SSN COLLEGE OF ENGINEERING, KALAVAKKAM, CHENNAI

UCS2201

FUNDAMENTALS AND PRACTICE OF SOFTWARE DEVELOPMENT

PROJECT REPORT

END SEMESTER REVIEW

EXAM SCHEDULING SYSTEM

Team Name

V C Sharper

Team Members (CSEA)

Karthik Vijayakumar (040)

Keerthan V S (043)

Mohith A (053)

Team Guide

Dr. S. Saraswathi

Associate Professor, Department of Computer Science and Engineering

Abstract:

The procedure for creating exam schedules using the provided data and within the specified constraints is covered by the software that was developed. For the given problem statement, the Brute Force method is applied. It requires input for the invigilators, hall information, and course details. These are used to determine the dates, hall assignments, and invigilator assignments for each exam. The goal of the exam scheduling problem is to create timetables for an examination. This exam schedule details which exam is given on each day of the week, as well as its specifics, including the course information, year, and semester information, and whether it is a theory, practical, or arrears exam.

Introduction:

Problem Analysis:

The timetabling problem is characterized in numerous areas and is difficult to solve since it comprises various restrictions that must be resolved. Exam scheduling is one form of problem that is difficult to resolve since each institution may incorporate hard and soft limitations that are unique to their regulations. The challenges like students being dissatisfied if their timetable does not allow them to study enough between two consecutive tests, vast number of tests to be planned, vast number of students who have completed different courses, the restricted number of rooms, no clash in a single student's exams, make experimental scheduling extremely challenging. This project addresses and constructs the scheduling problem and conducts experiments to create the timetable.

The project analyses the input file and creates a suitable exam schedule for everyone, ensuring that no exams overlap. The timetabling problem manifests itself in a variety of ways and is difficult to solve because of the multiple constraints that must be overcome. Exam scheduling is one type of difficulty that is tough to handle since each institution's standards may include both hard and soft constraints. Students will be dissatisfied if their timetable does not allow them to study enough between two consecutive tests, there will be many tests to plan, there will be many students who have completed different courses, there will be a limited number of rooms, and there will be no clash in a single student's exams, all these factors make experimental scheduling extremely difficult. This project addresses and constructs the scheduling problem and conducts experiments to create the timetable.

The exam scheduling problem deals with the assignment of a set of events to a specific timeslot and room within a working week subject to a variety of hard and soft constraints. Hard constraints should not be violated under any circumstances, and we call a timetable that satisfies all such constraints as a feasible solution:

- No student is assigned to more than one examination at the same time
- The room should satisfy the features required by the examination
- The number of students attending the examination is less than or equal to the capacity of the room

- No more than one examination is allowed at a timeslot in each room
- There is no exam scheduled on Sunday or University declared holidays
- The time slots are scheduled within the working hours of the Institution.

The soft constraints on the other hand, are treated as goals to be reached, where the overall objective is to get as close as possible to these goals. The following are the soft constraints:

- A student has no consecutive examinations.
- There is a gap of at least one day between two exams of a particular stream.
- A student to invigilator ratio of 25:1 is maintained while allocating invigilators to halls.
- The halls and invigilators are allocated such that the number of halls and invigilators required are minimum.
- Practical examinations are grouped together and are not scheduled simultaneously with theory examinations.
- An invigilator is not assigned more than once per day.
- Arrear examinations are scheduled separately along with theory examinations.
- Examinations are scheduled such that a student can appear for arrear examinations without compensating for his current semester examinations.
- Common subjects for all departments and for some departments are taken into consideration for scheduling.

Existing Work:

1)Examination Scheduling using Graph coloring by Emre Kumas

This C project was developed to schedule exams (in universities for example). It takes a text input file which contains name and lessons for every single person.
The program analyses the input file and determine an appropriate exam scheduling so for every person no exam will be overlapping. A common problem in universities since no person can enter two exams at the same time.
It uses graph coloring algorithm to solve the problem.

Key Takeaways:

- Modular approach to solve the scheduling problem.
- Use of files to get the pre input and other details about examinations.

2) **Examination Scheduling using Genetic Algorithm**

The project developed in the National University of Computer and Emerging Sciences; Lahore was to find a generic solution that will facilitate generating schedule for university using

“Genetic Algorithm”. The success of solution in this project was estimated based on the fulfillment of given constraints and criteria.

Key Takeaways:

- Extraction of course details from the subject code.
- Implementation of hall-wise and invigilator-wise timetable.

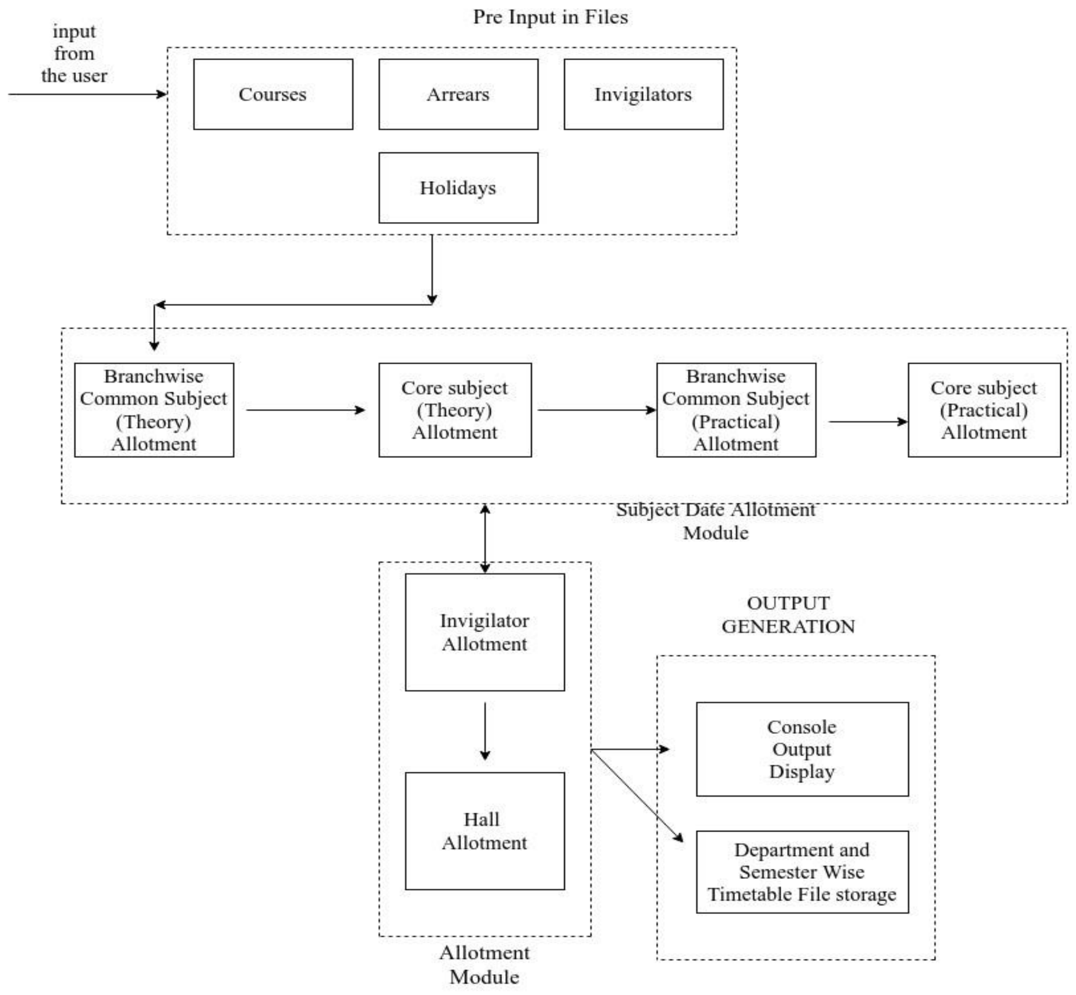
3) Modular exam scheduling problem with brute force algorithms developed by Dominik Czarnota

The software discusses about solving test scheduling challenges for huge datasets. The problem of modular test scheduling uses brute force algorithms. Problems with modular test scheduling in MESP are expected that exams must be dispersed into timeslots for the specified terms to be met. That means that no student will have two tests at the same time, and it is improbable that a student would take two exams one after the other. The number of limitations varies according to the university, the number of courses, tests, and rooms.

Key Takeaways:

- Limitations of brute force algorithm in generating an efficient timetable
- Creation of data structures for the allocation of resources like Invigilators and Halls.

Architectural Diagram:


Pre Input in Files
input from the user
Courses
Arrears
Invigilators
Holidays
Branchwise Common Subject (Theory) Allotment
Core subject (Theory) Allotment
Branchwise Common Subject (Practical) Allotment
Core subject (Practical) Allotment
Subject Date Allotment Module
Invigilator Allotment
Hall Allotment
Allotment Module
OUTPUT GENERATION
Console Output Display
Department and Semester Wise Timetable File storage

Module Design:

1) Header files (Pre-processor directives) and macros

This includes all header files, pre-processor directives and macors required for the program

2) Read courses module

The ‘Read courses module’ is composed of the ‘Read courses function’ and the ‘write courses function’.

The **‘Read courses function’** reads List of Courses, its corresponding course codes, List of Arrear courses and its corresponding course codes and List of invigilators from the pre-input given in the form of files for the corresponding operation.

The **‘Write courses function’** creates a separate file for each Department and Semester (Whether it is a main exam or an arrear exam) to write its corresponding contents to them after the process of scheduling is completed.

3) Course allocation module

This module primarily consists of three functions namely:

i) **Common subject allotment:** This function is used to schedule courses which are common to all the existing departments in a semester. These exams are scheduled at the start.

It takes a structure containing all the details for a particular course, semester, year, day of the week, course count, current date, course type as input and Returns a DATE on which the exam can be scheduled.

ii) **Common subject allotment across 2 Departments:** This function is used to schedule courses which are common only to two or more departments, but not all of them. These exams are scheduled after the scheduling of common subjects.

It takes a structure containing all the details for a particular course, semester, year, day of the week, the two departments which the course is common to, course count, current date, course type as input and Returns a DATE on which the exam can be scheduled.

iii) **Core subject allotment:** This function is used to schedule the courses which are unique to a particular department. These exams are generally scheduled at the end of the timetable.

It takes a structure containing all the details for a particular course, semester, year, day of the week, department, course count, current date, course type as input and Returns a DATE on which the exam can be scheduled.

4) Holiday check module

This module is used to read the pre-input Sundays, University Holidays and Public holidays from a file and makes sure that no exam is scheduled on that day.

5) Hall and invigilator allotment module

This module is used to read the pre-input of the list of exam halls and invigilators, given in the form of individual files and allot the exam halls and invigilators to a particular examination (which takes the department, course code, course name, semester into account) on a certain DATE and DAY.

This is the most important module in the whole scheduling process as it makes sure that:

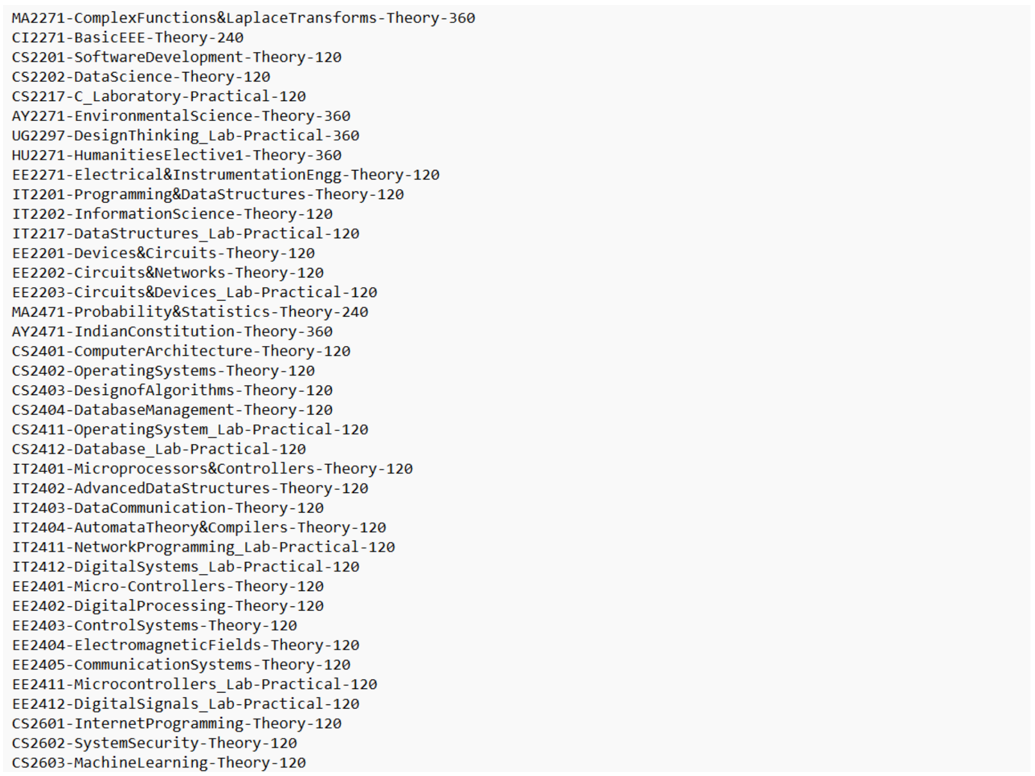
i) No invigilator is allotted multiple exam halls at the same time and ensures that an invigilator is allotted only one slot per day

ii) Each exam is scheduled in a separate room with the Ratio of the Invigiltor:Student being 1:25 and that multiple exams are not scheduled in the same exam hall in the same time slot.

6) Print Exam timetable module

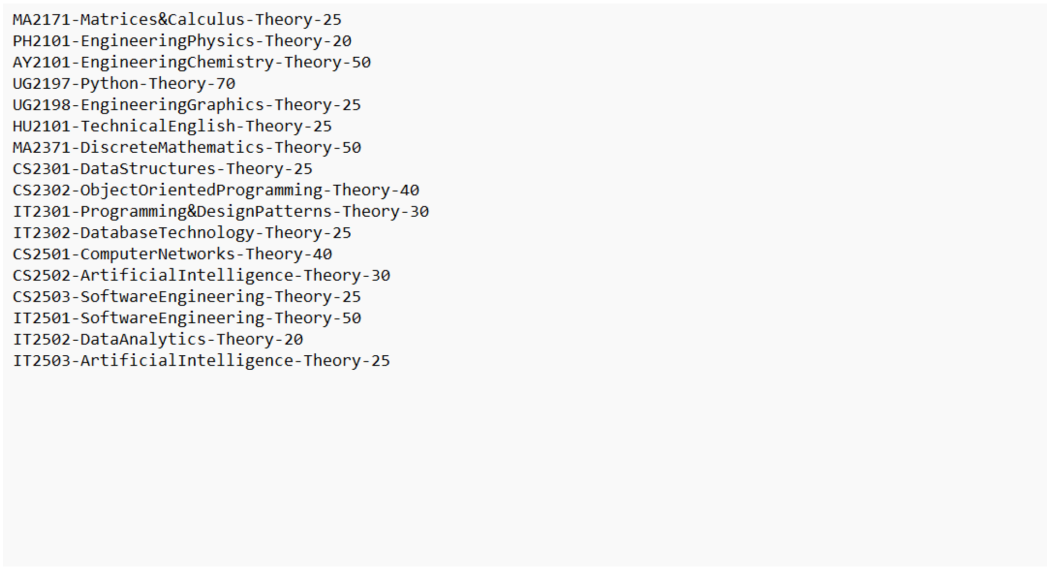
This module is used to print the exam timetable on the console as well as write the same to a file for further use, which is done for a particular semester (including arrears) and sorted by date. A hall wise timetable is also generated along with the list of invigilators on duty during the exam and it is duly written to the corresponding files.

Implementation and Result:



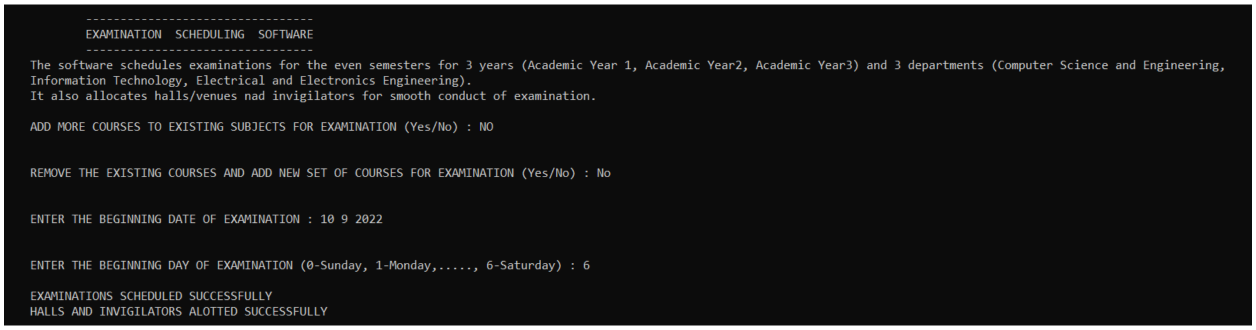
```
MA2271-ComplexFunctions&LaplaceTransforms-Theory-360
CI2271-BasicEEE-Theory-240
CS2201-SoftwareDevelopment-Theory-120
CS2202-DataScience-Theory-120
CS2217-C_Laboratory-Practical-120
AY2271-EnvironmentalScience-Theory-360
UG2297-DesignThinking_Lab-Practical-360
HU2271-HumanitiesElective1-Theory-360
EE2271-Electrical&InstrumentationEngg-Theory-120
IT2201-Programming&DataStructures-Theory-120
IT2202-InformationScience-Theory-120
IT2217-DataStructures_Lab-Practical-120
EE2201-Devices&Circuits-Theory-120
EE2202-Circuits&Networks-Theory-120
EE2203-Circuits&Devices_Lab-Practical-120
MA2471-Probability&Statistics-Theory-240
AY2471-IndianConstitution-Theory-360
CS2401-ComputerArchitecture-Theory-120
CS2402-OperatingSystems-Theory-120
CS2403-DesignofAlgorithms-Theory-120
CS2404-DatabaseManagement-Theory-120
CS2411-OperatingSystem_Lab-Practical-120
CS2412-Database_Lab-Practical-120
IT2401-Microprocessors&Controllers-Theory-120
IT2402-AdvancedDataStructures-Theory-120
IT2403-DataCommunication-Theory-120
IT2404-AutomataTheory&Compilers-Theory-120
IT2411-NetworkProgramming_Lab-Practical-120
IT2412-DigitalSystems_Lab-Practical-120
EE2401-Micro-Controllers-Theory-120
EE2402-DigitalProcessing-Theory-120
EE2403-ControlSystems-Theory-120
EE2404-ElectromagneticFields-Theory-120
EE2405-CommunicationSystems-Theory-120
EE2411-Microcontrollers_Lab-Practical-120
EE2412-DigitalSignals_Lab-Practical-120
CS2601-InternetProgramming-Theory-120
CS2602-SystemSecurity-Theory-120
CS2603-MachineLearning-Theory-120
```

Fig1.1 Contents of file ‘Courses.txt’



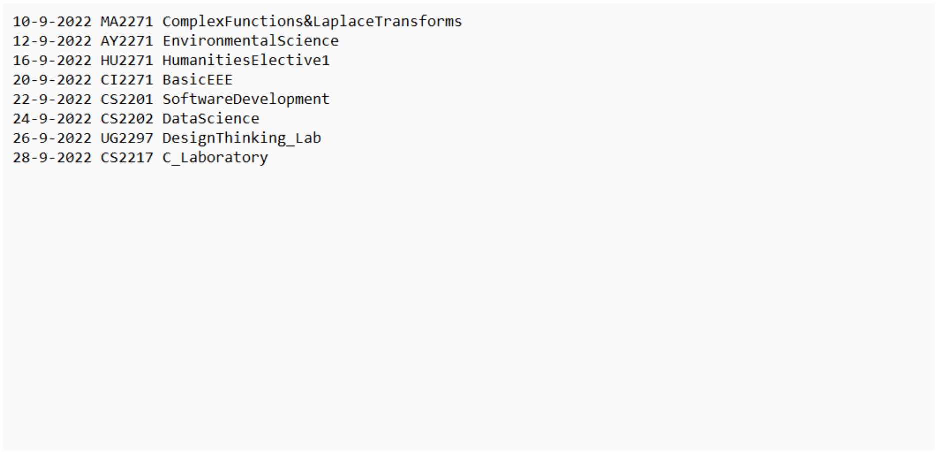
```
MA2171-Matrices&Calculus-Theory-25
PH2101-EngineeringPhysics-Theory-20
AY2101-EngineeringChemistry-Theory-50
UG2197-Python-Theory-70
UG2198-EngineeringGraphics-Theory-25
HU2101-TechnicalEnglish-Theory-25
MA2371-DiscreteMathematics-Theory-50
CS2301-DataStructures-Theory-25
CS2302-ObjectOrientedProgramming-Theory-40
IT2301-Programming&DesignPatterns-Theory-30
IT2302-DatabaseTechnology-Theory-25
CS2501-ComputerNetworks-Theory-40
CS2502-ArtificialIntelligence-Theory-30
CS2503-SoftwareEngineering-Theory-25
IT2501-SoftwareEngineering-Theory-50
IT2502-DataAnalytics-Theory-20
IT2503-ArtificialIntelligence-Theory-25
```

Fig1.2 Contents of file ‘Arrears.txt’



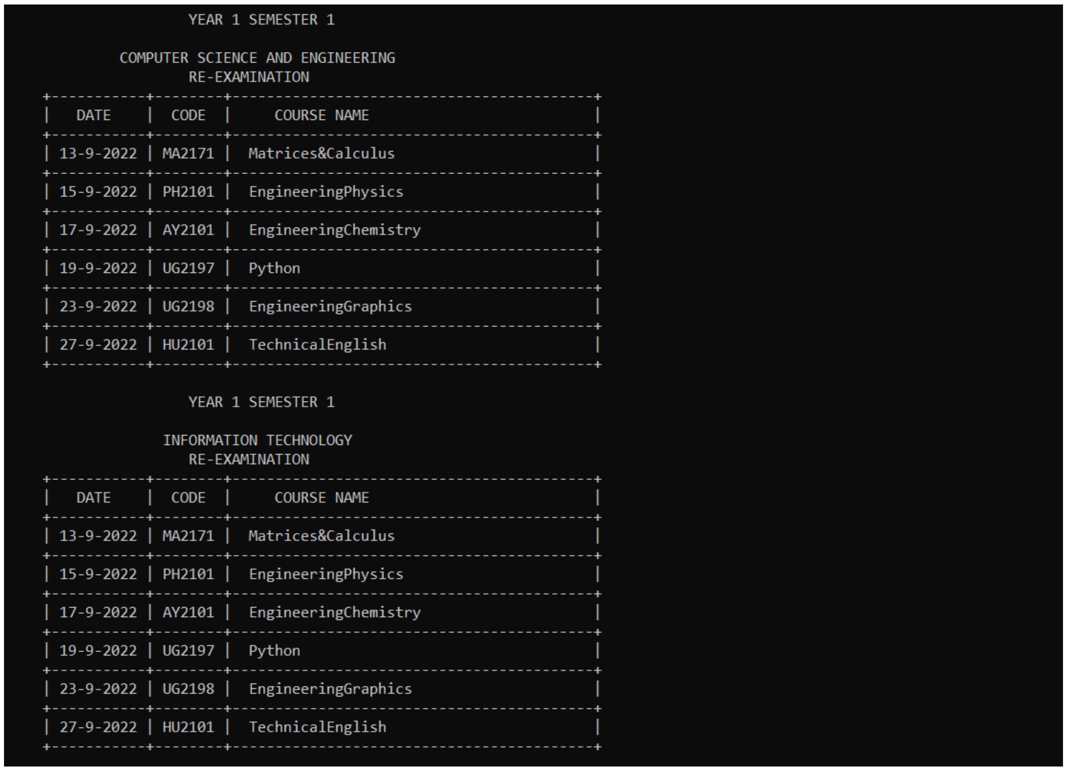
```
        ---------------------------------
        EXAMINATION  SCHEDULING  SOFTWARE
        ---------------------------------
The software schedules examinations for the even semesters for 3 years (Academic Year 1, Academic Year2, Academic Year3) and 3 departments (Computer Science and Engineering,
Information Technology, Electrical and Electronics Engineering).
It also allocates halls/venues nad invigilators for smooth conduct of examination.

ADD MORE COURSES TO EXISTING SUBJECTS FOR EXAMINATION (Yes/No) : NO


REMOVE THE EXISTING COURSES AND ADD NEW SET OF COURSES FOR EXAMINATION (Yes/No) : No


ENTER THE BEGINNING DATE OF EXAMINATION : 10 9 2022


ENTER THE BEGINNING DAY OF EXAMINATION (0-Sunday, 1-Monday,....., 6-Saturday) : 6

EXAMINATIONS SCHEDULED SUCCESSFULLY
HALLS AND INVIGILATORS ALOTTED SUCCESSFULLY
```

Fig1.3 Input from user (Starting date of exam)



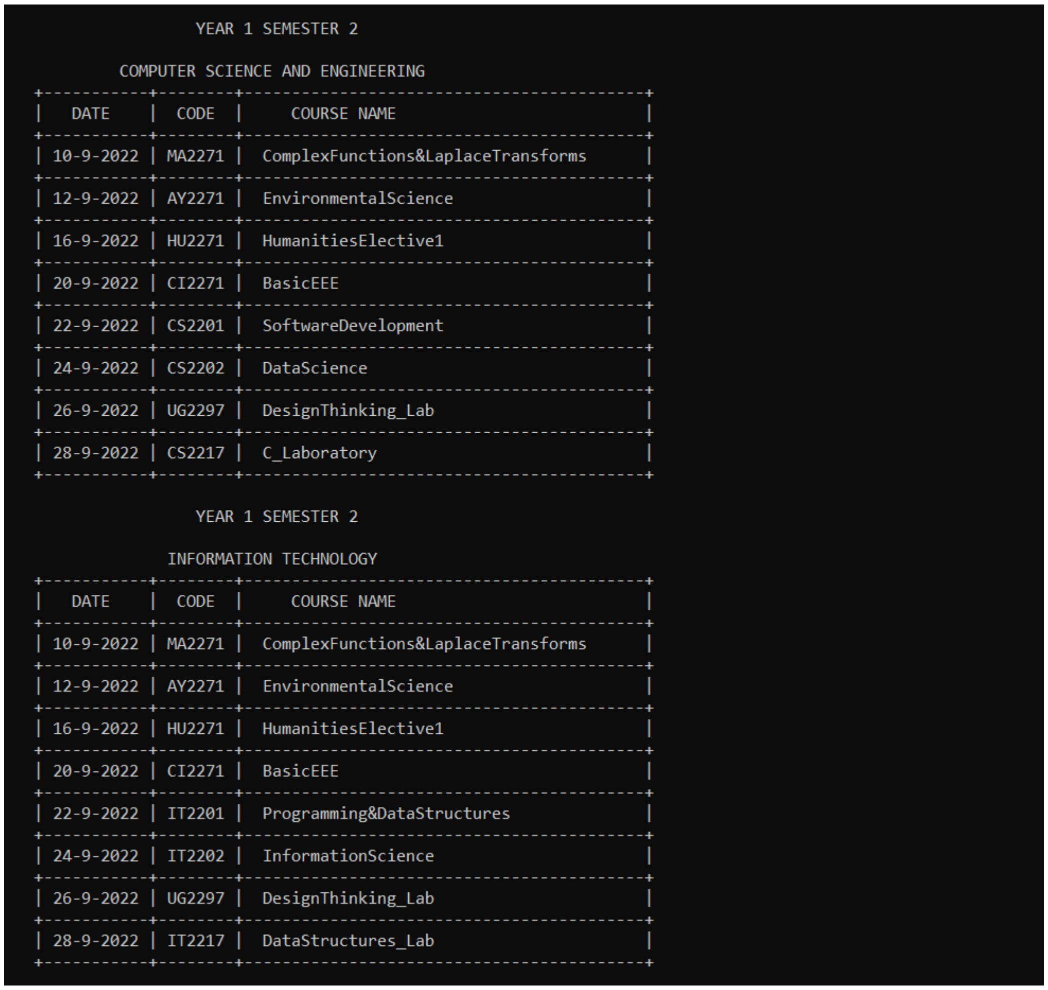
```
10-9-2022 MA2271 ComplexFunctions&LaplaceTransforms
12-9-2022 AY2271 EnvironmentalScience
16-9-2022 HU2271 HumanitiesElective1
20-9-2022 CI2271 BasicEEE
22-9-2022 CS2201 SoftwareDevelopment
24-9-2022 CS2202 DataScience
26-9-2022 UG2297 DesignThinking_Lab
28-9-2022 CS2217 C_Laboratory
```

Fig1.4 File Storing the Examination Schedule



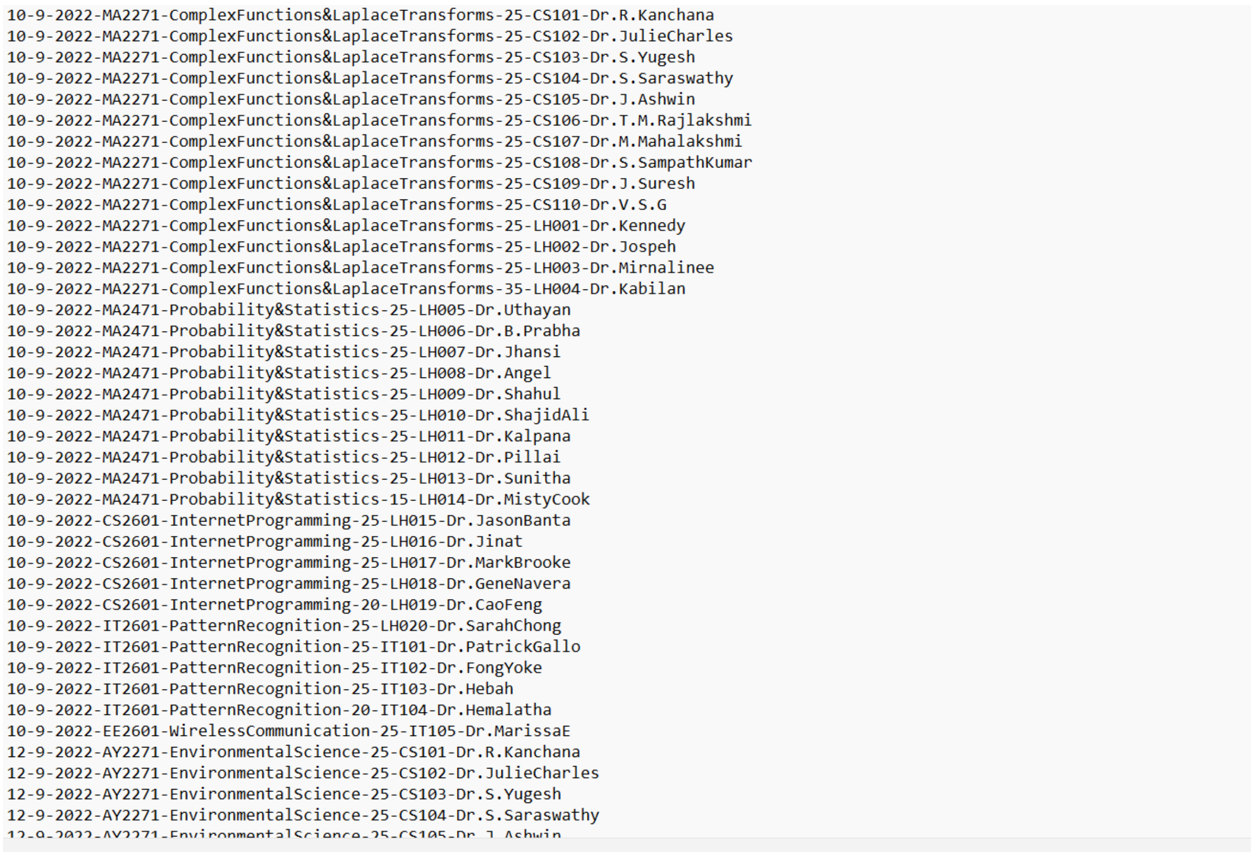
```
                 YEAR 1 SEMESTER 1

         COMPUTER SCIENCE AND ENGINEERING
                 RE-EXAMINATION
+-----------+--------+-------------------------------------------+
|   DATE    |  CODE  |     COURSE NAME                           |
+-----------+--------+-------------------------------------------+
| 13-9-2022 | MA2171 |  Matrices&Calculus                        |
+-----------+--------+-------------------------------------------+
| 15-9-2022 | PH2101 |  EngineeringPhysics                       |
+-----------+--------+-------------------------------------------+
| 17-9-2022 | AY2101 |  EngineeringChemistry                     |
+-----------+--------+-------------------------------------------+
| 19-9-2022 | UG2197 |  Python                                   |
+-----------+--------+-------------------------------------------+
| 23-9-2022 | UG2198 |  EngineeringGraphics                      |
+-----------+--------+-------------------------------------------+
| 27-9-2022 | HU2101 |  TechnicalEnglish                         |
+-----------+--------+-------------------------------------------+

                 YEAR 1 SEMESTER 1

              INFORMATION TECHNOLOGY
                 RE-EXAMINATION
+-----------+--------+-------------------------------------------+
|   DATE    |  CODE  |     COURSE NAME                           |
+-----------+--------+-------------------------------------------+
| 13-9-2022 | MA2171 |  Matrices&Calculus                        |
+-----------+--------+-------------------------------------------+
| 15-9-2022 | PH2101 |  EngineeringPhysics                       |
+-----------+--------+-------------------------------------------+
| 17-9-2022 | AY2101 |  EngineeringChemistry                     |
+-----------+--------+-------------------------------------------+
| 19-9-2022 | UG2197 |  Python                                   |
+-----------+--------+-------------------------------------------+
| 23-9-2022 | UG2198 |  EngineeringGraphics                      |
+-----------+--------+-------------------------------------------+
| 27-9-2022 | HU2101 |  TechnicalEnglish                         |
+-----------+--------+-------------------------------------------+
```

Fig1.5 Display of the Schedule (Arrear Examination)



```
                YEAR 1 SEMESTER 2

        COMPUTER SCIENCE AND ENGINEERING
+-----------+--------+-------------------------------------------+
|   DATE    |  CODE  |     COURSE NAME                           |
+-----------+--------+-------------------------------------------+
| 10-9-2022 | MA2271 |  ComplexFunctions&LaplaceTransforms       |
+-----------+--------+-------------------------------------------+
| 12-9-2022 | AY2271 |  EnvironmentalScience                     |
+-----------+--------+-------------------------------------------+
| 16-9-2022 | HU2271 |  HumanitiesElective1                      |
+-----------+--------+-------------------------------------------+
| 20-9-2022 | CI2271 |  BasicEEE                                 |
+-----------+--------+-------------------------------------------+
| 22-9-2022 | CS2201 |  SoftwareDevelopment                      |
+-----------+--------+-------------------------------------------+
| 24-9-2022 | CS2202 |  DataScience                              |
+-----------+--------+-------------------------------------------+
| 26-9-2022 | UG2297 |  DesignThinking_Lab                       |
+-----------+--------+-------------------------------------------+
| 28-9-2022 | CS2217 |  C_Laboratory                             |
+-----------+--------+-------------------------------------------+

                YEAR 1 SEMESTER 2

             INFORMATION TECHNOLOGY
+-----------+--------+-------------------------------------------+
|   DATE    |  CODE  |     COURSE NAME                           |
+-----------+--------+-------------------------------------------+
| 10-9-2022 | MA2271 |  ComplexFunctions&LaplaceTransforms       |
+-----------+--------+-------------------------------------------+
| 12-9-2022 | AY2271 |  EnvironmentalScience                     |
+-----------+--------+-------------------------------------------+
| 16-9-2022 | HU2271 |  HumanitiesElective1                      |
+-----------+--------+-------------------------------------------+
| 20-9-2022 | CI2271 |  BasicEEE                                 |
+-----------+--------+-------------------------------------------+
| 22-9-2022 | IT2201 |  Programming&DataStructures               |
+-----------+--------+-------------------------------------------+
| 24-9-2022 | IT2202 |  InformationScience                       |
+-----------+--------+-------------------------------------------+
| 26-9-2022 | UG2297 |  DesignThinking_Lab                       |
+-----------+--------+-------------------------------------------+
| 28-9-2022 | IT2217 |  DataStructures_Lab                       |
+-----------+--------+-------------------------------------------+
```

Fig1.6 Display of the Schedule (Main Examination)



```
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS101-Dr.R.Kanchana
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS102-Dr.JulieCharles
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS103-Dr.S.Yugesh
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS104-Dr.S.Saraswathy
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS105-Dr.J.Ashwin
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS106-Dr.T.M.Rajlakshmi
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS107-Dr.M.Mahalakshmi
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS108-Dr.S.SampathKumar
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS109-Dr.J.Suresh
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-CS110-Dr.V.S.G
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-LH001-Dr.Kennedy
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-LH002-Dr.Jospeh
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-25-LH003-Dr.Mirnalinee
10-9-2022-MA2271-ComplexFunctions&LaplaceTransforms-35-LH004-Dr.Kabilan
10-9-2022-MA2471-Probability&Statistics-25-LH005-Dr.Uthayan
10-9-2022-MA2471-Probability&Statistics-25-LH006-Dr.B.Prabha
10-9-2022-MA2471-Probability&Statistics-25-LH007-Dr.Jhansi
10-9-2022-MA2471-Probability&Statistics-25-LH008-Dr.Angel
10-9-2022-MA2471-Probability&Statistics-25-LH009-Dr.Shahul
10-9-2022-MA2471-Probability&Statistics-25-LH010-Dr.ShajidAli
10-9-2022-MA2471-Probability&Statistics-25-LH011-Dr.Kalpana
10-9-2022-MA2471-Probability&Statistics-25-LH012-Dr.Pillai
10-9-2022-MA2471-Probability&Statistics-25-LH013-Dr.Sunitha
10-9-2022-MA2471-Probability&Statistics-15-LH014-Dr.MistyCook
10-9-2022-CS2601-InternetProgramming-25-LH015-Dr.JasonBanta
10-9-2022-CS2601-InternetProgramming-25-LH016-Dr.Jinat
10-9-2022-CS2601-InternetProgramming-25-LH017-Dr.MarkBrooke
10-9-2022-CS2601-InternetProgramming-25-LH018-Dr.GeneNavera
10-9-2022-CS2601-InternetProgramming-20-LH019-Dr.CaoFeng
10-9-2022-IT2601-PatternRecognition-25-LH020-Dr.SarahChong
10-9-2022-IT2601-PatternRecognition-25-IT101-Dr.PatrickGallo
10-9-2022-IT2601-PatternRecognition-25-IT102-Dr.FongYoke
10-9-2022-IT2601-PatternRecognition-25-IT103-Dr.Hebah
10-9-2022-IT2601-PatternRecognition-20-IT104-Dr.Hemalatha
10-9-2022-EE2601-WirelessCommunication-25-IT105-Dr.MarissaE
12-9-2022-AY2271-EnvironmentalScience-25-CS101-Dr.R.Kanchana
12-9-2022-AY2271-EnvironmentalScience-25-CS102-Dr.JulieCharles
12-9-2022-AY2271-EnvironmentalScience-25-CS103-Dr.S.Yugesh
12-9-2022-AY2271-EnvironmentalScience-25-CS104-Dr.S.Saraswathy
12-9-2022-AY2271-EnvironmentalScience-25-CS105-Dr.J.Ashwin
```

Fig1.7 Hall and Invigilators Allotted in ‘HallAllotment.txt’

Conclusion:

The project is being developed in a way that leaves room for the code to be reused in other scheduling-related issues. The solution is flexible and sustainable because it is divided into easily adaptable modules. The flow is unaffected by changing the number of modules as needed because each module has its own functionality. The scheduling issue provides a thorough understanding of the resources and work required to plan an exam, and using programming expertise to solve the issue has been extremely insightful. The vision we had in mind as we developed the project was to satisfy the needs of all the project's stakeholders, including teachers, students, management etc. and all the efforts were directed in realizing that vision.

References:

[1] Gunawan, Aldy & Ng, Kien & Poh, Kim. (2006). A Mathematical Programming Model For A Timetabling Problem.. 42-47. ---- Mathematical Research

[2] Kumar Bania, Rubul & Duarah, Pinkey. (2018). Exam Timetable Scheduling using Graph Coloring Approach. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 6. 10.26438/ijcse/v6i5.8493.

[3] Asmuni, Hishammudin and Burke, Edmund K and Garibaldi, Jonathan M and McCollum, Barry and Parkes, Andrew J*, An investigation of fuzzy multiple heuristic orderings in the construction of university examination timetables*, Computers & Operations Research 2009

[4] Burke, Edmund & Elliman, David & Weare, Rupert. (1994). A Genetic Algorithm Based University Timetabling System. 1.

[5] D. Corne, H.L. Fang, C. Mellish “Solving the Modular Exam Scheduling Problem with Genetic Algorithms”, DAI Research Paper No. 622

[6] Qu, Rong and Burke, EK and McCollum, Barry and Merlot, Liam TG and Lee, Sau Y, *A survey of search methodologies and automated approaches for examination timetabling*, Computer Science Technical Report No. NOTTCS-TR-2006-4, UK, 2006