

Dynamic Faculty Allocation and Timetable Generator

Abdul Reehan¹, Balaji H V², Bhuvaneshwari S³, Dhanush R B⁴

^{1, 2, 3, 4} Department of Computer Science & Engineering

^{1, 2, 3, 4} Mysuru Royal Institute of Technology, Mandya, Karnataka, India

Under the guidance of Dr. Lanka Lakshmi Durga, HOD, Dept. of CSE, MRIT, Mandya

Abstract- Manual scheduling of faculty and class timetables in engineering colleges following the VTU scheme is time-consuming, error-prone, and difficult to update whenever subjects or faculty availability change. This paper presents a web-based Dynamic Faculty Allocation and Timetable Generator that automates the allocation of faculty to subjects based on expertise and workload, and generates a clash-free weekly timetable. The system is developed using the Flask framework with an SQLite database, and allows administrators to upload subject schemes and faculty details in CSV/XLSX format. An allocation engine matches faculty expertise to subjects, applies a no-repeat rule, and balances teaching workload before a timetable-generation module builds a conflict-free weekly grid. Results show that the proposed system considerably reduces the manual effort and errors involved in academic scheduling while producing consistent, clash-free timetables.

Keywords- Faculty allocation, timetable generation, workload balancing, clash-free scheduling, Flask web application.

I. INTRODUCTION

Academic institutions affiliated to VTU must prepare, every semester, a faculty allocation register and a weekly class timetable that respects subject schemes, faculty expertise, and workload limits. When this process is carried out manually, it is slow, prone to human error, and has to be redone almost from scratch whenever a subject list changes or a faculty member's availability changes.

The Dynamic Faculty Allocation and Timetable Generator addresses this problem by providing a web-based system that lets an administrator upload the subject scheme and faculty list once, and then automatically allocates subjects to faculty and produces a clash-free weekly timetable. The system reduces the time and effort required for manual timetable preparation and provides a simple, user-friendly interface for department administrators.

II. PROBLEM STATEMENT

- Managing faculty workload and availability is difficult to do manually.

- Changes in subjects or faculty require repeated timetable modifications.
- A dynamic, automated system is needed to reduce manual work and errors.
- The system must generate an efficient, clash-free timetable based on the VTU scheme.

III. OBJECTIVES

- Automate faculty allocation based on subject expertise and workload.
- Generate timetables automatically according to the VTU scheme.
- Manage faculty, subjects, classrooms, and time slots efficiently.
- Provide a simple and user-friendly system for administrators.
- Improve the accuracy and efficiency of academic scheduling.

IV. LITERATURE REVIEW

A number of existing approaches to automated timetabling and faculty allocation were studied before designing the proposed system. Table 1 summarizes the surveyed works along with their key limitations, which motivated the design choices made in this project.

Table 1: Literature Review Summary

Title	Author	Year	Summary	Limitation
Smart Timetable Generator	Sharma et al.	2025	Uses genetic algorithms to create timetables.	Not suitable for very large colleges.
Automated Timetable Generation	U. Naik et al.	2024	Solves teacher and classroom conflicts.	Cannot easily handle changing faculty availability.
Dynamic Faculty Allocation	Preethi V M	2025	Assigns teachers based on expertise and availability.	Takes more time for large data.
AI-Powered Timetable Generator	K. Reshma et al.	2026	Uses AI to create better timetables from old data.	Depends on past data.
Hybrid Timetable Generator	Kronika Jounal	2025	Uses two algorithms to reduce conflicts and balance workload.	More complex and slower.

V. PROPOSED METHODOLOGY

The proposed system follows a four-step methodology, moving from raw scheme and faculty data to a published, clash-free timetable.

Faculty Allocation and Timetable Generation Workflow

1. Upload Subject Scheme: The department administrator uploads the VTU subject scheme for the semester in CSV or XLSX format, listing every course along with its credits, contact hours and branch details.

2. Upload Faculty Details: The administrator uploads the faculty list, including each member's designation, maximum teaching hours per week, and the subject expertise codes they are qualified to teach.

3. Run Allocation Engine: The allocation engine matches faculty expertise to the uploaded subjects, applies a no-repeat rule so that no faculty member is assigned the same subject in consecutive terms, and balances teaching workload across the department.

4. Generate Timetable: The timetable-generation module builds a clash-free weekly grid from the resulting allocation register, checking every section already scheduled for the academic year before the schedule is published.

VI. SYSTEM ARCHITECTURE

The faculty allocation process takes two inputs — the faculty list with expertise data, and the subject scheme with its courses and slots. These inputs are processed in three stages: expertise matching, application of the no-repeat rule, and workload balancing, before subjects are finally allocated. The output of this process is an allocation register and a weekly timetable grid, as summarized in Fig. 1.

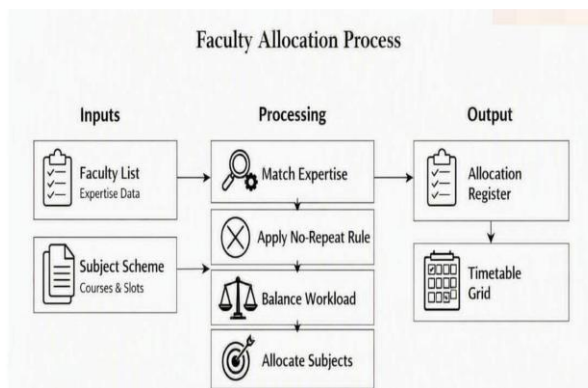


Fig. 1: Faculty Allocation Process

At the system level, the web application (built with Flask) accepts three categories of input — faculty details (expertise, workload, availability), subjects following the VTU scheme (uploaded as CSV/XLSX), and resource details such as classrooms, labs, and time slots. Internally, the application runs a faculty allocation engine and a timetable-generation engine, validates the result for clashes, and stores all data in an SQLite database. The system produces two outputs: a clash-free weekly class timetable, and a subject-wise faculty allocation list along with each faculty member's workload. The overall architecture is illustrated in Fig. 2.

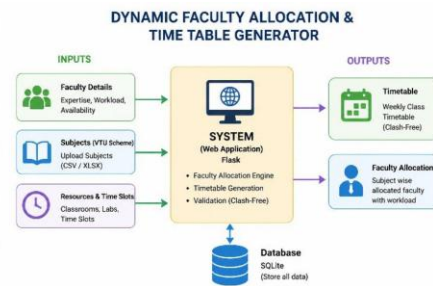


Fig. 2: System Architecture of the Dynamic Faculty Allocation and Timetable Generator

VII. TOOLS AND TECHNOLOGIES

- Languages: Python, HTML, CSS, Jinja2
- Framework: Flask
- Database: SQLite
- Libraries: Werkzeug (authentication), OpenPyXL (Excel/CSV handling)
- Environment: Windows CMD setup

VIII. IMPLEMENTATION (MODULES)

- Authentication Module: Handles administrator registration and login.
- Faculty Module: Add or upload faculty records and map their subject expertise.
- Subjects Module: Add or upload scheme subjects.
- Allocation Engine: Performs expertise matching, workload balancing, and enforces the no-repeat rule.
- Timetable Generator: Builds the weekly grid and performs clash-free scheduling.

IX. RESULTS

The developed system was tested with sample faculty and subject data for the Department of Computer Science and Engineering. The dashboard allows the administrator to add faculty, load the VTU scheme, run the allocation engine, and generate the timetable in four guided steps. Faculty records were added through a profile form capturing designation, department, contact details, and the maximum teaching hours per week. Subjects were added following the VTU 2022 OBE convention, including subject code, credits (L:T:P:C), and contact hours per week.

Once faculty and subjects were registered, the timetable-generation module was run for a selected branch, semester, academic year, and section. The system automatically checked for clashes across every section already scheduled for that academic year and produced a conflict-free

weekly grid, confirming that the allocation engine and timetable generator work correctly together. Representative screens of the administrator dashboard are shown in Fig. 3.

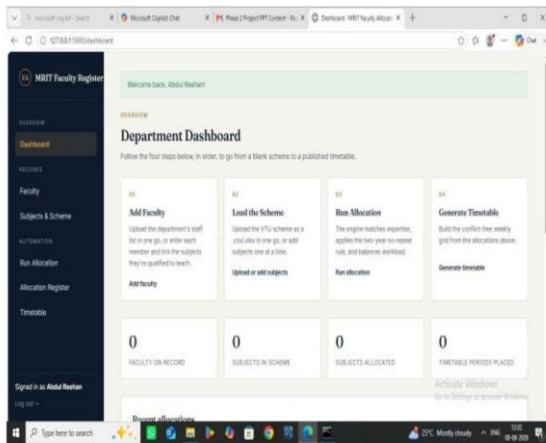


Fig. 3(a): Department Dashboard

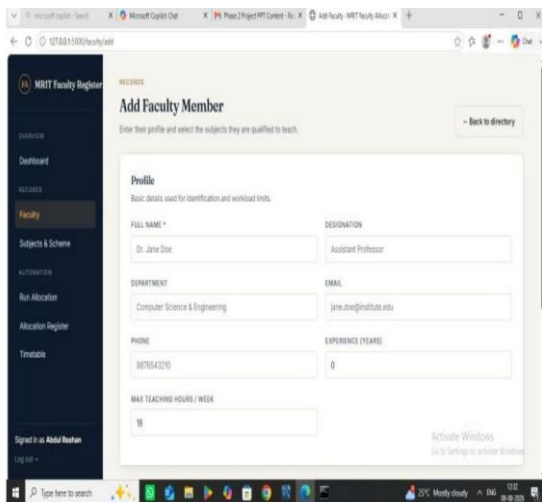


Fig. 3(b): Add Faculty Member

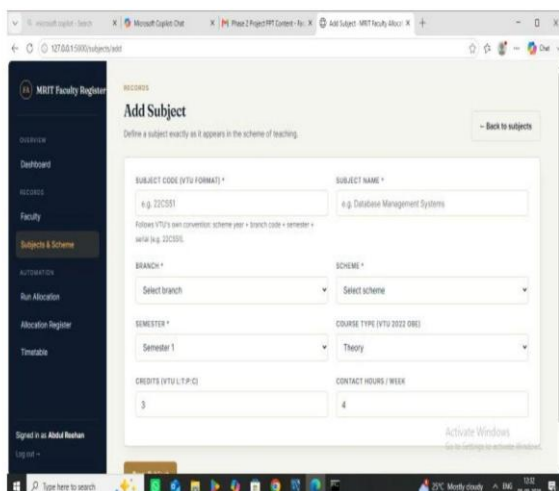


Fig. 3(c): Add Subject

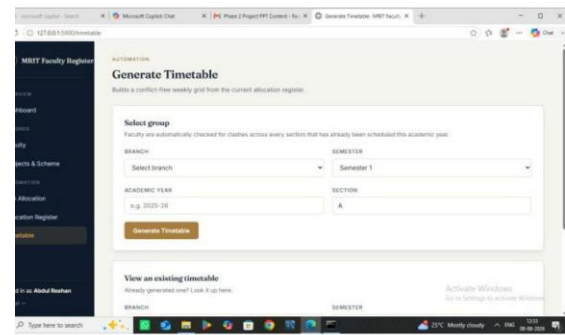


Fig. 3(d): Generate Timetable

CONCLUSION

The Dynamic Faculty Allocation and Timetable Generator successfully automates a process that is traditionally manual, slow, and error-prone. By combining expertise-based faculty allocation, workload balancing, and a no-repeat rule with an automated timetable-generation engine, the system produces accurate, clash-free weekly timetables while significantly reducing administrative effort. Future work may extend the system to handle multiple departments simultaneously, incorporate faculty preference scores, and support automatic conflict resolution across shared elective subjects.

ACKNOWLEDGMENT

The authors would like to thank Dr. Lanka Lakshmi Durga, HOD, Department of Computer Science & Engineering, MRIT, Mandya, for her valuable guidance and support throughout this project.

REFERENCES

- [1] Sharma et al., "Smart Timetable Generator Using Genetic Algorithms," International Journal of Computer Applications, 2025.
- [2] U. Naik et al., "Automated Timetable Generation for Teacher and Classroom Conflict Resolution," IJSART, 2024.
- [3] Preethi V M, "Dynamic Faculty Allocation Based on Expertise and Availability," IJSART, 2025.
- [4] K. Reshma et al., "AI-Powered Timetable Generator Using Historical Data," IJSART, 2026.
- [5] Kronika Jounal, "Hybrid Timetable Generator: A Two-Algorithm Approach for Conflict Reduction and Workload Balancing," 2025.
- [6] X. Y. Jing, F. Wu, Z. Li, R. Hu and D. Zhang, "Multi-Label Dictionary Learning for Image Annotation," IEEE Transactions on Image Processing, vol. 25, no. 6, pp. 2712-2725, June 2016.