KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science and Engineering

Project Report on

AUTOMATED TIME-TABLE SCHEDULER (USING GENETIC ALGORITHM)

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

Bachelor of Engineering

In

8th Semester

Submitted by

Akshata Chinchakhandi	2GI16CS018
Aparna M V	2GI16CS026
Bhagyashree Dudni	2GI16CS031
Gouri Benni	2GI16CS043

Guide **Co- Guide**

1.

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science and Engineering

CERTIFICATE

Certified that the project entitled Automated Time-Table Scheduler (Using Genetic Algorithm)carried out by Ms. Akshata Chinchankhandi USN-2GI16CS006 , Ms. Aparna M V USN-2GI16CS026,Ms. Bhagyashree Dudni USN-2GI16CS031, Ms. Gouri Benni USN-2GI16CS043 students of KLS Gogte Institute of Technology, Belagavi, can be considered as a bonafide work for partial fulfillment for the award of **Bachelor of Engineering** in COMPUTER SCIENCE AND ENGINEERINGof the Visvesvaraya Technological University, Belagavi during the year 2019 – 2020. It is certified that all corrections/suggestions indicated have been incorporated in the report. The project report has been approved as it satisfies the academic requirements prescribed for the said Degree.

Guide **Co-Guide** **HOD** **Principal**

Date:

Final Viva-Voce

	Name of the examiners	**Date of Viva -voce**	**Signature**
1.			
2.			

DECLARATION BY THE STUDENT(S)

I/We, *name of the student(s)*, hereby declare that the project report entitled Automated Time-Table Scheduling(Using Genetic Algorithm) submitted by me/us to KLS Gogte Institute of Technology, Belagavi, in partial fulfillment of the Degree of **Bachelor of Engineering** in Computer Science and Engineering is a record of the project carried out at 2019-2020. This report is for the academic purpose.

I/We further declare that the report has not been submitted and will not be submitted, either in part or full, to any other institution and University for the award of any diploma or degree.

Name of the student	USN	Signature
Akshata Chinchakhandi	2GI16CS018	
Aparna M V	2GI16CS026	
Bhagyashree Dudni	2GI16CS031	
Gouri Benni	2GI16CS043	

Place:Belagavi

Date:

ACKNOWLWDGEMENT

This feels greatly indebted to Computer Science and Engineering Department, for the opportunity given us to undertake this “Automated Time-Table Scheduling(Using Genetic Algorithm)” project. This project includes thoughts and contribution of many individuals. And we wish to express our sincere appreciation and gratitude to them.

First and foremost we want to extend entirely our gratitude to our guide prof. Jyoti Amboji for sharing her knowledge and profound wisdom with us. We appreciate all her comments and suggestions, which are incorporated into this project.

We would also like to express our gratitude towards colleagues and group members. Without their help, support, and encouragement, this project would never had been completed.

In our respect, this project is an outcome of the learning

Experience we have shared with our fellow students. We dedicate this project to all our fellow engineering students.

ABSTRACT

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 Automated time table scheduling. This project introduces a practical timetabling approach capable of taking care of both hard and soft constraints required specially for preparing time table in colleges with large number of students and limited resources like class-rooms or labs.

The automated time table scheduling provides easier ways for teachers and student to view their timetable once they are finalized over the application, having individual login id and passwords, and also some staff usually takes the previous year's timetable and modify it but still it is a tedious job to incorporate changes. To overcome all these problems we propose to make an automated system

Table of Contents

Chapter 1

1.Introduction:

Time Table Scheduling is an NP-hard problem and hence polynomial time verifiable using genetic algorithms. It a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year. In earlier days, time table scheduling was done manually with a single person or some group involved in task of scheduling it manually, which takes a lot of effort and time.

Planning timetables is one of the most complex and error-prone applications. Timetabling is the task of creating a timetable while satisfying some constraints. There are basically two types of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output is still valid, but hard constraints are those which if we violate them; the timetable is no longer valid. The search space of a timetabling problem is too vast, many solutions exist in the search space and few of them are not feasible. Feasible solutions here mean those which do not violate hard constraints and as well try to satisfy soft constraints.

We need to choose the most appropriate one from feasible solutions. Most appropriate ones here mean those which do not violate soft constraints to a greater extent. In this project hard-constraints have been taken care of strictly and it has been ensured that soft-constraints are as well followed as much as possible.

1.1.Problem Statement:

- Automated Time Table Scheduler (Using Genetic Algorithm) is an NP-hard problem and hence polynomial time verifiable using genetic algorithms. It a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year.The timetable must take inputs, and store them, the inputs include subjects, classes(the class indicates all the students present in in),teachers.

.

1.2 Objectives:

The main objective of this project is to develop an efficient and intelligent time table generation and maintenance system with powerful algorithms which can generate the Timetable by considering all the constraints and limitations. On satisfying all these constraints the algorithm will generate the best possible Timetable and can make universally accessible to faculty and students.

1.3.Scope of Poject:

1. The scope of the system includes

- Generation of Timetable.
- Access the available Timetable.
- Modify/Delete an existing Timetable.
- Add or modify faculty details of a department.
- Add or modify subject details of a department.
- Add or modify time scheduling and period allocation process.

2.Design and implementation Constraint:

Objects of Time Table Scheduler

□**Students Group**

The StudentGroup class has the ID, name of the student group, number of subjects, array of subject names and hours of study required for each subject per week. It also contains the id of teachers who will teach those subjects.

□**Teacher**

It is a class to hold the faculty information. It has an id, name of faculty, subject that he/she teaches and an in assigned which represents the no. of batches assigned to the teacher.

□**Slot**

A slot here is the most basic unit of Genetic algorithm. It represents a single characteristic of a Gene.

□**Time Table**

This class' object holds an array of Slot. This is basically a class to generate new slots initially for each Student group.

□**Gene**

It is the main constituent of a Chromosome and is made up of a sequence of slot numbers. It represents a Time table of a single class group

□**Chromosome**

A chromosome here is a collection or an array of Genes. It is the main class of algorithm and it undergoes crossover and mutation to furnish fitter individuals.

Automated Time Table Scheduler Using Genetic Algorithm

Pranav Khurana

□**Utility**

It is basically for testing purpose only. It contains some mehods like printSlots() which help to keep track of algorithm through console.

□**Inputdata**

This is a class mainly to fetch the input from user either through text file or

through form and provide it to the working classes of the algorithm.
□**SchedulerMain**
This is the main class of the algorithm which invokes other classes and calls methods for crossover, mutation , selection etc.

1.4 Methodology:

The methodology used in this application enables us to approach Evolutionary Algorithms. Evolutionary Algorithms are a class of direct, probabilistic search and optimisation algorithms gleaned from the model of organic evolution. A Genetic Algorithm (GA) is a type of EA and is regarded as being the most widely known EA in recent times. A GA differs from other search techniques in the following ways: GAs optimises the trade off between exploring new points in the search space and exploring the information discovered thus far.

GAs is randomised algorithms, in that they use operators whose results are governed by probability. The results for such operations are based on the value of a random number. This means GAs use probabilistic transition rules, not deterministic rules. GA was combined with a heuristic specific greedy algorithm to take advantage of the global search of feasible solutions and specific technique efficiency in local solution optimization. This approach resulted in considerably smaller execution times.

❑ **Soft-constraints (flexible):**

- More or less equal load is given to all faculties · Required time (hours per week) is given to every Batch

❑ **Hard-constraints (rigid):**

- There should not be any single instance of a faculty taking two classes simultaneously . A class group must not have more than one lectures at the same time

Chapter 2

2.1.Literature Survey:

The problem of timetable scheduling is described as a highly constrained NP-hard problem. It is known as the timetabling problem by most researchers. Planning timetable is one of the most complex and error prone task. A lot of complex constraints need to be addressed for development of an efficient algorithm to solve this problem. There are still serious problems like generation of high cost time table. Therefore there is a great requirement for an application distributing the course evenly and without collisions. Our aim here is to develop a simple, easily understandable, efficient and portable application, which could automatically generate good quality time tables within seconds.

Chapter 3

3.Requirement Specifications:

3.1.Interface Requirement:

<u>SOFTWARE REQUIREMENTS:-</u>

- Struts - The web framework used
- Bootstrap - Mobile first Frontend Framework
- JSP
- Servlets
- Ajax
- JavaScript
- HTML
- CSS
- MySQL (database)

<u>HARDWARE REQUIREMENTS:-</u>

1)Processor:	PENTIUM IV PROCCESSOR
2)Ram:	Min 4GB MB RAM
3)Hard Disk:	Min 80GB OF HARDDISK

3.2.Functional requirements:

- Give the inputs, to the software system using the frontend ,the input given must get stored into the database.
- The main algorithm must fetch the data that is stored in the database via ajax and Servlets.
- The algorithm is made using Struts-2 framework and java server pages, the main output is represented using the HTML and CSS pages.

3.3.Non-Functional Requirements

- The software must be able to perform error free reconciliations

Chapter 4

4.1.Architectural Diagram:

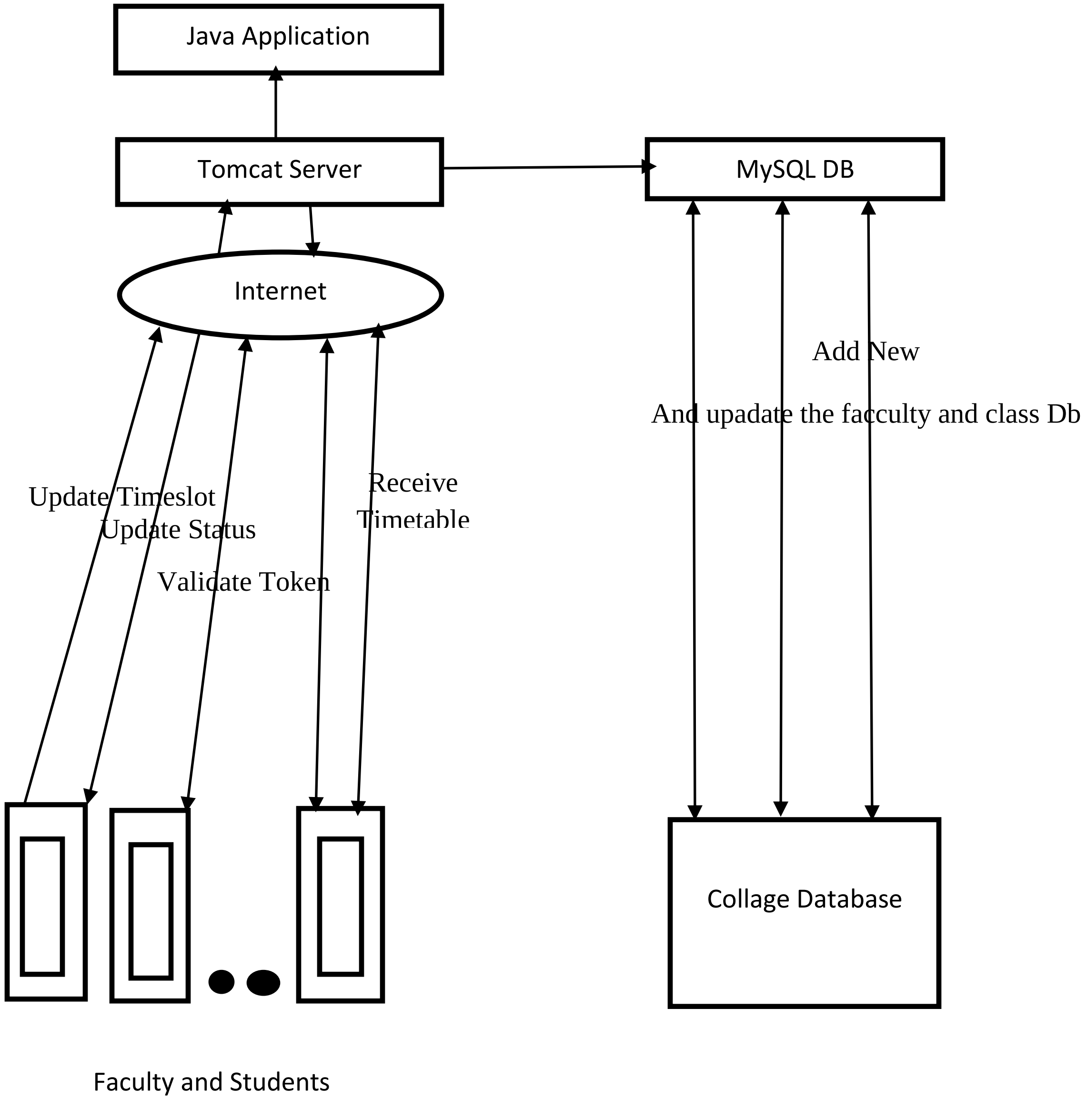


Fig:4.1 Architecture Design

4.2. Data Flow Diagram

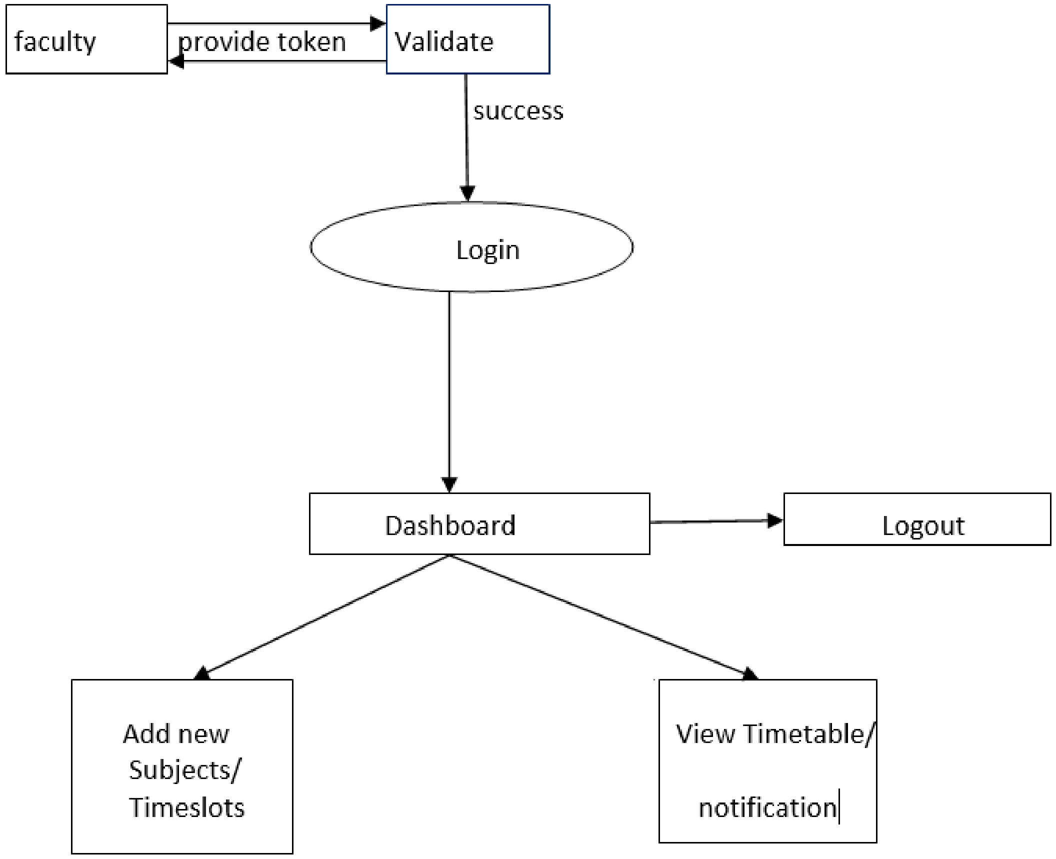


Fig 4.2 Data flow Diagram

4.3. Use case diagram for Faculty:

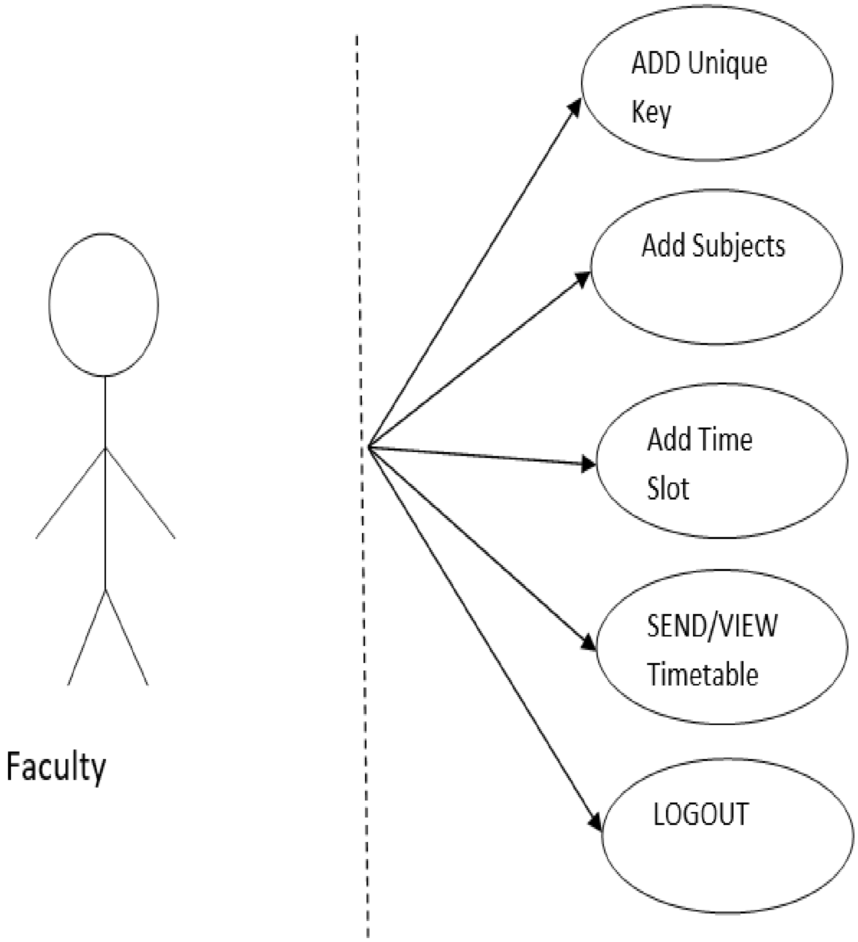
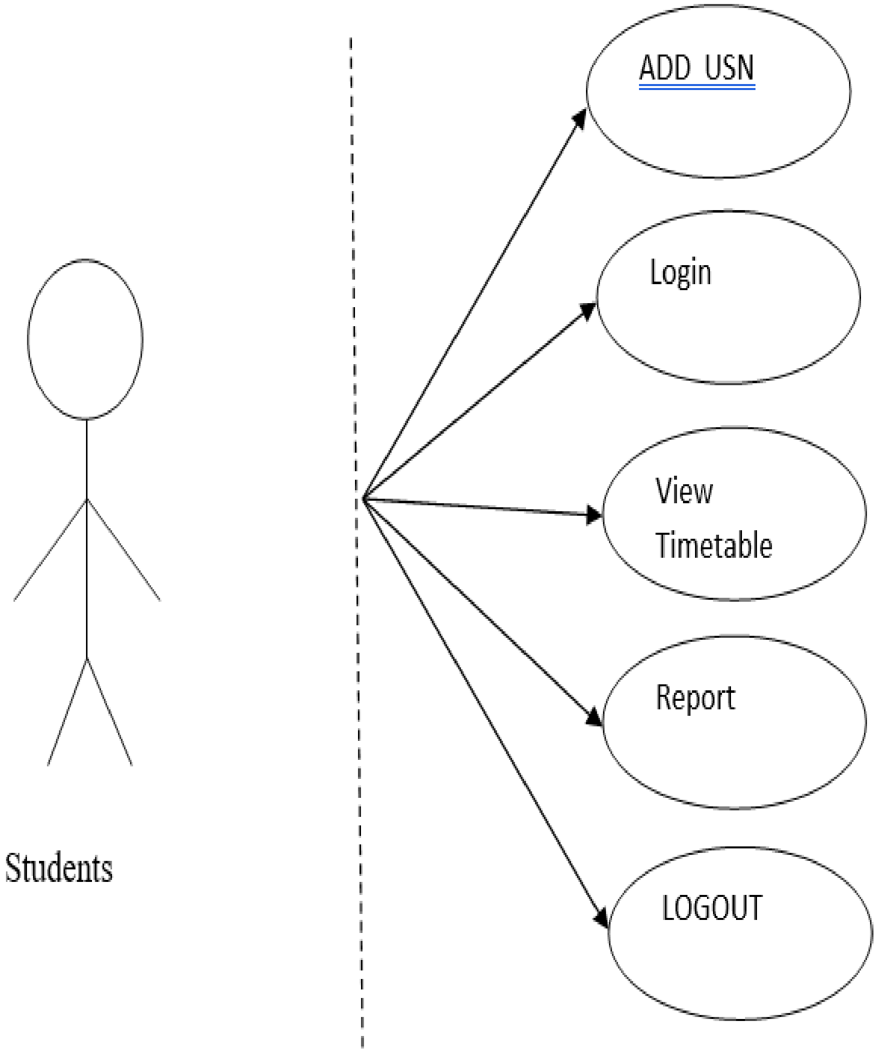


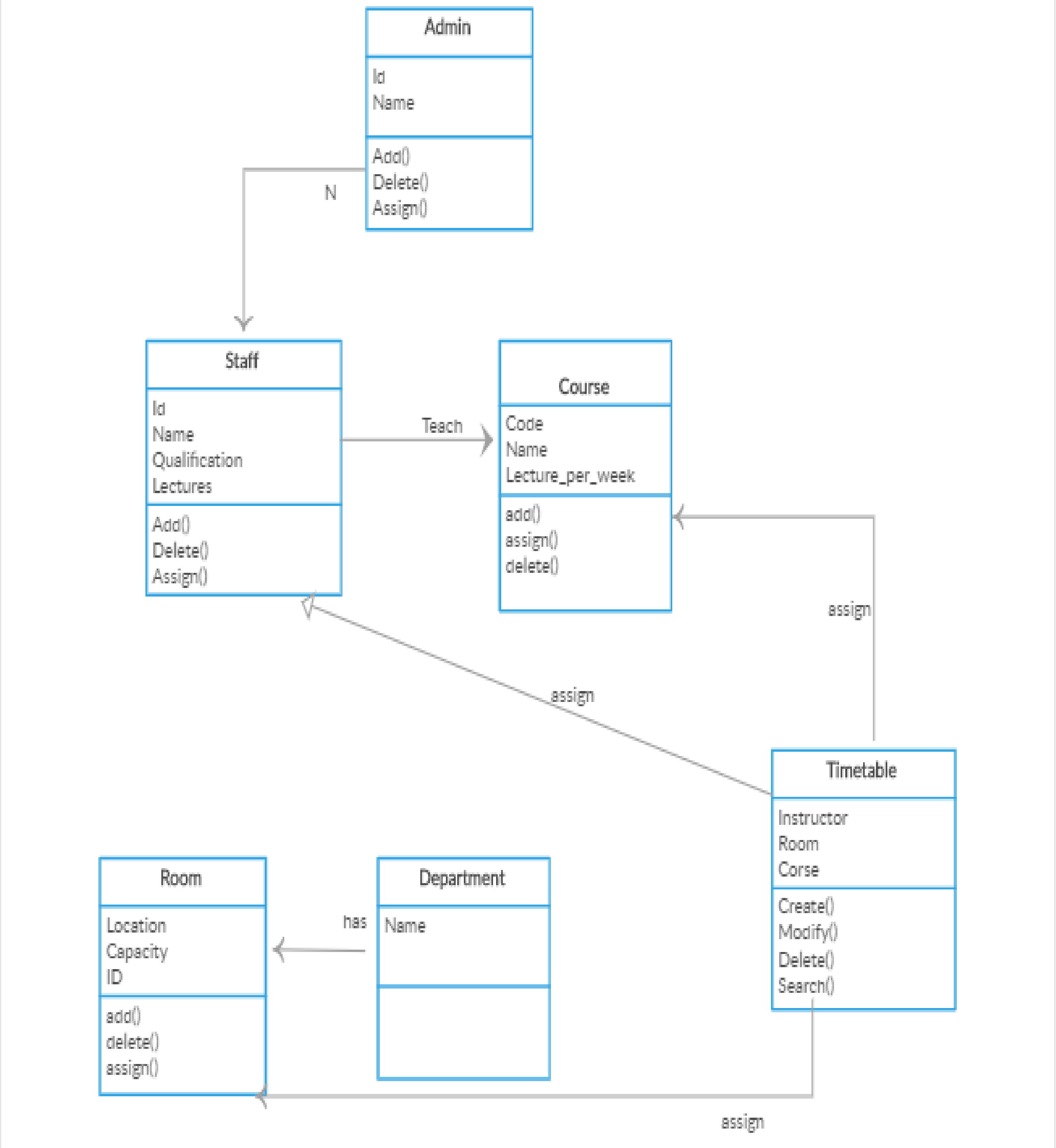
Fig 4.3 Use case Diagram for Faculty

4.4 Use case Diagram for students



Use case diagram for Students

4.5Class Diagram



Chapter 5

5.1.SQL Create Table

uuseruseruse

```
CREATE TABLE IF NOT EXISTS `users` (

  `id` int(4) NOT NULL,

  `username` varchar(50) NOT NULL,

  `password` varchar(50) NOT NULL,

  `country` varchar(50) NOT NULL,

  `email` varchar(50) NOT NULL) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT
CHARSET=latin1;
```

5.2.SQL Queries used

```
INSERT INTO `users` (`id`, `username`, `password`, `country`, `email`) VALUES

(1, 'firstname', 'lastname', 'india', 'firstname98@gmail.com');


ALTER TABLE `users`

  ADD PRIMARY KEY (`id`);


ALTER TABLE `users`

 MODIFY `id` int(4) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=2
```

5.3.Module Description:

- First of all an initial generation of chromosomes is created randomlyand their fitness value is analysed.

- New Generations are created after this. For each generation, itperforms following basic operations:

    a. First of all preserve few fittest chromosomes from the previousgeneration as it is. This is called Elitism and is necessary topreserve desired characteristics in the coming generations .

    b. Randomly select a pair of chromosomes from the previousgeneration. Roulette wheel selection method has been used herein this project.

    c. Perform crossover depending on the crossover rate which ispretty high usually. Here single point crossover has been used.

    d. Perform mutation on the more fit chromosome so obtaineddepending on the mutation rate which is kept pretty smallusually.

- Now analyze the fitness of the new generation of chromosomes andorder them according to fitness values.

- Repeat creating new generations unless chromosomes of desiredfitness value i.e. fitness=1, are obtained.

Database:

The admin will login to its database. Database will have all the details of the users.

5.4.Algorithm/Pseudo Code:

```
package scheduler;

import java.util.*;

publicclass SchedulerMain{

    /*
     * Time Table scheduling is an np-hard problem which can best be solved
     * using Genetic Algorithms (of Artificial Intelligence).
     * Conceps used here are Permutation encoding, elitism, roulette wheel selection,
     * single pt crossover,swap mutation
     */

    List<Chromosome> firstlist;
    List<Chromosome> newlist;
    double firstlistfitness;
    double newlistfitness;
    int populationsize=1000;
    int maxgenerations=100;

    publicstatic Chromosome finalson;

    public SchedulerMain() {

        //printing input data (on console for testing)
        Utility.printInputData();

        //generating slots
        new TimeTable();

        //printing slots (testing purpose only)
        Utility.printSlots();

        //initialising first generation of chromosomes and puting in first arraylist
        initialisePopulation();

        //generating newer generation of chromosomes using crossovers and mutation
        createNewGenerations();

    }


    //Creating new Generations using crossovers and mutations
    publicvoid createNewGenerations(){
```

```
Chromosome father=null;
Chromosome mother=null;
Chromosome son=null;

int nogenerations=0;

//looping max no of generations times or until suitable chromosome found
while(nogenerations<maxgenerations){

    newlist=new ArrayList<Chromosome>();
    newlistfitness=0;
    int i=0;

    //first 1/10 chromosomes added as it is- Elitism
    for(i=0;i<populationsize/10;i++){
        newlist.add(firstlist.get(i).deepClone());
        newlistfitness+=firstlist.get(i).getFitness();
    }

    //adding other members after performing crossover and mutation
    while(i<populationsize){

        father=selectParentRoulette();
        mother=selectParentRoulette();

        //crossover
        if(new Random().nextDouble()<inputdata.crossoverrate){
            son=crossover(father,mother);
        }else
            son=father;

        //mutation
        customMutation(son);


        if(son.fitness==1){
            System.out.println("Selected Chromosome is:-");
            son.printChromosome();
            break;
        }

        newlist.add(son);
        newlistfitness+=son.getFitness();
        i++;
```

```
            }

            //if chromosome with fitness 1 found
            if(i<populationsize){


    System.out.println("********************************************************
*********************************");
                System.out.println("\n\nSuitable Timetable has been generated in
the "+i+"th Chromosome of "+(nogenerations+2)+" generation with fitness 1.");
                System.out.println("\nGenerated Timetable is:");
                son.printTimeTable();
                finalson=son;
                break;

            }

            //if chromosome with required fitness not found in this generation
            firstlist=newlist;
            Collections.sort(newlist);Collections.sort(firstlist);
            System.out.println("***************************
Generation"+(nogenerations+2)+"     *********************************************\n");
            printGeneration(newlist);
            nogenerations++;

        }
    }

    //selecting using Roulette Wheel Selection only from the best 10% chromosomes
    public Chromosome selectParentRoulette(){

        firstlistfitness/=10;
        double randomdouble=new Random().nextDouble()*firstlistfitness;
        double currentsum=0;
        int i=0;

        while(currentsum<=randomdouble){
            currentsum+=firstlist.get(i++).getFitness();
        }
        return firstlist.get(--i).deepClone();

    }


    //custom mutation
    publicvoid customMutation(Chromosome c){
```

```
    double newfitness=0,oldfitness=c.getFitness();
    int geneno=new Random().nextInt(inputdata.nostudentgroup);

    int i=0;
    while(newfitness<oldfitness){

        //c.printChromosome();
        //System.out.println("getf="+c.getFitness()+" fit= "+c.fitness);

        c.gene[geneno]=new Gene(geneno);
        newfitness=c.getFitness();

        //c.printChromosome();
        //System.out.println("getf="+c.getFitness()+" fit= "+c.fitness);

        i++;
        if(i>=500000) break;
    }

}


//Two point crossover
public Chromosome crossover(Chromosome father,Chromosome mother){

    int randomint=new Random().nextInt(inputdata.nostudentgroup);
    Gene temp=father.gene[randomint].deepClone();
    father.gene[randomint]=mother.gene[randomint].deepClone();
    mother.gene[randomint]=temp;
    if(father.getFitness()>mother.getFitness())return father;
    elsereturn mother;

}

//initialising first generation of population
publicvoid initialisePopulation(){

    //generating first generation of chromosomes and keeping them in an arraylist
    firstlist=new ArrayList<Chromosome>();
    firstlistfitness=0;

    for(int i=0;i<populationsize;i++){

        Chromosome c;
        firstlist.add(c=new Chromosome());
```

```
                firstlistfitness+=c.fitness;

            }
            Collections.sort(firstlist);
            System.out.println("----------Initial Generation-----------\n");
            printGeneration(firstlist);

    }


    //printing important details of a generation
    publicvoid printGeneration(List<Chromosome> list){

            System.out.println("Fetching details from this generation...\n");

            //to print only initial 4 chromosomes of sorted list
            for(int i=0;i<4;i++){
                System.out.println("Chromosome no."+i+": "+list.get(i).getFitness());
                list.get(i).printChromosome();
                System.out.println("");
            }

            System.out.println("Chromosome no. "+(populationsize/10+1)+"
:"+list.get(populationsize/10+1).getFitness()+"\n");
            System.out.println("Chromosome no. "+(populationsize/5+1)+"
:"+list.get(populationsize/5+1).getFitness()+"\n");
            System.out.println("Most fit chromosome from this generation has fitness =
"+list.get(0).getFitness()+"\n");

    }


    //selecting from best chromosomes only(alternate to roulette wheel selection)
    public Chromosome selectParentBest(List<Chromosome> list){

            Random r=new Random();
            int randomint=r.nextInt(100);
            return list.get(randomint).deepClone();

    }


    //simple Mutation operation
    publicvoid mutation(Chromosome c){
            int geneno=new Random().nextInt(inputdata.nostudentgroup);
            int temp=c.gene[geneno].slotno[0];
```

```
                for(int i=0;i<inputdata.daysperweek*inputdata.hoursperday-1;i++){
                        c.gene[geneno].slotno[i]=c.gene[geneno].slotno[i+1];
                }
                c.gene[geneno].slotno[inputdata.daysperweek*inputdata.hoursperday-1]=temp;
        }


        //swap mutation
        publicvoid swapMutation(Chromosome c){

                int geneno=new Random().nextInt(inputdata.nostudentgroup);
                int slotno1=new
Random().nextInt(inputdata.hoursperday*inputdata.daysperweek);
                int slotno2=new
Random().nextInt(inputdata.hoursperday*inputdata.daysperweek);

                int temp=c.gene[geneno].slotno[slotno1];
                c.gene[geneno].slotno[slotno1]=c.gene[geneno].slotno[slotno2];
                c.gene[geneno].slotno[slotno2]=temp;
        }




        publicstaticvoidmain(String[] args) {
                inputdata id = new inputdata(); id.takeinput();
                new SchedulerMain();
        }
}
```

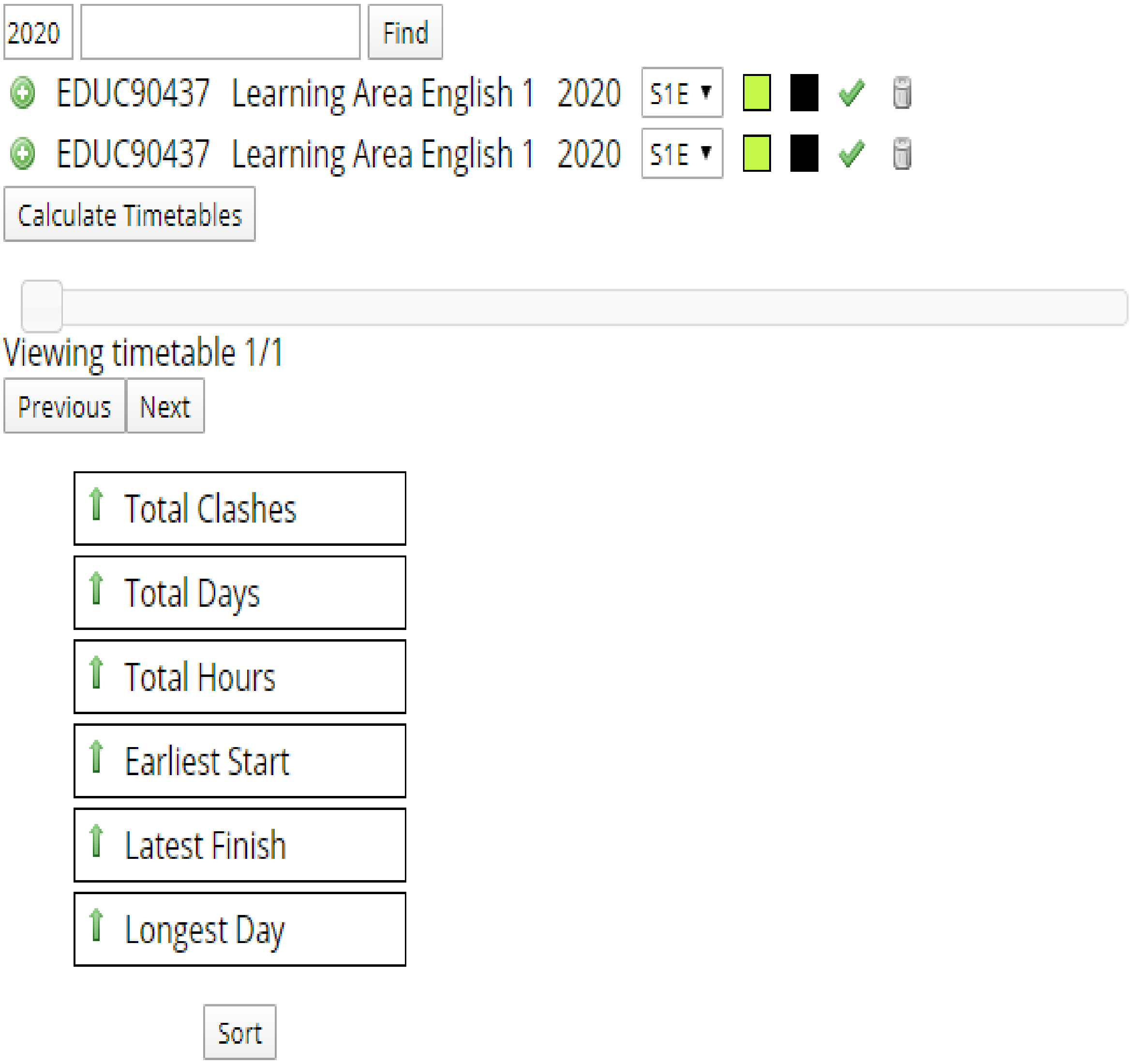
Chapter 6

6.1.Testing

For the ease of testing and tracking, a lot of information is printed on the console itself. It involves input information, slots generated,few chromosomes from each generation of chromosome, fitness ofthese chromosomes, maximum fitness in a generation and final selected chromosome.

Chapter 7

7.1 Results and Discussion:



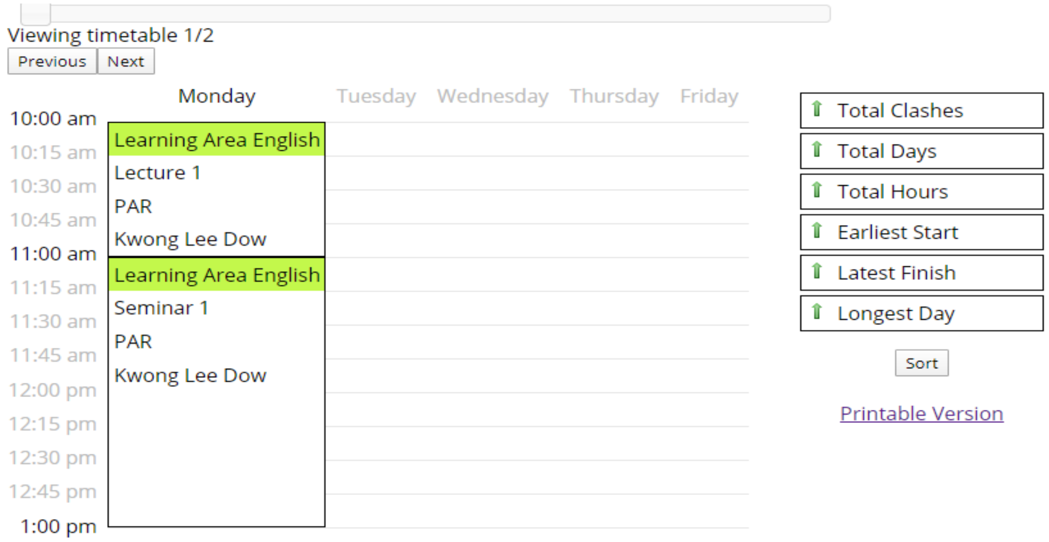
2020

EDUC90437 Learning Area English 1 2020 S1E

EDUC90437 Learning Area English 1 2020 S1E

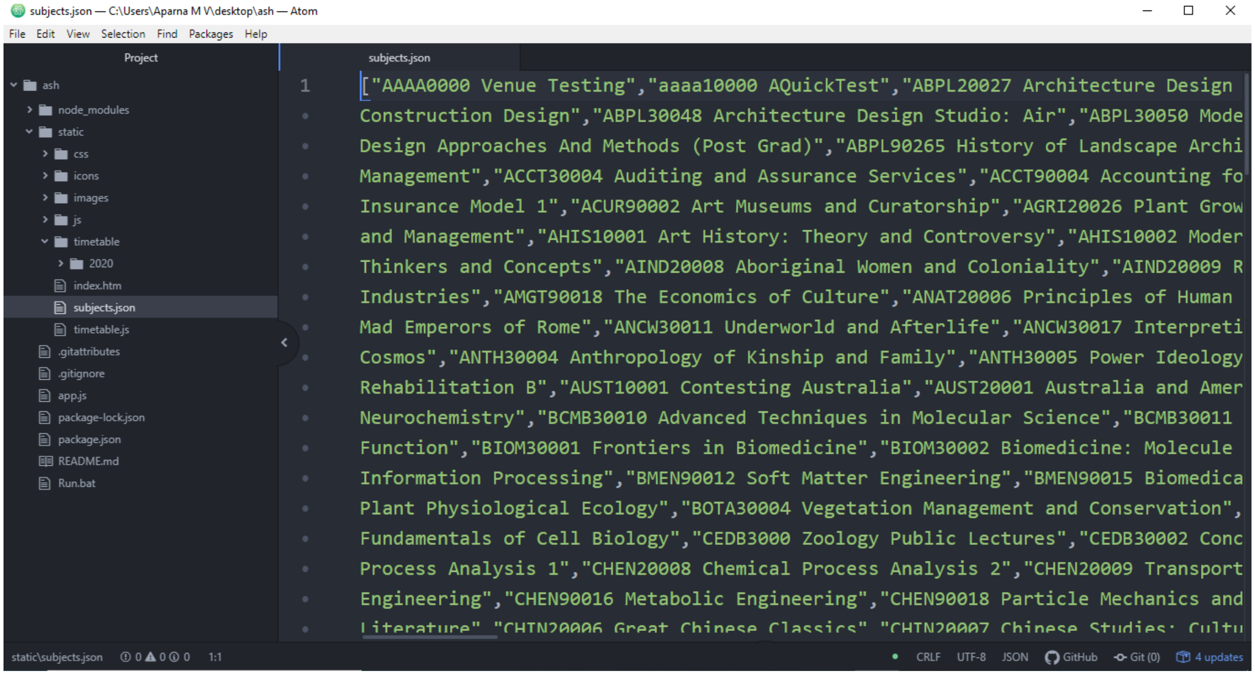
Viewing timetable 1/1

Fig 6.1 Start page



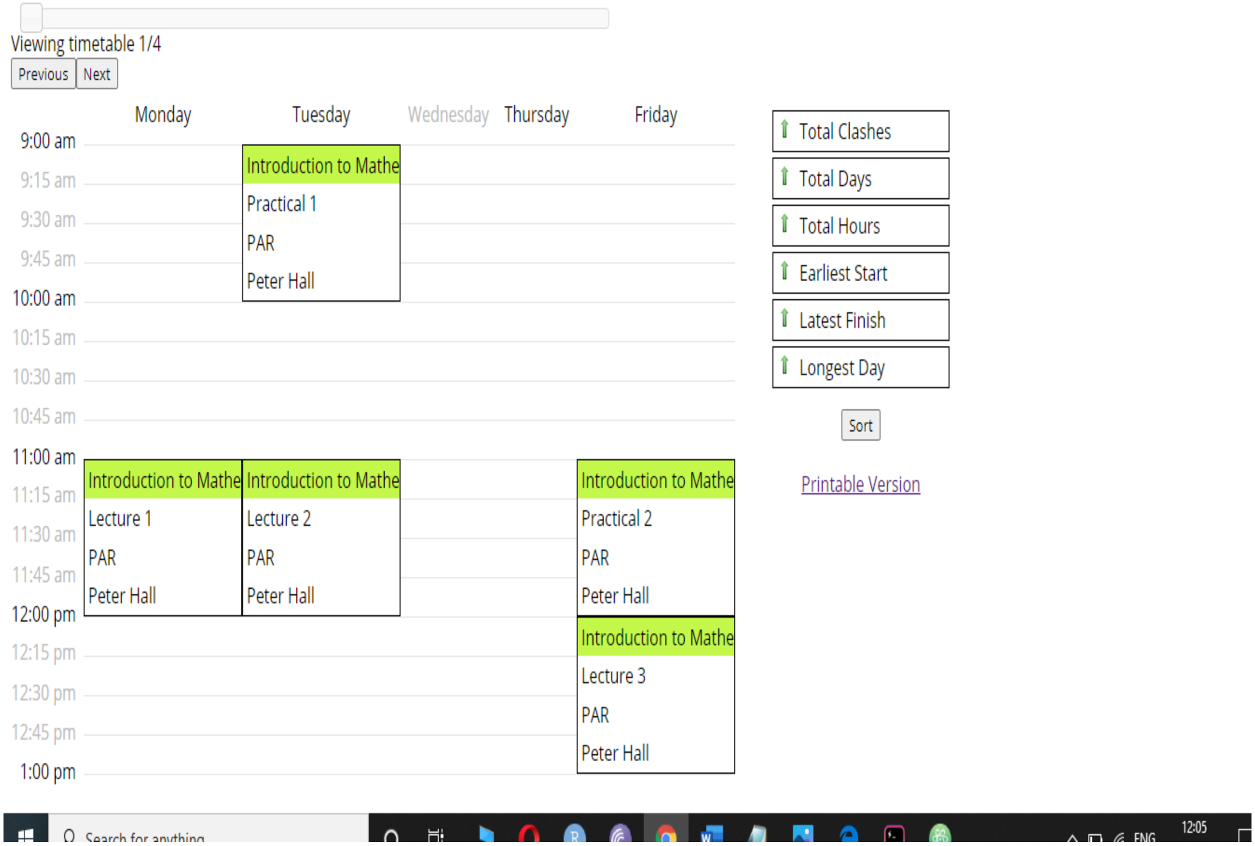
	Monday	Tuesday	Wednesday	Thursday	Friday
10:00 am					
10:15 am	Learning Area English				
10:30 am	Lecture 1				
10:45 am	PAR				
11:00 am	Kwong Lee Dow				
11:15 am	Learning Area English				
11:30 am	Seminar 1				
11:45 am	PAR				
12:00 pm	Kwong Lee Dow				
12:15 pm					
12:30 pm					
12:45 pm					
1:00 pm					

Fig 6.2. Inserting Into the table



```
["AAAA0000 Venue Testing","aaaa10000 AQuickTest","ABPL20027 Architecture Design
Construction Design","ABPL30048 Architecture Design Studio: Air","ABPL30050 Mode
Design Approaches And Methods (Post Grad)","ABPL90265 History of Landscape Archi
Management","ACCT30004 Auditing and Assurance Services","ACCT90004 Accounting fo
Insurance Model 1","ACUR90002 Art Museums and Curatorship","AGRI20026 Plant Grow
and Management","AHIS10001 Art History: Theory and Controversy","AHIS10002 Moder
Thinkers and Concepts","AIND20008 Aboriginal Women and Coloniality","AIND20009 R
Industries","AMGT90018 The Economics of Culture","ANAT20006 Principles of Human
Mad Emperors of Rome","ANCW30011 Underworld and Afterlife","ANCW30017 Interpreti
Cosmos","ANTH30004 Anthropology of Kinship and Family","ANTH30005 Power Ideology
Rehabilitation B","AUST10001 Contesting Australia","AUST20001 Australia and Amer
Neurochemistry","BCMB30010 Advanced Techniques in Molecular Science","BCMB30011
Function","BIOM30001 Frontiers in Biomedicine","BIOM30002 Biomedicine: Molecule
Information Processing","BMEN90012 Soft Matter Engineering","BMEN90015 Biomedica
Plant Physiological Ecology","BOTA30004 Vegetation Management and Conservation",
Fundamentals of Cell Biology","CEDB3000 Zoology Public Lectures","CEDB30002 Conc
Process Analysis 1","CHEN20008 Chemical Process Analysis 2","CHEN20009 Transport
Engineering","CHEN90016 Metabolic Engineering","CHEN90018 Particle Mechanics and
Literature","CHIN20006 Great Chinese Classics","CHIN20007 Chinese Studies: Cultu
```

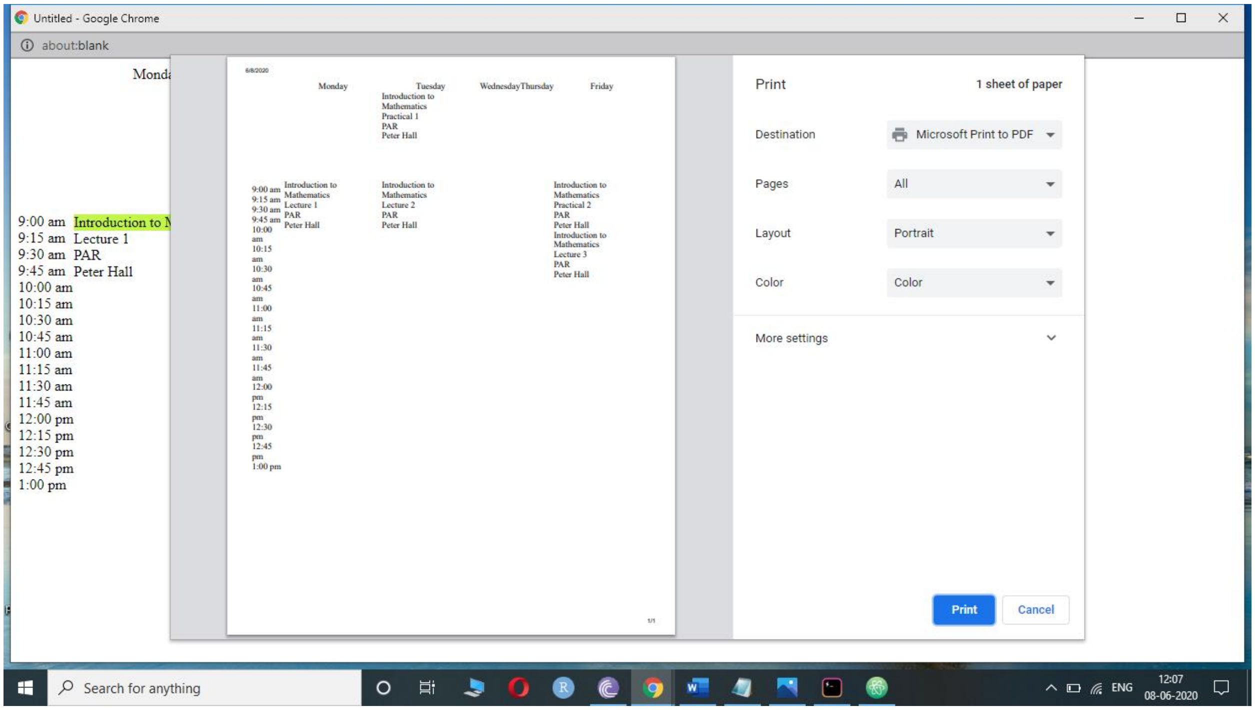
Subjects as JSON objects



Viewing timetable 1/4

	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 am		Introduction to Mathe Practical 1 PAR Peter Hall			
9:15 am					
9:30 am					
9:45 am					
10:00 am					
10:15 am					
10:30 am					
10:45 am					
11:00 am	Introduction to Mathe Lecture 1 PAR Peter Hall	Introduction to Mathe Lecture 2 PAR Peter Hall			Introduction to Mathe Practical 2 PAR Peter Hall
11:15 am					
11:30 am					
11:45 am					
12:00 pm					Introduction to Mathe Lecture 3 PAR Peter Hall
12:15 pm					
12:30 pm					
12:45 pm					
1:00 pm					

Fig 6.3 Review report



6/8/2020

	Monday	Tuesday	Wednesday	Thursday	Friday
		Introduction to Mathematics Practical 1 PAR Peter Hall			
9:00 am 9:15 am 9:30 am 9:45 am	Introduction to Mathematics Lecture 1 PAR Peter Hall	Introduction to Mathematics Lecture 2 PAR Peter Hall			Introduction to Mathematics Practical 2 PAR Peter Hall Introduction to Mathematics Lecture 3 PAR Peter Hall
10:00 am					
10:15 am					
10:30 am					
10:45 am					
11:00 am					
11:15 am					
11:30 am					
11:45 am					
12:00 pm					
12:15 pm					
12:30 pm					
12:45 pm					
1:00 pm					

Print — 1 sheet of paper

Destination: Microsoft Print to PDF

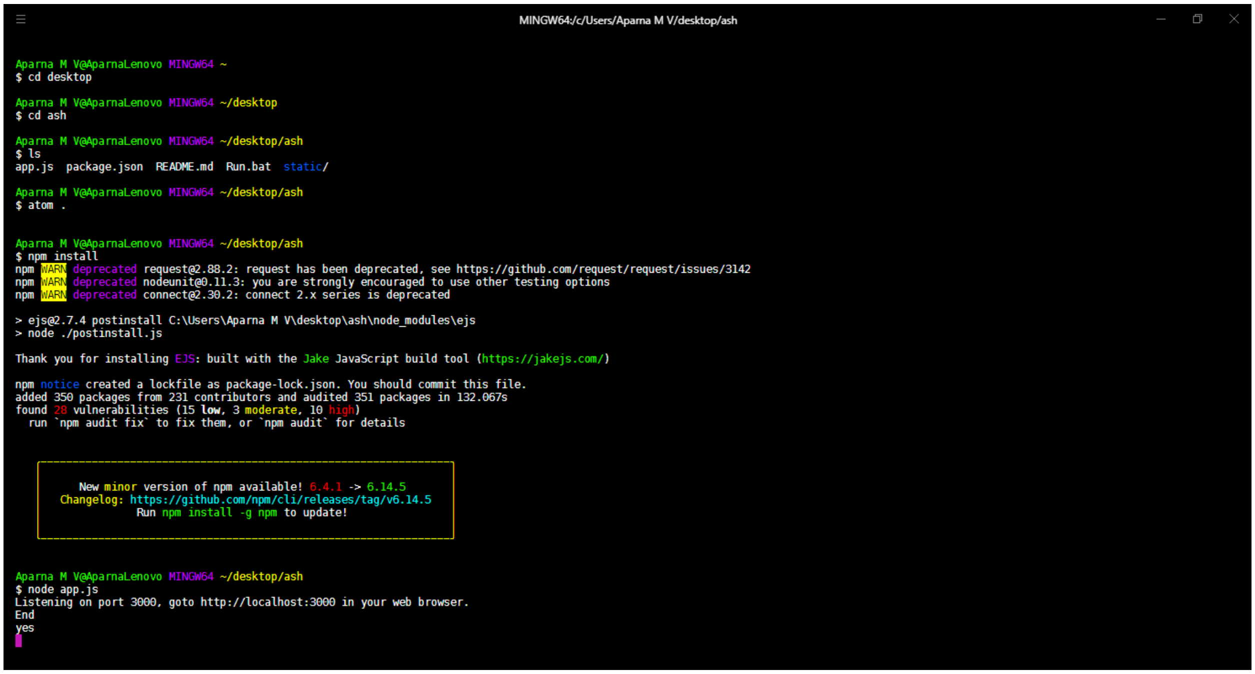
Pages: All

Layout: Portrait

Color: Color

More settings

Printable Timetable



```
Aparna M V@AparnaLenovo MINGW64 ~
$ cd desktop

Aparna M V@AparnaLenovo MINGW64 ~/desktop
$ cd ash

Aparna M V@AparnaLenovo MINGW64 ~/desktop/ash
$ ls
app.js  package.json  README.md  Run.bat  static/

Aparna M V@AparnaLenovo MINGW64 ~/desktop/ash
$ atom .


Aparna M V@AparnaLenovo MINGW64 ~/desktop/ash
$ npm install
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated nodeunit@0.11.3: you are strongly encouraged to use other testing options
npm WARN deprecated connect@2.30.2: connect 2.x series is deprecated

> ejs@2.7.4 postinstall C:\Users\Aparna M V\desktop\ash\node_modules\ejs
> node ./postinstall.js

Thank you for installing EJS: built with the Jake JavaScript build tool (https://jakejs.com/)

npm notice created a lockfile as package-lock.json. You should commit this file.
added 350 packages from 231 contributors and audited 351 packages in 132.067s
found 28 vulnerabilities (15 low, 3 moderate, 10 high)
  run `npm audit fix` to fix them, or `npm audit` for details


         New minor version of npm available! 6.4.1 -> 6.14.5
     Changelog: https://github.com/npm/cli/releases/tag/v6.14.5
                   Run npm install -g npm to update!



Aparna M V@AparnaLenovo MINGW64 ~/desktop/ash
$ node app.js
Listening on port 3000, goto http://localhost:3000 in your web browser.
End
yes
```

Fig 6.4 Opening Commands

Conclusion and Future Scope:

The process of Time Table generation has been fully automated with this software. This web app can now cater to multiple colleges, universities and schools which can rely on it for their Time Table scheduling which earlier had to be done by hand. Using Genetics Algorithm, a number of trade-off solutions, in terms of multiple objectives of the problem, could be obtained very easily. Moreover, each of the obtained solutions has been found much better than a manually prepared solution which is in use

Chapter 9

9.1. References

[1]http://www.javatpoint.com

[2] Automatic Timetable Generation using Genetic Algorithm-InternationalJournal of Advanced Research in Computer and CommunicationEngineering Vol. 4, Issue 2, February 2015.

[3] https://en.wikipedia.org/wiki/Genetic_algorithm

[4] http://www.obitko.com/tutorials/genetic-algorithms/encoding.php

Books:

1.Artificial Intelligence by Stuart J. Russell and Peter Norvig

2.Genetic Algorithms by David E. Goldberg

APPENDIX

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.Bootstrap can be boiled down to three main files:

bootstrap.css – a CSS framework

bootstrap.js – a JavaScript/jQuery framework

glyphicons – a font (an icon font set)

HTML is written in the form of HTML elements consisting of *tags*, enclosed in angle brackets (like <html>), within the web page content. HTML tags normally come in pairs like <h1> and </h1>. The first tag in a pair is the *start tag*, the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). In between these tags web designers can add text, tables, images, etc.The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

PHP is a general-purpose scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. It also has evolved to include a command-line interface capability and can be used in standalonegraphical applications. PHP can be deployed on most web servers and as a standalone interpreter, on almost every operating system and platform free of charge. PHP is installed on more than 20 million websites and 1 million web servers

JSPJavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offers several advantages in comparison with the CGI.

- Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having separate CGI files.
- JSP are always compiled before they are processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
- JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP, etc.

- JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere.