EXAMINATION TIMETABLE GENERATION

A PROJECT REPORT

Submitted by,

Trishanya	**-20211CSE0361**
Bhavith Mourya R	**-20211CSE0394**
Dinson Sam Thomas	**-20211CSE0377**
Ajay R	**-20211CSE0384**

Under the guidance of,

Ms. Rakheeba Taseen

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY

BENGALURU

DECEMBER 2024

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report **"EXAMINATION TIMETABLE GENERATION"** being submitted by TRISHANYA, BHAVITH MOURYA R, DINSON SAM THOMAS, AJAY R bearing roll numbers 20211CSE0361, 20211CSE0394, 20211CSE0377, 20211CSE0384 in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

15/1/25

Ms. Rakheeba Taseen
Assistant professor
School of CSE&IS
Presidency University

15/1/25

Dr. Asif Mohammed
Assistant professor& HoD
School of CSE&IS
Presidency University

Dr. L. SHAKKEERA
Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR
Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean -School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **EXAMINATION TIMETABLE GENERATION** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms. Rakheeba Taseen, Assistant Professor, School of Computer Science Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME OF THE STUDENTS	ROLL NUMBER	SIGNATURE
TRISHANYA	20211CSE0361	Trishanya
BHAVITH MOURYA R	20211CSE0394	Bhavith Mourya.R
DINSON SAM THOMAS	20211CSE0377	Dinson
AJAY R	20211CSE0384	Ajay.R

ABSTRACT

The goal of the Examination Timetable Generation Project is to create a sophisticated, automated system that uses HTML, CSS, Java, and Spring Boot to expedite and improve Presidency University's exam scheduling procedure. By addressing the major difficulties and complications involved in creating manual timetables, this project provides a methodical and effective way to create exam schedules free of conflicts. The system makes use of contemporary technology to guarantee that a number of limitations, including workloads for faculty and students, invigilation assignments, and resource availability, are carefully taken into account and managed. In big universities like Presidency University, making exam schedules by hand takes a lot of time and is prone to mistakes. Ineffective use of faculty time, scheduling problems, and fatigue for both students and professors are the results of the old approach's inability to meet the demands. The goal of this project is to automate the procedure, which will guarantee correctness, efficiency, and fairness while lowering administrative burdens. The system provides an optimal schedule that satisfies institutional requirements by taking into account real-time inputs such faculty availability, elective choices, and course registrations. Additionally, automating this procedure reduces the need for human intervention, increasing dependability and freeing up administrative staff to concentrate on other important duties.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and Dr. Asif Mohammed, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Ms. Rakheeba Taseen, Assistant Professor** and Reviewer **Ms. Ayesha Taranum, Assistant Professor**, School of Computer Science Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Trishanya

Bhavith Mourya R

Dinson Sam Thomas

Ajay R

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Generating a timetable is crucial for effectively managing educational institutions and has a significant impact on the academic experience of students and faculty. It requires balancing various factors, including course accessibility, exam scheduling, and resource allocation, while also addressing conflicting demands such as student registrations, faculty availability, and room capacities.

The creation of schedules is essential to the efficient operation of educational establishments. In order to maximize the use of time, resources, and space, a timetable guarantees that lectures, lab sessions, and other academic activities are arranged methodically. Organizing timetables becomes extremely difficult and time-consuming in major universities with several departments, faculty, and courses. Timetable creation has historically been done by hand, requiring administrators to strike a balance between a number of limitations, including classroom capacity, student preferences, and faculty availability. Academic activities may be disrupted by this manual approach's propensity for mistakes, inefficiency, and scheduling conflicts.

The difficulties in managing schedules have been exacerbated by the quick expansion of educational establishments and rising student enrollment. Institutions are currently searching for automated methods that can guarantee conflict-free, optimal timetables, streamline the scheduling process, and minimize manual labor. The goal of this project is to create an Automatic Timetable Generation System that automates and expedites the timetable creation process by utilizing contemporary technologies. Conventional approaches often involve manual processes, which can result in errors and inefficiencies, such as scheduling overlapping exams for students and faculty unavailability. As educational institutions grow and diversify, there is an increasing demand for automated and optimized solutions to ensure a seamless and efficient academic experience.

One of the most important duties in educational institutions is creating exam schedules. It entails allocating classrooms, faculty invigilators, and time slots appropriately while organizing numerous exams across different departments, courses, and elective subjects. Conflicts, mistakes, and inefficiencies are frequently the result of manual scheduling, particularly in educational institutions with sizable student bodies and intricate course designs. Using cutting-edge technologies like Spring Boot, Java, HTML, CSS, and MySQL, this project presents an automated examination timetable generation system that tackles these

issues.

1.1 Need of automation

In modern educational institutions, managing timetables manually has become impractical due to increasing complexities. Some of the key reasons for automation include:

- **Complexity of Constraints**: Balancing teacher availability, classroom allocation, and student course preferences manually is tedious and error-prone. Automated systems handle these constraints systematically.
- **Time Efficiency**: Manual scheduling consumes significant time and effort. Automation reduces the time taken from days to mere minutes.
- **Scalability**: Large institutions with thousands of students and multiple courses need systems that can handle scale. Automated tools can generate schedules for multiple departments simultaneously.
- **Conflict Management**: Manual systems often result in overlapping classes, unassigned classrooms, or double-booked faculty. Automated systems validate schedules to avoid such conflicts.
- **Resource Optimization**: Automated tools ensure optimal use of classrooms, faculty, and available time slots, improving institutional efficiency.
- **Adaptability**: Automated systems can easily adjust schedules based on changes, such as faculty unavailability, rescheduling requirements, or new courses being added.

1.2 Scope of the Project

The Examination Timetable Generation System aims to automate the process of generating exam schedules for educational institutions. The project focuses on resolving the challenges related to efficiently allocating resources (examination rooms, faculty invigilators), handling large datasets, and minimizing conflicts.

Key components of the project's scope include:

1. System Objectives:
    - Automate the examination timetable generation process while ensuring adherence to the institution's constraints and policies.
    - Ensure that the timetable maximizes the usage of available resources, including examination rooms and faculty invigilators, while minimizing conflicts.
    - Provide a user-friendly interface for administrators, allowing them to input

relevant data (such as course and student information) and view generated timetables.

2. Input Data:
    - Courses: Information about the courses being examined, including exam dates, times, durations, and student enrollments.
    - Examination Rooms: Details of the available exam rooms, including their capacity and availability during different time slots.
    - Faculty: A list of faculty members available for invigilation, including their availability and any preferences regarding which exams they can invigilate.
    - Student Enrollment: Data regarding which students are enrolled in which courses, to help detect and resolve any conflicts in their examination schedules.
    - Institutional Constraints: Specific rules or policies that the system must follow, such as no exams after a certain hour or not scheduling exams during lunch breaks.
3. Features:
    - Examination Timetable Generation: Automatically generate a conflict-free timetable that respects all the constraints, including faculty availability, room availability, and student schedules.
    - Conflict Resolution: Detect and resolve conflicts where a student is scheduled for two exams at the same time or a faculty member is assigned to multiple invigilation duties in the same time slot.
    - Optimization: Ensure that the timetable optimizes the use of resources such as exam halls and faculty invigilators while minimizing idle times.
    - User Interface: Provide a dashboard where administrators can view, edit, and manage generated timetables, as well as upload and update input data.
    - Real-Time Adjustments: Allow the system to adapt to changes such as room maintenance, faculty unavailability, or student course changes by regenerating timetables dynamically.
    - Reports and Analytics: Generate reports on resource utilization, scheduling conflicts, and other relevant metrics for administrators to assess the efficiency of the timetable.
4. System Boundaries:
    - The project will focus exclusively on generating examination timetables and

will not include other types of academic scheduling (e.g., regular class timetables or extracurricular activities).

- The project will focus on the academic exams and will not address non-academic events or practical exam setups, such as labs or sports exams.

5. Limitations:
    - The system will not handle every edge case, such as extreme student requests or non-standard examination requirements.
    - Real-time updates may require manual intervention for certain changes that fall outside the scope of automated scheduling (e.g., last-minute faculty absences or emergency room unavailability).

1.3 Significance

This project holds significant value for educational institutions, particularly universities like Presidency University, as it addresses an ongoing problem-optimizing the exam schedule.
In addition to saving a significant amount of time and money, automating this procedure improves the administration, staff, and student experience. Among the particular advantages are:

- Efficiency: Ensures a more streamlined procedure by lowering the amount of manual labor needed to create timetables.
- Fairness: Makes ensuring that no instructor or student is overworked or faced with conflicting demands at the same time.
- Scalability: As the number of students and courses rises, larger institutions can use the system because it is scalable.
- Optimization: By eliminating needless overlaps or gaps, algorithms that optimize the schedule guarantee better use of available resources.

1.4 Technologies used

The following contemporary technologies are used in the implementation of the Automatic Timetable Generation System:

- Java: Java's stability, object-oriented features, and platform independence make it the main programming language. It serves as the cornerstone for data processing and backend logic.
- Spring Boot: By offering tools for quick setup and configuration, Spring Boot streamlines

the development of Java applications. It is employed in the development of backend RESTful APIs for managing and creating timetables. Spring Boot is perfect for this project because of its features, which include dependency injection, MVC architecture, and JPA integration.

- MySQL: Classes, instructors, classrooms, and timetable entries are all stored in MySQL, a relational database management system. It makes it possible to handle and query structured data with relationships between different tables in an efficient manner.
- Java Persistence API, or JPA: Object-Relational Mapping (ORM) is made possible by JPA, which facilitates seamless communication between the MySQL database and Java applications. By transforming Java objects into database entities, it streamlines data persistence.
- Thymeleaf (Frontend): Thymeleaf templates are used to render dynamic HTML pages that are integrated with backend data when a web interface is established.
- Eclipse IDE: The Java application code is written, tested, and debugged using these IDEs.
- Maven: The build tool Maven controls project dependencies, guaranteeing that Spring Boot and other libraries integrate seamlessly.

1.5 Challenges

Complex scheduling logic can result from managing several constraints, including workload distribution, clash-free scheduling, and resource availability.

It can be computationally difficult to guarantee that all constraints are met without creating conflicts.

- Problems with Scalability: If not appropriately optimized, the system may experience performance bottlenecks as the number of courses, students, and constraints rises.
- Limitations on Resources: Scheduling issues may arise from a lack of classrooms or invigilators during exam peak periods.
- Accuracy of Real-Time Data: An improper schedule may result from inaccurate or missing input data (such as incorrect faculty availability or course registration problems).
- The complexity of algorithms: For huge datasets, it might be difficult to implement and optimize algorithms like the Greedy Algorithm to guarantee timely results.
- Connecting to Current Systems: It can take more work to integrate the timetable system if the university uses its current administrative tools.

- Adoption and Training of Users: To utilize the system effectively, administrative staff may require training, which could initially postpone implementation**.**

CHAPTER-2

LITERATURE SURVEY

Exam scheduling is a critical task for colleges and universities that requires careful consideration of several factors, such as available classroom space, and instructor availability. This overview examines the technologies and methods used in the production of exam schedules, highlighting advancements and challenges over time.

At the outset of the discussion, the efficiency and efficacy of contemporary automated techniques are contrasted with those of conventional manual procedures. It looks at ways to deal with complex constraints, such as avoiding scheduling conflicts, modifying for overlapping needs, and ensuring equitable resource allocation. Optimization methods, including algorithmic and heuristic approaches, are analyzed in detail to demonstrate how they improve the scheduling process.

2.1 Existing Methods or Frameworks for Creating Timetables

In educational institutions, creating timetables has changed from being done by hand to being done by highly automated systems. To automate and optimize the process, a number of tools, algorithms, and tactics have been created over time. Here, we'll go over current tools, research gaps, and both conventional manual and contemporary automated ways.

2.2 Manual Techniques for Creating Timetables

Historically, administrative personnel created timetables by hand in educational institutions. This procedure entails:

- Classroom Assignment: Depending on space availability, classrooms are assigned to various courses.
- Instructor Assignment: Assigning instructors to classes based on their qualifications and availability.

Making sure that classes are planned during times that don't conflict is known as time slot management.

However, there are a number of drawbacks to creating timetables by hand:

- Time-consuming: Creating, modifying, and rescheduling timetables using manual techniques takes a lot of administrative work.
- Error-prone: Common errors include disputes in assignments, double-booked teachers, and overlapping classes.

- Limited Scalability: It becomes impractical to manage a large number of students, teachers, and courses as institutions expand.
- Inefficiency: Teachers or classrooms may be underutilized as a result of administrative staff's inability to deploy resources as efficiently as possible.

2.3 Automatic Systems for Generating Timetables

Automated timetable generation solutions have been developed to address the drawbacks of manual scheduling. These systems create conflict-free and efficient schedules by utilizing algorithms and optimization approaches. Typical methods include of:

- Rule-based Systems: These systems create timetables by adhering to preset guidelines and limitations (such as class scheduling, teacher availability, and room capacity). Although they are easier to use, they might not provide really well-optimized solutions.
- Solutions for the Constraint Satisfaction Problem (CSP): A CSP is frequently used to describe schedule creation, in which the system aims to allocate resources (teachers, classrooms, and time slots) while meeting a number of requirements. These limitations can include room capacities, teacher availability, and the avoidance of course overlap. Choco Solver and Google OR-Tools are well-known CSP solvers.
- Genetic Algorithms: Genetic algorithms (GAs) develop a population of possible solutions by applying the laws of natural selection. By choosing the optimal solutions based on fitness factors (like minimizing conflicts), the algorithm iterates and improves the schedule across generations. GAs can be computationally costly, yet they are quite effective for large-scale issues.
- Greedy Algorithms: These algorithms choose the best option at every stage (for example, allocating a course to the first teacher available) without taking into account earlier decisions. They might not always offer the optimal global solution, despite their computational efficiency.

A probabilistic optimization method called "simulated annealing" looks for the optimal schedule by experimenting with various combinations and progressively lowering the likelihood of making significant adjustments over time. Although it can be slow, it's very helpful for avoiding local minima.

2.4 Algorithms Used in Timetable Generation

- Greedy Algorithm: Quick and simple approach where resources are assigned based on availability, but may not optimize overall efficiency.
- Genetic Algorithms: Mimics natural selection to explore and optimize timetable generation through iterative processes.
- Simulated Annealing: A probabilistic technique used to explore the solution space and minimize conflicts by gradually cooling the system.
- Backtracking Algorithms: Used for constraint satisfaction problems, where resources are assigned and checked recursively to ensure no conflicts.

Sl.No	TITLE OF THE PAPER/AUTHOR/PUBLISHER/YEAR	ADVANTAGES	LIMITATIONS
1.	Automatic Timetable Generator Authors: Prof. Jyothi Patil, Shambhavi V, Sneha N T, Sweta Jadhav, Tahura Sadaf IJRASET,2023	The Automatic Timetable Generator efficiently automates the scheduling process, significantly reducing manual effort and minimizing conflicts in examination and class schedules.	potential scalability issues with very large datasets, sensitivity to initial conditions in genetic algorithms
2.	AUTOMATIC TIMETABLE GENERATION SYSTEM Authors:Rajshri Firke, Pratiksha Bhabad, Omkar Gangarde, Abhimannyu Magar, Prof. Anuja Tawlare IJCRT,2023	The Automatic Timetable Generation System streamlines the scheduling process by automating timetable creation, ensuring optimal use of resources and reducing scheduling conflicts.	handling large-scale user demands, complexities in ensuring data privacy and security, and the need for ongoing updates to address user feedback

3.	Design and Implementation of An Automatic Examination Timetable Generation and Invigilation Scheduling System Using Genetic Algorithm Authors:Abdulaziz Aminu;WahyuCaesarendra;UmarSHaruna;Abubakar Sani;MansurSa'id;Daniel S Pamungkas;Sumantri R Kurniawan; Endang Kurniawan IEEE,2019	The system utilizes a genetic algorithm to efficiently generate optimal examination timetables and invigilation schedules, minimizing conflicts and enhancing resource allocation.	limited accessibility as it's currently desktop-based, and challenges in handling complex timetable conflicts
4.	An integer programming approach to curriculum-based examination timetabling Authors:Cataldo, A., Ferrer, J.-C., Miranda, J., Rey, P. A., & Saure, A. Annals of Operations Research-2017	The study employs an integer programming approach to effectively address curriculum-based examination timetabling, providing optimal scheduling solutions that adhere to various constraints.	computationally expensive and may struggle with larger datasets due to the complexity of the integer programming formulation, leading to longer solution times.
5.	Multi-objective optimization for exam scheduling to enhance the educational service performance Authors:Abdallah, K. S The Journal of Management and Engineering Integration-2016	The study introduces a multi-objective optimization approach that improves exam scheduling efficiency, thereby enhancing overall educational service performance.	The reliance on heuristic methods like genetic algorithms can result in long computational times and difficulty in ensuring consistent quality of solutions across diverse scheduling scenarios.
6.	A great deluge algorithm for a real-world examination timetabling problem Authors:Mohmad Kahar, M. N., & Kendall, G. Journal of the Operational Research Society-2015	The study demonstrates the effectiveness of a great deluge algorithm in solving real-world examination timetabling	tuning of parameters, and its performance can vary significantly depending on the specific characteristics

		problems, providing a robust method for optimizing scheduling while handling various constraints.	of the timetabling problem being addressed.
7.	Real-life curriculum-based timetabling Authors: Müller, T., & Rudová, H Proceedings of the 9th International Conference on the Practice and Theory of Automated Timetabling—PATAT-2012	The research explores real-life curriculum-based timetabling challenges and solutions, emphasizing practical applications of automated scheduling techniques to enhance efficiency and effectiveness.	Constraint programming can be limited by scalability issues, as it may struggle with performance in larger, more complex scheduling scenarios due to the increased number of constraints and variables involved.
8.	A survey of search methodologies and automated system development for examination timetabling Authors:Qu, R., Burke, E., McCollum, B., Merlot, L., & Lee, S. Journal of Scheduling-2009	This survey analyzes diverse search methodologies for automated examination timetabling, offering insights into their effectiveness and guiding future developments in scheduling systems.	The potential inefficiencies in finding optimal solutions are a limitation of these methodologies, as many of them depend on shortcuts that could result in local optima instead of globally optimal schedules.
9.	A survey and case study of practical examination timetabling problems Authors: Cowling, P., Kendall, G., & Hussin, N. M. Journal of Scheduling-2001	The survey presents a detailed analysis of practical examination timetabling challenges, offering valuable insights and solutions based on real-world case studies	The use of metaheuristic techniques can result in unreliable outcomes, as their performance is greatly influenced by

			the configuration of parameters and the intricacy of the scheduling scenarios they are applied to.
10.	Examination timetabling: Algorithmic strategies and applications. Authors:Carter, M. W., Laporte, G., & Lee, S. Y. Journal of the Operational Research Society-1996	The study provides a comprehensive overview of various algorithmic strategies for examination timetabling, highlighting their practical applications and effectiveness in optimizing scheduling solutions.	The techniques can struggle with scalability and efficiency, as greedy algorithms may yield suboptimal solutions and integer programming can be computationally intensive for large datasets.

1.1 Table of literature survey

CHAPTER-3

RESEARCH GAPS OF EXISTING METHODS

There has been a lot of research on the topic because developing an examination timetable is a difficult combinatorial optimization problem with many constraints and variables. The current manual and automated methods both provide answers, but they are still limited in their applicability in real-world situations by problems and constraints. There are numerous notable research gaps in the methods used today to create exam timetables, including the following:

- Inefficiency in Constraint Handling Problem: To create examination schedules, current approaches frequently employ basic heuristic algorithms (such as greedy algorithms, simulated annealing, or genetic algorithms). These methods may not be able to effectively manage the increasing complexity of limitations in larger institutions or more sophisticated examination systems, even while they are effective for smaller datasets.

  Research Gap: A lot of approaches fail to effectively balance conflicting restrictions such as:

  Availability of faculty: Making sure that instructors aren't overworked by tests that are too close to one another. Keeping students from taking too many tests in a short amount of time or on consecutive days might lead to student tiredness. Allocating rooms: Organizing a small number of rooms and making sure that tests with lots of students don't take place in the same space. More sophisticated optimization methods, such machine learning and multi-objective optimization. Algorithms that consider these intricate limitations may provide superior outcomes.

- Scalability to Larger Data Sets Problem: A lot of conventional methods perform well with smaller datasets, including those with comparatively few students and courses. However, the current algorithms encounter performance constraints as the number of students, courses, and teachers rises.

  Research Gap: Scalable technologies that can efficiently manage the massive volumes of data usually involved in creating exam schedules for the entire university are lacking. Large datasets of student registrations, course offers, staff availability, and room scheduling are harder to manage as universities expand. Solutions must be made to scale effectively when the quantity of resources, students, and courses increases.

- Multi-Objective Optimization Problem: A lot of current systems just optimize for one goal, such lowering the quantity of time slots or conflicts. But in practice, there are frequently several conflicting goals like reducing student exhaustion by distributing tests across several days, balancing the responsibilities of faculty, making the most of the rooms that are available, ensuring that tests are distributed evenly throughout various timeslots.

  Research Gap: There aren't many strategies that can successfully balance several competing goals at once. Research on multi-objective optimization methods that can offer trade-offs between these goals is still in progress.

- Lack of Real-Time Performance and Efficiency Problem: Conventional timetable production techniques frequently require a substantial amount of computation time, especially for big institutions with a high volume of students and courses. These systems are usually batch-oriented, which means that in order to produce a solution, the entire data set must be cycled through.

  Research Gap: Real-time or almost real-time schedule generation systems that can swiftly create and modify timetables despite dynamic data changes are required. Techniques like parallel processing, cloud-based solutions, and distributed computing could greatly increase the timetable creation process's performance and efficiency.

- Insufficient Attention to User-Friendly Interfaces Problem: A lot of current systems prioritize back-end optimization methods above front-end interfaces and user experiences. As a result, administrators, teachers, and students could find it difficult to comprehend or use the system efficiently.

  Research Gap: User-friendly interfaces that make it simple for administrators to enter data, make changes, and view the final schedule are becoming more and more necessary. It should also be simple for teachers and students to examine the schedules and report problems. In addition to optimization, modern systems should prioritize usability.

CHAPTER-4

PROPOSED METHODOLOGY

4.1 Problem Definition:

Timetable scheduling in educational institutions can become quite complicated for a number of reasons. Managing and creating timetables by hand for numerous classes, instructors, classrooms, and students is more challenging as institutions get bigger and more. This complexity results from the necessity to take into account a number of dependencies and limitations in order to make sure that the schedules are effective, free from conflicts, and compliant with institutional norms.

The process of generating examination timetables in educational institutions is a highly complex task, especially when dealing with large numbers of students, courses, and constraints. Managing the examination schedules manually often leads to conflicts, inefficient use of resources, and overall scheduling challenges. This complexity is exacerbated by the need to consider various factors, such as the availability of examination halls, the number of students appearing for each exam, faculty invigilation requirements, and student workload.

Some of the challenges in managing examination timetables include:

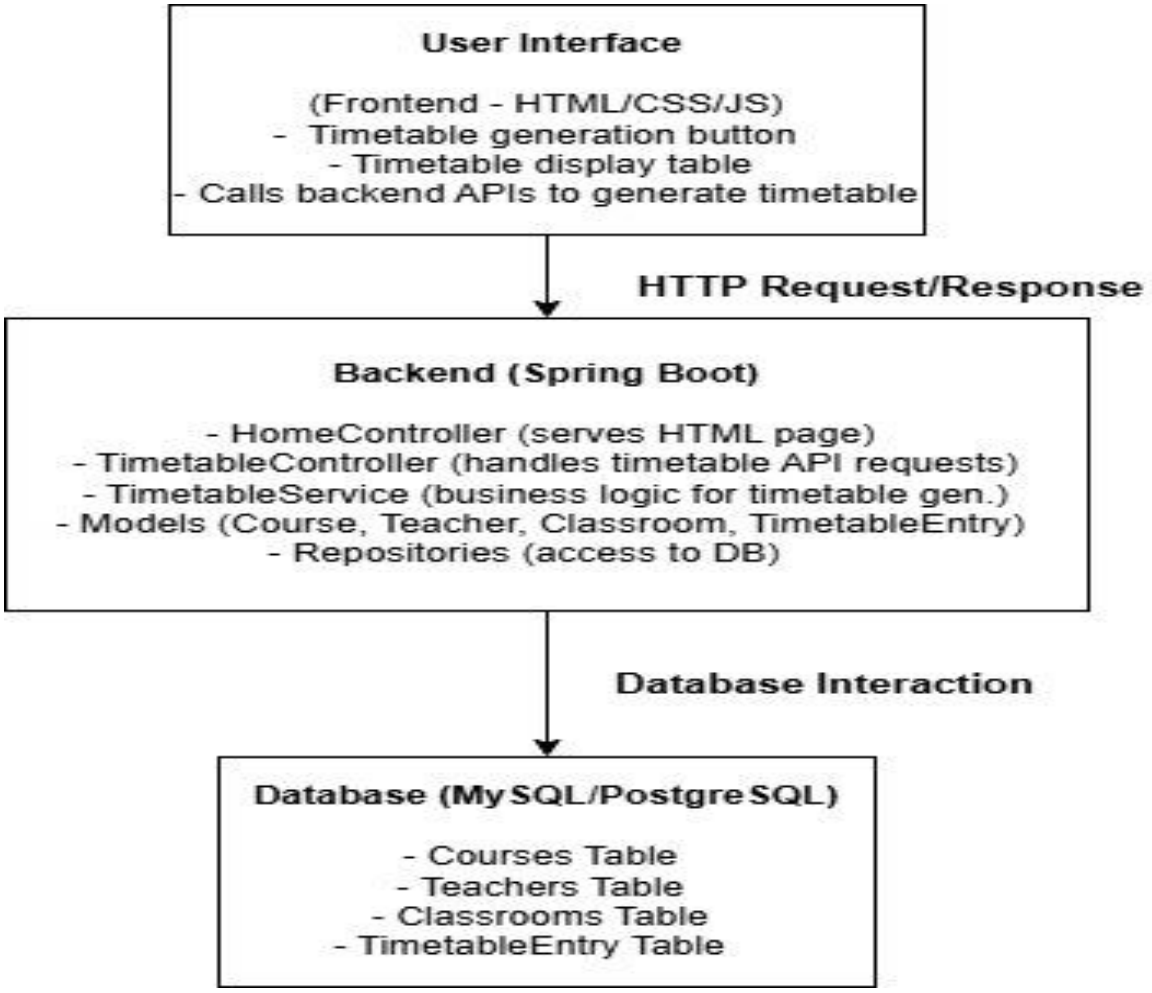
- Large Dataset Handling: Educational institutions often have hundreds of courses with many sections, each requiring individual exams. This results in a large dataset of courses, students, and exam rooms, making manual scheduling impractical.
- Room Availability: Exam halls have varying capacities and require careful allocation. The availability of these rooms during specific timeslot windows must be considered to avoid scheduling conflicts.
- Faculty Invigilation: Different exams require faculty members to be assigned as invigilators, and faculty availability must be taken into account to ensure that exams are properly invigilated without overloading any single instructor.
- Student Conflict: Students may be enrolled in multiple courses that have conflicting examination times. Identifying and resolving such conflicts is crucial to prevent students from being scheduled for multiple exams at the same time.
- Real-time Changes: Unexpected changes like room maintenance, faculty unavailability, or sudden student enrollment changes can occur, making it necessary to dynamically adjust the examination timetable.

Constraints to Consider

The examination timetable generation system needs to handle several constraints that impact the scheduling process:

- Room Availability: Each exam requires a suitable room based on the number of students, with some exams needing larger rooms or specialized setups. The system must allocate rooms considering their capacity and availability at specific times.
- Faculty Availability for Invigilation: Faculty members must be assigned to invigilate exams, and the system should ensure that no faculty is assigned more than one exam during a time slot. Faculty schedules must be checked to ensure they are available to invigilate exams.
- Examination Duration: Different exams may have different durations, and the system should account for this by not scheduling longer exams in time slots that cannot accommodate them.

4.2 Design Procedure:



4.3 Overview of the design

The project utilizes the Java Spring Boot framework for back-end development and MySQL for database management. The system will include various features for managing examination schedules, resources, and constraints. The code is structured in a modular way, allowing each component (course, timetable entry, teacher, classroom) to interact smoothly and ensuring a flexible timetable generation process.

Key Components of the Code:

1. Controller Classes:
    - The HomeController serves the home page with a simple GET request.
    - The TimetableController is responsible for handling API endpoints like generating the timetable and fetching all timetable entries.
2. Model Classes:
    - Classroom: Represents examination rooms with attributes like id, name, and capacity.
    - Course: Represents courses with attributes like id, name, and code.
    - Teacher: Represents teachers with id, name, and email for assigning invigilation duties.
    - TimetableEntry: Represents a timetable entry with details like courseId, teacherId, classroomId, dayOfWeek, startTime, and endTime.
3. Repository Classes:

    These interface classes interact with the MySQL database for CRUD operations on Classroom, Course, Teacher, and TimetableEntry entities.
4. Service Class:

    The TimetableService handles the logic of generating the examination timetable. It fetches all courses, teachers, classrooms, and other relevant data from the database and allocates them based on availability, student enrollments, and other constraints.
5. Spring Boot Application:

    The TimetableGeneratorApplication is the entry point for the Spring Boot application. It runs the system, manages resources, and connects all the components together.

4.4 Steps:

Step 1: Data Collection and Storage

- Input Data:
    - Course registration and elective information will be gathered from the university's records and stored in a MySQL database.
    - Faculty availability for invigilation duties and room capacities will also be stored.
- Database Setup:
    - Design the database schema using MySQL to store details such as courses, students, exam durations, rooms, and faculty schedules.
    - Each student's registered courses and electives will be linked to avoid scheduling conflicts.

Step 2: Define Constraints

- Hard Constraints:
    - A student cannot have two exams at the same time.
    - Faculty members cannot invigilate multiple exams simultaneously.
    - Room capacities must be respected during exam assignments.
    - No exams should exceed the room's available hours.
- Soft Constraints:
    - Minimize the number of exams scheduled per day for students and faculty to avoid exhaustion.
    - Distribute exams evenly over the examination period to prevent student and faculty burnout.

Step 3: Backend Development (Java and Spring Boot)

- Java will be used to implement the core logic that handles the scheduling algorithm.
- Spring Boot will serve as the application framework, facilitating communication between the frontend, backend, and database.
- Scheduling Algorithm: Use optimization techniques to slot exams in the timetable while respecting the constraints.
    - The timetable generation logic will pull data from the MySQL database and process it to generate valid, conflict-free exam slots.

Step 4: Frontend Development (HTML/CSS/JavaScript)

- A user-friendly web interface will be developed using HTML/CSS/JavaScript.

- The interface will allow administrators to:
  - View and adjust the generated timetable.
- The interface will display timetables in a structured, easy-to-read format.

Step 5: Timetable Generation

Spring Boot will act as the middle layer, receiving input from the frontend, processing it with the scheduling algorithm, and storing the results in the database.

4.5 Key Functionalities and Features

- Timetable Generation: The system handles the automatic assignment of exams to rooms, teachers, and timeslots. It works by ensuring that:
  - Rooms are assigned based on their capacity.
  - Teachers are not assigned multiple exams at the same time.
  - Timetable slots are optimized to avoid overlaps and maximize resource utilization.
- Conflict Management: The system prevents conflicts where:
  - A student has multiple exams scheduled at the same time.
  - A teacher is assigned to invigilate multiple exams at the same time.
- Room and Faculty Management: The system ensures that rooms are allocated based on the number of students and exam requirements (e.g., lab exams, lecture-based exams). It also ensures that faculty are assigned as invigilators in a conflict-free manner, ensuring no double-booking.
- Database Interactions: The system uses MySQL to store and manage data such as courses, teachers, classrooms, and timetable entries. The repository classes handle data retrieval and storage seamlessly.

CHAPTER-5

OBJECTIVES

The creation of an effective, automated method for creating exam schedules that are optimal while accounting for different institutional constraints is the main objective of the Examination Timetable Generation project. The project's specific goals are listed below:

- Automate Exam Timetable Generation Goal: The primary objective is to create and put into place an automated system that creates university exam schedules. The initiative aims to save administrative personnel a great deal of time and money by automating the process, which will remove human mistake and manual labor from creating timetables.

  Method: The system will automatically create a schedule that complies with the given restrictions after receiving input data, such as course registrations, faculty availability, and room allocations.
- Allocation of Resources Should Be Optimized Goal: Reduce waste and underutilization by making ensuring that all available resources—including rooms, invigilators, and faculty—are used effectively during the exam time.

  Method: To reduce disputes, the system will allocate resources as efficiently as possible. For instance, by distributing tests evenly among available days, the system will prevent many exams from being scheduled in the same room at the same time, guarantee that faculty members are not overworked, and reduce student tiredness.
- Manage Several Restrictions at Once Goal: The system must manage several conflicting constraints and provide the optimal solution while taking into account each one. This comprises:

  Faculty Availability: Making certain that test dates are set in accordance with faculty members' availability.

  Room Availability: Making the most use of the rooms that are available and making sure that large classes with lots of students are given rooms that are the right size.

  Ensuring that there are adequate invigilators available for all tests throughout the designated time periods is known as invigilator availability.

  Method: To balance these limitations and provide an ideal schedule that meets each of these requirements, the system will integrate sophisticated algorithms.

- Offer a User Interface That Is Easy to Use Goal: Create an intuitive user interface that enables administrators and other university employees to quickly enter the required information, view the timetable that is produced, and make any necessary modifications. The final schedule should also be simple for teachers and students to access.

  Method: To create a clear, responsive, and user-friendly interface, HTML, CSS, and JavaScript will be used in the system's front-end design.
- Make sure it's flexible and scalable Goal: As the institution grows, the system should be scalable, which means it should be able to accommodate the increasing number of staff, students, and courses.

  Method: Java and Spring Boot will be used to construct the back-end system, guaranteeing that the architecture can grow with bigger datasets. Additionally, the system should be adaptable enough to manage unforeseen circumstances (such faculty absences or last-minute course enrollment) without requiring a lot of manual labor.
- Create Timetables Without Conflict Goal: Make sure the created schedule is free of conflicts and steers clear of typical scheduling problems like overlapping tests, double-booked rooms, or a lack of invigilators.

  Method: The system will use optimization techniques to find a workable timetable that adheres to the guidelines for each limitation (e.g., room capacity, invigilator limits, etc.) and minimize conflicts.
- Facilitate Updates and Modifications in Real Time Goal: If there are any last-minute cancellations, staff absences, or room shortages, the system should make it simple and quick to amend the schedule.

  Method: By enabling administrators to make modifications in real time, the system will save time and effort by immediately updating the schedule rather than requiring a complete regeneration.
- Ensure Fairness and Balance in Timetable Generation Goal: Make sure that no group is unduly taxed and that all students, faculty, and resources receive equitable treatment. For instance, unless it is absolutely required, faculty members should not have tests scheduled back-to-back, and students should not have exams on consecutive days.

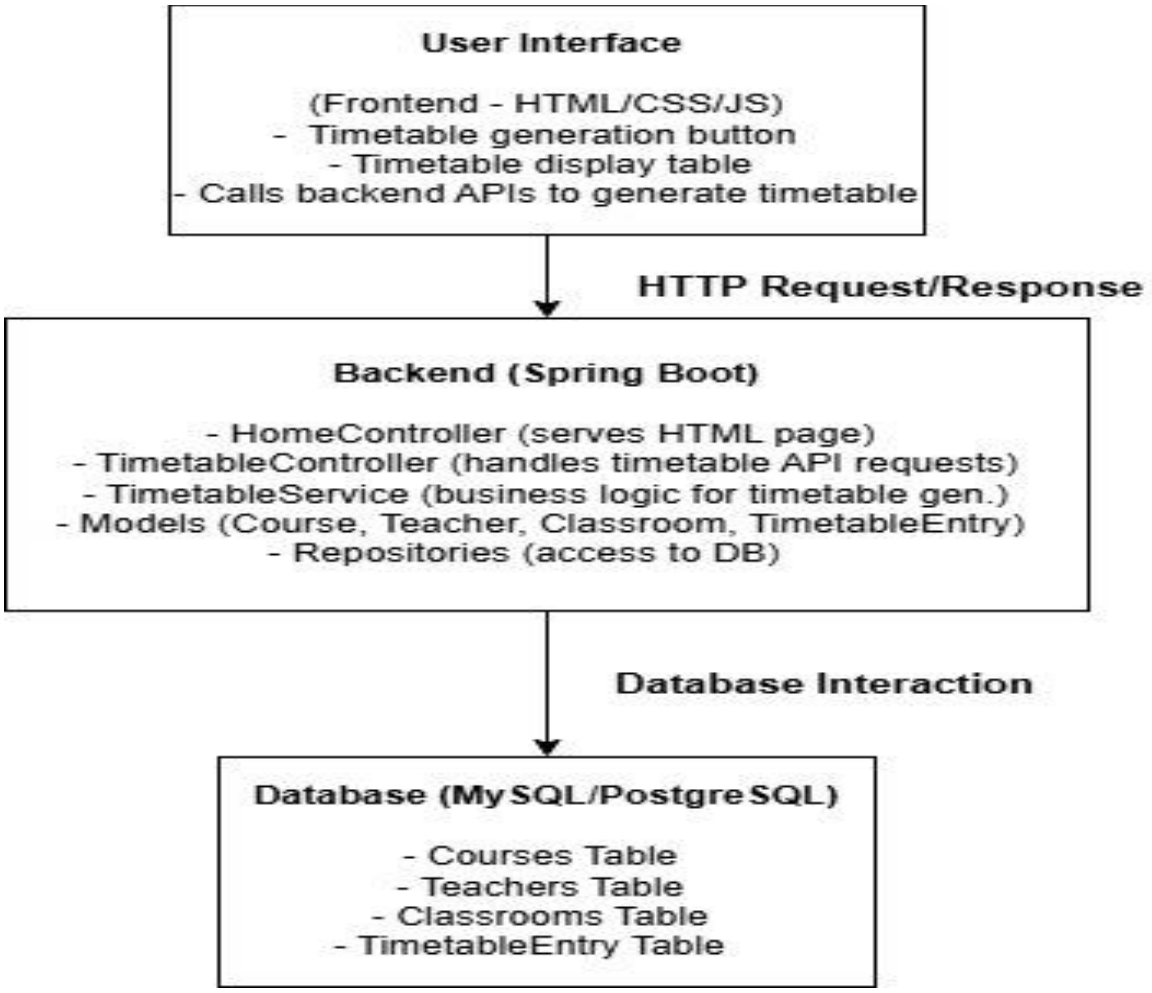
  Method: To make sure that everyone's schedule is balanced, the system will integrate fairness criteria into its optimization algorithms.

- Encourage Upcoming Improvements: The system should be built with the intention of being easily enhanced in the future, for example, by adding new features or incorporating user feedback.

  Method: Because the system will be modular and flexible, it will be possible for further iterations to include features like managing more intricate limitations or connecting with student portals to provide real-time notifications.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION



The purpose of the system is to automate the creation of class, instructor, and classroom schedules. It combines a relational database (like MySQL or PostgreSQL) with a front-end user interface and a backend Spring Boot application. The system's architecture and implementation, broken down into the Frontend, Backend (Spring Boot), and Database levels, are shown below.

1. **The user interface, or frontend Technology**

   JavaScript (using the Thymeleaf template engine), HTML, and CSS. A straightforward user interface for interacting with the system is offered via the frontend. It possesses: a button that initiates the creation of the schedule.A table that shows the timetable that was created.

Essential Features:

Creating a Timetable: The "Generate Timetable" button causes a JavaScript code to call the backend API in order to build the schedule when the user clicks it.Display Timetable: Following the creation of the timetable, the table is populated by a second API call that retrieves the timetable items.

2. Spring Boot backend

All business logic pertaining to the creation of timetables and database interactions is managed by the backend. It is made up of various parts:

- Controllers: Manage requests and answers using HTTP.
- Services: Business logic used to create and oversee the schedule.
- Database activities are managed by repositories.

Important Elements:

When a user accesses the home URL (/), the HomeController serves the HTML page.

TimetableController: Provides access to APIs for creating and retrieving schedule entries.

The business logic used to create the schedule is contained in the TimetableService.

Models or entities: Show the Course, Teacher, Classroom, and TimetableEntry data structures.

Storage facilities:

- TimetableRepository: Manages timetable entry CRUD operations.
- CourseRepository: Manages course CRUD functions.
- TeacherRepository: Manages teachers' CRUD tasks.
- In the classroomRepository: Manages classroom CRUD operations

3. Database (PostgreSQL/MySQL)

Course, teacher, classroom, and timetable data are all stored in the database. Relational tables are used to keep the entities' relationships intact.

Database Entities:

- Course information (id, name, and code) is stored in the Courses Table.
- Teachers Table: Holds teacher information (name, email, and ID).
- Classrooms Table: Holds information about the classroom (id, name, and capacity).
- TimetableEntry Table: Holds the created schedule, including the day, start time, and end time, as well as references to the class, teacher, and course.

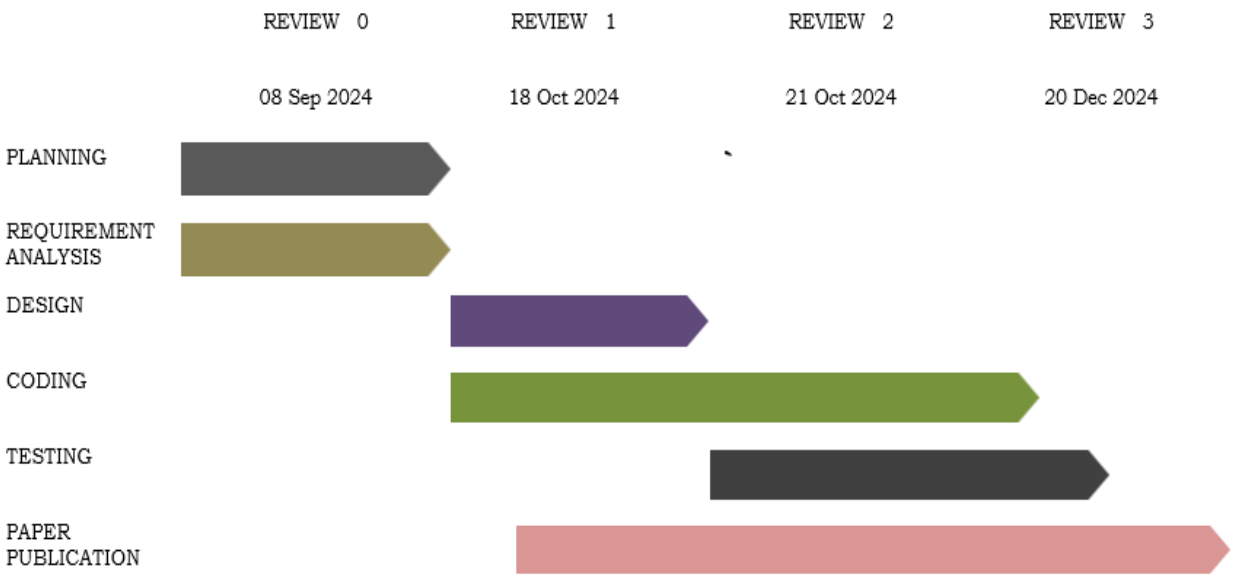
4. User Interaction and System Flow:

The frontend (HTML page) is how the user accesses the system. To request that the schedule be generated, the user hits the "Generate Timetable" button.

- Calls to APIs: The timetable generation process is started by sending a GET request to /api/timetable/generate.
- To produce the timetable, the backend invokes the TimetableService's generateTimetable () function. The user receives a success message once the timetable has been generated.
- Showing the Timetable: To retrieve all of the created timetable entries, another GET request is made to /api/timetable/entries. The frontend refreshes the HTML table with the timetable data after receiving it in JSON format.
- Database Communication: JPA repositories are used by the backend to communicate with the database, saving and retrieving information as required.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)



CHAPTER-8

OUTCOMES

The Presidency University's automated examination timetable generation project aims to achieve the following outcomes:

- Creation of a well-structured timetable that prevents exam overlaps for students taking multiple courses, ensuring fair access to exams.

- Efficient allocation of faculty and examination rooms to minimize conflicts and maximize resource utilization.

- Reduction of administrative workload through automation, enabling staff to focus on other important tasks while producing accurate and timely schedules.

- Improvement of the academic experience for students by providing a fair and manageable examination schedule, thus reducing stress and fatigue.

- Development of a flexible system that can easily adapt to changes in course offerings, student enrollments, and faculty assignments, ensuring long-term usability.

CHAPTER-9

RESULTS AND DISCUSSIONS

9.1 Results

The following tasks are successfully completed by the system:

- Creating a Timetable: The backend creates a schedule based on pre-established rules. It ensures that limitations like instructor availability, classroom capacity, and course duration are honored by allocating courses, teachers, and classrooms to particular days and time intervals. The system can manage numerous classes, instructors, and classrooms, allocating them to a schedule that spans several days.
- Display of the Timetable: The created timetable is dynamically shown in tabular form on the frontend. The user is notified that the schedule was successfully generated after selecting the "Generate Timetable" button. When it is successful, the system retrieves the schedule data from the backend and adds the course, teacher, classroom, day, start time, and end time details to the database.
- Integrity of Data: All pertinent information (course, teacher, classroom, day, start and end hours) is precisely recorded, and the schedule entries are successfully stored in the database. Because every course, instructor, and classroom assignment is verified before being saved, the system makes sure that there are no scheduling conflicts.
- User Engagement: The UI is engaging and easy to use. Users can clearly view the schedule thanks to the table structure.A clear call to action is provided by the "Generate Timetable" button, which allows users to create the timetable with just one click.

9.2 Discussions

- Scalability: More classes, instructors, and classrooms may be accommodated by the system with ease. Future additions and modifications (such adding more restrictions or managing more complicated scheduling requirements) are possible due to the modular and adaptable logic used to allocate courses to time slots and classrooms.
- Automatic Conflict Resolution: Because every timetable entry is verified before being saved, the system makes sure that there are no scheduling conflicts, such as teachers or classrooms being booked twice.
- Usability: The frontend offers a straightforward and user-friendly interface for creating and viewing timetables. The user experience is improved by the use of notifications

and automatic table population.

- Separation of Concerns: The system is made easier to manage by keeping the frontend, backend, and database separate from one another. While the frontend concentrates on user interaction and presentation, the backend manages business logic.

9.3 Possible Drawbacks and Enhancements:

- Complex restrictions: The existing system may not be able to manage increasingly complicated scheduling requirements, even while it manages fundamental restrictions like teacher and classroom availability. To further improve the timetable creation process, for example, restrictions like student preferences, course prerequisites, and overlapping course timings should be introduced.

  Enhancement: To handle more complicated constraints and provide a more optimized schedule, we can implement optimization techniques (e.g., Genetic Algorithms) or algorithmic approaches like Constraint Programming (CP).
- Error Handling: There is not much error handling in the current implementation. The system might not react correctly if the database malfunctions or if an unforeseen circumstance occurs (such as an invalid teacher or classroom).

  Enhancement: Including thorough error handling in the frontend and backend, along with input data validation and failure recovery techniques, will guarantee more seamless operation in practical settings.
- Performance: If the system is not optimized, the timetable creation process may become sluggish for large-scale applications with thousands of classes, instructors, and classrooms. Large datasets may render the existing algorithm ineffective.

  Enhancement: Performance would be improved by putting in place more effective methods, including parallel processing or optimization algorithms. Caching frequently used data may also enhance responsiveness and decrease database requests.
- UI/UX Improvements: Although the existing user interface works well, it might be improved with features like filtering choices, a drag-and-drop interface for manual timetable alterations, and more user-friendly feedback systems.

  Enhancement: The user experience would be improved by including graphical representations of the schedule (such as a calendar or grid style) and offering real-time updates.

- Security: At the moment, the system lacks security features like authorization and authentication. If the system is to be deployed in a real-world situation, this would be required, particularly to prevent illegal access to timetable data.

  Enhancement: Production-grade systems must incorporate security features including secure API endpoints, role-based access control (RBAC), and login systems.
- Design of Databases: Only the most important fields pertaining to classes, instructors, classrooms, and timetable entries are stored in the present, straightforward database design. To offer more sophisticated capabilities, it might be expanded to incorporate more specific information (such as room capacity, teacher credentials, and student enrollment).

  Enhancement: To make the system more resilient and adaptable, the schema might incorporate other things like Students, Rooms, Course Types, and Preferences.

9.4 Upcoming Improvements

- Optimization: As previously indicated, more sophisticated optimization algorithms or AI-based strategies might be used to automatically modify the schedule in response to other limitations like class capacity, student preferences, etc.
- Mobile Application: By creating a mobile version of the system, users would be able to view the schedule while on the go and get notifications or updates in real time.
- Integration with Other Systems: To reduce human input and increase productivity, the system might be coupled with current college and school administration systems to automatically pull data for classes, instructors, and classrooms.

CHAPTER-10

CONCLUSION

The goal of the Examination Timetable Generation project is to provide a solution that tackles the difficulties and complexities associated with developing exam schedules by streamlining and automating the scheduling process for educational institutions. Exam scheduling is successfully automated by the technology, which lessens human mistake and the administrative burden. Time is saved and resources are used more effectively when the process is streamlined. In summary, a major advancement in the administration of academic calendars is the Examination Timetable Generation system. Many of the issues that educational institutions confront are addressed by the system, which offers a more equitable and effective schedule, optimizes resources, and automates the scheduling process. It has a significant effect on faculty, students, and administrators, increasing productivity, lowering stress levels, and boosting the quality of the academic experience in general. With further improvements, the technology might completely transform exam scheduling in higher education, offering expanding institutions a flexible option and guaranteeing efficient management of workloads and resources.

REFERENCES

[1] Lindahl, M., Sørensen, M., & Stidsen, T. R. (2018). A fix-and-optimize matheuristic for university timetabling. Journal of Heuristics, 24(4), 645–665. (https://doi.org/10.1007/s10732-018-9371-3)

[2] Cataldo, A., Ferrer, J.-C., Miranda, J., Rey, P. A., & Saure, A. (2017).An integer programming approach to curriculum-based examination timetabling. Annals of Operations Research, 258(2), 369–393. (https://doi.org/10.1007/s10479-016-2321-2)

[3] Abdallah, K. S. (2016). Multi-objective optimization for exam scheduling to enhance the educational service performance. The Journal of Management and Engineering Integration, 9(1), 14–23.

[4] Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society, 47(3), 373–383. (https://doi.org/10.1057/jors.1996.37)

[5]https://programmer2programmer.net/live_projects/Synopsis/download_project_synopsis.aspx?JAVA-Time-Table-Generation-System-JAVA-MySQL&id=145

[6] https://stackoverflow.com/questions/12209268/scheduling-timetable-algorithm

[7] R. Firke, P. Bhabad, O. Gangarde, A. Magar, and A. Tawlare, "Automatic Timetable Generation System," *IJCRT*, vol. 2023, pp. 1-5, 2023.

[8] Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society, 47(3), 373–383. (https://doi.org/10.1057/jors.1996.37)

[9] Kendall, G., & Hussin, N. M. (2004).A tabu search hyper-heuristic approach to the examination timetabling problem. Informs Journal on Computing, 16(3), 270-276. (https://doi.org/10.1287/ijoc.1030.0017)

[10] Murray, K., & Wilson, C. (2014). Solving the university examination timetabling problem with a multi-stage algorithm. Computers & Operations Research, 56, 81-90. (https://doi.org/10.1016/j.cor.2014.01.005)

APPENDIX-A

PSUEDOCODE

1. Controller layer

1.1 Home Controller

HomeController {

Define method `index` that returns the "index" view.

}

1.2 TimetableController

TimetableController {

Define method `generateTimetable` that calls TimetableService's `generateTimetable` method and returns a success message.

Define method `getAllTimetableEntries` that fetches all timetable entries from the TimetableService and returns them.

}

2. Model layer

2.1 Classroom

Classroom {

Define fields `id`, `name`, `capacity`.

}

2.2 Course

Course {

Define fields `id`, `name`, `code`.

}

2.3 Teacher

Teacher {

Define fields `id`, `name`, `email`.

}

2.4 TimetableEntry

TimetableEntry {

Define fields `id`, `courseId`, `teacherId`, `classroomId`, `dayOfWeek`, `startTime`, `endTime`.

}

3. Repository layer

Repository Layer {

Define repositories for `Classroom`, `Course`, `Teacher`, `TimetableEntry`.

}

4. Service layer

4.1 TimetableService

TimetableService {

Method `generateTimetable`:

Fetch all `courses`, `teachers`, and `classrooms`.

Define `daysOfWeek`, `startTime`, `endTime`, `duration`.

For each course:

For each day of the week:

Set current time to start time.

While current time + class duration < end time:

Assign teacher and classroom.

Create `TimetableEntry` with course, teacher, classroom, day, start time, and end time.

Save the entry to the database.

Move to the next available time slot.

Move to next teacher and classroom in a round-robin fashion.

```
    Method `getAllTimetableEntries`:
        Fetch and return all `TimetableEntry` from the repository.
}
```

5. Main Application

5.1 Timetablegenerator

```
MainApplication {
    Run the Spring Boot application with
`TimetableGeneratorApplication`.
}
```

6. HTML code

```
BEGIN

    // Initialize web page
    Display Title "Timetable Generator"
    Display Header "Welcome to the Timetable Generator"
    Display a Button "Generate Timetable"
    Display a Table with headers: "Day", "Course", "Teacher",
"Classroom", "Start Time", "End Time"

    // Button click triggers timetable generation
    On Button "Generate Timetable" click:
        CALL generateTimetable()

    FUNCTION generateTimetable()
        // Send request to server to generate timetable
        SEND request to '/api/timetable/generate'
        IF request is successful THEN
            DISPLAY alert "Timetable generated successfully!"

            // Fetch and display the generated timetable entries
```

```
SEND request to '/api/timetable/entries'
IF request is successful THEN
   PARSE JSON response into timetable entries

   // Empty the table body before populating it
   EMPTY timetable-body

   // Loop through each timetable entry and create table rows
   FOR each entry in timetable entries DO
      CREATE a new table row (<tr>)
      FILL row with the following data:
         - Day of Week
         - Course ID
         - Teacher ID
         - Classroom ID
         - Start Time
         - End Time
      APPEND row to timetable-body

END
```

APPENDIX-B

SCREENSHOTS

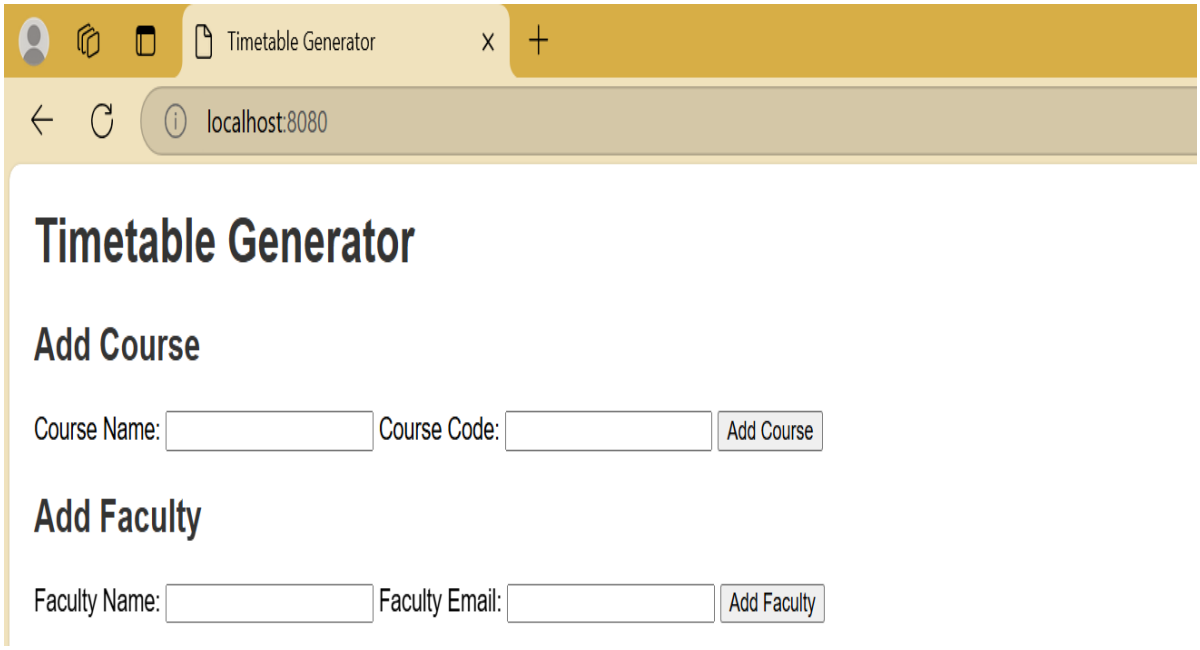


Fig 1. Course and faculty details entry

This page allows you to enter the details of the course and also details of the faculty and entered details are stored in the database

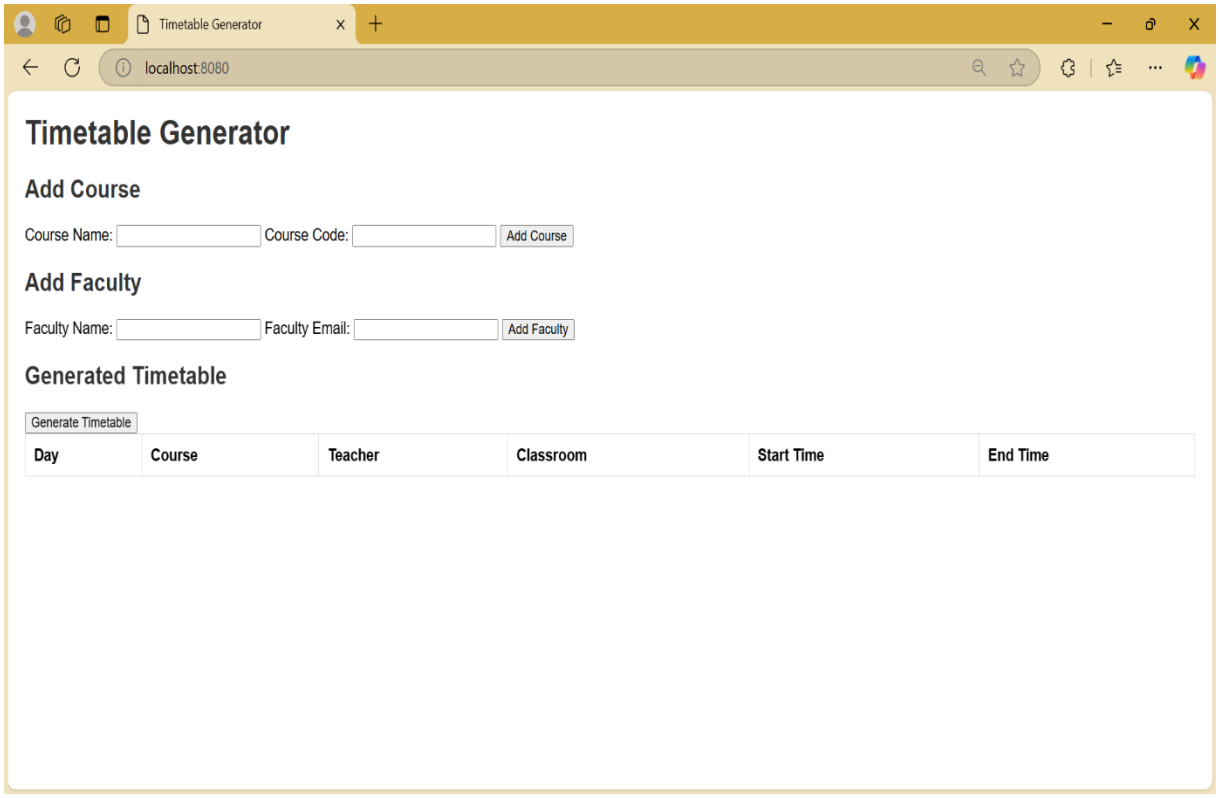


Fig 2. Home page

This page displays the generated timetable.

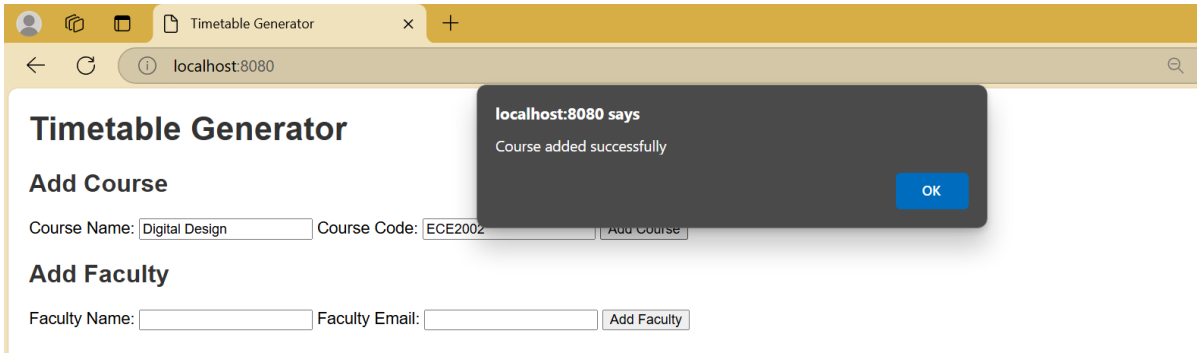


Fig 3. Course details

Course details entered and are stored in the database

```
mysql> select * from course;
+----+----------------+---------+
| id | name           | code    |
+----+----------------+---------+
|  1 | java           | CSE3001 |
|  2 | C++            | CSE2001 |
|  3 | C#             | CSE2002 |
|  4 | dbms           | CSE3010 |
|  5 | html           | CSE0354 |
|  6 | Digital Design | ECE2002 |
+----+----------------+---------+
6 rows in set (0.00 sec)
```

Fig 4.sql course page

This is the demonstration of the course details which are stored.

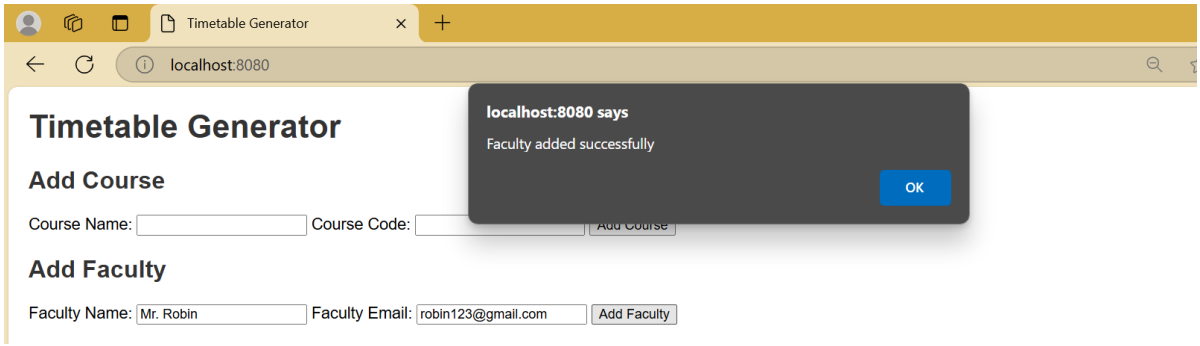


Fig 5. Faculty details

Faculty details entered and are stored in the database

```
mysql> select * from teacher;
+----+---------------+-----------------------------------------+
| id | name          | email                                   |
+----+---------------+-----------------------------------------+
|  1 | Mr Sandeep    | sandeep123@gmail.com                    |
|  2 | Mr Alex       | alex23@gmail.com                        |
|  3 | Mr Suresh     | suresh3@gmail.com                       |
|  4 | ABC           | abc@gmail.com                           |
|  5 | Ms. Rakheeba  | rakheeba.taseen@presidencyuniversity.in |
|  6 | Mr. Robin     | robin123@gmail.com                      |
+----+---------------+-----------------------------------------+
```

Fig 6.sql faculty page

This is the demonstration of the faculty details which are stored.

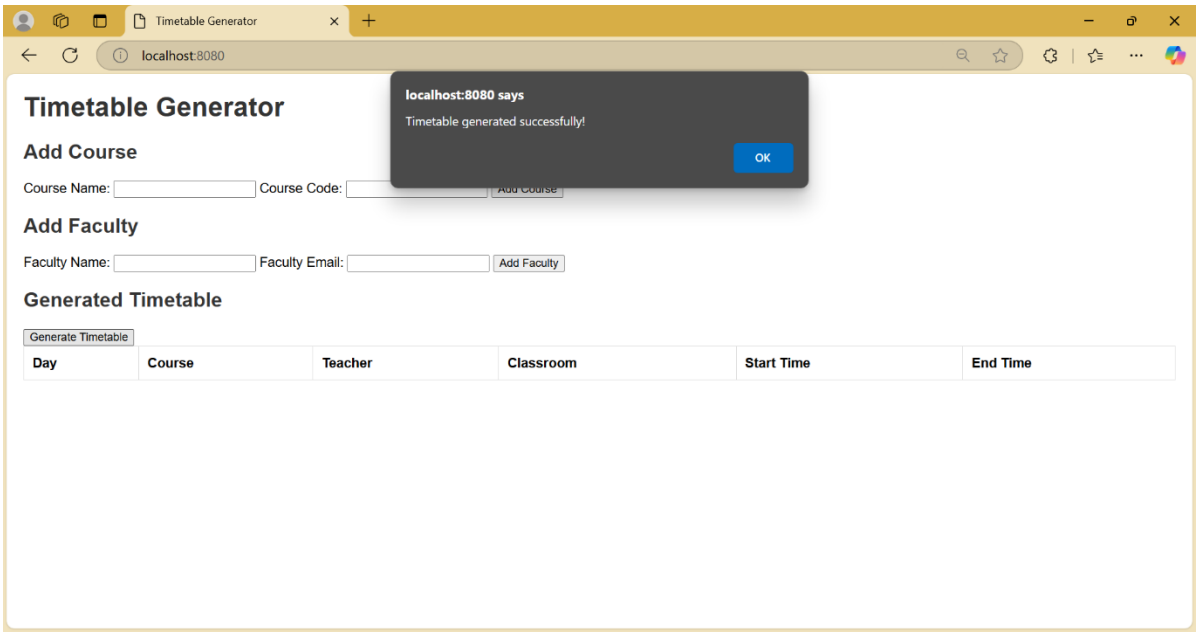
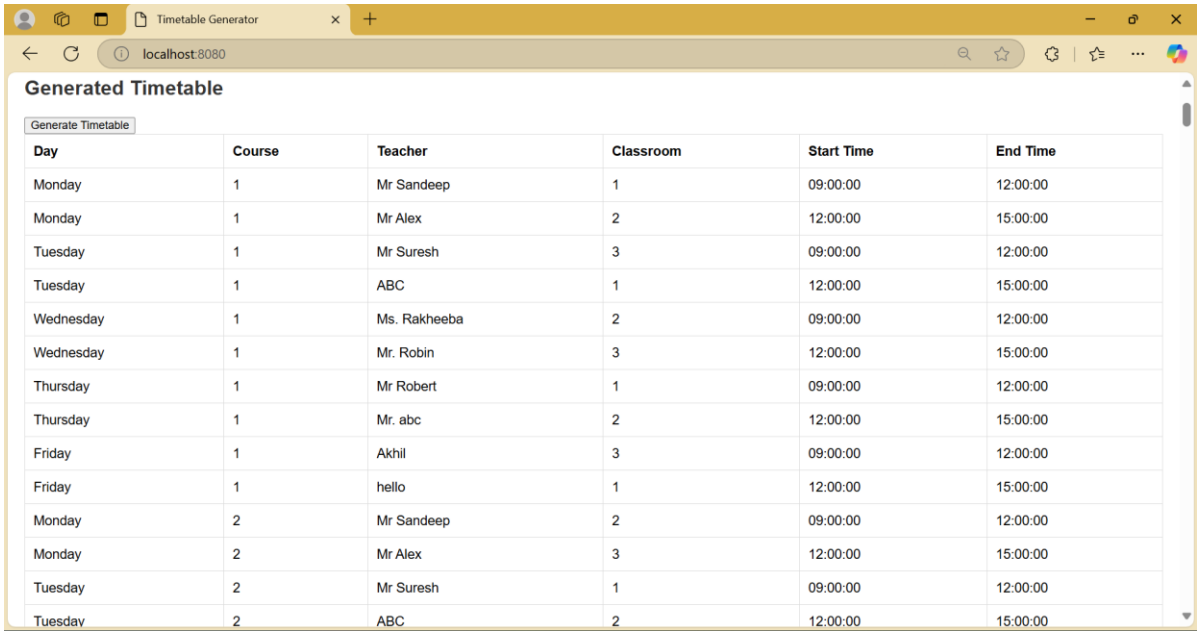


Fig 7. Msg dialog box

After clicking submit button, timetable is generated and msg displays as "timetable generated successfully"



Day	Course	Teacher	Classroom	Start Time	End Time
Monday	1	Mr Sandeep	1	09:00:00	12:00:00
Monday	1	Mr Alex	2	12:00:00	15:00:00
Tuesday	1	Mr Suresh	3	09:00:00	12:00:00
Tuesday	1	ABC	1	12:00:00	15:00:00
Wednesday	1	Ms. Rakheeba	2	09:00:00	12:00:00
Wednesday	1	Mr. Robin	3	12:00:00	15:00:00
Thursday	1	Mr Robert	1	09:00:00	12:00:00
Thursday	1	Mr. abc	2	12:00:00	15:00:00
Friday	1	Akhil	3	09:00:00	12:00:00
Friday	1	hello	1	12:00:00	15:00:00
Monday	2	Mr Sandeep	2	09:00:00	12:00:00
Monday	2	Mr Alex	3	12:00:00	15:00:00
Tuesday	2	Mr Suresh	1	09:00:00	12:00:00
Tuesday	2	ABC	2	12:00:00	15:00:00

Fig 8. Timetable

This page displays the generated timetable based on all the given constraints

TRISHANYA Examination Time Table Generation

ORIGINALITY REPORT

4%	2%	4%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Anandakumar H, Arulmurugan R. "Next Generation Wireless Communication Challenges and Issues", 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019 Publication	2%
2	Olavo Alves Diogo, Eduardo Raupp de Vargas, Peter Fernandes Wanke, Abdollah Hadi-Vencheh. "Longitudinal bibliometric analysis applied to home care services", Computer Methods and Programs in Biomedicine, 2021 Publication	1%
3	Peter Demeester, Burak Bilgin, Patrick De Causmaecker, Greet Vanden Berghe. "A hyperheuristic approach to examination timetabling problems: benchmarks and a new problem from practice", Journal of Scheduling, 2011 Publication	1%
4	Submitted to University of Sheffield Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Examination Time Table Generation

Ms. Rakheeba Taseen

School of computer science and engineering

Presidency University

Rakheeba.taseen@presidencyuniversity.in

Trishanya

School of computer science and engineering

Presidency University

TRISHANYA.20211CSE0361@presidencyuniversity.in

Bhavith Mourya R

School of computer science and engineering

Presidency University

BHAVITH.20211CSE0394@presidencyuniversity.in

Ajay R

School of computer science and engineering

Presidency University

AJAY.20211CSE0384@presidencyuniversity.in

Dinson Sam Thomas

School of computer science and engineering

Presidency University

DINSON.20211CSE0377@presidencyuniversity.in

ABSTRACT-Effective organization of examination forms is essential for educational institutions to ensure fair and efficient administration of examinations while meeting various requirements. The objective of this project is to create an automated system for generating assessment forms that takes into account important input data such as elective courses, course registration numbers, and predefined restrictions. The system will take into account important aspects including availability of teaching staff, effective accountability, and student teacher balance. In addition, it will limit the duration of the daily examination to reduce student fatigue. The methodology used ensures optimal allocation of time slots while maintaining fairness and practicality through a structured approach that defines constraints, domains and variables. The main objective is to improve operational efficiency and facilitate regular examination of exams for both students and faculties.

KEYWORDS-Constraint Optimization Faculty Availability, Student Workload Management, Invigilation Scheduling Educational Timetabling, Fair Examination Allocation.

I. INTRODUCTION

The handle of producing exam plans is a significant perspective of scholarly administration that straightforwardly impacts understudies, staff and instructive teach. With the expanding number of understudies and the expanding complexity of course offering, the requirement for proficient and reasonable exam planning has never been more prominent. In today's instructive scene, it is basic to adjust scholastic thoroughness with understudy well-being, making reasonable and down to earth plan era a foundation of regulation operations. This complicated prepare includes decision-making that takes into account different limitations, counting asset accessibility, understudy course determinations, and organization approaches, all of which contribute to the by and large proficiency and reasonableness of the exam process. Several thinks about have investigated strategies to address the complexity of

exam planning. Procedures such as hereditary calculations, as illustrated by Abdulaziz Aminu et al. (2019), optimize programs by taking into account components such as supervisory, assignments and weariness imperatives. Numbers programming approaches (Cataldo et al., 2017) and multi-objective optimization systems (Abdallah, 2016) assist emphasize the significance of efficiently tending to planning clashes and asset utilization. Heuristic strategies, such as the "great flood" calculation (Kahar and Kendall, 2015) and curriculum-based planning arrangements (Muller and Rudova, 2012) highlight the versatility of algorithmic techniques in tending to real-world challenges. This extends builds on these techniques to plan a programmed exam plan era framework that coordinating key variables such as course enrolment, elective data, workforce accessibility, and supervisory duties. Extra contemplations, such as constraining the everyday exam term to maintain a strategic distance from burnout and keeping up an even-handed dispersion of workload, upgrade the convenience of the proposed arrangement. By methodically characterizing imperatives, scopes and factors, the framework points to create optimized plans that adjust with organization plans that adjust with organizations approaches and back the well-being of all stakeholders. By tending to challenges and leveraging mechanical progresses, this extends points to make a vigorous system for producing exam plans. The result not as it were rearranging authoritative assignments, but moreover advances a learning-friendly environment by guaranteeing value, proficiency and operational brilliance in scholastic planning.

II. LITERATURE SURVEY

Creating exam timetables is a crucial job for schools and universities. It involves managing several factors like when teachers are available, how many students can fit in each room, and what students prefer. This review looks at how methods and technologies for making these timetables have changed over time. It covers old and new methods, ways to handle constraints, optimization techniques, software tools, adjustments for changes, scalability, and real-life examples. The goal is to give clear picture of current research and suggest areas where further study could be helpful.

III. METHODOLOGY

The advancement of the Programmed Examination Timetable Era Framework is organized into five key steps, guaranteeing a precise approach to address the complexities of planning whereas following to organization constraints.

Step 1: Information Collection and Storage.

The starting stage includes gathering and organizing input information basic for timetable generation.

Input Data: Course enrolment is collected from the university's records. Faculty accessibility for invigilation obligations and room capacities are too documented.

Database Setup: a MySQL database

pattern is outlined to store point by point data, counting courses, understudies, exam lengths, rooms, and workforce schedules. Relationships are built up between understudy enrolments and courses to dodge planning conflicts.

Step 2: Characterize Constraints

To guarantee the produced timetable meets organization necessities, imperatives are isolated into two categories:

Hard Constraints: An understudy cannot have two exams planned at the same time. Faculty individuals cannot invigilate numerous exams simultaneously. Room capacities must not be exceeded. Exams must follow to room accessibility hours.

Soft Constraint: Minimize the number of exams per day for understudies and staff to diminish exhaustion. Distribute exams equally over the examination period to dodge burnout.

Step 3: Backend Development

The backend improvement includes actualizing the centre rationale for timetable generation.

Technologies: Java and Spring Boot are utilized to oversee planning calculations and encourage consistent communication between the frontend, backend and database.

Scheduling Algorithm: The calculation forms information from the MySQL database to create substantial, conflict-free exam slots. Optimization methods guarantee adherence to difficult and

delicate imperatives whereas apportioning slots.

Step 4: Frontend Development

A user-friendly web interface is created utilizing HTML and CSS. The interface shows timetables in a clear and organized for improved usability.

Step 5: Timetable Generation

The last step coordinating all components for consistent timetable generation.

IV. RESULTS

The results section presents the outcomes of the examination timetable generation system after its implementation. The timetable was generated based on the input data, including course registration, faculty schedules, and room capacities, while adhering to both hard and soft constraints. The system successfully produced a timetable that avoided conflicts such as overlapping exams for students, faculty scheduling issues and room capacity violations. The output was presented in a clear, structured format, which administrators could easily review.

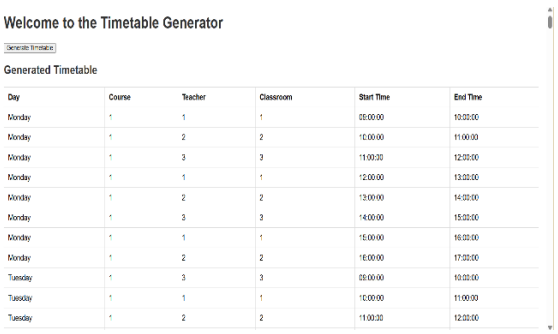


Fig 1. Output

V. DISCUSSIONS

The produced examination timetable was assessed against the characterized

imperatives and the by and large prerequisites of the institution. The comes about were deciphered to survey the viability of the planning calculation in dealing with the complexities of real-world timetabling. The framework effectively produced timetables with negligible clashes and optimized room utilization. The conveyance of exams over a few days made a difference decrease understudy and staff depletion, affirming the system's adherence to both difficult and delicate constraints. Compared to earlier inquire about, such as the work of Cataldo et al. (2017) and Aminu et al. (2019), which utilized optimization calculations to address comparative planning issues, this framework moreover connected a data-driven approach for conflict-free timetables. The comes about adjust with build-up procedures but offer down to earth enhancements in guaranteeing more adjusted workloads for understudies and faculty. The discoveries recommend that computerized planning frameworks can altogether improve the proficiency of examination timetable era. To utilize of an organized approach to characterize and coordinated limitations guarantees decency and common sense, which can lead to moved forward fulfilment among understudies, staff and administrators. A key quality of this framework is its capacity to handle complex limitations, guaranteeing conflict-free timetables. In any case, the framework may confront challenges in teach with exceedingly energetic course offerings or last-minute changes in understudy enlistments. Additionally, the dependence on MySQL databases may restrain versatility for amazingly expansive institutions. Future inquire about might investigate the joining of machine learning calculations to powerfully adjust to changing planning needs or to optimize for extra variables such as understudy inclinations. Extending the framework to incorporate more different limitations, such as availability contemplations for crippled understudies, seem to encourage upgrade its value.

VI. CONCLUSION

In conclusion, the improvement and execution of the Programmed Examination Timetable Era Framework has illustrated its capacity to create conflict-free timetables that regard both difficult and delicate imperatives. The discoveries appear that computerized framework can move forward operational proficiency and reasonableness in examination planning, profiting both understudies and staff. The think about highlights the significance of leveraging computational methods in tending to complex authoritative assignments, eventually contributing to a more conductive learning environment. Future advancements to the framework, such as consolidating machine learning and taking care of last-minute changes, will proceed to upgrade its commonsense pertinence and adequacy.

VII. REFERENCES

[1] Lindahl, M., Sørensen, M., & Stidsen, T. R. (2018). A fix-and-optimize matheuristic for university timetabling. Journal of Heuristics, 24(4),645–665. (https://doi.org/10.1007/s10732-018-9371-3)

[2] Cataldo, A., Ferrer, J.-C., Miranda, J., Rey, P. A., & Saure, A. (2017).An integer programming approach to curriculum-based examination timetabling. Annals of Operations Research, 258(2), 369–393. (https://doi.org/10.1007/s10479-016-2321-2)

[3] Abdallah, K. S. (2016). Multi-objective optimization for exam scheduling to enhance the educational service performance. The Journal of Management and Engineering Integration, 9(1), 14–23.

[4] Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society, 47(3), 373–383. (https://doi.org/10.1057/jors.1996.37)

[5]https://programmer2programmer.net/live_projects/Synopsis/download_project_synopsis.aspx?JAVA-Time-Table-Generation-System-JAVA-MySQL&id=145

[6] https://stackoverflow.com/questions/12209268/scheduling-timetable-algorithm

[7] R. Firke, P. Bhabad, O. Gangarde, A. Magar, and A. Tawlare, "Automatic Timetable Generation System," *IJCRT*, vol. 2023, pp. 1-5, 2023.

[8] Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. Journal of the Operational Research Society, 47(3), 373–383. (https://doi.org/10.1057/jors.1996.37)

[9] Kendall, G., & Hussin, N. M. (2004).A tabu search hyper-heuristic approach to the examination timetabling problem. Informs Journal on Computing, 16(3), 270-276. (https://doi.org/10.1287/ijoc.1030.0017)

[10] Murray, K., & Wilson, C. (2014). Solving the university examination timetabling problem with a multi-stage algorithm. Computers & Operations Research, 56, 81-90. (https://doi.org/10.1016/j.cor.2014.01.005)

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Rakheeba Taseen

In recognition of the publication of the paper entitled

Examination Time Table Generation

Published In IJRAR (www.ijrar.org) UGC Approved - Journal No : 43602 & 7.17 Impact Factor

Volume 11 Issue 4 December 2024, Date of Publication: 19-December-2024

PAPER ID : IJRAR24D3051

Registration ID : 303325

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Trishanya

In recognition of the publication of the paper entitled

Examination Time Table Generation

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 11 Issue 4 December 2024, Date of Publication: 19-December-2024

PAPER ID : IJRAR24D3051

Registration ID : 303325

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Bhavith Mourya R

In recognition of the publication of the paper entitled

Examination Time Table Generation

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 11 Issue 4 December 2024, Date of Publication: 19-December-2024

PAPER ID : IJRAR24D3051

Registration ID : 303325

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Ajay R

In recognition of the publication of the paper entitled

Examination Time Table Generation

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 11 Issue 4 December 2024, Date of Publication: 19-December-2024

PAPER ID : IJRAR24D3051

Registration ID : 303325

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

Certificate of Publication

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

E-ISSN: 2348-1269, P-ISSN: 2349-5138

The Board of

International Journal of Research and Analytical Reviews (IJRAR)

Is hereby awarding this certificate to

Dinson Sam Thomas

In recognition of the publication of the paper entitled

Examination Time Table Generation

Published In IJRAR (www.ijrar.org) UGC Approved (Journal No : 43602) & 7.17 Impact Factor

Volume 11 Issue 4 December 2024, Date of Publication: 19-December-2024

PAPER ID : IJRAR24D3051

Registration ID : 303325

R.B.Joshi

EDITOR IN CHIEF

UGC and ISSN Approved - Scholarly open access journals, Peer-reviewed, and Refereed Journals, Impact factor 7.17 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly Journal

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS | IJRAR

An International Scholarly, Open Access, Multi-disciplinary, Indexed Journal

Website: www.ijrar.org | Email: editor@ijrar.org | ESTD: 2014

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org

IJRAR | E-ISSN: 2348-1269, P-ISSN: 2349-5138

SDG Mappings with the project

- SDG 4: Quality Education

  The system helps improve education by making the examination process smooth and error-free. It ensures students don't have too many exams on the same day and balances faculty workloads, creating a better environment for learning.

- SDG 9: Industry, Innovation, and Infrastructure

  Using modern technologies like Spring Boot and Java, the project shows how innovation can solve problems. It improves the university's infrastructure by introducing a smart way to manage timetables.

- SDG 10: Reduced Inequalities

  The system ensures fairness in scheduling exams and invigilation duties, so no one is overburdened. It prevents problems like overlapping exams that could affect certain students more than others.

- SDG 11: Sustainable Cities and Communities

  It uses resources like classrooms and faculty time efficiently, reducing waste.

  By making better use of what's already available, it promotes sustainable practices within the university.

- SDG 16: Peace, Justice, and Strong Institutions

  Automating the timetable process makes it reliable and fair, reducing mistakes and biases. It helps build trust in the university's administrative processes.


SUSTAINABLE DEVELOPMENT GOALS
17 GOALS TO TRANSFORM OUR WORLD
1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS