iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

# SCHEDEASY — SMART TIMETABLE GENERATOR

**V. Aarthi**
Department of Computer Applications, Vels institute of science and technology, Chennai, India

**ABSTRACT**
SchedEasy is a web-based Smart Timetable Generator developed as a BCA final year project. The application is designed to simplify and automate the process of creating weekly academic timetables for college students and administrators. Traditional timetable creation is a manual, time-consuming task that is highly prone to scheduling conflicts, such as the same subject being assigned to multiple slots on the same day. SchedEasy addresses this problem by implementing an intelligent scheduling algorithm that automatically generates a conflict-free weekly timetable with a single click. The system is built using Python and the Flask web framework for the backend, SQLite as the lightweight file-based database, and HTML5 with CSS3 for the frontend interface. The Jinja2 templating engine is used to render dynamic HTML pages, and the ReportLab library enables users to download the generated timetable as a professionally formatted PDF document. The entire application runs on a local server, requiring no internet connection after initial setup. The core scheduling algorithm iterates through each weekday and time slot, filters subjects by their availability, and uses a Python set data structure to ensure no subject is repeated within the same day. Classrooms are randomly assigned to each scheduled slot from the pool of available rooms. The application features a clean, modern user interface with a pastel pink and green theme, card-based layout, and responsive design, making it accessible and easy to use. Key features of SchedEasy include subject management with progress tracking, classroom registration, automatic timetable generation, a progress dashboard with visual progress bars, and PDF export functionality. The project demonstrates practical application of full-stack web development concepts, database design, and algorithmic thinking, making it a well-rounded and resume-worthy academic project.

**Keywords:**
Python, Flask, SQLite, HTML5, CSS3, Jinja2, ReportLab, Timetable Generator, Web Application, Scheduling Algorithm, Full-Stack Development, BCA Project

## INTRODUCTION

The process of creating academic timetables in educational institutions has traditionally been a labor-intensive, error-prone manual task. Scheduling conflicts — such as a subject being assigned to multiple slots within the same day, or classrooms being double-booked — are common and difficult to detect without automated support. As institutions grow and subject offerings expand, the complexity of timetable generation increases exponentially. SchedEasy was conceived to address these challenges by providing a simple, efficient, and reliable automated scheduling solution built using modern web development technologies, specifically Python, Flask, SQLite, HTML5, CSS3, and the ReportLab library.

## OBJECTIVES

The main objectives of the SchedEasy project are: (1) to automate the generation of conflict-free weekly academic timetables; (2) to provide an intuitive subject and classroom management interface for administrators; (3) to implement a robust scheduling algorithm using Python that prevents day-level subject repetition; (4) to enable PDF export of generated timetables for offline use; and (5) to demonstrate full-stack web development proficiency using Python, Flask, SQLite, HTML5, and CSS3 within a single, deployable web application.

## METHODOLOGY

SchedEasy was developed using a full-stack web development approach. The backend is powered by Python's Flask framework, which handles routing, form processing, and database interaction via SQLite. The Jinja2 templating engine renders dynamic HTML pages populated with real-time data from the database. The frontend is built with HTML5 and CSS3, featuring a card-based, responsive layout with a pastel pink and green color scheme.

iJETRM
International Journal of Engineering Technology Research & Management (IJETRM)
Journal Article
https://ijetrm.com/issue/

The core scheduling algorithm works as follows: for each weekday (Monday through Friday) and each defined time slot, the algorithm retrieves the list of available subjects from the database. It maintains a Python *set* to track subjects already assigned on a given day, ensuring no subject is repeated within the same day. A subject is randomly selected from the filtered pool of eligible subjects for each slot, and a classroom is randomly assigned from the registered rooms. The generated timetable is stored in the database and rendered dynamically on the dashboard.

## RESULTS AND DISCUSSION

The SchedEasy application successfully demonstrated automated, conflict-free weekly timetable generation. Upon testing with multiple sets of subjects and classrooms, the scheduling algorithm consistently produced timetables free of day-level repetition. The use of Python sets as the core data structure for tracking assigned subjects proved to be both computationally efficient and logically sound.

The subject management module allowed administrators to add, view, and delete subjects, while progress bars on the dashboard provided a visual representation of scheduling completion. The classroom registration feature ensured room assignments were drawn from a validated pool. The PDF export feature, implemented using the ReportLab library, produced cleanly formatted,downloadable timetable documents that could be printed or shared digitally. The application's lightweight architecture — relying on SQLite and Flask's built-in development server — made it easy to set up and run on any standard laptop or desktop without external infrastructure.

## ACKNOWLEDGEMENT

The authors express sincere gratitude to the faculty of the Department of Computer Applications for their guidance and support throughout the development of this project. Special thanks are due to the project supervisor for providing valuable insights and constructive feedback during all phases of the project lifecycle. The authors also thank their families and peers for their encouragement and inspiration throughout this journey.

## CONCLUSION

SchedEasy successfully achieves its primary goal of automating academic timetable generation for college environments. The application eliminates manual scheduling errors, reduces administrative workload, and provides a user-friendly interface for managing subjects, classrooms, and timetables. The project demonstrates a comprehensive application of full-stack web development skills including backend logic with Python and Flask, database design with SQLite, frontend development with HTML5 and CSS3, and document generation with ReportLab. As a BCA final year project, SchedEasy represents a well-rounded, technically sound, and practically relevant solution with strong potential for real-world deployment in educational institutions. Future enhancements may include multi-user authentication, faculty-specific scheduling constraints, and integration with institutional ERP systems.

## REFERENCES

1) Flask Documentation. Pallets Projects. Retrieved from https://flask.palletsprojects.com/
2) SQLite Documentation. SQLite Consortium. Retrieved from https://www.sqlite.org/docs.html
3) ReportLab User Guide. ReportLab Inc. Retrieved from https://www.reportlab.com/docs/reportlab-userguide.pdf
4) Jinja2 Documentation. Pallets Projects. Retrieved from https://jinja.palletsprojects.com/
5) Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python* (2nd ed.). O'Reilly Media.
6) Python Software Foundation. Python 3 Documentation. Retrieved from https://docs.python.org/3/
7) Mozilla Developer Network. HTML5 Reference. Retrieved from https://developer.mozilla.org/en-US/docs/Web/HTML
8) Mozilla Developer Network. CSS3 Reference. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS
9) Scheduling Algorithms in Academic Timetabling — A Review. *International Journal of Computer Applications*, 2020.