SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune

AN INTERNSHIP REPORT ON

Timetable Generation System

SUBMITTED BY
Name: Prathmesh Pawar
Class: Third Year Computer Engineering
Roll no: 31359

Under the guidance of
Internship Guide: Prof. Rutuja Kulkarni

DEPARTMENT OF COMPUTER ENGINEERING
ACADEMIC YEAR 2023-24

DEPARTMENT OF COMPUTER ENGINEERING

SCTR's Pune Institute of Computer Technology
Dhankawadi, Pune
Maharashtra 411043

CERTIFICATE

This is to certify that the SPPU Curriculum-based internship report entitled
"Timetable Generation System"

Submitted by
Prathmesh Pawar
Roll No. 31359

has satisfactorily completed the curriculum-based internship under the guidance of *Prof. Rutuja Kulkarni* towards the fulfillment of third year Computer Engineering Semester VI, Academic Year 2023-24 of Savitribai Phule Pune University.

Prof. Rutuja Kulkarni
Internship Guide
PICT, Pune

Dr. G. V. Kale
Head
Department of Computer Engineering
PICT, Pune

Place: PICT Pune
Date: April 18, 2024

Acknowledgement

It gives me great pleasure in presenting the internship report on "Timetable Generation System".

First of all, I would like to express my heartfelt gratitude to my internship guide Prof. Rutuja Kulkarni for providing me with invaluable guidance, support, and encouragement throughout the duration of this internship. Their expertise and insights have been instrumental in shaping this project.

I would like to extend my sincere thanks to the Head of the Computer Engineering Department, Dr. G.V.Kale, for her continuous support and encouragement.

I am also thankful to the Department Internship Coordinator, Prof. P.P.Joshi, for her assistance and guidance at various stages of the internship.

Finally, I would like to thank my peers and colleagues for their support and encouragement.

Contents

List of Figures

List of Tables

1 Title: Timetable Generation System

1.1 Introduction

In today's fast-paced educational environment, efficient management of resources is crucial for the smooth functioning of educational institutions. One of the key aspects of resource management in educational institutions is the scheduling of classes, teachers, and resources to ensure optimal utilization and maximum productivity. Traditional methods of timetable generation, which rely on manual intervention and paper-based processes, often prove to be inefficient and error-prone.

To address these challenges, the Timetable Generation System offers an automated solution that streamlines the process of timetable creation. By leveraging the power of technology, the system aims to simplify the complex task of scheduling by automating the allocation of classes, teachers, and resources based on predefined constraints and requirements.

The Timetable Generation System is designed to cater to the unique needs of educational institutions, allowing administrators to define various parameters such as class timings, teacher availability, classroom capacities, and subject requirements. Using this input, the system employs advanced scheduling algorithms to generate optimized timetables that minimize conflicts and maximize efficiency.

By automating the timetable generation process, the system not only saves time and effort but also reduces the likelihood of errors and inconsistencies. Additionally, it provides administrators with greater flexibility and control over the scheduling process, allowing them to quickly adapt to changes and disruptions.

Overall, the Timetable Generation System represents a significant step forward in the quest for efficient resource management in educational institutions. By harnessing the power of technology, it empowers administrators to create timetables that meet the diverse needs of students, teachers, and staff, ultimately enhancing the overall educational experience.

1.2 Problem Statement

The manual process of generating timetables is time-consuming and prone to errors. It often leads to conflicts and inefficiencies in scheduling classes, teachers, and resources. The Timetable Generation System addresses these challenges by providing an automated solution that optimizes the allocation of resources and minimizes conflicts.

1.3 Objectives and Scope

1.3.1 Objectives

The main objectives of the Timetable Generation System are as follows:

- Automate the process of generating timetables for educational institutions.
- Optimize the allocation of classes, teachers, and resources.
- Minimize conflicts and inefficiencies in scheduling.
- Provide a user-friendly interface for administrators to define constraints and requirements.

1.3.2 Scope

The scope of the project includes the development of a web-based application that allows administrators to input constraints and requirements, such as class timings, teacher availability, and classroom capacities. The system then generates optimized timetables based on these inputs.

1.4 Methodological Details

The timetable generation system follows a systematic approach to efficiently allocate subjects, teachers, classrooms, and time slots while adhering to various constraints and objectives. Here are the key methodological details outlining the process:

1. Problem Understanding and Requirement Analysis:
    - The initial step involved a comprehensive analysis of the problem domain, understanding the requirements, constraints, and objectives of the timetable generation system.
    - This phase included gathering requirements from stakeholders such as academic administrators, faculty members, and students to ensure the system meets their needs.
2. Data Collection and Representation:
    - Essential data such as subject details, teacher availability, classroom information, time slot configurations, and division-specific constraints were collected and organized.
    - Data representation involved the use of data structures such as arrays, vectors, and maps to efficiently manage and manipulate the information within the system.
3. Algorithm Design:
    - The core algorithm for timetable generation was designed to intelligently allocate subjects, teachers, classrooms, and time slots while considering various constraints.
    - Techniques such as randomized selection, constraint satisfaction, and optimization were employed to ensure the generation of feasible and efficient timetables.
4. Constraint Handling:
    - The system accommodated multiple constraints such as teacher availability, subject-teacher associations, classroom capacities, and division-specific requirements.
    - Constraint handling mechanisms were integrated into the algorithm to ensure that timetables satisfy all constraints while maintaining balance and fairness.
5. Fitness Evaluation:
    - A fitness evaluation function was developed to quantitatively measure the quality of generated timetables based on predefined criteria.
    - Metrics such as hard fitness (e.g., teacher workload balance, subject-hour distribution) and soft fitness (e.g., student preferences, room utilization) were considered to assess the effectiveness of timetables.
6. Optimization and Iterative Improvement:
    - The system employed optimization techniques to iteratively improve generated timetables based on fitness evaluation results.
    - Iterative processes such as genetic algorithms, simulated annealing, or local search were utilized to refine timetables and explore better solutions within the solution space.
7. Testing and Validation:
    - Extensive testing and validation procedures were conducted to ensure the correctness, efficiency, and robustness of the timetable generation system.
    - Test cases covering various scenarios, edge cases, and constraint violations were executed to verify the system's functionality and reliability.

8. User Interface Design (Optional):
    - A user-friendly interface may be designed to allow stakeholders to interact with the system, input preferences, view generated timetables, and provide feedback.
    - The interface design focused on simplicity, intuitiveness, and accessibility to facilitate effective communication and usage.
9. Documentation and Reporting:
    - Detailed documentation was prepared to describe the system architecture, algorithmic approach, data structures, constraints, and functionalities.
    - Project reports, user manuals, and technical documentation were generated to provide comprehensive insights into the development process and system operation.
10. Deployment and Maintenance:
    - Upon completion, the timetable generation system was deployed in the educational institution's infrastructure for practical usage.
    - Regular maintenance, updates, and support services were provided to ensure the system's continued functionality, compatibility, and adaptability to evolving requirements.

1.4.1 Section: Implementation of Genetic Algorithm for Timetable Generation[1]

1. Introduction to Genetic Algorithms

- Provide an overview of genetic algorithms (GAs) and their application in optimization problems.
- Highlight the relevance of GAs in solving complex scheduling problems, such as timetable generation.

2. Representation of Timetable

- Describe how the timetable is represented within the context of the genetic algorithm.
- Define the chromosome structure, which encapsulates a potential solution to the timetable generation problem.
- Explain how each gene in the chromosome corresponds to specific aspects of the timetable, such as class schedules and room assignments.

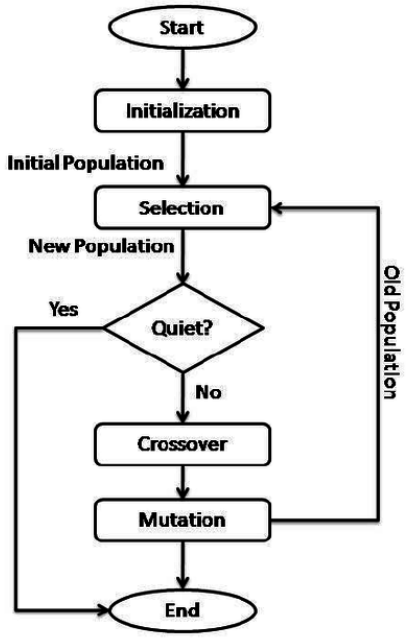


Figure 1: Flowchart illustrating the genetic algorithm implementation for timetable generation[5].

3. Fitness Function

- Define the fitness function used to evaluate the quality of a timetable solution.
- Discuss the considerations involved in designing the fitness function, including constraints and objectives such as minimizing conflicts and maximizing resource utilization.

4. Initialization

- Explain the process of generating the initial population of timetable solutions.
- Discuss whether random assignment or domain-specific knowledge is utilized to create the initial solutions.

Table 1: Constraints for Timetable Generation

Constraint	**Description**
Teacher Availability	Teachers' available time slots
Classroom Capacity	Maximum number of students a classroom can accommodate
Subject Prerequisites	Prerequisites for certain subjects
Class Durations	Duration of each class
Break Time	Time slots allocated for breaks
Student Preferences	Preferences for specific time slots
Special Events	Events that affect scheduling (e.g., exams, holidays)

5. Genetic Operators

- Selection: Describe the mechanism used to select parent solutions for reproduction.
- Crossover: Explain how genetic material from selected parent solutions is combined to create offspring solutions.
- Mutation: Discuss the operator responsible for introducing random changes to offspring solutions to maintain genetic diversity.

6. Genetic Algorithm Parameters

- Define the parameters of the genetic algorithm, such as population size, crossover rate, and mutation rate.
- Explain the impact of these parameters on the algorithm's performance and convergence.

Table 2: Genetic Algorithm Parameters

Parameter	**Description**
Population Size	Number of individuals in each generation
Generations	Number of iterations or generations
Selection Method	Method for selecting parents for crossover
Crossover Rate	Probability of crossover for two parents
Mutation Rate	Probability of mutation for each gene
Elitism	Whether to keep the best individual from each generation
Fitness Function	Evaluation function to determine the fitness of individuals

7. Iterative Optimization

- Describe how the genetic algorithm iteratively improves the population of timetable solutions over multiple generations.
- Explain the sequence of operations involved in each generation, including selection, crossover, mutation, and evaluation.

8. Stopping Criteria

- Discuss the criteria used to determine when the genetic algorithm should terminate.
- Consider factors such as reaching a maximum number of generations or achieving a satisfactory fitness threshold.

9. Implementation Details

- Provide specific details regarding the implementation of the genetic algorithm.
- Include information about the programming language, libraries used, data structures, and algorithmic optimizations.

10. Example or Case Study

- Present an example or case study illustrating the application of the genetic algorithm to timetable generation.
- Include visualizations or diagrams to demonstrate the evolution of the algorithm and the resulting optimized timetable solution.

1.4.2 MERN Stack Overview

The MERN stack is a popular technology stack used for building full-stack web applications. It consists of four main components:

- **MongoDB:** MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It is a popular choice for web applications due to its scalability and ease of use.
- **Express.js:** Express.js is a minimalist web application framework for Node.js. It provides a robust set of features for building web and mobile applications. Express.js simplifies the process of handling HTTP requests, routing, and middleware integration.
- **React.js:** React.js is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and efficiently update the UI in response to changes in data. React.js follows a component-based architecture, making it easy to build complex UIs.
- **Node.js:** Node.js is a server-side JavaScript runtime environment. It allows developers to build scalable and high-performance web applications using JavaScript. Node.js is known for its non-blocking, event-driven architecture, which makes it ideal for building real-time applications.

The MERN stack provides a cohesive development experience, allowing developers to use JavaScript across the entire stack. This enables rapid development and seamless integration between the frontend and backend components of the application.[2]

1.5 Modern Engineering Tools Used

The development of the timetable generation system leveraged a range of modern engineering tools and technologies to streamline the development process and enhance productivity. Key tools and technologies utilized in the project include:

- **MERN Stack**: The project was developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack, which provided a comprehensive and modern web development environment for building scalable and efficient web applications.[1]
- **Visual Studio Code (VS Code) IDE**: Visual Studio Code, a powerful and lightweight integrated development environment, was used for coding, debugging, and version control. Its extensive plugin ecosystem facilitated seamless integration with various tools and enhanced development workflows.
- **GitHub**: GitHub served as the primary version control and collaboration platform for the project. Git, combined with GitHub's features such as pull requests, branching strategies, and issue tracking, enabled effective team collaboration and code management.
- **Netlify Hosting Services**: The project was hosted on Netlify, a modern web hosting platform that provides continuous deployment, global content delivery network (CDN), and HTTPS support. Netlify's integration with Git allowed for automated deployment from GitHub repositories, simplifying the deployment process.[3]
- **Postman**: Postman, a popular API testing tool, was utilized for testing and debugging the backend API endpoints. Its intuitive interface and extensive features facilitated API development, testing, and documentation.[4]
- **Browser Inspection Tools**: Browser inspection tools such as Chrome Developer Tools and Firefox Developer Tools were instrumental in debugging frontend issues, analyzing network requests, and optimizing frontend performance. These tools provided insights into client-side behavior and aided in frontend development and troubleshooting.

These modern engineering tools and technologies played a crucial role in the successful development, deployment, and maintenance of the timetable generation system, contributing to its efficiency, reliability, and scalability. .

1.6 Outcome/Results of Internship Work

The Timetable Generation System has been successfully developed and tested in a simulated environment. The system demonstrates significant improvements in the efficiency and accuracy of timetable generation compared to manual methods.

1.6.1 Home

This link will redirect the user back to the TimeTable Generator main page. The page will look like shown below:

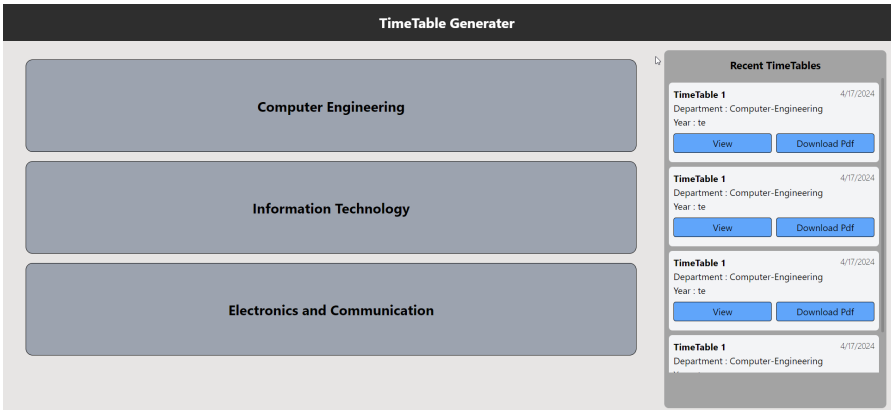


Figure 2: Home Screen UI.

1.6.2 Data Acquisition

In this section user will be able to provide all the necessary inputs such as subject, teachers, labs, audit courses, Elective courses related information. Left side Menus can be used for navigating between them.

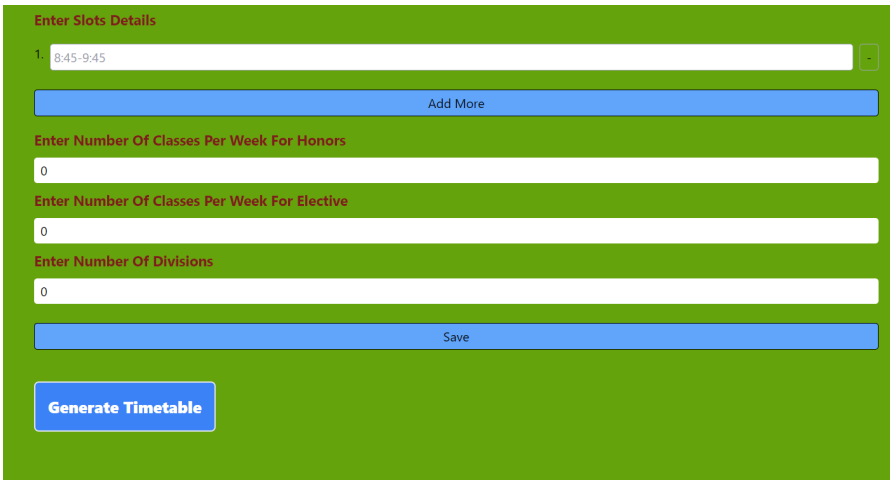
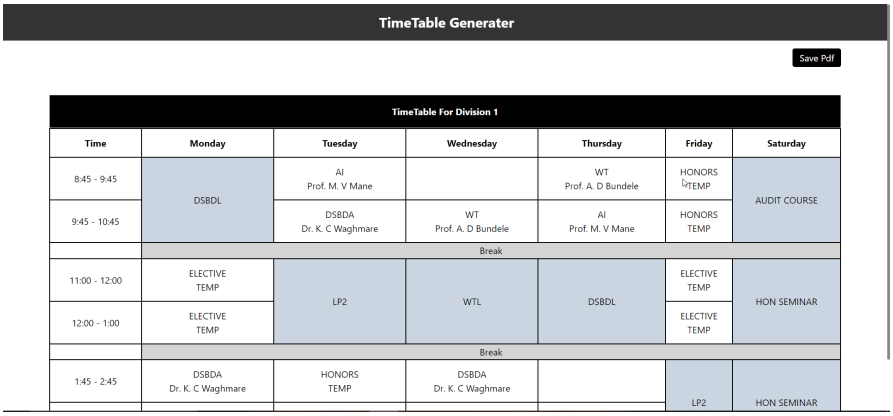


Figure 3: Screenshot of User Input UI.

1.6.3 Display Of Generated Timetable

Generated timetable is visible in browser. User can download it in pdf format as well.



TimeTable Generater

Save Pdf

TimeTable For Division 1

Time	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
8:45 - 9:45	DSBDL	AI Prof. M. V Mane		WT Prof. A. D Bundele	HONORS TEMP	AUDIT COURSE
9:45 - 10:45		DSBDA Dr. K. C Waghmare	WT Prof. A. D Bundele	AI Prof. M. V Mane	HONORS TEMP	
	Break					
11:00 - 12:00	ELECTIVE TEMP	LP2	WTL	DSBDL	ELECTIVE TEMP	HON SEMINAR
12:00 - 1:00	ELECTIVE TEMP				ELECTIVE TEMP	
	Break					
1:45 - 2:45	DSBDA Dr. K. C Waghmare	HONORS TEMP	DSBDA Dr. K. C Waghmare		LP2	HON SEMINAR

Figure 4: Screenshot of final Output.

1.7 Conclusion

In conclusion, the Timetable Generation System stands as a pivotal tool for educational institutions, streamlining the scheduling process and offering invaluable benefits. By automating the generation of timetables, it reduces administrative burden, saves time, and ensures optimal utilization of resources such as classrooms, faculty, and time slots. This efficiency translates to smoother operations, improved student-teacher interactions, and enhanced overall productivity within the institute. With its user-friendly interface and adaptable architecture, the system caters to the unique scheduling needs of each institution, making it a versatile asset for academic management. By adopting this system, institutes can expect not only to alleviate scheduling challenges but also to foster a conducive learning environment that promotes academic success and organizational effectiveness.

1.8 References

1. https://medium.com/@Nitin_Indian/genetic-algorithms-215a1e5abf87
2. https://www.freecodecamp.org/news/mern-stack-roadmap-what-you-need-to-know-to-build-full-stack-apps/
3. https://www.netlify.com/blog/2016/09/29/a-step-by-step-guide-deploying-on-netlify/
4. https://medium.com/sliit-foss/postman-api-101-3bb12dd553f4
5. https://www.researchgate.net/figure/Flow-chart-of-genetic-algorithm$_f ig2_2$61028418