Project Report

On

Automated System for Timetable Generation

Submitted for partial fulfillment of requirement for the degree of

BACHELOR OF ENGINEERING

(Computer Science and Engineering)

Submitted By

Arpit Sharma

Bhushan Chopade

Rutvik Agrawal

Vinay Dhamale

Sanmay Kawale

Under the Guidance of

Prof. S. P. Ingale

Department of Computer Science & Engineering,

PRM Institute of Technology & Research, Badnera.

2019-2020

Department of Computer Science & Engineering

Prof. Ram Meghe Institute of Technology & Research, Badnera

2019 - 2020

CERTIFICATE

This is to certify that the Project (8KS07) entitled

Automated System for Timetable Generation

is a bonafide work and it is submitted to the

Sant Gadge Baba Amravati University, Amravati

By

Arpit Sharma
Bhushan Chopade
Rutvik Agrawal
Sanmay Kawale
Vinay Dhamale

in the partial fulfillment of the requirement for the degree of ***Bachelor of Engineering*** *in* ***Computer Science & Engineering****, during the academic year* ***2019-2020*** *under my guidance.*

Prof. S. P. Ingale
Guide
Department of Computer Sci. & Engg.
PRM Institute Of Technology & Research, Badnera

Dr. G. R. Bamnote
Head,
Department of Computer Sci. & Engg.
PRM Institute Of Technology & Research, Badnera

External Examiner

_____

ACKNOWLEDGEMENT

With great pleasure we hereby acknowledge the help given to us by various individuals throughout the project. This Project itself is an acknowledgement to the inspiration, drive and technical assistance contributed by many individuals. This project would have never seen the light of this day without the help and guidance we have received.

We would like to express our profound thanks to **Prof. S. P. Ingale** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We would also like to express our gratitude towards **Dr. G. R. Bamnote** (Head of Computer Science and Engineering Department), for encouraging us for our betterment. We would also thank the faculties of the Department of Computer Science & Engineering, for their kind co-operation and encouragement which help us in completion of this project. We owe an incalculable debt to all staffs of the Department of Computer Science & Engineering for their direct and indirect help.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

We extend our heartfelt thanks to our parents, friends and well wishers for their support and timely help. Last but not the least; we thank the God Almighty for guiding us in every step of the way.

Arpit Sharma _____

Bhushan Chopade _____

Rutvik Agrawal _____

Vinay Dhamale _____

Sanmay Kawale _____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

ABSTRACT

Most colleges have number of different courses and each course has a number of subjects. Now there are limited faculties, each faculty teaching more than one subjects. So now the time table needed to schedule the faculty at provided time slots in such ways that their timings do not overlap and the time table schedule makes best use of all faculty subject demands. We use genetic algorithm for this purpose. In our Timetable Generation algorithm we propose to utilize a timetable object. This object comprises of Classroom objects and the timetable for them likewise a fitness score for the timetable. Fitness score relates to the quantity of crashes the timetable has regarding alternate calendars for different classes. Classroom-object comprises of week objects. Week objects comprise of Days, Days also comprises of Timeslots. Timeslot has an address in which a subject, student gathering and going to the address and educator showing that the subject is related. Also further on discussing the imperatives, we have utilized composite configuration design, which make it well extendable to include or uproot as numerous obligations. In every obligation class the condition as determined in the inquiry is now checked between two timetable objects. On the off chance that condition is fulfilled i.e. there is a crash available.

1. INTRODUCTION

Time table scheduling has been in human requirements since all thought of managing time effectively. It is widely used in schools, and other fields of teaching and working like crash courses, coaching center, training programs etc. In early days, time table scheduling was done manually with one single person or some group involved in task of scheduling it with their hands, which takes a lot of effort and time. While scheduling even the smallest constraints can take a lot of time and the cases are even worse when the number of constraints or the amount of data to deal with increases.

In such cases perfectly designed time table is reused for the whole generation without any changes, proving to be dull in such situations. Other cases that cause problem is when the number of employers/workers are weak, resulting in the rescheduling of time table or they need to fill on empty seats urgently. They need to schedule their course to meet the need of current duration and facilities that are available to them. However, their schedule should meet the requirements of the new course additions and newly enrolled students to fresh batches. This may result in rescheduling the entire time table once again for its entire batches and to be scheduled in shortest possible time before the batch courses start.

Another problem that occur when scheduling time table for exams i.e. when multiple batches have their exam on same day, they need to be schedules effectively taking into account all problems related to the facilities that are available to conduct these exams simultaneously.

Even though most college administrative work has been computerized, the lecture timetable scheduling is still mostly done manually due to its inherent difficulties. The manual lecture-timetable scheduling demands considerable time and efforts. The lecture-timetable scheduling is a constraint satisfaction problem in which we find a solution that satisfies the given set of constraints.

A college timetable is a temporal arrangement of a set of lectures and classrooms in which all given constraints are satisfied. Creating such timetables manually is complex and time-consuming process. By automating this process with computer assisted timetable generator can save a lot of precious time of administrators who are involved in creating and managing course timetables.

Generating a timetable is one of the challenging time consuming problems facing institutions belonging to the complete class of problems. Our main challenge is to be able to automatically time table four year associate degree.

Programs courses' so that students belonging to different programs can easily register for courses with no timetable clashes for the semester they are studying for. A general studied paper by investigates examination and course timetabling providing up to-date important information and citations for further research and possible implementations of automated timetabling for use in educational settings. The college lecture-timetabling problem asks us to find some time slots and classrooms which satisfy the constraints imposed on offered.

An evolutionary technique has been used to solve the time table scheduling problem Algorithms, have been used with mixed success. In this paper, we have referred the problem of educational time table scheduling and solving it with resource scheduling algorithm.

Examination is a core activity of any educational institution. In colleges the examinations are carried in the semester method that takes place once in a six month. Thus examinations control system automates the activities carried around the examination processes.

This includes the automation system of timetable generation for the student attending the lectures. This project focuses on developing an timetable management system for the Faculty of Engineering in order to manage the lecture. It will also manage timetable when any teacher is absent, late coming or early going.

Maximum and minimum work load for a Faculty for a day, week and month will be specified for the efficient generation of timetable. It is a comprehensive timetable management solution for College which helps to overcome the challenges in manually setting the timetable. The goal of this work is to develop the database and to implement the module as a system-based application.

1.1.Existing System

In the existing system, each task is carried out manually and processing is a very tedious job. The Organization is not able to achieve its need in time and the results too may not be accurate. Due to all the manual maintenance, there are number of difficulties and drawbacks that exist in this system.

As we know all Institutions or organizations have its own timetable, managing and maintaining these will not be difficult. Considering workload with this scheduling will make it more complex. As mentioned, when Timetable generation is being done, it should consider the maximum and minimum workload that is in a college. In those cases, timetable generation will become more complex. Also, it is a time consuming process. . After a successful preparing in the seating arrangement, a critical work is carried on the prepared one that is typing the information is a paper using the typewriting machine or manually adding information in the computer.

Following are the main problems in current system

- Manual work
- Lots of paper work
- Difficult to recover data in case of loss
- Time consuming
- Allocation of seats is Critical task
- Increased transaction leads to increased source document & hence maintenance becomes difficult.
- If any student or staff entry is wrongly made then the maintenance becomes very difficult.

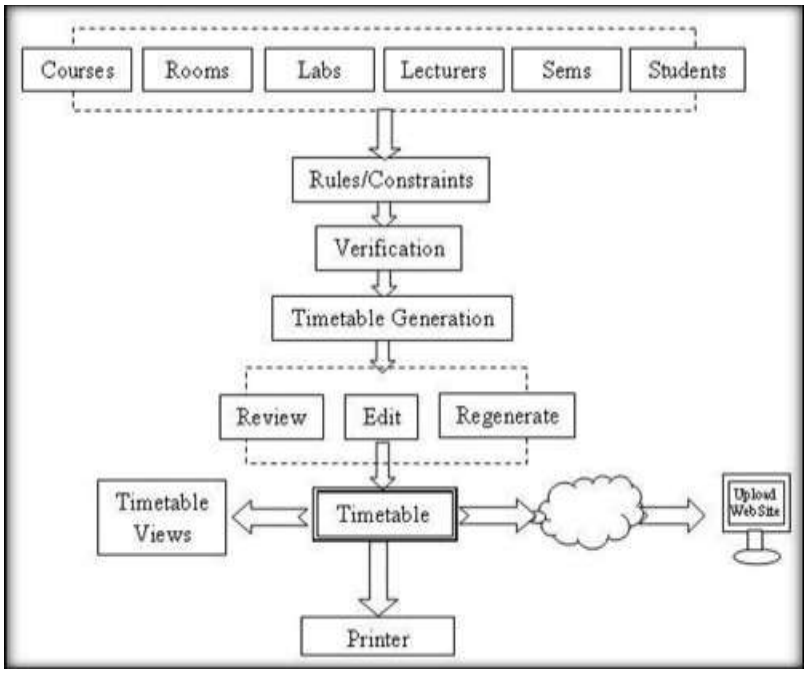


Figure 1.1: General view of Timetable

1.2.Objective of the project

- Reduce manual paper work.
- Easy to manage seating arrangement of exams.
- To view all details of each class seating room.
- To send pre information of seating arrangement to student.
- Exam controller can easily view, Edit, Create exam, seating arrangements of classroom and duty charts of faculty.
- Reduce Exam Controller efforts related to exam.
- Easy to handle this application.
- Less man power consumption.
- Easy future reference.
- Attractive GUI interface.

2. LITERATURE REVIEW

Literature survey represents a study of previously existing material. It described the existing and established theory. Existing system shows where you can fill a perceived gap in the existing theory. The literature survey should be structured in such a way as to logically represent the development of ideas in that field. Timetable Scheduling has been studied from a variety of perspectives. One are of emphasis is class preference, if some person have to write all the data to arrange the class for the student it's critical and takes a lot time. We can use online timetable arrangement to arrange all work done manually.

There exist various problems such as Sports Timetabling, Examination Timetabling, Employee Timetabling and university timetabling. [13]Carter and Laporte (1998) considered different categories to solve the timetabling problem. They are – Cluster method, Sequential method, Meta-Heuristics and Constraint Based method. Meta Heuristics is a higher level procedure which is used to provide good enough solutions for optimization problems. On some class of problems, they do not guarantee a globally optimum solution. This method is used when the classical methods are too slow or fail to give a solution. This is achieved at the cost of optimality and precision for speed. In this paper we consider the following Meta-Heuristic methods.

To Finding a feasible lecture/tutorial timetable in a large university department is a challenging problem faced continually in educational establishments. The approach uses a problem-specific chromosome representation. Heuristics and context-based reasoning have been used for obtaining feasible timetables in a reasonable computing time. An intelligent adaptive mutation scheme has been employed for speeding up the convergence. The comprehensive course timetabling system presented in this paper has been validated, tested and discussed using real world data from a large university.

[9]According to Kong, S. C. & Kwok, L.F. (1999), timetabling system involves a heuristic function to increase the scheduling performance, as well as producing a best outcome. Currently, the well known solutions for the timetabling system are Genetic Algorithms and Memetic Algorithms (Mohd. Dain, A. A., Shaari, N. S. Gom,Y. S. & Bacheck).[9]Hybrid Genetic Algorithms for the vehicle routing problem, which they argues to be faster, more cost-effective and highly competitive than the best-known heuristic routing procedures and solutions. On the other hand, there are many solutions for the timetabling system, while each of them has their strengths and weaknesses.

2.1.Course Timetabling Problem

The course timetabling problem is a kind of personnel scheduling problem that appears in the education domain. However, arranging objects such as pairing a course and a lecturer in the scheduling process can be viewed as assigning a lecturing job for a lecturer to take care of that course. Moreover, arranging a pair of course and lecturer at certain time slots can also be viewed as allocating or assigning resources (teachers and time slots) for an event.

[6]The course timetabling problem has to consider two types of constraints, namely hard constraints and soft constraints (Burke, Jackson, Kingston and Weare, 1997). Hard constraints are essential and must be rigidly enforced to the schedule, while soft constraints are not essential but desirable (Burke and Petrovic, 2002)[6]. Quality of the schedule is measured by an objective function.

The sequential approach selects the events one by one using a certain order, and then places the event into a safe tune slot. In order to prevent a potential conflict, a data structure, like a graph in the graph coloring problem, must be prepared in advance. The vertices of the graph represent the events and the edges connect conflicting events. The cluster approach provides a number of clusters to place a group of conflict events. To construct the schedule, the group is then assigned into a time slot using a certain method.

The constraint based method treats schedule components as variables. It constructs the schedule from an empty set by selecting element by element that comprise of schedule component. The improvement approach selects any arbitrary solution then improves the solution by heuristic methods.

A scheduling activity usually produces a kind of table called a timetable so that some researchers referred the scheduling activity as timetabling activity (Duong and Lam, 2004; Gudes, Kuflik and Meisels, 1990). Relationship between scheduling and timetabling in detail can be found in (Wren, 1995).

2.2.Solution for Generation Algorithms

There are many solution generation techniques that can be used to find a solution for the course timetabling problem such as

2.2.1. Simulated Annealing (SA)

Simulated annealing is a 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. It is frequently used when the search space is distinct. Simulated annealing is a technique for finding a good result to an optimization dilemma. If there is a condition where we want to maximize or reduce something, our problem can likely be tackle with simulated annealing.

Chainate, Thapatsuwan and Pongcharoen (2008) proposed a repair process during neighborhood search process in the Simulated Annealing Algorithm to the course scheduling problem that consider lecturer preference to certain time slots. The aim of this experiment is to set some Simulated Annealing parameters that have better performance for their problems that include initial temperature, final temperature, neighborhood search operators and cooling schemes. The objective function is to minimize the total violation index which is the sum of weighted penalty of the constraint violation indices.

2.2.2. Tabu Search

Tabu Search is a Global Optimization algorithm and a Metaheuristic or Meta-strategy for calculating an surrounded heuristic method. Tabu Search is a parent for a huge relations of derivative approach that establish memory structure in Metaheuristic, such as Tabu Search and Parallel Tabu Search (S, A Kavya Reddy, & K Panimozhi, April 2015)[14].

Amol C. Adamuthe[15] and Rajankumar S Bichkar used a set of heuristic algorithms in a program for solving course timetabling related problems. A Tabu Search procedure had several strategies developed for it and tested, leading to a potent and quick algorithm which produced satisfactory results.

2.2.3. Genetic algorithm (GA)

Genetic Algorithms (GA) was imaginary by John Holland and has described this thought in his book “Adaptation in natural and artificial systems” in the year 1975.

Genetic algorithm is a metaheuristic motivated by the procedure of natural selection that belong to the bigger class of evolutionary algorithms (EA).

Genetic Algorithms are motivated by Darwin's evolutionary theory. GA comes below the class of Evolutionary algorithms that use the principle of natural collection to develop a set of solution towards the best result. It is a search heuristic which generates solutions to optimization problems using technique motivated by natural evolution like mutation, inheritance, crossover and selection (S, A Kavya Reddy, & K Panimozhi, April 2015)[14].

Genetic Algorithms (GA) was introduced by John Holland (Aziz M. A.,2002). It utilizes several iterations to choose the best solution from a set of solutions for a problem (Yingsong, Z. & Kiyooka, S., 1999). Tzafestas . S. G. (1999).

2.2.4. Constraint Based Approaches

Constraint satisfaction problems or CSPs are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations. CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which are solved by constraint satisfaction methods. CSPs are the subject of intense research in both artificial intelligence and operations research, since the regularity in their formulation provides a common basis to analyze and solve problems of many unrelated families. A solution to a timetabling problem using CSP is described as follows: (Deris et al., 1997).

A different approach by Brailsforsd, Potts and Smith (1999) asserts that many scheduling and timetabling problems can be formulated as Constraint Satisfaction Problems (CSP's). They defined and examined basic techniques for solving CSP's, then compared them with other approaches such as integer programming, branch and bound and simulated annealing. As opposed to more developed OR methods the authors viewed CP as being a relatively new area, however they still see that there are problems for which it is competitive. With further improvements to CP methodology being anticipated, it is considered that awareness of CP as a technique for tackling combinatorial optimization problems is important for researchers in this area.

2.2.5. Hybrid Algorithm

Difficulties of scheduling or NP problems in general and the availability some algorithms direct researchers to combine some methods or algorithms as a hybrid algorithm to enhance their algorithm. (Alvarez-Valdes, Martin and Tamarit, 1996) used hybrid algorithm

that construct Spanish school timetabling using hybrid of parallel heuristic algorithm and Tabu search.

[6]Burke, Elliman and Weare (1994) discussed an automatic timetable generation method that used applications such as Graph Colouring and Genetic Algorithms. It was claimed that their method was being implemented at some United Kingdom Universities.

Weare and Elliman[6] showed that several types of constraints may be modeled using the unifying framework of partially renewable resources. Presented are two-phase parallel greedy randomized and genetic methods. An instance generator for the generation of a representative set of instances was also given. Computational results showed that this method solved the instances investigated close to optimality.

Cagdas Hakan Aladag [16] solved job shop schedule by hybridizing evolutionary Algorithm into Tabu Search. Malek Rahoual and Rachid Saad (2006) proposed hybrid approach for Solving Timetabling Problems by combining Genetic Algorithms and Tabu Search. Gunawan, Ng and Poh (2007) used hybrid integer programming, Greedy Algorithm and Simulated Annealing Algorithm to solve course timetable.

3. PROPOSED SYSTEM

The proposed system tends to overcome the problems in the existing system. This system is implemented as an Timetable Control System for manage all controls regarding timetable. With the help of this system, all the manual work done by timetable controller can be made easily with the help of this simple application. The proposed system is a android application represents a system design which reflects all the basics needed for the current system.

The data is stored in backend database instead on paper. The overhead of maintaining large data for the large number of subjects get reduced accordingly. The new system will overcome all the problems that find in the current system.It invokes all base tasks that are now carried out manually, such as the forms transactions and reports which is added advantage.

Description:

- Most colleges have a number of different courses and each course has ‘n’ number of subjects.
- Now there are limited faculties, and each faculty might be teaching more than one subjects.
- So now the time table needed to schedule all the faculty at provided time slots in such a way that their timings do not overlap and the time table schedule will make the best use of all faculty subject demands.
- We use a customized algorithm for this purpose.
- In our Timetable Generation algorithm we propose to utilize a timetable object.
- This object comprises of Classroom objects and the timetable for them likewise a fitness score for the timetable.
- Fitness score relates to the quantity of crashes the timetable has regarding alternate calendars for all the different classes.
- Classroom object comprises of week objects. Week objects comprise of Days, Days comprises of Timeslots.
- Timeslot has an address in which a subject, student gathering going to that particular address and educator showing to the subject it is related will be shown.

- Also further on discussing, we have utilized composite configuration (design), which make it well extendable numerous obligations.
- In every obligation class the condition as determined in our inquiry is now checked between both the timetable objects. On the off chance that condition is fulfilled, there is a crash is available then score is augmented by one.

To make a timetable system generic so that I can work equally well for different School, Colleges and Universities. User defined constraints handling. Ease of use for user of system so that he/she can make automatic time table. Focus on optimization of resources i.e. teachers, labs and rooms etc. Provide a facility for everyone to view timetable. Generate multiple useful views from time table.

Outcomes depends on:-

3.1.Interface for Input

The system will be having an easy to use and interactive interface to enter all the inputs like the teacher name, the data for the rooms and data for the labs and the data for subject.

3.2.Database Capabilities

The system will have well designed database to store all the information which will be entered in as the input. Separate database maintaining basic information, subjects, teachers, batches and their associations and other details Database for holding generated timetable and for storing required timetables.

3.3.Processing Capabilities

The system will have algorithms to process all the data present in the database and keeping in view the various constraints like that a teacher should not have two consecutive lectures/labs, students have minimum one hour gaps, proper rooms are allocated for the lectures and tutorials, labs are used optimally so that they are used for the maximum possible time, it will generate the time table.

3.4.Salient Features of the System

- Automatic Timetable manger is a PHP based software used to generate timetable automatically.
- Will help you to manage all the periods automatically nd also will be helpful for

faculty who will get timetable in their phone as a notification.

- It will also manage timetable when any teacher is absent, late coming or early going.
- Proposed system will help to generate it automatically also helps save the time.
- There is no need for Faculty to worry about their timetable.
- It is a comprehensive timetable management solutions for Colleges which help to overcome the challenges in current system.

3.5.Advantages of proposed system

- **Data Consistency**

  Each module creates a data in a tabular format. Once the data is feed in a system, cache module generates the data as per the inputs provided.

- **Better Data Accessibility**

  By providing minimum inputs, the system provides the necessary data.

- **Less Effort**

  Since it is a desktop application, the controller can easily maintain all the exam related work. It reduces man power and duplication of data. It is easy to lookout the previously filled data.

- **Compactness**

  The system saves lot of paper work and therefore saves the time.

- **Easy to handle and understand**

  All the data is available in a single database, need to maintain a computer system instead of lot of papers.

- **No loss of data**

  Data can be secured and data leakage can be avoided.

- **Less Cost**

  Software like android, PHP is freely available.

4. SYSTEM OVERVIEW

Further to the study of data flow in timetable production, we are able to propose architecture for implementing the system.

4.1.Architecture for Timetable Production

Abstractly speaking, software architecture describes the elements of a system. It also shows the interactions between these elements, the models governing its composition and the constraints of these models. Generally, when facing a complex problem, the best approach is to break it down into parts that become easier to solve with simple solutions. Then, when we combine all these small solutions, we can find the solution to our complex problem.

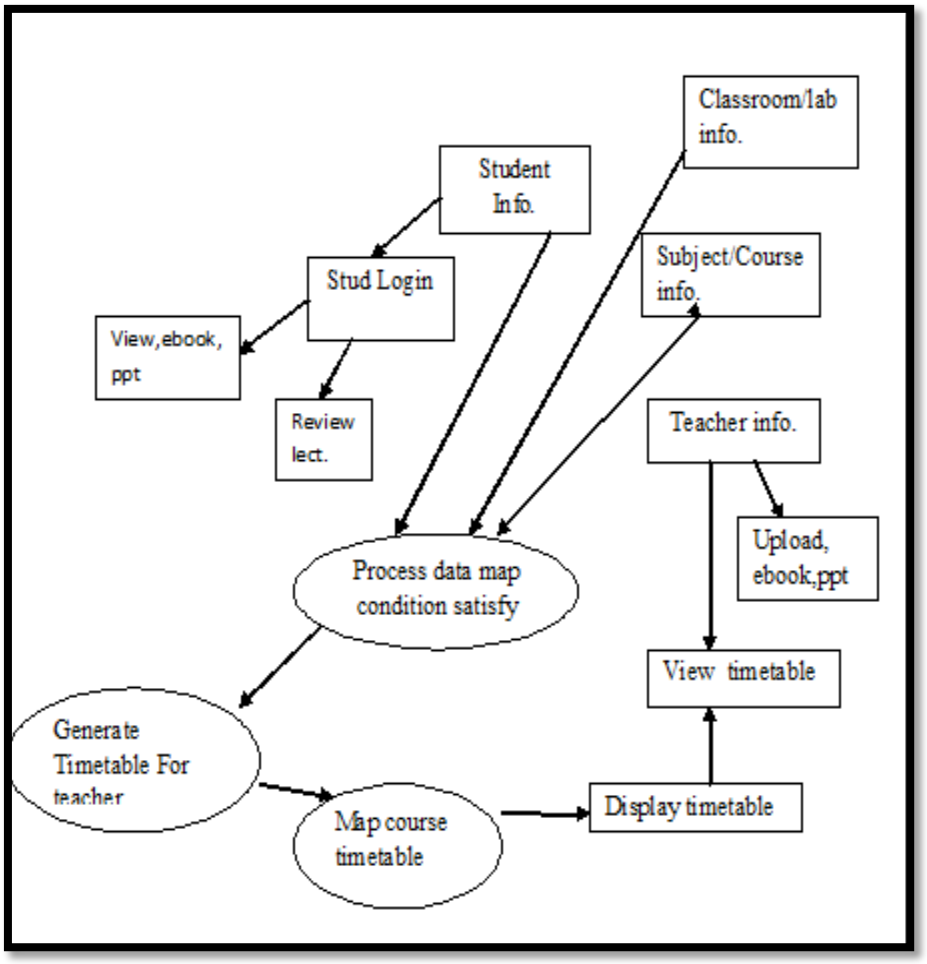


Figure 4.1: System Architecture

Figure 4.1 contain following things:

1) Enter Classroom Details, teacher details, Lab Details as classroom information.
2) When Student details as student information, Subject details as subject information.
3) Teacher details as teacher information.
4) Process Data map and Conditions to satisfy: Process all the information as per Graph Coloring Algorithm. try to satisfy constraints accordingly depending upon priorities associated. Course Information can be viewed at this stage.
5) Generate timetable for teachers: Generated timetable satisfying hard constraints is to be viewed by teachers and define timetable that will try to satisfy maximum soft constraints.
6) Map course Timetable: After defining a timetable it will try to map timetable for whole course.
7) View Timetable: Timetable will be available for display, update for staff and teachers.

4.2.Flow Chart of Time Table Scheduling Process

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. This diagrammatic representation illustrates a solution model to a given problem as in Fig.4.2:

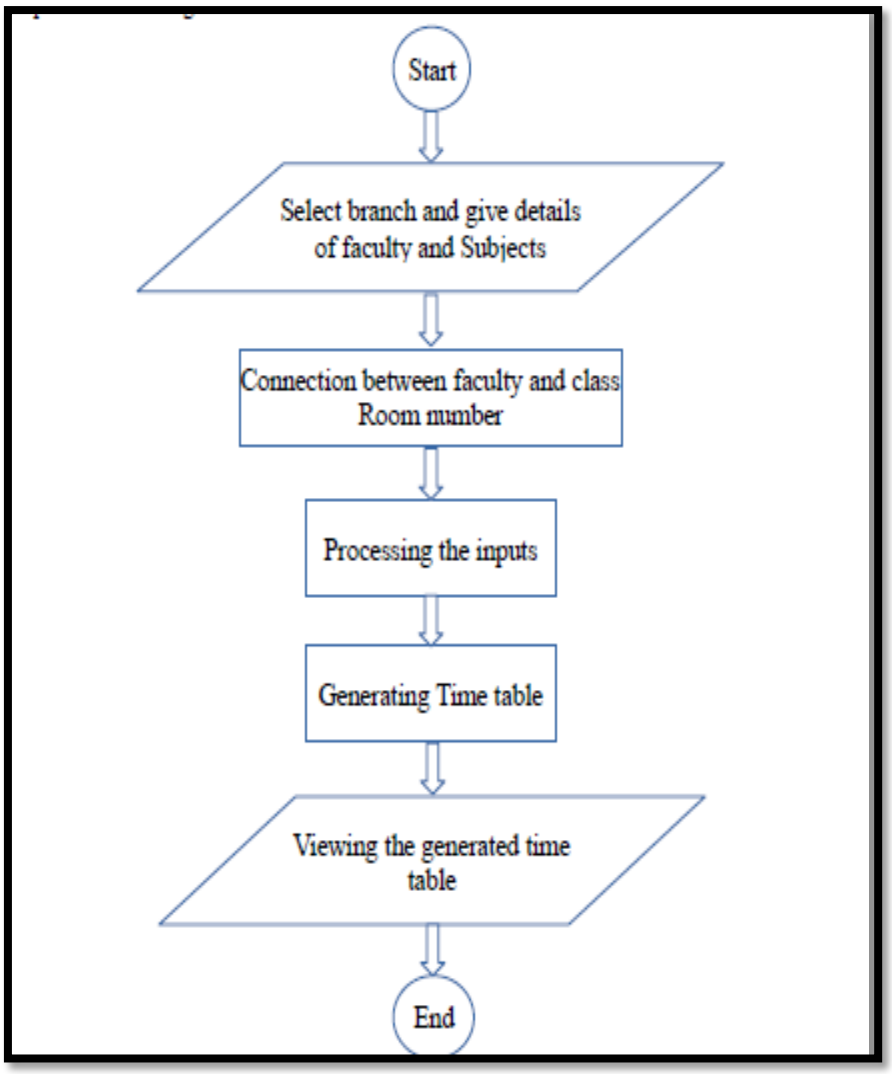


Figure 4.2: Data flow diagram for Timetable Generation

5. MATHEMATICAL MODEL

5.1.Constraints

There are a variety of constraints to be satis the time to instantiate variables about time slots and classrooms. Constraints are the most vital aspect of any scheduling problem. These are the various restrictions involved in creating a schedule. Based on satisfaction of these a schedule can be accepted or get rejected. Depending on the degree of strictness, constraints are broadly classified into- Hard and Soft Constraints.

5.1.1. Hard Constraints

Hard Constraints are those essential conditions which must be satisfied to have a legal schedule. If any of the hard constraints cannot be placed successfully by a schedule, then such a schedule is rejected. For example, no two subjects having common students can be scheduled in the same time-slot, courses cannot be assigned to more than maximum number of available time-slots or periods. In those scheduling datasets which involve resources like teachers and classrooms, no courses can be scheduled to the same classroom at same time-slot, more than one course taught by the same teacher cannot be assigned same time of the week.

HC1: A classroom is not assigned to more than one teacher at the same time.

HC2: A teacher cannot teach more than one class at the same time.

HC3: Courses for the similar year-session students of a department cannot take place at the same time.

HC4: The classroom for a course should have enough capacity to take students registered in the course.

HC5: The classroom should be well set of equipment required services for the classes.

5.1.2. Soft Constraints

Soft Constraints on the other hand are those preferential conditions which are optional. Mostly, it gets difficult to incorporate all the soft constraints in a schedule. A schedule is still said to be legal even if it fails to satisfy soft constraints, provided all hard constraints are met.

Soft constraints are less significant than hard constraints, and it is typically not possible to avoid breaking at least some of them. Either timetable technique is functional,

which calculates the level to which a timetable has violated its soft constraints. Some soft constraints are more vital than others, and this is often specified with a priority value.

For example, a teacher may prefer to take practical classes only in the second half, honours and pass classes are preferred to be scheduled in non-overlapping time-slots, etc.

SC1: The teachers are not assigned to time sl in the teacher's prohibited time zones.

SC2: Teachers daily teach hours should be within the acceptable utmost hours.

SC3: As far as possible, classes are scheduled in the lecturer's preferred time zones.

SC4: A lunch break must be scheduled.

SC5: The practical courses are scheduled in morning session, and the theory courses are such session.

SC6: The lecture hours for a course should be scheduled consecutively.

SC7: As distant as possible, classes should be corresponding department's exclusive-use classrooms.

SC8: The classrooms should be allocated in a manner to reduce the distances between adjacent classes and classrooms.

It is coveted for timetables to satisfy all strong and weak constraints. However, it is usually not simply to meet all these constraints. The virile constraints must be satisfied all the times, but impotent constraints can be somewhat sacrificed to find practice timetables. Among the weak constraints, constraints from SC1 to SC6 are imposed on the allocation of slots. Constraints SC7 and SC8 are imposed on the allocation of classrooms. The constraints are arranged in the sequence of importance in the scheduling. For example, if it is unthinkable to find schedules satisfying both SC1 and SC2 simultaneously, it is preferred to select a schedule that satisfies SC1 but SC2 rather than a schedule satisfying SC2 but SC1.

It is describe for timetable to satisfy all hard and soft constraints. It is typically hard to meet all these constraints. Any hard constraint must not be despoiled in any case, but some soft constraint can be sacrificed to find reasonable timetables.

The timetable practice is made more difficult by the fact that so many people are affected by it's outcome.

5.1.3. Temporal Constraints

Temporal Constraints can also be viewed as time-related . Time related constraints are called Temporal constraints.

For example, Computer Science theory and practical classes cannot be scheduled at the same time-slot or period because of common students. Also there must be a fixed number

of theory and practical classes scheduled in a week its outcome.

5.1.4. Spatial Constraints

Spatial Constraints are viewed as space-related constraints. In course scheduling problem, spatial constraints mainly involve classroom related issues. Any educational institute has a fixed number of available rooms with specified capacity. Also, classrooms can be theory or laboratory based. While making schedules, courses having student capacity compatible to the classroom size is an essential condition. Courses which need specific classroom type, need to be assigned accordingly. Both temporal and spatial constraints are mainly hard type constraints whose fulfillment determines the effectiveness of a schedule.

5.2.Constraint Based Approaches

Constraint satisfaction problems or CSPs are mathematical problems defined as a set of objects whose state must satisfy a number of constraints or limitations. CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which are solved by constraint satisfaction methods. CSPs are the subject of intense research in both artificial intelligence and operations research, since the regularity in their formulation provides a common basis to analyze and solve problems of many unrelated families. A solution to a timetabling problem using CSP is described as follows: (Deris et al., 1997).

The problem depicts each course as offering several subjects per semester and each subject having a specified number of lessons per week. Each lesson can thus be defined as the contact hours between lecturers and students at a specific time and place (room). Each lesson lasts for a specific period of time.

Thus the timetabling problem can be defined as an assignment of time ti 1<j<m and rooms rk , 1< k < p to lessons s(i), 1<i<n taught by lecturer L(S(i)) such that all constraints C(S(i)) are satisfied . L(S(i)) and C(S(i)) are lectures and constraints of a lesson respectively, while m, p and n are variables .

Thus the general CSP for a timetabling problem is as follows.

- A finite set of variables X1 …… Xn.
- For each variable Xi a set of domains D1 …… Dn containing possible values of Xi.
- A finite set of constraints, C1 ….. Cq representing relations between variables.

A solution to the CSP involves assigning values from domains of all variables such that all constraints are satisfied.

In terms of the CSP, a timetabling problem can be formulated by representing a timeslot and a room of a lesson as variables of the CSP, available timeslots and rooms as values of the CSP, whereas constraints are the various relationships between lessons. Therefore the CSP model for timetabling problems can be formulated by deciding the variables, values and constraints.

CSP is a decision making tool that satisfies all constraints and its major advantages are as follows:

- Constraint propagation reduces the search space so it takes less time to search thus minimizing backtracking;
- Memory requirement is smaller since the search space has been reduced;
- All available resources are represented in the form of constraints and hence user preferences and requirements can be easily satisfied.

6. ALGORITHMIC STRATEGY

There are many solution generation techniques that can be used to find a solution for the course timetabling problem. But we use Graph Coloring Algorithm approach for Timetable Generation.

6.1.Graph Coloring Algorithm

For solving Course Timetable scheduling problems using graph coloring, the problem is first formulated in the form of a graph where courses act as vertices. Depending on type of graph created, edges are drawn accordingly. One type is conflicting graph where edges are drawn between conflicting courses having common students. Other one is non-conflicting graph, where edges are drawn between mutually exclusive courses having no students in common. Sometimes it is found that creating a non-conflicting graph from the given input set and constraints is easier costing less time. This non-conflicting graph needs to be complemented to obtain the required conflict graph whose proper coloring provides the desired solution. This two-step method is efficient in certain cases, while in some conflict graph is created directly.

Some problems involve few resources while others may require many at a time. Courses can also conflict due to common teachers, common classrooms in addition to common students. In such cases, the conflict graph must consider course, teacher and room conflicts simultaneously.

As mentioned earlier, there can be various aspects of a scheduling problem. When teachers are involved in resources, other factors like availability of teachers, subject area preferred by each teacher acts as additional data inputs which needs to be provided for making a complete schedule.

Input: The course conflict graph G thus obtained act as the input of graph coloring algorithm.
Output: The minimum number n of non-conflicting time-slots required to schedule courses.

Degree sequence is the array having degree of each vertices of the input graph G. Colors being used are stored in Used_Color array. And the chromatic number will be the total number of elements in the Used_Color array.

Step 1: Input the conflict graph G.

Step 2: Compute degree sequence of the input conflict graph G.

Step 3: Assign color1 to the vertex vi of G having highest degree.

Step 4: Assign color1 to all the non-adjacent uncoloured vertices of vi and store color1 into Used_Color array.

Step 5: Assign new color which is not previously used to the next uncoloured vertex having next highest degree.

Step 6: Assign the same new color to all non-adjacent uncoloured vertices of the newly colored vertex.

Step 7: Repeat step-5 and step-6 until all vertices are colored.

Step 8: Set minimum number of non-conflicting time-slots n= chromatic number of the colored graph=total number of elements in Used_Color array.

Step 9: End.

6.2.Case Studies

Undergraduate colleges under Indian Universities offer a variety of subject combinations to its students. In streams like B.E or B.Sc., students can take one subject as Honours (Major) and two subjects as General (Minor/Pass) papers. Also, to conduct such courses teachers of respective subjects are needed to be scheduled according to their availabilities in minimum number of time slots without any conflict. In the following subsection, we have presented such typical case of scheduling problems and their conflict free solution timetables.

Example 1: Teacher-Subject problem

Problem Definition:

For a given ‘T’ number of teachers, ‘N’ number of subjects and available ‘P’ number of periods, a timetable should be prepared. The number of classes for each subject needed by a particular teacher is given in Table 2. This problem is mentioned earlier in some papers , but only a partial solution was provided in both.

Input Dataset:

Number of teachers- 4

Number of subjects- 5

Periods P	N1	N2	N3	N4	N5
T1	2	0	1	1	0
T2	0	1	0	1	0
T3	0	1	1	1	0
T4	0	0	0	1	1

Table 1: Teacher-Subject Requirement Matrix

List of constraints:

Hard Constraints-

- At any one period, each subject can be taught by maximum one teacher.
- At any one period, each teacher can teach at most one subject

Soft Constraints-

- Teacher taking two classes of same subject to be scheduled in consecutive periods.
- No more than two consecutive theory classes can be assigned to same teacher for teaching same subject.

Solution:

This problem is solved using bipartite graph which acts as the conflict graph. The set of teachers and subjects are the two disjoint independent sets. Edges are drawn connecting a vertex from Teacher set to a vertex in Subject set, indicating that the subject is taught by the respective teacher. This data is obtained from the Teacher-Subject requirement matrix given in Table 1. The solution to the problem is obtained file by proper edge coloring of the bipartite graph as shown in Fig. 6.1. The chromatic number acts as the minimum number of

periods.

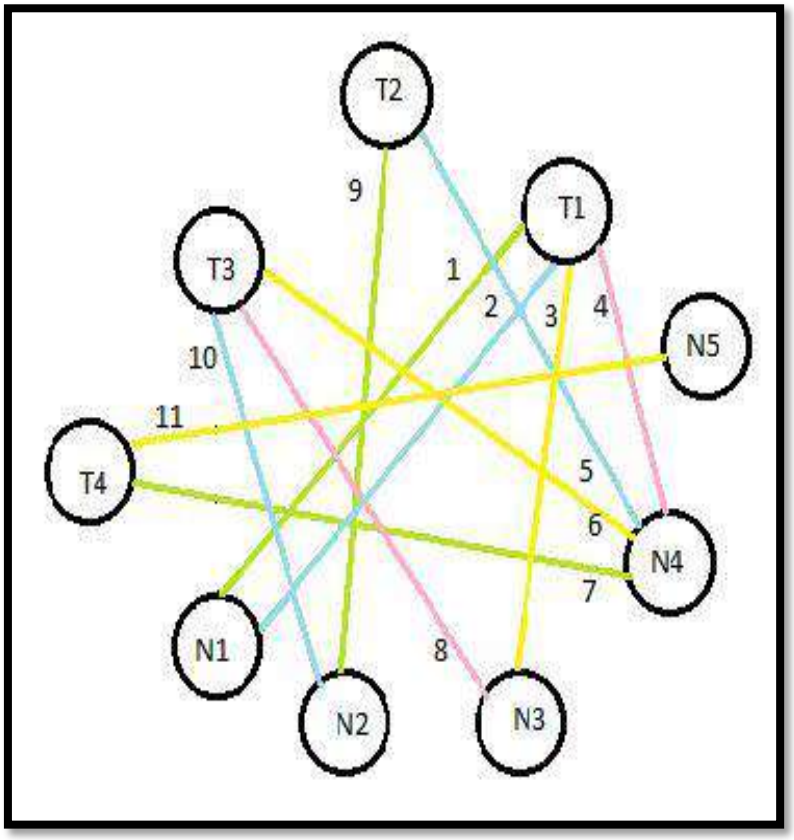


Figure 6.1: Bipartite Graph G

An alternative way of solving the above problem is by converting the edge coloring problem into a vertex coloring problem. For that the bipartite graph is converted into its equivalent line graph L(G) , and a proper vertex coloring of the line graph shown in Fig. 6.2 gives the same solution. This is simply an alternative procedure and is used depending on the type of graph coloring needed to be applied. The eleven edges present in the bipartite graph in Fig. 6.1 acts as the vertices of L(G) in Fig. 6.2.

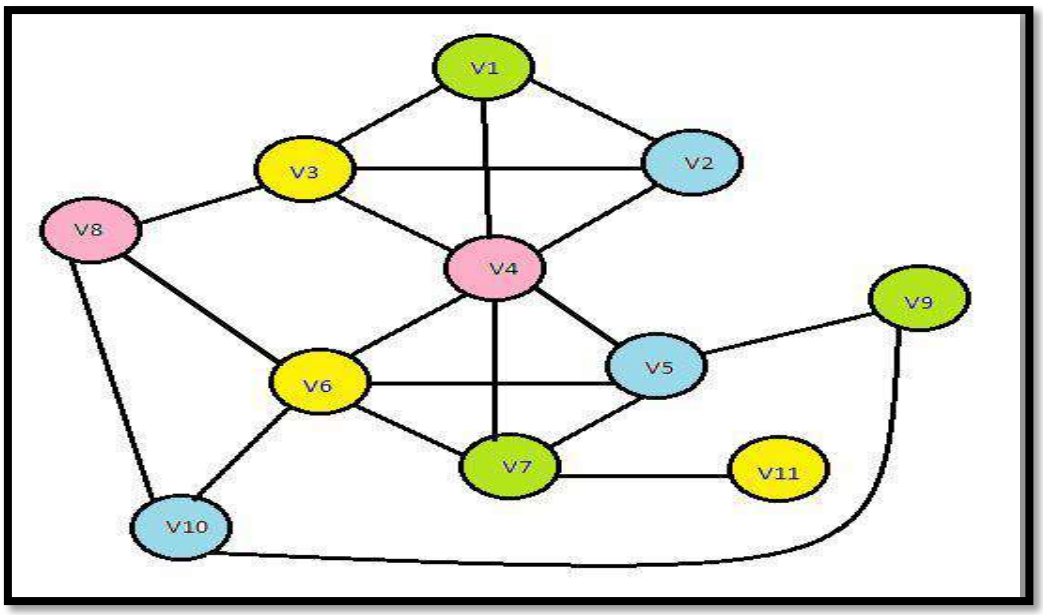


Figure 6.2: Line graph L(G)

Result:

Graph Coloring of the line graph L(G) in Fig. 6-2 has been plotted in the solution Table 2 below.

PINK	GREEN	BLUE	YELLOW
V4	V1	V2	V3
V8	V7	V5	V6
	V9	V10	V11

Table 2: Graph Coloring Solution Table of Fig. 6.2

Now, each of these colors in Table 2 represents periods in Table 3 and the vertex in L(G) (Fig. 6-2) that corresponds to a particular edge in the bipartite graph G (Fig. 6-1) represents the teacher-subject combination scheduled under that period.

Thus, the final complete schedule is obtained and shown in Table 3.

Period 1	Period 2	Period 3	Period 4
T1-N4	T1-N1	T1-N1	T1-N3
T3-N3	T4-N4	T2-N4	T3-N4
	T2-N2	T3-N2	T4-N5

Table 3: Final Teacher-Subject Allotment Table

Proof of satisfaction:

It can be easily established that the specified constraints are satisfied by the above schedule.

- No common subject in any column indicates that at any particular period, a subject is taught by only one teacher.
- No duplicate data in any cell indicates that at any particular period, a teacher can teach only one subject.

7. SYSTEM DESIGN

7.1.Timetable Scheduling Algorithm

The flow of timetable generation using the Automated Timetable generator which contains the contribution of Graph coloring algorithm.

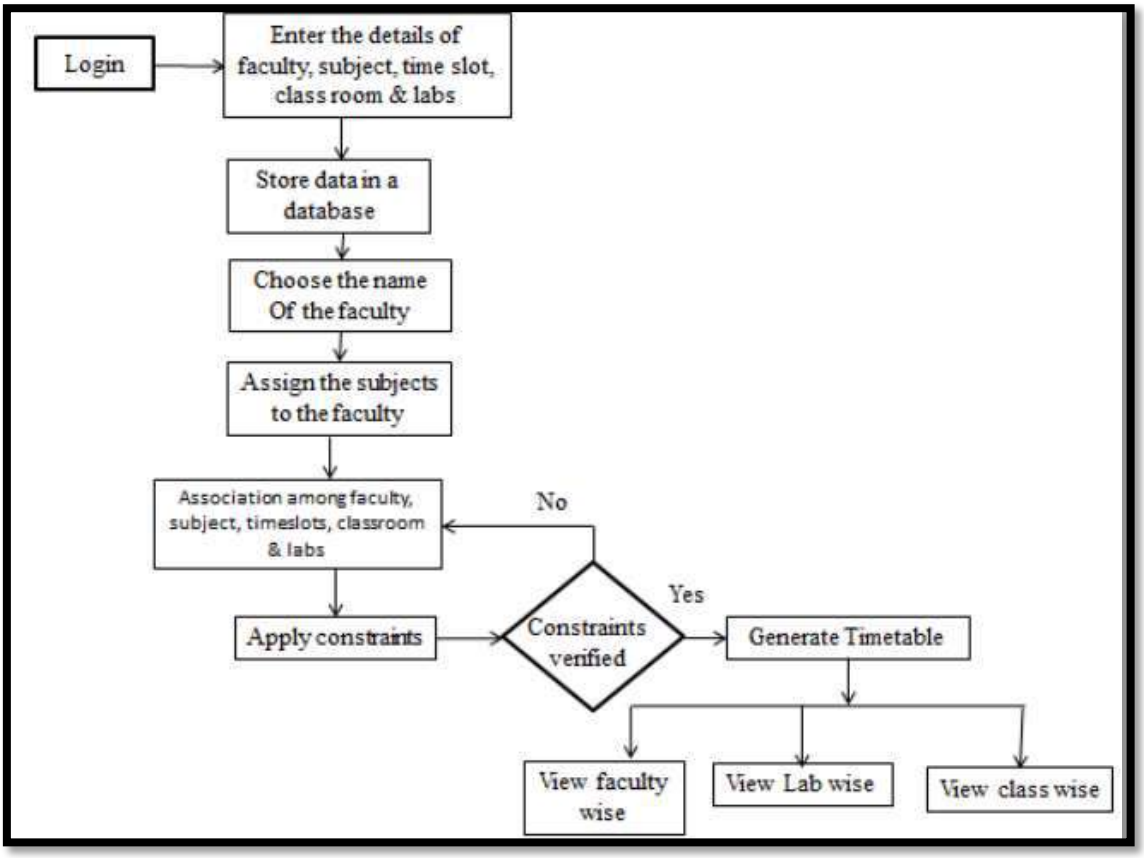


Figure 7.1: System Design of Timetable

The above figure shows a flow of following procedures:

There is one admin in the software who will handle the software. The admin have to login into the system by entering their id and respected password. The admin have to enter each of the records as number of faculties available, counts of subjects, class-rooms, labs, lectures, students and timeslots. The data will be store in the database for further process. The admin will choose the name of the faculty and allot each subject to them and also assign classrooms and the students whom they will teach. The software then starts association among the faculty with their subjects and classrooms.

The constraints will be applied as specified in the algorithm so that no conflicts occur. Same procedure will be repeated until satisfactory results will be generated. After successful verification the Timetable will be ready for the staffs and students to view with respected to class, faculty and labs.

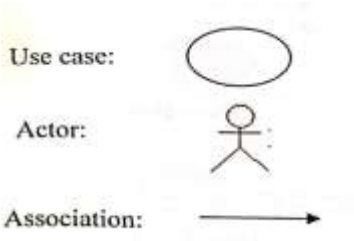
7.2.Use Case Diagram

A use case diagram is a type of behavioral diagram defined by the Unified Modeling Language (UML). Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals- represented as use cases- and any dependencies between those use cases.

Use case diagram component:

- **Use cases:** A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- **Actors:** An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- **Associations:** Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modelled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.
- **System boundary boxes (optional):** You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not. System boundary boxes are rarely used, although on occasion I have used them to identify which use cases will be delivered in each major release of a system.

Following are the representation of use case diagram component:



UML use case diagram for Automatic Timetable Generation is shown below. The various participants of the same are detailed below:

Actors :- Admin, Faculty
Use cases :- Add and Delete Teachers, Add and Delete Subjects, Add and Delete Classrooms, Allot subjects, Allot Classrooms ,View Timetable.

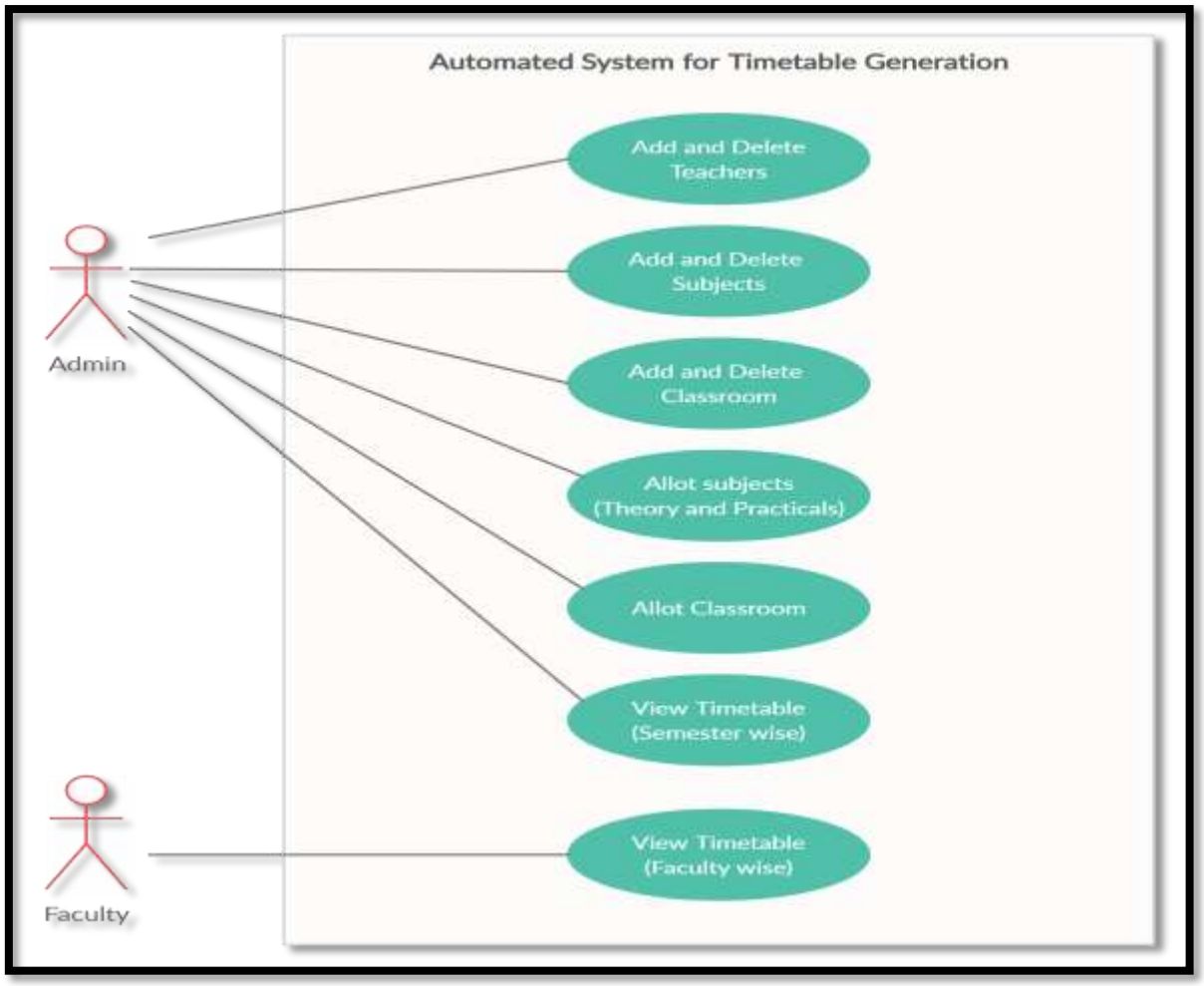


Figure 7.2: Use Case Diagram for Timetable Generation

7.3.Flow of User

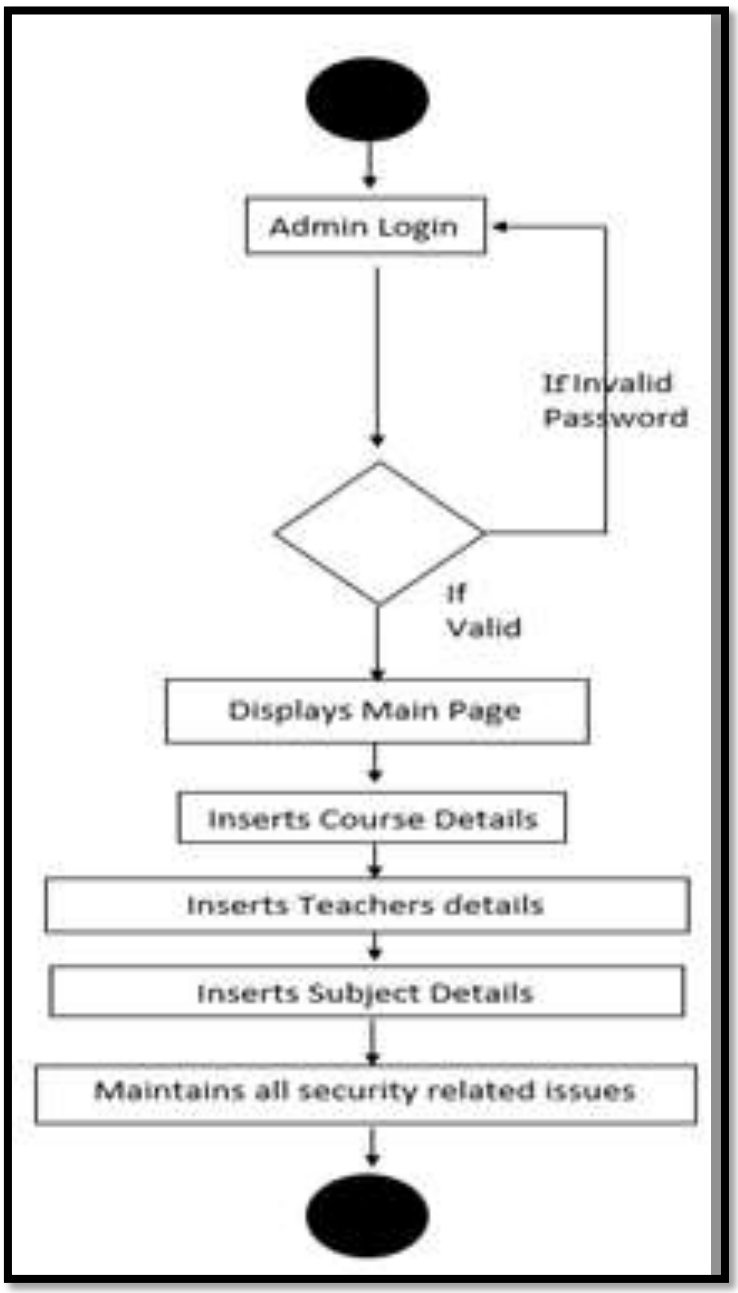


Figure 7.3: Data Flow Diagram of User

7.4.Class Diagram for Timetable Generation

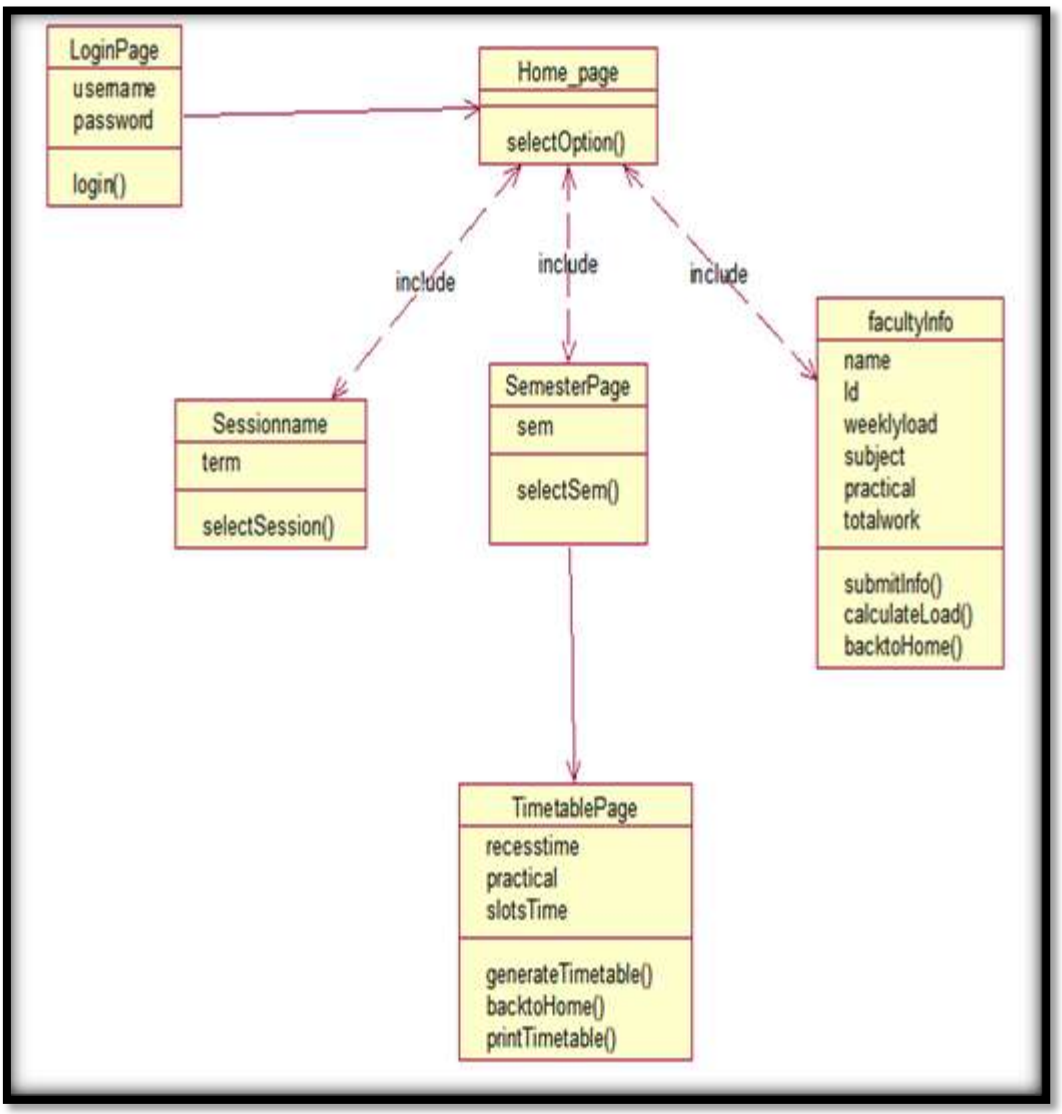


Figure 7.4: Class Diagram for Timetable Generation

I. Login Page

Contains attributes such as username and password and operation for allowing authenticate user to get login ().

II. Home Page

This class contain the selectOption() operation for selecting select option.

III. Semester Page

This class contain attributes sem and operation selectsem() .

IV. Faculty

This class contain attributes name, id weekly load, subject, practical, total work and

operation submit info(), calculate workload() backtohome().

V. Timetable Page

This class contain attributes recess time, practical,slotsTime and operation GenreteTimeTable (), backtohomepage(), printTimetable() . This a main class of our project.

7.5.Flowchart of Timetable Generation

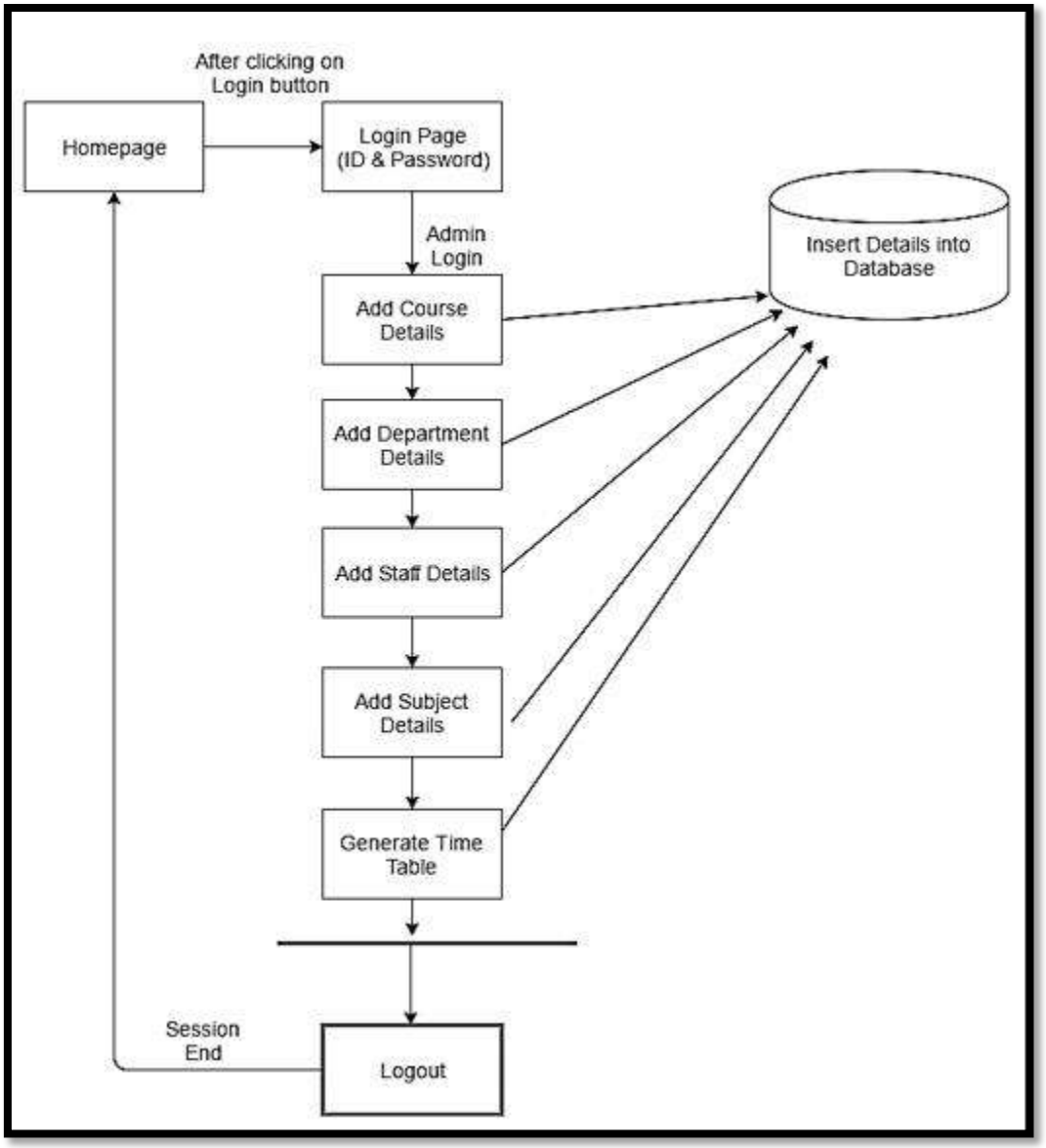


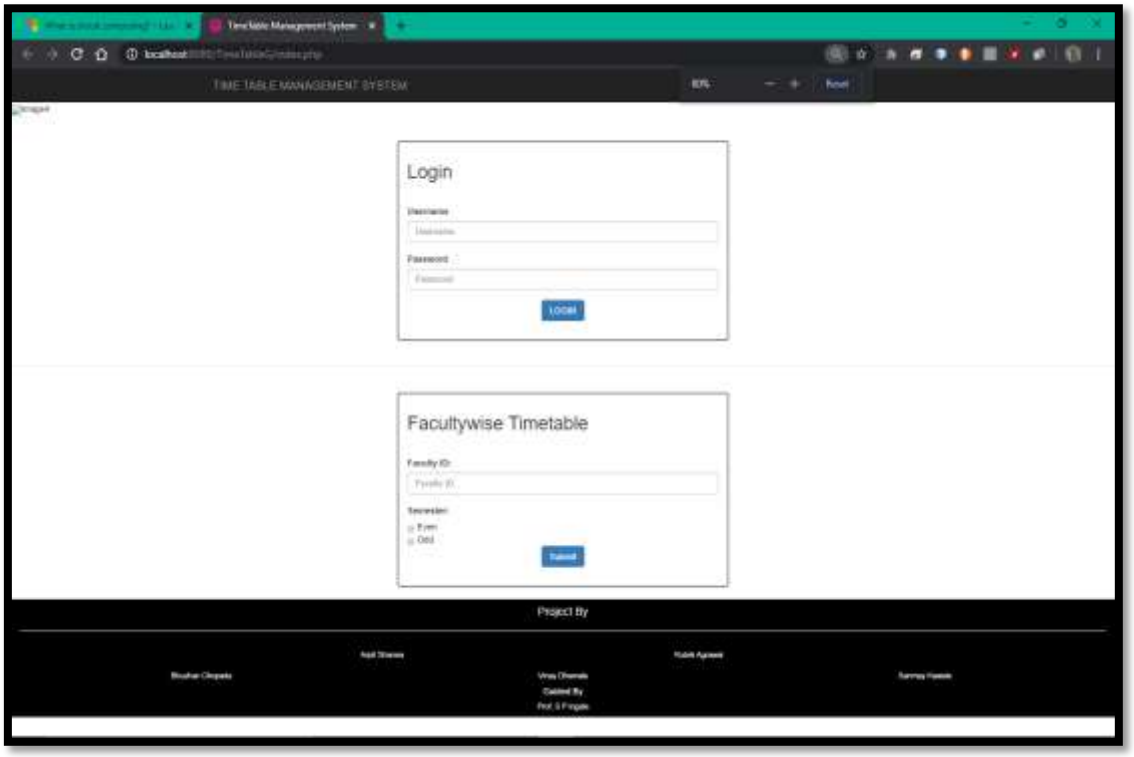
Figure 7.5: Flowchart of Automatic Timetable Generation

8. IMPLEMENTATION & RESULT

Software testing is an investigation conducted to provide stake owners with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs (errors or other defects). Software testing can be stated as the process of validating and verifying that a computer program/application/product:

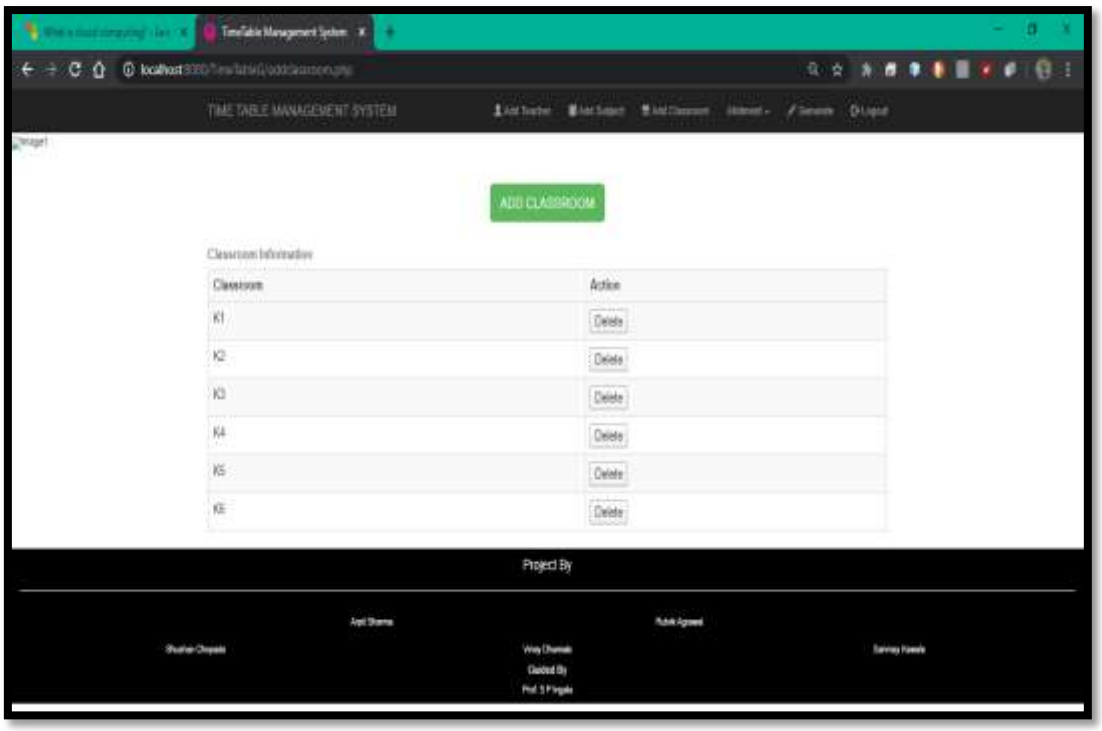
- Meets the requirements that guided its design and development
- Works as expected

8.1.Results and Screenshots



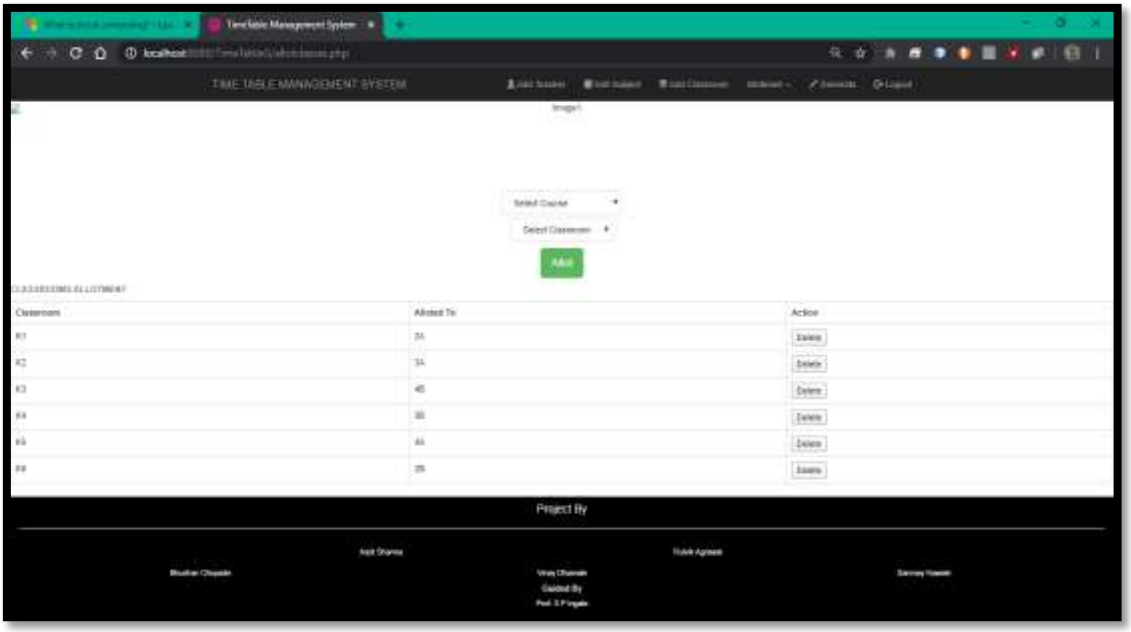
Screenshot 8.1: Login Screen for Admin

Above screenshot gives the frame of Admin layout. Where admin and Faculty can login using unique username and password. This interface is specified to admin and faculty only.

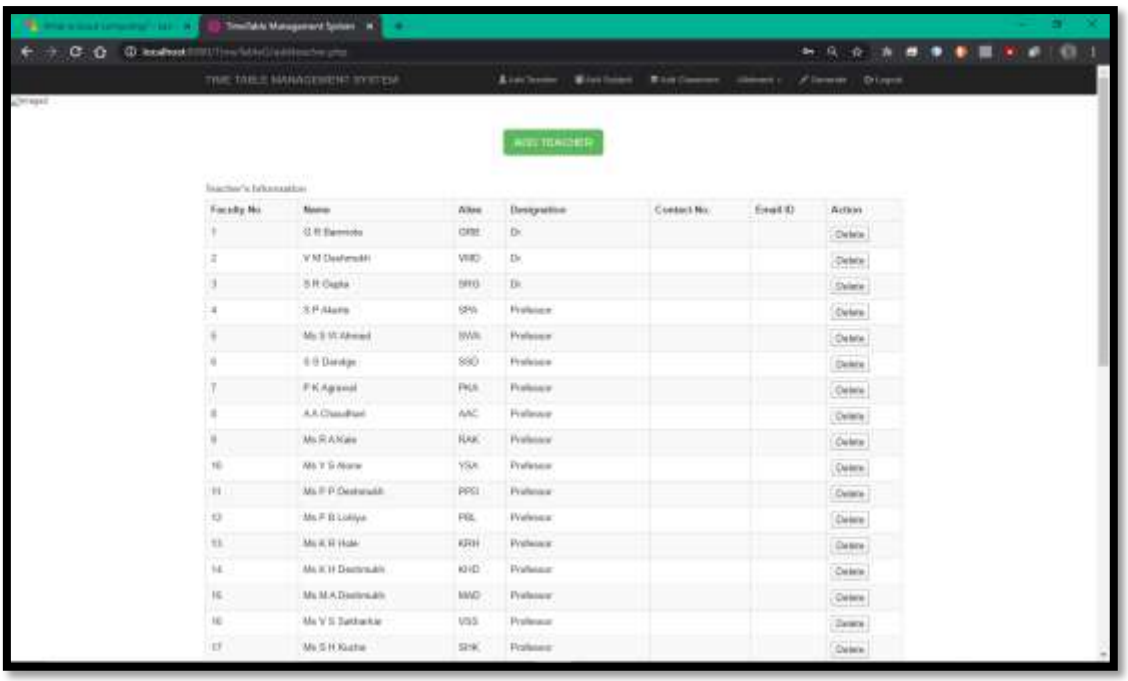


Screenshot 8.2: Add Classroom Screen

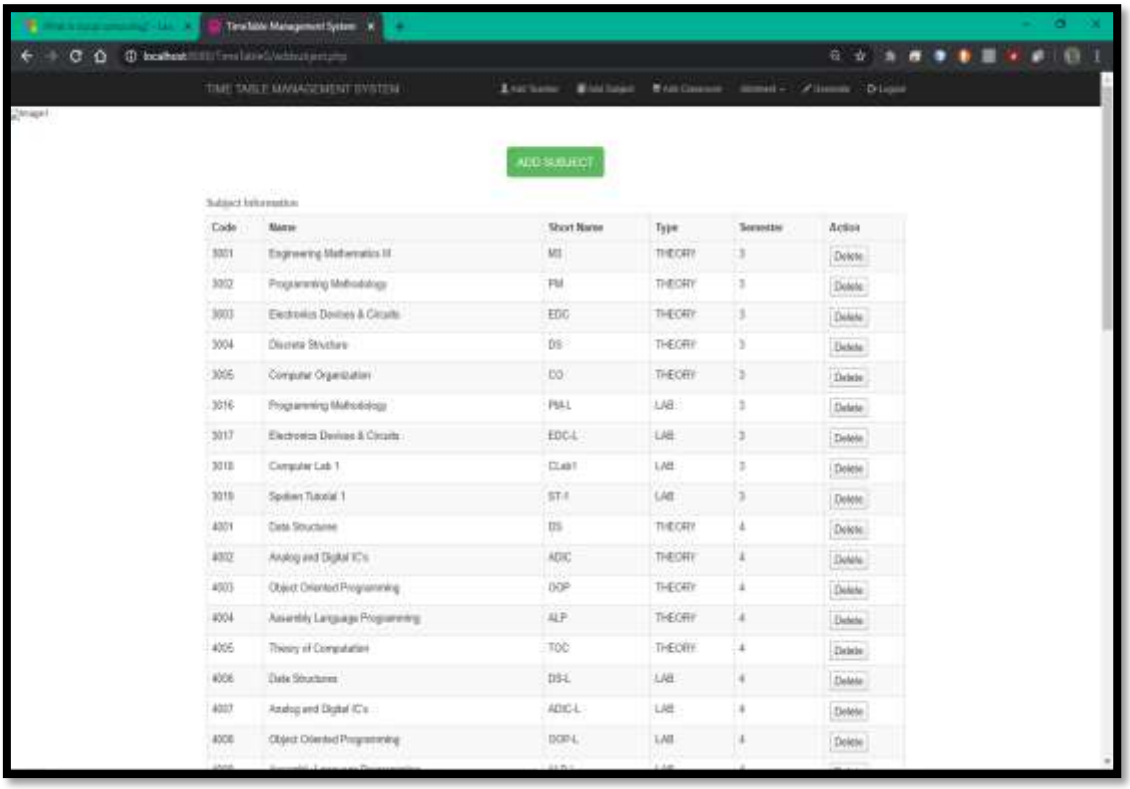
After login process completed 'Add Classroom' frame sequenced where admin can add classroom.



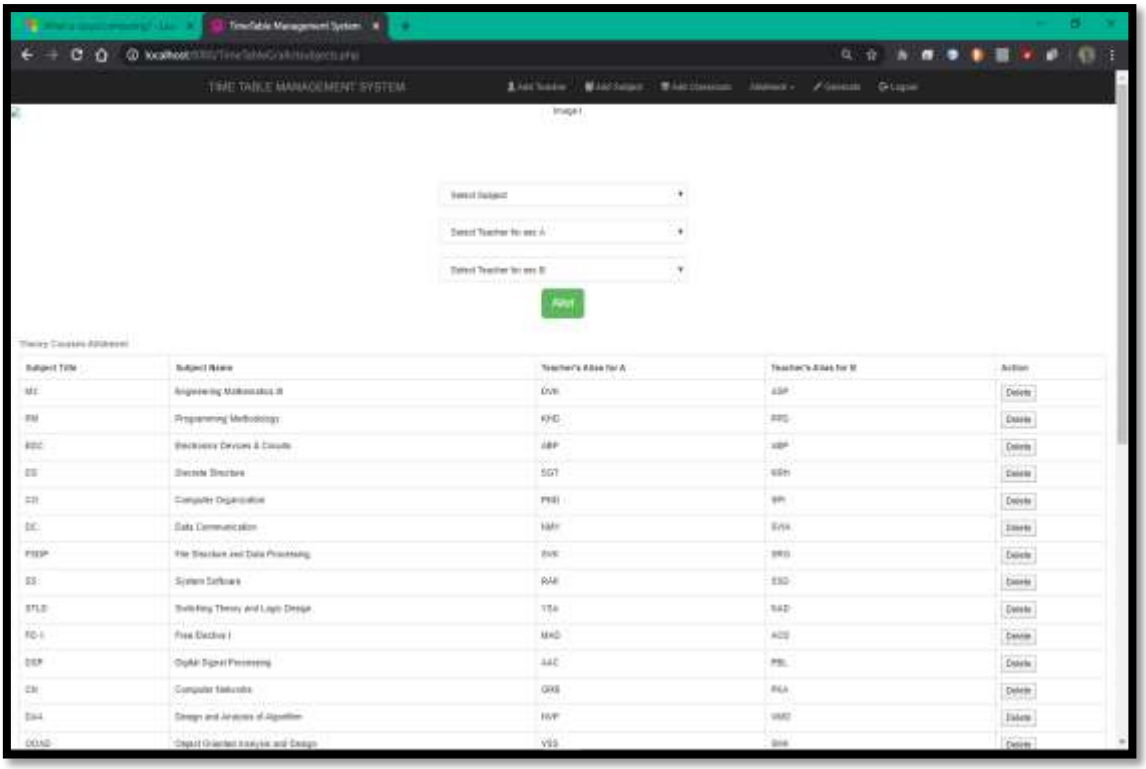
Screenshot 8.3: Classroom Allotment Screen



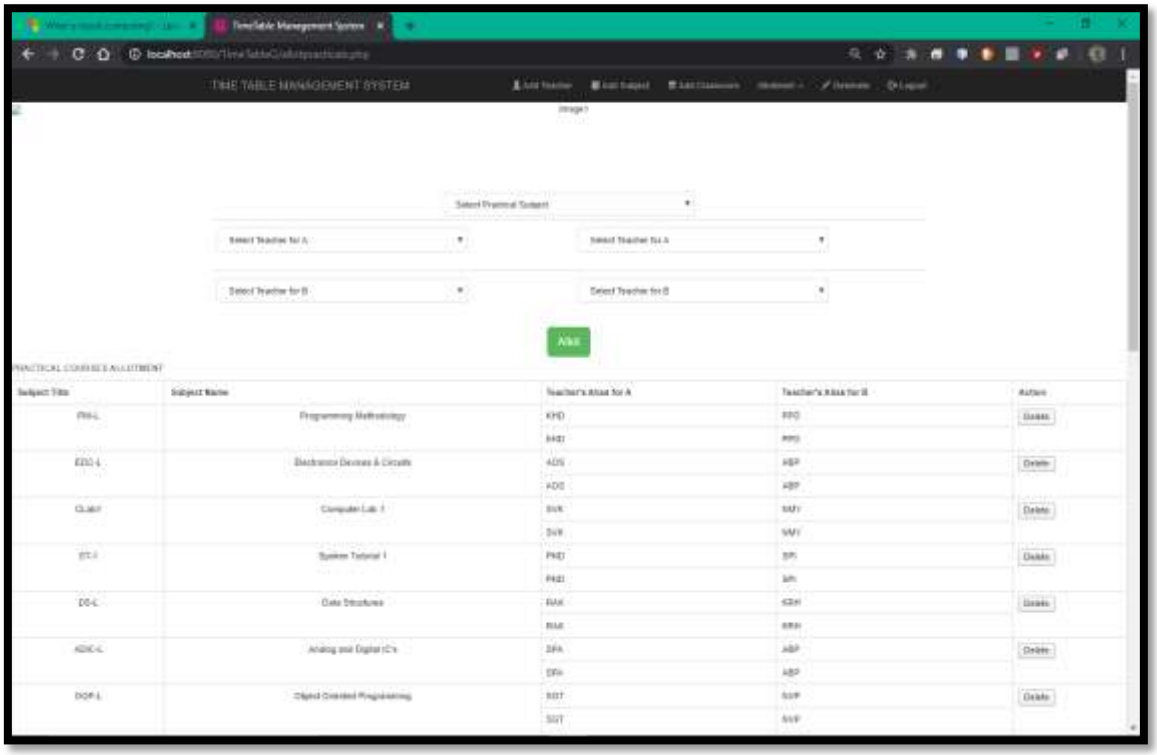
Screenshot 8.4: Add Teacher Screen



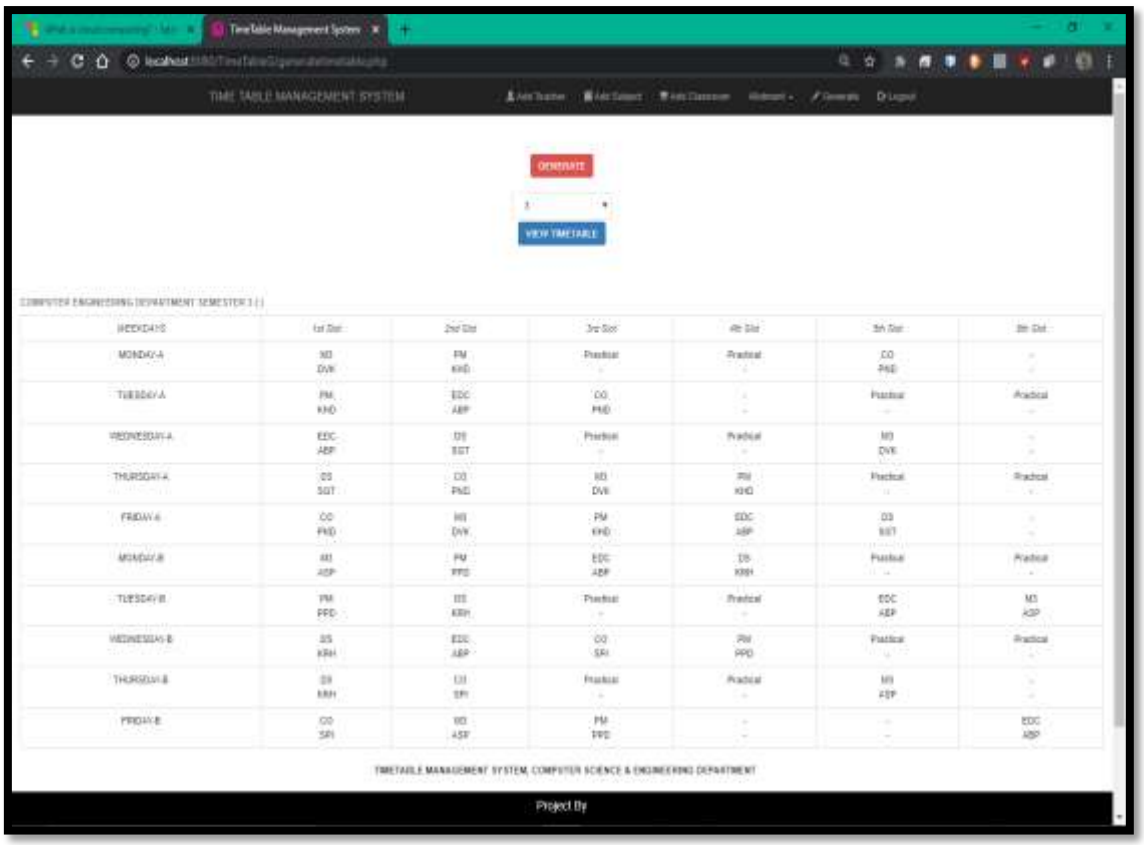
Screenshot 8.5: Add Subject Screen



Screenshot 8.6: Teacher Allotment Screen for 3rd Year



Screenshot 8.7: Teacher Allotment Screen for 3rd Semester



Screenshot 8.8: Output Timetable Screen for 3rd Semester

9. APPLICATION, ADVANTAGES & DISADVANTAGES

9.1.Applications

- Allocation of periods to classes and instructors is completely mechanized and does not restrict the user any processing.
- A compatible and accurate timetable is guaranteed and the system is therefore well organized and reliable.
- Enables the collaboration of teachers and students.
- The study material can be used in common with others.
- Time and Labour is saved.
- This system can be used by Schools and Colleges to create Time-Table.

9.2.Advantages

- Reduce the paper work and storage area.
- Improve accuracy in result.
- It is user friendly having quick authentication.
- Secure log in logout and validation.
- It is easy to maintain.
- Work efficiency is improved.
- Provides up to date information.
- Friendly Environment by providing warning messages.

9.3.Disadvantages

- The project lacks the implementation of load balancing technique.
- The project fails to handle larger set of database.
- Lot of modification has to be done if the examination pattern is change.

10.CONCLUSION AND FUTURE WORKS

10.1. Conclusion

The intention of the algorithm to generate a time-table schedule automatically is satisfied. It also, addresses the important hard constraint of clashes between the availability of teachers. The non-rigid soft constraints i.e. optimization objectives for the search operation are also effectively handled. Given the generality of the algorithm operation, it can further be adapted to more specific scenarios, e.g. University, examination scheduling and further be enhanced to create railway time tables. Thus, through the process of automation of the time-table problem, many an-hours of creating an effective timetable have been reduced eventually. The most interesting future direction in the development of the algorithm lies in its extension to constraint propagation. When there is a value assigned to a variable, such assignment can be propagated to unassigned variables to prohibit all values which come into conflict with the current assignments. The information about such prohibited values can be propagated as well.

The Taken the generality of the rule operation, it will additional be tailored to a lot of specific eventualities, e.g. School, examination planning and additional be increased to make railway time tables. Thus, through the method of automation of the time-table drawback, several an-hours of making a good timetable are decreased eventually. The foremost fascinating future directions within the development of the rule arise in its extension to constraint propagation. Once there's a price appointed to a variable, such assignment is propagated to unassigned variables to allow all values that get struggle with the present assignments the data concerning such prohibited worth’s is propagated furthermore.

10.2. Future Scope

This software is a solution for the time table generation problem manually. It’s main scope is to save the time and efforts for time table generation process.

- The data of faculty in the data base can further be used to maintain record of faculty’s experience for particular subjects.
- Attribute Correctness of project will give more corrective approach toward generation of this timetable. This project will generate most corrective output with no errors.

- The future enhancement that can be developed from the project is to generate the master timetable for the departments and to the entire college. This enhancement can be achieved my making further modifications keeping the approach and techniques used in this project.

REFERENCES

[1] K. “A Study on Course Timetable Scheduling using Graph Coloring Approach” International Journal of Computational and Applied Mathematics. ISSN 1819- 4966 Volume 12, Number 2 (2017), pp. 469-485.

[2] “Web Application for Automatic Time Table Generation” International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 –,5161.

[3] “Automated timetable generator for educational institutions using graph colouring technology” International Journal of Applied Research 2018; 4(4).

[4] “A Literature Review on Timetable generation algorithms based on Genetic Algorithm and Heuristic approach” International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 4, April 2015.

[5] “Automatic Timetable Generation using Genetic Algorithm”, IJARCCE Vol. 4, Issue 2, February 2015.

[6] Burke, E. K., Elliman, D. G., and Weare, R., 1993, “A university timetabling. system based on graph coloring and constraint manipulation,” Journal of Research on Computing in Education, 27(1).

[7] Akbulut, A., and Yılmaz, G., 2013, “University Exam Scheduling System Using Graph Coloring Algorithm and RFID Technology,” International Journal of Innovation, Management and Technology, 4(1).

[8] Shirinivas, S. G., Vetrivel, S., and Elango, N. M., 2010, “Application of Graph. Theory in Computer Science An Overview,” International Journal of Engineering Science and Technology, 2(9).

[9] S. C.& Kwok ,Kong “TIME TABLE GENERATION SYSTEM” .Vol.3 Issue.2, February- 2014,pg.410-414.

[10] Anirudha Nanda “An Application to Automatically Generate Schedule for School Lectures Using a Heuristic Approach”.International Journal of Machine Learning and Computig Vol. 2. No 4, August 2012.

[11] DiptiShrinivasn “automated time table generation using multiple context resoning For university modules”.Published in: evolutionary computation, 2002.cec 02.

[12] C. Blum and A. Roli, “Metaheuristics in combinatiorial optimization: Overview and conceptual comparison”. ACM Comput. Surv.,vvol. 35, no. 3, pp. 268-308, 2003.

[13] Carter , M. W., Laporte ,G., and Lee, S. Y., 1996,"Exminationn timetableling: Algorithmic strategies and applications." Journal of the Operational Research Society47(3),, pp, 373-383.

[14] Deeksha C S, A Kavya Reddy, Nagambika A, Akash Castelio and K Panimozhi,"Automated Timetable Generator" (2015).

[15] Amol C. Adamuthe and Rajankumar S Bichkar ," Tabu Search for Solving Personal Scheduling Problem".

[16] Cagdas Hakan Aladag and Gulsum Hocaoglu ," A Tabu search Algorithm To Solve a Course Timetabling Problem", Hacettepe Journal of Mathematics and Statistics, Vol.36(1)

RESUME

PERSONAL DETAILS	
NAME	Arpit Sharma
DATE OF BIRTH	20-Oct-1998
ADDRESS	Ishwardeep Apt., Gajanan Colony, Kathora Road, Amravati
MOBILE NO	8806524040
EMAIL_ID	arpeet.sharma.1998@gmail.com

EDUCATION DETAILS

Name of Board		Passing Year	% of Marks/ CGPA
10th SSC	Maharashtra State Board	2014	81.20
12th HSC	Maharashtra State Board	2016	77.08
Diploma	Maharashtra State Board Of Technical Education	-	-

Bachelor Of Engineering (B.E)		Passing Year & Month	Marks Obt/ Out of	% of Marks	Pointer
I st Year	I - SEM	**Dec-2016**	**419/600**	**69.83**	**7.84**
	II - SEM	**Jun-2017**	**445/600**	**74.16**	**8.79**
II nd Year	III - SEM	**Dec-2017**	**422/650**	**64.92**	**7.35**
	IV - SEM	**Jun-2018**	**495/700**	**70.71**	**8**
III rd Year	V - SEM	**Dec-2018**	**428/700**	**61.14**	**6.54**
	VI - SEM	**Jun-2019**	**455/700**	**65**	**7.31**
IV th Year	VII - SEM	**Dec-2019**	**456/700**	**65.14**	**7.33**
	VIII - SEM				

PLACEMENT DETAILS	
Campus Placement (If Any)	Yes
(If Any) Name Of Company	**Tata Consultancy Services(TCS)**

FUTUTRE PLNNING		
Higher Studies/ Job Preferences	Higher Studies	-
	Job	-
	Training	-
	Business	-

Place: Amravati
Date: 06/04/2020

Signature
Arpit Sharma

RESUME

PERSONAL DETAILS	
NAME	Bhushan Chopade
DATE OF BIRTH	12-Dec-1998
ADDRESS	Shri Krupa House,Sawavlambi Nagar,Kathora Road, Amravati
MOBILE NO	7378499954
EMAIL_ID	b.chopade12@gmail.com

EDUCATION DETAILS

Name of Board		Passing Year	% of Marks/ CGPA
10th SSC	Maharashtra State Board	2014	89
12th HSC	Maharashtra State Board	2016	73.85
Diploma	Maharashtra State Board Of Technical Education	-	-

Bachelor Of Engineering (B.E)		Passing Year & Month	Marks Obt/ Out of	% of Marks	Pointer
I st Year	I - SEM	**Dec-2016**	**386/600**	**64.33**	**7.04**
	II - SEM	**Jun-2017**	**449/600**	**74.83**	**8.8**
II nd Year	III - SEM	**Dec-2017**	**398/650**	**61.23**	**6.73**
	IV - SEM	**JUN-2018**	**459/700**	**65.57**	**6.88**
III rd Year	V - SEM	**Dec-2018**	**424/700**	**60.57**	**6.77**
	VI - SEM	**Jun-2019**	**477/700**	**68.14**	**8.08**
IV th Year	VII - SEM	**Dec-2019**	**479/700**	**68.42**	**8.07**
	VIII - SEM				

PLACEMENT DETAILS	
Campus Placement (If Any)	No
(If Any) Name Of Company	-

FUTUTRE PLNNING		
Higher Studies/ Job Preferences	Higher Studies	
	Job	
	Training	
	Business	

Place: Amravati
Date: 06/04/2020

Signature
Bhushan S.Chopade

RESUME

PERSONAL DETAILS	
NAME	Rutvik S. Agrawal
DATE OF BIRTH	10-JAN-1999
ADDRESS	Ranet, Devrankar Nagar, Badnera Road, Amravati
MOBILE NO	9403621961
EMAIL_ID	rutviksagrawal@gmail.com

EDUCATION DETAILS

Name of Board		Passing Year	% of Marks/ CGPA
10th SSC	ICSE	2014	82.50
12th HSC	Maharashtra State Board	2016	78.31
Diploma	Maharashtra State Board Of Technical Education	-	-

Bachelor Of Engineering (B.E)		Passing Year & Month	Marks Obt/ Out of	% of Marks	Pointer
I st Year	I - SEM	**Dec-2016**	**404/600**	**67.33**	**7.64**
	II - SEM	**Jun-2017**	**404/600**	**67.33**	**8.04**
II nd Year	III - SEM	**Dec-2017**	**452/650**	**69.54**	**8.23**
	IV - SEM	**Jun-2018**	**471/700**	**67.29**	**7.2**
III rd Year	V - SEM	**Dec-2018**	**436/700**	**62.29**	**6.88**
	VI - SEM	**Jun-2019**	**398/700**	**56.86**	**6.12**
IV th Year	VII - SEM	**Dec-2019**	**416/700**	**59.43**	**6.41**
	VIII - SEM				

PLACEMENT DETAILS	
Campus Placement (If Any)	NO
(If Any) Name Of Company	-

FUTUTRE PLNNING		
Higher Studies/ Job Preferences	Higher Studies	-
	Job	-
	Training	-
	Business	Yes

Place: Amravati
Date: 06/04/2020

Signature
Rutvik Agrawal

RESUME

PERSONAL DETAILS	
NAME	Sanmay Kawale
DATE OF BIRTH	06-Oct-1995
ADDRESS	C/o Narendra Ingole, Jyotiba Colony, Near Chaitanya Colony, Amravati.
MOBILE NO	7709570345
EMAIL_ID	sanmaykawale@gmail.com

EDUCATION DETAILS

Name of Board		Passing Year	% of Marks/ CGPA
10th SSC	Maharashtra State Board	2011	87.09
12th HSC	Maharashtra State Board	2013	61.67
Diploma	Maharashtra State Board Of Technical Education	-	-

Bachelor Of Engineering (B.E)		Passing Year & Month	Marks Obt/ Out of	% of Marks	Pointer
I st Year	I - SEM	**Jul-2015**	**334/600**	**55.66**	**5.83**
	II - SEM	**Nov-2014**	**350/600**	**58.33**	**5.88**
II nd Year	III - SEM	**Feb-2015**	**339/650**	**52.15**	**5.23**
	IV - SEM	**Sep-2016**	**372/700**	**53.14**	**4.84**
III rd Year	V - SEM	**Jan-2020**	**262/700**	**37.42**	**2.57**
	VI - SEM	**Sep-2019**	**376/700**	**53.71**	**5.31**
IV th Year	VII - SEM	**Jan-2019**	**402/700**	**57.42**	**5.96**
	VIII - SEM				

PLACEMENT DETAILS	
Campus Placement (If Any)	No
(If Any) Name Of Company	-

FUTUTRE PLNNING		
Higher Studies/ Job Preferences	Higher Studies	Yes
	Job	-
	Training	-
	Business	-

Place: Amravati
Date: 06/04/2020

Signature
Sanmay Kawale

RESUME

PERSONAL DETAILS	
NAME	Vinay Govind Dhamale
DATE OF BIRTH	24-Feb-1999
ADDRESS	Fatema Apartment, Sundarlal Square, Camp, Amravati
MOBILE NO	8806503870
EMAIL_ID	vinaydhamale9@gmail.com

EDUCATION DETAILS

Name of Board		Passing Year	% of Marks/ CGPA
10th SSC	Maharashtra State Board	2014	86.40
12th HSC	Maharashtra State Board	2016	65.24
Diploma	-	-	-

Bachelor Of Engineering (B.E)		Passing Year & Month	Marks Obt/ Out of	% of Marks	Pointer
I st Year	I - SEM	**Dec-2016**	**400/600**	**66.67**	**7.6**
	II - SEM	**Jun-2017**	**380/600**	**63.33**	**6.83**
II nd Year	III - SEM	**Dec-2017**	**399/650**	**61.38**	**6.92**
	IV - SEM	**Jun-2018**	**437/700**	**62.43**	**6.56**
III rd Year	V - SEM	**Dec-2018**	**438/700**	**62.57**	**7.04**
	VI - SEM	**Jun-2019**	**441/700**	**63.00**	**7.23**
IV th Year	VII - SEM	**Dec-2019**	**439/700**	**62.71**	**6.89**
	VIII - SEM				

PLACEMENT DETAILS	
Campus Placement (If Any)	No
(If Any) Name Of Company	-

FUTUTRE PLNNING		
Higher Studies/ Job Preferences	Higher Studies	-
	Job	-
	Training	-
	Business	-

Place: Amravati
Date: 06/04/2020

Signature
Vinay Dhamale