CHAPTER 1

INTRODUCTION

1.1 Introduction to the Project

A timetable is a key academic component in any educational institution. Coordinators must consider various factors during scheduling, which becomes particularly stressful in colleges due to multiple departments, diverse courses, and faculty members with varying designations. They must carefully allocate courses, classrooms, and faculty based on subject requirements, and patiently construct an efficient timetable to ensure an organized schedule and smooth functioning of the institution. Despite the increasing automation of institutional tasks in colleges, lecture timetables are still often prepared manually due to the inherent complexities of the process.

Timetable scheduling problem is inherently complex due to the numerous interrelated factors and constraints, including teaching plans, courses, faculty and teaching classes, classrooms, and time slots. In universities, the complexity increases significantly during semester scheduling, as it must accommodate multiple student batches, various student groups, elective courses, and common mandatory courses taken by all the students. It is essential to ensure that there are no clashes among student, faculty members during the scheduling process.

Timetable scheduling is widely recognized as a combinatorial optimization problem and a constraint satisfaction problem where the goal is to find a solution that satisfies a predefined set of constraints. These constraints are generally categorized as hard and soft constraints. Hard constraints are mandatory and must be satisfied for a timetable to be feasible, such as preventing classes from clashing in the same room or a teacher having more than one class simultaneously. Soft constraints, on the other hand, represent preferences or desired conditions that ideally should be met but are not strictly required for feasibility. Meanwhile violating soft constraints is acceptable, but an optimal timetable minimizes such violations.

This project aims to address the difficulties of generating timetables by providing an automated scheduling system. Automating the process helps save time, avoid the complexity of manual management, and reduce documentation work. The goal is to generate timetables that are more accurate, precise, and free of human errors. An automated system ensures up-to-date and accurate information.

To achieve automatic timetable generation, the project proposes the use of algorithms such as genetic, heuristic, and resource scheduling. Genetic algorithms are frequently applied to timetabling problems, which are known to be NP-hard optimization problems, because Genetic algorithm is effective for finding optimal or near-optimal feasible solutions among a complex set of variables and constraints. They are based on natural selection and evolution principles and are known for their robustness in solving complex combinatorial problems. Heuristic approaches are also commonly used in timetabling, either independently or as components within algorithms like GAs, often focusing on scheduling the most constrained elements first. Resource scheduling is another algorithmic approach listed for addressing these problems. These algorithms incorporate various strategies aimed at improving the efficiency, scalability and reliability of the timetable generation process.

The system will generate the timetables based on various inputs, including the number of course and faculty, faculty workload, semester details, and subject priorities. Additional necessary inputs include faculty details, course details (including name and code), workload based on faculty designation, faculty and course allotment based on time slots, and details of theory and lab courses handled by each faculty. Classroom including availability and capacity, are also crucial inputs. By relying on these inputs and utilizing optimization algorithms, the system will generate possible timetables for the working days.

In conclusion, the proposed solution aims to streamline the scheduling process, enhance accuracy, and offer a more efficient, scalable, and reliable approach to timetable generation, ultimately contributing to smoother academic operations and a better teaching-learning experience.

1.2 Introduction to Technology Used

The Web-Based Automatic Timetable Scheduler integrates a set of modern technologies that together enable efficient, accurate, and scalable timetable generation. The core development is carried out in Python, chosen for its simplicity, flexibility, and the wide availability of libraries for optimization, database interaction, and system integration.

The user interface is built using Streamlit, a Python-based framework that allows the rapid creation of interactive web applications without requiring front-end technologies like HTML, CSS, or JavaScript. Through Streamlit, users can easily log in, manage academic data, configure faculty preferences, and view generated timetables in a visually clear and user-friendly format. For data management, the system uses a PostgreSQL relational database, ensuring persistent storage of users, departments, faculty, courses, and preferences. The database layer, provides structured functions for CRUD operations such as adding faculty, updating courses, or retrieving stored preferences. This design ensures secure and reliable handling of institutional data while keeping the system lightweight and portable. At the heart of the scheduler lies the Genetic Algorithm. The GA is responsible for optimizing timetables by simulating natural selection through operations like selection, crossover, and mutation.

The overall system, integrates the Streamlit UI, PostgreSQL database, and Genetic Algorithm logic into a cohesive workflow. This modular architecture improves maintainability, readability, and scalability of the project. By leveraging these technologies together, the Web-Based Automatic Timetable Scheduler provides a practical and automated solution that reduces manual effort, minimizes human error, and ensures optimized utilization of academic resources.