

# Online Application of Automatic Time-Table Generator

V. Abhinaya<sup>1</sup>, K. Sahithi<sup>2</sup>, K. Akaanksha<sup>3</sup>

*1,2,3B.Tech Student, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*

\*\*\*

**ABSTRACT** - Professional colleges have different streams of courses and each has its own structure of syllabus which contains diverse subjects. In these colleges, faculty are teaching different subjects in different semesters and also within same semester faculty are handling two different subjects. The major challenge is that the time table required to schedule according to the faculty provided time slots where timetables are prepared in such a way that faculty timings do not overlap. The time table does not overlap with their other schedules and these timetables efficiently utilized by faculty. In this work, we develop the application of time table which can be automatically generating time table according faculty available time slots. This system provides benefits to the faculty need not worry for time clashes; a human do not need to perform permutation and combination and they can concentrate on other activities rather than wasting time by generating Time-Table. This system gives efficient time table generated according to professional college requirement.

**Keywords:** colleges, time table, faculty, courses, system

## 1.INTRODUCTION

Time table generation [1] is tedious job for educationalist with respect to time and man power. This system allows creating time table automatically. Our automatic time generator application will help to create time tables which save the time and reduce the burden to the people who are continuously working on time tables. This automatic generated timetable application reduces the complication of manually setting and managing Timetable. In implementation result we are utilize resource scheduling to decrease the difficulties of producing timetable. Our proposed method integrates a numeral of approach, intended to advance the operativeness of the explore operation. The system will take various inputs like number of subjects, teachers, and workload of a faculty, semester, and precedence of subject. By applying on these contributions, we are able to generate probable time tables for operational days of the week for faculty.

Although majority college organization work has been mechanized, the lecture timetable preparation is still commonly done by hand due to its inherent difficulties. The physical lecture-timetable preparation demands significant time and efforts. The manual lecture-timetable scheduling is a limitation fulfillment problem in which we find a result that satisfies the given set of constraints. There have been numerals of approaches made in the earlier period to the difficulty of constructing timetables for colleges and schools. Timetabling problems may be solve by diverse methods inherited from operation study such as graph coloring, local search measures such as tabu search, simulated annealing, genetic algorithms or from backtracking based constraint fulfillment handling. In this work, timetable problem is formulated as a constraint fulfillment problem and we proposed a realistic timetable algorithm which is capable of taking care of both hard and soft constraints. It is a complete time table solution for Colleges which help to overcome the challenges in manually constructing the timetable.

It is large and highly constrained, but above all the problem differs greatly for diverse colleges and learning institutions. It is hard to write a universal agenda, fitting for all possible timetable problems. Even though manual creation of timetable is sustained, it is still universal, because of the lack of suitable computer programs.

There exist a lot of diverse timetable problems such as:

- University Timetable
- Exam Timetable
- School Timetable
- Sports Timetable
- Worker Timetable

Moreover, there exist a lot of problem solving methods, which typically use the concept of customary optimization algorithms such as genetic algorithms, Backtracking, Constraint Logic Programming.

In recent years two major approaches appear to have been victorious.

- Local Search Procedures
- Constraint Programming (CP)

Probabilistic method used for similar to the global optimum of a given function. Purposely, it is a metaheuristic to fairly accurate global optimization in a huge search space. If there is a condition where we want to maximize or reduce something, our problem can likely be tackle with simulated annealing.

**Search Panel:** The system will have an easy to use to search panel to search according his need on the time tables stored in the database. The system would give the response to the user quires in the proper format and errors messages will be shown properly to tell user about his mistakes and to guide him/her for proper use of the system.

### Constraints

- The system would run only on the windows operating system.
  - The system may give unexpected errors if the input is not proper.
  - The system may work very slowly if the hardware is of very low configuration.
- 
- ▶ Security of data
  - ▶ Ensure data accuracy
  - ▶ Greater efficiency
  - ▶ Minimum time needed for various processing

## 2. RELATED WORK

EA (Evolutionary algorithm) [2] is used to solve a large limitations based university timetable issues. In their approach Heuristics and context-based reasoning methods are utilized for achieving realistic timetables in minimum time. A combinatorial optimization problem [3] developed to solve the university timetabling problem where a set of measures has to be planed into time periods and established into suitable rooms. Manually generated timetables [4] is maintained, it is universal problem due to lack of suitable methods implemented.

A practical timetabling algorithm [5] is developed and capable of handling all constraints efficiently where each faculty and student can access their timetable for a given semester but they will not able to edit. Heuristic [6] method proposes a general solution for the timetabling though, this method works from the faculty point of view which includes faculty available.

## 3. EXPERIMENTAL ANALYSIS

For implementation purpose we use java script and MySQL database utilized.

# TIME-TABLE

Year :

Semester :

No of hours :

No of faculty:

---

Subject 1:

faculty :

No of hours in a week:

Subject 2:

faculty:

No of hours in a week:

Subject 3:

faculty:

No of hours in a week:

Subject 4:

faculty:

No of hours in a week:

Fig 1. Our input page where we enter details to generate the time table.

For implementation purpose, we consider only for a single section in a department .And 8 subjects, each subject has its own corresponding faculty.

Let us presently consider the following subjects:

- Web technologies
- Theory of computation
- Operating system
- Software engineering and object oriented analysis and design
- Python programming
- Internet of things
- Management science
- Quantitative aptitude

And they are 4 labs (web technologies, operating system and case tools, python programming, and internet of things). And two more phases in the schedule related to technical seminar and group project.

This is our basic input page. This page contains Subject name and faculty name and number of hours for each subject. The developer has to enter the subject name to know which subject has to be allotted and the faculty name to know which faculty is going to deal that particular subject. He should also enter the number of periods in a week . This tells us how many hours should be allotted to that particular subject in a week. For example, a core subject like java should at least have 5 periods in a week and for a subject like English 2 or 3 hours are enough. So we will come to know about it from this particular field. Here as were are considering the time table only for one section we by default taking 8 subjects. The same details should be filled for all the 8 subjects.

The same details like lab name and faculty name and number of hours for each lab should be filled for the labs also. Here as we are considering the time table only for 1 section we by default talking 2 labs for 2 core subjects. The faculty may be different for theory and lab for same subject. So it is compulsory to write the details of the faculty name again even though the subject is same. Once all these details are filled, the developer or user should click on either submit button or reset button. The submit button sends the details to the code and prints the output in another tab. If you click on reset button, the information which was entered till then will be lost and you can enter the correct details again. We click on reset only if the entire details are wrong and we want to erase all the details.

|                                |   |
|--------------------------------|---|
| <b>Subject 2:</b>              | <input type="text" value="TOC"/>          |
| <b>Faculty name:</b>           | <input type="text" value="Kranthikumar"/> |
| <b>Already allotted hours:</b> |   |
| <b>Monday:</b>                 | <input type="text" value="1"/>            |
| <b>Tuesday:</b>                | <input type="text" value="None"/>         |
| <b>Wednesday:</b>              | <input type="text" value="3,5"/>          |
| <b>Thursday:</b>               | <input type="text" value="1,2,3,4"/>      |
| <b>Friday:</b>                 | <input type="text" value="None"/>         |
| <b>Saturday:</b>               | <input type="text" value="7"/>            |

Fig 3. The user interface where we entering the faculty details

This shows us the basic allotment of our time table. It just works in the same way as N-Queens algorithm with a few changes. First we consider one subject and fill in the table so that not more than 2 classes should be there on each day. Next we consider the other subjects. If there is no exact time slot available, backtracking will take place. Here we allotted the schedule for WT and TOC. Then the next subjects will also be allotted in the same way. This is the basic idea of our code in the project. This is how our final time table will look like. This will be opened in another tab on our browser. So that, it will be easy for us to access.

| Day       | I                    | II             | III            | Column2        | IV            | V              | VI             | VII            | VIII           |
|-----------|----------------------|----------------|----------------|----------------|---------------|----------------|----------------|----------------|----------------|
|           | 09.30 to 10.20       | 10.20 to 11.10 | 11.10 to 12.00 | 12.00 to 12:45 | 12:45 to 1.30 | 01.30 to 02.15 | 02.15 to 03.00 | 03.00 to 03.45 | 03.45 to 04.30 |
| Monday    |                      | WT             |                |                | TOC           |                |                |                | WT             |
| Tuesday   | WEB TECHNOLOGIES LAB |                |                |                |               |                | TOC            | TOC            |                |
| Wednesday |                      |                |                |                |               | WT             |                |                |                |
| Thursday  |                      |                |                |                |               |                |                |                |                |
| Friday    | WT                   | WT             |                |                |               | TOC            |                |                |                |
| Saturday  | TOC                  |                |                |                |               |                |                |                |                |

Fig 4. Individual faculty time table who having two subjects

| Day       | I                    | II             | III            | Column2        | IV            | V                                  | VI             | VII               | VIII           |
|-----------|----------------------|----------------|----------------|----------------|---------------|------------------------------------|----------------|-------------------|----------------|
|           | 09.30 to 10.20       | 10.20 to 11.10 | 11.10 to 12.00 | 12.00 to 12:45 | 12:45 to 1.30 | 01.30 to 02.15                     | 02.15 to 03.00 | 03.00 to 03.45    | 03.45 to 04.30 |
| Monday    | Web Technologies LAB |                |                |                | LUNCH         | TOC                                | QA             | MS                | WT             |
| Tuesday   | QA                   | OS             | WT             | LUNCH          | SE & OOAD     | IOT                                | IOT            | PP                | PP             |
| Wednesday | PP LAB               | PP LAB         | IOT LAB        | IOT LAB        | Lunch         | OS                                 | TOC            | Technical seminar |                |
| Thursday  | PP                   | PP             | IOT            | IOT            | Lunch         | OS AND CT LAB                      |                |                   |                |
| Friday    | WT                   | WT             | OS             | LUNCH          | MS            | SE & OOAD                          | TOC            | GROUP PROJECT     |                |
| Saturday  | TOC                  | SE & OOAD      | MS             | OS             | LUNCH         | SPORTS & EXTRA CIRCULAR ACTIVITIES |                |                   |                |

Fig 5. Weekly subject wise time table

#### 4. CONCLUSION

Generally, this system can be considered a useful system since it helps the lecturer to improve their process of preparing the timetable. By providing support through the Timetable Management system, the usage can be increased to any faculties. If the system is successfully upload to host, to assist administrator, lecturer and student on how to use the system.

#### 5. FUTURE ENHANCEMENT

In our system, there are some problems those are User has to format it a bit after it is prepared. The transaction are executed in off-line mode, hence on-line data for room Student capture and modification is not possible. In future work, we will overcome these disadvantages by using fuzzy logic approaches.

#### 6. REFERENCES:

- 1) Parkavi A, et al., "A STUDY ON AUTOMATIC TIMETABLE GENERATOR", INTERNATIONAL JOURNAL OF SCIENCE AND INNOVATIVE ENGINEERING & TECHNOLOGY, MAY 2018 ISSUE VOLUME 5 ISBN No. 978-81-923607-3-7.
- 2) Dipti Srinivasan Tian Hou Seow Jian Xin Xu "Automated timetable generation using multiple context reasoning for univer-sity models", 2002 IEEE conference.
- 3) AnujaChowdhary,PriyankaKakde, ShrutiDhoke, Sonali Ingle, RupalRushiya, Dinesh Gawande "TIMETABLE GENERATION SYSTEM" A paper published in IJCSMC Vol. 3, Issue. 2, February 2014.
- 4) M, S., & Pranav Kiran Vaze, P. M. (MAY 2017). Automatic Time Table Generator. International Journal of Advanced Research in Computer Science and Software Engineering, 7(5), 8.
- 5) AnujaChowdhary "TIME TABLE GENERATION SYSTEM" .Vol.3 Issue.2, February- 2014, pg. 410-414
- 6) Anirudha Nanda "An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach". International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012.