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

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

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.

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

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]

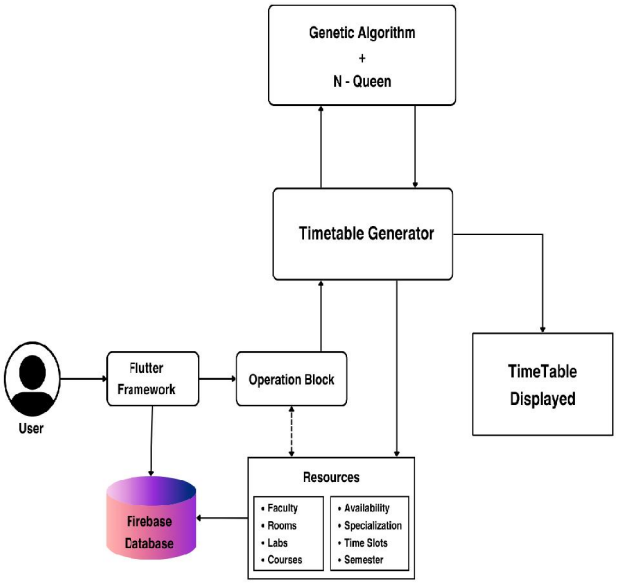

Figure 1. Proposed System Architecture

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

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.

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

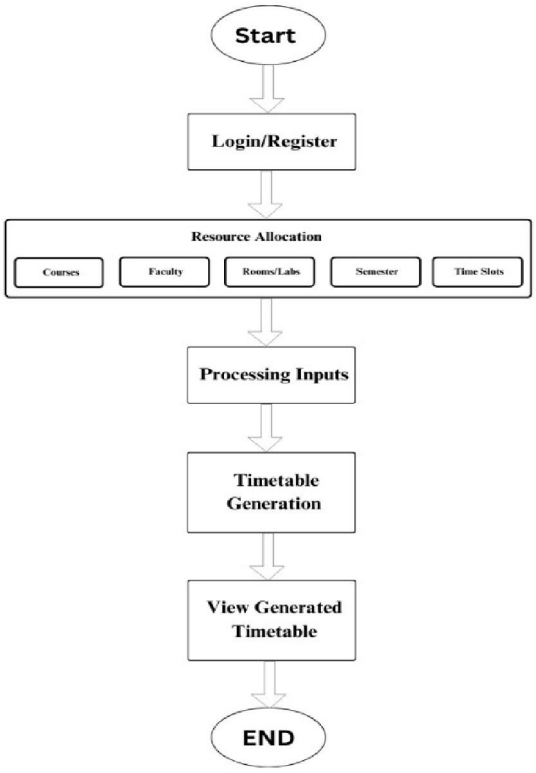

Figure 2. Flowchart of System

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]

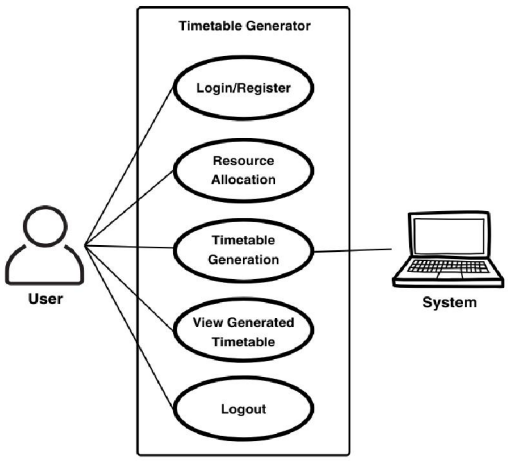

Figure 3. Use Case Diagram of Proposed System

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

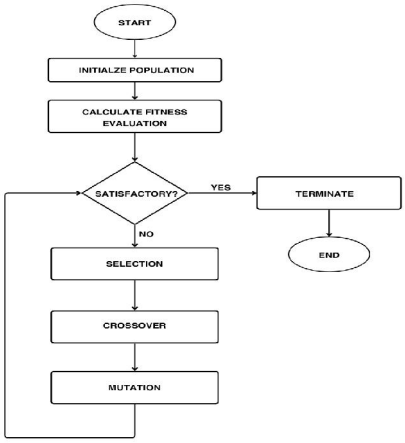

Figure 4. Genetic Algorithm Flowchart

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	ML&AI/S&SS	BREAK	OOAD	PE&M	BREAK	OFF	
Tue	DLT LAB(A)			ML&AI/S&SS	PE&M		OFF	
Wed	DLT	OOAD		DLT LAB(B)			OFF	
Thu	DLT LAB(C)			DLT	OOAD		OFF	
Fri	PE&M	ML&AI/S&SS		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	S&SS LAB(F)/ML&AI LAB(E)			S&SS 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
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

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.