A

Project Report

on

Smart Timetable Generator

Submitted to

Sant Gadge Baba Amravati University, Amravati

Submitted in partial fulfilment of

the requirements for the Degree of

Bachelor of Engineering in

Computer Science and Engineering

Submitted by

Disha Wasnik
(PRN: 203120081)

Harish Barhate
(PRN: 203120154)

Sakshi Bhombe
(PRN: 203120159)

Sanika Sapkale
(PRN: 203120165)

Under the Guidance of

Dr. P. K. Bharne

Asst. Prof., Computer Science & Engineering Dept.

Department of Computer Science and Engineering

Shri Sant Gajanan Maharaj College of Engineering,
Shegaon – 444 203 (M.S.)

Session 2023-2024

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,

SHEGAON – 444 203 (M.S.)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that **Ms. Disha Wasnik, Mr. Harish Barhate, Ms. Sakshi Bhombe and Ms. Sanika Sapkale** students of final year Bachelor of Engineering in the academic year 2023-24 of Computer Science and Engineering Department of this institute have completed the project work entitled "**Smart Timetable Generator**" and submitted a satisfactory work in this report. Hence recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.

[illegible] 6/5/24.

Dr. P. K. Bharne
Project Guide

[illegible]

Dr. J. M. Patil
Head of Department

[illegible]

Dr. S. B. Somani
Principal
SSGMCE, Shegaon

SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,

SHEGAON – 444 203 (M.S.)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that **Ms. Disha Wasnik, Mr. Harish Barhate, Ms. Sakshi Bhombe and Ms. Sanika Sapkale** students of final year Bachelor of Engineering in the academic year 2023-24 of Computer Science and Engineering Department of this institute have completed the project work entitled "**Smart Timetable Generator**" and submitted a satisfactory work in this report. Hence recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.

[illegible]	[illegible]
Internal Examiner	**External Examiner**
Dr. P.K. Bharne	Prof. A.M. Bajad.
Name and Signature	**Name and Signature**
Date: 10/5/24	**Date:** 10/5/24

ACKNOWLEDGEMENT

It is our utmost duty and desire to express gratitude to various people who have rendered valuable guidance during our project work. We would have never succeeded in completing our task without the cooperation, encouragement and help provided to us by then. There are a number of people who deserve recognition for their unwavering support and guidance throughout this report.

We are highly indebted to our guide **Dr. P. K. Bharne** for his guidance and constant supervision as well as for providing necessary information from time to time. We would like to take this opportunity to express our sincere thanks, for his esteemed guidance and encouragement. His suggestions broaden our vision and guided us to succeed in this work.

We are sincerely thankful to **Dr. J. M. Patil** (HOD, CSE Department, SSGMCE, Shegaon), and to **Dr. S. B. Somani** (Principal, SSGMCE, Shegaon) who always has been kind to extend their support and help whenever needed.

We would like to thank all teaching and non-teaching staff of the department for their cooperation and help. Our deepest thank to our parents and friends who have consistently assisted us towards successful completion of our work.

–Project Members
Disha A. Wasnik
Harish B. Barhate
Sakshi N. Bhombe
Sanika S. Sapkale

ABSTRACT

The Smart Timetable Generator project introduces an innovative solution for creating efficient timetables, applicable in various contexts for example colleges. Generating time table manually occurs conflicts. This automatic timetable generation software generates the timetables automatically by taking the inputs like the number of subjects, teachers, the workload of a teacher, semester, and priority of subjects. Leveraging the versatile Flutter framework for cross-platform app development, we've designed an easy-to-use interface that empowers users to input their scheduling requirements effortlessly.

The real magic happens behind the scenes, where we employ genetic algorithms to generate optimized schedules that maximize resource utilization while minimizing conflicts. These genetic algorithms work like problem-solving wizards, iteratively refining schedules until they meet all constraints and preferences. What sets this solution apart is its user-friendliness; anyone can interact with the app, visualize the generated timetables, and make real-time adjustments. Moreover, the app ensures synchronization across multiple devices, facilitating seamless collaboration and decision-making. Ultimately, this project streamlines the often-troublesome task of timetable creation, resulting in improved resource allocation and reduced manual effort, all thanks to the power of genetic algorithms and the simplicity of the Flutter framework.

In conclusion, the Smart Timetable Generator, with its combination of Flutter's user-friendly framework and the intelligence of genetic algorithms, stands as a comprehensive solution for timetable creation. It not only simplifies the often-troublesome task but also adapts, secures, and scales to meet the evolving needs of educational and organizational scheduling.

Keywords: Genetic Algorithm, Resource Scheduling, Timetable Automation, Cross Platform Based Timetable Generation, Flutter Framework.

Contents

List of Figures

List of Tables

CHAPTER 01

INTRODUCTION

1. INTRODUCTION

Introducing "Smart Time Table Generator" – an innovative scheduling solution precisely crafted with Flutter and Dart, ensuring seamless functionality across various platforms. At the core of its expertise are advanced genetic algorithms, serving as intelligent scheduling assistants to streamline the complicated process of timetable creation. With "Smart Time Table Generator" articulate your scheduling requisites, specifying class times, meetings, and preferences, and witness the app's algorithmic precision in action. It observes to predefined rules and user preferences, generating optimal schedules that align seamlessly with your requirements.

Not merely a feat of technological complexity, this project claims a user-friendly interface, providing users with a clear visualization of generated schedules and the flexibility to make real-time modifications. The synchronization feature ensures that updates seamlessly propagate across all linked devices, guaranteeing a harmonized scheduling experience.

Our commitment with this project is explicit – to provide a technologically advanced, efficient, and user-centric solution that simplifies the complex task of scheduling. "Smart Time Table Generator" is composed to revolutionize your scheduling experience, offering a refined, stress-free approach to time management and organization.

1.1 Background:

Educational institutions, ranging from schools to universities, face a common challenge: creating efficient and conflict-free timetables. Manual timetable generation is a complex task that demands significant time and effort from administrators, often leading to suboptimal schedules, clashes, and inefficiencies.

In response to this challenge, the Smart Timetable Generator project is conceived. The aim is to leverage technology to automate the timetable creation process, ensuring the production of well-organized, time-efficient schedules while minimizing conflicts and meeting the diverse requirements of students, teachers, and staff.

Manual timetable creation is prone to human errors, and it becomes increasingly challenging as the size and complexity of educational institutions grow. The Smart Timetable Generator addresses these issues by employing advanced algorithms and user-friendly interfaces, allowing educational institutions to streamline the scheduling process, save time, and allocate resources more effectively.

The project is inspired by the need for a scalable and adaptable solution that caters to the specific requirements of different educational levels and institutions. By automating the timetable generation, the Smart Timetable Generator intends to not only enhance the overall efficiency of educational institutions but also contribute to a better learning environment by minimizing disruptions and optimizing resource utilization.

As educational institutions increasingly embrace digital transformation, the Smart Timetable Generator emerges as a strategic solution, aligning with the broader goal of improving educational processes through technology.

In summary, the Smart Timetable Generator project addresses a critical pain point in the educational sector, offering a sophisticated yet user-friendly solution to streamline and optimize the timetable creation process for institutions of all sizes. By automating this intricate task, the project aims to contribute to the overall improvement of educational experiences for both administrators and students.

1.2 Problem Statement:

The problem we're addressing is the inefficiency and issues with manual timetable creation in college. The current manual process of creating college timetables is time-consuming, error-prone, and inflexible, leading to scheduling conflicts.

1.3 Aim and Objectives

The aim of the Smart Timetable Generator is to automate the process of creating class schedules.

To fulfill the aim of the project, the following are the objectives:

1. To develop a cross-platform application using Flutter.
2. To utilize optimization algorithms such as Genetic Algorithm to create efficient class schedules.
3. To employ Firebase for data storage and retrieval to ensure seamless integration and scalability.
4. To implement Data Analytics techniques to analyse and optimize scheduling outcomes for improved efficiency.

1.4 Project Scope and Limitations:

Scope:

The scope of Smart Timetable Generator is mainly focused on resolving the issue of college lecture timetabling and therefore will not address scheduling of exams. It will be able to get assessed in various institutes in different departments and will also consider different criteria based on the form of classes, lecturers, rooms and laboratories affiliated with the relevant department. Efforts would be placed in motion to ensure that the method built is sufficiently generic to satisfy the different limitations that come with each individual educational institution. Also, the study incorporates continuous improvement through user feedback, exploring scalability, and assessing modularity for seamless integration with third-party systems.

Limitations:

1. Limited Applicability: The system is designed exclusively for web pages, restricting its use across other platforms or mediums.
2. Absence of Faculty Appointments: Faculty appointments are not visible or integrated into the timetable, potentially impacting scheduling transparency and organization.
3. Inability to Schedule Dual Lab Slots: The system is unable to display or manage laboratory slots that span two time slots on the timetable, leading to potential scheduling conflicts or inaccuracies.

1.5 Organization of Project:

The Smart Timetable Generator project is organized systematically, starting with an overview that outlines the project's motivation and significance. The literature review explores existing scheduling solutions, while the methodology details the use of genetic algorithms and Flutter for cross-platform app development. The technical architecture emphasizes the integration of algorithms and data security. User interface design highlights user-friendly features, followed by an implementation section detailing coding and algorithm integration. Testing and validation ensure reliability, with results and analysis assessing efficiency and user feedback. The discussion interprets findings and suggests future directions, leading to a conclusion. The project structure also includes recommendations, references, and appendices for supplementary materials.

The project is organized as follows:

Chapter 1: Introduction of the Project

This section serves as an introductory overview of the project, outlining its objectives and scope. The project aims to streamline timetable creation processes through the development of an intuitive application. Leveraging the Flutter framework for its versatility in cross-platform development, the application seeks to empower users to input their scheduling requirements effortlessly. Additionally, the utilization of advanced genetic algorithms underpins the generation of optimized schedules, prioritizing resource efficiency and conflict resolution.

Chapter 2: Literature Review

This chapter provides a comprehensive review of existing literature pertinent to timetable creation and optimization methodologies. Through an examination of prior research, methodologies, and technological frameworks, this section aims to identify key insights, trends, and gaps in the field. By drawing upon established knowledge and best practices, the literature survey informs the project's approach and methodology.

Chapter 3: Analysis of Project Requirements

This chapter entails a detailed analysis of the specific requirements and constraints inherent to the project. By elucidating the needs and expectations of stakeholders, particularly educational institutions, this section delineates the essential features and functionalities necessary for effective timetable management.

Chapter 4: Architecture and Design

In this chapter, the architectural framework and design principles of the application are delineated. The system architecture, comprising various components and their interactions, is expounded upon to provide a holistic understanding of the application's structure.

Chapter 5: Implementation

This section elucidates the implementation phase of the project, detailing the development process, methodologies, and technologies employed. Through a systematic approach to coding, testing, and integration, the application is realized according to the defined requirements and design specifications.

Chapter 6: Result and Discussion

This chapter presents the outcomes and evaluation of the project, showcasing the generated timetables and their optimization metrics. Through a comprehensive analysis of results against predefined criteria, such as resource utilization and conflict resolution, the efficacy of the application is assessed.

Chapter 7: Conclusion

This chapter summarizes the project's objectives, methodologies, and outcomes, while also suggesting future research directions.

CHAPTER 02
LITERATURE REVIEW

2. LITERATURE REVIEW

A literature review involves systematically reviewing existing research on a specific topic, providing a critical summary, identifying gaps, and offering insights for future studies. It serves to contextualize, evaluate, and synthesize the current state of knowledge in a particular field.

2.1 Existing System:

The Research paper [1] titled "Automated Timetable Generation Using Machine Learning Algorithms," authored by Shraddha Ambhore, Pooja Walke, Rohit Ghundgrudkar, Akshay Alone, Anushree Khedkar and published in the International Journal of Research in Engineering, Science and Management, Volume-3, Issue-3, March-2020, presents an automated system to address the manual and time-consuming process of creating timetables in educational institutions. By leveraging machine learning algorithms such as Evolutionary Algorithm, Tabu Search, Simulated Annealing, and Scatter search, the proposed system optimizes timetable generation while considering constraints like university scheme requirements and faculty workload. Inputs such as semester-wise subjects and faculty workload are used to generate feasible timetables for the working days of the week. The methodology involves creating an initial population, defining fitness functions, selection, crossover, and mutation. Implementation includes modules for registration, login, and timetable generation, aiming to save time and manpower while providing a faster and better generation of timetables for any number of courses and multiple semesters.

The research paper [2] titled "Smart TimeTable System Using AI and ML," authored by Rutuja Kavade, Sohail Qureshi, Nikita Veer, Vaishanavi Ugale, and Priyanka Agrawal, introduces a novel approach to timetable creation using machine learning algorithms, particularly a genetic algorithm. The methodology involves the optimization of resources through Django, creating a generic and efficient timetable system. The paper explores various solutions, including local search procedures, multiple context reasoning, genetic algorithms, and constraint programming. While local search procedures like taboo search and simulated annealing are common, the genetic algorithm leverages natural selection principles for optimal solution development. Multiple context reasoning ensures a nuanced understanding of specific situations within the overall system. However, the drawbacks include the reliance on context reasoning, leading to reduced accuracy and optimization compared to the more robust genetic algorithm. Despite these limitations, the paper contributes valuable insights into the integration of AI and ML for timetable optimization, emphasizing the trade-offs between different approaches in the pursuit of an efficient scheduling system.

The research paper [3] titled "Automatic Time Table Generation Using Genetic Algorithm," authored by Mrs. G. Maneesha and students T. Deepika, S. BhanuSri, N. RaviKumar, and P. SivaNagamani, published in the July 2021 issue of the JETIR Journal, addresses the problem of conflicts in timetable generation through the application of genetic and heuristic algorithms. The proposed solution employs a stochastic search algorithm, combining Simulated Annealing (SA) and Genetic Algorithm (GA), leveraging inherent randomness to refine designs gradually. While SA mimics annealing to enhance designs, GA operates on a population of designs, evolving solutions through selection, reproduction, and mutation. The optimization process is determined by convergence and efficiency indicators, signaling the conclusion of the algorithm's attempts to improve overall performance. However, drawbacks include slow convergence speed, time-consuming tuning problems for manually adjusting parameters like population size, a lack of guarantee on solution quality, and challenges in differentiating between better and worse solutions due to noisy objectives. Despite these limitations, the research contributes valuable insights into utilizing genetic and heuristic algorithms for automatic timetable generation, acknowledging trade-offs and areas for further improvement in optimization efficiency and solution quality.

The research paper [4] titled "Design and Implementation of a Web-Based Timetable System for Higher Education Institutions," authored by Henry Techie-Menson and Paul Nyagorme, published in the International Journal of Educational Research and Information Science in 2021, presents a comprehensive approach to the development of a web-based timetable system. The methodology involves the evaluation of the existing system using the Joint Application Design (JAD) approach to foster collaboration among stakeholders. The chosen system development life cycle is Rapid Application Development (RAD), known for its incremental model facilitating swift development aligned with project goals. Utilizing tools such as MYSQL, PHP, JQUERY, and HTML, the proposed system aims for quick operational program creation and seamless collaboration. However, drawbacks include dependency on internet connectivity, technical challenges, potential user resistance, data security concerns, maintenance overhead, limited offline access, costs, learning curve, customization challenges, and system downtime. Despite these limitations, the research contributes valuable insights into the design and implementation of web-based timetable systems for higher education institutions, acknowledging the trade-offs involved in adopting such technologies.

The research paper [5] titled "Automated Timetable Generation using Genetic Algorithm," authored by Shraddha Thakare, Tejal Nikam, and Prof. Mamta Patil, published in the International Journal of Engineering Research & Technology (IJERT) in July 2020, addresses the complex process of timetabling in educational institutes. The paper highlights timetabling as a non-polynomial complete problem and employs a heuristic approach to generate a set of good solutions. The primary objective is to optimize the algorithm used in recent timetabling systems, aiming to produce optimal timetables with fewer or no clashes. The outcomes emphasize the significance of automated timetabling in achieving reduced manpower, time savings, and improved data accuracy. However, drawbacks include potential issues with creativity, quality assurance, adaptability, biases, maintenance complexity, resistance to change, privacy concerns, and limitations in customization. Despite these challenges, the research contributes valuable insights into the application of genetic algorithms for automated timetable generation, acknowledging the trade-offs and considerations essential for the successful implementation of such systems in educational institutions.

The research paper [6] titled "Automatic Timetable Generation Using Genetic Algorithm," authored by Kehinde Williams and Micheal Ajinaja, published in IISTE on May 31st, 2019, outlines a comprehensive methodology for solving a heavily constrained university timetabling problem. The genetic algorithm employed utilizes a problem-specific chromosome representation and integrates heuristics and context-based reasoning to obtain feasible timetables within reasonable computing time. An intelligent adaptive mutation scheme is implemented to enhance convergence speed. However, drawbacks include the inherent limitation of genetic algorithms in providing a 100% optimal timetable and a slightly slower processing speed. Despite these limitations, the research contributes valuable insights into addressing complex scheduling problems in educational institutions through the application of evolutionary algorithms, acknowledging the trade-offs involved in terms of optimality and computational efficiency.

The research paper [7] titled "Automated Timetable Generator Using Machine Learning" by Prashanta Kumar, Shreedhar Sanakar, Praveen Kumar, Syed Muhammad Usman, Vani introduces a Python-based software for generating timetables automatically, aiming to alleviate the manual effort and time involved in scheduling classes in professional colleges. The system ensures that faculty timings do not overlap and effectively utilizes available time slots. The methodology involves artificial intelligence techniques, specifically optimization, with an iterative development approach for system refinement. The evaluation methodology combines Monte Carlo methods and surrogate modeling. The main problem addressed is the manual assignment of subjects to faculty, which is time-consuming and prone to errors. The proposed solution involves a decision tree algorithm and linear regression for timetable generation, allowing administrators to assign subjects, classrooms, and manage substitutions efficiently. The system architecture supports the seamless functioning of the software, providing a comprehensive timetable management solution for colleges. The results indicate that while the system provides valid solutions, some manual adjustments may still be required. However, the system's simplicity and flexibility enable users to experiment with different configurations to find optimal solutions, despite the challenges posed by the complexity of scheduling constraints.

The paper [8] presents the development and implementation of a lecture and examination timetable generator (TTGen) for the Department of Electrical and Electronics Engineering at the University of Agriculture, Makurdi, Nigeria. Authored by Joseph M. Mom and Jonathan A. Enokela, the software is built using Microsoft's Visual Basic.Net and MySQL database server. It effectively manages students, lecturers, venues, periods, and courses, featuring image handling for various objects and utilizing a randomizer module for timetable generation. The TTGen addresses the challenges of manual timetable creation and offers a cost-effective, flexible solution tailored to academic institutions. It employs iterative refinement to generate clash-free timetables efficiently and can produce visual representations and export options in HTML and Excel formats. The software's versatility extends to generating timetables for multiple departments and faculties, with potential applications beyond timetable management, such as student course registration.

Table 2.1 Literature Review

S. N.	**Title & Author of Research Paper**	**Publication Details**	**Methodology & Solution**	**Limitations**
1.	Automatic Timetable Generator Authors: Shraddha Ambhore Pooja Walke Rohit Ghundgrudkar Akshay Alone Anushree Khedkar	International Journal of Research in Engineering, Science and Management Volume-3, Issue-3, March-2020 www.ijresm.com | ISSN (Online): 2581-5792 381	Machine learning algorithms, including Algorithm, Tabu Search, Simulated Annealing, and Scatter search, are employed to optimize timetable generation by considering constraints like university scheme requirements and faculty workload, using semester-wise subjects and faculty workload as inputs.	Limitations may arise from scheduling complexities and the need for ongoing system refinement to ensure optimal solutions.
2.	Smart Time Table System Using AI and ML Authors: Prof. Priyanka Agrawal Rutuja Kavade Sohail Qureshi Nikita Veer Vaishanavi Ugale	2023 IJCRT (International Journal of Creative Research Thoughts) | Volume 11, Issue 5 May 2023 | ISSN: 2320-2882	The smart timetable system employs machine learning, specifically a genetic algorithm, to address complex problems by replacing suboptimal solutions. Utilizing Django, the system focuses on resource optimization through ML and provides a customizable timetable view using a SQL database.	It has less accuracy and optimization compared to the genetic algorithm.

3.	Automatic Time Table Generation Using Genetic Algorithm Authors: Mrs. G. Maneesha T. Deepika S. BhanuSri N. RaviKumar P. SivaNagamani	July 2021[JETIR Journal – Journal of Emerging Technologies and Innovative Research	The methodology employs a hybrid algorithm, integrating Simulated Annealing and Genetic Algorithm, leveraging inherent randomness to refine designs and optimize overall performance. Convergence and efficiency indicators govern the optimization endpoint	Slow convergence speed, time-consuming manual tuning of parameters like population and lack of guarantee on solution quality
4.	Design and Implementation of a Web-Based Timetable System for Higher Education Institutions Authors: Henry Techie-Menson Paul Nyagorme	March 4, 2021-IJERIS (International Journal of Educational Research and Information Science). Vol. 7, No. 1, 2021, pp. 1-13.	Utilizing JAD for system evaluation, the project follows RAD for swift development, emphasizing quick program creation and seamless collaboration. The system employs MYSQL, PHP, JQUERY, and HTML for backend, programming, special effects, and complementary language.	Challenges include internet dependency, potential user resistance, data security, maintenance overhead, limited offline access, costs, learning curve, and system downtime.
5.	Automated Timetable Generation using Genetic Algorithm Authors: Shraddha Thakare Tejal Nikam Prof. Mamta Patil	International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 ISSN: 2278-0181 Vol. 9 Issue 07, July-2020	Timetabling in educational institutes involves scheduling and assigning lecturers to time slots and allocating resources without time clashes, addressing a non-polynomial complete problem. A heuristic approach yields a set of good solutions, aiming to optimize	Potential issues with creativity, quality assurance, adaptability, biases, maintenance complexity, resistance to change, privacy concerns, and

			recent timetabling systems for fewer clashes. Advantages include reduced manpower, time savings, and improved data accuracy.	limitations in customization.
6.	Automatic Timetable Generation Using Genetic Algorithm Authors: Kehinde Willams Micheal Ajinaja	International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV9IS070568 Published by: www.ijert.org Vol. 9 Issue 07, July-2020	Utilizing an evolutionary algorithm, the approach tackles a heavily constrained university timetabling problem with a problem-specific chromosome representation. Heuristics and context-based reasoning ensure feasible timetables within reasonable computing time, and an intelligent adaptive mutation scheme accelerates convergence.	Drawbacks include the Genetic Algorithm not providing a 100% optimal timetable and being relatively slower in performance.
7.	Automated Timetable Generator Using Machine Learning Authors: Prashanta Kumar Shreedhar Sanakar Praveen Kumar Syed Muhammad Usmad Vani	International Research Journal of Modernization in Engineering Technology and Science Volume:02/Issue:08/August-2020 e-ISSN: 2582-5208	Utilizes artificial intelligence techniques such as optimization and iterative development, combined with Monte Carlo methods and surrogate modeling for evaluation. Employs a decision tree algorithm and linear regression for automated timetable generation, addressing manual assignment inefficiencies in scheduling classes.	Manual adjustments may still be necessary despite automation, and evaluating results is challenging due to the system's configurability and the complexity of scheduling constraints.

8.	Implementation of a Time Table Generator Using Visual Basic.Net Authors: Joseph M. Mom Jonathan Enokela	ARPN Journal of Engineering and Applied Sciences ISSN 1819-6608 Published by: www.arpnjournals.com VOL. 7, NO. 5, MAY 2012	The methodology for developing the timetable generator includes defining classes, implementing separate modules, designing a graphical interface, and visualizing the implementation flow through flow charts and block diagrams.	Manual adjustments needed, software and database technology dependence, and potential complexity with large datasets.

2.2 Conclusion drawn from literature review:

The above literature review proposes an automated system for generating timetables in educational institutions using algorithm like Genetic Algorithm. By optimizing resource utilization and streamlining the process, the system aims to overcome manual timetable generation challenges. The implementation of a user-friendly interface and software modules simplifies lecture allocation, saving time and manpower. Overall, the automated timetable generation system presented offers a more efficient and optimized solution for educational institutions.

2.3 Scope of the Research Work:

Automated timetable generation systems, specifically those utilizing Genetic Algorithm, present promising solutions for educational institutions. These systems enhance efficiency and stability in timetable creation by optimizing resource utilization and addressing various constraints, thereby improving user experience. With intuitive interfaces and user-friendly features, they streamline the scheduling process, leading to significant time and resource savings. Potential future improvements, like automated notifications and integration with management systems, offer avenues for further enhancement. In summary, these conclusions emphasize the substantial positive impact of automated timetable generators on scheduling processes within educational institutions.

CHAPTER 03

REQUIREMENTS ANALYSIS

3. REQUIREMENTS ANALYSIS

Developing a Smart Timetable Generation system begins by understanding the unique needs of educational institutions. Through discussions with stakeholders and careful data collection, we lay the groundwork for a system that simplifies timetable creation and boosts scheduling efficiency. This initial phase guides decisions on how the system will be built, ensuring it meets the specific demands of educational scheduling.

3.1 Stakeholder Requirements:

This Project involved Stakeholder Requirements for understanding the scheduling needs and preferences of educational institutions, administrators, teachers, and students. This input guides the development process to create a system that aligns with the diverse demands of the stakeholders. Following is the required constraints given by the user:

3.1.1 Administrators:

- User-Friendly Interface: The Smart Timetable Generator should have an intuitive and easy-to-use interface, allowing administrators to navigate through the system effortlessly.
- Customization Options: Administrators should have the ability to customize the timetable creation process, specifying rules, preferences, and constraints.

3.1.2 End Users (Faculty and Students):

- User Feedback Mechanism:

End users need a user-friendly feedback mechanism within the application to provide insights into their experiences with the schedules.

End users desire a system that actively incorporates their feedback to continuously improve the scheduling algorithms and user experience.

3.2 Functional Requirements:

Functional requirements for the Smart Timetable Generation system encompass the specific capabilities and features needed, such as efficient scheduling algorithms, real-time updates, and user-friendly interfaces. These requirements guide the development to ensure the system meets its intended functionalities, contributing to streamlined timetable creation and enhanced user experience.

3.2.1 Cross-Platform Application:

• The Smart Timetable Generator application should be developed using the Flutter framework to ensure a consistent and user-friendly experience across iOS, Android, and Web platforms.

• The interface should be responsive and adaptable to various screen sizes and resolutions.

• Following are the technologies involved in cross-platform application development:

(1) Flutter:

Flutter is an open-source UI software development toolkit created by Google.
It is used to build natively compiled applications for mobile, web, and desktop from a single codebase. It provides a reactive framework, expressive UI, and a set of pre-designed widgets.

Key Features:

• Hot Reload: Flutter's hot reload feature allows developers to see the results of their code changes instantly, without restarting the entire application.

• Single Codebase: Developers can write a single codebase for both iOS and Android applications, reducing development time and effort.

• Widgets: Flutter uses a widget-based architecture, where everything, including the application itself, is a widget. This makes UI development highly modular and customizable.

• Expressive UI: Flutter offers a rich set of customizable widgets that allow developers to create highly expressive and visually appealing user interfaces.

• Dart Programming Language: Flutter uses the Dart programming language, which is designed for building scalable and performant applications.

Use Cases:

• Flutter is used to build mobile applications for both iOS and Android.

• It is increasingly used for building web and desktop applications as well.

(2) Dart:

Dart is a programming language developed by Google and is the primary language used for building applications with Flutter. It is designed for building scalable and efficient web, mobile, and server applications.

Key Features:

• Object-Oriented: Dart is an object-oriented language with classes and interfaces, making it suitable for building modular and maintainable code.

• Just-in-Time (JIT) and Ahead-of-Time (AOT) Compilation: Dart supports both JIT and AOT compilation, enabling developers to choose the best compilation strategy based on the application's requirements.

• Strong Typing: Dart is a statically-typed language, providing better tooling support and catching potential errors during development.

• Asynchronous Programming: Dart has built-in support for asynchronous programming, making it well-suited for handling asynchronous tasks efficiently.

Use Cases:

• Dart is primarily used for building applications with Flutter.

• It can be used for server-side development, web applications, and command-line tools.

(**3) Android Studio:**

Android Studio is the official integrated development environment (IDE) for Android app development. It is based on JetBrains' IntelliJ IDEA and tailored for Android development.

Key Features:

• Intelligent Code Editor: Android Studio provides a powerful code editor with features like code completion, code analysis, and smart refactoring tools.

• Visual Layout Editor: The visual layout editor allows developers to design UI layouts visually, making it easier to create complex user interfaces.

• Gradle Build System: Android Studio uses the Gradle build system for managing project dependencies and building Android applications.

• Emulator: The built-in emulator allows developers to test their applications on various

Android device configurations.

• Version Control Integration: Android Studio integrates with version control systems like Git, making it easy for developers to manage and track changes to their code.

Use Cases:

• Android Studio is the preferred IDE for developing native Android applications.

• It supports Kotlin and Java for Android app development.

Flutter and Dart work together to provide a comprehensive framework for building cross-platform applications, while Android Studio serves as the primary IDE for native Android development. Developers can leverage these tools to create powerful and visually appealing applications for various platforms.

3.2.2 Efficient Class Schedules Algorithms:

• Implement genetic algorithms and simulated annealing for timetable generation, considering f actors such as subject preferences, teacher availability, and classroom constraints.

• Provide options for administrators to customize algorithmic parameters and constraints based on institutional requirements.

• Genetic Algorithm described as in following way:

A genetic algorithm (GA) is a search and optimization technique inspired by the principles of natural selection and genetics. It is part of a broader class of algorithms known as evolutionary algorithms and is commonly used for solving optimization and search problems. Genetic algorithms are particularly effective in finding approximate solutions to complex problems where finding an exact solution might be impractical or computationally expensive.

3.2.3 Data Storage and Retrieval (Firebase):

Firebase is a comprehensive mobile and web application development platform provided by Google. It offers a set of tools and services that enable developers to build, deploy, and scale applications quickly. Firebase is known for its real-time database, authentication services, cloud functions, hosting, and various other features.

Key Features of Firebase:

• Real-Time Database: Firebase provides a NoSQL cloud database that supports real-time synchronization. This is particularly useful for applications that require instant updates across multiple clients.

• Authentication: Firebase Authentication offers secure and straightforward user authentication, supporting email/password, social media logins, and more.

• Cloud Firestore: An alternative to the Real-Time Database, Cloud Firestore is a scalable NoSQL document database that allows for more complex queries and hierarchical data structures.

• Cloud Functions: Firebase Cloud Functions enable serverless computing, allowing developers to run backend code in response to events triggered by Firebase features or HTTP requests.

• Hosting: Firebase Hosting provides fast and secure web hosting for static and dynamic content, serving as a global content delivery network (CDN).

• Cloud Storage: Firebase Storage allows developers to store and serve user-generated content such as images, videos, and other files.

• Authentication: Firebase Authentication offers secure and straightforward user authentication, supporting email/password, social media logins, and more.

• Analytics: Firebase Analytics provides insights into user behavior, allowing developers to make data-driven decisions for app improvement.

3.2.4 Data Analytics for Schedule Optimization:

• Utilize data analytics tools to analyze historical scheduling data and identify patterns for optimizing future schedules.

• Provide administrators with actionable insights and recommendations derived from data analytics.

3.2.5 Effort Reduction and Resource Allocation Improvement:

• Minimize manual effort in the timetable creation process by automating tasks related to scheduling and resource allocation.

• Provide features that optimize resource allocation based on class sizes, room capacities, and teacher availability.

3.3 Non-Functional Requirements:

3.3.1 Usability:

• The application should have an intuitive and user-friendly interface, ensuring that users can easily navigate and interact with the Smart Timetable Generator.

• Provide clear and concise documentation and onboarding materials to assist users in understanding and using the application effectively.

3.3.2 Performance:

• The system should generate timetables promptly, with a maximum response time of [define specific time] seconds to ensure a responsive user experience.

• The application should handle a concurrent user load of [define specific number] users without significant performance degradation.

3.3.3 Reliability:

• The Smart Timetable Generator should have a high level of availability, with a target uptime of [define specific percentage] during normal operating hours.

• Implement robust error handling and recovery mechanisms to minimize disruptions in case of unexpected errors.

3.3.4 Security:

• Ensure data confidentiality and integrity by implementing secure authentication and authorization mechanisms for user access.

• Protect against common security threats such as SQL injection, cross-site scripting (XSS), and data breaches.

3.3.5 Compatibility:

• Ensure cross-browser compatibility, supporting the latest versions of popular web browsers (e.g., Chrome, Firefox, Safari).

• The mobile application should be compatible with the latest versions of iOS and Android operating systems.

3.3.6 Maintainability:

• Use modular and well-documented code to facilitate ease of maintenance and updates.

• Provide a mechanism for administrators to apply system updates and patches seamlessly.

3.3.7 Flexibility and Adaptability:

• The Smart Timetable Generator should be adaptable to different educational institutions, accommodating various scheduling needs, policies, and constraints.

• Allow administrators to easily configure and customize scheduling parameters and rules based on institutional requirements.

3.4 Constraints:

3.4.1 Hard Constraints:

1. No Teacher Conflict: Explain the necessity of ensuring that no teacher has overlapping classes.
2. Room & Lab Availability: Detail the importance of scheduling classes only when rooms and labs are available.
3. Break Between Every Two Lectures: Elaborate on the significance of allowing breaks to prevent fatigue and aid concentration.
4. No Lecture and Lab Back-to-Back: Explain why lectures and labs should not be scheduled consecutively to allow for transition time and prevent conflicts.

3.4.2 Soft Constraints:

1. Arrangement of P.E. Lectures and Labs: Justify the need for specific arrangements of physical education classes.
2. Adjustment in O.E. Slots with respect to Interbranch Timetables: Explain the necessity of coordinating timetables across different branches or departments.

CHAPTER 04

DESIGN AND ARCHITECTURE

4. DESIGN AND ARCHITECTURE

In crafting the design and architecture for a Smart Timetable Generation system, the focus is on creating a robust and efficient framework that seamlessly integrates scheduling algorithms, user interfaces, and database management. This involves careful consideration of scalability, modularity, and the effective utilization of technologies to meet the unique demands of educational scheduling. The design phase lays the foundation for a system that not only optimizes timetable creation through advanced algorithms but also provides an intuitive and responsive user experience. Through thoughtful architectural decisions, the Smart Timetable Generation system aims to deliver a reliable solution that adapts to the evolving needs of educational institutions.

4.1 System Architecture:

The system architecture for the Smart Timetable Generation encompasses a multi-tiered structure, integrating front-end, back-end, and database components to ensure a comprehensive and efficient solution. The front-end, developed using technologies like Flutter for cross-platform compatibility, provides a user-friendly interface for administrators, teachers, and students. The back-end, powered by a robust server infrastructure, hosts the scheduling algorithms, WebSocket for real-time updates, and integrates with a database, allowing seamless data retrieval and storage through a chosen Database Management System (DBMS) such as MySQL. This scalable and modular architecture is designed to handle the complexities of educational scheduling, ensuring optimal performance and adaptability to future enhancements. Below is a high-level system architecture for a Smart Timetable Generator based on the specified objectives:

4.1.1 User Interface (UI) Layer:

1. Flutter Application:

• Develop a cross-platform mobile and web application using the Flutter framework to provide a consistent and user-friendly interface for administrators, teachers, and students.

• Utilize Flutter's widgets to create responsive and visually appealing UI components.

4.1.2 Application Logic Layer:

1. Genetic Algorithms Module:

• Implement the core scheduling algorithm using genetic algorithms and simulated annealing for efficient class schedule generation.

• Provide customization options for algorithmic parameters based on institutional requirements.

2. Data Analytics Module:

• Incorporate a Data Analytics module to analyze historical scheduling data, identify patterns, and optimize future schedules.

• Use analytics insights to improve the efficiency and effectiveness of the scheduling process.

3. User Feedback Module:

• Integrate a user feedback mechanism within the application to collect feedback from teachers and students regarding generated schedules.

• Implement a system to analyze and prioritize user feedback for continuous improvement.

4.1.3 Data Management Layer:

1. Firebase Integration:

• Utilize Firebase for data storage and retrieval to ensure secure and efficient handling of timetable data.

• Implement Firebase Realtime Database or Cloud Firestore for real-time synchronization across devices.

2. WebSocket Integration:

• Implement WebSocket technology for real-time updates, ensuring that users receive instant notifications about any changes in their schedules.

• Enable bidirectional communication for seamless interaction between the application and the server.

4.1.4 Security Layer:

1. Authentication and Authorization:

• Implement secure user authentication and authorization mechanisms to protect sensitive data.

• Use Firebase Authentication for user identity verification.

2. Data Security Measures:

• Apply encryption techniques to secure data transmission between the application and Firebase.

• Implement secure coding practices to prevent common security threats.

4.1.5 Cross-Cutting Concerns Layer:

1. Logging and Monitoring:

• Implement logging mechanisms to record system activities and errors for debugging and analysis purposes.

• Integrate monitoring tools to track system performance, identify issues, and ensure optimal operation.

2. Error Handling and Recovery:

• Implement robust error handling mechanisms to gracefully manage unexpected errors and prevent disruptions in service.

• Design a recovery system to restore the application to a stable state after encountering errors.

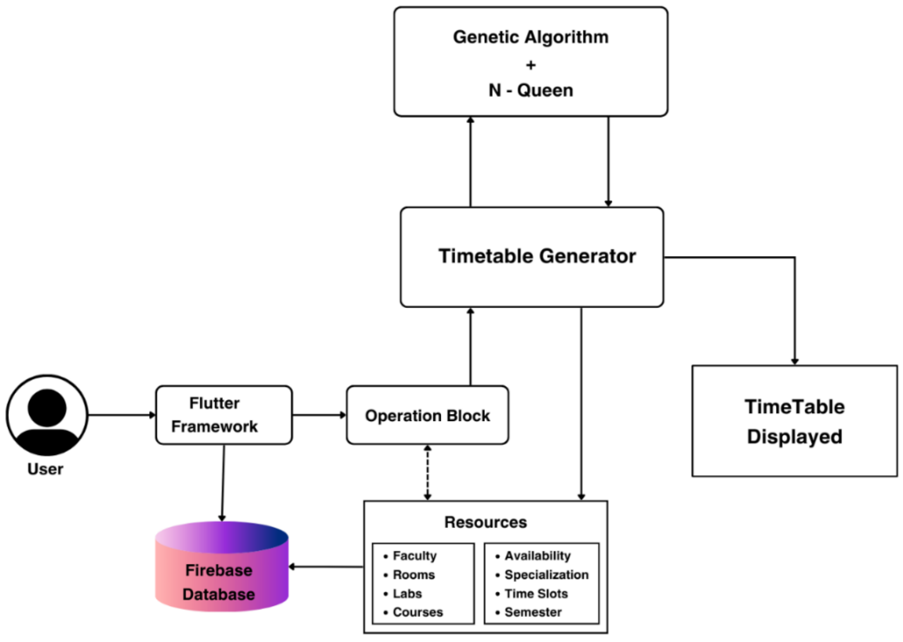


Figure 4.1 Proposed System Architecture

4.2 Database Design:

Designing the database for a Smart Timetable Generator involves defining the structure of the database, specifying the tables, relationships, and constraints necessary to store and manage the data efficiently. Below is a detailed database design for the Smart Timetable Generator based on the specified objectives:

- Implement proper data types for each field to optimize storage and retrieval.
- Utilize database triggers and stored procedures for enforcing business rules and maintaining data integrity.
- Implement appropriate indexing strategies based on query patterns to optimize data retrieval speed.
- Consider data partitioning and archiving strategies for managing large datasets efficiently.

4.3 Advantages of Proposed System:

1. In contrast to traditional manual timetabling methods, the system provides enhanced flexibility.
2. It requires only minimal processing or computing power.

3. Significantly reducing the time required to generate error-free timetables.
4. Facilitating easy data entry and revision through an intuitive interface.
5. Timetables generated typically fall within the range of 60% to 80% optimal solutions.
6. Virtually eliminating the need for paperwork.

4.4 System Design:

4.4.1 System Flowchart:

A flowchart is a graphical representation used to depict the steps, sequences, and decisions of a process or workflow. It serves as a valuable tool in planning, visualizing, documenting, and enhancing processes across different domains. Initially introduced by industrial engineers Frank and Lillian Gilbreth in 1921, flowcharts have become essential for analyzing and improving processes. Essentially, a flowchart is a visual map that outlines a series of steps in a structured manner.

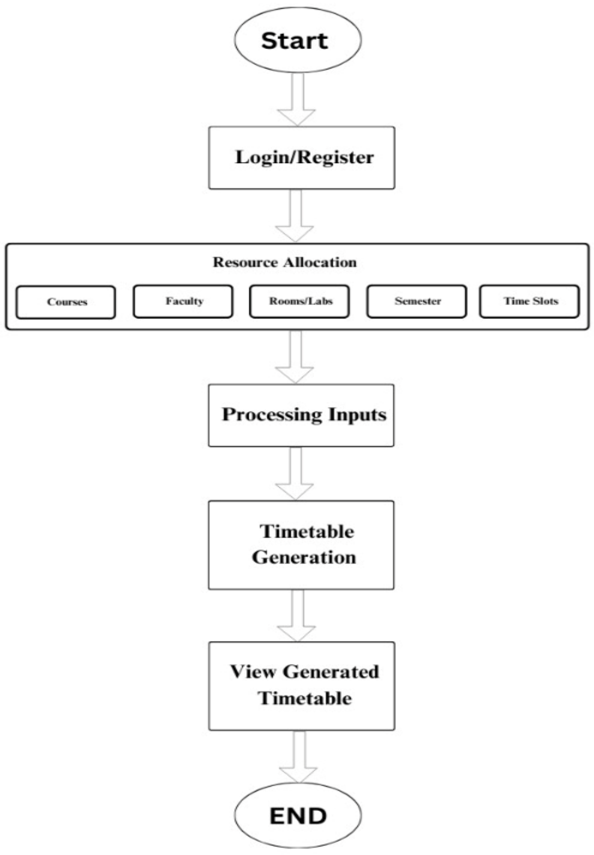


Figure 4.2: Flowchart of System

4.4.2 Use case Diagram:

A use-case diagram, in its fundamental display, depicts the interaction between a user and a system while delineating the specifications of a specific usage instance. It visually portrays the various user categories within a system and the varied methods by which they interact with it. Usually, this diagram is utilized in conjunction with textual usage instances and is frequently complemented by other types of diagrams. In this instance, personnel enter semester and subject particulars into the system, which subsequently processes this data to produce the timetable.

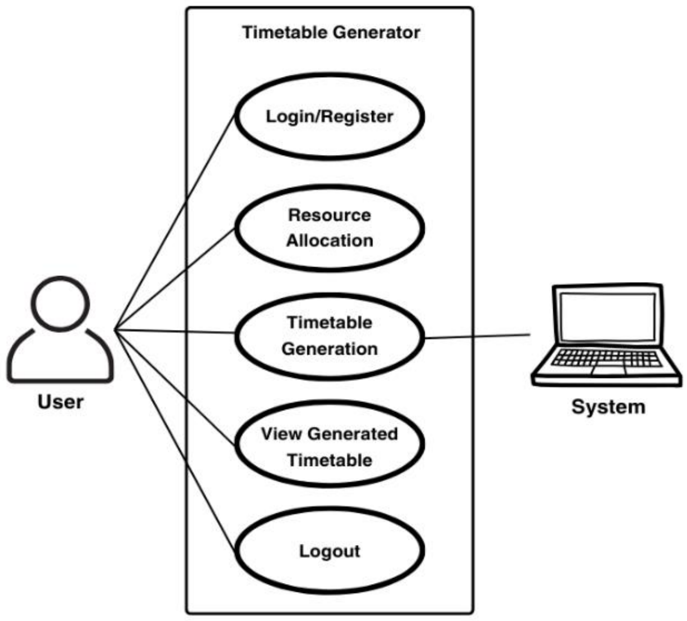


Figure 4.3: Use Case Diagram

4.5 ALGORITHM DESIGN

Algorithm design is a foundational step in developing a Smart Timetable Generator, focusing on creating intelligent and efficient methodologies to automate complex scheduling tasks. By employing optimization techniques like genetic algorithms or simulated annealing, the goal is to optimize resource allocation, minimize conflicts, and deliver a scalable solution tailored to the dynamic scheduling needs of educational institutions. This computational phase forms the backbone of the system, translating scheduling requirements into systematic and algorithmic approaches for the generation of intelligent and optimized timetables.

4.5.1 Genetic Algorithm:

A genetic algorithm is a heuristic search method inspired by Charles Darwin’s theory of natural evolution. It simulates the survival process where the most fit individuals are chosen for reproduction to generate offspring in the subsequent generation. The survival process begins with selecting the fittest individuals from a population. These individuals then produce offspring inheriting their parents' characteristics, which are added to the next generation. Offspring of parents with superior fitness are expected to have a better chance of survival. This iterative process continues until a generation with the fittest individuals is attained. This concept can be applied to problem-solving, where a group of solutions is considered, and the best ones are selected. Five phases are typically involved in a genetic algorithm

- Initial population
- Fitness function
- Selection
- Crossover
- Mutation

4.5.2 Working:

1. **Initialization:** Generate an initial population of chromosomes.
2. **Evaluation:** Evaluate the fitness of each chromosome in the population using the fitness function.
3. **Selection:** Select chromosomes from the current population for reproduction based on their fitness.
4. **Crossover:** Generate offspring by combining genetic material from selected parent chromosomes.
5. **Mutation:** Introduce random changes to the offspring chromosomes to maintain diversity.

6. **Replacement:** Create a new population for the next generation, possibly using elitism to retain the best solutions.
7. **Termination:** Repeat steps 2-6 until a termination condition is met (e.g., a maximum number of generations or a satisfactory solution is found).

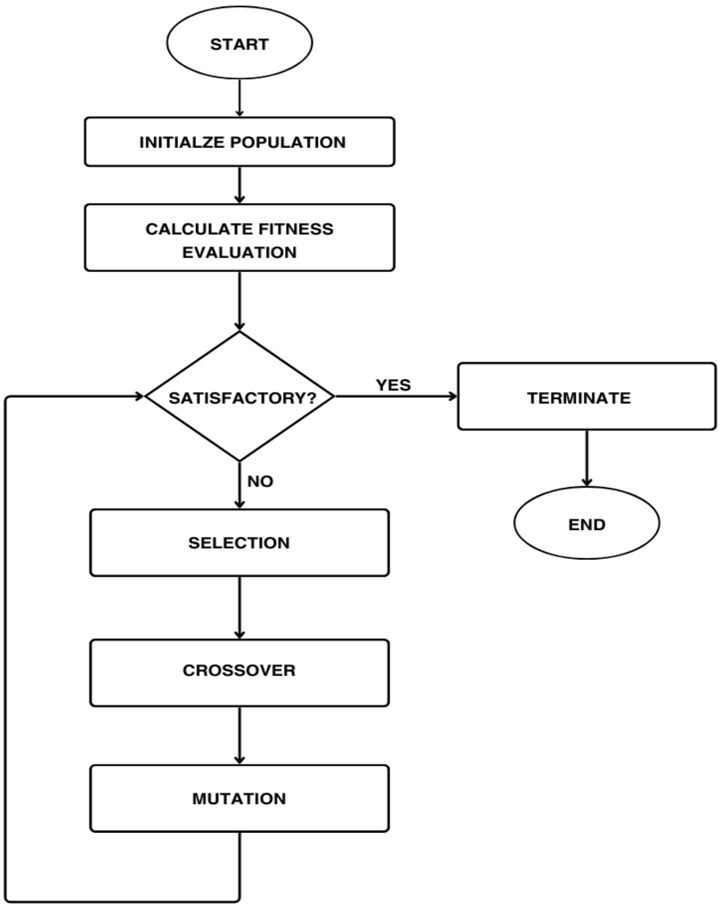


Figure 4.4: Genetic Algorithm Flow Chart

CHAPTER 05
IMPLEMENTATION

5. IMPLEMENTATION

The implementation phase is where our project takes shape, with all the envisioned components coming together. We employ various software development methodologies to build the modules of our project. Specifically, we utilize Dart as our programming language, Firebase as our database solution, Flutter as our framework, and the server infrastructure is supplied by our college. Our development environment operates on the Windows 10 operating system.

Modules that are developed in our project are as follows:

5.1 User Management:

User Management is a crucial component of our system, responsible for handling user accounts, authentication, and access control. This module ensures that only authorized users can access the system's functionalities while maintaining the security and integrity of user data.

5.1.1 Registration: The registration process allows users to create accounts by providing necessary information such as username, email address, and password. This information is securely stored in the system's database.

// Firebase Authentication package used

'package: firebase_auth/firebase_auth.dart';

// Function for user registration

void registerUser(String username, String password)

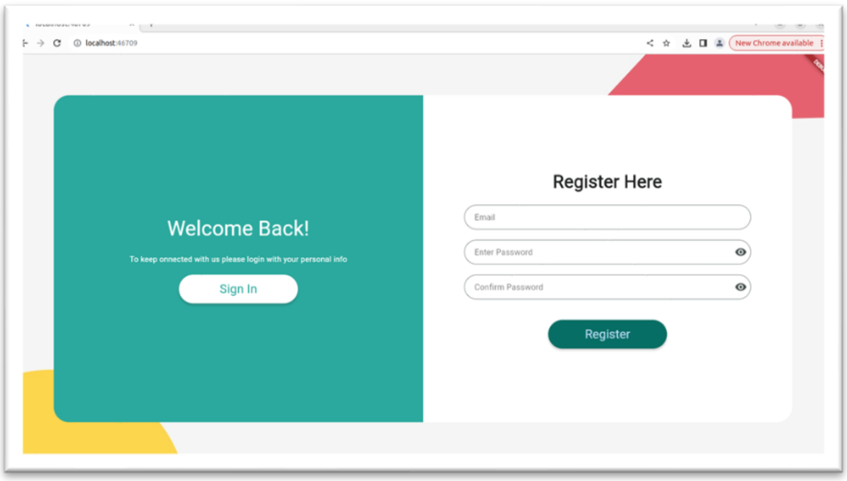


Figure 5.1 Register Page

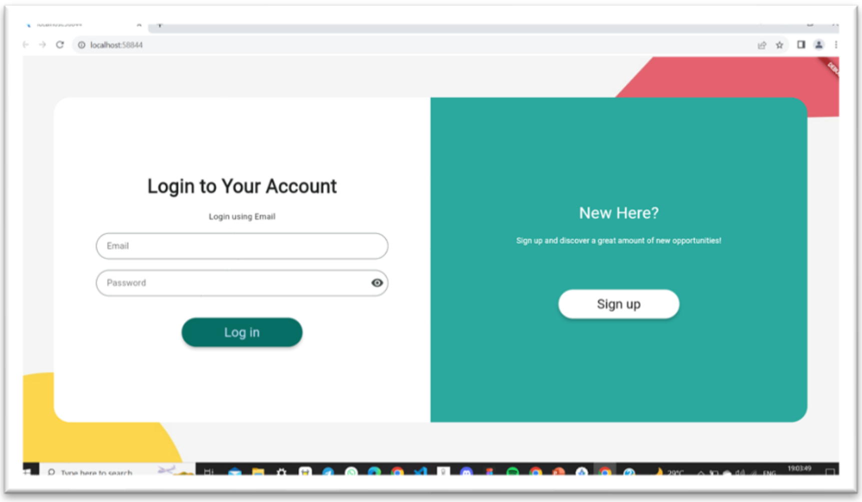
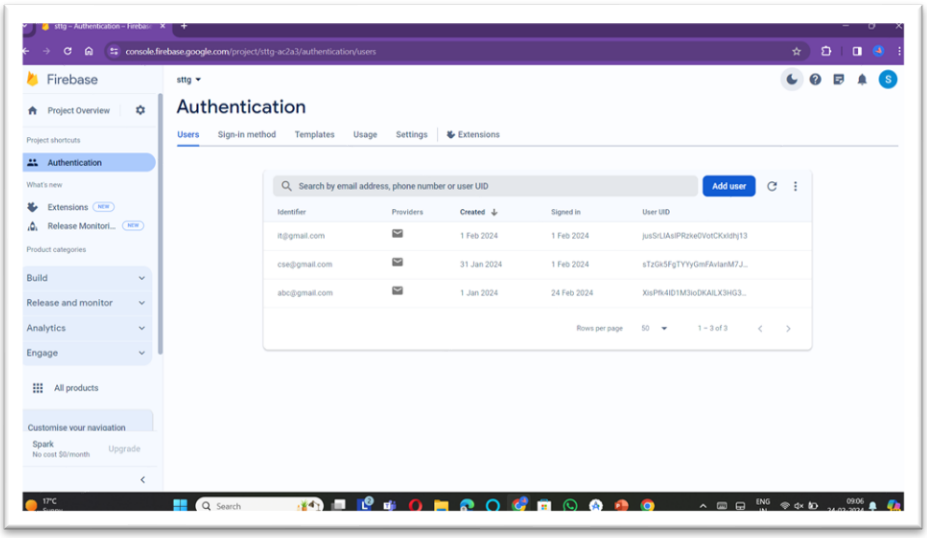


Figure 5.2 Login Page

5.1.2 Authentication: Once registered, users need to authenticate themselves to access the system. Authentication mechanisms, such as email verification or two-factor authentication, verify the identity of users before granting them access to the system.

```
// Function for user authentication
bool authenticateUser(String username, String password) {
// Implement user authentication logic
  return true; // Placeholder,
}
```



Identifier	Providers	Created	Signed in	User UID
it@gmail.com	✉	1 Feb 2024	1 Feb 2024	jusSrLIAsIPRzke0VotCKxldhj13
cse@gmail.com	✉	31 Jan 2024	1 Feb 2024	sTzGk5FgTYYyGmFAvlanM7J...
abc@gmail.com	✉	1 Jan 2024	24 Feb 2024	XisPfk4ID1M3ioDKAILX3HG3...

Rows per page 50 1 – 3 of 3

Figure 5.3 Authentication Page

5.1.3 Access Control: User roles and permissions are defined to regulate access to different parts of the system. For example, administrators may have access to administrative functions, while regular users may only have access to view their own timetables.

```
// Function for access control
void regulateAccess(String username, String resource, String role) {
// Implement access control logic
}
```

5.2 Data Input:

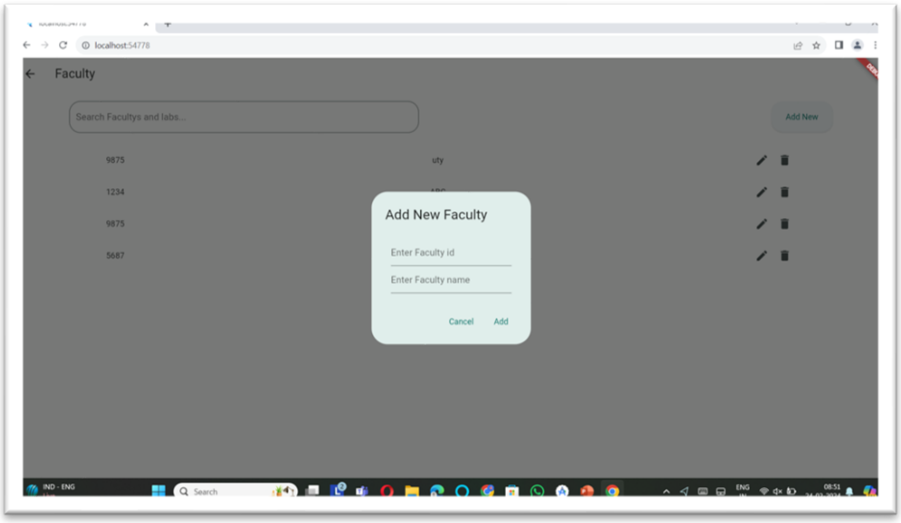
The Data Input module collects essential information required for timetable generation. This information includes details about courses, faculty members, classrooms, and user preferences.

5.2.1 Courses: Information about courses includes course names, codes, and timings. This data helps in scheduling classes and allocating resources accordingly.

```
// Firebase Firestore package used
package:cloud_firestore/cloud_firestore.dart';
// Add course data to Firestore
Future<void> addCourseData(String courseName, String instructor, String room)
async
```

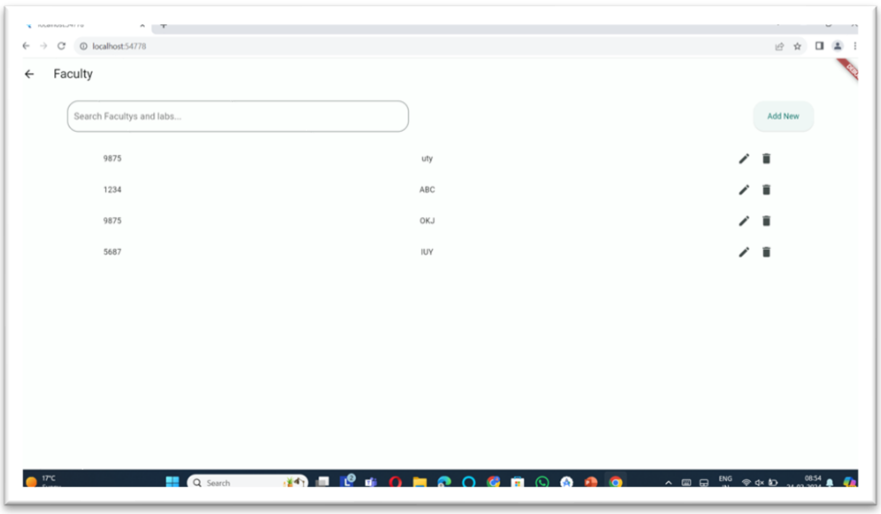
5.2.2 Faculty: Details about faculty members, such as their names, specialties, and availability, are collected to assign teaching responsibilities and avoid scheduling conflicts.

```
// Add faculty data to Firestore
Future<void> addFacultyData(String name, String email, String department) async
```



9875	uty
1234	ABC
9875	
5687	

Figure 5.4 Faculty Edit Page

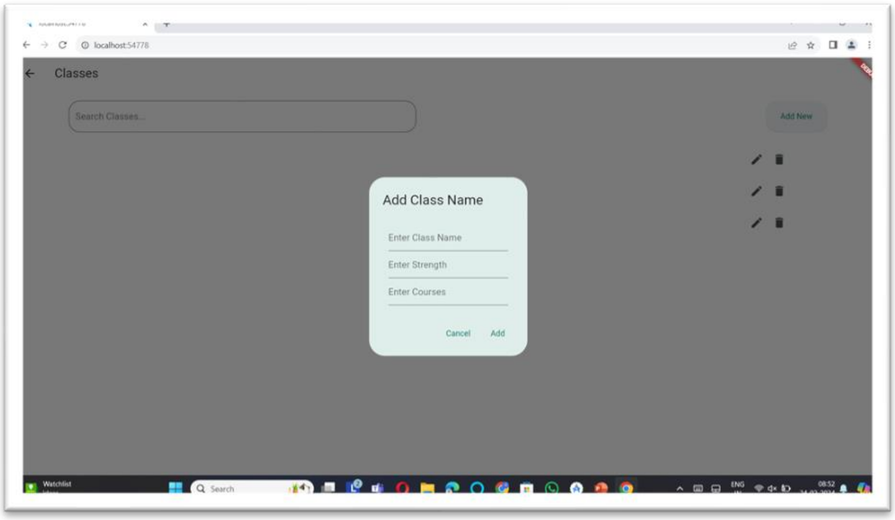


9875	uty
1234	ABC
9875	OKJ
5687	IUY

Figure 5.5 Faculty Input Page

5.2.3 Classrooms: Information about available classrooms, including their capacities, facilities, and availability slots, is gathered to allocate appropriate venues for classes.

```
// Add classroom data to Firestore
Future<void> addClassroomData(String ClassName, String id, String room) async
```


Classes

Search Classes...

Add New

Add Class Name

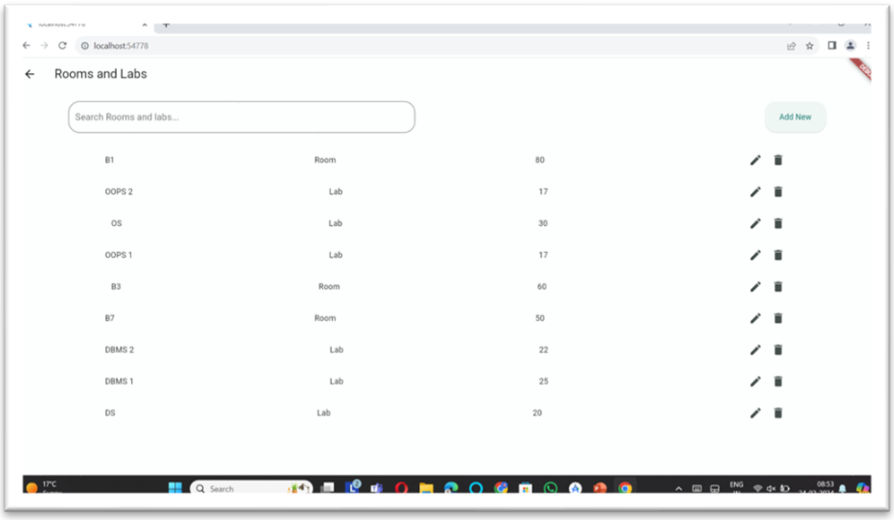
Enter Class Name

Enter Strength

Enter Courses

Cancel Add

Figure 5.6 Rooms and Labs Edit Page


Rooms and Labs

Search Rooms and labs...

Add New

B1	Room	80
OOPS 2	Lab	17
OS	Lab	30
OOPS 1	Lab	17
B3	Room	60
B7	Room	50
DBMS 2	Lab	22
DBMS 1	Lab	25
DS	Lab	20

Figure 5.7 Rooms and Labs Input Page

5.2.4 Preferences: User preferences, such as preferred class timings, days off, or specific teaching assignments, are collected to personalize the timetable generation process and meet user requirements.

5.3 Timetable Generation:

Timetable Generation is the core functionality of our system, where schedules are created using advanced algorithms to optimize resource utilization and meet specified criteria.

5.3.1 Algorithmic Approach: The timetable generation algorithm utilizes techniques like constraint satisfaction and optimization to navigate the solution space and produce optimal schedules. It considers constraints such as resource availability, time limitations, and user preferences to generate feasible timetables.

5.3.2 Resource Allocation: The algorithm allocates resources, such as classrooms and faculty members, to courses based on their availability and preferences. It aims to minimize conflicts and optimize resource utilization to ensure efficient scheduling.

5.3.3 Conflict Resolution: Conflicts, such as overlapping class timings or double bookings, are resolved intelligently by the algorithm to ensure that schedules are feasible and practical.

Following are the step-by-step functions used in the code:

```
// Timetable Generation Functions
// Function to perform selection
Timetable selection(List<Timetable> population) {
  // Implement selection logic to choose a parent timetable
  return population.first; // Placeholder, replace with actual logic
}
// Function to perform crossover
Timetable crossover(Timetable parent1, Timetable parent2) {
  // Implement crossover logic to generate a child timetable
  return parent1; // Placeholder, replace with actual logic
}
// Function to perform mutation
void mutation(Timetable child) {
  // Implement mutation logic (e.g., swap two subjects in the timetable)
}
// Function to calculate fitness
double calculateFitness() {
  // Implement your fitness calculation logic here
  // Return a value indicating how good the timetable is
  return 0.0;
}
```

5.4 User Interface:

The User Interface provides a user-friendly platform for interacting with the system, facilitating data input and timetable viewing.

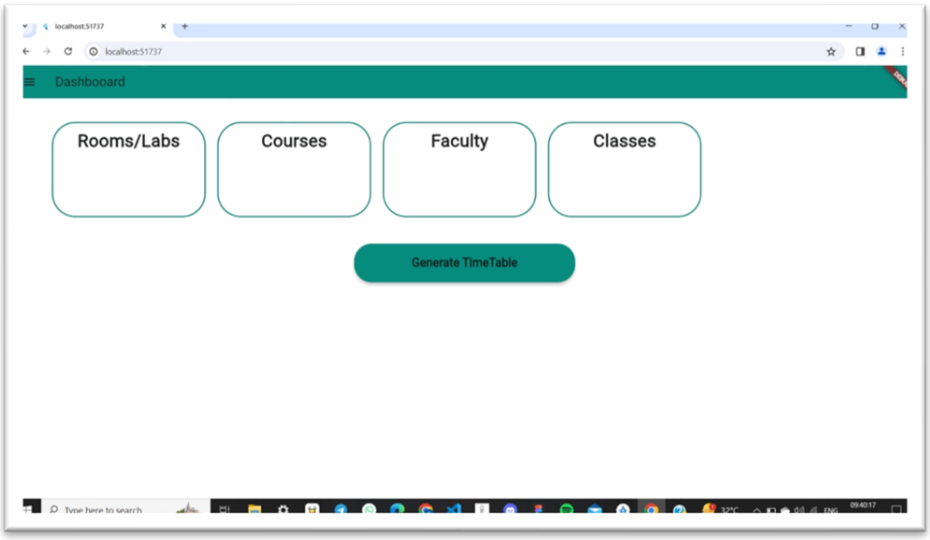


Figure 5.8 Dashboard

5.4.1 Input Forms: Intuitive input forms allow users to input their preferences and requirements easily. Clear instructions and user-friendly controls enhance the usability of the interface.

```
// User Interface Functions
// Function to input timetable requirements
void inputTimetableRequirements() {
  // Implement logic to input timetable requirements
}
```

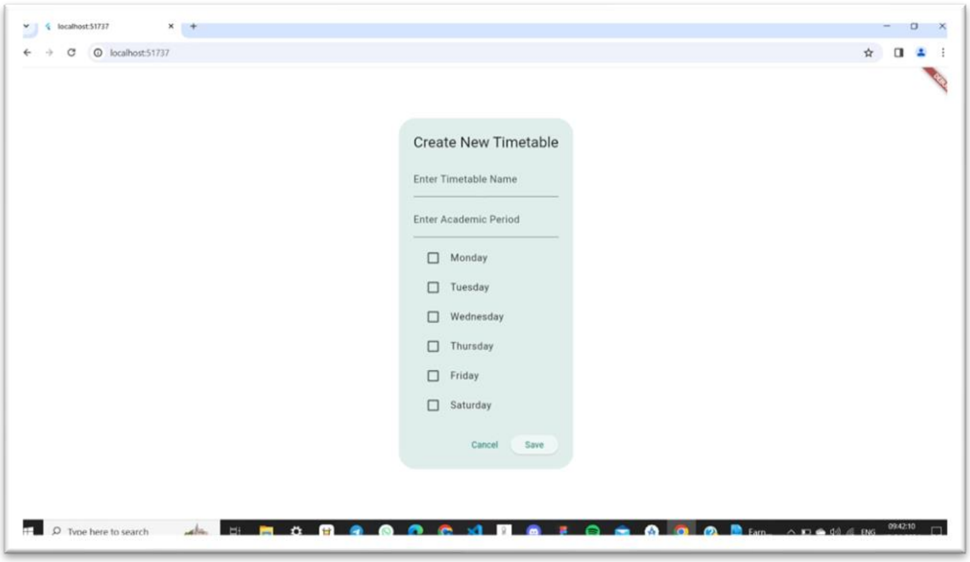
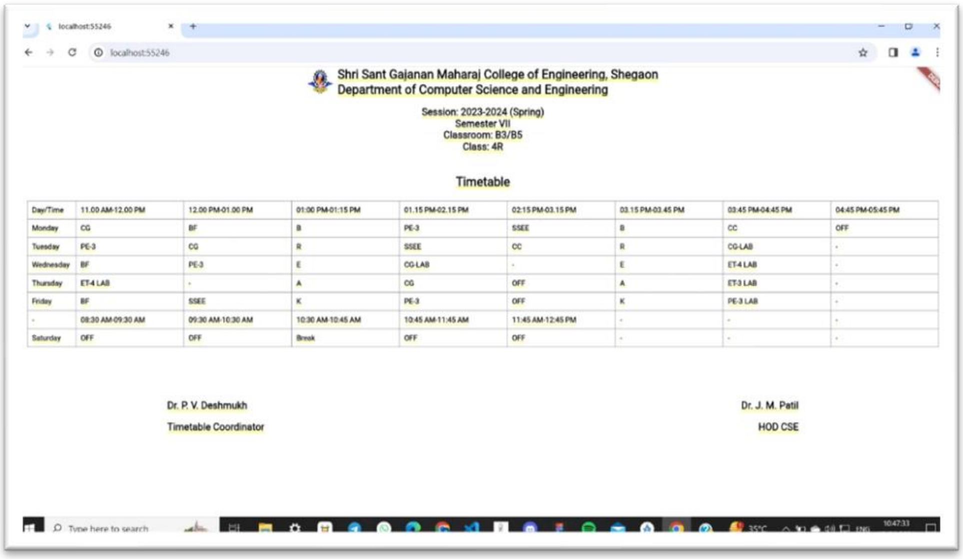


Figure 5.9 Information Fill Up Page

5.4.2 Timetable Display: Generated timetables are displayed in a visually appealing and easy-to-understand format. Color-coded schedules, interactive features, and filtering options enhance the user experience and make it easy to navigate through the timetable.

```
// Function to display generated timetables
void displayTimetables(List<Timetable> timetables) {
// Implement logic to display timetables in a user-friendly format
}
```


Shri Sant Gajanan Maharaj College of Engineering, Shegaon
Department of Computer Science and Engineering

Session: 2023-2024 (Spring)
Semester VII
Classroom: B3/B5
Class: 4R

Timetable

Day/Time	11.00 AM-12.00 PM	12.00 PM-01.00 PM	01:00 PM-01:15 PM	01.15 PM-02.15 PM	02:15 PM-03.15 PM	03.15 PM-03.45 PM	03:45 PM-04:45 PM	04:45 PM-05:45 PM
Monday	CG	BF	B	PE-3	SSEE	B	CC	OFF
Tuesday	PE-3	CG	R	SSEE	CC	R	CG-LAB	-
Wednesday	BF	PE-3	E	CG-LAB	-	E	ET-4 LAB	-
Thursday	ET-4 LAB	-	A	CG	OFF	A	ET-3 LAB	-
Friday	BF	SSEE	K	PE-3	OFF	K	PE-3 LAB	-
-	08:30 AM-09:30 AM	09:30 AM-10:30 AM	10:30 AM-10:45 AM	10:45 AM-11:45 AM	11:45 AM-12:45 PM	-	-	-
Saturday	OFF	OFF	Break	OFF	OFF	-	-	-

Dr. P. V. Deshmukh
Timetable Coordinator

Dr. J. M. Patil
HOD CSE

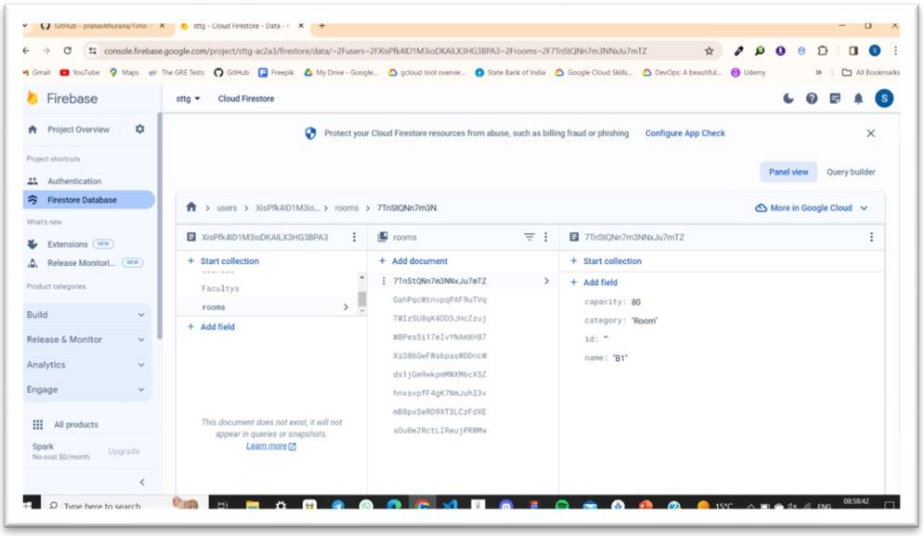
Figure 5.10 Generated Timetable Page

5.5 Database and Security:

The Database & Security module manages the storage of data and ensures its protection from unauthorized access or breaches.

5.5.1 Data Storage: The system's database securely stores all relevant data, including user information, course details, timetable schedules, and preferences. Data integrity and consistency are maintained to ensure the accuracy and reliability of information.

```
// Database and Security Functions
// Function to store data securely
void storeDataSecurely(dynamic data) {
  // Implement logic to store data securely
}
```



users > XisPfk4ID1M3io... > rooms > 7TnStQNn7m3N

XisPfk4ID1M3ioDKAILX3HG3BPA3	rooms	7TnStQNn7m3NNxJu7mTZ
Facultys	7TnStQNn7m3NNxJu7mTZ	capacity: 80
rooms	GahPqcWtnvpqPAF9uTVq	category: "Room"
	TWIzSU8qK4DD3JHcZzuj	id: ""
	W8Pes5i17eIvYNAmXH87	name: "B1"
	Xi086GwFWs6pasWDDncW	
	ds1jGm9wkpmMWXM6cX5Z	
	hnxsvpfF4gK7NmJuh13v	
	mB8pxSeRD9XT3LCzFdXE	
	sOuBe2RctLIReujPRWMw	

Figure 5.11 Cloud Firestore Page

5.5.2 Security Measures: Strong security measures, such as encryption, access controls, and regular backups, are implemented to protect sensitive data from unauthorized access, data loss, or corruption. Security protocols are continuously monitored and updated to mitigate potential risks and vulnerabilities.

```
// Function to retrieve data securely
dynamic retrieveDataSecurely() {
  // Implement logic to retrieve data securely
  return null; // Placeholder, replace with actual retrieval logic
}
```

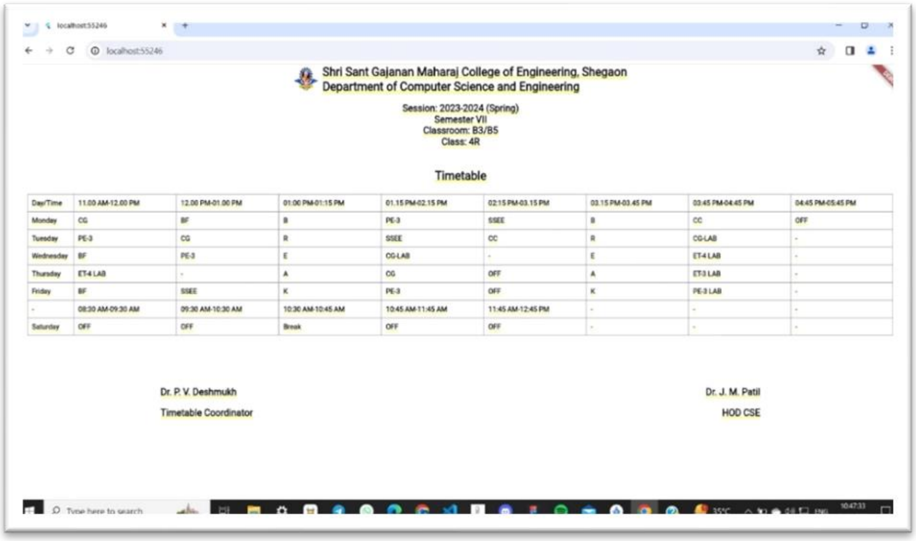
By elaborating on each module in this manner, we provide a comprehensive understanding of their functionalities and contributions to the overall system. These modules work together seamlessly to create a robust and user-friendly timetable management system that meets the diverse needs of educational institutions, corporate organizations, and event planners.

CHAPTER 06
RESULT AND DISCUSSION

6. RESULT AND DISCUSSION

1. Implemented secure login functionality allowing user access to the timetable system.
2. Established role-based authorization, ensuring appropriate access controls for different user roles.
3. Integrated Firebase for efficient storage and real-time data input, enhancing data management.
4. Utilized Firestore to segregate and store timetable data, ensuring scalability and reliability.
5. Successfully implemented a genetic algorithm for timetable generation, incorporating selection, crossover, and mutation strategies to optimize resource allocation, resolve conflicts, and efficiently produce schedules meeting specified criteria.

The result of this research work is the timetable generated for the college/ university.



Shri Sant Gajanan Maharaj College of Engineering, Shegaon
Department of Computer Science and Engineering

Session: 2023-2024 (Spring)
Semester VII
Classroom: B3/B5
Class: 4R

Timetable

Day/Time	11.00 AM-12.00 PM	12.00 PM-01.00 PM	01:00 PM-01:15 PM	01.15 PM-02.15 PM	02:15 PM-03.15 PM	03.15 PM-03.45 PM	03:45 PM-04:45 PM	04:45 PM-05:45 PM
Monday	CG	BF	B	PE-3	SSEE	B	CC	OFF
Tuesday	PE-3	CG	R	SSEE	CC	R	CG-LAB	-
Wednesday	BF	PE-3	E	CG-LAB	-	E	ET-4 LAB	-
Thursday	ET-4 LAB	-	A	CG	OFF	A	ET-3 LAB	-
Friday	BF	SSEE	K	PE-3	OFF	K	PE-3 LAB	-
-	08:30 AM-09:30 AM	09:30 AM-10:30 AM	10:30 AM-10:45 AM	10:45 AM-11:45 AM	11:45 AM-12:45 PM	-	-	-
Saturday	OFF	OFF	Break	OFF	OFF	-	-	-

Dr. P. V. Deshmukh
Timetable Coordinator

Dr. J. M. Patil
HOD CSE

Figure 6.1: Generated Timetable

CHAPTER 07

CONCLUSION, CONTRIBUTION AND FUTURE SCOPE

7. CONCLUSION, CONTRIBUTION AND FUTURE SCOPE

7.1 Conclusion:

Our system addresses the challenges of managing faculty members and scheduling lectures effectively. Through automation and advanced algorithms, it streamlines the process of generating timetables for various courses and semesters. With its user-friendly interface and efficient processing, it significantly reduces the time and effort required for timetable creation. By minimizing manual intervention and errors, our system optimizes productivity and resource utilization, ultimately saving valuable time and manpower.

7.2 Contribution:

The implementation of automation and advanced algorithms in our system marks a significant advancement in efficiently managing faculty members and scheduling lectures. Through the utilization of these technologies, we have successfully streamlined the cumbersome process of timetable generation across various courses and semesters. A standout feature of our system is its user-friendly interface, designed to ensure accessibility even for users with limited technical expertise. This accessibility not only enhances user experience but also encourages widespread adoption among faculty members and administrative staff. Moreover, the system's efficient processing capabilities play a crucial role in reducing the time and effort traditionally associated with timetable creation. By automating repetitive tasks and optimizing resource allocation, we have significantly minimized manual intervention and errors, resulting in more accurate timetables and reducing the need for constant adjustments.

7.3 Future scope:

1. Optimized Faculty Scheduling: This feature would automate faculty allocation within timetables by factoring in workload, expertise, and availability. Imagine a system that generates conflict-free schedules that maximize faculty utilization while meeting course needs. This could significantly reduce administrative burden, ensure fairer workload distribution, and potentially contribute to increased faculty satisfaction.

2. Cross-Platform Availability: Expanding the system's reach by making it accessible across various platforms (web, mobile apps, etc.) would enhance user flexibility and convenience. Students and faculty could access their schedules, submit requests, and receive updates on any device, at any location. This would not only improve user experience but also cater to the growing trend of mobile-first interaction.

3. Integrated Feedback Loop: Implementing a system for gathering and analysing user feedback on the generated timetables would be invaluable. This could involve faculty and student surveys, allowing for adjustments to the allocation algorithm based on real-world experiences. This feedback loop would ensure continuous improvement and a system that adapts to the evolving needs of the educational institution.

REFERENCES

REFERENCES

[1] Shraddha Ambhore, Pooja Walke, Rohit Ghundgrudkar, Akshay Alone, Anushree Khedkar, Automatic Timetable Generator , IJRESM | Volume-3, Issue-3, March-2020 | ISSN (Online): 2581-5792 | Published by: www.ijresm.org.

[2] Shraddha Thakare, Tejal Nikam, Prof. Mamta Patil, Automated Timetable Generation using Genetic Algorithm, International Journal of Engineering Research & Technology, IJERT | Vol. 9 Issue 07, July-2020 | ISSN: 2278-0181 Published by: www.ijert.org.

[3] Rutuja Kavade, Sohail Qureshi, Nikita Veer, Vaishnavi Ugale, Prof. Priyanka Agrawal, Smart Time Table System Using AI and ML,2023 IJCRT | Volume 11, Issue 5 May 2023 | ISSN: 2320-2882 | Published by: www.ijcrt.org.

[4] Mrs. G. Maneesha, T. Deepika, S. BhanuSri ,N. Ravi Kumar, P. Siva Nagamani, Automatic Time Table Generation Using Genetic Algorithm, JETIR Journal – Journal of Emerging Technologies and Innovative Research, July 2021 | ISSN-2349-5162 | Published by: www.jetir.org.

[5] Henry Techie-Menson, Paul Nyagorme, Design and Implementation of a Web-Based Timetable System for Higher Education Institutions, March 4, 2021 | International Journal of Educational Research and Information Science. Vol. 7, No. 1, 2021, pp. 1-13 | Published by: www.openscienceonline.com/journal/eirs.

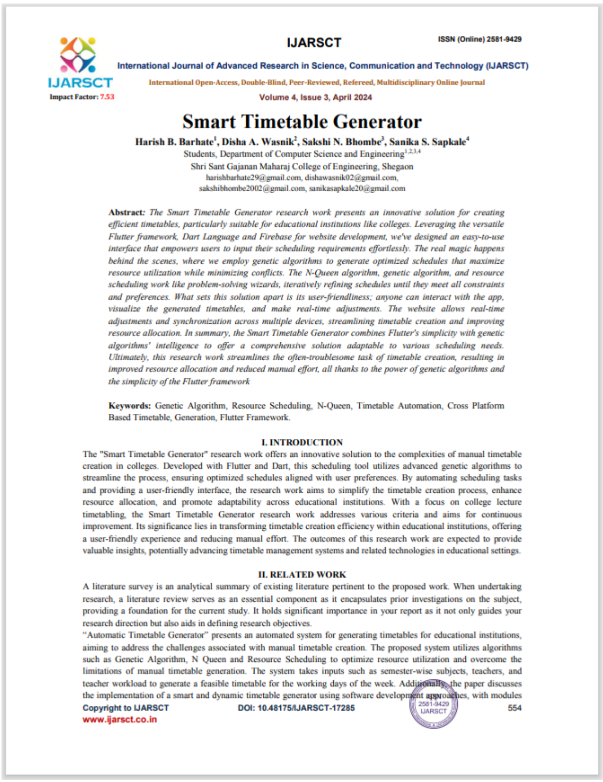
[6] Kehinde Williams, Micheal Ajinaja, Automatic Timetable Generation Using Genetic Algorithm, International Journal of Engineering Research & Technology | IJERT | ISSN: 2278-0181 | Vol. 9 Issue 07, July-2020 | Published by: www.ijert.org.

[7] Prashanta Kumar, Shreedhar Sanakar, Praveen Kumar, Syed Muhammad Usman, Vani, AUTOMATED TIMETABLE GENERATOR USING MACHINE LEARNING, e-ISSN: 2582-5208 International Research Journal of Modernization in Engineering Technology and Science Volume:02| Issue:08 | August 2020 Impact Factor- 5.354 Published by: www.irjmets.com.

[8] Joseph M. Mom and Jonathan A. Enokela, IMPLEMENTATION OF A TIME TABLE GENERATOR USING VISUAL BASIC.NET VOL. 7, NO. 5, MAY 2012 ISSN 1819-6608 ARPN Journal of Engineering and Applied Sciences | Published by: www.arpnjournals.com.

LIST OF DELIVERABLES ON PRESENT WORK

PUBLISHED RESEARCH PAPER



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, April 2024

Smart Timetable Generator

Harish B. Barhate[1], Disha A. Wasnik[2], Sakshi N. Bhombe[3], Sanika S. Sapkale[4]
Students, Department of Computer Science and Engineering[1,2,3,4]
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
harishbarhate29@gmail.com, dishawasnik02@gmail.com,
sakshibhombe2002@gmail.com, sanikasapkale20@gmail.com

Abstract: *The Smart Timetable Generator research work presents an innovative solution for creating efficient timetables, particularly suitable for educational institutions like colleges. Leveraging the versatile Flutter framework, Dart Language and Firebase for website development, we've designed an easy-to-use interface that empowers users to input their scheduling requirements effortlessly. The real magic happens behind the scenes, where we employ genetic algorithms to generate optimized schedules that maximize resource utilization while minimizing conflicts. The N-Queen algorithm, genetic algorithm, and resource scheduling work like problem-solving wizards, iteratively refining schedules until they meet all constraints and preferences. What sets this solution apart is its user-friendliness; anyone can interact with the app, visualize the generated timetables, and make real-time adjustments. The website allows real-time adjustments and synchronization across multiple devices, streamlining timetable creation and improving resource allocation. In summary, the Smart Timetable Generator combines Flutter's simplicity with genetic algorithms' intelligence to offer a comprehensive solution adaptable to various scheduling needs. Ultimately, this research work streamlines the often-troublesome task of timetable creation, resulting in improved resource allocation and reduced manual effort, all thanks to the power of genetic algorithms and the simplicity of the Flutter framework*

Keywords: Genetic Algorithm, Resource Scheduling, N-Queen, Timetable Automation, Cross Platform Based Timetable, Generation, Flutter Framework.

I. INTRODUCTION

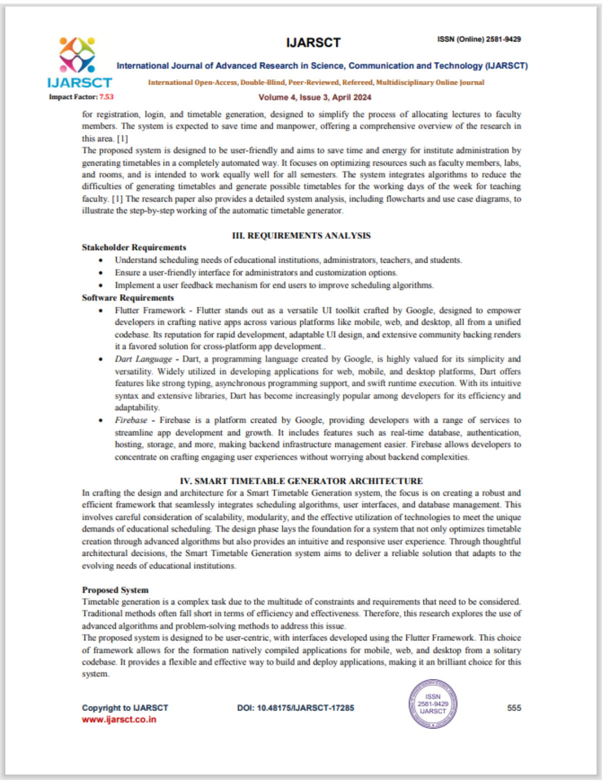
The "Smart Timetable Generator" research work offers an innovative solution to the complexities of manual timetable creation in colleges. Developed with Flutter and Dart, this scheduling tool utilizes advanced genetic algorithms to streamline the process, ensuring optimized schedules aligned with user preferences. By automating scheduling tasks and providing a user-friendly interface, the research work aims to simplify the timetable creation process, enhance resource allocation, and promote adaptability across educational institutions. With a focus on college lecture timetabling, the Smart Timetable Generator research work addresses various criteria and aims for continuous improvement. Its significance lies in transforming timetable creation efficiency within educational institutions, offering a user-friendly experience and reducing manual effort. The outcomes of this research work are expected to provide valuable insights, potentially advancing timetable management systems and related technologies in educational settings.

II. RELATED WORK

A literature survey is an analytical summary of existing literature pertinent to the proposed work. When undertaking research, a literature review serves as an essential component as it encapsulates prior investigations on the subject, providing a foundation for the current study. It holds significant importance in your report as it not only guides your research direction but also aids in defining research objectives.

"Automatic Timetable Generator" presents an automated system for generating timetables for educational institutions, aiming to address the challenges associated with manual timetable creation. The proposed system utilizes algorithms such as Genetic Algorithm, N Queen and Resource Scheduling to optimize resource utilization and overcome the limitations of manual timetable generation. The system takes inputs such as semester-wise subjects, teachers, and teacher workload to generate a feasible timetable for the working days of the week. Additionally, the paper discusses the implementation of a smart and dynamic timetable generator using software development approaches, with modules

ISSN 2581-9429 IJARSCT



for registration, login, and timetable generation, designed to simplify the process of allocating lectures to faculty members. The system is expected to save time and manpower, offering a comprehensive overview of the research in this area. [1]

The proposed system is designed to be user-friendly and aims to save time and energy for institute administration by generating timetables in a completely automated way. It focuses on optimizing resources such as faculty members, labs, and rooms, and is intended to work equally well for all semesters. The system integrates algorithms to reduce the difficulties of generating timetables and generate possible timetables for the working days of the week for teaching faculty. [1] The research paper also provides a detailed system analysis, including flowcharts and use case diagrams, to illustrate the step-by-step working of the automatic timetable generator.

III. REQUIREMENTS ANALYSIS

Stakeholder Requirements

- Understand scheduling needs of educational institutions, administrators, teachers, and students.
- Ensure a user-friendly interface for administrators and customization options.
- Implement a user feedback mechanism for end users to improve scheduling algorithms.

Software Requirements

- Flutter Framework - Flutter stands out as a versatile UI toolkit crafted by Google, designed to empower developers in crafting native apps across various platforms like mobile, web, and desktop, all from a unified codebase. Its reputation for rapid development, adaptable UI design, and extensive community backing renders it a favored solution for cross-platform app development..
- *Dart Language* - Dart, a programming language created by Google, is highly valued for its simplicity and versatility. Widely utilized in developing applications for web, mobile, and desktop platforms, Dart offers features like strong typing, asynchronous programming support, and swift runtime execution. With its intuitive syntax and extensive libraries, Dart has become increasingly popular among developers for its efficiency and adaptability.
- *Firebase* - Firebase is a platform created by Google, providing developers with a range of services to streamline app development and growth. It includes features such as real-time database, authentication, hosting, storage, and more, making backend infrastructure management easier. Firebase allows developers to concentrate on crafting engaging user experiences without worrying about backend complexities.

IV. SMART TIMETABLE GENERATOR ARCHITECTURE

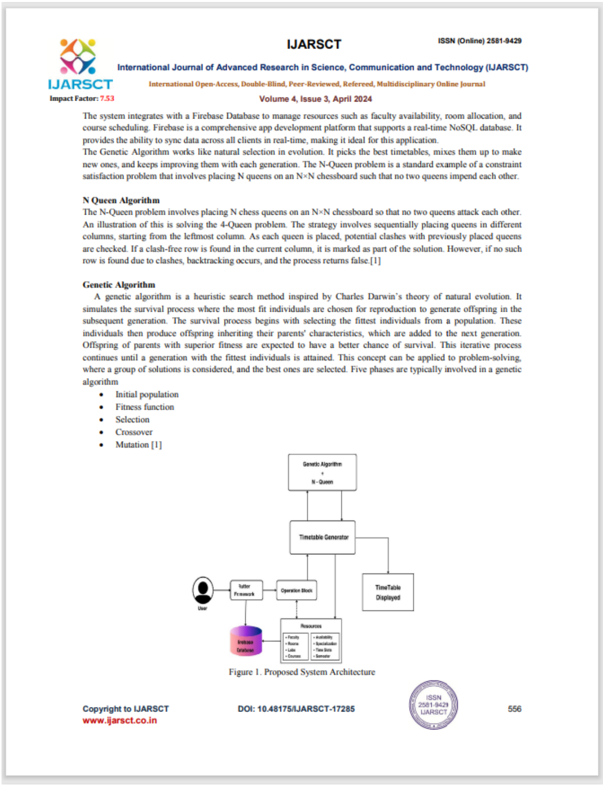
In crafting the design and architecture for a Smart Timetable Generation system, the focus is on creating a robust and efficient framework that seamlessly integrates scheduling algorithms, user interfaces, and database management. This involves careful consideration of scalability, modularity, and the effective utilization of technologies to meet the unique demands of educational scheduling. The design phase lays the foundation for a system that not only optimizes timetable creation through advanced algorithms but also provides an intuitive and responsive user experience. Through thoughtful architectural decisions, the Smart Timetable Generation system aims to deliver a reliable solution that adapts to the evolving needs of educational institutions.

Proposed System

Timetable generation is a complex task due to the multitude of constraints and requirements that need to be considered. Traditional methods often fall short in terms of efficiency and effectiveness. Therefore, this research explores the use of advanced algorithms and problem-solving methods to address this issue.

The proposed system is designed to be user-centric, with interfaces developed using the Flutter Framework. This choice of framework allows for the formation natively compiled applications for mobile, web, and desktop from a solitary codebase. It provides a flexible and effective way to build and deploy applications, making it an brilliant choice for this system.

ISSN 2581-9429 IJARSCT



The system integrates with a Firebase Database to manage resources such as faculty availability, room allocation, and course scheduling. Firebase is a comprehensive app development platform that supports a real-time NoSQL database. It provides the ability to sync data across all clients in real-time, making it ideal for this application.

The Genetic Algorithm works like natural selection in evolution. It picks the best timetables, mixes them up to make new ones, and keeps improving them with each generation. The N-Queen problem is a standard example of a constraint satisfaction problem that involves placing N queens on an N×N chessboard such that no two queens impend each other.

N Queen Algorithm

The N-Queen problem involves placing N chess queens on an N×N chessboard so that no two queens attack each other. An illustration of this is solving the 4-Queen problem. The strategy involves sequentially placing queens in different columns, starting from the leftmost column. As each queen is placed, potential clashes with previously placed queens are checked. If a clash-free row is found in the current column, it is marked as part of the solution. However, if no such row is found due to clashes, backtracking occurs, and the process returns false.[1]

Genetic Algorithm

A genetic algorithm is a heuristic search method inspired by Charles Darwin's theory of natural evolution. It simulates the survival process where the most fit individuals are chosen for reproduction to generate offspring in the subsequent generation. The survival process begins with selecting the fittest individuals from a population. These individuals then produce offspring inheriting their parents' characteristics, which are added to the next generation. Offspring of parents with superior fitness are expected to have a better chance of survival. This iterative process continues until a generation with the fittest individuals is attained. This concept can be applied to problem-solving, where a group of solutions is considered, and the best ones are selected. Five phases are typically involved in a genetic algorithm

- Initial population
- Fitness function
- Selection
- Crossover
- Mutation [1]

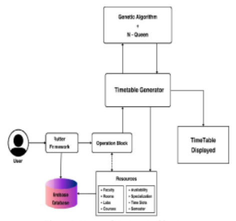
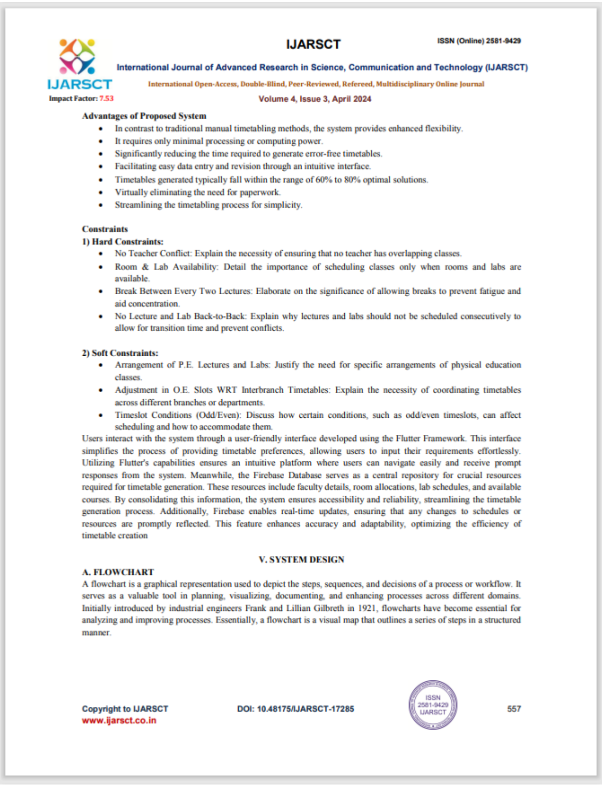


Figure 1. Proposed System Architecture

ISSN 2581-9429 IJARSCT



Advantages of Proposed System

- In contrast to traditional manual timetabling methods, the system provides enhanced flexibility.
- It requires only minimal processing or computing power.
- Significantly reducing the time required to generate error-free timetables.
- Facilitating easy data entry and revision through an intuitive interface.
- Timetables generated typically fall within the range of 60% to 80% optimal solutions.
- Virtually eliminating the need for paperwork.
- Streamlining the timetabling process for simplicity.

Constraints

1) Hard Constraints:

- No Teacher Conflict: Explain the necessity of ensuring that no teacher has overlapping classes.
- Room & Lab Availability: Detail the importance of scheduling classes only when rooms and labs are available.
- Break Between Every Two Lectures: Elaborate on the significance of allowing breaks to prevent fatigue and aid concentration.
- No Lecture and Lab Back-to-Back: Explain why lectures and labs should not be scheduled consecutively to allow for transition time and prevent conflicts.

2) Soft Constraints:

- Arrangement of P.E. Lectures and Labs: Justify the need for specific arrangements of physical education classes.
- Adjustment in O.E. Slots WRT Interbranch Timetables: Explain the necessity of coordinating timetables across different branches or departments.
- Timeslot Conditions (Odd/Even): Discuss how certain conditions, such as odd/even timeslots, can affect scheduling and how to accommodate them.

Users interact with the system through a user-friendly interface developed using the Flutter Framework. This interface simplifies the process of providing timetable preferences, allowing users to input their requirements effortlessly. Utilizing Flutter's capabilities ensures an intuitive platform where users can navigate easily and receive prompt responses from the system. Meanwhile, the Firebase Database serves as a central repository for crucial resources required for timetable generation. These resources include faculty details, room allocations, lab schedules, and available courses. By consolidating this information, the system ensures accessibility and reliability, streamlining the timetable generation process. Additionally, Firebase enables real-time updates, ensuring that any changes to schedules or resources are promptly reflected. This feature enhances accuracy and adaptability, optimizing the efficiency of timetable creation

V. SYSTEM DESIGN

A. FLOWCHART

A flowchart is a graphical representation used to depict the steps, sequences, and decisions of a process or workflow. It serves as a valuable tool in planning, visualizing, documenting, and enhancing processes across different domains. Initially introduced by industrial engineers Frank and Lillian Gilbreth in 1921, flowcharts have become essential for analyzing and improving processes. Essentially, a flowchart is a visual map that outlines a series of steps in a structured manner.

ISSN 2581-9429 IJARSCT

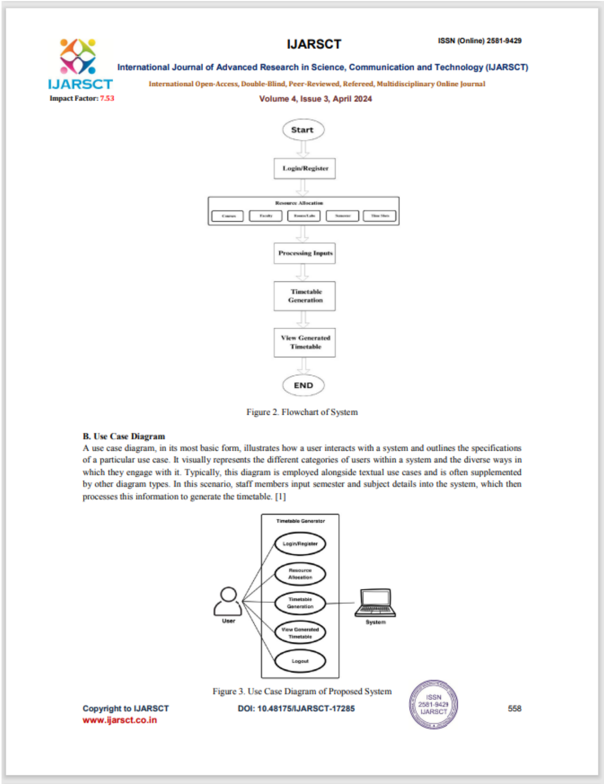
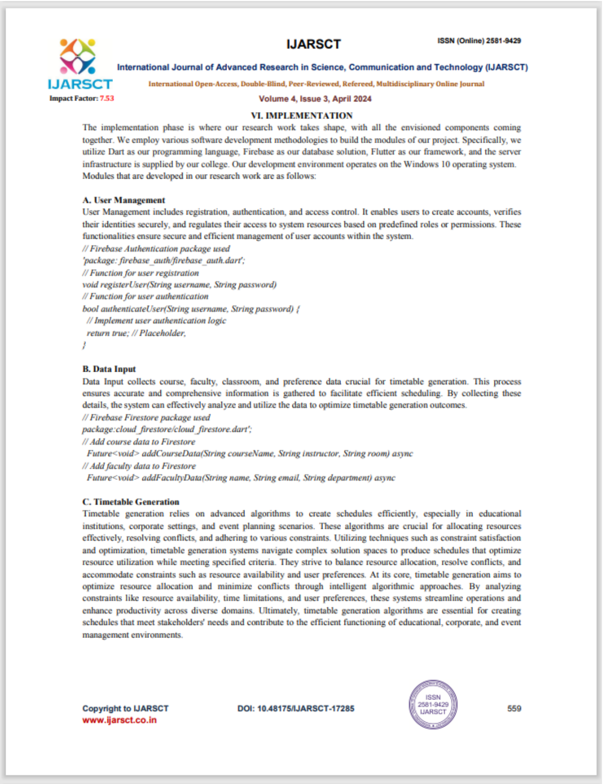

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, April 2024

Figure 2. Flowchart of System

B. Use Case Diagram

A use case diagram, in its most basic form, illustrates how a user interacts with a system and outlines the specifications of a particular use case. It visually represents the different categories of users within a system and the diverse ways in which they engage with it. Typically, this diagram is employed alongside textual use cases and is often supplemented by other diagram types. In this scenario, staff members input semester and subject details into the system, which then processes this information to generate the timetable. [1]

Figure 3. Use Case Diagram of Proposed System


IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, April 2024

VI. IMPLEMENTATION

The implementation phase is where our research work takes shape, with all the envisioned components coming together. We employ various software development methodologies to build the modules of our project. Specifically, we utilize Dart as our programming language, Firebase as our database solution, Flutter as our framework, and the server infrastructure is supplied by our college. Our development environment operates on the Windows 10 operating system. Modules that are developed in our research work are as follows:

A. User Management

User Management includes registration, authentication, and access control. It enables users to create accounts, verifies their identities securely, and regulates their access to system resources based on predefined roles or permissions. These functionalities ensure secure and efficient management of user accounts within the system.

```
// Firebase Authentication package used
'package: firebase_auth/firebase_auth.dart';
// Function for user registration
void registerUser(String username, String password)
// Function for user authentication
bool authenticateUser(String username, String password) {
  // Implement user authentication logic
  return true; // Placeholder,
}
```

B. Data Input

Data Input collects course, faculty, classroom, and preference data crucial for timetable generation. This process ensures accurate and comprehensive information is gathered to facilitate efficient scheduling. By collecting these details, the system can effectively analyze and utilize the data to optimize timetable generation outcomes.

```
// Firebase Firestore package used
package:cloud_firestore/cloud_firestore.dart';
// Add course data to Firestore
  Future<void> addCourseData(String courseName, String instructor, String room) async
// Add faculty data to Firestore
  Future<void> addFacultyData(String name, String email, String department) async
```

C. Timetable Generation

Timetable generation relies on advanced algorithms to create schedules efficiently, especially in educational institutions, corporate settings, and event planning scenarios. These algorithms are crucial for allocating resources effectively, resolving conflicts, and adhering to various constraints. Utilizing techniques such as constraint satisfaction and optimization, timetable generation systems navigate complex solution spaces to produce schedules that optimize resource utilization while meeting specified criteria. They strive to balance resource allocation, resolve conflicts, and accommodate constraints such as resource availability and user preferences. At its core, timetable generation aims to optimize resource allocation and minimize conflicts through intelligent algorithmic approaches. By analyzing constraints like resource availability, time limitations, and user preferences, these systems streamline operations and enhance productivity across diverse domains. Ultimately, timetable generation algorithms are essential for creating schedules that meet stakeholders' needs and contribute to the efficient functioning of educational, corporate, and event management environments.

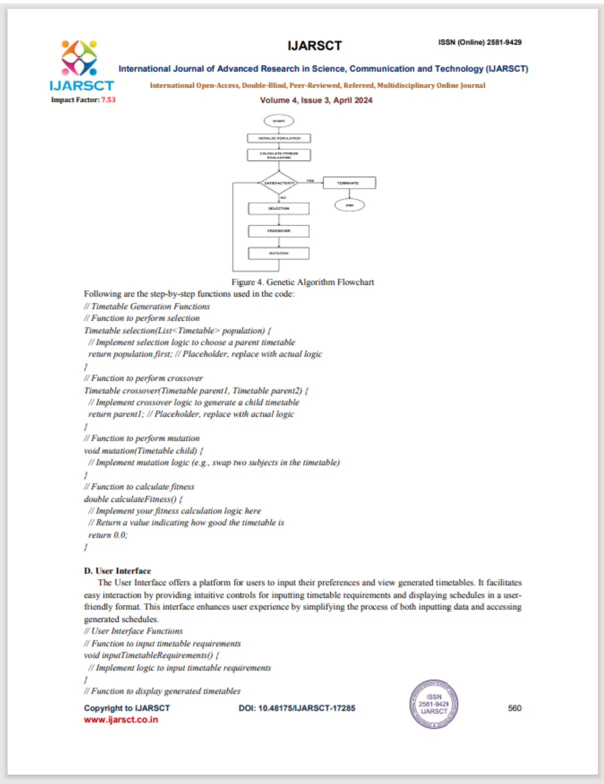

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, April 2024

Figure 4. Genetic Algorithm Flowchart

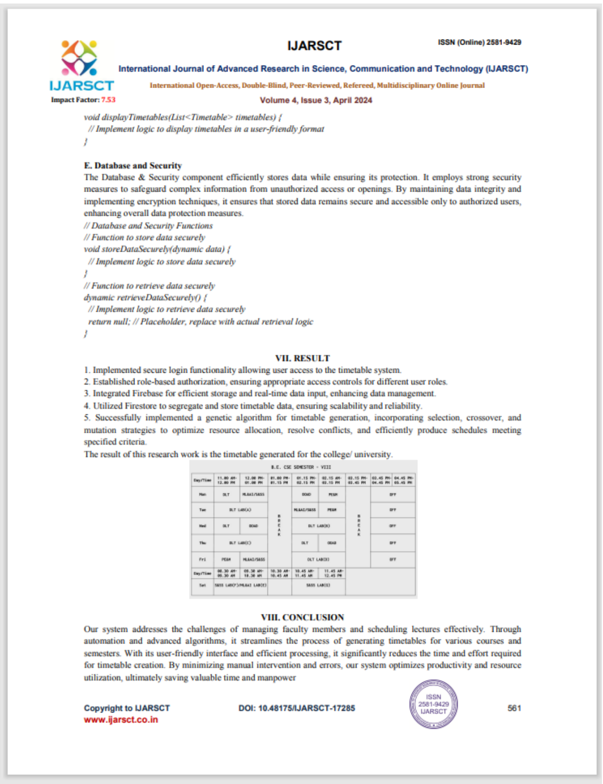
Following are the step-by-step functions used in the code:

```
// Timetable Generation Functions
// Function to perform selection
Timetable selection(List<Timetable> population) {
  // Implement selection logic to choose a parent timetable
  return population.first; // Placeholder, replace with actual logic
}
// Function to perform crossover
Timetable crossover(Timetable parent1, Timetable parent2) {
  // Implement crossover logic to generate a child timetable
  return parent1; // Placeholder, replace with actual logic
}
// Function to perform mutation
void mutation(Timetable child) {
  // Implement mutation logic (e.g., swap two subjects in the timetable)
}
// Function to calculate fitness
double calculateFitness() {
  // Implement your fitness calculation logic here
  // Return a value indicating how good the timetable is
  return 0.0;
}
```

D. User Interface

The User Interface offers a platform for users to input their preferences and view generated timetables. It facilitates easy interaction by providing intuitive controls for inputting timetable requirements and displaying schedules in a user-friendly format. This interface enhances user experience by simplifying the process of both inputting data and accessing generated schedules.

```
// User Interface Functions
// Function to input timetable requirements
void inputTimetableRequirements() {
  // Implement logic to input timetable requirements
}
// Function to display generated timetables
```


IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 3, April 2024

```
void displayTimetables(List<Timetable> timetables) {
  // Implement logic to display timetables in a user-friendly format
}
```

E. Database and Security

The Database & Security component efficiently stores data while ensuring its protection. It employs strong security measures to safeguard complex information from unauthorized access or openings. By maintaining data integrity and implementing encryption techniques, it ensures that stored data remains secure and accessible only to authorized users, enhancing overall data protection measures.

```
// Database and Security Functions
// Function to store data securely
void storeDataSecurely(dynamic data) {
  // Implement logic to store data securely
}
// Function to retrieve data securely
dynamic retrieveDataSecurely() {
  // Implement logic to retrieve data securely
  return null; // Placeholder, replace with actual retrieval logic
}
```

VII. RESULT

1. Implemented secure login functionality allowing user access to the timetable system.
2. Established role-based authorization, ensuring appropriate access controls for different user roles.
3. Integrated Firebase for efficient storage and real-time data input, enhancing data management.
4. Utilized Firestore to segregate and store timetable data, ensuring scalability and reliability.
5. Successfully implemented a genetic algorithm for timetable generation, incorporating selection, crossover, and mutation strategies to optimize resource allocation, resolve conflicts, and efficiently produce schedules meeting specified criteria.

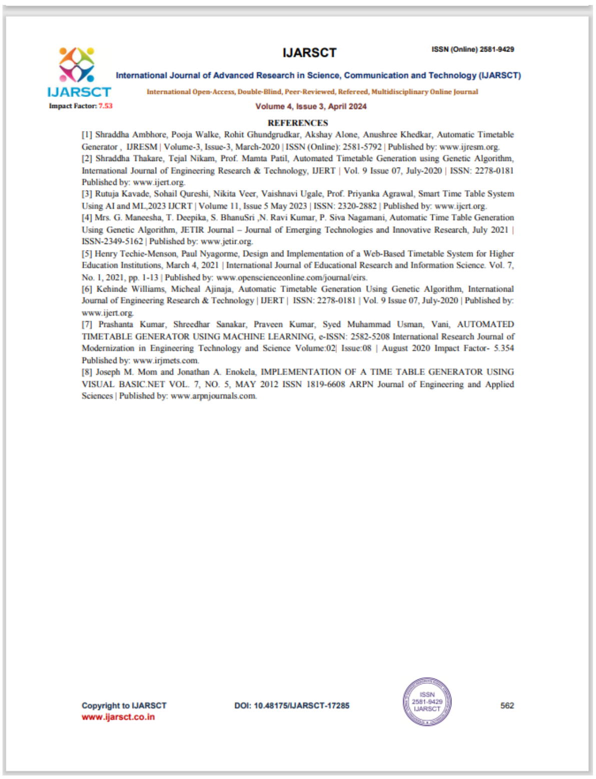
The result of this research work is the timetable generated for the college/ university.

B.E. CSE SEMESTER - VIII

Day/Time	11.00 AM-12.00 PM	12.00 PM-01.00 PM	01.00 PM-01.15 PM	01.15 PM-02.15 PM	02.15 AM-03.15 PM	03.15 PM-03.45 PM	03.45 PM-04.45 PM	04.45 PM-05.45 PM
Mon	DLT	MLAAI/SASS	BREAK	DOAD	PEAM	BREAK	OFF	
Tue	DLT LAB(A)			MLAAI/SASS	PEAM		OFF	
Wed	DLT	DOAD		DLT LAB(B)			OFF	
Thu	DLT LAB(C)			DLT	DOAD		OFF	
Fri	PEAM	MLAAI/SASS		DLT LAB(D)			OFF	
Day/Time	08.30 AM-09.30 AM	09.30 AM-10.30 AM	10.30 AM-10.45 AM	10.45 AM-11.45 AM	11.45 AM-12.45 PM			
Sat	SASS LAB(F)/MLAAI LAB(E)			SASS LAB(G)				

VIII. CONCLUSION

Our system addresses the challenges of managing faculty members and scheduling lectures effectively. Through automation and advanced algorithms, it streamlines the process of generating timetables for various courses and semesters. With its user-friendly interface and efficient processing, it significantly reduces the time and effort required for timetable creation. By minimizing manual intervention and errors, our system optimizes productivity and resource utilization, ultimately saving valuable time and manpower



IJARSCT ISSN (Online) 2581-9429

IJARSCT
Impact Factor: 7.53

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Volume 4, Issue 3, April 2024

REFERENCES

[1] Shraddha Ambhore, Pooja Walke, Rohit Ghundgrudkar, Akshay Alone, Anushree Khedkar, Automatic Timetable Generator , IJRESM | Volume-3, Issue-3, March-2020 | ISSN (Online): 2581-5792 | Published by: www.ijresm.org.

[2] Shraddha Thakare, Tejal Nikam, Prof. Mamta Patil, Automated Timetable Generation using Genetic Algorithm, International Journal of Engineering Research & Technology, IJERT | Vol. 9 Issue 07, July-2020 | ISSN: 2278-0181 Published by: www.ijert.org.

[3] Rutuja Kavade, Sohail Qureshi, Nikita Veer, Vaishnavi Ugale, Prof. Priyanka Agrawal, Smart Time Table System Using AI and ML,2023 IJCRT | Volume 11, Issue 5 May 2023 | ISSN: 2320-2882 | Published by: www.ijcrt.org.

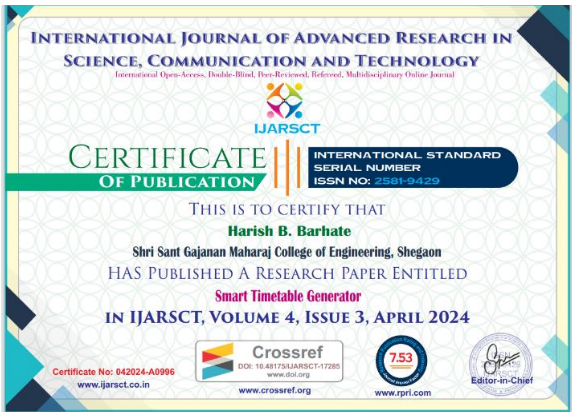
[4] Mrs. G. Maneesha, T. Deepika, S. BhanuSri ,N. Ravi Kumar, P. Siva Nagamani, Automatic Time Table Generation Using Genetic Algorithm, JETIR Journal – Journal of Emerging Technologies and Innovative Research, July 2021 | ISSN-2349-5162 | Published by: www.jetir.org.

[5] Henry Techie-Menson, Paul Nyagorme, Design and Implementation of a Web-Based Timetable System for Higher Education Institutions, March 4, 2021 | International Journal of Educational Research and Information Science. Vol. 7, No. 1, 2021, pp. 1-13 | Published by: www.openscienceonline.com/journal/eirs.

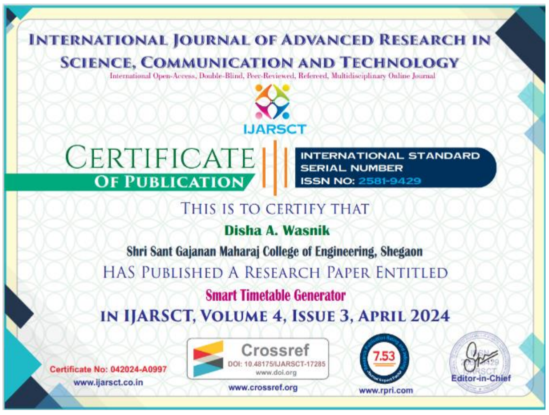
[6] Kehinde Williams, Micheal Ajinaja, Automatic Timetable Generation Using Genetic Algorithm, International Journal of Engineering Research & Technology | IJERT | ISSN: 2278-0181 | Vol. 9 Issue 07, July-2020 | Published by: www.ijert.org.

[7] Prashanta Kumar, Shreedhar Sanakar, Praveen Kumar, Syed Muhammad Usman, Vani, AUTOMATED TIMETABLE GENERATOR USING MACHINE LEARNING, e-ISSN: 2582-5208 International Research Journal of Modernization in Engineering Technology and Science Volume:02| Issue:08 | August 2020 Impact Factor- 5.354 Published by: www.irjmets.com.

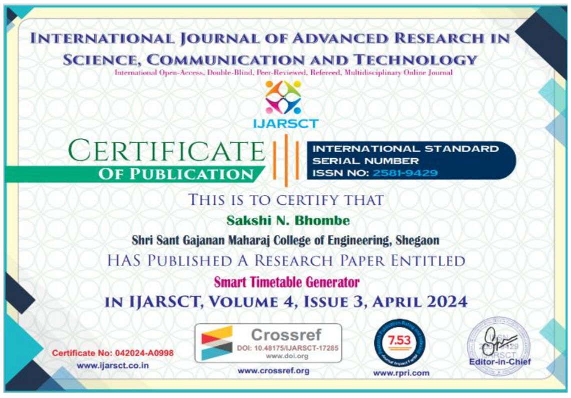
[8] Joseph M. Mom and Jonathan A. Enokela, IMPLEMENTATION OF A TIME TABLE GENERATOR USING VISUAL BASIC.NET VOL. 7, NO. 5, MAY 2012 ISSN 1819-6608 ARPN Journal of Engineering and Applied Sciences | Published by: www.arpnjournals.com.



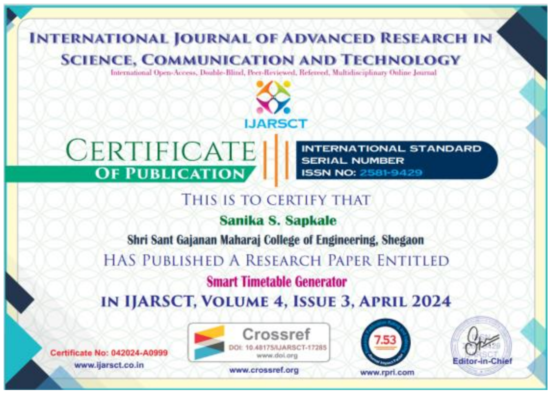
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
IJARSCT
CERTIFICATE OF PUBLICATION
INTERNATIONAL STANDARD SERIAL NUMBER ISSN NO: 2581-9429
THIS IS TO CERTIFY THAT
Harish B. Barhate
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
HAS PUBLISHED A RESEARCH PAPER ENTITLED
Smart Timetable Generator
IN IJARSCT, VOLUME 4, ISSUE 3, APRIL 2024
Certificate No: 042024-A0996
www.ijarsct.co.in
Crossref DOI: 10.48175/IJARSCT-17285 www.doi.org
www.crossref.org
7.53
www.rpri.com
Editor-in-Chief



International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
IJARSCT
CERTIFICATE OF PUBLICATION
INTERNATIONAL STANDARD SERIAL NUMBER ISSN NO: 2581-9429
THIS IS TO CERTIFY THAT
Disha A. Wasnik
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
HAS PUBLISHED A RESEARCH PAPER ENTITLED
Smart Timetable Generator
IN IJARSCT, VOLUME 4, ISSUE 3, APRIL 2024
Certificate No: 042024-A0997
www.ijarsct.co.in
Crossref DOI: 10.48175/IJARSCT-17285 www.doi.org
www.crossref.org
7.53
www.rpri.com
Editor-in-Chief

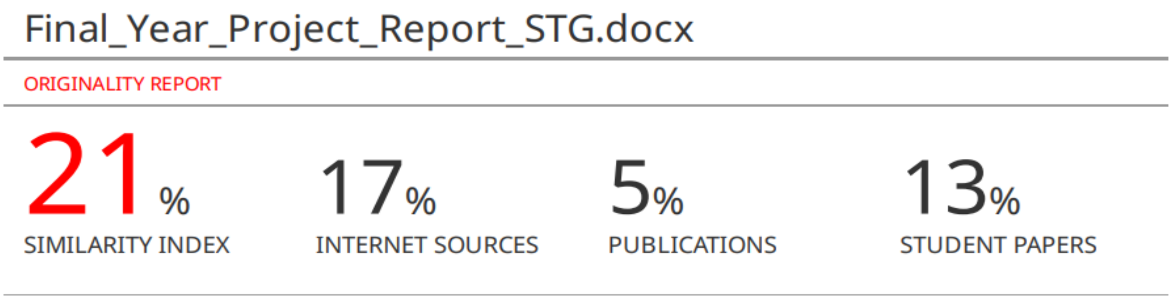


International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
IJARSCT
CERTIFICATE OF PUBLICATION
INTERNATIONAL STANDARD SERIAL NUMBER ISSN NO: 2581-9429
THIS IS TO CERTIFY THAT
Sakshi N. Bhombe
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
HAS PUBLISHED A RESEARCH PAPER ENTITLED
Smart Timetable Generator
IN IJARSCT, VOLUME 4, ISSUE 3, APRIL 2024
Certificate No: 042024-A0998
www.ijarsct.co.in
Crossref DOI: 10.48175/IJARSCT-17285 www.doi.org
www.crossref.org
7.53
www.rpri.com
Editor-in-Chief



International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
IJARSCT
CERTIFICATE OF PUBLICATION
INTERNATIONAL STANDARD SERIAL NUMBER ISSN NO: 2581-9429
THIS IS TO CERTIFY THAT
Sanika S. Sapkale
Shri Sant Gajanan Maharaj College of Engineering, Shegaon
HAS PUBLISHED A RESEARCH PAPER ENTITLED
Smart Timetable Generator
IN IJARSCT, VOLUME 4, ISSUE 3, APRIL 2024
Certificate No: 042024-A0999
www.ijarsct.co.in
Crossref DOI: 10.48175/IJARSCT-17285 www.doi.org
www.crossref.org
7.53
www.rpri.com
Editor-in-Chief

PLAGIARISM REPORT

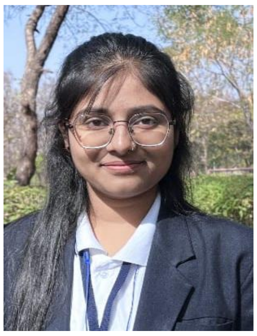


Final_Year_Project_Report_STG.docx

ORIGINALITY REPORT

21%	17%	5%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PROJECT GROUP DETAILS

	Name: Disha Abhaykumar Wasnik **Address: Dhadiwal Layout, Suyog Nagar, Nagpur – 440027** **Mobile No: 9511750862** **Email ID: dishawasnik02@gmail.com**
	Name: Harish Bhagwan Barhate **Address: Diwan Layout, Manewada, Nagpur - 440034** **Mobile No: 9637948416** **Email ID: harishbarhate29@gmail.com**
	Name: Sakshi Nandu Bhombe **Address: Renuka Nagar, Shri Ram Chowk, Dabki Road, Akola – 444001** **Mobile No: 8378817915** **Email ID: sakshibhombe2002@gmail.com**
	**Name: Sanika Sudhir Sapkale** **Address: Bansi Nagar, Hingna Road, Nagpur - 440016** **Mobile No: 9657844610** **Email ID: sanikasapkale20@gmail.com**