A Major Project Report on

AUTOMATIC TIMETABLE GENERATION

Submitted in partial fulfilment of the requirements for the award of

the degree Of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

P.NITISH GOUD **20EG105137**

Under the guidance of

Dr. G. PRABHAKAR RAJU

Assistant Professor Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Venkatapur(V), Ghatkeshar(M), Medchal(D) – 500088

2020-2024

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the report / dissertation entitled “**AUTOMATIC TIMETABLE GENERATION”** that is being submitted by **P.NITISH GOUD [20EG105137]** in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering to the Anurag University, Hyderabad is a record of bonafide work carried out by him under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Internal Guide
Dr. G. Prabhakar Raju
Assistant Professor, Dept. of CSE

Dr. G. Vishnu Murthy
Professor &Dean, Dept. of CSE

External Examiner

ACKNOWLEDGMENT

I would like to express our sincere thanks and deep sense of gratitude to project supervisor **Dr. G. Prabhakar Raju** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

I would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. I also express our deep sense of gratitude to **Dr. V V S S S Balaram,** Academic coordinator, **Dr. Pallam Ravi**, Project in-Charge, **Dr. G. Prabhakar Raju** Project Coordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage our project work.

I would like express our special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for his encouragement and timely support in our B.
Tech program.

P. NITISH GOUD

(20EG105137)

DECLARATION

I, hereby declare that the Report entitled **“AUTOMATIC TIMETABLE GENERATION”** submitted for the award of Bachelor of technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

P.NITISH GOUD
(20EG105137)

ABSTRACT

In response to the complex scheduling demands of educational institutions, this project proposes the development of a sophisticated computer program leveraging genetic algorithms for the automatic generation of optimized timetables. The application of genetic algorithms, inspired by principles of natural selection and genetics, offers a novel approach to tackling the intricate task of timetable creation by iteratively evolving solutions that satisfy diverse constraints and objectives. The proposed program operates by soliciting user inputs, including but not limited to the number of courses, classrooms, teachers, and available timeslots. Utilizing this information, the program initiates a genetic algorithm-driven process to produce a diverse population of potential timetable solutions. Through successive iterations of selection, crossover, and mutation operations, the program refines these solutions, continually evaluating their fitness based on predefined criteria until an optimal timetable configuration is reached. The overarching objective of this project is twofold: to introduce an efficient and automated system for timetable generation that alleviates the time and effort burden on educational institutions, and to enhance the quality of timetables generated by considering a multitude of factors such as resource availability, conflict resolution, and equitable distribution of courses. By employing genetic algorithms, this project aims to capitalize on several inherent advantages. These include the ability to navigate expansive search spaces effectively, adeptly manage complex constraints, and generate timetables that are both diverse and adaptable to changing circumstances. Furthermore, the genetic algorithm framework offers a robust solution for optimizing scheduling processes, ensuring that the resultant timetables align closely with the operational needs and preferences of educational institutions. Through the integration of advanced optimization techniques within the realm of timetable generation, this project endeavors to provide a comprehensive and scalable solution that addresses the evolving scheduling challenges faced by educational institutions. By

harnessing the power of genetic algorithms, this endeavor represents a significant step towards enhancing efficiency, effectiveness, and overall satisfaction in the realm of educational scheduling.

Table Of Contents

Table Of Figures

List Of Tables

1. INTRODUCTION

Automated Planning utilizing genetic algorithms heralds a transformative approach to enhancing planning processes, particularly within educational institutions. This innovative computational tool harnesses the potency of genetic algorithms to craft streamlined programs that align with specific criteria and objectives. Leveraging heuristic optimization principles inspired by natural selection and genetics, this methodology enables iterative refinement of schedules, class allocations, teacher availability, and other pertinent factors to yield optimal solutions amidst evolving constraints.

This article elucidates the methodologies and advantages inherent in employing genetic algorithms for planning purposes, illuminating its potential to revolutionize school management practices. Within educational management, the judicious allocation of resources and time is paramount to fostering effectiveness and efficiency. However, conventional manual scheduling processes often entail labour intensive workflows fraught with errors and inefficiencies. In response, automatic job scheduling utilizing genetic algorithms emerges as a promising avenue poised to supplant traditional syllabus planning methodologies.

Automatic planning facilitated by genetic algorithms represents a paradigm shift in strategic planning, leveraging evolutionary principles to iteratively refine project parameters and enhance resource utilization. This decision-making framework is engineered to heighten project efficiency and optimize resource allocation by preemptively crafting plans that align with predetermined criteria and objectives.

This research article endeavors to delve into the methodologies and outcomes of automatic planning leveraging genetic algorithms, particularly within the context of curriculum development for educational institutions. By elucidating the intricacies of this approach, the aim is to underscore its adaptability in optimizing planning processes and elevating overall project efficacy. Moreover, this article seeks to articulate the rationale behind automatic scheduling generation employing genetic algorithms, emphasizing its facets such as efficiency, quality assurance, flexibility, and scalability.

Through comprehensive evaluation of these objectives, this study endeavors to underscore the myriad benefits and practical applications of this innovative technology.

Additionally, it offers a nuanced exploration of the criticality of selecting pertinent data sets for successful planning utilizing genetic algorithms. By delineating inputs, constraints, and outputs, the article underscores their pivotal role in facilitating the formulation of robust plans.

In essence, this study advocates for the widespread adoption of automatic planning via genetic algorithms as a viable alternative to conventional scheduling practices within educational settings. By harnessing the potential of this pioneering technology and leveraging appropriate data, institutions can unlock novel avenues for enhanced operational efficiency and efficacy across their programs.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

The university's manual scheduling system is exceedingly tedious and timeconsuming, often resulting in the overlap of classes taught by the same instructor or clashes between multiple classes. Many universities offer numerous courses spanning various subjects, with a limited number of instructors assigned to each department. The current manual approach to scheduling is inefficient due to its lack of automation. To address these challenges, an automated system equipped with a schedule generator can be devised. This system would take into consideration several parameters, including the number of subjects, available teachers, maximum lectures per teacher, and the priority of subjects and topics to be covered. By analysing these inputs, the automated system would generate viable timetables for each working day of the week, effectively optimizing the allocation of resources within the specified constraints.

In the past, timetable generators utilized various techniques to create schedules for educational institutions. Initially, manual methods were employed, which proved to be time-consuming and error-prone. These manual approaches involved assigning teachers and classes to specific time slots and iteratively refining schedules. With advancements in computer technology, a variety of algorithms have been developed to automate the scheduling process. These algorithms encompass brute force, simulated annealing, and genetic algorithms. Nonetheless, older algorithms often struggled with complex constraints or lacked flexibility in creating schedules. Recently, hybrid algorithms have gained popularity, combining multiple optimization techniques such as genetic algorithms, taboo search, and simulated annealing.

The earlier timetable generators also suffered from limited user interfaces, requiring users to possess a strong understanding of the algorithm and timetable constraints. This posed a challenge for users lacking a technical background, hindering their effective utilization of these systems. Over time, timetable generators have evolved into more efficient and user-friendly systems. They now possess enhanced capabilities to handle intricate constraints and generate tailored timetables for diverse educational institutions, marking a significant advancement from their earlier counterparts.

2.2 RELATED WORK

- In a paper authored by E.K. Burke and B. McCollum, titled "An Evolutionary Algorithmic Approach to University Course Timetabling" *[2],* a method employing genetic algorithms is presented to address the university course timetabling problem. The proposed algorithm integrates local search techniques and crossover operators to produce optimal timetables.

- F. Firdausi et al. discuss their genetic algorithm-based solution for the school timetabling problem in their paper titled "Genetic Algorithms Applied to SchooL Timetabling". The algorithm utilizes a blend of crossover and mutation operators to generate optimized and feasible timetables.

- M. R. A. Rahman et al. conduct a comprehensive examination of genetic algorithmbased methodologies for university course timetabling in their paper "Exploring Genetic Algorithms for University Course Timetabling" *[5]*. This review critically evaluates various genetic algorithm-based approaches, identifying their strengths, weaknesses, and areas for future research.

- In "Optimizing Nurse Rostering Using Genetic Algorithms" *[6]* authored by E.K. Burke et al., a genetic algorithm framework is proposed to solve the nurse rostering problem. The algorithm integrates crossover, mutation, and local search operators to generate high-quality nurse rosters.

S. No	**Author(s)**	**Method**	**Advantages**	**Disadvantages**
1	Esraa A. Abdelhalim and Ghada A.El Khayat	Population Generation, Fitness Function, Selection, Crossover, Mutation, Elitism, Termination.	1.Maximizes resource utilization for efficient scheduling. 2.Effectively manages various timetabling constraints.	1.High computational requirements, especially for large datasets. 2.Effectiveness depends on tuning

				crossover and mutation rates
2	Meysam Shahvali Kohshori and Mohammad Saniee Abadeh	Problem Encoding, Initial Population, Fitness Function, Crossover, Mutation, HybridFeatures, Termination.	1.Hybrid features accelerate convergence and enhance solution quality. 2.Diverse crossover and mutation operators explore varied solution spaces	1.GAs can be computation ally expensive, especially for complex constraints. 2.Effectiven ess depends on tuning parameters, which can be timeconsuming.

3	Mrunmayee V. Rane;
	Vikram M. Apte;
	Vishakha N. Nerkar; Mani Roja Edinburgh;
	K.Y. Rajput
		Input Module, Operational Module, Display Module.	1.System ensures efficient resource allocation with more optimized and accurate timetables.
			2. Saves administrators significant time, allowing focus on more productive tasks.
				1.Algorithm complexity (cube root of n) may challenge computation al resources for larger datasets or real-time adjustments .
				2.User unfamiliarit y with the system may lead to a learning curve, potentially causing initial resistance or

				errors during the transition.

Table 2.2 Related work

3. PROPOSED METHOD

3.1 PROBLEM IDENTIFICATION

Automatic timetable generation using genetic algorithm is a project aimed at developing a computer program that can generate an optimized timetable automatically for educational institutions. The program uses genetic algorithms, which are a type of heuristic optimization algorithm based on the principles of natural selection and genetics, to generate a timetable that satisfies certain constraints and objectives. The program takes input from the user, including the number of courses, the number of classrooms, the number of teachers, and the timeslots available for scheduling. It then uses the genetic algorithm to generate a population of candidate solutions, which are evaluated based on their fitness, i.e., how well they satisfy the constraints and objectives. The genetic algorithm then applies selection, crossover, and mutation operations to generate new candidate solutions, which are evaluated and ranked again. This process continues until a satisfactory solution is found, which is then output as the final timetable.

The objective of the project is to create an efficient and automated system for timetable generation, which can save time and effort for educational institutions. The project also aims to improve the quality of timetables generated, by considering multiple factors such as the availability of resources, avoiding conflicts, and ensuring fairness in course scheduling. The benefits of using genetic algorithms for timetable generation include the ability to find optimal solutions in a large search space, the ability to handle complex constraints, and the ability to generate diverse and flexible timetables.

Overall, the project aims to provide an effective and efficient solution for automatic timetable generation using genetic algorithms.

3.2 PROPOSED SYSTEM

The proposed system for Automatic timetable Generation utilizes genetic algorithms to efficiently generate optimal timetables for educational institutions. It features a userfriendly interface for inputting constraints such as teacher availability, course prerequisites, and classroom availability. The genetic algorithm creates a population of possible timetables, employing crossover and mutation operations to produce new generations. A fitness function evaluates each timetable based on constraints and objectives, selecting the fittest for the next generation. Users can visualize, evaluate, and refine schedules, and can adjust limits as needed. The system handles complex constraints, offers flexibility, and is designed for ease of use, saving time and effort for educational institutions.

3.2.1 Proposed Method Illustration

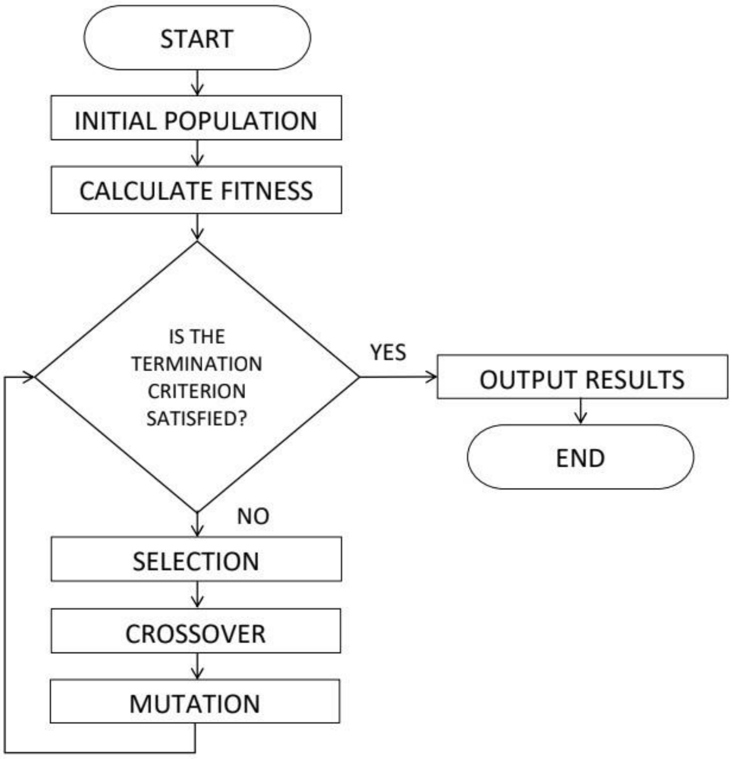


Fig 3.1 : Proposed Method Illustration

Initialization:

In the initialization phase, the process begins by generating an initial population of chromosomes randomly. Each chromosome serves as a blueprint for a potential configuration, embodying various permutations and combinations of class schedules. Following this, a meticulous evaluation ensues, assessing the fitness of each chromosome against predefined constraints and preferences. This evaluation process serves as a groundwork for subsequent genetic operations.

Genetic Operations

a. Elitism:

As a foundational genetic operation, elitism plays a pivotal role in preserving the most robust chromosomes from the previous generation. These elite chromosomes represent the pinnacle of fitness within the population and serve as a cornerstone for maintaining desirable traits across subsequent generations.

b. Selection:

Employing the Roulette Wheel Selection method, chromosome pairs are chosen from the previous generation in a randomized fashion. This selection process introduces an element of genetic diversity, facilitating the exploration of different genetic combinations and pathways. By incorporating randomness, the selection mechanism ensures that all chromosomes have an equal opportunity to contribute to the evolution of the population.

c. Crossover:

The crossover operation is executed through the application of Single Point Crossover. This intricate process involves the exchange of genetic information between selected chromosome pairs, fostering genetic recombination and diversification. Careful consideration is given to ensure that the crossover points do not intersect student group timetables, thus preserving the integrity of class schedules and minimizing disruptions.

d. Mutation:

Mutation serves as a mechanism for introducing controlled alterations to the genetic makeup of chromosomes. Through the application of Swap Mutation, minor changes are made to the genetic code of the more fit chromosome. These subtle modifications inject variability into the population, preventing premature convergence and facilitating the exploration of novel solutions.

Fitness Evaluation of New Generation:

With the advent of a new generation, a comprehensive fitness evaluation ensues. This process entails a meticulous analysis of the newly created chromosomes, scrutinizing their ability to meet the objectives and constraints set forth for the current generation. Through rigorous assessment, chromosomes are assigned fitness scores reflective of their efficacy in addressing the specified criteria.

Sorting and Selection:

The sorting and selection phase entails the meticulous ordering of chromosomes based on their respective fitness levels. Chromosomes are arranged in descending order of fitness, with the most robust solutions occupying precedence. This iterative process continues until chromosomes meeting the desired fitness threshold, such as a fitness value of 1, are obtained, signifying the attainment of optimized solutions.

Output:

The culmination of the automated timetable generation process yields the selection of the final chromosome, representing the optimized timetable configuration. This meticulously curated solution embodies the synthesis of genetic operations, reflecting a harmonious balance between hard constraints and soft preferences. The output encapsulates the culmination of extensive computational efforts and embodies a refined timetable solution tailored to the specific needs of the educational institution.

Testing and Console Output:

For validation and diagnostic purposes, comprehensive console output is generated, providing detailed insights into the timetable generation process. This includes pertinent information such as input data, generated time slots, chromosome configurations, corresponding fitness values, and ultimately, the final selected chromosome. By facilitating transparency and traceability, console output serves as a crucial tool for assessing the efficacy and robustness of the implemented genetic algorithm.

Conclusion and Further Work:

In conclusion, the automated timetable generation process driven by Genetic Algorithms showcases immense promise in revolutionizing scheduling practices within educational institutions. Through meticulous experimentation and evaluation, the effectiveness of Genetic Algorithms in obtaining superior timetable solutions is affirmed. Looking ahead, future research endeavors may explore the integration of additional soft constraints, such as faculty preferences, classroom capacities, lab facilities, and considerations for faculty managing multiple subject selections. By expanding the scope of optimization, these enhancements aim to further refine the timetable generation process, fostering greater efficiency and adaptability in educational scheduling practices.

3.2.2 Parameters

The parameter formulas can be derived from the explanation of the Genetic Algorithm (GA) process for Time Table Scheduling. Here are some of the key parameters mentioned:

1. Initialization:

- Population Size (N): Number of chromosomes in the initial population.

2. Genetic Operations:

a. Elitism:

- Elitism Ratio (ER): Percentage of fittest chromosomes preserved from the previous generation.

b. Selection:

- Selection Probability (SP): Probability of selecting a chromosome based on its fitness during the Roulette Wheel Selection.

c. Crossover:

- Crossover Rate (CR): Probability of crossover occurring between two selected chromosomes.
- Crossover Point (CP): Randomly chosen point for the Single Point Crossover, ensuring it doesn't intersect student group timetables.

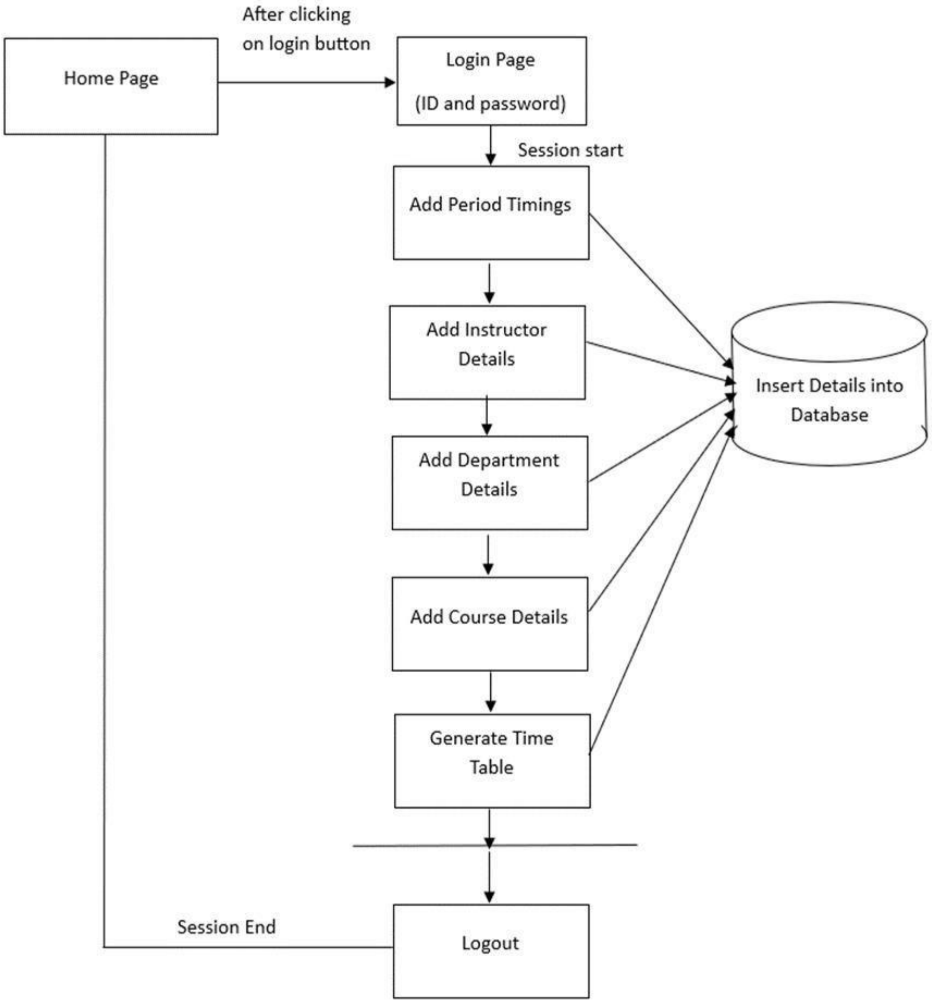
d. Mutation:

- Mutation Rate (MR): Probability of mutation occurring on the more fit chromosome.
- Swap Mutation: Specific technique involving the exchange of two portions of the chromosome.

3. Sorting and Selection:

- There are no specific parameters, but sorting involves ordering the chromosomes in a specific way according to their fitness of each chromosome and some algorithms like Roulette wheel selection for selecting chromosomes for the next Generation.

3.3 ARCHITECTURE

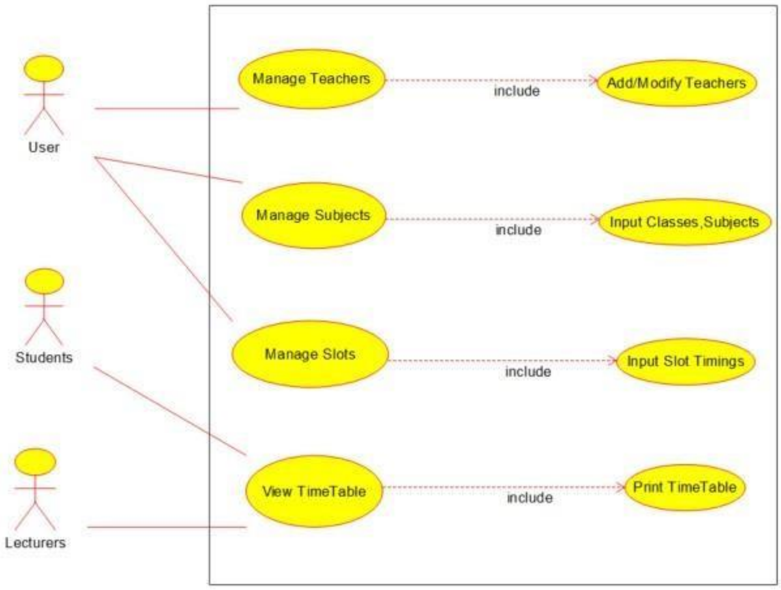


3.3.1 Architecture Diagram

The project's architecture is structured as follows: Users initially log in via the home page, proceeding to input period timings, instructor details, department information, and course specifics. All entered data, alongside user details, are stored within the database. Subsequently, the system generates a comprehensive timetable. Users have the option to view this timetable before logging out, ensuring accessibility and convenience. This streamlined process enhances user experience and facilitates efficient management of scheduling and resources. With its user-centric design and database integration, the system offers a seamless and effective solution for timetable generation and management.

3.4 UML DIAGRAMS

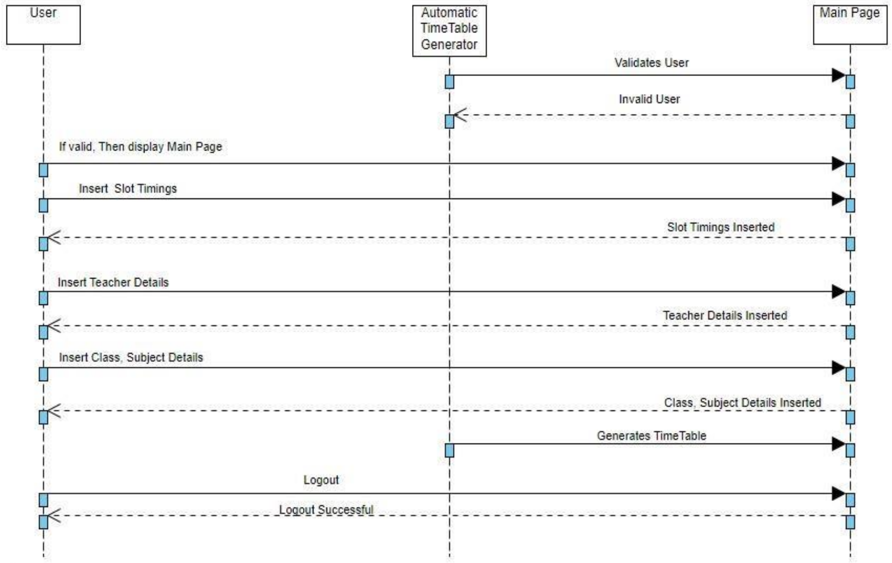
3.4.1 Use Case Diagram



3.4.1 Automatic Time Table Generator Use Case Diagram

Use case diagram of Automatic timetable generator includes Manage Teachers, Add/Modify Teachers, Manage Subjects, Input Classes and Subjects, Manage Slots, Input Slot Timings, View TimeTable, Print TimeTable use cases. It includes User, Students, Lecturers actors

3.4.2 Sequence Diagram



3.4.2 Automatic Time Table Generator Sequence Diagram

Sequence Diagram of this system includes User, Automatic time table generator, Main Page Components. Messages between the components indicates the actions such as validating the user and inserting the required details along with user session management.

3.4.3 Class Diagram

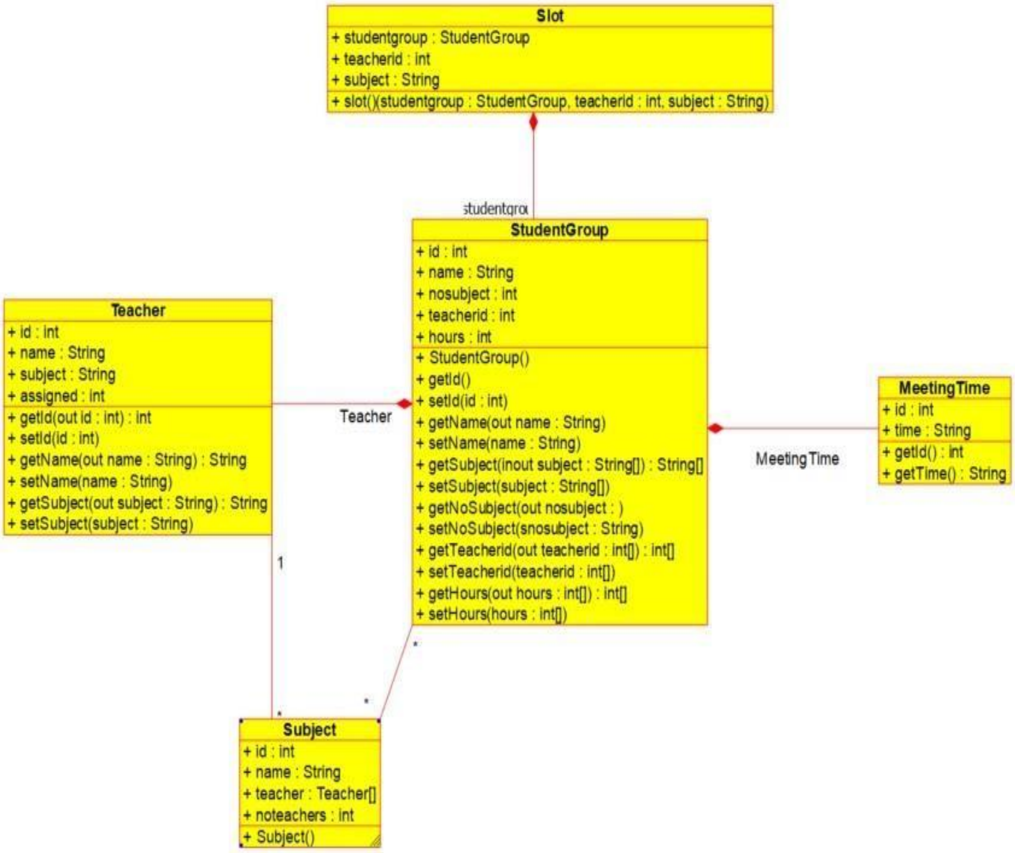
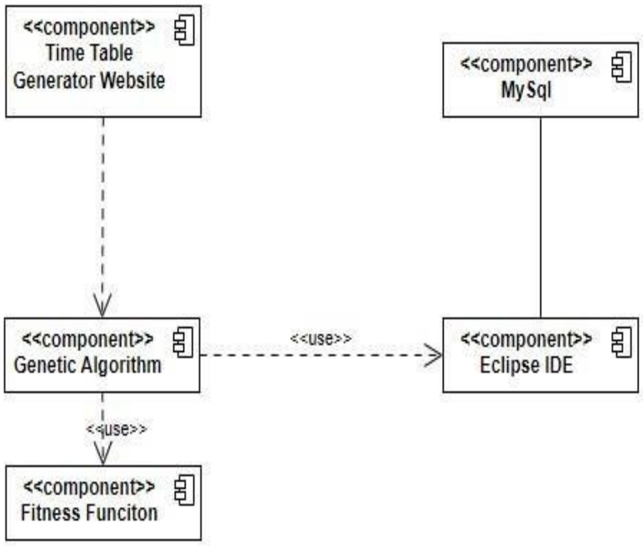

Slot
+ studentgroup : StudentGroup
+ teacherid : int
+ subject : String
+ slot()(studentgroup : StudentGroup, teacherid : int, subject : String)
studentgrou
StudentGroup
+ id : int
+ name : String
+ nosubject : int
+ teacherid : int
+ hours : int
+ StudentGroup()
+ getId()
+ setId(id : int)
+ getName(out name : String)
+ setName(name : String)
+ getSubject(inout subject : String[]) : String[]
+ setSubject(subject : String[])
+ getNoSubject(out nosubject :)
+ setNoSubject(snosubject : String)
+ getTeacherid(out teacherid : int[]) : int[]
+ setTeacherid(teacherid : int[])
+ getHours(out hours : int[]) : int[]
+ setHours(hours : int[])
Teacher
+ id : int
+ name : String
+ subject : String
+ assigned : int
+ getId(out id : int) : int
+ setId(id : int)
+ getName(out name : String) : String
+ setName(name : String)
+ getSubject(out subject : String) : String
+ setSubject(subject : String)
Teacher
MeetingTime
MeetingTime
+ id : int
+ time : String
+ getId() : int
+ getTime() : String
1
*
*
*
Subject
+ id : int
+ name : String
+ teacher : Teacher[]
+ noteachers : int
+ Subject()

Fig 3.4.3 Automatic Time Table Generator Class Diagram

Class diagram for an automatic timetable generator includes class components of Slot, Teacher, StudentGroup, Subject, MeetingTime. In each components their attributes and operations are specified accordingly. Relationships between them are specified as mentioned in the above diagram.

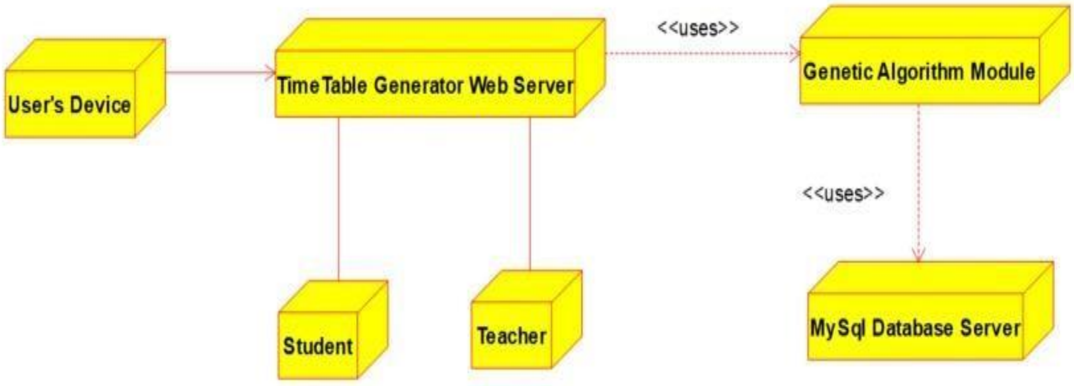
3.4.4 Component Diagram



3.4.4 Automatic Time Table Generator Component Diagram

Component diagram of Automatic timetable generator includes TimeTable generator website, execution environment, algorithm used as components. Website is dependent on algorithm to generate time table. Algorithm uses IDE to get executed and generate time tables.

3.4.5 Deployment Diagram



3.4.5 Automatic Time Table Generator Deployment Diagram

Deployment diagram of Automatic timetable generator includes User's device, Timetable generator web server, Genetic algorithm module, Student, Teacher, MySql Database server as Nodes and relationships between them as depicted in the above figure.

3.5 FEASIBILITY STUDY

The feasibility study is an important phase in software development. It helps to determine whether a proposed system is practical and viable. In the case of an Automatic Timetable Generator using Genetic Algorithm, the feasibility study evaluates the practicality and viability of developing such a system. This study involves the assessment of technical, economic, operational, and scheduling feasibility

3.5.1 Technical Feasibility

Technical feasibility is the evaluation of whether the proposed system can be developed using the current technology and infrastructure. The Automatic Time Table Generator using Genetic Algorithm requires a high level of technical expertise in programming, algorithm development, and database management. The development team must have experience in working with the latest technologies, such as Python, Java, and SQL, to develop such a system. The development team must also have access to a powerful computing system to run the genetic algorithm and generate the timetables. Based on the current state of technology, it is technically feasible to develop the Automatic Time Table Generator using Genetic Algorithm.

3.5.2 Economic Feasibility

Economic feasibility is the evaluation of the cost-effectiveness of the proposed system. Developing an Automatic Time Table Generator using Genetic Algorithm requires a significant investment in terms of human resources, hardware and software infrastructure, and time. The cost of software and hardware licenses and other resources must also be taken into account. However, the potential benefits of the system far outweigh the costs. An efficient and effective timetable generator would significantly reduce the workload of administrators and faculty, increase student satisfaction and retention, and improve the overall efficiency of the institution. Therefore, the Automatic Time Table Generator using Genetic Algorithm is economically feasible.

3.5.3 Operational Feasibility

Operational feasibility is the evaluation of whether the proposed system can be integrated into the existing operational environment of the organization. The Automatic

Time Table Generator using Genetic Algorithm must be compatible with existing systems and processes of the institution. The development team must work closely with the administration, faculty, and IT staff to ensure that the system meets the specific needs and requirements of the institution. The system must also user-friendly and easy to navigate, with minimal training required for users. Based on the operational requirements of institution, it is operationally feasible to develop it.

3.5.4 Scheduling Feasibility

Scheduling feasibility is the evaluation of whether the proposed system can be developed and implemented within the available timeframe. The development team must have a clear understanding of the time required for the various phases of the project, including requirements gathering, design, development, testing, and implementation. The team must also work within the constraints of the academic calendar to ensure that the system is ready for use at the start of the academic year. Based on the available resources and timeline, it is feasible to develop and implement the Automatic Time Table Generator using Genetic Algorithm.

4. IMPLEMENTATION

4.1 LANGUAGE/TECHNOLOGY USED

Java and Genetic Algorithm from Machine Learning. Java is a general-purpose programming language that is designed to be platform-independent, meaning that it can run on any platform or operating system. Java was first introduced in 1995 by James Gosling at Sun Microsystems, which was later acquired by Oracle Corporation in 2010. Java is one of the most popular programming languages in the world, and is widely used in various fields, such as mobile app development, web development, game development, and more.

One of the key features of Java is its portability. Java code can be compiled into bytecodes, which can be run on any machine that has a Java Virtual Machine (JVM) installed. This allows developers to write a single piece of code that can run on multiple platforms without any modifications.

OOP is a programming paradigm that is based on the concept of objects, which are instances of classes. A class is a blueprint for creating objects, and defines the properties and methods of the objects. Java supports all the OOP concepts, such as encapsulation, inheritance, and polymorphism. Java code runs inside a sandbox environment, which provides a layer of protection against malicious code. Java applets, which are small programs that run inside a web browser, were once a popular way to add interactivity to web pages. However, due to security concerns, Java applets are no longer supported inmost modern web browsers.

Java has a vast standard library, which includes classes and interfaces for various tasks, such as input/output operations, networking, threading, and more. The standard library makes it easier for developers to write code without having to reinvent the wheel. Java is also used for Android application development. Android apps are written in Java and compiled into bytecodes, which are then run on the Android Virtual Machine (AVM). Java is a popular language for game development, particularly for developing mobile games. The Java- based game engine, Unity, is widely used for creating games for mobile devices. In conclusion, Java is a powerful and versatile programming language that has numerous applications in various fields. Its portability, security features, and vast standard library make it a popular choice for developers. Java's popularity is likely to continue in the future, as it continues to evolve and adapt to new technologies and platforms

4.1.1 Genetic Algorithm

A genetic algorithm (GA) is used to solve complex optimization problems, especially when the problem space is large and complex. GA is based on the principles of evolution, reproduction, selection, and mutation. It is widely used in engineering, computer science, economics, biology, and other fields. The basic idea of GA is to create a population of candidate solutions to a problem, evaluate the fitness of each solution, and then use the principles of natural selection and genetics to produce new and better solutions. Each solution is represented as a string of binary digits (genes), and the fitness of the solution is evaluated based on an objective function. The objective function measures the quality of the solution and provides a numerical score that represents the fitness of the solution.

The selection process in GA is based on the principle of survival of the fittest. Solutions with higher fitness scores are more likely to be selected for reproduction than solutions with lower fitness scores. The reproduction process involves combining two solutions to create a new solution that inherits some of the traits of its parents. This is done by selecting two solutions from the population, and then using a crossover operator to create a new solution that combines some of the genes from each parent. The new solution is then mutated, meaning that some of its genes are randomly altered, in order to introduce new genetic material into the population.

The process of selection, reproduction, and mutation is repeated over multiple generations. Each generation produces a new population of solutions, and the fitness of each solution is evaluated. The best solutions from each generation are retained, and they are used to create the next generation of solutions. The process continues until a satisfactory solution is found or a maximum number of generations is reached. One of the advantages of GA is that it is well-suited for solving complex optimization problems where the problem space is large and complex. It can handle problems with multiple objectives and constraints. Additionally, GA does not require any knowledge of the problem domain, making it a powerful and flexible optimization technique.

In summary, genetic algorithm is a powerful optimization technique that is based on the principles of natural selection and genetics. It is widely used in engineering, computer science, economics, biology, and other fields to solve complex optimization problems. The basic idea of GA is to create a population of candidate solutions, evaluate the fitness of each solution, and then use the principles of natural selection and genetics to produce new and better solutions over multiple generations.

Genetic algorithms (GAs) have become a cornerstone in various optimization problems, ranging from engineering to computer science. One intriguing application is their integration into automatic timetable generation systems, where the goal is to efficiently allocate resources, such as classrooms, teachers, and time slots, to meet predefined constraints and objectives.

The essence of a genetic algorithm lies in its mimicry of natural selection and evolution processes to find optimal solutions within a vast search space. In the context of automatic timetable generation, the algorithm operates on a population of potential solutions, each represented as a chromosome encoding the timetable configuration. These chromosomes undergo evolution through a series of genetic operators, including selection, crossover, and mutation, to iteratively refine and improve the timetable quality. The first step in applying a genetic algorithm to timetable generation is to define the chromosome representation. Typically, a timetable chromosome comprises a sequence of genes, each representing a particular time slot and assigned resource (e.g., course, classroom, teacher). This representation allows for the exploration of various timetable configurations within the search space.

The next crucial aspect is the definition of fitness functions. These functions evaluate the quality of a timetable solution based on predefined criteria and constraints. Common objectives include minimizing conflicts (e.g., overlapping classes, resource double-booking), balancing resource utilization, and optimizing preferences (e.g., minimizing gaps between classes). The fitness function guides the genetic algorithm towards solutions that satisfy these objectives.Once the chromosome representation and fitness function are defined, the genetic algorithm iterates through generations of populations, applying genetic operators to produce offspring with potentially improved fitness. Selection mechanisms, such as roulette wheel selection or tournament selection, choose chromosomes based on their fitness to participate in crossover and reproduction.In conclusion, genetic algorithms offer a robust and versatile approach to automatic timetable generation, enabling the efficient allocation of resources while satisfying diverse constraints and objectives. By emulating the principles of natural selection and evolution, genetic algorithms have emerged as a powerful tool for tackling complex optimization problems in various domains, including education management.

4.2 DATASETS

Automatic timetable generator using genetic algorithm is a powerful tool that can be used in various educational institutions to generate an optimized time table. The performance of the system depends largely on the quality and quantity of data used in the algorithm. In this context, selecting the right datasets plays a crucial role in the successful implementation of the system. The datasets used in automatic timetable generator can be broadly classified into three categories: input datasets, constraint datasets, and output datasets.

1. Input Datasets: Input datasets are the datasets that are required to generate a timetable. These datasets include information about the courses, faculties, rooms, and timeslots. The system needs to be fed with the data related to the courses such as course name, course code, course duration, course type, etc. Similarly, the system needs to be fed with the data related to faculties such as faculty name, faculty code, faculty specialization, faculty availability, etc. Information about the rooms such as room capacity, room type, and room availability also need to be fed into the system. Timeslots information such as time duration, time availability, and day availability also need to be fed into the system. All of these inputs are necessary to create an optimized timetable that satisfies all the constraints.

2. Constraint Datasets: The constraint datasets are the datasets that define the constraints that the system needs to adhere to while generating a timetable. These datasets include information about course and faculty availability, room availability, student preferences, and administrative policies. These constraints ensure that the system generates a feasible timetable that satisfies all the requirements. Some of the common constraints that need to be considered while generating a timetable are:

- No two courses can be scheduled in the same timeslot
- No faculty can be scheduled for two different courses at the same time - Each course should be scheduled in a room with sufficient capacity
- A faculty should be assigned to a course only if he/she is available during the scheduled timeslot
- Certain courses should be scheduled in a specific time slot
- Student preferences such as avoiding early morning classes, evening classes, consecutive classes, etc.
- Administrative policies such as no classes on weekends, national holidays, etc.

3. Output Datasets: Output datasets are the datasets that are generated as the output of the system. These datasets include the final timetable that satisfies all the constraints. The output datasets may include the details such as the course name, course code, faculty name, room number, and timeslot. The output datasets can also include the graphical representation of the timetable to make it easy to understand and visualize. Apart from these datasets, historical data such as previous timetables, faculty performance, student feedback, and other relevant data can be used to improve the accuracy of the algorithm. This data can be used to analyze the performance of the system and to identify the areas where the system needs improvement. Automatic timetable generator using genetic algorithm requires input data in the form of datasets to produce a timetable. The quality and quantity of the dataset directly impact the effectiveness and accuracy of the generated timetable.

Therefore, it is essential to have a reliable dataset to improve the performance of the automatic timetable generator. In this section, we will discuss the different datasets used in automatic timetable generator using genetic algorithm.

1. Course Dataset: This dataset contains information about the courses offered by the institution, including course code, course name, course duration, and credits. The course dataset is essential in designing the curriculum and creating the timetable as it provides information about the subjects to be taught in a particular semester or academic year.
2. Faculty Dataset: This dataset contains information about the faculty members available in the institution, including their names, designations, qualifications, and experience. The faculty dataset is essential in assigning the right faculty member for a particular course and slot in the timetable.
3. Time Slot Dataset: This dataset contains information about the available time slots for conducting classes, including start time, end time, and duration. The time slot dataset is essential in designing the timetable and allocating the right slot for a particular course and faculty member.
4. Student Dataset: This dataset contains information about the students enrolled in the institution, including their names, roll numbers, and the courses they have enrolled in. The student dataset is essential in designing the timetable and allocating the right slot for a particular course based on the availability of the students.
5. Constraint Dataset: This dataset contains information about the constraints and preferences of the institution, including the availability of faculty members, the availability of rooms, and the maximum number of classes that can be conducted in a

day. The constraint dataset is essential in designing the timetable and ensuring that the timetable meets the institution's requirements.

The quality and quantity of the dataset are critical in generating an effective timetable using genetic algorithm. Incomplete or incorrect data can lead to inaccurate or inefficient timetables, which can cause problems for the institution and the students. Therefore, it is essential to ensure that the dataset is reliable and up-to-date. In conclusion, automatic time table generator using genetic algorithm requiresinput data in the form of datasets to generate a timetable. The quality and quantity of the dataset directly impact the effectiveness and accuracy of the generated timetable. Therefore, it is essential to have a reliable dataset to improve the performance of the automatic time table generator.

In conclusion, selecting the right datasets is crucial to the success of an automatic time table generator using genetic algorithm. The datasets should be accurate, reliable, and sufficient to generate an optimized timetable that satisfies all the constraints. By selecting the right datasets and using them effectively, the automatic time table generator can help educational institutions to create a feasible and optimized timetable that satisfies all the requirements.

5. EXPERIMENTAL RESULT/OBSERVATIONS

5.1 EXPERIMENTAL SETUP

5.1.1. Development Language and Framework:

Java 8: Java version 8, released in March 2014, introduced several significant features and enhancements to the Java programming language. One of the most notable additions was the introduction of lambda expressions, which revolutionized the way developers write code by enabling functional programming constructs. This release also included the Streams API, providing a powerful way to process collections of objects in a functional style. Additionally, Java 8 introduced the java.time package, offering a modern date and time API to address the shortcomings of the legacy Date and Calendar classes. With its enhanced performance and productivity features, Java 8 has become a cornerstone for developers, facilitating the development of more concise, readable, and maintainable code.

Java Server Pages (JSP): JavaServer Pages (JSP) is a server-side technology that enables the creation of dynamic web content using Java programming language. With JSP, developers can embed Java code directly into HTML pages, facilitating the seamless integration of dynamic content and logic within web applications. By leveraging JSP's tag-based syntax and Java standard library, developers can build robust web applications with ease. JSP pages are compiled into servlets by the server at runtime, offering high performance and scalability. Additionally, JSP provides support for various Java EE technologies, such as servlets, JavaBeans, and JDBC, enabling developers to build feature-rich web applications that interact with databases, handle user input, and manage session data effectively. Overall, JSP empowers developers to create dynamic and interactive web experiences efficiently, making it a popular choice for building modern web applications.

Servlets: Servlets are Java-based server-side components that facilitate the development of dynamic web applications. Acting as intermediaries between the client request and server response, servlets enable the execution of server-side logic, processing user input, and generating dynamic content in various formats such as HTML, XML, or JSON. Leveraging the robustness and portability of the Java platform, servlets provide a scalable and efficient solution for handling web requests, supporting functionalities like session management, authentication, and database access. Servlets adhere to the Java Servlet API specifications, offering developers a standardized framework for building web applications that can run on any Servlet container, ensuring compatibility across different web servers and platforms. With their versatility and performance, servlets play a pivotal role in the development of enterprise-grade web applications, powering interactive and responsive user experiences across diverse domains.

5.1.2. Database

MySQL database: MySQL is a powerful relational database management system (RDBMS) renowned for its scalability, reliability, and ease of use. Developed by Oracle Corporation, MySQL is an open-source platform that has gained widespread popularity across various industries, from small businesses to large enterprises. Its robust features include support for multiple storage engines, comprehensive security mechanisms, and efficient transaction management capabilities. MySQL offers seamless integration with popular programming languages and frameworks, making it a preferred choice for web applications, data warehousing, and e-commerce platforms. With its rich ecosystem of tools and community support, MySQL remains a cornerstone in the realm of database management, empowering organizations to store, retrieve, and manipulate data efficiently and effectively.

5.1.3. Frontend Design

HTML 5: HTML5, the fifth iteration of the Hypertext Markup Language, represents a significant advancement in web development standards. Released in 2014, HTML5 introduces a plethora of new features and enhancements aimed at modernizing the web experience. One of its hallmark attributes is its robust support for multimedia elements, including native video and audio playback without the need for third-party plugins. Additionally, HTML5 offers enhanced semantic markup, allowing developers to structure web content more intuitively, which not only improves accessibility but also aids search engine optimization. Furthermore, HTML5 introduces several APIs that facilitate rich interactive experiences, such as the Canvas API for dynamic graphics rendering and the Web Storage API for client-side data storage. With its focus on accessibility, interoperability, and performance, HTML5 has become the cornerstone of contemporary web development, empowering developers to create immersive, responsive, and feature-rich web applications across various platforms and devices.

Cascading Style Sheets (CSS): Cascading Style Sheets (CSS) represent a cornerstone of modern web design, serving as a fundamental tool for controlling the visual presentation of web pages. Through CSS, developers can precisely dictate the layout, typography, colors, and other stylistic attributes of HTML elements, thus imbuing websites with aesthetic coherence and user-friendly interfaces. One of the key strengths of CSS lies in its cascading nature, allowing styles to be defined at various levels and then inherited or overridden as needed. This hierarchical approach fosters modularity and efficiency in code management, facilitating easier maintenance and updates across large-scale web projects. Moreover, CSS enables responsiveness and adaptability, empowering designers to create fluid layouts that seamlessly adjust to different screen sizes and devices, thereby ensuring optimal user experiences across diverse platforms. As web technologies continue to evolve, CSS remains a versatile and indispensable tool for shaping the visual identity and functionality of websites in an ever-changing digital landscape.

JavaScript: JavaScript, often abbreviated as JS, is a versatile and dynamic programming language primarily utilized for web development. Renowned for its ability to create interactive and engaging web experiences, JavaScript is an essential component of modern web applications. With its lightweight syntax and cross-platform compatibility, JavaScript empowers developers to enhance the functionality and interactivity of websites, ranging from simple animations to complex web applications. As a client-side scripting language, JavaScript executes code directly within a user's browser, enabling dynamic content manipulation and real-time updates without the need for server interaction. Additionally, JavaScript has expanded its domain beyond web development, finding applications in server-side development (Node.js), mobile app development (React Native), and even desktop application development (Electron). Its ubiquity and versatility make JavaScript a cornerstone of the digital landscape, driving innovation and pushing the boundaries of what is possible on the web.

Bootstrap: Bootstrap is a powerful front-end framework that has revolutionized web development by offering a comprehensive suite of tools and components for building responsive and visually appealing websites and web applications. Developed by Twitter, Bootstrap provides a robust grid system, pre-styled UI components, and

JavaScript plugins, enabling developers to create sleek and professional-looking interfaces with minimal effort. Its grid system facilitates the creation of layouts that seamlessly adapt to various screen sizes, ensuring optimal viewing experiences across devices. Moreover, Bootstrap's extensive library of CSS classes allows developers to easily customize the appearance and functionality of their projects, while its JavaScript plugins enhance interactivity and user experience. With its intuitive design and extensive documentation, Bootstrap has become the go-to framework for developers looking to streamline their workflow and deliver high-quality web solutions efficiently.

Ajax: Ajax, short for Asynchronous JavaScript and XML, is a fundamental technology in web development renowned for its ability to facilitate dynamic, asynchronous communication between web servers and client-side scripts. By enabling seamless updates to web content without necessitating a full page reload, Ajax revolutionized the user experience, empowering developers to create highly interactive and responsive web applications. Leveraging a combination of JavaScript, XML, HTML, CSS, and XMLHttpRequest object, Ajax enables data to be exchanged with the server in the background, allowing for the real-time updating of specific sections of a webpage. This capability has catalysed the proliferation of rich internet applications (RIAs) and ushered in an era of enhanced user engagement and interactivity on the web. From live chat features to dynamic form submissions and interactive maps, Ajax continues to underpin the functionality of numerous web applications, enriching the browsing experience for users worldwide.

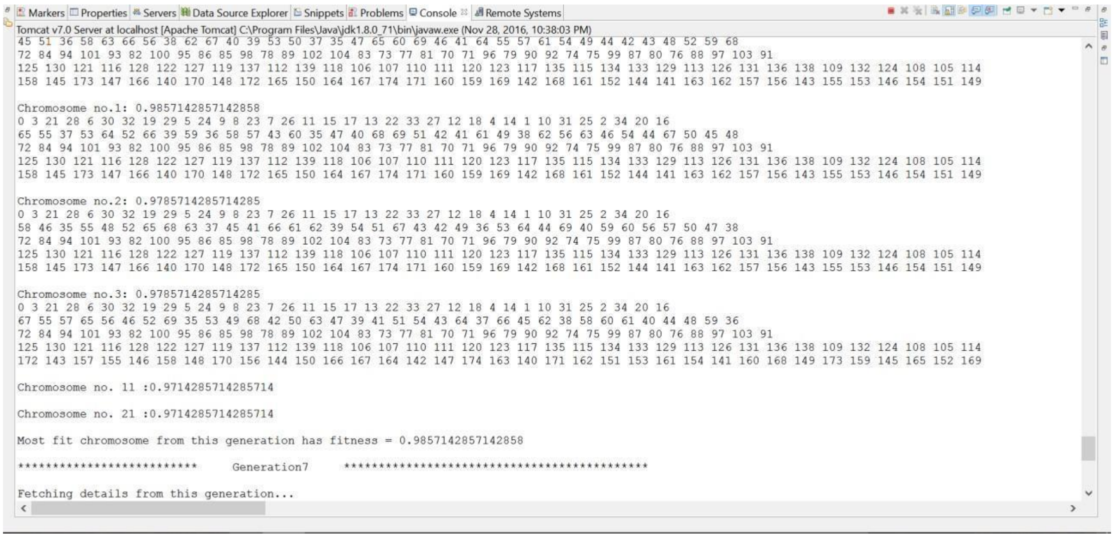
4. Web Server:

Apache Tomcat-9 (locally hosted during development):Building upon its predecessor's legacy, Tomcat 9 introduces a slew of enhancements and optimizations, solidifying its position as a cornerstone in the realm of Java application servers. With its lightweight architecture and impressive scalability, Tomcat 9 offers unparalleled performance, making it the preferred choice for developers seeking a reliable and efficient environment for hosting their web applications. Boasting comprehensive support for the latest Java EE specifications, Tomcat 9 empowers developers to leverage cutting-edge technologies while ensuring seamless compatibility with existing applications.

6. DISCUSSION OF RESULTS

6.1 RESULTS

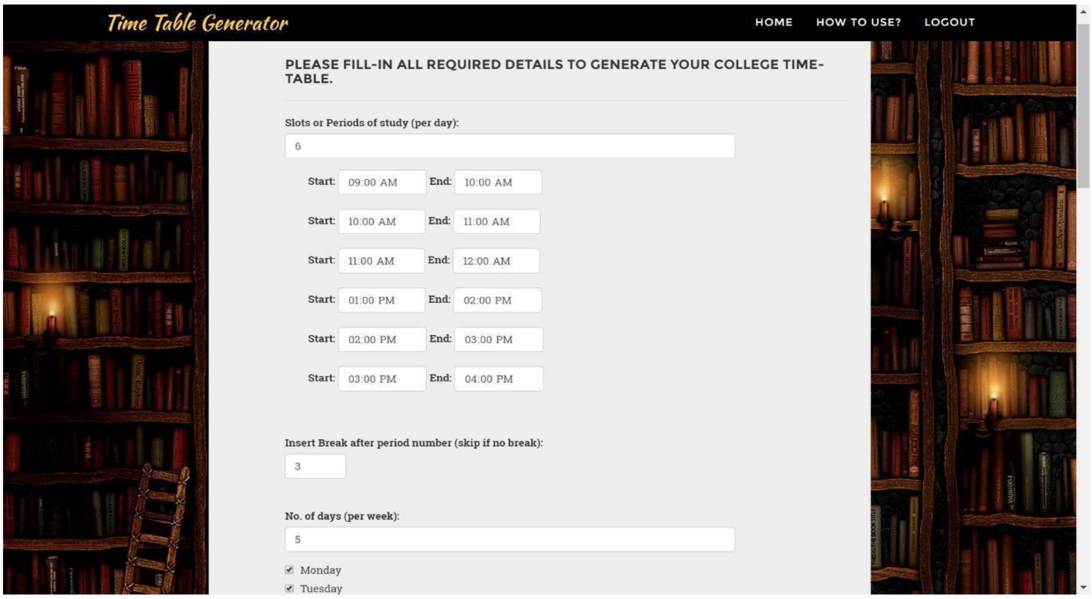
For the ease of testing and tracking, a lot of information is printed on the console itself. It involves input information, slots generated, few chromosomes from each generation of chromosome, fitness of these chromosomes, maximum fitness in a generation and final selected chromosome.



```
Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_71\bin\javaw.exe (Nov 28, 2016, 10:38:03 PM)
45 51 36 58 63 66 56 38 62 67 40 39 53 50 37 35 47 65 60 69 46 41 64 55 57 61 54 49 44 42 43 48 52 59 68
72 84 94 101 93 82 100 95 86 85 98 78 89 102 104 83 73 77 81 70 71 96 79 90 92 74 75 99 87 80 76 88 97 103 91
125 130 121 116 128 122 127 119 137 112 139 118 106 107 110 111 120 123 117 135 115 134 133 129 113 126 131 136 138 109 132 124 108 105 114
158 145 173 147 166 140 170 148 172 165 150 164 167 174 171 160 159 169 142 168 161 152 144 141 163 162 157 156 143 155 153 146 154 151 149

Chromosome no.1: 0.9857142857142858
0 3 21 28 6 30 32 19 29 5 24 9 8 23 7 26 11 15 17 13 22 33 27 12 18 4 14 1 10 31 25 2 34 20 16
65 55 37 53 64 52 66 39 59 36 58 57 43 60 35 47 40 68 69 51 42 41 61 49 38 62 56 63 46 54 44 67 50 45 48
72 84 94 101 93 82 100 95 86 85 98 78 89 102 104 83 73 77 81 70 71 96 79 90 92 74 75 99 87 80 76 88 97 103 91
125 130 121 116 128 122 127 119 137 112 139 118 106 107 110 111 120 123 117 135 115 134 133 129 113 126 131 136 138 109 132 124 108 105 114
158 145 173 147 166 140 170 148 172 165 150 164 167 174 171 160 159 169 142 168 161 152 144 141 163 162 157 156 143 155 153 146 154 151 149

Chromosome no.2: 0.9785714285714285
0 3 21 28 6 30 32 19 29 5 24 9 8 23 7 26 11 15 17 13 22 33 27 12 18 4 14 1 10 31 25 2 34 20 16
58 46 35 55 48 52 65 68 63 37 45 41 66 61 62 39 54 51 67 43 42 49 36 53 64 44 69 40 59 60 56 57 50 47 38
72 84 94 101 93 82 100 95 86 85 98 78 89 102 104 83 73 77 81 70 71 96 79 90 92 74 75 99 87 80 76 88 97 103 91
125 130 121 116 128 122 127 119 137 112 139 118 106 107 110 111 120 123 117 135 115 134 133 129 113 126 131 136 138 109 132 124 108 105 114
158 145 173 147 166 140 170 148 172 165 150 164 167 174 171 160 159 169 142 168 161 152 144 141 163 162 157 156 143 155 153 146 154 151 149

Chromosome no.3: 0.9785714285714285
0 3 21 28 6 30 32 19 29 5 24 9 8 23 7 26 11 15 17 13 22 33 27 12 18 4 14 1 10 31 25 2 34 20 16
67 55 57 65 56 46 52 69 35 53 49 68 42 50 63 47 39 41 51 54 43 64 37 66 45 62 38 58 60 61 40 44 48 59 36
72 84 94 101 93 82 100 95 86 85 98 78 89 102 104 83 73 77 81 70 71 96 79 90 92 74 75 99 87 80 76 88 97 103 91
125 130 121 116 128 122 127 119 137 112 139 118 106 107 110 111 120 123 117 135 115 134 133 129 113 126 131 136 138 109 132 124 108 105 114
172 143 157 155 146 158 148 170 156 144 150 166 167 164 142 147 174 163 140 171 162 151 153 161 154 141 160 168 149 173 159 145 165 152 169

Chromosome no. 11 :0.9714285714285714

Chromosome no. 21 :0.9714285714285714

Most fit chromosome from this generation has fitness = 0.9857142857142858

**************************     Generation7     ********************************************

Fetching details from this generation...
```

Fig 6.1.1 Testing the algorithm

- When tested with new input through the form for generation of time-table :



PLEASE FILL-IN ALL REQUIRED DETAILS TO GENERATE YOUR COLLEGE TIME-TABLE.

Slots or Periods of study (per day):
6

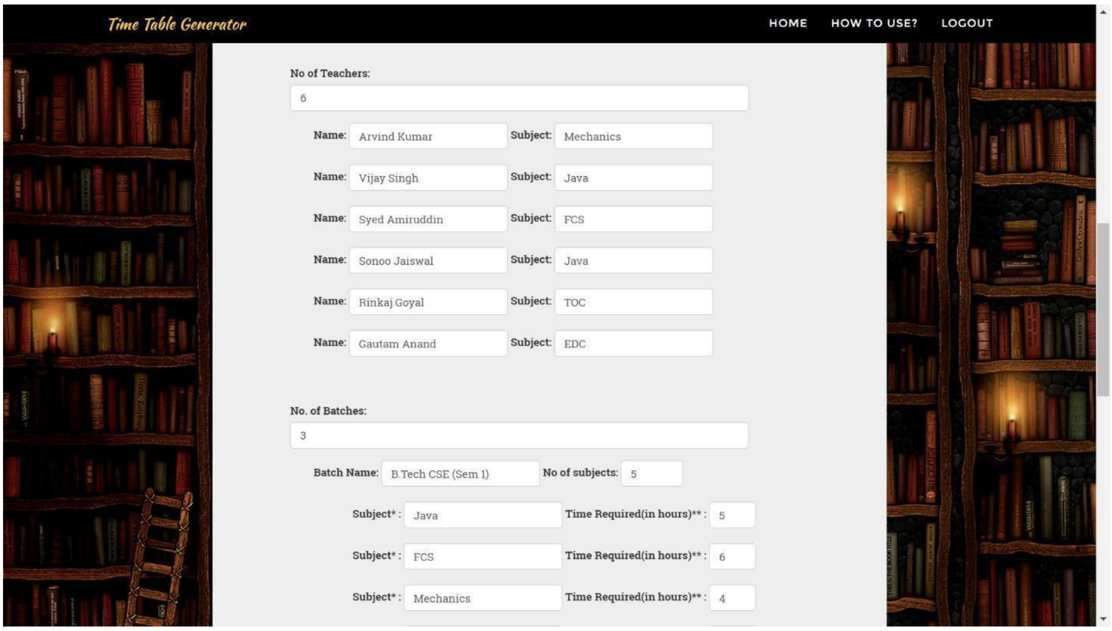
Start: 09:00 AM End: 10:00 AM
Start: 10:00 AM End: 11:00 AM
Start: 11:00 AM End: 12:00 AM
Start: 01:00 PM End: 02:00 PM
Start: 02:00 PM End: 03:00 PM
Start: 03:00 PM End: 04:00 PM

Insert Break after period number (skip if no break):
3

No. of days (per week):
5

☑ Monday
☑ Tuesday

Fig 6.1.2.1 Inputs on webpage(ex.1)



No of Teachers: 6

Name: Arvind Kumar Subject: Mechanics

Name: Vijay Singh Subject: Java

Name: Syed Amiruddin Subject: FCS

Name: Sonoo Jaiswal Subject: Java

Name: Rinkaj Goyal Subject: TOC

Name: Gautam Anand Subject: EDC

No. of Batches: 3

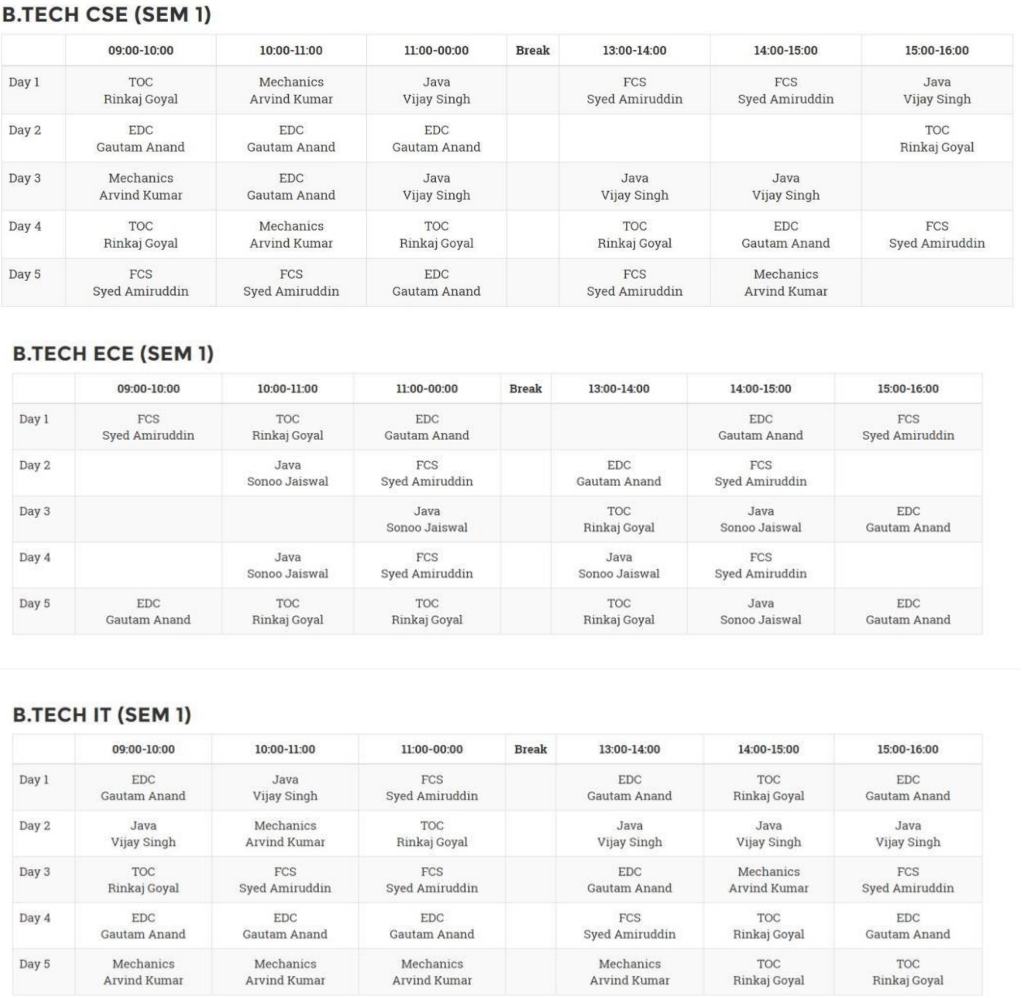
Batch Name: B.Tech CSE (Sem 1) No of subjects: 5

Subject* : Java Time Required(in hours)** : 5

Subject* : FCS Time Required(in hours)** : 6

Subject* : Mechanics Time Required(in hours)** : 4

Fig 6.1.2.2 Inputs on webpage(ex.1) **Output:**



B.TECH CSE (SEM 1)

	09:00-10:00	10:00-11:00	11:00-00:00	Break	13:00-14:00	14:00-15:00	15:00-16:00
Day 1	TOC Rinkaj Goyal	Mechanics Arvind Kumar	Java Vijay Singh		FCS Syed Amiruddin	FCS Syed Amiruddin	Java Vijay Singh
Day 2	EDC Gautam Anand	EDC Gautam Anand	EDC Gautam Anand				TOC Rinkaj Goyal
Day 3	Mechanics Arvind Kumar	EDC Gautam Anand	Java Vijay Singh		Java Vijay Singh	Java Vijay Singh	
Day 4	TOC Rinkaj Goyal	Mechanics Arvind Kumar	TOC Rinkaj Goyal		TOC Rinkaj Goyal	EDC Gautam Anand	FCS Syed Amiruddin
Day 5	FCS Syed Amiruddin	FCS Syed Amiruddin	EDC Gautam Anand		FCS Syed Amiruddin	Mechanics Arvind Kumar	

B.TECH ECE (SEM 1)

	09:00-10:00	10:00-11:00	11:00-00:00	Break	13:00-14:00	14:00-15:00	15:00-16:00
Day 1	FCS Syed Amiruddin	TOC Rinkaj Goyal	EDC Gautam Anand			EDC Gautam Anand	FCS Syed Amiruddin
Day 2		Java Sonoo Jaiswal	FCS Syed Amiruddin		EDC Gautam Anand	FCS Syed Amiruddin	
Day 3			Java Sonoo Jaiswal		TOC Rinkaj Goyal	Java Sonoo Jaiswal	EDC Gautam Anand
Day 4		Java Sonoo Jaiswal	FCS Syed Amiruddin		Java Sonoo Jaiswal	FCS Syed Amiruddin	
Day 5	EDC Gautam Anand	TOC Rinkaj Goyal	TOC Rinkaj Goyal		TOC Rinkaj Goyal	Java Sonoo Jaiswal	EDC Gautam Anand

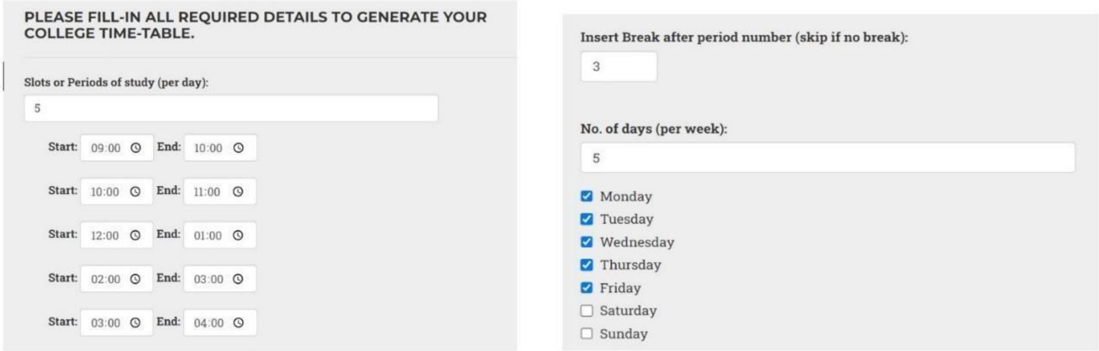
B.TECH IT (SEM 1)

	09:00-10:00	10:00-11:00	11:00-00:00	Break	13:00-14:00	14:00-15:00	15:00-16:00
Day 1	EDC Gautam Anand	Java Vijay Singh	FCS Syed Amiruddin		EDC Gautam Anand	TOC Rinkaj Goyal	EDC Gautam Anand
Day 2	Java Vijay Singh	Mechanics Arvind Kumar	TOC Rinkaj Goyal		Java Vijay Singh	Java Vijay Singh	Java Vijay Singh
Day 3	TOC Rinkaj Goyal	FCS Syed Amiruddin	FCS Syed Amiruddin		EDC Gautam Anand	Mechanics Arvind Kumar	FCS Syed Amiruddin
Day 4	EDC Gautam Anand	EDC Gautam Anand	EDC Gautam Anand		FCS Syed Amiruddin	TOC Rinkaj Goyal	EDC Gautam Anand
Day 5	Mechanics Arvind Kumar	Mechanics Arvind Kumar	Mechanics Arvind Kumar		Mechanics Arvind Kumar	TOC Rinkaj Goyal	TOC Rinkaj Goyal

Fig 6.1.2.3 Results(ex.1)

- In the experiment described above, we observed that our model functions adequately with basic inputs. Now, let us examine its performance with inputs where multiple teachers teach the same subject and analyze the results.

Input:



PLEASE FILL-IN ALL REQUIRED DETAILS TO GENERATE YOUR COLLEGE TIME-TABLE.

Slots or Periods of study (per day): 5

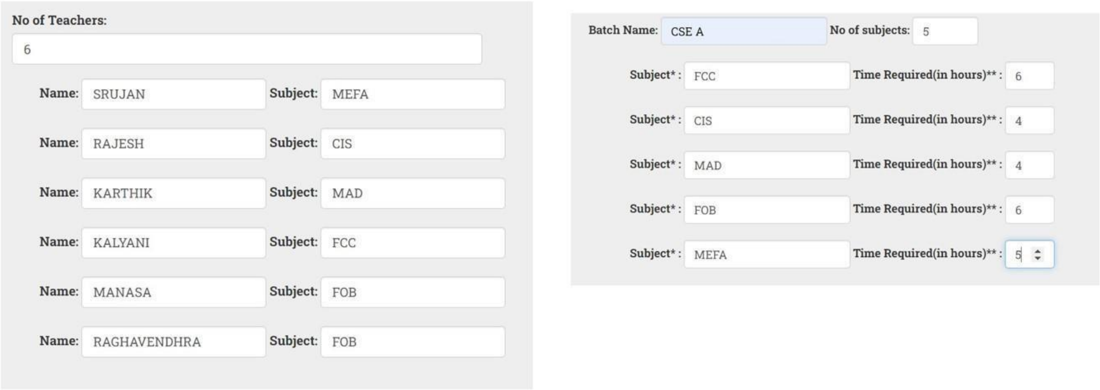
Start:	09:00	End:	10:00
Start:	10:00	End:	11:00
Start:	12:00	End:	01:00
Start:	02:00	End:	03:00
Start:	03:00	End:	04:00

Insert Break after period number (skip if no break): 3

No. of days (per week): 5

☑ Monday
☑ Tuesday
☑ Wednesday
☑ Thursday
☑ Friday
☐ Saturday
☐ Sunday

Fig 6.1.3.1 Inputs on webpage(ex.2)



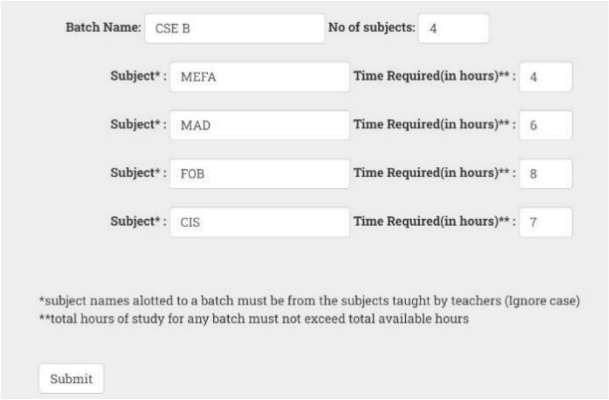
No of Teachers: 6

Name:	SRUJAN	Subject:	MEFA
Name:	RAJESH	Subject:	CIS
Name:	KARTHIK	Subject:	MAD
Name:	KALYANI	Subject:	FCC
Name:	MANASA	Subject:	FOB
Name:	RAGHAVENDHRA	Subject:	FOB

Batch Name: CSE A No of subjects: 5

Subject* :	FCC	Time Required(in hours)** :	6
Subject* :	CIS	Time Required(in hours)** :	4
Subject* :	MAD	Time Required(in hours)** :	4
Subject* :	FOB	Time Required(in hours)** :	6
Subject* :	MEFA	Time Required(in hours)** :	5

Fig 6.1.3.2 Inputs on webpage(ex.2)



Batch Name: CSE B No of subjects: 4

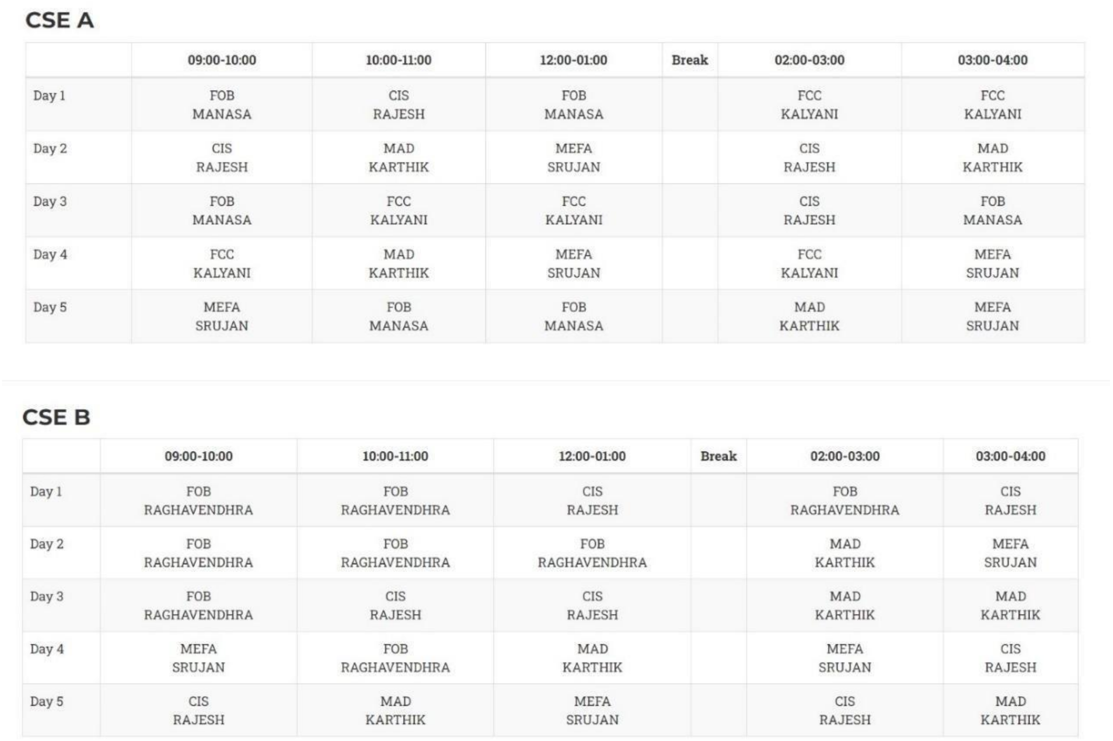
Subject* :	MEFA	Time Required(in hours)** :	4
Subject* :	MAD	Time Required(in hours)** :	6
Subject* :	FOB	Time Required(in hours)** :	8
Subject* :	CIS	Time Required(in hours)** :	7

*subject names alotted to a batch must be from the subjects taught by teachers (Ignore case)
**total hours of study for any batch must not exceed total available hours

Fig 6.1.3.3 Inputs on webpage(ex.2)

Output:

In the input we have given 2 faculties for FOB, the algorithm gave the equal load for both the faculties as below.



CSE A

	09:00-10:00	10:00-11:00	12:00-01:00	Break	02:00-03:00	03:00-04:00
Day 1	FOB MANASA	CIS RAJESH	FOB MANASA		FCC KALYANI	FCC KALYANI
Day 2	CIS RAJESH	MAD KARTHIK	MEFA SRUJAN		CIS RAJESH	MAD KARTHIK
Day 3	FOB MANASA	FCC KALYANI	FCC KALYANI		CIS RAJESH	FOB MANASA
Day 4	FCC KALYANI	MAD KARTHIK	MEFA SRUJAN		FCC KALYANI	MEFA SRUJAN
Day 5	MEFA SRUJAN	FOB MANASA	FOB MANASA		MAD KARTHIK	MEFA SRUJAN

CSE B

	09:00-10:00	10:00-11:00	12:00-01:00	Break	02:00-03:00	03:00-04:00
Day 1	FOB RAGHAVENDHRA	FOB RAGHAVENDHRA	CIS RAJESH		FOB RAGHAVENDHRA	CIS RAJESH
Day 2	FOB RAGHAVENDHRA	FOB RAGHAVENDHRA	FOB RAGHAVENDHRA		MAD KARTHIK	MEFA SRUJAN
Day 3	FOB RAGHAVENDHRA	CIS RAJESH	CIS RAJESH		MAD KARTHIK	MAD KARTHIK
Day 4	MEFA SRUJAN	FOB RAGHAVENDHRA	MAD KARTHIK		MEFA SRUJAN	CIS RAJESH
Day 5	CIS RAJESH	MAD KARTHIK	MEFA SRUJAN		CIS RAJESH	MAD KARTHIK

Fig 6.1.3.4 Results(ex.2)

6.2 PARAMETER COMPARISION TABLE

Parameter	Previous methods	Proposed method
Equal load to all faculties	Previous models used to fill the slots with the single faculty and if that faculty is busy, model will assign slot for other faculty.	In the proposed method, the model will consider all faculties for a same subject and allocates slots equally.
Selection of working days	There is no selection of number of working days.	User can select number of working days and also select the specific working days.

Break slot and periods	User cannot select break timings, number of slots and their timings	User can specify break timings and number of slots along with the timings for each slot.

Table 6.2 Parameter Comparision Table

7. CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The automatic time table generator using genetic algorithm is a software system that aims to automate the process of generating optimized timetables for educational institutions. The system is designed to meet the functional and non- functional requirements of the system, ensuring that it is easy to use, reliable, and scalable. The use of genetic algorithms allows the system to generate a large number of possible timetables and select the best possible option based on fitness function criteria. This allows the system to produce optimal timetables that meet the requirements of the educational institution while taking into account various constraints and preferences. The proposed system provides various modules such as course management, faculty management, classroom management, and timetable generation. The course management module allows the user to add, edit and delete courses and their details, including course name, code, duration, and number of sessions per week. The faculty management module allows the user to manage faculty details, including name, code, and subject specialization.

The classroom management module allows the user to manage classroom details, including room number, capacity, and availability. The timetable generation module is the core of the system, which uses genetic algorithms to generate optimized timetables based on user-defined constraints and preferences. The system is developed using various software development tools and technologies, including Java programming language, Eclipse Integrated Development Environment (IDE), and MySQL database management system. The system is designed to be user-friendly, with a simple and intuitive user interface that allows the user to perform various tasks with ease. The system is tested using various software testing techniques, such as unit testing, integration testing, functional testing, and acceptance testing. Unit testing involves testing individual modules and components of the system to ensure that they are working correctly. Functional testing involves testing the functionality of the system to ensure that it meets the user's requirements and expectations. Acceptance testing involves testing the system with real-world data to ensure that it meets the customer's requirements and expectations. The functional requirements of the system include course management, faculty management, classroom management, and timetable generation.

The non-functional requirements of the system include usability, reliability, performance, and scalability. The system is designed to be easy to use, reliable, and scalable, ensuring that it can be used by educational institutions of all sizes and complexities. The system has several advantages, including saving time and resources, improving operational efficiency, and increasing student and faculty satisfaction. The system can generate optimized timetables quickly and efficiently, allowing educational institutions to focus on other important tasks. The system can also reduce conflicts and errors in the timetables, improving the overall quality and reliability of the educational institution. In conclusion, the automatic timetable generator using genetic algorithm is an efficient and effective solution for generating optimized timetables for educational institutions.

The system is designed to meet the functional and non-functional requirements of the system, ensuring that it is easy to use, reliable, and scalable. The system is tested using various software testing techniques, ensuring that it is free from defects and meets the customer's requirements and expectations. The system has several advantages, including saving time and resources, improving operational efficiency, and increasing student and faculty satisfaction. The system is a valuable tool for educational institutions looking to automate the process of generating optimized timetables while ensuring the quality and reliability of the timetables.

7.2 FUTURE SCOPE

The use of automatic timetable generators based on genetic algorithms has the potential for a wide range of applications in various domains such as schools, colleges, universities, and other educational institutions. As the technology continues to evolve and improve, there are several areas where automatic timetable generators can be further developed in the future. One of the areas where automatic timetable generators can be improved is the optimization of the objective function.

While current systems optimize the objective function to minimize constraints such as conflicts in scheduling and the number of teacher or classroom changes, future systems can aim to further optimize the objective function to consider other factors such as student performance, teacher preferences, and student engagement levels. Another area of improvement is the incorporation of machine learning algorithms into the system. With machine learning algorithms, automatic timetable generators can learn from past data and improve their performance over time.

For instance, by analysing past timetables and the performance of the students in those timetables, the algorithm can learn to generate timetables that lead to better student outcomes. Another future area of development is the integration of real-time data feeds into the system. This could include information about teacher absences, classroom availability, and student enrollment changes. By integrating this information into the system, the algorithm can generate timetables that account for these changes in real-time.

Furthermore, the use of automatic timetable generators can be extended to other domains beyond the education sector. For instance, it can be used in hospitals to schedule appointments for patients, assign doctors to specific shifts, and manage the availability of medical equipment. It can also be used in the transportation sector to schedule the arrival and departure of buses and trains, optimize routes, and manage vehicle maintenance. Another area of future development is the use of cloud computing. With cloud computing, automatic timetable generators can be made accessible to more users and institutions, thereby increasing their reach and impact. Cloud computing can also enable the sharing of data and insights across institutions and allow for better collaboration between different stakeholders.

In addition, the use of automatic timetable generators can be extended to other related tasks such as room allocation and event scheduling. For instance, in a university setting, the automatic timetable generator can be extended to allocate rooms for conferences, seminars, and other events. Finally, there is also potential for the integration of automatic timetable generators with other technologies such as virtual reality and augmented reality.

With virtual reality and augmented reality, students can visualize their timetable in a more interactive and immersive way, leading to better engagement and motivation. In conclusion, automatic timetable generators using genetic algorithms have a wide range of applications in various domains beyond the education sector. While current systems have shown great promise, there is still a lot of room for improvement and future development. With the integration of machine learning algorithms, real-time data feeds, cloud computing, and other emerging technologies, the potential for these systems is enormous, and we can expect to see significant advancements in the years to come.

8. REFERENCES

[1] A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA) by Esraa A. Abdelhalim, Ghada A. El Khayat, Information Systems and Computers Department, Faculty of Commerce, Alexandria University, Alexandria, Egypt, Received 6 September 2014, Revised 10 February 2016, Accepted 20 February 2016, Version of Record 25 June 2016.

[2] Hybrid Genetic Algorithms for University Course Timetabling, March 2012,International Journal of Computer Science Issues 9(2) by Meysam Shahvali Kohshori and Mohammad Saniee Abadeh.

[3] Automatic Timetabling System for University Course" by Mrunmayee V. Rane, Vikram M. Apte, Vishakha N. Nerkar, Mani Roja Edinburgh, K.Y. Rajput published in 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) AISSMS Institute of Information Technology, Pune, India.

[4] Abramson, D. (1991). Constructing Schools Timetables Using Simulated Annealing:
Sequential and Parallel Algorithms Management Science.

[5] Adejuwon, O. S. (2012). Development of a University timetable automation system.

[6] Anisha, J., Ganapathy, S. C., Harshita, G., & Rishabh, B. (2015). A Literature Review on Timetable generation algorithms based on Genetic Algorithm and Heuristic approach. International Journal of Advanced Research in Computer and Communication Engineering, 159-163.

[7] Back, T. (1995). Evolutionary Algorithms in theory and practice.

[8] Baricelli Nils Aall. (1957). Symbiogenetic evolution processes realized by artificial methods. 143- 182.

[9] Barkha, N., Ambika, G., & Rashmi, B. (2013, July). Use of Active Rules and Genetic Algorithm to Generate the Automatic Timetable. International Journal of Advances in Engineering Sciences Vol.3