International Journal of Research in Engineering, Science and Management
Volume-3, Issue-3, March-2020
www.ijresm.com | ISSN (Online): 2581-5792

Automatic Timetable Generator

Shraddha Ambhore[1], Pooja Walke[2], Rohit Ghundgrudkar[3], Akshay Alone[4], Anushree Khedkar[5]
[1,3,4,5]*Student, Department of Information Technology, St. Vincent Palloti College of Engineering and Technology, Nagpur, India*
[2]*Assistant Professor, Department of Information Technology, St. Vincent Palloti College of Engineering and Technology, Nagpur, India*

***Abstract*: "Timetable" generation is a tedious job for educationalists as they need to prepare a timetable manually for each semester. The major challenge is that the time table should be prepared according to the University scheme predefined for each subject. Another constraint that plays a major role in preparing a timetable is the workload of faculty members that leads to the availability of that faculty member for respective subjects. While preparing a time table one should take these constraints in mind and prepare timetable in such a way that it does not overlap with their other schedules and can efficiently be utilized. The proposed system of the project will help to avoid the complexity of setting and managing Timetable manually. In our project, we are going to use algorithms like Evolutionary Algorithm, Tabu Search, Simulated Annealing, Scatter search to overcome difficulties of generating timetable manually. The system will take inputs like semester wise subjects, teachers, and the workload of the teacher. By relying on these inputs, it will generate a possible time table for the working days of the week. This will integrate by making optimal use of all resources during a way that will best suit the constraints. So, our proposed system will help to overcome the constraints that were appearing in the system, resulting in timetables for any number of courses and multiple semesters. This system will help to create a dynamic solution with the best approach.**

***Keywords*: N Queen, Genetic Algorithm, Time tabling, Scheduling, Timetable Generator.**

1. Introduction

A timetable is a kind of schedule that sets out times at which specific events are intended to occur is. When an educationalist tends to prepare a timetable manually the main task is to check the availability and work of the faculty members for a respective subject. The other main job is generating a timetable keeping in mind the predefined university scheme for each subject of a particular semester. The scheduling of time slots for each subject of a semester should be so smooth that there is no overlapping of slots of the same faculty member teaching to different semesters. The proposed system of the project will help to avoid the complexity of setting and managing Timetable manually multiple times. By using Machine Learning, the system will take inputs like semester wise subjects, faculty members, and the workload of the faculty member. By relying on these inputs, it will generate a possible time table for the working days of the week. So our proposed system will help to overcome the constraints that were appearing in the system by reducing the tedious job of manual timetable generator.

2. Literature survey

A literature survey is an evaluative report of information found in the literature related proposed work. When conducting research, a literature review may be a crucial part of the project because it covers all previously researched the subject and sets the platform for current research. It is the foremost important part of your report because it gives you a direction within the area of your research. It helps you to set a goal for your analysis.

"Time Table Scheduling using Genetic Artificial Immune Network" proposes Scheduling is one of the important tasks encountered in real-life situations. Various scheduling problems are present, like personnel scheduling, production scheduling, education schedule, etc. Educational schedule scheduling could also be a difficult task thanks to the varied constraints that are needed to be satisfied so on urge a feasible solution. Methodologies like Genetic Algorithms (GAs), has been used with mixed success [1].

We've reviewed the scheduling system of our institute and later tried solving it with the genetic algorithm. Finding a feasible lecture/tutorial timetable for a department may be a challenging problem faced continually in educational establishments. N Queen algorithm based approach to unravel a heavily constrained university timetable problem proved to a feasible solution for solving the problem. Using N Queen seamed to be useful but still, certain issues have to be overcome [2].

A. N Queen Algorithm

The N Queen is that the problem of placing N chess queens on an N×N chessboard in order that no two queens attack one another. For example, the following is a solution for the 4 Queen problem. The idea is to put queens one by one in several columns, ranging from the leftmost column. When we place a queen during a column, we check for clashes with already placed queens. In the current column, if we discover a row that there's no clash, we mark this row and column as a part of the answer. If we don't find such a row thanks to clashes, then we backtrack and return false.

B. Genetic Algorithm

A genetic algorithm may be a search heuristic that's inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the method of survival where the fittest individuals are selected for a copy to supply the off spring of a subsequent generation. The process of survival starts with the choice of fittest individuals from a population. They produce offspring which inherit the characteristics of the oldsters and can be added to a subsequent generation. If parents have better fitness, their offspring are going to be better than parents and have a far better chance of surviving. This process keeps on iterating and in the end, a generation with the fittest individuals will be found. This notion is often applied to an inquiry problem. We consider a group of solutions for drag and choose the set of best ones out of them.

Five phases are considered in a genetic algorithm.

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

1) Initial population

The first step in the performance of a GA is the production of an initial population. Each member of this population encodes a possible solution to a retardant. Each unit is evaluated and assigned a fitness value in line with the fitness function. It has been acknowledged that if the initial population to the GA is good, then the algorithm has an enhanced option of finding a good result and if the initial supply of construction blocks is not large enough or good enough, then it would be hard for the algorithm to search out an honest result.

2) Selection

This operator selects chromosomes within the population for reproduction. The fitter the chromosome, the more times, it's likely to be chosen to breed. During each successive production, a portion of the accessible population is selected to select a new generation. Individual solutions are chosen through a fitness-based process, where fitter results are usually more likely to be chosen.

3) Crossover

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one creation to the succeeding It is parallel to reproduction and organic crossover, upon which genetic algorithms are based. A crossover takes more than one parent solution and producing a child solution. There are techniques for collection of the chromosomes. Crossover arbitrarily exchanges the subsequences before and afterward locus between two chromosomes to form two children. The crossover operator roughly does as it is natural recombining between two single chromosome organisms.

4) Mutation

Mutation is used to sustain genetic diversity from one creation of a population of genetic algorithm chromosomes to the next. It is parallel to natural mutation. Alteration (mutation) alters one or more gene values during a chromosome from its initial situation. In mutation, the result may alter totally from the previous result. Hence GA can come to enhanced results by using mutation. Mutation can take place at each bit position in a string with some possibility, usually very small.

5) Fitness function

The fitness function is described over the genetic representation and procedures the standard of the represented result. In particular, within the fields of genetic programming and genetic algorithms, each design result. After each round of testing, the thought is to remove the 'n' worst design solution. Therefore, desires to be awarded a shape of merit, to suggest how close it came to meeting the general necessity, and this is often generated by applying the fitness function to test, results obtained from that solution.

3. Proposed system

Timetables are generated manually by faculty members today. It takes up a lot of time. At the start of every academic year, one of the foremost responsibilities is to create a timetable for semesters. As simple as this sounds, in actuality, it can be a complex challenge as the faculty members need to make a time table with the availability of the faculty members across all semesters. Normally timetable generation is done manually and can be a frustrating and highly time-consuming task for faculty members.

The final system should able to generate time tables in a completely automated way which can save tons of your time and effort of an institute administration. It will generate a timetable system such that it can work equally well for all the semesters. The time table should be scheduled according to the University scheme predefined for each subject and the workload of faculty members that leads to the availability of that faculty member for respective subjects. This also focuses on the optimization of resources i.e. faculty members, labs and rooms, etc. By counting on these inputs, it'll generate possible time tables for working days of the week for teaching faculty. This will integrate by making optimal use of all resources in a way that will best suit the constraints.

A. Constraints

- Predefined university scheme for each subject.
- The workload of Faculty members.
- Avoid slot clashes of a faculty member.
- Priority should be given to the scheduling of practical slot, then tutorials slot.
- Provide labs for tutorials, if needed.

At the start of each academic year, the foremost responsibility is to prepare a timetable for each semester. As simple as this sounds, in actuality, it can be a complex challenge as the faculty members need to make a time table with the availability of the faculty members across all semesters. The final system should be ready to generate time tables in a

completely automated way which can save tons of your time and energy of institute administration. It will generate a timetable system such that it can work equally well for all the semesters. This also focuses on the optimization of resources i.e. faculty members, labs and rooms, etc. By counting on these inputs, it'll generate possible time tables for working days of the week for teaching faculty. This will integrate by making optimal use of all resources during a way that will best suit the constraints.

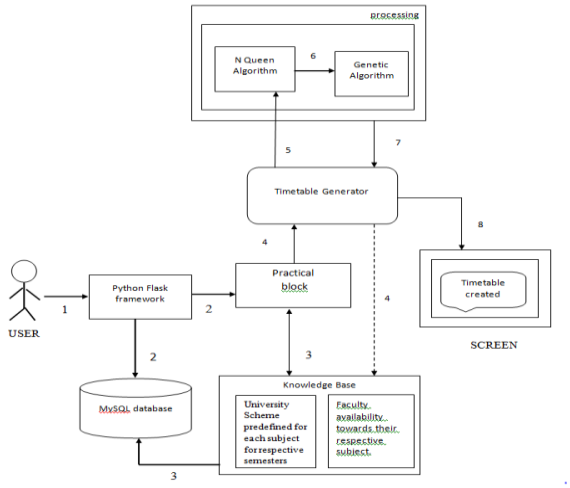

Fig. 1. Proposed system

Every faculty member has their faculty identification number or an employee identification number, through which they will register themselves on the portal. The system will take various inputs like a branch, year, semester, number of subjects, the workload of the faculty member.

For checking the availability of slots for any particular subject of a semester, there are certain rules to follow. Working on the constraints inserted in the knowledge base, the system will provide all the possible slots. In our project, we are going to use algorithms which will reduce the difficulties of generating time tables. The algorithms incorporate a numeral of strategies.

4. Methodology

A design methodology is a methodical approach to creating a plan by applying a set of methods.

1. The total requirement of the system including the framing of timetable strategy should be concerned.
2. A database should be formed.
3. Firstly, we should generate a timetable for the final year class, by entering details of the final year.
4. The workload allotted for a faculty member has to be followed.
5. Read the details like faculty, subjects into the database.
6. Retrieve the final year timetable timeslots and assign the timeslots to the respective faculty.
7. Labs must be assigned for tutorials if needed.
8. After practical's and tutorials, remaining slots should be assigned for theory lectures.
9. When generating the timetable for the third year, faculty id should be entered.
10. All the engaged slots for a particular faculty member must be shown as invalid.

5. System analysis

A. *Flowchart*

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. The diagram below shows the step by step working of the automatic timetable generator.

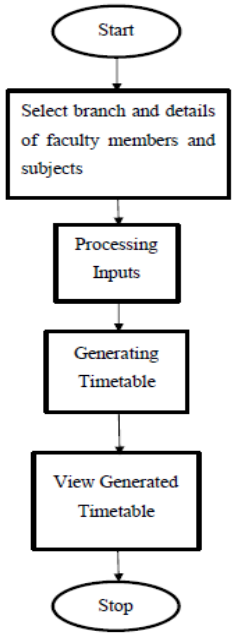

Fig. 2. Flowchart of system

B. *Use case diagram*

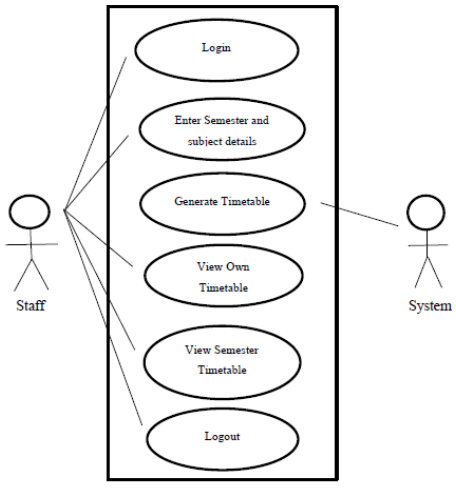

Fig. 3. Use case diagram of System

A Use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well. Here the staff will enter semester and subject details. The system will process these inputs and generate the timetable.

C. *Data Flow Diagram*

A data flow diagram is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

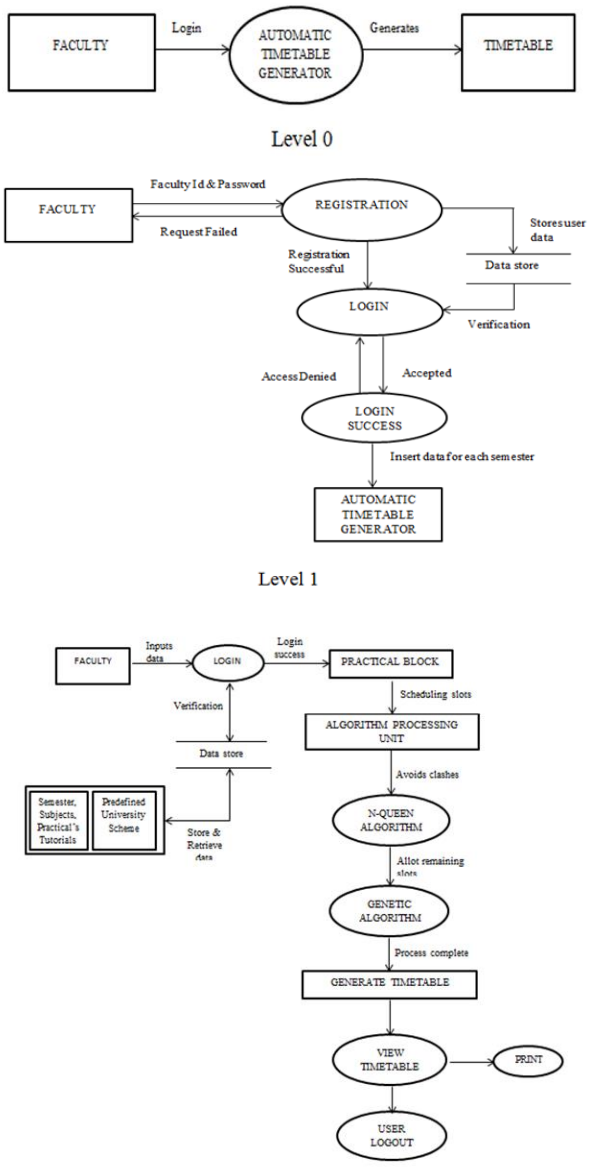

Fig. 4. Data Flow Diagram

6. Implementation

Implementation is a phase where all that we thought of comes into the picture. All the modules of our project are building using software development approaches. In our project, languages that we have used are HTML, PHP, Database used is MYSQL, Framework is Python Flask, the server used here is Machine Learning server (provided by our college), Operating system is Windows 7+.

Modules that are developed in our project are as follows:

A. *Registration*

In this module, every faculty needs to register themselves onto the portal. We provide the user various inputs to our system which gets store into the database.

B. *Login*

This is a submodule of registration. Once registered a user only needs to log in every time they want to access the system. In this module, the user will have to provide details about the semesters they are teaching in, subjects and practical's they will take. Every faculty member has their faculty identification number which will act as registration and login credentials.

C. *Generate*

Once the details are inserted into the database, the timetable generator can now do its task. Processing the data provided, and using the algorithm wisely, the timetable for even/odd semester will be generated. Display of timetable generated by our system provides viewing of timetable semester wise.

7. Result analysis

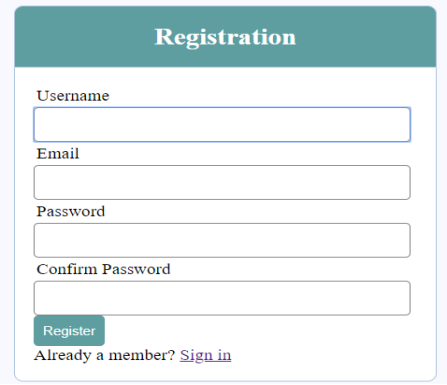

Fig. 5. Registration Window

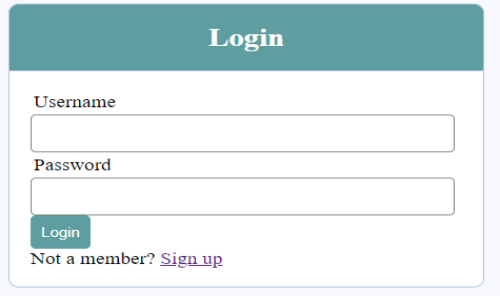

Fig. 6. Login Window

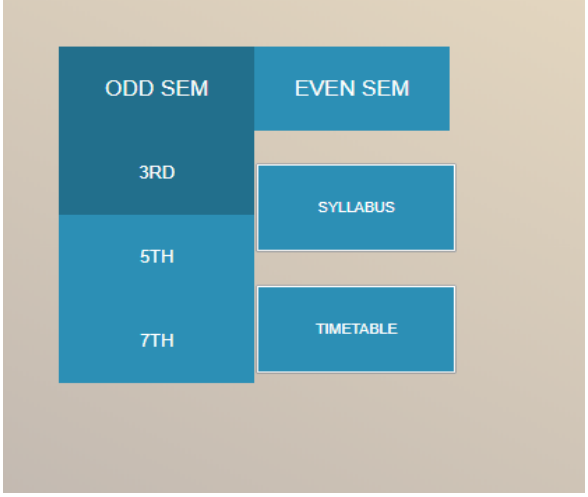

Fig. 7. Semester-wise timetable window

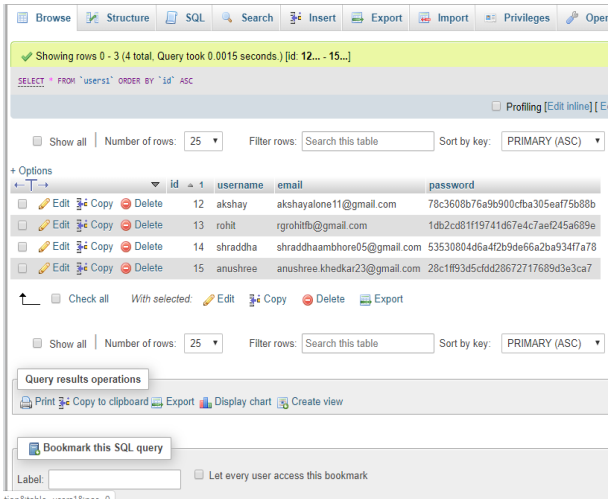

Fig. 8. Database Window

8. Conclusion

It is a complicated task to handle Faculty members and allocate lectures to them physically. So our proposed system will help to beat this disadvantage. Thus we can produce a timetable for any number of courses and multiple semesters. This system is user-friendly and provides a faster and better generation of timetable, which in turn saves time and manpower. The project reduces time consumption and the pain in making the timetable manually.

References

[1] M. D. Boomija, R. Ambika, J. Sandhiya, P. Jayashree, “Smart and Dynamic Timetable Generator”, International Journal for Research in Applied Science and Engineering Technology, March 2019.
[2] V. Abhinaya, K. Sahithi, K. Akaanksha, “Online Application of Automatic Timetable generator”, International Research Journal of Engineering and Technology, February 2019.
[3] Akshay Puttaswamy, H. M. Arshad Ali Khan, Chandan S. V, Parkavi. A, “A Study on Automatic Timetable Generator”, International Journal of Science and Innovative Engineering and Technology, May 2019.
[4] Adithya R Pai, Ashwitha S, Raksha Shetty, Geethalaxmi, “Automated college timetable generator”, International Journal of Scientific & Engineering Research, vol. 9, no. 4, April 2018.
[5] Saritha M, Pranav Kiran Vaze, Pradeep, Mahesh NR, “Automatic time table generator”, International Journal of Advanced Research in Computer Science and Software Engineering, May 2017.
[6] Dipesh Mittal, Hiral Doshi, Mohammed Sunasra, Renuka Nagpure, “Automatic Timetable Generator using Genetic Algorithm”, International Journal of Advanced Research in Computer and Communication Engineering, February 2015.
[7] Anuja Chowdhary, Priyanka Kakde, Shruti Dhoke, Sonali Ingle, Rupal Rushiya, Dinesh Gawande “Timetable Generation System,” IJCSMC, vol. 3, no. 2, February 2014.
[8] https://www.analyticsvidhya.com/blog/2017/07/introduction-to-genetic-algorithm/
[9] V. Abhinaya, K. Sahithi, and K. Akaanksha, “Online Application of Automatic Time-Table Generator,” in International Research Journal of Engineering and Technology, vol. 6, no. 2, pp. 1028-1301, February 2019.
[10] https://www.includehelp.com/algorithms/n-queens-problem-and-solution-using-backtracking-algorithm.aspx