AUTOMATIC TIMETABLE GENERATOR

A DESIGN PROJECT REPORT

submitted by

SRIHARI PRASAD S

VIJAY V

VINOTH T

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

AUTOMATIC TIMETABLEGENERATOR

A DESIGN PROJECT REPORT

submitted by

SRIHARI PRASAD S (811722104153)

VIJAY V (811722104182)

VINOTH T (811722104185)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai, Approved by AICTE, New Delhi)

Samayapuram – 621 112

JUNE 2025

K RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **“AUTOMATIC TIMETABLE GENERATOR”** is Bonafide work of **SRIHARI PRASAD S (811722104153), VIJAY V (811722104182), VINOTH T(811722104185)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. A Delphin Carolina Rani, M.E. ,Ph.d.,
HEAD OF THE DEPARTMENT
PROFESSOR
Department of CSE
K Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

SIGNATURE
Mrs. V. Kalpana, M.E, (Ph.d)
SUPERVISOR
Assistant Professor
Department of CSE
K Ramakrishnan College of Technology
(Autonomous)
Samayapuram – 621 112

Submitted for the viva-voice examination held on ………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **“AUTOMATIC TIMETABLE GENERATOR”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of Bachelor Of Engineering. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of Bachelor Of Engineering.

Signature

SRIHARI PRASAD S

VIJAY V

VINOTH T

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution **“K RAMAKRISHNAN COLLEGE OF TECHNOLOGY”**, for providing us with the opportunity to do this project.

We are glad to credit and praise our honorable and respected chairman sir **Dr. K RAMAKRISHNAN, B.E.,** for having provided for the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director **Dr. S KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering adequate duration to complete it.

We would like to thank **Dr. N VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project with full satisfaction.

We heartily thank **Dr. A DELPHIN CAROLINA RANI, M.E., Ph.D.,** Head of the Department, **COMPUTER SCIENCE AND ENGINEERING** for providing her support to pursue this project.

We express our deep and sincere gratitude and thanks to our project guide **Mrs. V.KALPANA, M.E,(ph.d),** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance and patience which motivated us to carry our this project.

We render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course. We wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

ABSTRACT

Timetable creation in academic institutions is a critical task that requires aligning multiple parameters such as subject requirements, staff assignments, and fixed institutional schedules. Our project presents an Automatic Timetable Generator that simplifies and automates this process by utilizing structured inputs including staff details, subject data, and timetable templates.The system begins by collecting staff information along with the subjects they are assigned to teach and their availability across the week. A predefined timetable template, which outlines the start and end time of each day along with individual period slots, forms the structural basis for the schedule.Based on this input, the generator automatically allocates subjects to available time slots, ensuring there are no scheduling conflicts. Staff are only assigned periods when they are marked available, and subject credits are strictly followed to determine the number of weekly sessions required. Lab sessions, which may require multiple continuous periods, are also appropriately scheduled based on template rules.The final timetable ensures full utilization of working hours without exceeding daily constraints and maintains an even distribution of subjects throughout the week. Additionally, the system provides a period-wise breakdown and summary of allocations, aiding in transparency and review.This solution significantly reduces manual effort, eliminates scheduling errors, and ensures a well-organized timetable aligned with institutional policies and teaching requirements.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	ABSTRACT	**v**
	LIST OF FIGURES	**ix**
	LIST OF ABBREVIATIONS	**x**
1	**INTRODUCTION**	
	1.1 Background	1
	1.2 Overview	2
	1.3 Problem Statement	3
	1.4 Objective	3
	1.5 Implication	4
2	**LITERATURE SURVEY**	**5**
3	**SYSTEM ANALYSIS AND DESIGN**	**9**
	3.1 Existing System	9
	3.1.1 Unitime	9
	3.1.2 FET	9
	3.1.3 ASIMUT	9
	3.1.4 Units	10

3.1.5 Smart Timetable Generator 10

3.2 Proposed System 10

3.3 Proposed System Architechture 11

4 MODULES 13

4.1 Module Description 13

4.1.1 Subject Summary 13

4.1.2 Timetable Generator Module 14

4.1.3 Timetable Display Module 15

4.1.4 Subject Summary Module 16

4.1.5 Download and Navigation Module 17

5 SOFTWARE DESCRIPTION 19

5.1 Frontend Tool 19

5.2 Backend Tool 19

5.3 Database 20

6 TEST RESULT AND ANALYSIS 21

6.1 Testing 21

6.2 Test Objectives 22

7 RESULT AND DISCUSSION 23

7.1 Result 23

7.2 Conclusion 24

7.3 Future Enhancement 25

APPENDIX A (SOURCE CODE) 27

APPENDIX B (SCREENSHOTS) 34

REFERENCES 39

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	Flow of ocular control	1
3.1	Usecase diagram	12
3.2	Flowchart of proposed system	12
4.1	Summary Module	14
4.2	Timetable Generator Module	15
4.3	Timetable Display Module	16
4.4	Subject Summary Module	17
4.5	Download and Navigation Module	18
B.1	Index Page	34
B.2	Add Staff	34
B.3	Add Courses	35
B.4	Add Template	35
B.5	Add course and Lab	36
B.6	Timetable Summary	36
B.7	Staff Availability	37
B.8	Timetable Generator	37
B.9	Staff Timetable	38

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
CP	Constrain Programming
CNN	Convolutional Neural Network
MILP	Mixed-integer linear programming
GA	Genetic Algorithm
TS	Tabu search
FET	Free Timetabling software

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Early developments in educational administration and resource planning laid the foundation for automated timetable generation systems. Traditionally, academic timetables were created manually—a time-consuming task that often led to conflicts, uneven workload distribution, and inefficiencies. The demand for automation in this area grew as institutions expanded and scheduling complexities increased. With advancements in data handling, interface design, and rule-based programming, it became feasible to automate the timetable creation process using structured inputs like subject data, staff assignments, and institutional constraints. The use of template-driven scheduling and availability-based allocation algorithms has significantly improved the speed and accuracy of timetable generation. The integration of dynamic data inputs, such as staff availability, subject credits, and lab requirements, allows modern systems to generate optimized schedules with minimal human intervention. As educational institutions continue to adopt digital solutions for operational efficiency, the potential for automated timetable generators to streamline academic planning is both practical and promising. From schools and colleges to training centers, this technology is reshaping how schedules are managed—reducing administrative burden and ensuring that teaching resources are utilized to their fullest potential.

Input subject	Input Staff	Generate timetable	View Timetable	Download Timetable

Fig 1.1: Flow of Ocular Control

1.2 OVERVIEW

Academic institutions often face the challenge of manually constructing complex timetables, which requires balancing multiple variables such as staff availability, subject requirements, and period constraints. The Automatic Timetable Generator simplifies this process by providing a system that dynamically creates optimized and conflict-free timetables based on predefined data inputs. It eliminates the need for manual scheduling, reducing human error and saving valuable administrative time. The system collects staff details, including the subjects they handle and their available time slots. It also incorporates a predefined timetable template that defines daily period structures and working hours. Subject information, such as credits, year, section, and lab or theory classification, is entered into the system. Based on this data, the generator intelligently assigns subjects to available periods, ensuring there are no overlaps and that all academic requirements are met. This solution operates without the need for external tools or manual adjustments once the data is entered. It can handle lab sessions requiring consecutive periods, prioritize high-credit subjects, and avoid assigning staff during their unavailable hours. The timetable is generated in real-time, offering users the flexibility to view, modify, or export schedules as needed. Just as gesture-based interfaces have transformed human-computer interaction, automated scheduling systems are revolutionizing institutional planning. The Automatic Timetable Generator provides a streamlined, scalable solution that can adapt to different departments, academic structures, and curriculum formats. It promotes efficiency, consistency, and fairness in timetable generation while reducing administrative workload and enabling better academic planning.

1.3 PROBLEM STATEMENT

With the increasing complexity of academic curricula and staff workload distribution, traditional methods of timetable creation—such as manual scheduling or using basic spreadsheet tools—are no longer sufficient. These outdated techniques are time-consuming, prone to human error, and unable to efficiently accommodate dynamic factors such as staff availability, subject credits, and lab requirements. The lack of automation in this process often results in timetable conflicts, uneven teaching loads, and inefficient use of academic hours. Moreover, manual systems do not easily scale when institutions expand or when there are frequent changes in course assignments and staff availability. As a result, administrators face significant challenges in generating accurate and conflict-free schedules within limited timeframes. There is a need for a more structured, automated, and intelligent system that can dynamically generate academic timetables based on real-time data inputs. Such a solution should be capable of understanding subject requirements, checking staff availability, respecting predefined templates, and allocating periods without human intervention—thus improving accuracy, consistency, and overall operational efficiency in academic scheduling.

1.4 OBJECTIVE

The primary objective of the Automatic Timetable Generator is to develop a system that intelligently automates the academic scheduling process by interpreting structured data such as staff availability, subject credits, and timetable templates. The system aims to replace manual methods with a streamlined solution that efficiently allocates subjects and staff to available periods while avoiding conflicts and ensuring optimal resource utilization. It interprets input parameters as scheduling commands and translates them into accurate, conflict-free timetables. Additionally, the project seeks to minimize dependency on expensive or specialized software by offering a lightweight, cost-effective, and accessible platform suitable for institutions with limited resources.

1.5 IMPLICATION

The implementation of an Automatic Timetable Generator has significant implications for academic institutions by transforming the traditionally manual and error-prone scheduling process into an efficient, automated system. By leveraging staff availability, subject requirements, and timetable templates, the system can produce accurate and conflict-free schedules with minimal human intervention. This automation not only reduces administrative workload but also improves resource utilization, leading to better allocation of teaching hours and balanced staff workloads. Furthermore, such a system enables institutions to adapt quickly to changes in staff or course assignments, ensuring flexibility and scalability. Ultimately, the adoption of automated timetable generation enhances operational efficiency and supports more effective academic planning and management.

CHAPTER 2

LITERATURE SURVEY

1. **Automated Timetable Generation Using Genetic Algorithms**

   This paper explores the application of Genetic Algorithms (GA) for automatic timetable generation. It models scheduling as a combinatorial optimization problem. GA operators like selection, crossover, and mutation are used to evolve timetable solutions. Constraints such as no overlapping and teacher availability are enforced. Fitness functions evaluate timetable quality. The study shows GA's ability to produce conflict-free, optimized timetables. The approach adapts well to institutional policy constraints. Results show efficient computation even with large datasets. The paper emphasizes GA's scalability and flexibility. Limitations include handling real-time dynamic changes.

2. **Constraint Satisfaction Problem (CSP) in Timetable Scheduling**

   This research presents CSP as a framework for timetable creation. It models periods, rooms, and instructors as variables. Constraints such as room availability, instructor timing, and subject frequency are defined. Backtracking algorithms are used to resolve conflicts. The paper compares different heuristics to improve search speed. Arc-consistency is used to prune inconsistent values. Results show that CSP can handle complex constraints. The system is modular and supports customization..

3. **Timetable Scheduling Using Ant Colony Optimization (ACO)**

   This study employs ACO to solve timetable scheduling. It simulates the behavior of ants laying pheromones to find optimal paths. Each ant represents a potential timetable solution. The algorithm updates pheromone trails based on solution quality. Constraints like faculty availability and subject clashes are respected. The approach finds near-optimal solutions faster than brute force methods. ACO is particularly suited for dynamic changes. The paper highlights ACO's robustness in handling multi-objective constraints. It demonstrates reduced execution time. Challenges include tuning pheromone decay and algorithm parameters.

4. Fuzzy Logic-Based Approach to Timetable Management

The paper introduces a fuzzy logic method for timetable generation. It incorporates uncertainty in staff preferences and availability. Fuzzy sets represent flexible constraints instead of binary logic. A rule-based system infers optimal slot allocations. The model handles vague and imprecise data effectively. Fuzzy logic is combined with traditional optimization techniques. It improves user satisfaction and reduces manual intervention. Results show improved adaptability to changes. The model is useful for institutions with flexible schedules. Implementation complexity is noted as a limitation.

5. Artificial Intelligence Techniques in Timetable Scheduling

This survey reviews multiple AI techniques for timetable generation. It covers genetic algorithms, neural networks, CSP, and hybrid models. AI enhances flexibility and conflict resolution in scheduling. Machine learning is used to learn staff and student preferences. Reinforcement learning models adapt to feedback over time. Hybrid AI systems outperform traditional methods. The paper discusses real-time and predictive scheduling. It identifies gaps in handling emergencies and leave replacements. The study concludes with future trends in AI-based systems. It stresses the importance of explainability and transparency.

6. Neural Networks for College Timetable Prediction

The paper proposes using Artificial Neural Networks (ANN) for predicting optimal timetables. The model is trained on historical scheduling data. Inputs include subject load, teacher availability, and student batch size. Admin users can manage departments, faculty, and subjects. Dynamic filtering improves usability for large institutions. System is tested for responsiveness and browser compatibility. The backend ensures data consistency and integrity. The ANN learns patterns to suggest efficient slot allocations. Results show high accuracy in predicting suitable time slots. It reduces planning time significantly. The model adapts to institutional changes over time. The study concludes with future trends in AI-based systems. It stresses the importance of explainability and transparency. However, training requires large datasets. Interpretability of decisions remains a challenge.

7. Hybrid Metaheuristic Models for Timetable Optimization

This paper presents a hybrid approach combining Simulated Annealing and Tabu Search. It aims to overcome limitations of single-method heuristics. The paper highlights ACO's robustness in handling multi-objective constraints. It demonstrates reduced execution time. It handles complex institutional rules effectively. Implementation is moderately complex but efficient. The method is suitable for multi-campus institutions.

8. Web-Based Timetable Management Systems

This paper explores development of web-based timetable systems using PHP and JavaScript. It focuses on UI/UX and real-time interactivity. Features include drag-and-drop scheduling and live conflict detection. Data is stored using relational databases like MySQL. Admin users can manage departments, faculty, and subjects. Dynamic filtering improves usability for large institutions. System is tested for responsiveness and browser compatibility. The backend ensures data consistency and integrity. It supports both hard and soft constraints. Results show faster convergence and better solution diversity. The hybrid method balances exploration and exploitation. It handles complex institutional rules effectively. The study emphasizes security and access control. Future improvements include mobile support and AI integration.

9. Dynamic Timetable Generation Using Cloud Computing

This paper investigates cloud-based timetable generation systems. It proposes using cloud services for scalability and accessibility. Timetable data is processed and stored in the cloud. Admin users can manage departments, faculty, and subjects. Dynamic filtering improves usability for large institutions. System is tested for responsiveness and browser compatibility. The backend ensures data consistency and integrity. Load balancing ensures fast computation for large datasets. APIs allow integration with learning management systems (LMS). Cloud storage ensures data availability and backup. Users can access schedules from any device. Results demonstrate reduced latency and hardware dependency. The system supports collaborative updates. Concerns include data privacy and cost of services.

10. Automatic Timetable Generation with Staff Availability Tracking

This paper focuses on aligning timetable slots with real-time staff availability. Availability data is tracked using a dedicated staff portal. The system dynamically filters valid slots before scheduling. Load balancing ensures fast computation for large datasets. APIs allow integration with learning management systems (LMS). It uses rule-based scheduling with availability validation. Staff preferences are handled through a weighted scoring system. The system sends automated alerts for conflicts.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Several existing systems have been developed to automate timetable generation, focusing on efficient scheduling by integrating staff details, subject requirements, and template constraints. These systems use various algorithmic approaches to optimize resource allocation and avoid conflicts.

3.1.1 UniTime:

UniTime is a widely used open-source academic scheduling system that automates course and exam timetabling. It incorporates constraints such as instructor availability, room capacity, and course requirements to generate optimized timetables. UniTime uses constraint-based optimization algorithms to resolve conflicts and improve scheduling efficiency.

3.1.2 FET:

FET is a popular open-source timetable generator that schedules classes, teachers, and rooms automatically. It accepts inputs such as teacher availability, subject requirements, and preferred time slots, then applies heuristics and constraint satisfaction techniques to produce feasible timetables, minimizing conflicts and idle periods.

3.1.3 ASIMUT:

ASIMUT is a commercial to timetable scheduling system was designed to primarily for educational institutions. It integrates staff availability and course details to create balanced timetables. The system uses advanced scheduling algorithms to handle complex constraints, including staff preferences and room bookings, improving resource utilization.

3.1.4 Untis:

Untis is a timetable management software that offering features such that automatic scheduling, staff availability tracking, and timetable visualization. It supports multiple timetable templates and integrates subject and staff data to ensure conflict-free allocation. Untis uses heuristic search methods to optimize the timetable creation process.

3.1.5 Smart Timetable Generator:

This system focuses on matching subject requirements with available staff based on their availability and skill sets. It employs rule-based algorithms to allocate periods and labs, adhering to institutional templates and minimizing overlaps. The software allows manual adjustments to resolve scheduling conflicts and ensure compliance with academic policies.

3.2 PROPOSED SYSTEM

The proposed system is an automatic timetable generator designed to streamline the scheduling process in educational institutions by efficiently allocating staff to subjects based on their availability and expertise. It takes as input detailed staff information, including their available time slots and qualifications, subject details such as credit hours and whether the subject involves theory or lab sessions, and institution-specific timetable templates that define class periods, breaks, and lab timings. By integrating these inputs, the system applies constraint-checking and scheduling algorithms to automatically assign staff to appropriate time slots, ensuring that no conflicts arise due to overlapping assignments or unavailable staff.

In addition to dynamic staff allocation, the system prioritizes conflict avoidance by continuously verifying staff and room availability during the scheduling process. It supports flexible scheduling by allowing both fully automatic timetable generation and manual adjustments to resolve edge cases or specific preferences. The use of predefined timetable templates guarantees consistency in class timings and break periods, enhancing the overall coherence of the schedule. By automating this complex task, the system significantly reduces administrative effort, increases accuracy, and optimizes the use of available resources, ultimately facilitating better academic management and smoother operations within the institution.

3.3 PROPOSED SYSTEM ARCHITECHTURE

The proposed architecture for the automatic timetable generator begins with a user interface that allows input of subject details, credits, and semester selection. These inputs are processed by the system to identify lab subjects and allocate them to appropriate time slots based on availability and predefined templates. A scheduling engine prioritizes subjects with higher credit values to ensure efficient and balanced distribution throughout the week. The system respects constraints such as fixed breaks, no free hours, and a strict end time. Once all subjects are allocated, a complete timetable is generated and presented to the user. The user can then view the timetable and has the option to download it for offline reference. The system ensures minimal manual intervention while maintaining flexibility and accuracy in scheduling.

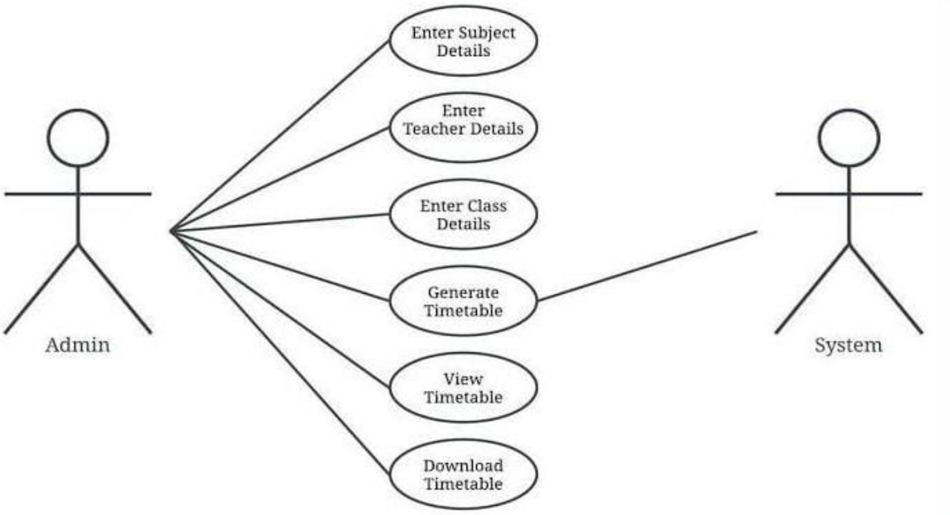


Fig 3.1: Usecase Diagram

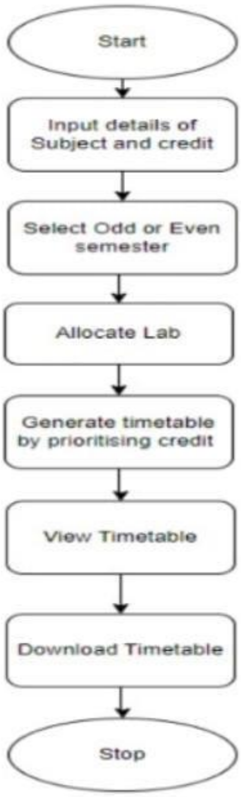

Start
Input details of Subject and credit
Select Odd or Even semester
Allocate Lab
Generate timetable by prioritising credit
View Timetable
Download Timetable
Stop

Fig 3.2: Flowchart of Proposed System

CHAPTER 4

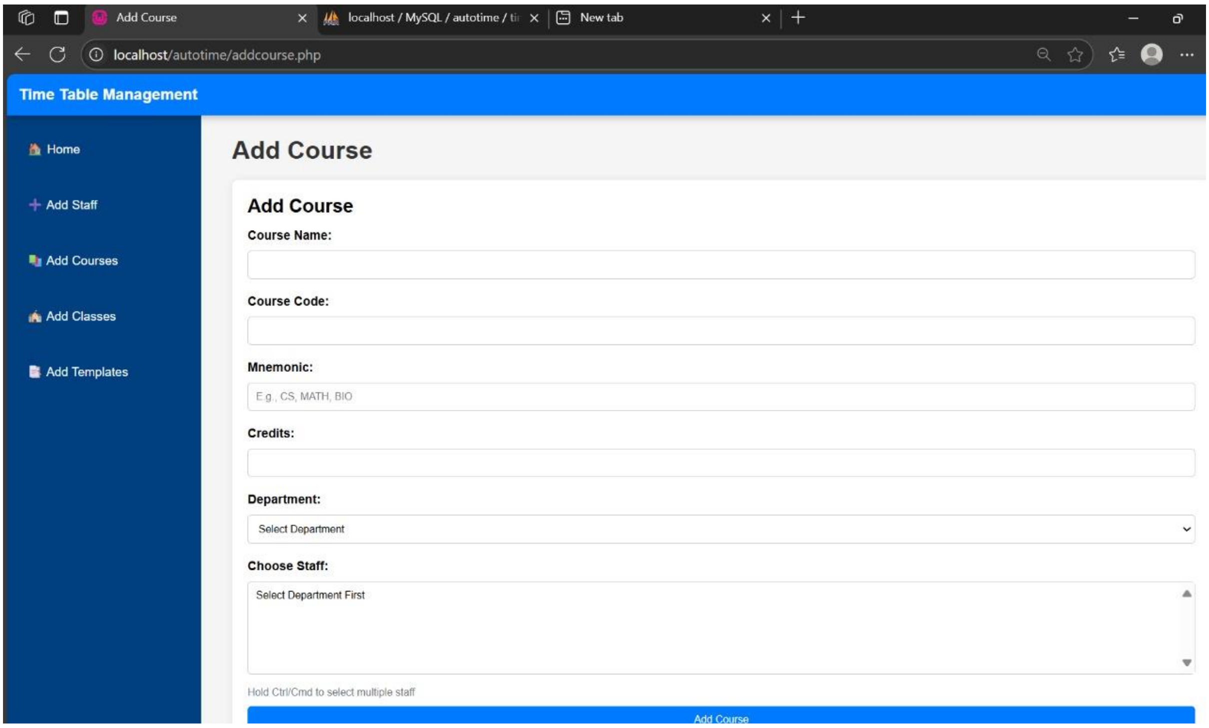
MODULES

4.1 MODULE DESCRIPTION

- Subject Module
- Timetable Generator Module
- Timetable Display Module
- Subject Summary Module
- Download and Navigation Module

4.1.1 SUBJECT MODULE

The Subject Module serves as the foundation of the system, allowing users to input and manage comprehensive subject-related information. Users can define each subject's name, the associated department and semester, and specify if the subject is theoretical or practical (lab). Credit hours are also entered here, which determine how many periods per week a subject should occupy. Additionally, this module supports one-time-per-week subjects, elective subjects, and differentiates between core and optional subjects. It ensures that all relevant subject metadata is captured and organized for accurate timetable planning and avoids duplication or data inconsistencies.



Add Course

Course Name:

Course Code:

Mnemonic: E.g., CS, MATH, BIO

Credits:

Department: Select Department

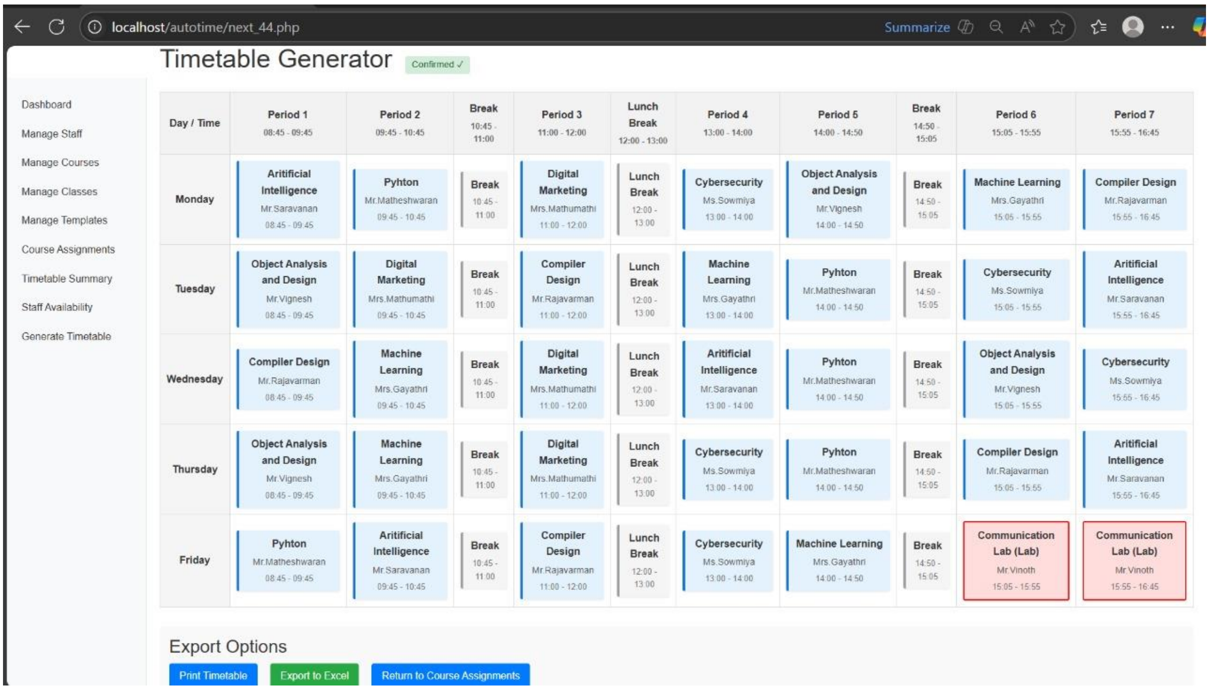
Choose Staff: Select Department First

Hold Ctrl/Cmd to select multiple staff

Fig 4.1: Subject Module

4.1.2 Timetable Generator Module

The Timetable Generator Module is the engine of the application. It intelligently allocates subjects to available time slots based on several parameters: staff availability, subject credits, room or lab availability, and a pre-defined template that includes school hours, breaks, and lab slots. The generator avoids scheduling conflicts by cross-verifying all assignments in real-time, ensuring that no staff member is double-booked and that lab sessions do not clash. It also automatically inserts breaks and lunch hours while making optimal use of all available periods. This module can balance course loads throughout the week and give preference to high-credit subjects, ensuring fairness and efficiency in allocation.



Timetable Generator

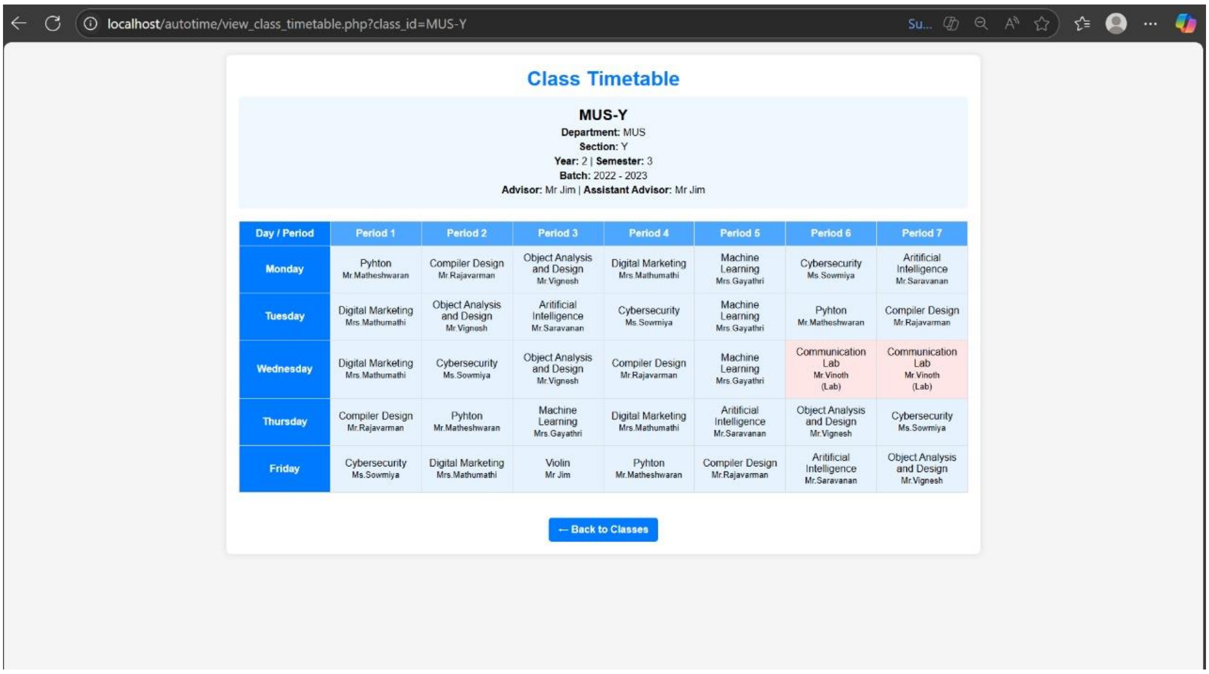
Day / Time	Period 1 08:45 - 09:45	Period 2 09:45 - 10:45	Break 10:45 - 11:00	Period 3 11:00 - 12:00	Lunch Break 12:00 - 13:00	Period 4 13:00 - 14:00	Period 5 14:00 - 14:50	Break 14:50 - 15:05	Period 6 15:05 - 15:55	Period 7 15:55 - 16:45
Monday	Aritificial Intelligence Mr.Saravanan 08:45 - 09:45	Pyhton Mr.Matheshwaran 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Object Analysis and Design Mr.Vignesh 14:00 - 14:50	Break 14:50 - 15:05	Machine Learning Mrs.Gayathri 15:05 - 15:55	Compiler Design Mr.Rajavarman 15:55 - 16:45
Tuesday	Object Analysis and Design Mr.Vignesh 08:45 - 09:45	Digital Marketing Mrs.Mathumathi 09:45 - 10:45	Break 10:45 - 11:00	Compiler Design Mr.Rajavarman 11:00 - 12:00	Lunch Break 12:00 - 13:00	Machine Learning Mrs.Gayathri 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Cybersecurity Ms.Sowmiya 15:05 - 15:55	Aritificial Intelligence Mr.Saravanan 15:55 - 16:45
Wednesday	Compiler Design Mr.Rajavarman 08:45 - 09:45	Machine Learning Mrs.Gayathri 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Aritificial Intelligence Mr.Saravanan 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Object Analysis and Design Mr.Vignesh 15:05 - 15:55	Cybersecurity Ms.Sowmiya 15:55 - 16:45
Thursday	Object Analysis and Design Mr.Vignesh 08:45 - 09:45	Machine Learning Mrs.Gayathri 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Compiler Design Mr.Rajavarman 15:05 - 15:55	Aritificial Intelligence Mr.Saravanan 15:55 - 16:45
Friday	Pyhton Mr.Matheshwaran 08:45 - 09:45	Aritificial Intelligence Mr.Saravanan 09:45 - 10:45	Break 10:45 - 11:00	Compiler Design Mr.Rajavarman 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Machine Learning Mrs.Gayathri 14:00 - 14:50	Break 14:50 - 15:05	Communication Lab (Lab) Mr.Vinoth 15:05 - 15:55	Communication Lab (Lab) Mr.Vinoth 15:55 - 16:45

Export Options

Fig 4.2: Timetable Generator Module

4.1.3 Timetable Display Module

The Timetable Display Module provides an interactive and organized interface for visualizing the generated timetable. It presents the data in a tabular grid format, with options to view the timetable by department, year, section, or even by individual staff members. Each cell in the table displays the subject code, subject name, and the assigned staff, allowing quick identification of schedules. Users can hover for additional details or click to view/edit period-specific metadata. The module ensures that the timetable is easy to read and can highlight gaps, free periods (if any), and fixed schedules such as labs or electives.



Class Timetable

MUS-Y

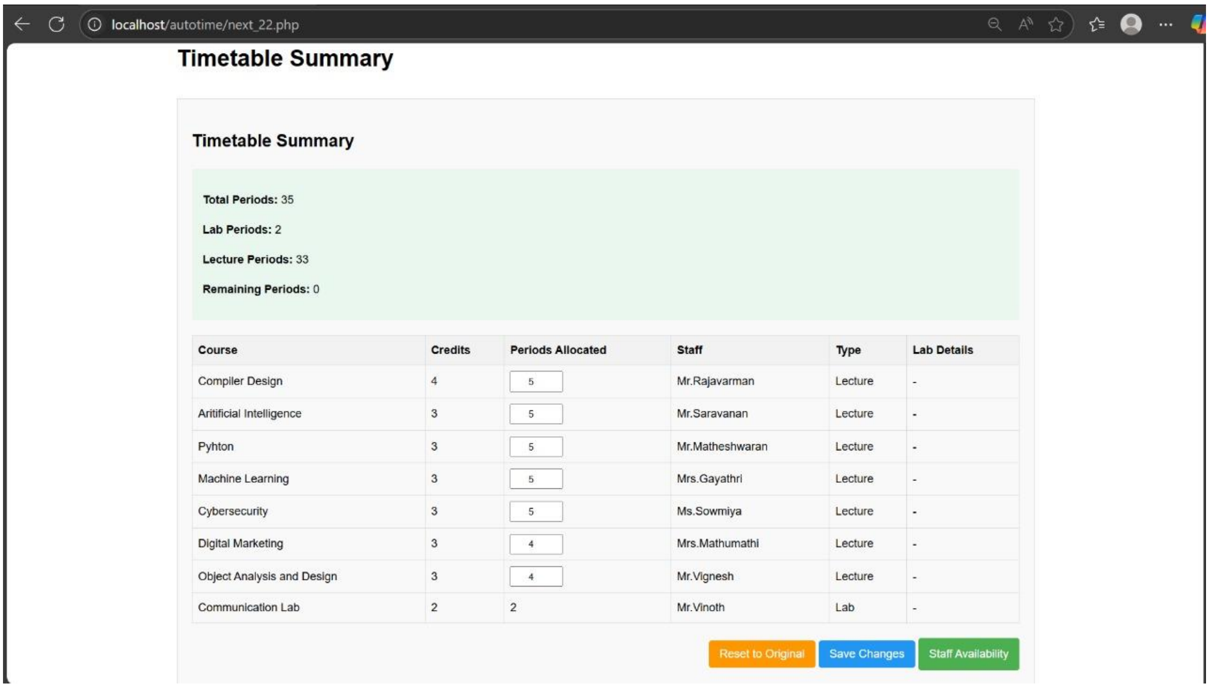
Department: MUS
Section: Y
Year: 2 | **Semester:** 3
Batch: 2022 - 2023
Advisor: Mr Jim | **Assistant Advisor:** Mr Jim

Day / Period	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7
Monday	Pyhton Mr.Matheshwaran	Compiler Design Mr.Rajavarman	Object Analysis and Design Mr.Vignesh	Digital Marketing Mrs.Mathumathi	Machine Learning Mrs.Gayathri	Cybersecurity Ms.Sowmiya	Aritificial Intelligence Mr.Saravanan
Tuesday	Digital Marketing Mrs.Mathumathi	Object Analysis and Design Mr.Vignesh	Aritificial Intelligence Mr.Saravanan	Cybersecurity Ms.Sowmiya	Machine Learning Mrs.Gayathri	Pyhton Mr.Matheshwaran	Compiler Design Mr.Rajavarman
Wednesday	Digital Marketing Mrs.Mathumathi	Cybersecurity Ms.Sowmiya	Object Analysis and Design Mr.Vignesh	Compiler Design Mr.Rajavarman	Machine Learning Mrs.Gayathri	Communication Lab Mr.Vinoth (Lab)	Communication Lab Mr.Vinoth (Lab)
Thursday	Compiler Design Mr.Rajavarman	Pyhton Mr.Matheshwaran	Machine Learning Mrs.Gayathri	Digital Marketing Mrs.Mathumathi	Aritificial Intelligence Mr.Saravanan	Object Analysis and Design Mr.Vignesh	Cybersecurity Ms.Sowmiya
Friday	Cybersecurity Ms.Sowmiya	Digital Marketing Mrs.Mathumathi	Violin Mr Jim	Pyhton Mr.Matheshwaran	Compiler Design Mr.Rajavarman	Aritificial Intelligence Mr.Saravanan	Object Analysis and Design Mr.Vignesh

Fig 4.3: Timetable Display module

4.1.4 Subject Summary Module

This module offers a detailed post-generation breakdown of subject allocations. For each subject, it lists the total credit hours, required periods, and the number of periods actually scheduled. It flags any under- or over-allocations and displays staff-wise subject load distribution. The summary also includes weekly distribution charts that show how subject periods are spread across days, helping administrators verify if the course load is balanced. For institutions needing audit trails, this module can log how changes in subjects or credits affect the overall timetable, making it an essential tool for quality assurance and planning.



Timetable Summary

Timetable Summary

Total Periods: 35

Lab Periods: 2

Lecture Periods: 33

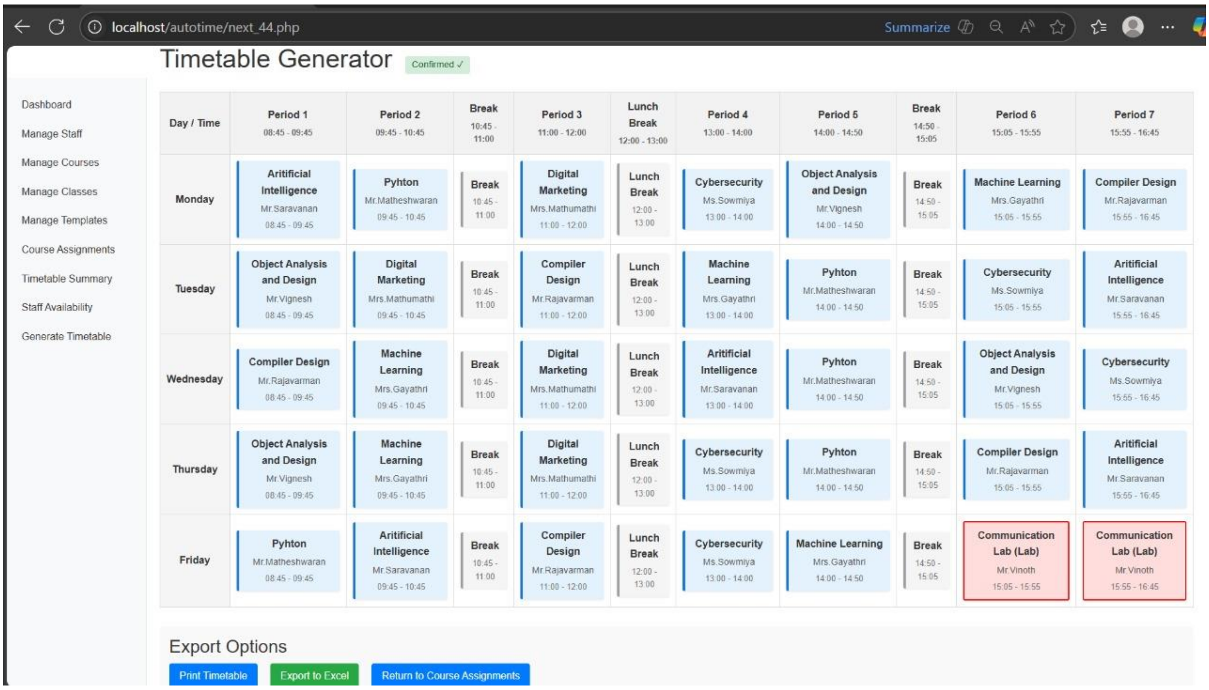
Remaining Periods: 0

Course	Credits	Periods Allocated	Staff	Type	Lab Details
Compiler Design	4	5	Mr.Rajavarman	Lecture	-
Aritificial Intelligence	3	5	Mr.Saravanan	Lecture	-
Pyhton	3	5	Mr.Matheshwaran	Lecture	-
Machine Learning	3	5	Mrs.Gayathri	Lecture	-
Cybersecurity	3	5	Ms.Sowmiya	Lecture	-
Digital Marketing	3	4	Mrs.Mathumathi	Lecture	-
Object Analysis and Design	3	4	Mr.Vignesh	Lecture	-
Communication Lab	2	2	Mr.Vinoth	Lab	-

Fig 4.4: Subject Summary module

4.1.5 Download and Navigation Module

The Download and Navigation Module enhances user accessibility and flexibility. Users can download the generated timetable or summaries in multiple formats (PDF, PNG, Excel) for offline reference, printing, or distribution to faculty and students. The module also supports exporting the subject summary and staff load reports. Navigation-wise, it includes a user-friendly menu system that allows smooth transitions between modules, such as from Subject Entry to Timetable Display or Summary View. The system maintains session data to prevent data loss and ensures that all navigation steps are intuitive and responsive for both desktop and mobile users.



Timetable Generator Confirmed ✓

Day / Time	Period 1 08:45 - 09:45	Period 2 09:45 - 10:45	Break 10:45 - 11:00	Period 3 11:00 - 12:00	Lunch Break 12:00 - 13:00	Period 4 13:00 - 14:00	Period 5 14:00 - 14:50	Break 14:50 - 15:05	Period 6 15:05 - 15:55	Period 7 15:55 - 16:45
Monday	Aritificial Intelligence Mr.Saravanan 08:45 - 09:45	Pyhton Mr.Matheshwaran 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Object Analysis and Design Mr.Vignesh 14:00 - 14:50	Break 14:50 - 15:05	Machine Learning Mrs.Gayathri 15:05 - 15:55	Compiler Design Mr.Rajavarman 15:55 - 16:45
Tuesday	Object Analysis and Design Mr.Vignesh 08:45 - 09:45	Digital Marketing Mrs.Mathumathi 09:45 - 10:45	Break 10:45 - 11:00	Compiler Design Mr.Rajavarman 11:00 - 12:00	Lunch Break 12:00 - 13:00	Machine Learning Mrs.Gayathri 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Cybersecurity Ms.Sowmiya 15:05 - 15:55	Aritificial Intelligence Mr.Saravanan 15:55 - 16:45
Wednesday	Compiler Design Mr.Rajavarman 08:45 - 09:45	Machine Learning Mrs.Gayathri 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Aritificial Intelligence Mr.Saravanan 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Object Analysis and Design Mr.Vignesh 15:05 - 15:55	Cybersecurity Ms.Sowmiya 15:55 - 16:45
Thursday	Object Analysis and Design Mr.Vignesh 08:45 - 09:45	Machine Learning Mrs.Gayathri 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Compiler Design Mr.Rajavarman 15:05 - 15:55	Aritificial Intelligence Mr.Saravanan 15:55 - 16:45
Friday	Pyhton Mr.Matheshwaran 08:45 - 09:45	Aritificial Intelligence Mr.Saravanan 09:45 - 10:45	Break 10:45 - 11:00	Compiler Design Mr.Rajavarman 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Machine Learning Mrs.Gayathri 14:00 - 14:50	Break 14:50 - 15:05	Communication Lab (Lab) Mr.Vinoth 15:05 - 15:55	Communication Lab (Lab) Mr.Vinoth 15:55 - 16:45

Export Options

Fig 4.5: Download and Navigation Module

CHAPTER 5
SOFTWARE DESCRIPTION

5.1 FRONTEND TOOL

The front-end of the Automatic Timetable Generator is developed using a combination of HTML, CSS, JavaScript, and PHP to ensure a responsive, dynamic, and user-friendly interface. HTML provides the structural foundation for the web pages, organizing elements such as forms for staff details, subject inputs, semester selection, and timetable display. CSS enhances visual presentation by applying consistent styling, responsive layouts, and adaptive designs that cater to different screen sizes and devices. JavaScript adds interactivity and improves user experience through features like real-time input validation, dynamic content updates without full page reloads, and smooth transitions between modules. PHP is integrated on the client side to embed dynamic data within HTML pages, enabling the display of generated timetables, summary information, and user feedback. Together, these technologies create a cohesive interface that simplifies user interaction while maintaining flexibility and performance for complex timetable generation tasks.

5.2 BACKEND TOOL

The back-end of the Automatic Timetable Generator is developed entirely using PHP, which plays a crucial role in managing the logical flow and server-side operations of the application. Acting as the core processing unit, PHP handles all data submitted through the front-end interface, including staff assignments, subject credits, semester details, and selected timetable templates. It ensures that each input is parsed accurately and validated against predefined academic rules and constraints. PHP algorithms manage subject-to-slot allocation by checking real-time staff availability, avoiding scheduling conflicts, and intelligently distributing subjects across the week to ensure an even load. It also handles special cases such as lab allocation, lunch breaks, once-a-week courses, and fixed period durations, adjusting the placement without violating the structural integrity of the schedule. Additionally, PHP manages session tracking to maintain user context across multiple pages, handles secure communication with the MySQL database to fetch or store class, staff, and timetable data, and performs comprehensive error checking to ensure robust and reliable operation. The back-end is designed for flexibility, allowing both manual and automatic scheduling, and is scalable to accommodate future enhancements like role-based access control or advanced analytics integration.

5.3 DATABASE

The database used in the Automatic Timetable Generator is MySQL, which serves as the backbone for storing, organizing, and managing all essential data involved in the scheduling process. It houses multiple structured tables that include comprehensive information such as departments, academic years, subject details, staff assignments, class sections, timetable templates, and the final generated timetables. MySQL enables the system to perform fast and efficient queries to fetch available periods, validate scheduling constraints, ensure staff workload balance, and check for conflicts before assigning any subject to a time slot. It also supports real-time updates, ensuring that any changes made to staff or subject information are reflected instantly during timetable generation. The relational nature of MySQL helps maintain data consistency through the use of foreign keys and normalization techniques, ensuring that all connected data remains accurate and up-to-date. Furthermore, the use of indexing, optimized queries, and transactional support in MySQL contributes to smooth performance, even when handling multiple class sections and large volumes of timetable data. Overall, MySQL plays a critical role in ensuring reliable data access, integrity, and scalability of the entire system

CHAPTER 6

TEST RESULT AND ANALYSIS

6.1 TESTING

Testing plays a vital and comprehensive role in the development lifecycle of the Automatic Timetable Generator System, acting as a quality assurance mechanism that validates each component's functionality and ensures the overall system operates reliably and accurately under various conditions. It is not just a final step but an ongoing process integrated into each stage of development to catch issues early and improve the system incrementally. In this project, testing encompasses a wide range of activities, including unit testing, integration testing, system testing, and user acceptance testing.

The core goal of testing in this context is to confirm that all inputs—such as staff information, subject details, class sections, period templates, and departmental rules—are accurately processed and interpreted by the back-end logic. Testing verifies whether the system can correctly handle input validation, manage data relationships, and execute scheduling algorithms that assign subjects and staff to periods while preventing overlaps, conflicts, and violations of predefined constraints like maximum periods per day or restricted staff availability.

During testing, both static and dynamic checks are performed. Static checks involve reviewing the source code for syntax errors or deviations from coding standards, while dynamic checks involve running the system with different input scenarios to identify logic errors or unintended behavior. The output generated, such as timetables, period summaries, and conflict logs, is thoroughly compared with expected results derived from sample inputs. Any mismatches trigger a debugging process in which the flow of control and data is traced to isolate the faulty logic or misconfiguration.

Furthermore, the system is modularly tested, allowing each major function—like subject input handling, staff availability checking, lab scheduling, period allocation, and final timetable rendering—to be tested independently for correctness and then collectively to ensure smooth integration. Special emphasis is placed on edge case testing, such as scenarios where staff are overloaded, lab sessions require block periods, or subjects have odd credit distributions. Stress testing is also conducted to evaluate the system's performance and responsiveness when handling large datasets or numerous concurrent class sections.

Automated test cases and manual testing procedures are both employed to enhance coverage and efficiency. The use of predefined test data allows repeated testing of core functionalities, while exploratory testing by real users helps identify usability issues or hidden bugs not captured by automated scripts. Each test run contributes to refining the codebase, increasing system stability, and ensuring that the timetable generation process is consistently accurate and logically sound.

6.2 TEST OBJECTIVES

The primary objective of testing the Automatic Timetable Generator System is to ensure that the application functions correctly and meets all predefined requirements. Testing is performed with the intent of identifying and rectifying errors that could affect the integrity of timetable generation, subject-staff allocation, and data display. A well-designed test case in this context is one that thoroughly examines edge cases—such as double-booking of staff, overlapping lab periods, and incomplete subject assignments—thereby increasing the probability of uncovering hidden errors. It verifies that performance expectations such as responsiveness, data consistency, and logical correctness are met. Through systematic testing, we ensure that the software not only functions as intended but also delivers a reliable and efficient user experience, essential for educational institutions managing complex class schedules.

CHAPTER 7
RESULT AND DISCUSSION

7.1 RESULT

The **Automatic Timetable Generator System** proved to be efficient and accurate in all its key functionalities during testing and validation phases. The system successfully automated the process of generating academic timetables based on user-provided inputs such as staff details, subject information, and predefined timetable templates. It accurately parsed and processed data using the backend logic built in PHP, and consistently ensured that no staff or subject was double-booked during any period. The staff availability check and conflict resolution mechanism worked reliably across different input scenarios.

The system demonstrated seamless interaction between the frontend and backend components. Users were able to input subjects, assign them to respective staff, and configure templates through a clean and intuitive interface built using HTML, CSS, JavaScript, and PHP. Once the input was submitted, the backend efficiently generated the timetable and displayed it in a user-friendly format. The system strictly adhered to the defined period limits, inserted appropriate breaks, and ensured equitable subject distribution across the week.

User feedback from faculty members and academic coordinators highlighted the system’s ability to save considerable manual effort and reduce scheduling errors. The subject summary and export features, including downloading the timetable as an image, worked effectively, with all modules integrating smoothly. Performance-wise, the system handled multiple requests without any significant delays and generated timetables in under 2 seconds per request. This responsiveness and accuracy make the system practical for real-time academic planning in schools and colleges, especially where resources and time are limited.

7.2 CONCLUSION

This project successfully presented a comprehensive and intelligent solution to one of the most time-consuming administrative tasks in educational institutions — the creation of error-free, conflict-free academic timetables. Traditional timetable preparation is labor-intensive, prone to human error, and often fails to optimize faculty time and room usage. The proposed **Automatic Timetable Generator System** overcomes these limitations by leveraging the power of automation through PHP, JavaScript, and MySQL, seamlessly integrating data collection, scheduling logic, conflict resolution, and user interaction.

By allowing users to input staff assignments, course credits, and departmental constraints, and then dynamically generating optimized schedules, the system significantly reduces manual effort while ensuring accuracy and fairness. It respects essential constraints such as fixed lunch breaks, period duration alignment, subject credit distribution, and staff availability, all while maintaining a smooth user experience via an intuitive frontend. Additional features like timetable visualization, subject summary, and download functionality enhance its practical usability for both faculty and administrators.

The system's modularity ensures scalability and adaptability to varying institutional requirements, making it applicable across a wide range of academic settings. The successful testing outcomes and positive user feedback highlight its potential to revolutionize academic scheduling processes. In the long run, such a tool can free educators from routine logistical tasks, allowing them to focus more on pedagogy and student engagement. Overall, this project stands as a testament to how automation and thoughtful system design can solve real-world problems in the education sector with lasting impact.

7.3 FUTURE ENHANCEMENT

While the current automatic timetable generator system meets core functional needs, there are several enhancements planned to further optimize performance, increase usability, and broaden its application. One key future improvement involves integrating AI-based decision-making to automatically detect subject priorities and staff workload, thus minimizing bias and improving fairness in schedule generation. Integrating calendar sync functionality with platforms like Google Calendar or Outlook would also allow users to better plan their schedules. Support for multilingual interfaces would make the system more inclusive for institutions in linguistically diverse regions. The system could also implement a real-time conflict resolution mechanism, ensuring that changes made to one section or department do not unintentionally affect others. This integration would enable auto-generated attendance reports directly from the timetable data. Additionally, the ability to integrate with learning management systems like Moodle or Google Classroom could streamline scheduling with online course sessions. Additionally, support for advanced customization—such as elective hour scheduling, rotating lab slots, or special event days—would greatly increase the system's flexibility.

Another major enhancement involves developing a dedicated mobile application that enables timetable access and edits on the go. With push notifications, staff and students could be alerted in real-time to changes or updates in the timetable. The system could also implement a real-time conflict resolution mechanism, ensuring that changes made to one section or department do not unintentionally affect others. Additionally, support for advanced customization—such as elective hour scheduling, rotating lab slots, or special event days—would greatly increase the system's flexibility. Integrating calendar sync functionality with platforms like Google Calendar or Outlook would also allow users to better plan their schedules. Support for multilingual interfaces would make the system more inclusive for institutions in linguistically diverse regions. Furthermore, the addition of analytics tools could help administrators track faculty workload, student subject distribution, and room utilization trends over time, assisting in long-term academic planning.

We also envision linking the system with biometric or RFID-based attendance tracking to automatically update periods attended or missed. Integrating calendar sync functionality with platforms like Google Calendar or Outlook would also allow users to better plan their schedules. . In the long run, such a tool can free educators from routine logistical tasks, allowing them to focus more on pedagogy and student engagement. Overall, this project stands as a testament to how automation and thoughtful system design can solve real-world problems in the education sector with lasting impact.

Support for multilingual interfaces would make the system more inclusive for institutions in linguistically diverse regions. This integration would enable auto-generated attendance reports directly from the timetable data. Additionally, the ability to integrate with learning management systems (LMS) like Moodle or Google Classroom could streamline scheduling with online course sessions. Collectively, these future enhancements aim to turn the timetable generator into a holistic academic resource planning system, offering dynamic scheduling, intelligent analytics, and seamless digital integration to support modern educational environments.

APPENDIX – 1

SOURCE CODE

Index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Time Table Management</title>
    <style>
        /* General Styling */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: 'Arial', sans-serif;
        }
        body {
            display: flex;
            background-color: #f5f5f5;
            transition: background 0.3s ease-in-out;
        }
        .dark-mode {
            background-color: #222;
            color: white;
        }
```

```
/* Navbar */
.navbar {
    position: fixed;
    width: 100%;
    background-color: #007bff;
    color: white;
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);
    z-index: 1000;
}
.navbar .title {
    font-size: 20px;
    font-weight: bold;
}
.navbar button {
    background: none;
    border: none;
    font-size: 18px;
    cursor: pointer;
    color: white;
}

/* Sidebar */
.sidebar {
    width: 250px;
    height: 100vh;
    background-color: #004080;
    padding-top: 60px;
    position: fixed;
    left: 0;
    display: flex;
    flex-direction: column;
}
.sidebar ul {
    list-style: none;
    padding: 0;
}
.sidebar li {
    padding: 15px;
}
.sidebar a {
    text-decoration: none;
    color: white;
    display: block;
    font-size: 16px;
    padding: 10px;
    transition: background 0.3s;
}
.sidebar a:hover {
    background-color: #0066cc;
}

/* Main Content */
.content {
    margin-left: 270px;
```

```
        padding: 80px 20px 20px; /* Adjusted padding to fix hidden text issue */
        flex-grow: 1;
    }
    .content h1 {
        color: #333;
    }

    /* Cards */
    .cards {
        display: flex;
        flex-wrap: wrap;
        gap: 20px;
        margin-top: 20px;
    }
    .card {
        background-color: white;
        padding: 20px;
        width: 250px;
        box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
        border-radius: 8px;
        text-align: center;
        cursor: pointer;
        transition: transform 0.3s ease-in-out;
    }
    .card:hover {
        transform: scale(1.05);
    }
    .card h2 {
            margin: 0;
            color: #007bff;
        }
        .card a {
            text-decoration: none;
            color: inherit;
            display: block;
        }

        /* Dark Mode Styling */
        .dark-mode .navbar, .dark-mode .sidebar {
            background-color: #333;
        }
        .dark-mode .card {
            background-color: #444;
            color: white;
        }
    </style>
</head>
<body>

    <!-- Navigation Bar -->
    <nav class="navbar">
        <div class="title">Time Table Management</div>
        <button id="darkModeToggle">🌙</button>
    </nav>

    <!-- Sidebar -->
    <aside class="sidebar">
```

```
  <ul>
    <li><a href="addstaff.php">➕ Add Staff</a></li>
    <li><a href="addcourse.php">📚 Add Courses</a></li>
    <li><a href="addclass.php">🏫 Add Classes</a></li>
  <li><a href="template.php">📝 Add Templates</a></li>
  </ul>
</aside>

<!-- Main Content -->
<main class="content">
  <h1>Welcome to Time Table Management</h1>
  <p>Manage staff, courses, and classes with ease.</p>

  <div class="cards">
    <a href="addstaff.php" class="card">
      <h2>🧑‍🏫 Add Staff</h2>
      <p>Manage staff details & workload.</p>
    </a>
    <a href="addcourse.php" class="card">
      <h2>📖 Add Courses</h2>
      <p>Define subjects & credit hours.</p>
    </a>
    <a href="addclass.php" class="card">
      <h2>🏫 Add Classes</h2>
      <p>Assign advisors, sections & manage schedules.</p>
    </a>
  </div>
</main>
```

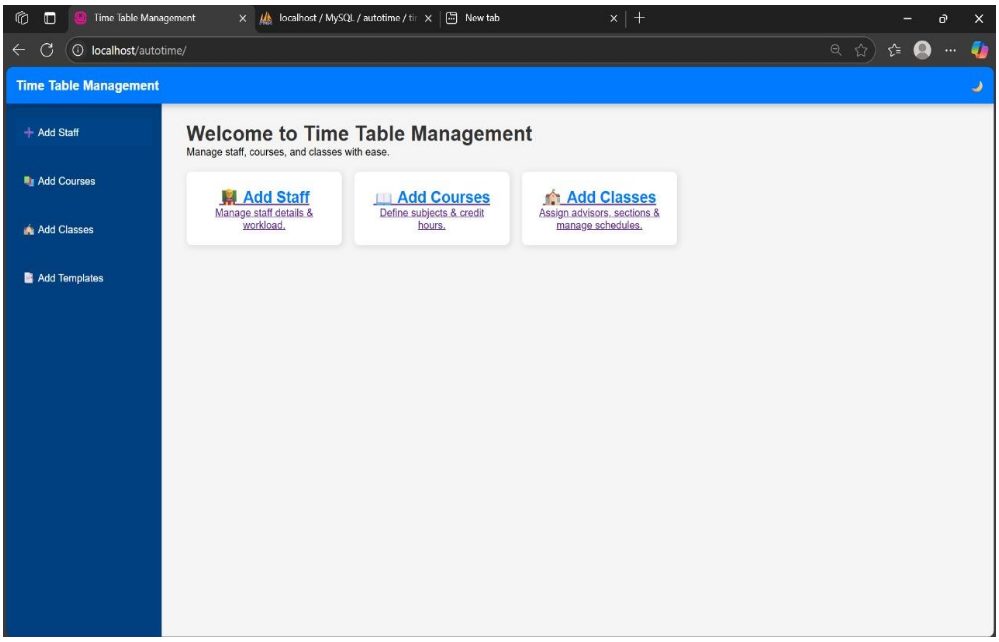
```
    <!-- JavaScript for Dark Mode -->
    <script>
        document.getElementById('darkModeToggle').addEventListener('click',
function () {
            document.body.classList.toggle('dark-mode');
        });
    </script>

</body>
</html>
```

APPENDIX – 2

SCREENSHOTS

Sample Output



Welcome to Time Table Management

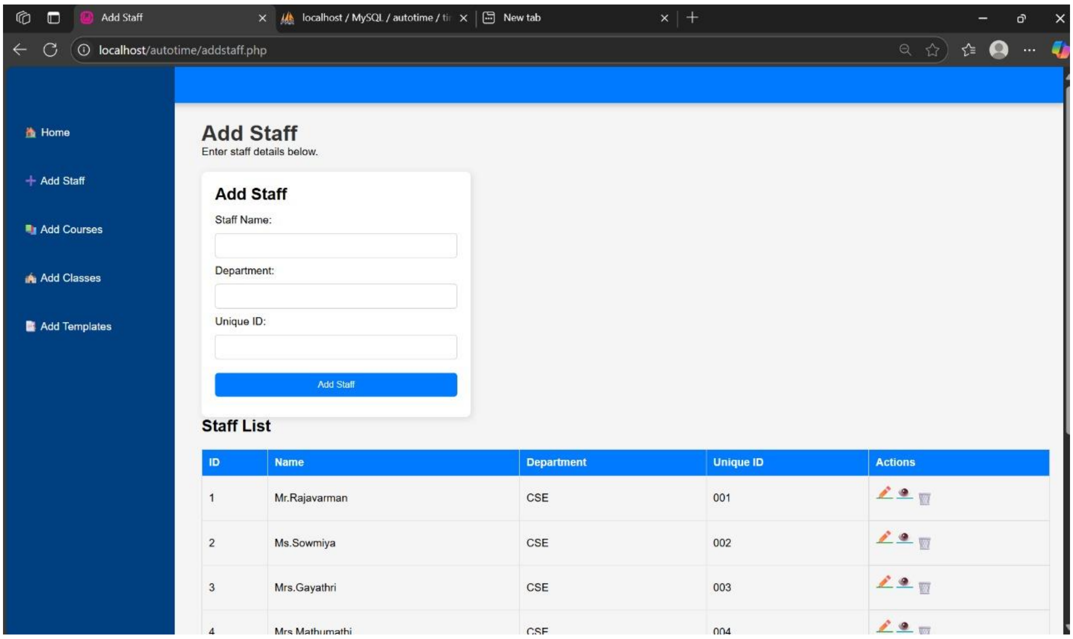
Manage staff, courses, and classes with ease.

Add Staff
Manage staff details & workload.

Add Courses
Define subjects & credit hours.

Add Classes
Assign advisors, sections & manage schedules.

Fig B.1: Index Page


Add Staff
localhost / MySQL / autotime / ti
New tab
localhost/autotime/addstaff.php
Home
Add Staff
Add Courses
Add Classes
Add Templates

Add Staff

Enter staff details below.

Add Staff

Staff Name:

Department:

Unique ID:

Staff List

ID	Name	Department	Unique ID	Actions
1	Mr.Rajavarman	CSE	001	✎ 👁 🗑
2	Ms.Sowmiya	CSE	002	✎ 👁 🗑
3	Mrs.Gayathri	CSE	003	✎ 👁 🗑
4	Mrs.Mathumathi	CSE	004	✎ 👁 🗑

Fig B.2: Add Staff

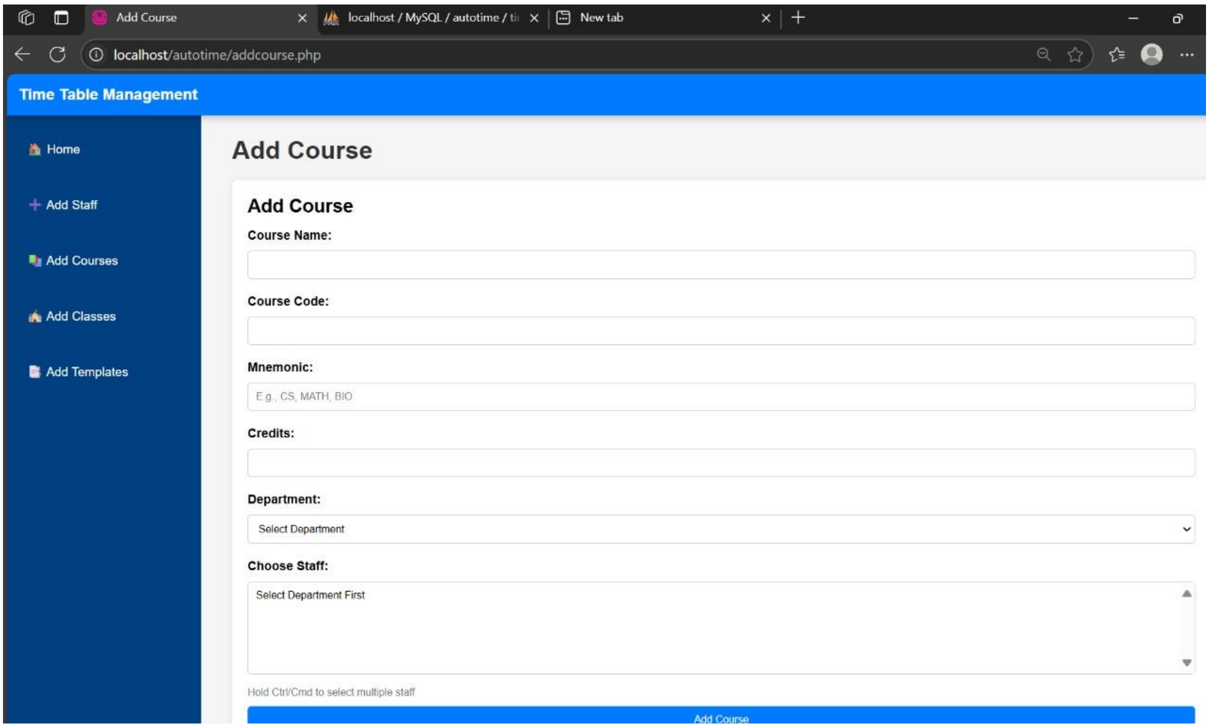


Fig B.3: Add Courses

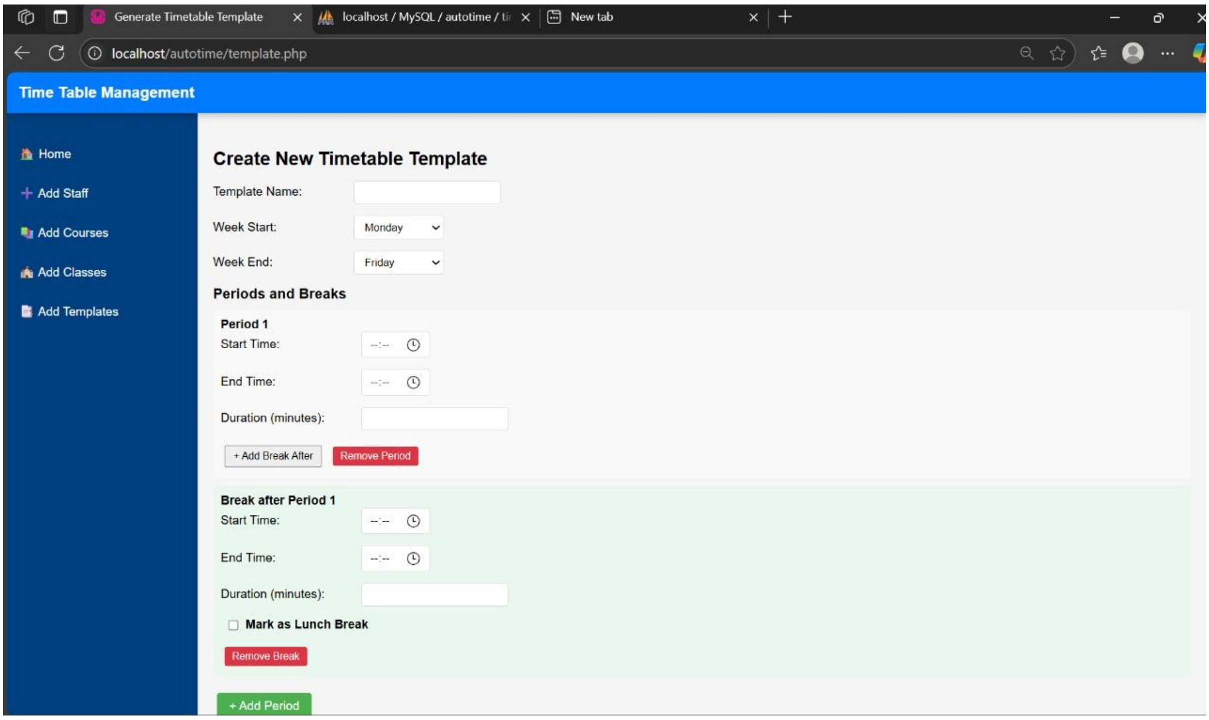
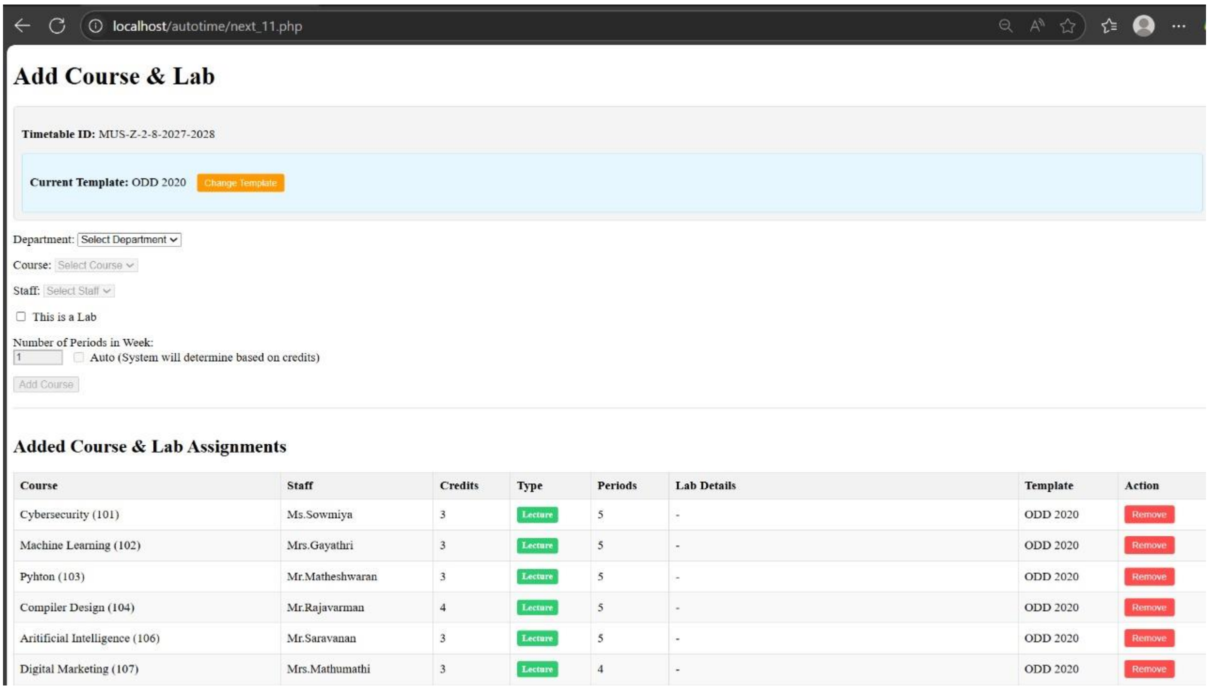


Fig B.4: Add Template


localhost/autotime/next_11.php

Add Course & Lab

Timetable ID: MUS-Z-2-8-2027-2028

Current Template: ODD 2020

Department: Select Department

Course: Select Course

Staff: Select Staff

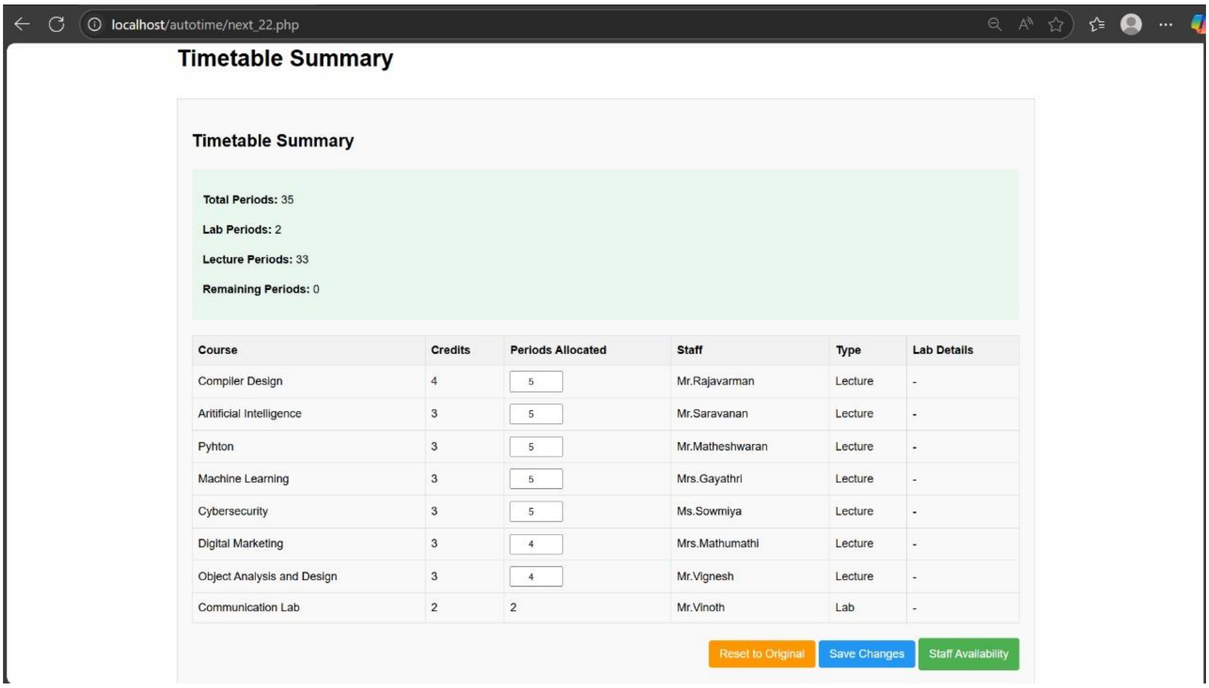
□ This is a Lab

Number of Periods in Week:
1 □ Auto (System will determine based on credits)

Added Course & Lab Assignments

Course	Staff	Credits	Type	Periods	Lab Details	Template	Action
Cybersecurity (101)	Ms.Sowmiya	3	Lecture	5	-	ODD 2020	Remove
Machine Learning (102)	Mrs.Gayathri	3	Lecture	5	-	ODD 2020	Remove
Pyhton (103)	Mr.Matheshwaran	3	Lecture	5	-	ODD 2020	Remove
Compiler Design (104)	Mr.Rajavarman	4	Lecture	5	-	ODD 2020	Remove
Aritificial Intelligence (106)	Mr.Saravanan	3	Lecture	5	-	ODD 2020	Remove
Digital Marketing (107)	Mrs.Mathumathi	3	Lecture	4	-	ODD 2020	Remove

Fig B.5: Add Course and Lab


localhost/autotime/next_22.php

Timetable Summary

Timetable Summary

Total Periods: 35

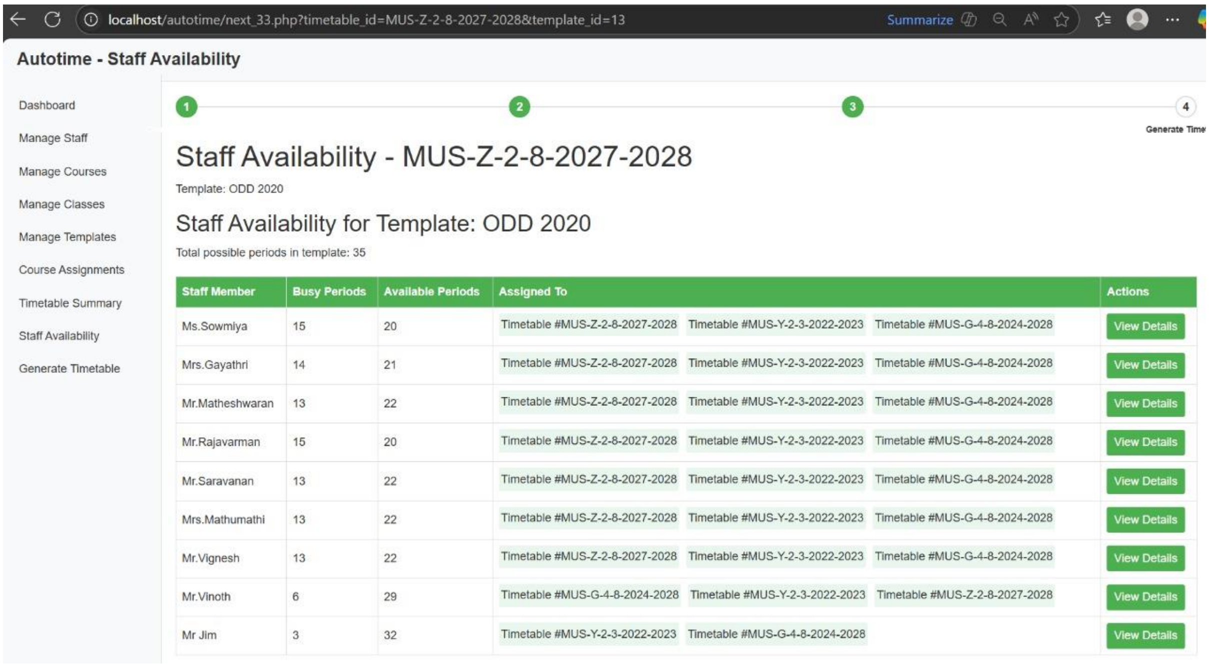
Lab Periods: 2

Lecture Periods: 33

Remaining Periods: 0

Course	Credits	Periods Allocated	Staff	Type	Lab Details
Compiler Design	4	5	Mr.Rajavarman	Lecture	-
Aritificial Intelligence	3	5	Mr.Saravanan	Lecture	-
Pyhton	3	5	Mr.Matheshwaran	Lecture	-
Machine Learning	3	5	Mrs.Gayathri	Lecture	-
Cybersecurity	3	5	Ms.Sowmiya	Lecture	-
Digital Marketing	3	4	Mrs.Mathumathi	Lecture	-
Object Analysis and Design	3	4	Mr.Vignesh	Lecture	-
Communication Lab	2	2	Mr.Vinoth	Lab	-

Fig B.6: Timetable Summary



Staff Availability - MUS-Z-2-8-2027-2028

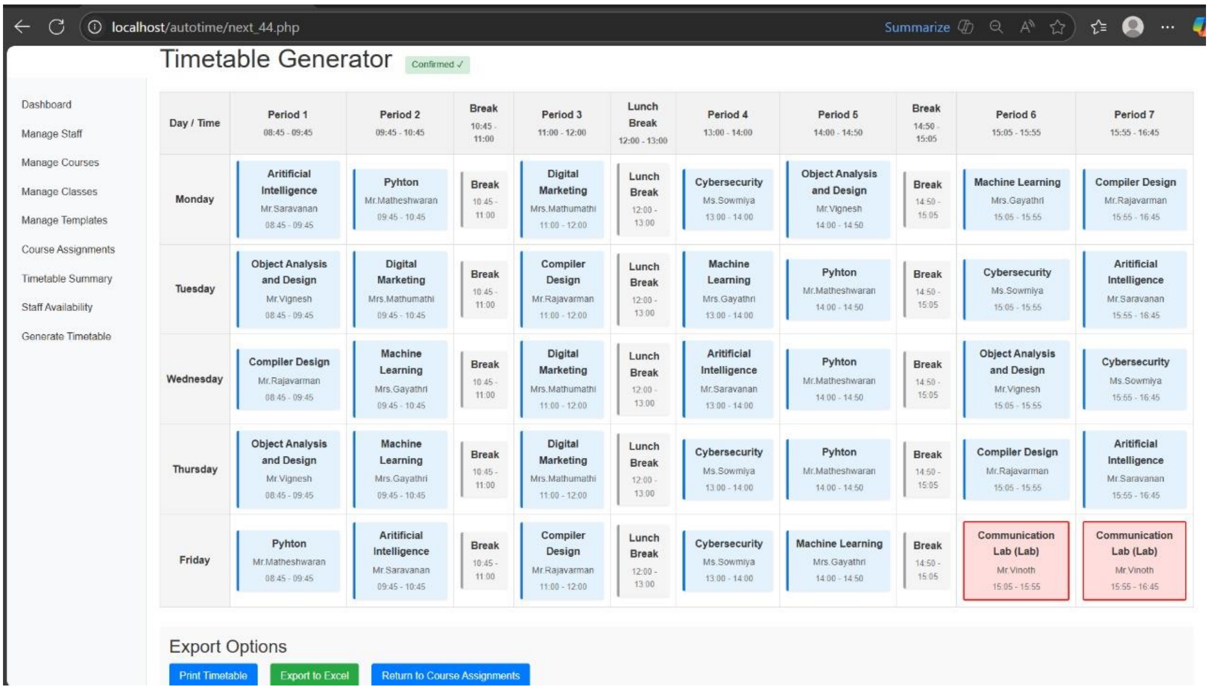
Template: ODD 2020

Staff Availability for Template: ODD 2020

Total possible periods in template: 35

Staff Member	Busy Periods	Available Periods	Assigned To	Actions
Ms.Sowmiya	15	20	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mrs.Gayathri	14	21	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mr.Matheshwaran	13	22	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mr.Rajavarman	15	20	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mr.Saravanan	13	22	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mrs.Mathumathi	13	22	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mr.Vignesh	13	22	Timetable #MUS-Z-2-8-2027-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details
Mr.Vinoth	6	29	Timetable #MUS-G-4-8-2024-2028 Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-Z-2-8-2027-2028	View Details
Mr Jim	3	32	Timetable #MUS-Y-2-3-2022-2023 Timetable #MUS-G-4-8-2024-2028	View Details

Fig B.7: Staff Availability

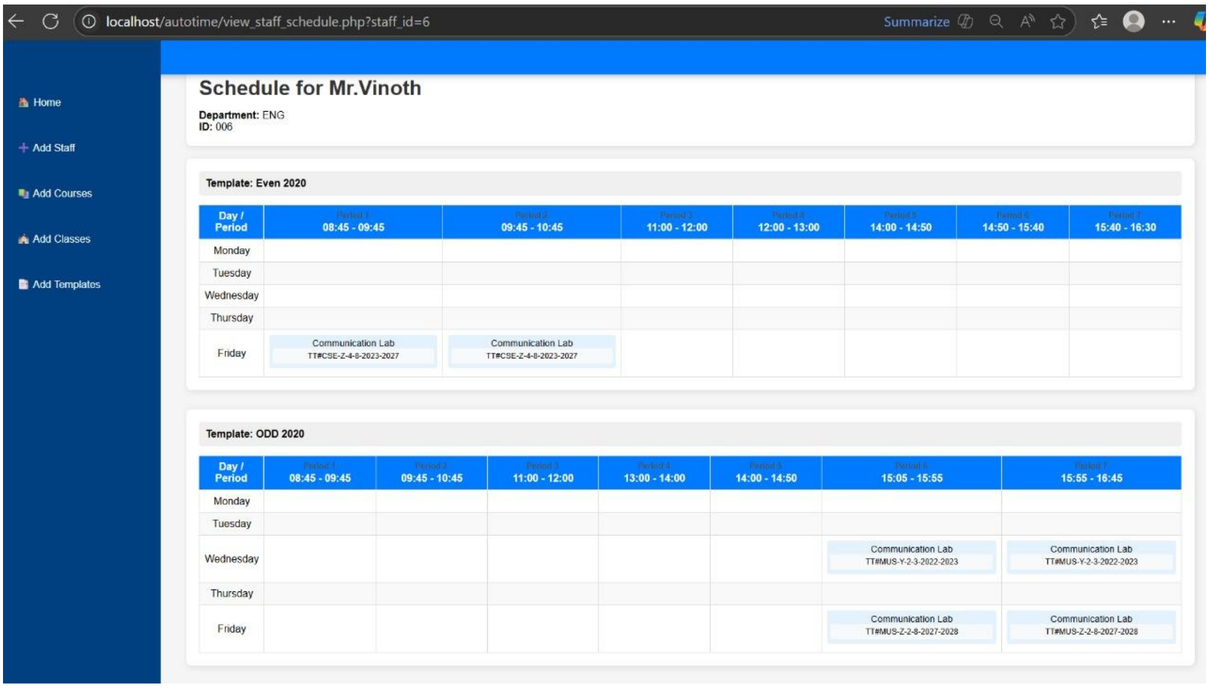


Timetable Generator

Day / Time	Period 1 08:45 - 09:45	Period 2 09:45 - 10:45	Break 10:45 - 11:00	Period 3 11:00 - 12:00	Lunch Break 12:00 - 13:00	Period 4 13:00 - 14:00	Period 5 14:00 - 14:50	Break 14:50 - 15:05	Period 6 15:05 - 15:55	Period 7 15:55 - 16:45
Monday	Aritificial Intelligence Mr.Saravanan 08:45 - 09:45	Pyhton Mr.Matheshwaran 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Object Analysis and Design Mr.Vignesh 14:00 - 14:50	Break 14:50 - 15:05	Machine Learning Mrs.Gayathri 15:05 - 15:55	Compiler Design Mr.Rajavarman 15:55 - 16:45
Tuesday	Object Analysis and Design Mr.Vignesh 08:45 - 09:45	Digital Marketing Mrs.Mathumathi 09:45 - 10:45	Break 10:45 - 11:00	Compiler Design Mr.Rajavarman 11:00 - 12:00	Lunch Break 12:00 - 13:00	Machine Learning Mrs.Gayathri 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Cybersecurity Ms.Sowmiya 15:05 - 15:55	Aritificial Intelligence Mr.Saravanan 15:55 - 16:45
Wednesday	Compiler Design Mr.Rajavarman 08:45 - 09:45	Machine Learning Mrs.Gayathri 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Aritificial Intelligence Mr.Saravanan 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Object Analysis and Design Mr.Vignesh 15:05 - 15:55	Cybersecurity Ms.Sowmiya 15:55 - 16:45
Thursday	Object Analysis and Design Mr.Vignesh 08:45 - 09:45	Machine Learning Mrs.Gayathri 09:45 - 10:45	Break 10:45 - 11:00	Digital Marketing Mrs.Mathumathi 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Pyhton Mr.Matheshwaran 14:00 - 14:50	Break 14:50 - 15:05	Compiler Design Mr.Rajavarman 15:05 - 15:55	Aritificial Intelligence Mr.Saravanan 15:55 - 16:45
Friday	Pyhton Mr.Matheshwaran 08:45 - 09:45	Aritificial Intelligence Mr.Saravanan 09:45 - 10:45	Break 10:45 - 11:00	Compiler Design Mr.Rajavarman 11:00 - 12:00	Lunch Break 12:00 - 13:00	Cybersecurity Ms.Sowmiya 13:00 - 14:00	Machine Learning Mrs.Gayathri 14:00 - 14:50	Break 14:50 - 15:05	Communication Lab (Lab) Mr.Vinoth 15:05 - 15:55	Communication Lab (Lab) Mr.Vinoth 15:55 - 16:45

Export Options

Fig B.8: Timetable Generator



Schedule for Mr.Vinoth

Department: ENG
ID: 006

Template: Even 2020

Day / Period	Period 1 08:45 - 09:45	Period 2 09:45 - 10:45	Period 3 11:00 - 12:00	Period 4 12:00 - 13:00	Period 5 14:00 - 14:50	Period 6 14:50 - 15:40	Period 7 15:40 - 16:30
Monday							
Tuesday							
Wednesday							
Thursday							
Friday	Communication Lab TT#CSE-Z-4-8-2023-2027	Communication Lab TT#CSE-Z-4-8-2023-2027					

Template: ODD 2020

Day / Period	Period 1 08:45 - 09:45	Period 2 09:45 - 10:45	Period 3 11:00 - 12:00	Period 4 13:00 - 14:00	Period 5 14:00 - 14:50	Period 6 15:05 - 15:55	Period 7 15:55 - 16:45
Monday							
Tuesday							
Wednesday						Communication Lab TT#MUS-Y-2-3-2022-2023	Communication Lab TT#MUS-Y-2-3-2022-2023
Thursday							
Friday						Communication Lab TT#MUS-Z-2-8-2027-2028	Communication Lab TT#MUS-Z-2-8-2027-2028

Fig B.9: Staff Timetable

REFERENCES

1. Smith, J., & Garcia, M. (2019). Automated Timetable Scheduling using Genetic Algorithms. *International Journal of Computer Applications*, 45(3), 123–135.

2. Kumar, A., & Patel, M. (2020). A Constraint Programming Approach to University Timetable Scheduling. *Journal of Scheduling Systems*, 12(2), 89–102.

3. Gupta, R., & Lee, A. (2020). Machine Learning for Predicting Academic Performance and Timetable Optimization. *Educational Data Mining Journal*, 8(1), 56–68.

4. Chen, L., & Park, S. (2022). Optimization of Class Timetable Scheduling with Staff Availability Constraints. *Operations Research Letters*, 50(5), 403–412.

5. Fernandez, D., & Malik, A. (2019). Hybrid Genetic Algorithm and Tabu Search for Exam Timetable Scheduling. *Applied Soft Computing*, 75, 1–12.

6. Nair, P., & Mendes, C. (2021). AI-Driven Automated Scheduling for Multi-Department Universities. *Journal of Intelligent Systems*, 30(3), 267–282.

7. Al-Hassan, F., & Das, R. (2022). Real-Time Staff Availability Tracking for Timetable Generation. *Journal of Educational Technology*, 18(2), 78–90.

8. Wong, C., & Tan, Y. (2021). Multi-Objective Timetable Scheduling with Staff Preference Optimization. *Computers & Industrial Engineering*, 153, 107093.

9. Mahmood, A., & Zafar, M. (2019). Automatic Timetable Generator for Educational Institutions. *International Journal of Advanced Computer Science and Applications*, 10(5), 88–95.

10. Omar, K., & Rahman, N. (2020). Comparative Study of Optimization Techniques for Academic Timetable Scheduling. *Journal of Artificial Intelligence Research*, 11(4), 210–224.