

A Project Report on

# **AUTOMATIC TIMETABLE GENERATOR USING GENETIC ALGORITHM**

Submitted in partial fulfillment of the requirements for the award  
of the degree of

**Bachelor of Engineering**

in

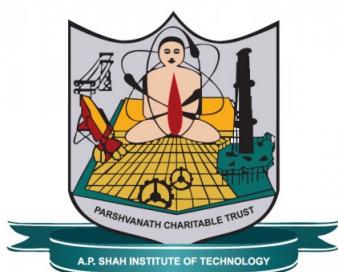
**Information Technology**

by

**Harsh Bhanushali(16104043)  
Greshma Sapra (16104038)  
Surbhi Saroliya (16104037)**

Under the Guidance of

**Ms. Rujata Chaudhari  
Ms. Nahid Shaikh**



**Department of Information Technology**

A.P. Shah Institute of Technology  
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615  
UNIVERSITY OF MUMBAI

**Academic Year 2019-2020**

## Approval Sheet

This Project Report entitled "***Automatic Timetable Generator using Genetic Algorithm***" Submitted by "***Harsh Bhanushali***" (16104043), "***Greshma Sapra***" (16104038), "***Surbhi Saroliya***" (16104037) is approved for the partial fulfillment of the requirement for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Ms. Nahid Shaikh  
Co-Guide

Ms. Rujata Chaudhari  
Guide

Mr. Kiran Deshpande  
Head Department of Information Technology

Place: A.P.Shah Institute of Technology, Thane  
Date:

## CERTIFICATE

This is to certify that the project entitled "***Automatic Timetable Generator using Genetic Algorithm***" submitted by "***Harsh Bhanushali***" (***16104043***), "***Greshma Sapra***" (***16104038***), "***Surbhi Saroliya***" (***16104037***) for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***, to the University of Mumbai, is a bonafide work carried out during academic year 2019-2020.

Ms. Nahid Shaikh  
Co-Guide

Ms. Rujata Chaudhari  
Guide

Mr. Kiran Deshpande  
Head Department of Information Technology

Dr. Uttam D.Kolekar  
Principal

External Examiner(s)

1.

2.

Place: A.P.Shah Institute of Technology, Thane

Date:

## **Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---

(Harsh Bhanushali 16104043)

---

(Greshma Sapra 16104038)

---

(Surbhi Saroliya 16104037)

Date:

## **Abstract**

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, 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. The system will take various inputs like number of subjects, teachers, workload of a teacher, semester, priority of subject. By relying on these inputs, it will generate possible time tables for working days of the week for teaching faculty. This will integrate by making optimal use of all resources in a way that will best suit the constraints. Most colleges have a 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 a way that their timings do not overlap and the time table schedule makes best use of all faculty subject demands. The traditional hand operated method of time table is very time consuming and usually ends up with various classes clashing either at same room or with same teachers having more than one class at a time which is being resolved by Automatic Timetable Generator System.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objectives . . . . .	3
1.2	Problem Definition . . . . .	3
1.3	Scope . . . . .	4
1.4	Technology Stack . . . . .	4
<b>2</b>	<b>Literature Review</b>	<b>5</b>
<b>3</b>	<b>Proposed System</b>	<b>9</b>
<b>4</b>	<b>Project Design</b>	<b>12</b>
	Basic Diagrams . . . . .	12
4.1	System Architecture . . . . .	12
4.2	Usecase Diagram . . . . .	13
4.3	Class Diagram . . . . .	14
4.4	Sequence Diagram . . . . .	15
4.5	Activity Diagram . . . . .	16
<b>5</b>	<b>Project Implementation</b>	<b>19</b>
5.1	Code Snippets . . . . .	19
5.2	Result . . . . .	22
<b>6</b>	<b>Testing</b>	<b>27</b>
<b>7</b>	<b>Conclusions and Future Scope</b>	<b>30</b>
7.1	Conclusion . . . . .	30
7.2	Future Scope . . . . .	30
	<b>Bibliography</b>	<b>31</b>
	<b>Appendices</b>	<b>32</b>
	Appendix-A . . . . .	32
	Appendix-B . . . . .	33
	Appendix-C . . . . .	33

# List of Figures

1.1	Flowchart of Genetic Algorithm . . . . .	2
1.2	Technology Stack . . . . .	4
2.1	Project Timeline Chart . . . . .	8
3.1	System Flow . . . . .	9
4.1	System Architecture . . . . .	12
4.2	Usecase Diagram . . . . .	13
4.3	Class Diagram . . . . .	14
4.4	Sequence Diagram . . . . .	15
4.5	Activity Diagram . . . . .	16
4.6	Activity Diagram for Generation of timetable . . . . .	17

# List of Tables

6.1 Testcases . . . . .	29
-------------------------	----

# **Chapter 1**

## **Introduction**

Even though most college administrative work has been computerized, timetable scheduling is still mostly done manually due to its inherent difficulties. The manual timetable scheduling demands considerable time and efforts. 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 practical's in which all given constraints are satisfied. Creating such timetables manually is complex and time consuming process. 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. Using this system Authority can concentrate on other things rather than wasting their time on preparing Timetable.

To generate such system our college sponsored our project which is built to efficiently generate college time mechanically, reducing the need of manual interference by timetable coordinator. Our system is designed in such a way that there are no slot collisions that provides optimality to the timetable, extra options that's enclosed within the system is that there's no headache of giving abundant input, only basic input like subject, semester, faculty, load, classroom number, lab number, etc, are required in csv format.

Inputs once received by system will be processed further and the given set of constraints provided in proposed system will be applied on it, further it will be validated and the system will check whether all the requirements are fulfilled or not. Once validated, system will generate a review of the timetable which if approved will be stored in the database and will be available with different views of the timetable like class timetable, lab timetable, faculty timetable and master timetable which can be exported as csv file anytime.

Our system can be used by all the educational institutes as we have built it in a generic method, only concern is the input that the system takes. For this, we have provided option of downloading a sample input file which is editable and the same can be uploaded as input after making changes, to the system. Input file cannot be altered as there are set on parameters on which the algorithm runs and these parameters if manipulated will cause system failure. System works on powerful and reliable algorithm Genetic Algorithm.

Genetic algorithm is as followed :

```
BEGIN /*genetic algorithm*/
Initial population generation;
Fitness check of individuals;
WHILE NOT finished DO LOOP
BEGIN
Mating done by selecting previous generation individuals;
Generation of offspring by crossing over/mutation of the
previous generation individuals;
Fitness check of new individuals;
Removing previous generation and using new generation;
IF Population has converged
THEN finishes:= TRUE;
END
END
```

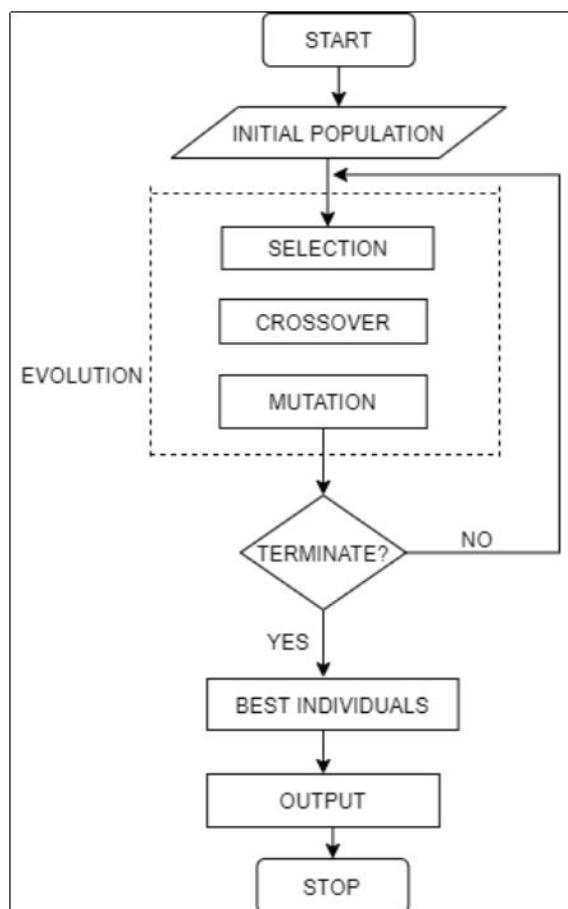


Figure 1.1: Flowchart of Genetic Algorithm

## 1.1 Objectives

The main objectives are:

1. To reduce the load on Time table coordinator.

Time table generation is a tedious and time consuming task also it takes lots of effort from the coordinator end to check the clashes of faculties while assigning slots to them,also the coordinator has to satisfy the load constraint.

2. To minimize the manual intervention in creation of the timetable.

Time table generation at present is completely based on human efforts but the system will reduce the human effort to minimum level,the effort from human end will be to provide the system with loadsheet ,review the data and forward it to the algorithm.

3. Generates multiple useful views from timetable.

Time table once generated has multiple data in it,the system will provide different views of the time table to the user such as faculty view,student view and lab view.

4. To make timetable system generic so that it can work equally well for different Schools, Colleges and Universities.

The system built will just require a standard format of data which once provided can generate time table for any school, college and university.

## 1.2 Problem Definition

Timetable generation manually requires time and it's difficult for the person to handle all the constraints that need to be considered. Generally, the hand operated system of time table preparation in colleges is very monotonous and time-consuming which results in either the same teachers ending up with more than one class at a time or a number of classes conflicting at the same classroom. Due to a non-automatic perspective, absolute utilization of resources has proven ineffective.

The solution for the above problems will be handled by the software ,as we will be providing a separate module of validation where all the constraints will be checked for providing an optimal solution and all the constraints will be satisfied by our system. The proposed system is used to generate time table automatically. This ensures the following features:

- Easier slot assigning
- Less time consumption
- No clashes of the resources

## 1.3 Scope

- Separate timetable for the individual class, faculty and labs will be generated automatically by this system.
- The project reduces time consumption and the pain in framing the timetable manually.
- The project is developed in such a way that, no slot clashes occur providing features to tailor the timetable as of wish.
- Additional features that is included in the project is that there is no headache of giving much input.
- This system can be used by Schools and Colleges to create Time-Table.

## 1.4 Technology Stack

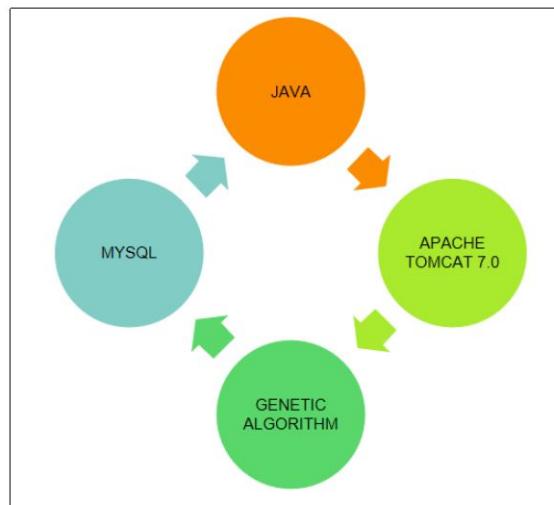


Figure 1.2: Technology Stack

- Java for coding
- Apache Tomcat as server
- Mysql for database
- Genetic Algorithm for processing

# Chapter 2

## Literature Review

[1] Asha, V. G., Babu, K. N. R. M. (2017). On-line help desk for college departmental activities. 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) IEEE (2017).

This paper refers to the varied computer machinery and software used to digitally create, collect, store, manipulate, and transfer the information needed for accomplishing basic tasks in a college department. The On-line help desk is a combination of multiple sub modules which are required for college department automation activities. This includes

1. Automatic Timetable generation which is used for generating the time table for the course, which also includes adding, deleting and updating a course, faculty, classroom and lab to generate the timetable.
2. Automatic Internals timetable generation which is used for generating the time table for the internal examination.
3. Student profile management which includes adding, deleting and updating student's personal information, previous education information, academic information and other non-academic information.
4. Library management which is used for barrow and return of books from department library by students and faculty.

Using Genetic algorithms all these modules are implemented. Genetic algorithm comes under the class of evolutionary algorithms that use the principle of natural selection to derive a set of solutions towards the optimal solution.

[2] Yang, X. F., Ayob, M., Nazri, M. Z. A. (2017). An investigation of timetable satisfaction factors for a practical university course timetabling problem. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI) IEEE (2017)

University Course Timetabling Problem (UCTP) is an educational timetabling problem that deals with the task of assigning educational events (lectures, tutorials, and laboratories) and lecturers to timeslots and classrooms or lab; to generate a weekly schedule. In this work, they have conducted a survey to investigate the satisfaction factors of a course timetable for a Practical University Course Timetabling Problem at Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia. This paper is organized as follows. Section 1 briefly describes the FTSM-UCTP i.e. Fakulti Teknologi dan Sains Maklumat-University Course

Timetabling Problem. Section 2 presents the questionnaires for both lecturers and students, and the result we obtained from this survey. In Section 3, they have discuss the result we gathered from the questionnaires. This survey has clearly demonstrated three types of schedule patterns based on students and lecturers satisfaction level:

The first type is the not-cared patterns which have no influence on the users' satisfaction when these patterns appear in the course timetable.

The second type, like in the literature, the penalty pattern, which will decrease the users' satisfaction level when this type of pattern exists in the timetable. the third type of schedule pattern is the reward patterns.

- [3] Ilham, N. I., Saat, E. H. M., Rahman, N. H. A., Rahman, F. Y. A., Kasuan, N. (2017). Auto- generate scheduling system based on expert system. 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE).

This paper presents an evolutionary algorithm (EA) based approach to solving a heavily constrained university timetabling problem. 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. The development of algorithm for timetabling in a large university department is illustrated in this paper by using second year electrical engineering (EE2) cohort as an example. Using following approach they have implemented their paper. Chromosome Representation, Initialization, Generation of initial population, crossover, mutation, evaluation, selection. This paper addresses the Timetabling Problem (TTP), which covers a very broad range of real problems faced continually in educational institutions, and we describe how Evolutionary Algorithms (EAs) can be employed to effectively address arbitrary instances of automated timetabling problem.

- [4] Anuja Chowdhary et al, TIMETABLE GENERATION SYSTEM, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February- 2014.

This project introduces a practical timetabling algorithm capable of taking care of both strong and weak constraints effectively, used in an automated timetabling system. So that each teacher and student can view their timetable once they are finalized for a given semester but they can't edit them. Timetable Generation System generates timetable for each class and teacher, in keeping with the availability calendar of teachers, availability and capacity of physical resources (such as classrooms, laboratories and computer room) and rules applicable at different classes, semesters, teachers and subjects level. In this paper the main component of is to produce the HTML based timetable even / odd semester sheet as the output. In which it takes various inputs from the user such as Teacher List, Course List, Semester List, Room List, Day List and Timeslot as well as various rules, facts and constraints using web based forms, which are stored in XML based knowledge base. This knowledge base serves as input to our Timetable Generator Algorithm residing on server machine. knowledge base is in the middle, because it is between our timetabling algorithm and GUI front end which is designed in the last. After the representation of KB is standardized, we designed the

timetabling algorithm.

[5] Bong Chia Lih, Sze San Nah, Bolhassan, N. A. (2015). A study on heuristic timetabling method for faculty course timetable problem. 2015 9th International Conference on IT in Asia (CITA) IEEE (2015).

This paper studies university course timetabling problem in a case study related to faculty management system, which is concerned with assigning students/lecturers to classes and time-slots. A two-stage heuristic approach is presented, where the initial stage groups the courses that are able to conduct simultaneously. The second stage then assigns the weekly timeslots for each group of courses, followed by venue for each course. Computational results are presented for the proposed solution using real data. It shows that the proposed solution is effective to handle the faculty course timetabling. A two stage heuristic is proposed to solve this course timetabling case study. This is due to its simplicity to cope with different hard and soft constraints in two stages:

Stage I (Course Grouping) in which all the courses will be divided into a few groups based on the following constraints:

1. Students can max enrol one course in each group.
2. Lecturers can max enrol one course in each group.
3. Takes into account the repeating students to enable them to enrol in certain courses.

Stage II (Timeslot Allocation) in this stage takes into account time-related constraints:

1. One timeslot allocated for only one group.
2. Spread the lecture time for each major/program evenly throughout the weekdays.

# Project Timeline Chart

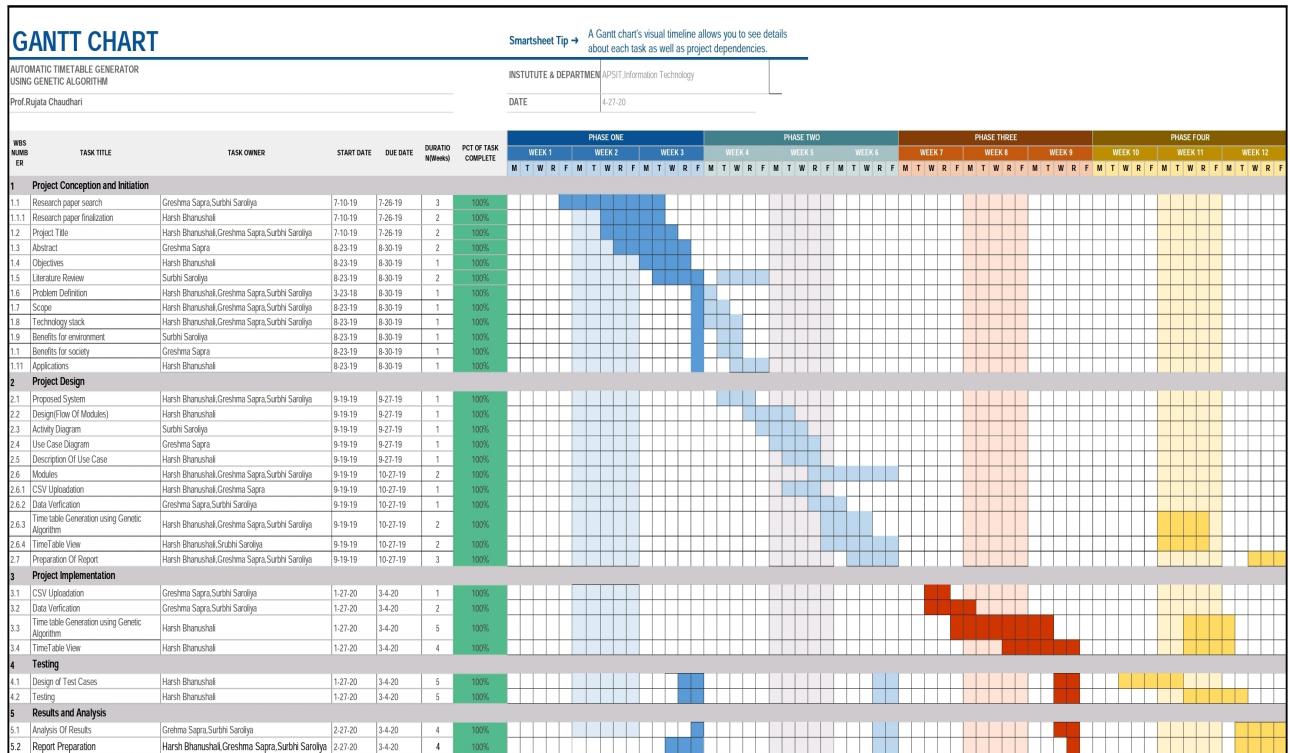


Figure 2.1: Project Timeline Chart

# Chapter 3

## Proposed System

Proposed system will take the basic inputs such as subjects, classrooms, faculties, laboratories, semesters through the loadsheet. These inputs will be processed further and the given set of constraints will be applied on it, further it will be validated and the system will check whether all the requirements are fulfilled or not. If all the requirements are fulfilled then only it will go for further processing. After validation, the system will generate the timetable which if approved will be stored in the database. It will provide different views of the timetable like class timetable for the students, laboratory time table for respective laboratories, faculty timetable for respective faculties and master timetable for timetable co-ordinator which can be printed anytime.

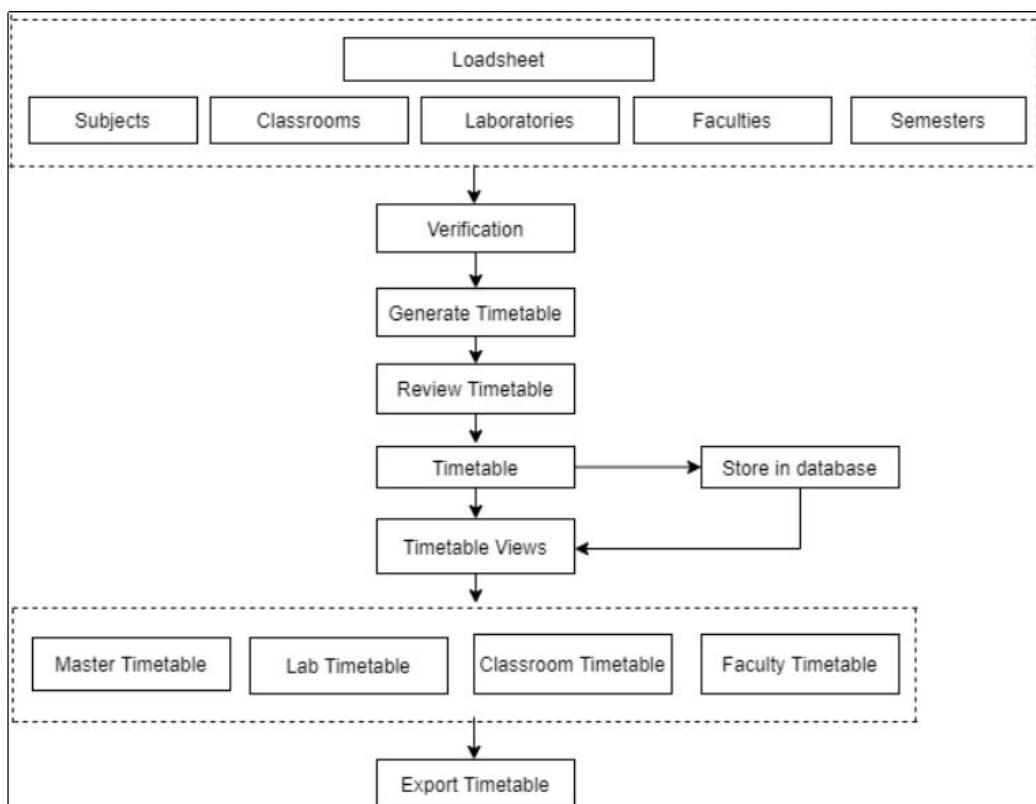


Figure 3.1: System Flow

The generated result will fulfill all the required constraints such as :

1. Slots for compulsory subjects should also get assigned properly taking into consideration there should be no conflict.
2. Slots should not overlap likewise if some faculty is teaching in second year then their slot should not be assigned in third year or final year during that particular time.
3. The generated timetable will also take care that each faculty workload should be maximum 10 hours per week for theory and 8 hours per week practical.
4. The generated timetable will also take care that each faculty workload should be equal to the load provided by the institute.

General Genetic Algorithm steps and their use in our project:

Step 1: Create initial population:

An initial population is created from a random selection of individuals. (which are analogous to chromosomes).

In our project population generation is the step where faculty abbreviation is the component which act as chromosome and the set of faculties abbreviation together forms our population.

Step 2: Evaluate fitness:

A fitness value is assigned to check solution depending on how close it actually is to solving the problem.

In our project the fitness value is the load assigned to faculty by the institute. There is a separate list created having data about the load of every faculty in reference with the index value of the faculty abbreviation list.

Step 3: Reproduce (and children mutate) :

The individuals with a greater fitness value are selected to generate new individuals (which can mutate after reproduction). New individual has features of both the parent individuals. (This generation is achieved by ‘crossing over’ method).

In our project this step comes into picture where the algorithm after assigning slots to a single faculty picks up another faculty from the faculty list and start placing that faculty into slots available, if the new faculty is provided a slot already occupied by the previous one the algorithm roll backs and assigns new slot to the faculty.

Step 4: Final generation:

If the latest generation has the fitness value equal to assigned value the solution is considered optimal.

In our project this stage is the stage at which the timetable gets generated trying all the combinations and providing the best solution.

## **Working of Project :**

As soon as the user opens the system he/she will see the home page where he/she will be asked to upload the load sheet ,there there will be a sample format file which the user can download and modify as per his/her requirement.Once,the user upload the loadsheet he/she will be shown the data of load sheet and will be asked to verify the data,after verification the data will go the algorithm where faculty name will be considered and used to generate the population for Genetic algorithm which will be stored in a array and according to the faculty name another array will be generated having load of that faculty,there will be 3 such pairs for second year,third year and final year respectively.

Once the population will be generated Genetic algorithm will start it's process by selecting any random faculty from array and placing the faculty at random position in time table grid according to the load assigned,this process will continue until all faculties have been assigned for their respective loads.The algorithm will check for clashes while assigning the faculty by checking the positions current status,if it would be free it will assign the faculty there or else go to another position.

In such a way the algorithm will assign all faculties without clashes for all the 3 semesters,once the time table will be generated it will be displayed to the user and if user approves the time table it will be stored in the database and if user doesn't likes the allotment he/she can regenerate the time table and the algorithm will run again and give a new output,user can get n combinations of output using the regenerate button.The approved and stored time table can be viewed any time.

# Chapter 4

## Project Design

### Basic Diagrams

Any real-world system is used by different users. The users can be developers, testers, business people, analysts, and many more. Hence, before designing a system, the architecture is made with different perspectives in mind. The most important part is to visualize the system from the perspective of different viewers. The better we understand the better we can build the system. Hence UML diagrams are used to represent the idea of users with the help of various diagrams. UML diagrams is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

### 4.1 System Architecture :

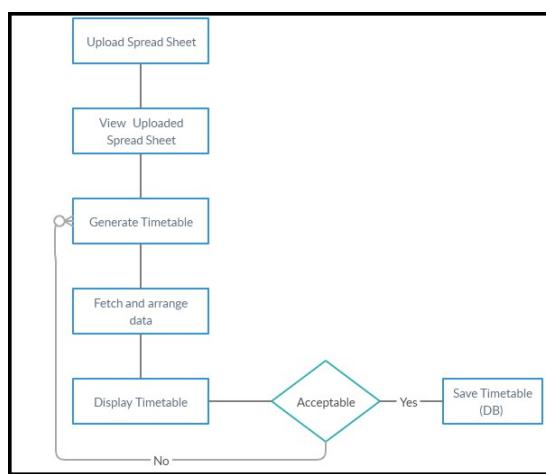


Figure 4.1: System Architecture

Architecture diagram can help system designers and developers visualize the high-level, overall structure of their system or application for the purpose of ensuring the system meets

their users' needs. You can also use architecture diagrams to describe patterns that are used throughout the design. It's somewhat like a blueprint that can be used as a guide for the convenience of discussing, improving, and following among your team.

Architecture diagram for Automatic Timetable Generation System involves overall structure of the system. To make this model working the project involves main modules such as upload spreadsheet (in which the loadsheet will be uploaded), View uploaded spreadsheet (in which the system will check if uploaded loadsheet is in proper format or not), Generate Timetable(in which the main genetic algorithm will work according to the given data), Fetch and arrange the data, Display Timetable(in which timetable will be displayed in different views), save Timetable(it will save the generated timetable in the databases).

## 4.2 Usecase Diagram :

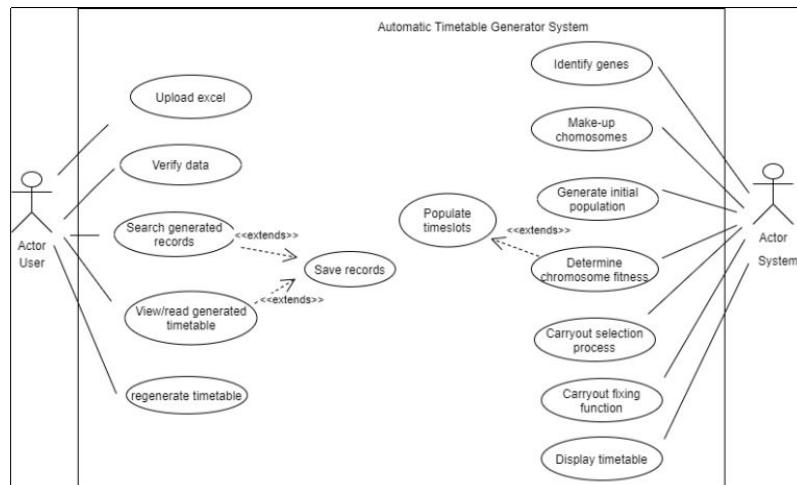


Figure 4.2: Usecase Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

Use case for Automatic Timetable Generation System involves various modules and working of it which is represented in diagrammatic way. Use case helps in understanding the function of managing system i.e. what are the role that are being played by managing system.

Hence use case provides an easier way for understanding the roles of the actors to the different users involved in project. In this project there are two actors, one is actor User which performs various tasks such as upload excel, verify data, view/read generated timetable etc and another actor is System which performs various tasks such as identify genes, generate initial population, display timetable etc. Thus different users can understand role of actor by having a glimpse of use case diagram.

### 4.3 Class Diagram :

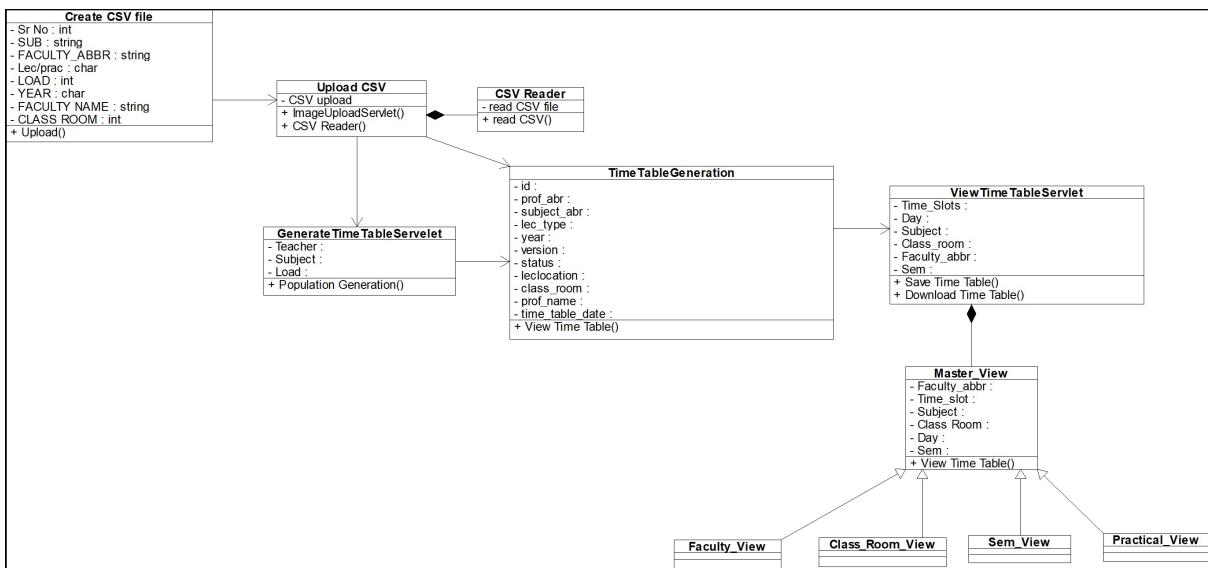


Figure 4.3: Class Diagram

Class diagram helps in understanding various different modules that are functioning such as create CSV file module, upload CSV module, CSV Reader module, Generate TimeTable-Servelet module(how timetable will be generated using Genetic algorithm), TimeTableGeneration module, ViewTimetableServelet module, and Masterview module(how timetable will be viewed in different views like classRoomview, Semview, Practicalview etc). Class diagram not only provide visual overview of project instead it helps in understanding various operations(such as upload() function, csv reader() function, Download Timetable() function etc.) It also helps in understanding various attributes(such as Timeslot, subjects, class, facultyabbr etc.) involved in the project.

Class diagram also explains the interfacing components that plays an vital role in functioning of main modules. For example in this project their are some interface modules one is, after creating csv file we will take this as a input to upload the csv file module for generating the timetable and other module is after timetable generation module using genetic algorithm we will take this as a input and interface it with view TimetableServelet module. Hence we can conclude that class diagram not only provides brief description of project and its module but also provides a layout of executable code along with parameters.

## 4.4 Sequence Diagram :

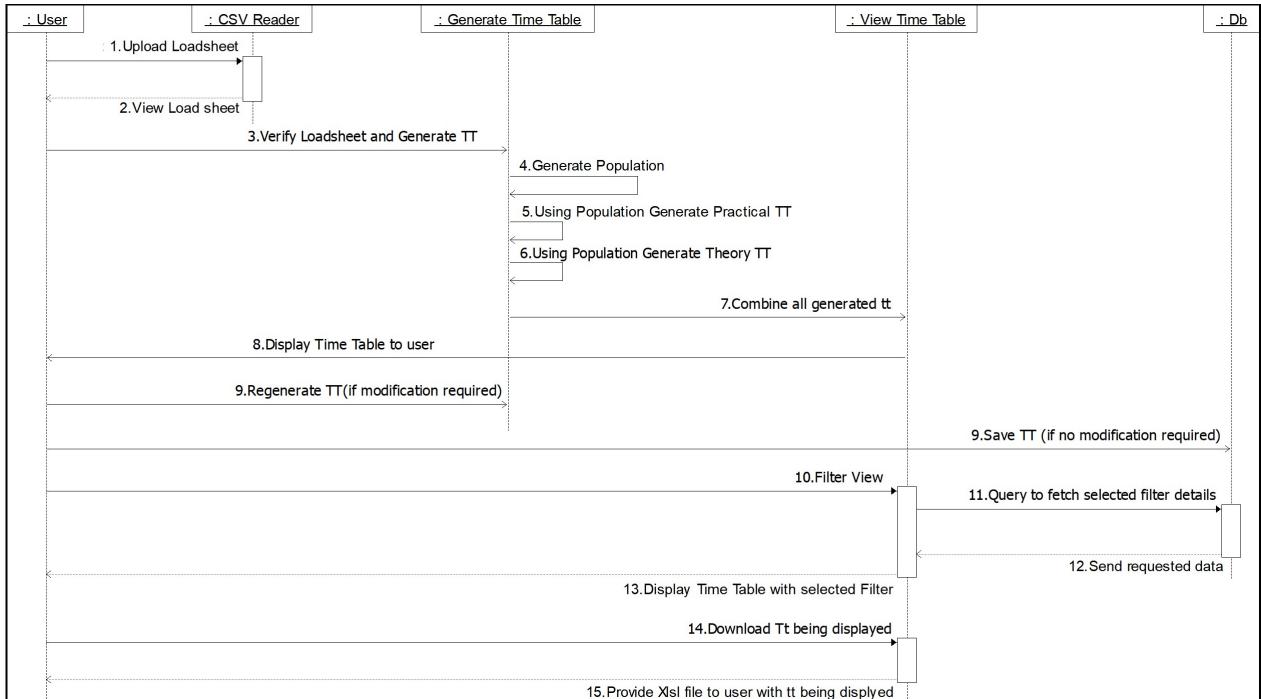


Figure 4.4: Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

Sequence diagram for Automatic Timetable Generation System involves various objects, classes and functions of it which is represented in graphical way. Sequence helps in understanding the object interactions arranged in time sequence i.e in which order or in which sequence the system is going to process. There are five different objects and classes are available and they are User, CSV Reader, Generate Timetable, View Timetable and Db. There are different sequence of messages exchanged between the objects are carried out the in this scenario which shows functionality of the system and they are upload loadsheet, Verify loadsheet and generate tt etc. Which will be exchanged between different objects.

#### 4.5 Activity Diagram :

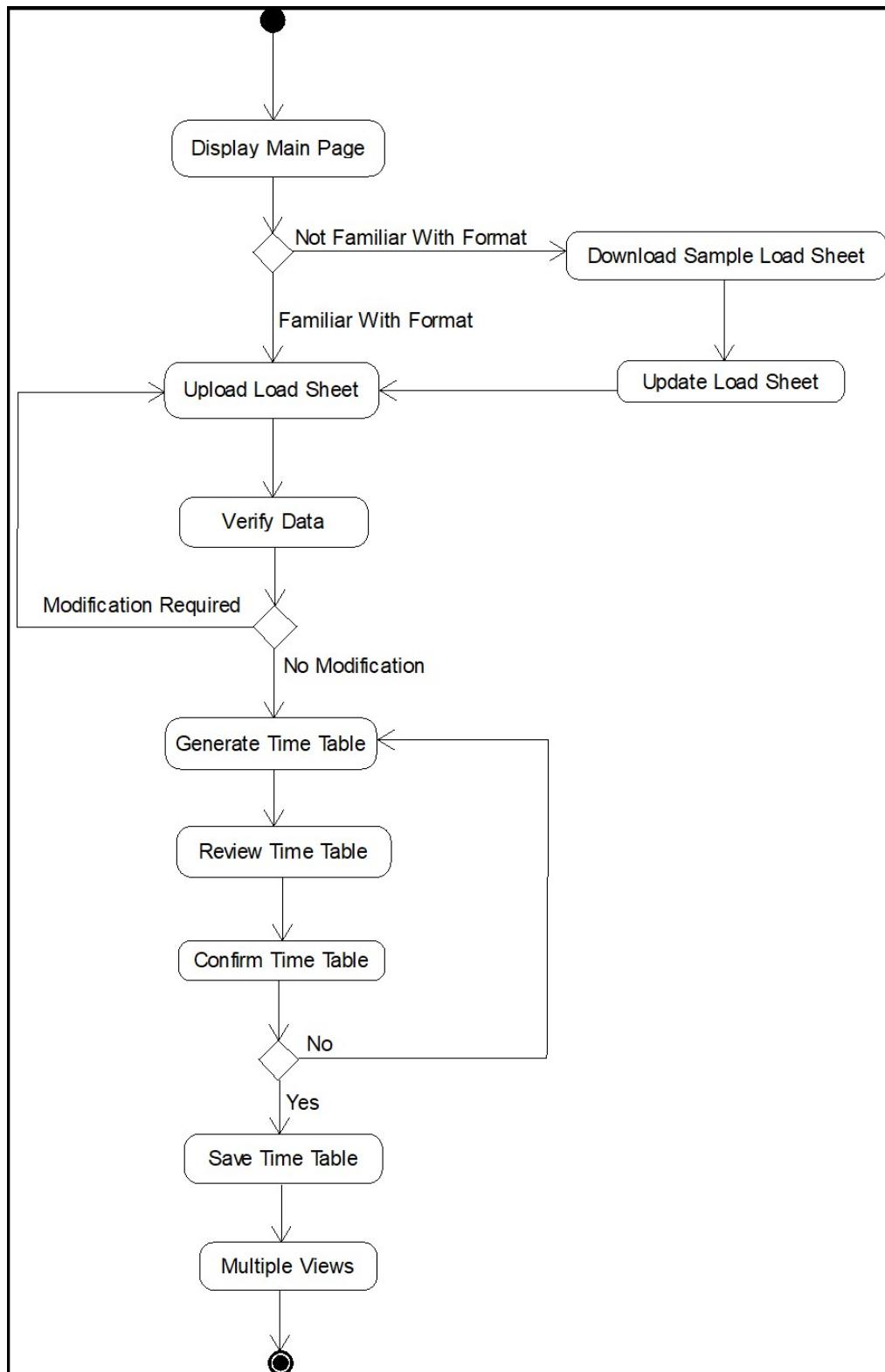


Figure 4.5: Activity Diagram

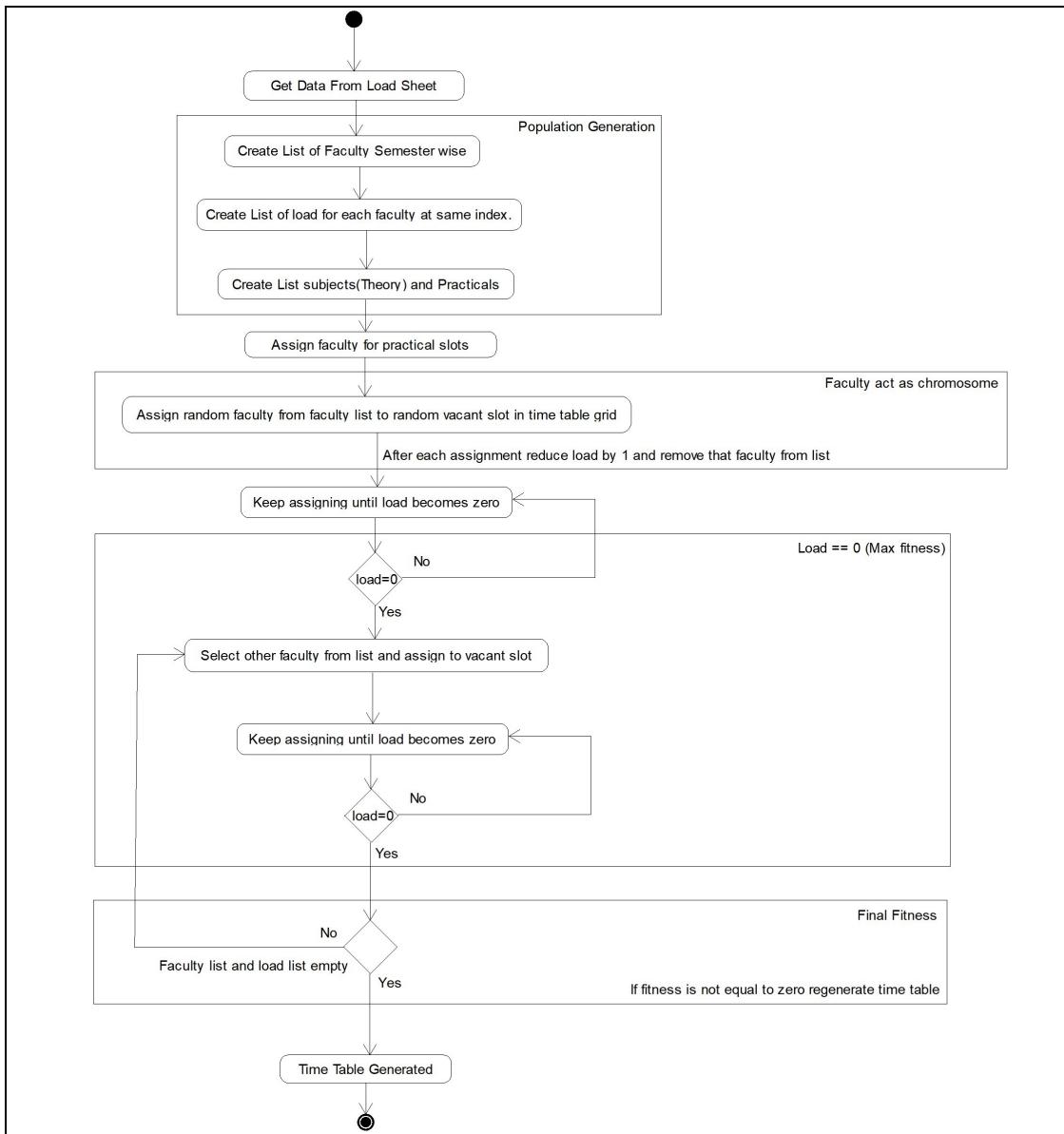


Figure 4.6: Activity Diagram for Generation of timetable

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The control flow is drawn from one operation to another. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

Activity diagram in this project includes how sequence of activity will be processed flowing from one activity to another activity represented by a flowchart. In our project Automatic Timetable Generation System there are two activity diagrams available one is for overall activity of project in which project flow is divided into of sequence of activity

following one after the another starting from Display main page, upload loadsheet, generate timetable, save timetable, modify views etc.

Another Activity of this project is Generating Timetable which includes flow of activity in which how timetable will be generated using genetic algorithm. It involves different activities such as population generation, faculty act as a chromosomes, fitness check, generating final timetable.

# Chapter 5

## Project Implementation

### 5.1 Code Snippets :

#### 1.Database Connection:

```
1 package com.timetable.utility;
2
3 import java.sql.Connection;
4
5 public class MyConnection {
6     static Connection conn=null;
7     public static Connection getConnectionobj(){
8         {
9             try {
10                 {
11                     Class.forName("com.mysql.jdbc.Driver");
12                     System.out.println("DRIVER LOADED");
13                     conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/timetabledb","root","123456");
14                     System.out.println("CONNECTION SUCCESS");
15
16                     return conn;
17                 }
18             catch (Exception e) {
19                 // TODO: handle exception
20             }
21         }
22         return null;
23     }
24 }
25 }
```

#### 2.Population Generation:

```
List<String> seTeachNameL=population.get("S").get("teacher");
List<String> teTeachNameL=population.get("T").get("teacher");
List<String> beTeachNameL=population.get("B").get("teacher");

String[] seTeachName = seTeachNameL.toArray(new String[seTeachNameL.size()]);
String[] teTeachName = teTeachNameL.toArray(new String[teTeachNameL.size()]);
String[] beTeachName = beTeachNameL.toArray(new String[beTeachNameL.size()]);

List<String> seSubNameL=population.get("S").get("subject");
List<String> teSubNameL=population.get("T").get("subject");
List<String> beSubNameL=population.get("B").get("subject");

String[] seSubName = seSubNameL.toArray(new String[seSubNameL.size()]);
String[] teSubName = teSubNameL.toArray(new String[teSubNameL.size()]);
String[] beSubName = beSubNameL.toArray(new String[beSubNameL.size()]);

List<String> seLoadL=population.get("S").get("load");
List<String> teLoadL=population.get("T").get("load");
List<String> beLoadL=population.get("B").get("load");

String[] seLoad = seLoadL.toArray(new String[seLoadL.size()]);
String[] teLoad = teLoadL.toArray(new String[teLoadL.size()]);
String[] beLoad = beLoadL.toArray(new String[beLoadL.size()]);
```

### 3.Random slot allocation to random faculty of final year:

```
Z3: for(int i=0;i<teacher2.length;i++)  
{  
    int rnd=abc.getRandomNumber(teacher2);  
    String teacherName2=teacher2[i];  
    int count2=load2[i];  
    while(count2>0 && thirdCount<1000){  
  
        for(int m=0;m<sem2[0].length;m++){  
  
            if(bePracDay!=m){  
                thirdCount++;  
                int rndNo2=abc.getRandomNumber(lecNo2);  
                int lecN2=lecNo2[rndNo2];  
                //String pracTime2[]=practical2[m].split(",");  
  
                if(sem2[lecN2][m]==null && count2>0 && checkPracs(lecN2, teacherName2, m) && !teacherName2.equals(sem[lecN2][m]) && !teacherName2.equals(sem1[lecN2][m]))  
                {  
                    sem2[lecN2][m]=teacherName2;  
                    count2 --;  
                }  
            }  
        }  
  
        if(thirdCount>=1000){  
  
            List<String[][]> arrayList=new ArrayList<String[][]>();  
            arrayList.add(sem);  
            arrayList.add(sem1);  
            arrayList.add(sem2);  
            for(Object obj:arrayList){  
                String curr[][]=(String[][])obj;  
                for(int k=0;k<curr.length;k++){  
                    for(int m1=0;m1<curr[i].length;m1++){  
                        curr[k][m1]=null;  
                    }  
                }  
            }  
  
            firstCount=0;  
            secondCount=0;  
            thirdCount=0;  
            continue L;  
        }  
    }  
}
```

### 4.Random slot allocation to random faculty of third year:

```
Z1: for(int i=0;i<teacher1.length;i++)  
{  
    int rnd=abc.getRandomNumber(teacher1);  
    String teacherName1=teacher1[i];  
    int count1=load1[i];  
  
    while(count1>0 && secondCount<1000){  
        for(int m=0;m<sem1[0].length;m++)  
        {  
            secondCount++;  
            int rndNo1=abc.getRandomNumber(lecNo1);  
            int lecN1=lecNo1[rndNo1];  
            //String pracTime1[]=practical1[m].split(",");  
  
            if(sem1[lecN1][m]==null && count1>0 && checkPracs(lecN1, teacherName1, m) && !teacherName1.equals(sem[lecN1][m]))  
            {  
                sem1[lecN1][m]=teacherName1;  
                count1 --;  
            }  
        }  
  
        if(secondCount>1000){  
            firstCount=0;  
            secondCount=0;  
  
            List<String[][]> arrayList=new ArrayList<String[][]>();  
            arrayList.add(sem);  
            arrayList.add(sem1);  
            arrayList.add(sem2);  
  
            for(Object obj:arrayList){  
                String curr[][]=(String[][])obj;  
                for(int k=0;k<curr.length;k++){  
                    for(int m1=0;m1<curr[i].length;m1++){  
                        curr[k][m1]=null;  
                    }  
                }  
            }  
            continue L;  
        }  
    }  
}
```

5. Random slot allocation to random faculty of second year:

```

E: for(int i=0;i<teacher.length;i++)
{
    int rnd=abc.getRandomNumber(teacher);
    String teacherName=teacher[i];
    int count=load[i];
    while(count>0 && firstCount<1000)
    {
        for(int m=0;m<sem[0].length;m++)
        {
            firstCount++;
            int rndNo=abc.getRandomNumber(lecNo);
            int lecN=lecNo[rndNo];
            String practime[] = practical[m].split(",");
            //String pNames = sem[lecN][Integer.parseInt(practime[0])];

            if(sem[lecN][m]==null && count>0 && checkPracs(lecN, teacherName, m))
            {
                sem[lecN][m]=teacherName;
                count--;
            }
        }
    }

    if(firstCount>1000){
        firstCount=0;
        List<String[][]> arrayList=new ArrayList<String[][]>();
        arrayList.add(sem);
        arrayList.add(sem1);
        arrayList.add(sem2);
        for(Object obj:arrayList){
            String curr[][]=(String[][])(obj);
            for(int k=0;k<curr.length;k++){

                for(int m1=0;m1<curr[i].length;m1++)
                {
                    curr[k][m1]=null;
                }
            }
        }
        continue L;
    }
}

```

## 6.Fixing slots for practical:

```
public class TimeTableGeneration {  
  
    static String [][] sem1=null;  
    static String [][] sem1=null;  
    static String [][] sem2=null;  
  
    static String practical1[]={ "3,4","0,1","3,4","4,5","6,7" };  
    static String practical1[]={ "3,4","6,7","6,7","0,1","0,1" };  
    static String practical2[]={ "6,7","3,4","0,1","6,7","3,4" };
```

#### **7.Design for displaying Timetable:**

## 5.2 Result

### 1. Uploaded load sheet verification:

Load Sheet Verification						
NO	Subject Abbreviation	Prof Abbreviation	L/P	LOAD	YEAR	Prof Name
1	AM	NN	L	4	S	Nancy
2	CN	ND	L	4	S	Neha
3	OS	YP	L	4	S	Yamini
4	COA	SJ	L	4	S	Sonal
5	AT	KU	L	4	S	Kaushiki
6	UL	SK	P	6	S	Sneha
7	MPL	SJ	P	6	S	Sonal
8	NL	KD	P	6	S	Kiran
9	PL	AA	P	6	S	Anagha
10	SEPM	RC	L	4	T	Rujata
11	DMBI	GK	L	4	T	Gitanjali
12	CCS	NS	L	4	T	Nahid
13	WN	SK	L	4	T	Sneha
14	AIP	NS	L	4	T	Nahid
15	DF	RC	L	4	T	Rujata
16	SDL	RC	P	6	T	Rujata
17	SNL	KU	P	6	T	Kaushiki
18	CSDL	NS	P	6	T	Nahid
19	BIL	GK	P	6	T	Gitanjali
20	BDA	VB	L	4	B	Vishal

### 2.Master Time table generated:

AUTOMATIC TIME-TABLE GENERATION SYSTEM										
DEPARTMENT OF INFORMATION TECHNOLOGY ( IT )										
TIME/DAY	9.10 to 10.05	10.05 to 11.00	11.00 to 11.55	11.55 12.25 to 1.20	1.20 to 2.15	2.15 to 3.10	3.10 3.30 to 4.25	3.30 to 4.25	4.25 to 5.20	
Monday	SE	AM - NN - 206	CN - ND - 206	(UL-SK-313)/(MPL-SJ-302)/(NL-KD-317) (SDL-RC-301)/(SNL-KU-308)/(CSDL-NS-317)	(UL-SK-313)/(MPL-SJ-302)/(NL-KD-317) (SDL-RC-301)/(SNL-KU-308)/(CSDL-NS-317)	AT - KU - 206	COA - SJ - 206			
	TE	DMBI - GK - 205	CCS - NS - 205	SEPM - RC - 205	UID - AA - 207	BDI - VB - 207	DMBI - GK - 205	WN - SK - 205	WN - SK - 205	
	BE	ERP - ND - 207	IOE - SJ - 207			BDI - VB - 207	(BDL-VB-301)/(BDL-VB-301)/(OEL-KU-308)	(BDL-VB-301)/(BDL-VB-301)/(OEL-KU-308)		
Tuesday	SE	(UL-SK-313)/(MPL-SJ-302)/(PL-AA-303)	(UL-SK-313)/(MPL-SJ-302)/(PL-AA-303)				AT - KU - 206	COA - SJ - 206	AM - NN - 206	
	TE	SEPM - RC - 205			CCS - NS - 205	SEPM - RC - 205	WN - SK - 205	(SDL-RC-301)/(SNL-KU-308)/(BIL-GK-303)	(SDL-RC-301)/(SNL-KU-308)/(BIL-GK-303)	
	BE	ERP - ND - 207		IOE - SJ - 207	(BDL-VB-301)/(BDL-VB-301)/(RPL-SF-313)	(BDL-VB-301)/(BDL-VB-301)/(RPL-SF-313)	EVM - CV - 207	UID - AA - 207	ERP - ND - 207	
Wednesday	SE	AM - NN - 206			(UL-SK-313)/(NL-KD-317)/(PL-AA-303)	(UL-SK-313)/(NL-KD-317)/(PL-AA-303)	OS - YP - 206	OS - YP - 206	CN - ND - 206	
	TE	SEPM - RC - 205		CCS - NS - 205	SEPM - RC - 205	DMBI - GK - 205	CCS - NS - 205	(SDL-RC-301)/(CSDL-NS-317)/(BIL-GK-303)	(SDL-RC-301)/(CSDL-NS-317)/(BIL-GK-303)	
	BE									
Thursday	SE	OS - YP - 206	COA - SJ - 206	AT - KU - 206	(MPL-SJ-302)/(NL-KD-317)/(PL-AA-303)	(MPL-SJ-302)/(NL-KD-317)/(PL-AA-303)	OS - YP - 206	OS - YP - 206	CN - ND - 206	
	TE	(SNL-KU-308)/(CSDL-NS-317)/(BIL-GK-303)	(SNL-KU-308)/(CSDL-NS-317)/(BIL-GK-303)	CCS - NS - 205	CCS - NS - 205	EVM - CV - 207	EVM - CV - 207	(BIL-VB-301)/(OEL-KU-308)/(RPL-SF-313)	SEPM - RC - 205	
	BE	HRM - SN - 207	HRM - SN - 207	IOE - SJ - 207	IOE - SJ - 207	BDI - VB - 207	WNN - SK - 205	(BIL-VB-301)/(OEL-KU-308)/(RPL-SF-313)	(BIL-VB-301)/(OEL-KU-308)/(RPL-SF-313)	
Friday	SE	AT - KU - 206	CN - ND - 206				AM - NN - 206	COA - SJ - 206		
	TE	CCS - NS - 205	CCS - NS - 205	SEPM - RC - 205	SEPM - RC - 205		WN - SK - 205		ERP - ND - 207	
	BE	UID - AA - 207	BDI - VB - 207	UID - AA - 207	(BDL-VB-301)/(OEL-KU-308)/(RPL-SF-313)	(BDL-VB-301)/(OEL-KU-308)/(RPL-SF-313)	HRM - SN - 207	IOE - SJ - 207		

### 3.BE time table with 1 day blank for project slot:

DEPARTMENT OF INFORMATION TECHNOLOGY (IT)										
TIME/DAY	9.10 to 10.05	10.05 to 11.00	11.00 to 11.55	11.55 - 12.20	12.20 to 1.20	1.20 to 2.15	2.15 to 3.00	3.10 - 3.20	3.30 to 4.25	4.25 to 5.20
<b>Monday</b>	SE									
	TE									
	BE	ERP - ND - 207	IOE - SJ - 207			UID - AA - 207	BDA - VB - 207	BDA - VB - 207		(BDL-VB-301)(BDL-VB-301)(OEL-KU-306)
<b>Tuesday</b>	SE									
	TE									
	BE	ERP - ND - 207		IOE - SJ - 207		(BDL-VB-301)(BDL-VB-301)(RPL-SF-212)	(BDL-VB-301)(BDL-VB-301)(RPL-SF-212)	EVM - CV - 207		UID - AA - 207
<b>Wednesday</b>	SE									
	TE									
	BE									
<b>Thursday</b>	SE									
	TE									
	BE	HRM - SN - 207	HRM - SN - 207	IOE - SJ - 207		BDA - VB - 207	EVM - CV - 207	EVM - CV - 207		(BDL-VB-301)(OEL-KU-306)(RPL-SF-313)
<b>Friday</b>	SE									
	TE									
	BE	UID - AA - 207	BDA - VB - 207	UID - AA - 207		(BDL-VB-301)(OEL-KU-306)(RPL-SF-212)	(BDL-VB-301)(OEL-KU-306)(RPL-SF-212)	HRM - SN - 207		ERP - ND - 207

### 4.Faculty List:

DEPARTMENT OF INFORMATION TECHNOLOGY									
	10.05 to 11.00	11.00 to 11.55	11.55 - 12.20	12.20 to 1.20	1.20 to 2.15	2.15 to 3.00	3.10 - 3.20	3.30 to 4.25	4.25 to 5.20
Gitanjali Rujuta									
Kausik Nahid									
Sneha Sonal									
Anagha Nancy									
Yamini Neha									
Kiran Sameer									
Vishal Selvin									
Civil									

### 5.Faculty View:

DEPARTMENT OF INFORMATION TECHNOLOGY (IT)										
TIME/DAY	9.10 to 10.05	10.05 to 11.00	11.00 to 11.55	11.55 - 12.20	12.20 to 1.20	1.20 to 2.15	2.15 to 3.00	3.10 - 3.20	3.30 to 4.25	4.25 to 5.20
<b>Monday</b>	SE					(SNL-KU-306)	(SNL-KU-306)			(OEL-KU-306) (OEL-KU-306)
	TE									
	BE									
<b>Tuesday</b>	SE									
	TE									
	BE									
<b>Wednesday</b>	SE									
	TE									
	BE									
<b>Thursday</b>	SE					AT - KU - 206				
	TE	(SNL-KU-306)		(SNL-KU-306)						
	BE									
<b>Friday</b>	SE	AT - KU - 206								
	TE									
	BE					(OEL-KU-306)	(OEL-KU-306)			

## 6. Class List:

DEPARTMENT OF INFORMATION TECHNOLOGY ( IT )							
05 to 11.00	11.00 to 11.55	11.55 - 12.29	12.25 to 1.20	1.20 to 2.15	2.15 to 3.10	3.30 - 3.39	3.30 to 4.25
			(SDL-RC-301)	(SOL-R-301)			
				(BDL-VB-301)/(BDL-VB-301)	(BDL-VB-301)/(BDL-VB-301)		
							(SDL-RC-301)

## 7. Class View:

AUTOMATIC TIME-TABLE GENERATION SYSTEM							
DEPARTMENT OF INFORMATION TECHNOLOGY ( IT )							
TIME/DAY	9.30 to 10.05	10.05 to 11.00	11.00 to 11.55	11.55 - 12.29	12.25 to 1.20	1.20 to 2.15	2.15 to 3.10
Monday	SE				(SDL-RC-301)	(SOL-R-301)	
	TE						(BDL-VB-301)/(BDL-VB-301)
	BE						(BDL-VB-301)/(BDL-VB-301)
Tuesday	SE				(BDL-VB-301)/(BDL-VB-301)	(BDL-VB-301)/(BDL-VB-301)	
	TE						(SDL-RC-301)
	BE						(SDL-RC-301)
Wednesday	SE						(SDL-RC-301)
	TE						(SDL-RC-301)
	BE						
Thursday	SE						(BDL-VB-301)
	TE						(BDL-VB-301)
	BE						
Friday	SE				(BDL-VB-301)	(BDL-VB-301)	
	TE						
	BE						

## 8. Practicals:

AUTOMATIC TIME-TABLE GENERATION SYSTEM							
DEPARTMENT OF INFORMATION TECHNOLOGY ( IT )							
TIME/DAY	9.30 to 10.05	10.05 to 11.00	11.00 to 11.55	11.55 - 12.29	12.25 to 1.20	1.20 to 2.15	2.15 to 3.10
Monday	SE				(UL-SH-313)/(MPL-RJ-303)/(PL-AA-303)	(UL-SH-313)/(MPL-RJ-303)/(PL-AA-303)	
	TE				(SOL-R-301)/(SOL-R-301)	(SOL-R-301)/(SOL-R-301)	
	BE						(BDL-VB-301)/(BDL-VB-301)
Tuesday	SE	(UL-SH-313)/(MPL-RJ-303)/(PL-AA-303)	(UL-SH-313)/(MPL-RJ-303)/(PL-AA-303)		(BDL-VB-301)/(BDL-VB-301)	(BDL-VB-301)/(BDL-VB-301)	
	TE						(BDL-VB-301)/(BDL-VB-301)
	BE						
Wednesday	SE				(UL-SH-313)/(MPL-RJ-303)/(PL-AA-303)	(UL-SH-313)/(MPL-RJ-303)/(PL-AA-303)	
	TE						(SOL-R-301)/(SOL-R-301)
	BE						(SOL-R-301)/(SOL-R-301)
Thursday	SE				(MPL-SJ-303)/(UL-HD-317)/(PL-AA-303)	(MPL-SJ-303)/(UL-HD-317)/(PL-AA-303)	
	TE	(SOL-RU-306)/(SOL-RU-306)	(SOL-RU-306)/(SOL-RU-306)				(BDL-VB-301)/(BDL-VB-301)
	BE						(BDL-VB-301)/(BDL-VB-301)
Friday	SE				(BDL-VB-301)/(CSEL-KU-303)	(BDL-VB-301)/(CSEL-KU-303)	
	TE						
	BE						

## 9.Previous available Time table:

DEPARTMENT OF INFORMATION TECHNOLOGY ( IT )								download	Previous Time-Table	
10.05 to 11.00									25-04-2020 version 1	
11.00 to 11.55									25-04-2020 version 2	
12.25 to 1.20									04-05-2020 version 1	
1.20 to 2.15									04-05-2020 version 2	
2.15 to 3.10									3.10	
3.30									3.30 to 4.25	

## 10.Master view of previous time table:

DEPARTMENT OF INFORMATION TECHNOLOGY ( IT )											download
TIME/DAY	9.10 to 10.05	10.05 to 11.00	11.00 to 11.55	11.55 12.25	12.25 to 1.20	1.20 to 2.15	2.15 to 3.10	3.10 3.30	3.30 to 4.25	4.25 to 5.20	
Monday	SE AM - NN - 206	AT - KU - 206			(UL-SK-313)/(MPL-SJ-302)/(NL-KD-303)	(UL-SK-313)/(MPL-SJ-302)/(NL-KD-303)				OS - YP - 206	
	TE WN - SK - 205			WN - SK - 205	(DF-RC-205)/(DF-RC-205)/(SNL-KU-308)	(DF-RC-205)/(DF-RC-205)/(SNL-KU-308)	SEPM - RC - 205			DMBI - GK - 205	
	BE EVM - CV - 207	IOE - SJ - 207		BDA - VB - 207	ERP - ND - 207	UID - AA - 207	EVM - CV - 207			(BDL-VB-301)/(BDL-VB-301)/(IOEL-KU-308)	(BDL-VB-301)/(BDL-VB-301)/(IOEL-KU-308)
Tuesday	SE (UL-SK-313)/(MPL-SJ-302)/(PL-AA-303)	(UL-SK-313)/(MPL-SJ-302)/(PL-AA-303)	OS - VP - 206		AM - NN - 206	COA - SJ - 206				CN - ND - 206	
	TE SEPM - RC - 205	CCS - NS - 205	CCS - NS - 205		CCS - NS - 205	DMBI - GK - 205				(DF-RC-205)/(CSDL-NS-317)/(BIL-GK-303)	(DF-RC-205)/(CSDL-NS-317)/(BIL-GK-303)
	BE BDA - VB - 207			ERP - ND - 207	(BDL-VB-301)/(BDL-VB-301)/(RPL-SF-313)	(BDL-VB-301)/(BDL-VB-301)/(RPL-SF-313)	UID - AA - 207			HRM - SN - 207	(BDL-VB-301)/(BDL-VB-301)/(IOEL-KU-308)
Wednesday	SE COA - SJ - 206			CN - ND - 206	(UL-SK-313)/(NL-KD-317)/(PL-AA-303)	(UL-SK-313)/(NL-KD-317)/(PL-AA-303)	AT - KU - 206			OS - YP - 206	COA - SJ - 206
	TE CCS - NS - 205			DMBI - GK - 205	ERP - ND - 207	IOE - SJ - 207	SEPM - RC - 205			(SNL-KU-308)/(DF-RC-205)/(BIL-GK-303)	(SNL-KU-308)/(DF-RC-205)/(BIL-GK-303)
	BE (BDL-VB-301)/(IOEL-KU-308)/(RPL-SF-313)	(BDL-VB-301)/(IOEL-KU-308)/(RPL-SF-313)	HRM - SN - 207			UID - AA - 207				ERP - ND - 207	BDA - VB - 207
Thursday	SE AM - NN - 206				(MPL-SJ-302)/(NL-KD-317)/(PL-AA-303)	(MPL-SJ-302)/(NL-KD-317)/(PL-AA-303)	AT - KU - 206			CN - ND - 206	
	TE (CSDL-NS-317)/(DF-RC-205)/(BIL-GK-303)	(CSDL-NS-317)/(DF-RC-205)/(BIL-GK-303)	SEPM - RC - 205		CCS - NS - 205	DMBI - GK - 205				CCS - NS - 205	WN - SK - 205
	BE				ERP - ND - 207	(BDL-VB-301)/(IOEL-KU-308)/(RPL-SF-313)	(BDL-VB-301)/(IOEL-KU-308)/(RPL-SF-313)	IOE - SJ - 207			UID - AA - 207
Friday	SE AM - NN - 206	COA - SJ - 206	AT - KU - 206		CN - ND - 206	CCS - NS - 205				OS - YP - 206	AT - KU - 206
	TE (DF-RC-205)/(SNL-KU-308)/(CSDL-NS-317)	(DF-RC-205)/(SNL-KU-308)/(CSDL-NS-317)	WN - SK - 205		ERP - ND - 207	(BDL-VB-301)/(IOEL-KU-308)/(RPL-SF-313)	(BDL-VB-301)/(IOEL-KU-308)/(RPL-SF-313)	IOE - SJ - 207			
	BE EVM - CV - 207	HRM - SN - 207	IOE - SJ - 207								

## 11.Required CSV file format for running the algorithm:

NO	SUB	FACULTY_ABBR	Lec/prac	LOAD	YEAR	FACULTY NAME	CLASS ROOM
1	AM	NN	L	4 S	Nancy	206	
2	CN	ND	L	4 S	Neha	206	
3	OS	YP	L	4 S	Yamini	206	
4	COA	SJ	L	4 S	Sonal	206	
5	AT	KU	L	4 S	Kaushiki	206	
6	UL	SK	P	6 S	Sneha	313	
7	MPL	SI	P	6 S	Sonal	302	
8	NL	KD	P	6 S	Kiran	317	
9	PL	AA	P	6 S	Anagha	303	
10	SEPM	RC	L	4 T	Rujata	205	
11	DMBI	GK	L	4 T	Gitanjali	205	
12	CCS	NS	L	4 T	Nahid	205	
13	WN	SK	L	4 T	Sneha	205	
14	AIP	NS	L	4 T	Nahid	205	
15	DF	RC	L	4 T	Rujata	205	
16	SDL	RC	P	6 T	Rujata	301	
17	SNL	KU	P	6 T	Kaushiki	308	
18	CSDL	NS	P	6 T	Nahid	317	
19	BIL	GK	P	6 T	Gitanjali	303	
20	BDA	VB	L	4 B	Vishal	207	
21	IOE	SI	L	4 B	Sonal	207	
22	UID	AA	L	4 B	Anagha	207	
23	ERP	ND	L	4 B	Neha	207	
24	BDA	VB	P	6 B	Vishal	301	
25	DL	VB	P	6 B	Vishal	317	
26	IOEL	KU	P	6 B	Kaushiki	308	
27	RPL	SE	P	6 B	Selvin	313	
28	EVM	CV	L	3 B	Civil	207	
29	HRM	SN	L	3 B	Sameer	207	

## 12.Database:

<b>id</b>	<b>prof_abr</b>	<b>subject_abr</b>	<b>lec_type</b>	<b>year</b>	<b>version</b>	<b>status</b>	<b>ledocation</b>	<b>class_room</b>	<b>prof_name</b>	<b>time_table_date</b>
269	GK	DMBI	L	TE	2	active	0\$0	205	Gitanjali	2020-05-04
270	RC	SEPM	L	TE	2	active	0\$1	205	Rujata	2020-05-04
271	RC	SEPM	L	TE	2	active	0\$2	205	Rujata	2020-05-04
272	KU-NS-GK	SNL-CSDL-BIL	P	TE	2	active	0\$3	308-317-303	Kaushiki-Nahid-Gitanjali	2020-05-04
273	NS	CCS	L	TE	2	active	0\$4	205	Nahid	2020-05-04
274	NS	CCS	L	TE	2	active	1\$0	205	Nahid	2020-05-04
275	KU-NS-GK	SNL-CSDL-BIL	P	TE	2	active	1\$3	308-317-303	Kaushiki-Nahid-Gitanjali	2020-05-04
276	NS	CCS	L	TE	2	active	1\$4	205	Nahid	2020-05-04
277	RC	SEPM	L	TE	2	active	2\$0	205	Rujata	2020-05-04
278	NS	CCS	L	TE	2	active	2\$2	205	Nahid	2020-05-04
279	NS	CCS	L	TE	2	active	2\$3	205	Nahid	2020-05-04
280	RC	SEPM	L	TE	2	active	2\$4	205	Rujata	2020-05-04
281	RC-KU-NS	SDL-SNL-CSDL	P	TE	2	active	3\$0	301-308-317	Rujata-Kaushiki-Nahid	2020-05-04
282	NS	CCS	L	TE	2	active	3\$1	205	Nahid	2020-05-04
283	RC	SEPM	L	TE	2	active	3\$2	205	Rujata	2020-05-04
284	GK	DMBI	L	TE	2	active	3\$3	205	Gitanjali	2020-05-04
285	RC	SEPM	L	TE	2	active	3\$4	205	Rujata	2020-05-04
286	RC-KU-NS	SDL-SNL-CSDL	P	TE	2	active	4\$0	301-308-317	Rujata-Kaushiki-Nahid	2020-05-04
287	RC	SEPM	L	TE	2	active	4\$1	205	Rujata	2020-05-04
288	GK	DMBI	L	TE	2	active	4\$2	205	Gitanjali	2020-05-04
289	NS	CCS	L	TE	2	active	4\$3	205	Nahid	2020-05-04
290	GK	DMBI	L	TE	2	active	5\$0	205	Gitanjali	2020-05-04
291	SK	WN	L	TE	2	active	5\$1	205	Sneha	2020-05-04
292	NS	CCS	L	TE	2	active	5\$2	205	Nahid	2020-05-04
293	SK	WN	L	TE	2	active	5\$4	205	Sneha	2020-05-04
294	SK	WN	L	TE	2	active	6\$0	205	Sneha	2020-05-04
295	RC-KU-GK	SDL-SNL-BIL	P	TE	2	active	6\$1	301-308-303	Rujata-Kaushiki-Gitanjali	2020-05-04
296	RC-NS-GK	SDL-CSDL-BIL	P	TE	2	active	6\$2	301-317-303	Rujata-Nahid-Gitanjali	2020-05-04
297	SK	WN	L	TE	2	active	7\$0	205	Sneha	2020-05-04
298	RC-KU-GK	SDL-SNL-BIL	P	TE	2	active	7\$1	301-308-303	Rujata-Kaushiki-Gitanjali	2020-05-04
299	RC-NS-GK	SDL-CSDL-BIL	P	TE	2	active	7\$2	301-317-303	Rujata-Nahid-Gitanjali	2020-05-04
300	RC	SEPM	L	TE	2	active	7\$3	205	Rujata	2020-05-04
301	SK-SJ-AA	UL-MPL-PL	P	SE	2	active	0\$1	313-302-303	Sneha-Sonal-Anagha	2020-05-04
302	NN	AM	L	SE	2	active	0\$2	206	Nancy	2020-05-04
303	YP	OS	L	SE	2	active	0\$3	206	Yamini	2020-05-04
304	KU	AT	L	SE	2	active	0\$4	206	Kaushiki	2020-05-04
305	NN	AM	L	SE	2	active	1\$0	206	Nancy	2020-05-04
306	SK-SJ-AA	UL-MPL-PL	P	SE	2	active	1\$1	313-302-303	Sneha-Sonal-Anagha	2020-05-04

# **Chapter 6**

## **Testing**

For the testing purpose we opted to go for the functional testing methods. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using use cases provided by the design team or business analyst. Function testing includes :

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

### **Unit Testing :**

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing also makes debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.

Therefore, opting for the unit testing method in our project played a crucial role in assessing each module of the application separately. This testing method best suited our project as we had various modules at the start which were to be tested and verified. It made the testing process easier by helping us discover the minute errors in each module and therefore we could rectify them efficiently.

### **Integration Testing :**

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

Integration testing was a necessity to check whether each individual module/unit was working well in synchronisation with one another. There were multiple problems while integrating the various modules which were only discovered with the help of integration testing methodology.

### **System Testing :**

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production. Finally, the entire system was tested as a whole using the System testing Methodology.

Here, the functional requirements of our applications that include CSV reading, generation of time table using genetic algorithm, multiple views of time table, fetching previous saved time tables, downloading time table in xls format were checked altogether. Application was given to other professionals (faculty) who checked and verified the proper working of all the modules and the system as a whole.

### **Acceptance Testing :**

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users.

Acceptance Testing method was adopted for our application to get a final review of our application by the actual end users. The application was given to fellow peers to try hands on and get their opinions and recommendations which were worked upon. Also, it was seen that all end user needs were met to their complete satisfaction after using the application.

## Various Testcases :

Test Case Name	Test Condition	Test Steps/procedure	Test Data	Expected Result	Actual Result	Pass/Fail
home.jsp	View Screen	Null	View screen	System need to show the Home page to the user	Home page run on the user screen	Pass
uploadImage1.jsp	Upload CSV format file	If already prepared the CSV file with required format, upload it OR Download the sample file and make required changes in it and then upload it.	CSV File Uploaded	User should be able to download the sample file and upload the CSV file	Sample file was downloaded successfully and also CSV file with data was uploaded	Pass
details.jsp	Data verification	Verify the Data	View screen	System need to show the Data in the CSV file to the user	System Shows the Data of CSV file	Pass
timeTable.jsp	Time Table Generation	Null	View screen with generated Time table	System needs to show generated Timetable to user which fulfills all constraints	System Shows Generated Time Table with proper format to user	Pass
timeTable.jsp	Save Time Table	If user is ok with the generated Time table he can save it	Save Time Table to database with time and date so that it can be viewed any time	Time table visible on the screen should be properly saved in database	Time table saved to database.	Pass
timeTable.jsp	Multiple Views	User needs to select the view to be displayed	Selected View to be displayed	Latest generated and saved time table should provide multiple views to user such as classroom, faculty, semester	User can see the view selected in proper format	Pass
timeTable.jsp	Previous Time-Table	User needs to select the time table to be viewed from previous time-table tab	Selected timetable to be displayed(only master view)	System should provide master view of selected time table	System provides master view of selected date	Pass
timeTable.jsp	Download Time Table	Null	View displayed should be downloaded in xlsx format	System should allow user to user to download any view in xlsx format	User downloaded current view in xlsx format	Pass

Table 6.1: Testcases

# **Chapter 7**

## **Conclusions and Future Scope**

### **7.1 Conclusion**

Automatic Timetable Generator using Genetic Algorithm has made it possible to automate the process of generating the timetable taking into consideration all the basic constraints kept in mind while generating the timetable manually, it uses genetic algorithm steps to implement the process. The system is efficiently generating the timetable fulfilling all the requirements and that too in few minutes, which in manual process took hours for same output to be generated. To use the system there is a basic input format which needs to be followed to get the algorithm working, any place where the input format is followed, this system can be used. For example, it can be used by all institutes, schools, etc.

### **7.2 Future Scope**

1. Implementation of various constraints like Aptitude slots, PBL (Project Based Learning) activities, Mentoring slots (same for every semester), Departmental and Institutional electives.
2. If required working on security factor.
3. Assignment of slots for various activities for different initiatives taken by the institute.
4. Generation of special timetable i.e. one day timetable or occasional timetable.

# Bibliography

- [1] Asha, V. G., Babu, K. N. R. M. (2017). On-line help desk for college departmental activities 2017. International Conference on Intelligent Computing and Control Systems (ICICCS) IEEE (2017).  
<https://ieeexplore.ieee.org/document/8250600/>
- [2] Yang, X. F., Ayob, M., Nazri, M. Z. A. (2017). An investigation of timetable satisfaction factors for a practical university course timetabling problem. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI) IEEE (2017).  
<https://ieeexplore.ieee.org/document/8312409>
- [3] Ilham, N. I., Saat, E. H. M., Rahman, N. H. A., Rahman, F. Y. A. Kasuan, N. (2017). Autogenerate scheduling system based on expert system 2017. 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE).  
<https://ieeexplore.ieee.org/document/8284370>
- [4] Anuja Chowdhary et al, TIMETABLE GENERATION SYSTEM, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February-2014.  
<https://ijcsmc.com/docs/papers/February2014/V3I2201474.pdf>
- [5] Bong Chia Lih, Sze San Nah, Bolhassan, N. A. (2015). A study on heuristic timetabling method for faculty course timetable problem 2015 9th International Conference on IT in Asia (CITA) IEEE(2015).  
<https://ieeexplore.ieee.org/document/7349832>

# Appendices

## Appendix-A: Installation and Configuration of Eclipse

### Step 1 : Installing Java

Before you start installing Eclipse, make sure that you have the Java Development Kit (the JDK) installed on your system.

### Step 2 : Download

You can download Eclipse at [eclipse.org/downloads](http://eclipse.org/downloads). The latest version, as of time of writing, is Eclipse SimRel.

### Step 3: Installation

Run the Eclipse installer. You should see a window like the one below; Select the first **”Eclipse IDE for Java Developers”** option. After that point, you can keep hitting “yes” and select all the default options (unless you want to change something). You should eventually see a screen like this. Click the “Launch” button.

### Step 4: Configuration

I) When you run Eclipse, it’ll ask you where you want your workspace to be. Your workspace will be the location where Eclipse will add any new projects you create.

You can change the location of the workspace if you want: just make sure you remember what you picked.

II) Once you’re done, you should see a “Welcome” screen. Close the “welcome” tab to open the regular editor.

III) Next, select “Windows  $\downarrow$  Preferences” (PC) or “Eclipse  $\downarrow$  Preferences” (Mac) in the menu. Then, select “Java  $\downarrow$  Installed JREs”:

IV) Click the “Search” button and select the “Java” folder. This folder should contain your installed JRE and JDK. (If it contains only the installed JDK, that’s also ok). You can probably find this folder located at:

Windows: C:Files

Mac: /Library/Java

V) After hitting “ok”, you should see a screen with a line for either both the JRE and the JDK, or just the JDK. Select the line for the JDK:

VI) Click the “Apply and close” button.

## **Appendix-B: Installation and Configuration of Apache Tomcat**

JDK or JRE will need to be installed on the Windows Server before you can configure Tomcat 9 on the server.

### **Installing Tomcat 9**

**Step 1 :** Download Apache Tomcat from this link <http://tomcat.apache.org/>  
click Download -> Tomcat 8.0

Choose Binary Distributions Core: 32-bit Windows zip / 64-bit Windows zip.

**Step 2 :** Extract it to Document folder.

**Step 3 :** Open Eclipse Environment

Click on Servers Tab

Click on No servers are available. Click this link to create a new server...

Click Tomcat v8.0 Server and Next

**Step 4 :** Select Apache installation Directory and click Finish.

**Step 5 :** You should see Tomcat v8.0 Server at localhost [Stopped, Republish] under Servers tab.

**Step 6 :** Now select the Server and click Start.

Now it should be up and running on port 8080.

## **Appendix-C: Installation of MySQL**

**Step 1 :** Download the MySQL Installer from [dev.mysql.com](http://dev.mysql.com)

The two download options are a web-community version and a full version.

i) **web-community version:** It will only download the server, by default, but you can select other applications (like Workbench) as desired.

ii) **full installer:** It will download the server and all the recommended additional applications.

**Step 2 :** Run the installer that you downloaded from its location on your server, generally by double-clicking.

**Step 3 :** Determine which setup type you would like to use for the installation:

i) **Developer Default:** This is the full installation of MySQL Server and the other tools needed for development. If you are building your database from the ground up or will be managing the data directly in the database, you'll want to use this setup type.

ii) **Server Only:** If you only need MySQL Server installed for use with a CMS or other application and will not be managing the database directly, you can install just the server.

iii) **Custom:** This setup type will allow you to customize every part of the installation from the server version to whichever additional tools you select.

**Step 4 :** Install the server instance and whichever additional products you selected. Then begin the configuration process by selecting the availability level (most users will use the

default, standalone version).

**Step 5 :** Complete the configuration process by following the on-screen instructions. You'll want to make sure to install MySQL as a Service so that Windows can automatically start the service after a reboot or can restart the service if it fails.

## Configuration of MySQL

**Step 1 :** Goto Database Perspective

Window-> Perspective->Open Perspective-> Other

**Step 2 :** Select Database Development Perspective

select the Database Development perspective and click on the Open button.

**Step 3 :** Create Database Connections

**3.1 :** There you can see the Database Connections folder, right click and click on the New.

**3.2 :** Then you will see the New Connection Profile window as below; there you will find all available connections, filter the connection profile type with “MySQL” string and select the MySQL connection. Name this connection as MYSQLDBCONNECTION and click on Next button.

**3.3 :** This window allows you to attach the driver details to connect with the MySQL DB. Click on the below-highlighted icon to attach MySQL connector.

**3.4 :** Then you will see the following window; select the MySQL version which you wanted to connect with and move to JAR List tab.

Here we have to attach the MySQL connector.

**Step 4 :** Download and attach MySQL connector

**4.1 :** Download the MySQL connector from <https://downloads.mysql.com/archives/c-j/>

**4.2 :** It will be downloaded as a .zip file.

Extract the zip and there you will see the mysql-connector-java-xxx.jar.

Add this jar file in Jar List tab by clicking on the Add JAR button and click on the Ok button.

**4.3 :** Specify the MySQL connection details like username, password and database name and click on the Test Connection button.

**4.4 :** If everything configured well, you would see the below Ping succeeded! Message.

Then Click on the Finish button.

Done.

**Step 5 :** Verify Again go to Data Source Explorer and refresh on Database Connections folder and expand the folders inside it.

You would see the MySQL database schema here.

## Acknowledgement

We have great pleasure in presenting the report on **Automatic Timetable Generator using Genetic Algorithm**. We take this opportunity to express our sincere thanks towards our guide **Ms. Rujata Chaudhari** & Co-Guide **Ms. Nahid Shaikh** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Mr. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Ms. Apeksha Mohite** and **Ms. Anagha Aher** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

**Student Name1:** Harsh Bhanushali  
**Student ID1:** 16104043

**Student Name2:** Greshma Sapra  
**Student ID2:** 16104038

**Student Name3:** Surbhi Saroliya  
**Student ID3:** 16104037