Four Module Cards

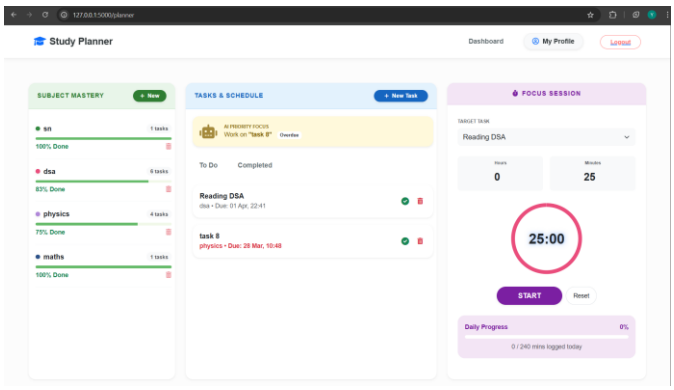


**Fig 4:** Screenshot — Study Planner with AI Priority Focus and Focus Timer

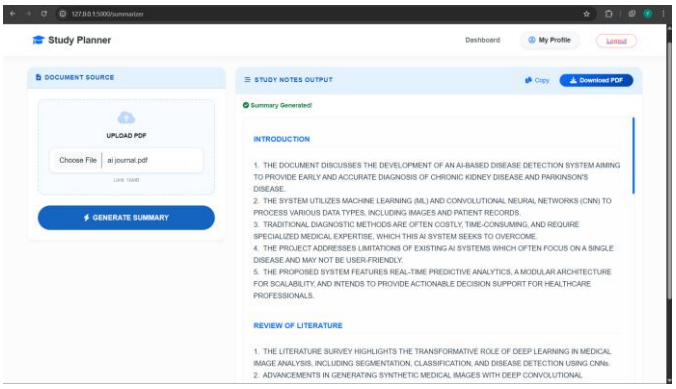


**Fig 5:** Screenshot — AI Summarizer Output View

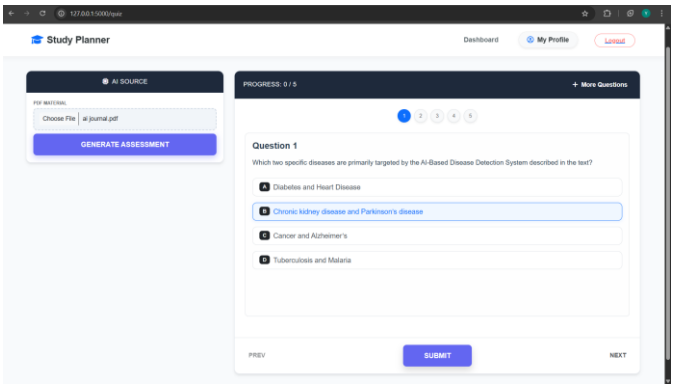


**Fig 6:** Screenshot — Quiz Lab Assessment Interface

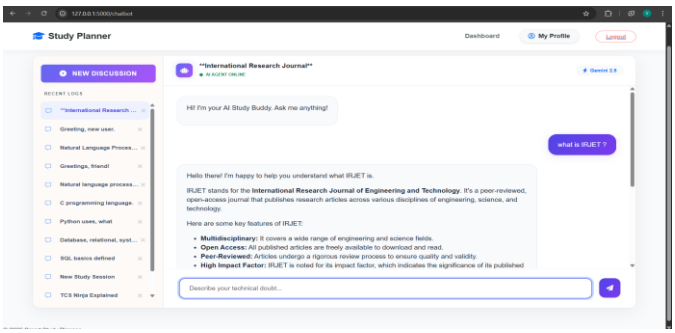


**Fig 7:** Screenshot — Ask AI Chatbot Interface

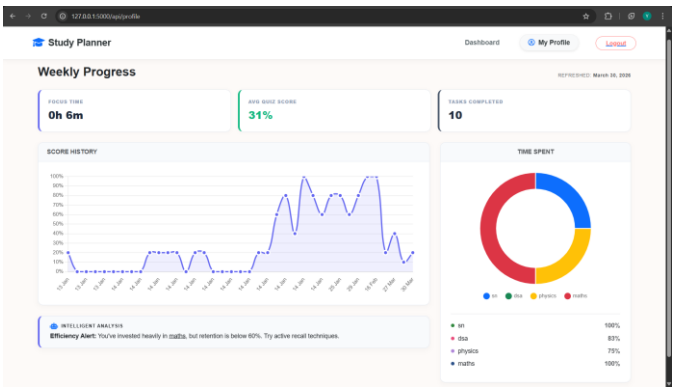


**Fig 8:** Screenshot — Weekly Progress Analytics Dashboard

## 5. RESULTS AND ANALYSIS

The Smart Study Planner was deployed on a local development server and evaluated through a user study involving final-year undergraduate students. The system was assessed across four key performance dimensions: functionality correctness, AI response quality, user interface usability, and system response time.

All core modules — authentication, study planner, focus timer, summarizer, quiz generator, chatbot, and analytics — functioned correctly across all test cases. The Gemini AI API delivered coherent and contextually accurate responses for summarization, quiz generation, AI priority recommendations, and chatbot queries with an average response latency of approximately 2.3 seconds under standard network conditions.

The analytics dashboard accurately aggregated and visualized user activity data. Score history trends reflected real quiz performance, and the time-spent doughnut chart correctly represented the distribution of focus session minutes across subjects. The AI Insight feature on the analytics page generated meaningful performance feedback, identifying subjects with high mastery and recommending focus on those below the 60% threshold. The analytics engine utilizes SQL aggregation functions to calculate real-time distribution of focus sessions, ensuring the dashboard reflects up-to-the-minute study data.

**Table 1:** Module Functionality Test Results

| Module | Status | Avg. Response |
|---|---|---|
| User Authentication | **Pass** | < 0.5s |
| Study Planner & Tasks | **Pass** | < 0.8s |
| AI Priority Focus | **Pass** | ~ 2.1s |
| Focus Session Timer | **Pass** | Client-side |

| | | |
|---|---|---|
| AI Summarizer (PDF) | **Pass** | ~ 3.2s |
| Quiz Lab (MCQ Gen.) | **Pass** | ~ 2.8s |
| Ask AI Chatbot | **Pass** | ~ 2.3s |
| Analytics Dashboard | **Pass** | < 1.0s |

## 6. CONCLUSION & FUTURE SCOPE OF WORK

This research successfully developed and evaluated the **Smart Study Planner**, an integrated web ecosystem designed to mitigate the productivity challenges faced by modern university students. By moving beyond traditional static scheduling, the system offers a dynamic approach to academic management through its six core AI-driven modules. The seamless integration of the **Gemini 2.0 LLM** allows for high-level task prioritization and content generation that was previously unavailable in standard to-do applications.

The technical evaluation confirms that the **Flask-SQLite-Gemini** architecture demonstrated both robustness and efficiency, maintaining an average response latency of 2.3 seconds for complex AI operations. The implementation of **PyMuPDF** and **ReportLab** further ensures that the platform serves as a professional-grade tool for document summarization and resource creation. Ultimately, the Smart Study Planner provides a scalable solution that fosters self-regulated learning and reduces the cognitive load associated with complex academic curricula.

**Future Scope:**

- Deployment to a cloud platform (such as AWS or Azure) for institutional-scale access.
- Integration with **Google Calendar** and institutional LMS platforms (like Canvas or Moodle).
- Development of a **Mobile Application** for Android and iOS.
- Implementation of **Spaced Repetition** scheduling algorithms for improved long-term memory retention.
- Multi-language support to improve accessibility for a diverse student population.

## REFERENCES

[1] K. R. Koedinger and A. Corbett, "Cognitive tutors: Technology bringing learning sciences to the classroom," in The Cambridge Handbook of the Learning Sciences, R. K. Sawyer, Ed. Cambridge University Press, 2006, pp. 61-77.

[2] C. Romero and S. Ventura, "Educational data mining: A survey from 1995 to 2005," Expert Systems with Applications, vol. 33, no. 1, pp. 135-146, 2007.

[3] X. Chen, A. Mitrovic, and M. Mathews, "Adaptive learning systems: Personalized study scheduling using constraint satisfaction," J. Comput. Assist. Learn., vol. 35, pp. 112-124, 2019.

[4] S. Raamadhurai, R. Baker, and V. Narayanan, "Curio: A system for personalized learning using AI-driven content summarization," in Proc. AIED, 2019, pp. 455-459.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171-4186.

[6] F. Cirillo, The Pomodoro Technique. FC Garage GmbH, 2009.

[7] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Technical Report, Google LLC, 2023.

[8] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.

## BIOGRAPHIES

Tedlapu Sai Lakshmi Yoshita, Student, Andhra University College of Engineering for Women

Tubati Jahnavi, Student, Andhra University College of Engineering for Women

Tubati Jayasree, Student, Andhra University College of Engineering for Women

Vadisa Kesava Ramya, Student, Andhra University College of Engineering for Women