JSS MAHAVIDYAPEETA

JSS SCIENCE AND TECHNOLOGY UNIVERSITY

JSS TI CAMPUS, MYSORE-570 006

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ADVANCED TIME TABLE GENRATOR BASED ON PERSONAL PREFERENCES

Synopsis submission for the partial fulfillment of the course

JAVA PROGRAMMING (EC663)
EVENT II & IV

Submitted by

Roll No.	USN	NAME	Sem	Section	Marks Awarded
34	01JST17EC062	PANNAGA S	6	B	
18	01JST17EC034	G N RANJAN	6	B	

Under the guidance of

Supreetha M
Assistant Professor
Department of Electronics & Communication Engineering
JSSSTU, Mysore
2019-2020

Educate Elevate Enlighten

JSS Mahavidyapeetha

JSS Science And Technology University

(Established Under JSS Science and Technology University Act No. 43 of 2013)

(Formerly Known as SJCE)

JSS SCIENCE AND TECHNOLOGY UNIVERSITY MYSURU

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

LTE-1 : Synopsis presentation and evaluation (20 marks)

Group No :12	Sem : VI	Section :B	Period of semester: Even -2019-2020	Faculty :MS

Title of the Project : Advanced time table generator based on personal preferences

Sl. No.	USN	NAME	Problem Identification (Max: 10 Marks)	Presentation (Max: 5 Marks)	Viva (Max: 5 Marks)	Total (Max: 20 marks)
1	01JST17EC062	PANNAGA S				
2	01JST17EC034	G N RANJAN				

Date of Submission :

Remarks :

Signature of Faculty :

Instructions for submission

1. Prepare 10 -20 pages of report
2. Font size: headings 16, sub headings 14, content 12 – Times new roman (font)
3. Direst download from website not allowed
4. Simulation task includes, introduction, writing codes, flow charts, results, graphs/plots,calculations, verification and conclusion.

CONTENTS

INTRODUCTION

Generating a timetable is a time consuming task and is done manually in more than 95% of universities. The process of designing, scheduling and executing a time table in universities is a hectic job for the time table committee. Although most of the college administrative work has been computerized, yet the lecture time table scheduling is still mostly done manually due to its technical difficulties. This process, that is the manual scheduling of lecture - timetable requires considerable time and effort to be put by the time table generating committee. Timetabling is known to be a non-polynomial complete problem i.e. there is no known efficient way to locate a solution. Also, the most striking characteristic of non-polynomial - complete problems is that, no best solution to them is known. Hence, in order to find a solution to a timetabling problem, a heuristic approach is chosen. This heuristic approach, therein, leads to a set of good solutions (but not necessarily the best solution).

1.1 MOTIVATION

In a general educational timetabling problem, a set of events (e.g. courses and labs, etc) are assigned into a certain number of time slots (time periods) subject to a set of constraints, which often makes the problem very difficult to solve in real-world circumstances . In fact, large-scale timetables such as university timetables may need many hours of work spent by qualified people or team in order to produce high quality timetables with optimal constraint satisfaction and optimization of timetable's objectives at the same time. Therefore in order to avoid all such mental stress on the time table committee, we propose an algorithm to generate a timetable that is housed inside a GUI , a graphical user interface in order to increase the dynamic and give freedom and ease of understanding by any member of the time table committee. While setting a timetable, importance is given to effective utilization of resources such as the classroom,the teacher, etc. This becomes a very tedious task which needs to be addressed at least once a year by every academic institute. Most institutes deal with this problem manually, i.e. a trial and error method is used to set a timetable.

1.2 PROBLEM STATEMENT

The need of the hour is an algorithm to generate a time table for a specific class, that takes in user defined constraints, here user being the time table setting committee or the faculty members, and based on the seniority of the lecturers, a time table is generated and is conviniently viewed on a graphical user interface (GUI).

1.3 OBJECTIVES

The main objectives of this JAVA program is as follows :

1. To create an algorithm that takes in multiple constraints such as :
    a) The number of teaching hours per week per subject by each individual lecturer
    b) The number of labs and the number of batches that the class should have
    c) The time slot division , I.e., the teaching hours (6hrs) into morning and afternoon sessions.
2. To build a GUI platform that is driven by the algorithm that we design
3. To ease out the workload of the timetable setting committee
4. To provide more flexibility to the lecturers as a whole

CHAPTER 2

LITERATURE SURVEY

Research papers, articles and journals were studied and the literature survey was conducted to draw the conclusions.

2.1 PREVIOUS RESEARCH

[1] Automated College time table generator : Many colleges use manual way of preparing timetables with large number of students is very time consuming. This usually ends up with various courses clashing this may be either at same room or with same teachers having more than one course at a time. These are just due to common human errors which are very difficult to prevent in the processes.

[2] A study on time table generator : time table generation is tedious job for educationalist with respect to time and man power. Providing a automatic time table generator will help to generate time table automatically. Proposed system of our project will help to generate it automatically also helps to save time. It avoids the complexity of setting and managing Timetable manually. In our project we are going to use algorithms like genetic, heuristic, resource scheduling to reduce these difficulties of generating timetable. These algorithms incorporate a numeral of strategy, aimed to improve the operativeness of the search operation.

2.2 SUMMARY OF LITERATURE REVIEW

Several time table generating hypothesis and heuristic approaches were done that could generate a time table according to certain constraints.

Table 2.2.1 Summary of literature review

Sl.No	Reference	Work done	Drawbacks
1	[1]	A hypothesis is proposed that takes in user defined constraints such as classrooms, teachers available, etc.	Does not talk anything regarding the lab allotment
2	[2]	Proposes to use genetic, heuristic algorithms to approach the problem.	Discusses the approach but not suitable for our university constraints.

CHAPTER 3

METHODOLOGY

The proposed solution basically works as described in the following sections by taking in various constraints from each of the faculty members and the time table setting committee as well.

3.1 PROPOSED BLOCK DIAGRAM

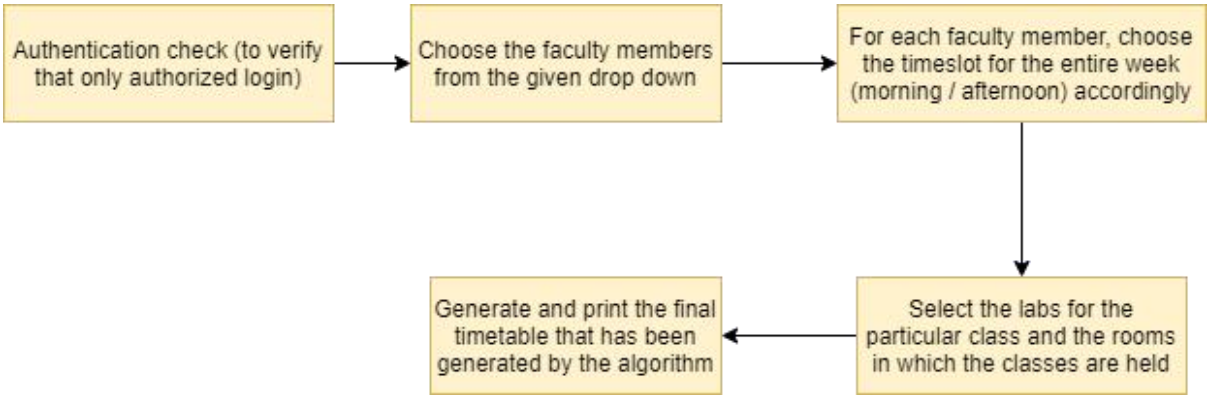

Fig 1.1 : Methodology

3.2 FLOW CHART AND REQUIREMENTS

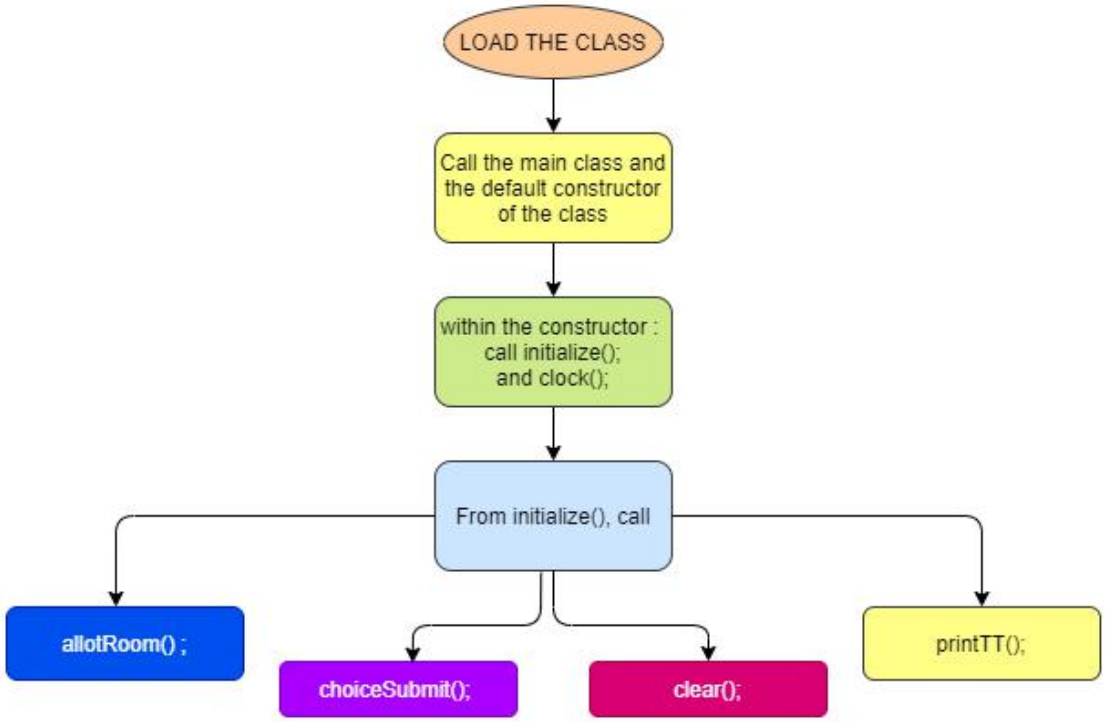

Fig 1.2 : flow chart depicting the overall run

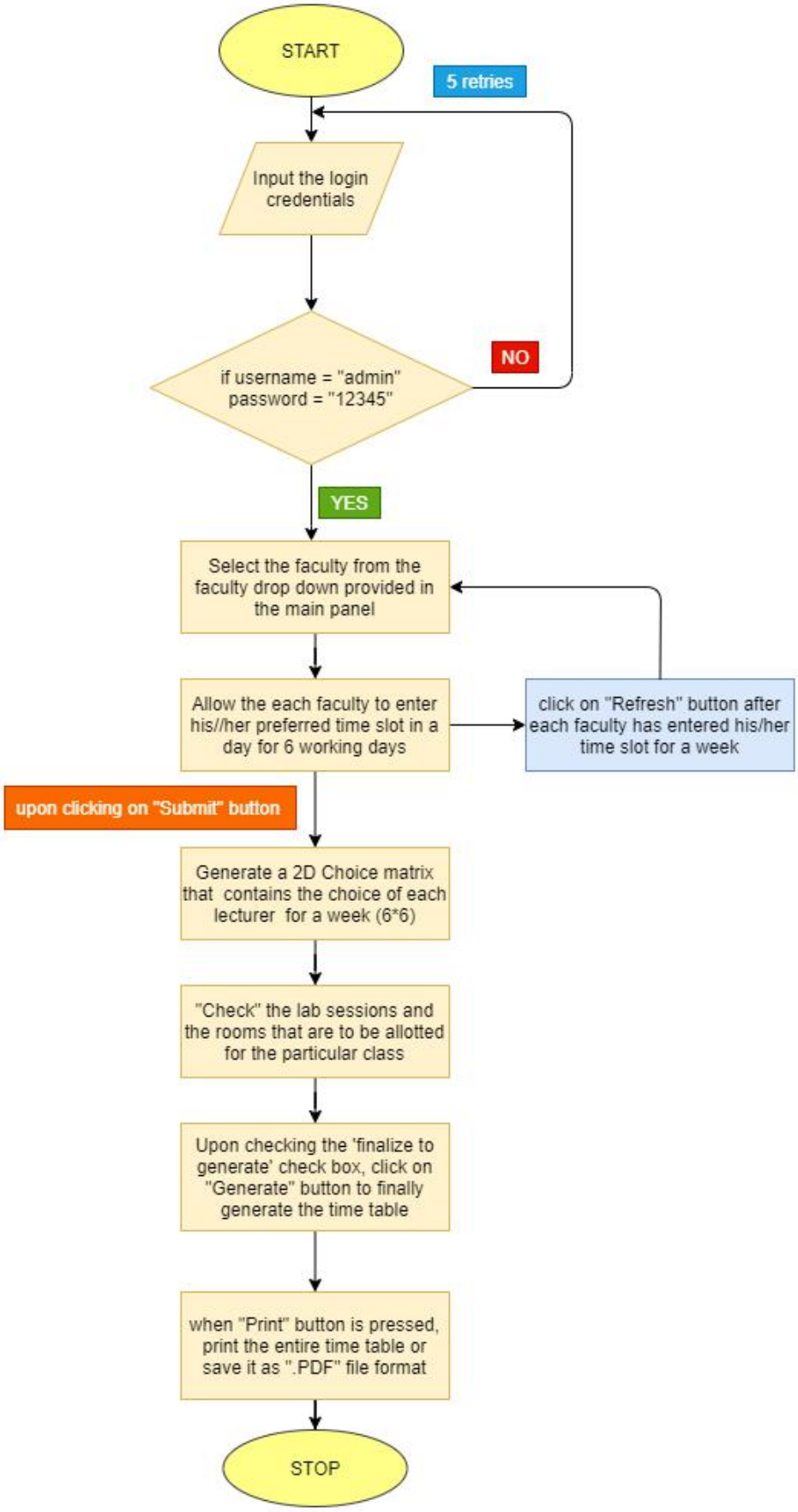

Fig 1.3 : Detailed flow chart of the algorithm

The primary requirement is the “javax.swing” package :
In general Swing is not thread safe. All Swing components and related classes, unless otherwise documented, must be accessed on the event dispatching thread. Typical Swing applications do processing in response to an event generated from a user gesture. For example, clicking on a JButton notifies all ActionListeners added to the JButton. As all events generated from a user gesture are dispatched on the event dispatching thread, most developers are not impacted by the restriction.

As the flow chart depicts, it first takes the user to the login window, where it asks for the user’s , that is the authorized personnel the credentials for login, and only lets in to go to the next window if they’ve entered the correct credentials. Inside the main panel, the faculties are selected from the drop down provided, and 2 radio buttons for each day , for 6 working days have been put for the time slot selection that is either 7:30am - 1:30pm or 11:00am to 5:30pm. Then the user has to select any 4 of the given time slots. After each faculty’s time slot selection, the user has to press “Ok” and “Refresh” button in the main panel to ensure that the slots are cleared and are ready to be entered for a new faculty member from the drop down provided. After all the 6 faculty members are entered with their choices, the user need to click on “Submit” button. When the “Ok” button is clicked, a choiceSubmit( ) function is called, where it inputs a 1D array of string type to store which time slot the faculty has booked. (7:30 am - 1:30 pm as slot 1 , and 11:00 am - 5:30 pm as Slot 2). The 1D array is unique to each faculty member, and finally, when “Submit” button is clicked, Generate( ) function is called which converts this 1D String array into a 2D string array where in the rows represents the each of the faculty members , and the columns represent the days of the week (6*6 matrix) , and the entries in this matrix would be “1, 2 or 0”, which represents slot 1, slot 2 , or no class for that day. This 2D string array is passed as one of the parameters to printTT( ) function, where it scans the entire choice matrix (6*6) matrix that is discussed earlier, and then checks if the choice[i][j] = 1 or 2 or 0.

Before we begin to assign the subjects based on the choice matrix, we assign the labs sessions first as it demands 3 consecutive slots. That is either from 7:30am - 10:30am or 11:00am to 1:30pm or 2:30pm -5:30pm, since once we begin assigning the subjects, there might be a possibility of not getting a continuous 3 slots to allot a lab session. Therefore, we assign the labs first. The lab assignment is also done completely randomly. The following flow charts exactly convey the algorithm explained above.

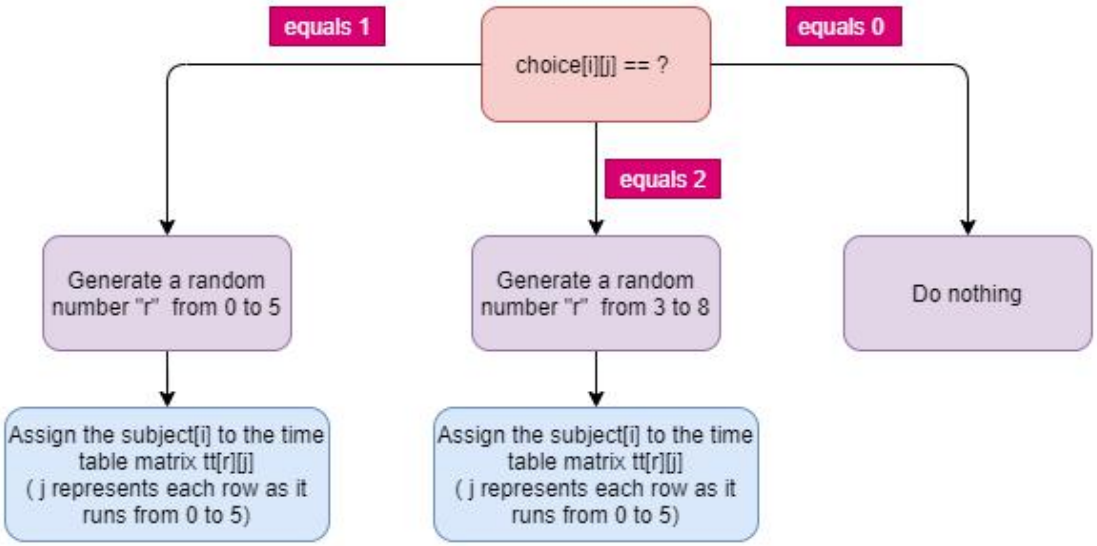

Fig 1.3 : flow that represents choice matrix generation

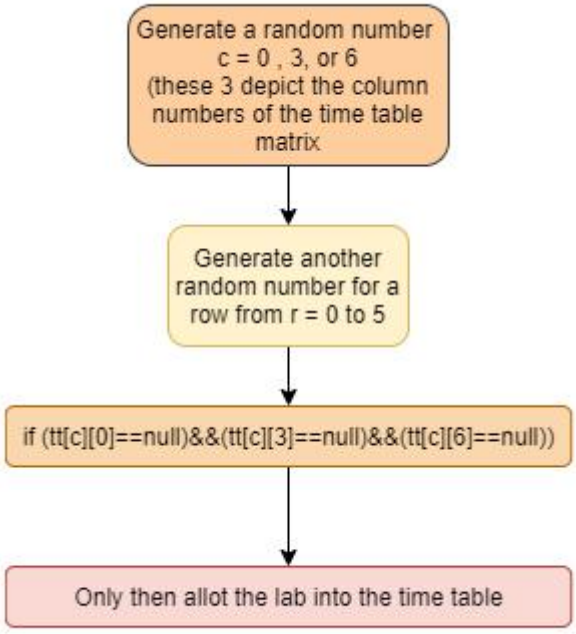

Fig 1.4 : flow that represents the lab session allotment

The algorithm is designed in such a way, such that the allotment of subjects into the timetable is based on the seniority order that is the 1st subject that is going to be allotted into the time table would be the senior most faculty, and would be having the most years of experience amongst the list of faculty members present.

CHAPTER 4

PLANNING AND FEASIBILITY OF WORK

Planning of the work was done and a feasibility analysis was conducted. The following conclusions were derived. A few anticipated bottlenecks and the method of overcoming them are discussed in this chapter.

4.1 ANTICIPATED BOTTLENECK

There would be a quite a few pit falls that we may face in this algorithm, such as (1) when a lecturer wishes to enter more than 4 hours of teaching slots per week, (2) when the admin fails to enter the correct credentials into the login window, (3) or, in a particular case, for example, say 4 lecturers have been opted for time slot 1, and by co incidence, a lab has been allotted in the time slot 11:00am to 1:30pm time frame. So, the remaining slots that are free to be able to accommodate the necessary lecturers would be only 3 , that is from 7:30am to 10:30am, but there are 4 lecturers that are wanting this particular time slot.

We over come each of the problems discussed above as follows :

(1) We initialize a count variable initially assigned to 4, and after each click on radio button (any) the count is decremented by 1, and upon clicking “Refresh” button, the count is again reassigned to 4.

(2) The second problem is also overcome by using a try variable for type int, that is initialized to 5, and then we run a separate thread in order to create a 5 second delay , only after the user has failed to enter even after 5 times. In order to create a delay of 5 seconds, we call the Thread.sleep class with try - catch blocks.

(3) The third problem is overcome by again initializing a separate count variable that is initialized to 10. Then this is decremented after the command enters the if - else block to check whether the choice[i][j] is equal to 1 or 2 or 0. When this count is 0 , it enters another while loop where it randomly generates two numbers in the range r1 = 0 to 5 and r2 = 0 to 8, and checks if tt[r1][r2] equal to null, and assigns the subject[i] to that particular time slot. This is done so that all the lecturer’s teaching hours is 4 in our university for a week. Therefore, this ensures that all the faculty members get 4 time slots per week as per they desire.

4.2 NOVELTY OF THE PROPOSED WORK

The graphical user interface developed is unique to our university, and is based on our lecturer's teaching hour flexibility and the constraints they have as per the higher management. Also, the GUI program gives apt results based on the number of teaching hours per faculty per week, and the lab sessions assigned including the number of batches for a particular batch. The generalized way of assigning the batches and the labs is also taken care off , such as if there are 3 labs and 4 batches, the maximum of the 2 numbers are to be considered for knowing the number of lab sessions per week. As per our university, it is 4 batches and hence there are 4 lab sessions per week, so that all batches get equal opportunity to have the lab sessions as per the course designed.

Also, the flexibility of selecting the time slots 7:30am to 1:30pm or, 11:00am to 5:30pm is given to each individual faculty. Since, the algorithm is designed in such a way that the assignment of the subjects into the time slots of the time table is done according to the seniority of the lecturer's, even in an ambiguity situation where a junior faculty member could not be accommodated in his/ her timeslot , they are ensured of getting 4 slots of teaching hours per week, and no 2 consecutive classes per day or in a day is also taken care off which is essential for students to take in.

4.3 ADVANTAGES AND APPLICATIONS

The advantages of this program is to largely decrease the amount of time and effort put by the time table setting committee into making a timetable that doesn't clash , accommodate everyone's schedule, relief from mental stress and strain.

The second advantage is that we are giving the faculty members to choose the time slot according to their convinience.

Applications of this program would be in any university, college, schools, but some of the constraints have to be altered according to the particular university, college, or school, where it is put in use.

4.4 PERT CHART

TASK	22-Jan	15-Feb	10-Mar	25-Mar	31-Mar	10-Apr
Final title selection						
Idea presentation						
Presentation in brief						
Synopsis submission						
Final submission						

REFERENCES

[1] International Journal of Scientific & Engineering Research Volume 9, Issue 4, April-2018 ISSN 2229-5518 IJSER © 2018 : Automated college timetable generator

[2] A, Parkavi. (2018). A STUDY ON AUTOMATIC TIMETABLE GENERATOR. International Journal of Innovative Research & Growth. May.

[3] M, Saritha & Vaze, Pranav & Pradeep, & R, Mahesh. (2017). Automatic Time Table Generator. International Journal of Advanced Research in Computer Science and Software Engineering. 7. 204-211. 10.23956/ijarcsse/SV7I5/0234.

[4] R., Mayuri & C., Sunil & N., Sunita & R., Pushkar & Kumavat, Sanjay. (2015). A Novel Approach for Automatic Timetable Generation. International Journal of Computer Applications. 127. 26-30. 10.5120/ijca2015906512.