PROJECT EAZY-Schedule : Timetable Generator

A PROJECT REPORT

Submitted by

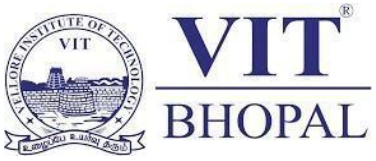
Shivalika Rastogi	**(20BCE10225)**
Adhishta Sharma	**(20BCE10307)**
Anurag Singh	**(20BCE10409)**
Subhadra Bansal	**(20BCE10861)**
Ananya M. Jha	**(20BCE11054)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING



SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRI KALAN, SEHORE MADHYA PRADESH - 466114

APRIL 2022

VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE MADHYA PRADESH – 466114

BONAFIDE CERTIFICATE

Certified that this project report titled " **PROJECT Eazy-Schedule :- Timetable Generator** " is the bonafide work of **Shivalika Rastogi (20BCE10225), Adhishta Sharma (20BCE10307), Anurag Singh (20BCE10409), Subhadra Bansal (20BCE10861), Ananya Jha (20BCE11054)** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr. Sandip Mal
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr. Anand Motwani.
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition II Examination is held on **29th April, 2022.**

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Lord Almighty for his presence and immense blessings throughout the project work.

We wish to express my heartfelt gratitude to Dr. Sandip Mal, Head of the Department, School of Computer Science, for much of his valuable support and encouragement in carrying out this work.

We would like to thank our internal guide, Dr. Anand Motwani, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Computer Science, who extended directly or indirectly all support.

Last, but not least, we are deeply indebted to our parents and group mates who have been the greatest support while working day and night for the project to make it a success.

LIST OF ABBREVIATIONS

S.No.	ABBREVIATION	MEANING
1	G.A.	Genetic Algorithm
2	FFCS	Fully Flexible Credit System
3	TT	Timetable
4	AI&ML	Artificial Intelligence & Machine Learning
5	UI	User Interface
6	CPU	Central Processing Unit
7	HDD	Hard Disk Drive

LIST OF FIGURES AND GRAPHS

ABSTRACT

The project focuses on getting away with the manual complexity students face while formulating their college timetables by providing them a ready to use, easy and efficient tool to automatically generate their timetables using machine learning algorithms.

Specific, to **VIT Bhopal** each semester a unique **FFCS (Fully Flexible Credit System)** is provided to the students so that they can make their own timetable. However, many inherent complexities are involved in this process and often the students have to toil hard manually to get their desired and suitable timetable.

This project **" Eazy Schedule – FFCS Timetable Generator "** focuses on creating a user friendly interface where students have the flexibility to add their available options like available teachers, slots, courses, slots, etc. and based on their input choices get a variety of automatically generated timetable samples through the machine learning and computing.

Artificial Intelligence & Machine Learning (AI&ML) is one of the rising computing solution due to an increase in computing power. More specifically **Genetic Algorithm** is a metaheuristic approach that mimics the process of natural selection. It can be applied to different types of problems such as non-deterministic polynomial time problems, to create optimal solutions, in our case the timetable scheduling problem.

Therefore, using evolving computational technology and emerging algorithms, the project generates multiple possible permutations for available timetables as **'good enough'** or **'optimal solutions'**, with each result fulfilling the **constraints** initially specified into the algorithm, mirroring real life limitations

involved in timetable scheduling.

TABLE OF CONTENTS

CHAPTER – 1
PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

Time table is a time management tool which allocates time to activities, events, or tasks that need to take place in a sequential manner. Timetable scheduling produces desired outcomes which follow all constraints and optimizes the user's time. Timetables can be customized in different ways to fulfill various requirements by changing the constraints and other requirements. Small scale scheduling can be done manually whereas on a larger population, computer assistance is needed. To solve this problem there are many algorithms which are specifically developed for problems such as timetable scheduling.

Timetable scheduling is a constraint satisfaction problem [CSP] in which the solution needs to satisfy a given set of constraints which may or may not be required. Satisfying the smallest of the constraints can take a lot of effort and manually making timetables can be a tedious task as population and number of constraints increase.

1.2 Motivation for the work

VIT Bhopal FFCS provides a great interface for timetable scheduling for its students. However, due to the large number of students there are many conflicts during the FFCS course registration. To make this process more streamlined and efficient for students as well as the administration, this project was proposed.

This project not only is limited till the FFCS timetable scheduling, but can be customized and used in other fields as well.

1.3 Introduction to the project

The project is a simple system wherein the user inputs the basic data set and an output structure will be generated as a result giving the solution.

The user inputs the choice of instructor, rooms, sections, subjects, and other settings for the program. The system processes the information using genetic algorithms, finds the most fit population and generates 5 most optimal timetables from the input data. The input data can also be added through a CSV file and works the same as manually inputting data in the interface. The output then takes feedback and retakes the input if necessary and the cycle goes on until the most optimal solution is generated.

1.4 Problem Statement

The University's timetable scheduling consists of many complex constraints such as a vast number of courses, different programs, and various slots which make timetable scheduling more complicated and time consuming. The manual way of selecting courses, teachers and time slots for each semester is a laborious task for students as it causes confusion and chaos among the students as the number of courses per semester increases. To make the process of scheduling more streamlined and automated, AI provides many new algorithms that can produce optimized results with customized constraints.

1.5 Objective of the work

The method will significantly minimize the time and effort required to generate a timetable, freeing up more time for the administrator. This will also be relevant to the institute's exam scheduling and several educational institutions can use the technology and profit from a calculated scheduling solution. It will assist to ease university management and maybe cut costs by having a more used timetable. The approach is designed to accommodate scheduling at all levels of academics.

1.6 Organization of the project

Designing system architecture, data schematics, and models was the first step in the planning process. When an iteration begins, the feature to be completed is determined and planned. Iterations are divided into two types: adding a feature or a model and modifying values. The purpose of each cycle is then implemented and personally tested.

1.7 Summary

The system was also intended to be basic and easy to use. This removes any misunderstanding produced by dispersed user interface controls and allows the product to be fully exploited.

CHAPTER - 2

RELATED WORK INVESTIGATION

2.1 Core area of the project

This section includes the research and investigation done before and during the making of the project including search for needed data as well as existing tools to deal with the same problem we are trying to solve. The pros and cons of the existing system(s) are also discussed. All the issues that presented themselves in the investigation are outlined as well.

2.2 Existing Approaches/Methods

- A Review of Optimization Algorithms for University Timetable Scheduling - https://www.etasr.com/index.php/ETASR/article/view/3832

➢ Because university course timetabling is an NP-hard problem, there are several methods that have been explored in attempts to find optimal solutions. The reviewed algorithms take into account the demands of institutional constraints for course timetable management.

➢ This study helped us to review several optimization algorithms that could be applied as possible solutions for the university student course timetable problem. However, the application of these methods to solve practical problems has not been verified at the same proportion as the rate of their development. One of the reasons is the complexities of the real behavior and the model generated, which are described by non-linear behavior functions and whose solution space may be non-convex.

➢ Scheduling problems are usually addressed by heuristic techniques due to their structure and complexity. Primarily, they involve values of the functions in the process, but on the other hand, they demand a large number of calculations. Among

the heuristic optimization techniques, there are metaheuristic methods, which promise accurate and optimal solutions to the timetable scheduling issue.

➢ After studying the comparison between the different scheduling and optimization algorithms, it was concluded that the most fit algorithm to use for the proposed work is ‘Genetic algorithm’.

- **Theses and dissertations on Genetic Algorithm For University Course Timetabling Problem - https://egrove.olemiss.edu/etd/443/**

➢ The aim of this thesis is to optimize a general university course scheduling process based on genetic algorithms using some defined constraints. Genetic algorithms, based on Darwin's theory of evolution, is the chosen algorithm to obtain optimal solutions to the timetabling problem.

➢ This thesis gave a detailed analysis about Fundamentals of Genetic Algorithm (GA), How Genetic Algorithms works, initial population of chromosomes, suitability of chromosomes to mate, selection to the mating pool, methods of change in reproduction, advantages of genetic algorithm, disadvantages of genetic algorithm, design and implementation strategy. This thesis helped us understand the concept of genetic algorithm better and solidified the decision to use this method for the proposed work.

CHAPTER 3
REQUIREMENT ARTIFACTS

3.1 Introduction

For creating the project several software tools are required both for frontend and backend development. This section coneys the various software requirements of this system, in terms of expectations and the outcomes for the users.

3.2 Functional Requirements

- It should provide/display details like available faculties, rooms, subjects, etc. to students (User) so that they can select their preferred choices to generate their timetable.
- It should generate a schedule/timetable without any of clashes among faculties, day, time and room keeping in mind the before set constraints to the system.
- It should generate the output as an easily viewable display or in a storable .csv (Excel format), which contains all the possible optimum solutions each iteration (Chromosome population) has generated.

3.2.1 Data Requirements

- In terms of the data, the system is expected to take the desired inputs from the USER – which depends either maybe the Admin – who can add/delete/edit the details or it may be the end Users - the Students who can select their choices.
- User (Admin) should be provided the choices to add new faculties, new subjects, slots and alter the time slot availability for each of them as per demand to generate timetable scheduling options.

3.2.2 UI Requirements

The user experience and utility of the system heavily depends upon how the user perceives the system in terms of its display and ease of operability.

The system is intended to be equipped with an attractive, clear, consistent and responsive UI that can be utilised efficiently. Content presentation and UI feature require that it should have

1. Easy Navigation
   - dedicated UI windows or tabs for each operation related to add/view/delete/modify timetable details such as timeslots, teachers, subjects, rooms, etc.
2. Simple and responsive interface
3. Meaningful and easily understandable logical layout
4. Appropriate use of colour and texture schemes.
5. User centric approach
   - Appropriate Click / Check-box buttons for above mentioned User choices.
6. User based settings and add – on features.
   - Ease of navigability and view/display/ generate results, etc tab options.

3.2 Summary

In this segment the various requirement artifacts of this project in terms of the expected performance and run – time efficiency and applicability of the project is mentioned.

CHAPTER 4
DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and goal :

The program can easily be described as a scheduling computing software wherein you input basic data sets and it will output a structured result. A simple input-process-model would be:

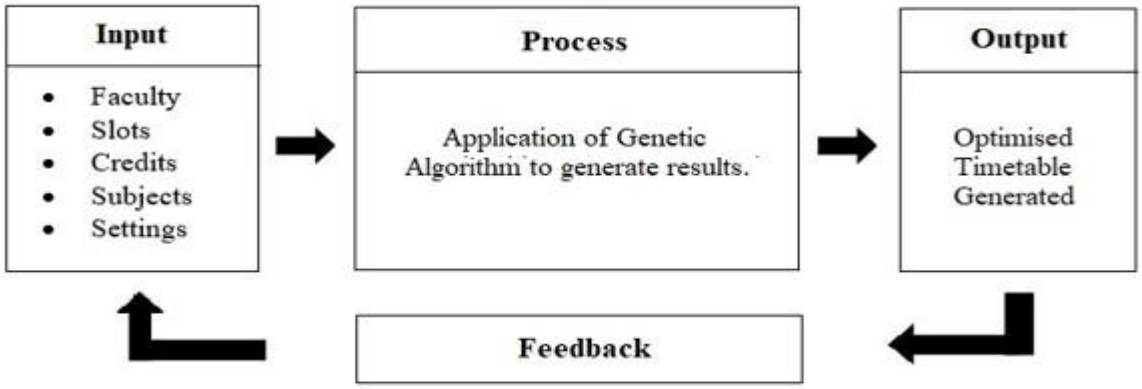


Fig : Simplified IPO-Model of the Program

Genetic algorithm (GA) is a metaheuristic approach which solves computational problems with a large population. It is in a category of evolutionary algorithms which is a subset of evolutionary computation in artificial intelligence. It is adapted from Darwinian theory of natural selection which is the natural process through which individual populations of organisms evolve. According to Darwin, the natural population of organisms changes according to individuals' adaptive traits. Organisms evolve according to their environment in order to survive. Individuals with adaptive traits—traits that give them some advantage—are more likely to survive and reproduce. These individuals then pass the adaptive traits on to their offspring. Over time, these advantageous traits become more common in the population. Through this process of natural selection, favorable traits are transmitted through generations.

GA mimics the biological process of gene and chromosome formation in the process of mutation and evolution. Each chromosome represents an individual organism and genes forming components of a solution to be employed with a genetic algorithm. As a result, a chromosome is a data structure. New chromosomes are created in the same way as they are in nature: by combining genetic material through crossover and mutation. Crossover is the biological equivalent of mating. Mutation is the means by which new information is introduced to the population. As a result, it is the chromosome that changes by changing the order and makeup of its genes. It's worth noting that only a small number of genes are employed in the procedure.

The basic steps of genetic algorithm are as follows:

1. Random population of chromosomes is initialized
2. Computing the fitness function for each chromosome to make a population
3. Selecting random chromosomes for mating from the resulted population
4. Producing new chromosomes from mating
5. Mutating genes randomly of the new offspring
6. Creating new population for with new chromosomes
7. Finding the most optimal solution from the best fitness function

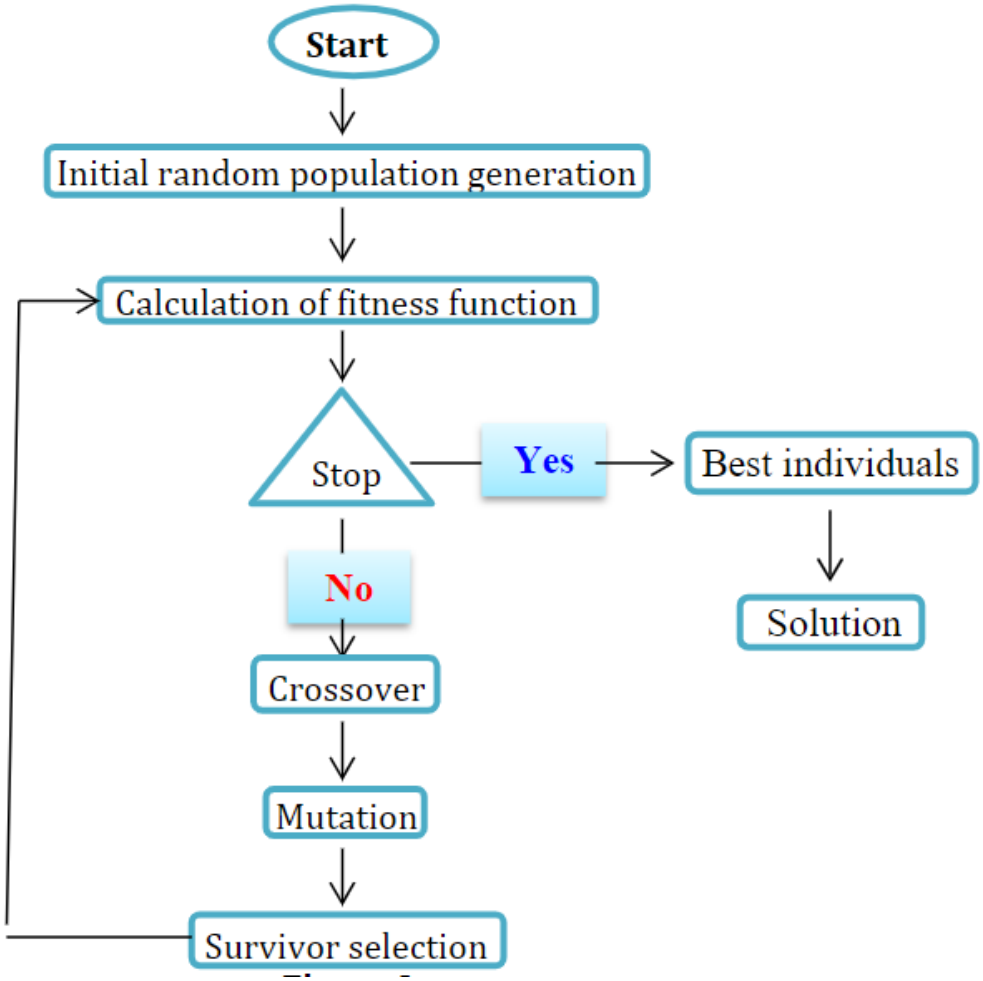


Fig 4.2 : Flowchart of Genetic Algorithm

The goal of a genetic algorithm is to improve a population by marrying the most suitable chromosomes in a population. It results in the development of children (chromosomes) that join the current population of chromosomes in following generations to become part of a mating population. A "fitness function" in the form of a numeric score must be designed to evaluate the solutions for the genetic algorithm to function properly (chromosomes). In nature, organisms do not receive a score; they simply perish or survive.

4.2 System design and implementation :

The fitness calculation would be largely based on the amount of plotted subject compared to the required subjects. However, the calculation would still depend on the

evaluation matrix provided by the user. The evaluation matrix is a set of constraint weights that has the capability to shape the solutions. It is a list of containing the prioritization of constraints using a distribution of a hundred percent (100%)

Name	**Function**
Subject Placement	Attempts to create valid schedule for all subjects of all sections.
Lunch Break	Attempts to at least have a 30-minute break between 11:00 AM and 1:00 PM
Section Rest	Attempts to at least give sections a 30-minute break for every 3 consecutive hours of session
Section Idle Time	Attempts to ensure that sections will have little or no idle time with respect to lunch break and section rest*. **Section rest is modified by removing the need for consecutive session.*
Instructor Rest	Attempts to at least give instructors a 30-minute break for every 3 consecutive hours of session
Instructor Load Balance	Attempts to balance the distribution of instructor load with respect to their pooled subjects.
Meeting Pattern	Attempts to ensure that all divided schedules are following a pattern instead of random date selection.

Fig 4.3 : Evaluation Parameters and Restraints

After evaluation, an additional step is added in order to perform the adaption of artificial intelligence to its current performance. Adaption is the process of changing running variables to cater for better results. This helps the artificial intelligence to avoid focusing on one set of problem which may cause pre-mature convergence. Adaption performs two important task which is population alignment and mutation rate adjustment.

4.2.1 Mutation Rate Adjustment

Mutation rate is the chance for each chromosome to get random change. Mutation rate adjustment happens when the trigger which is set before running is met. When a generation completes without triggering the adjustment, the mutation rate will decrement by 0.5% else it will be increased by 0.5%. Calculation of mutation rate adjustment trigger follows the formula below:

$$\boldsymbol{change} = 0.5 * (-1 \; if \; average \; fitness - previous \; average \; fitness < adjustment \; trigger \; settings \; else \; 1)$$

After evaluation, chromosomes are picked to participate for reproduction. Chromosomes with higher fitness is more likely to get picked. This is where the survival of the fittest comes. There are multiple ways of selecting chromosomes all of which have their pros and cons. Having the most appropriate type of selection will help avoid early convergence and promote diverse solutions. As an elitist variant of genetic algorithm, the top *n%* of population is guaranteed to proceed in the next generation with same genes depending on the settings.

The rest of the selection process is to be done by implementing multiple tournaments on the population. The participants in the tournament are selected randomly with a quantity of 4% of existing population but capped at twenty-five (25). The tournament will run until enough pairs are picked. The following example below shows a simulation where 40% of the 10 chromosomes are selected in the tournament.

Chromosomes	Fitness	Participant
#1	4.1	TRUE
#2	3.4	
#3	5.6	TRUE
#4	7.5	
#5	6.5	
#6	1.2	
#7	3.5	TRUE
#8	2.8	TRUE
#9	7.5	
#10	1.1	
Winner: Chromosome #3		

Fig 4.3 : Chromosome Generation and Fitness Evaluation

The mating pool consist of selected chromosomes that will undergo crossover. The crossover operation involves production of offspring based on genes of parents analogous to its biological counterpart. There are multiple ways to perform crossover and like selection, it is important to use the most appropriate one to have better outcome.

All offspring will take their parent's place in the population. This can be visualized below where:

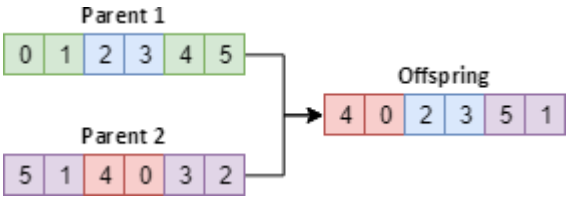


Fig 4.4 : Selection and Mutation Operations

Mutation is a genetic operator that helps maintain the variety of solutions. It can also prevent solutions from being stuck in local optima. Mutation is the process of altering the gene/s of a chromosome. Usually, mutation rate is low and fixed. However, adaptive genetic algorithm means that mutation rate may vary depending on the performance of the population.

Once the genetic algorithm terminates either by force or meeting the end criteria, the top 5 fittest chromosomes will be displayed for user review. The user may pick any from the proposed solution and get its output in a CSV file.

4.2.2 Database Design :

A Database Management System (DBMS) is a collection of computer programs that manages the creation, maintenance, and use of databases. It enables organizations to delegate database development control to database administrators (DBAs). A database management system (DBMS) is a system software package that facilitates the use of databases, which are integrated collections of data records and files.

It allows different user application programs to easily access the same database. DBMS's may use any of a variety of database models, such as the network model or relational model. In large systems, a DBMS allows users and other software to store and retrieve data in a structured way.

4.2.3 SQLITE

SqLite was initially built as a C- Language in order to implement a fast and reliable SQL database engine. Now SQLite is directly connected to core python in the form of a library known as ‘Sqlite3’, which does not need any external installation

In this project, input is taken from the user, so in order to run this algorithm, the data must be stored in a database. This project is a prototype demo software, hence basic DB has to be created.

For the database connectivity here, a function is created which is named as checksetup(). SQLite directly creates an SQL file and directly checks for the setup which creates the Queries and files after creating the table. Next the function calls the QTmain window which is the main GUI application for this timetable generator. It helps in running the executing and running components.

Fig 4.5 : Database Setup()

4.2.4 Designing of Application

The first table consists of the attributes given to the instructor entity. The first attribute is ID() which consists of unique value i.e faculty_id followed by Name() which has the names of the instructors and then comes hours() which means the number of hours

a faculty is supposed to take classes and the last attribute is schedule() which stores the information about the weekly timing schedule at which the classes will be held.

Add Instructor		
🔑	id	integer
	student_id	integer
	name	string
	hours	integer
	Schedule	text

The next table is named as ROOMS() which takes in the ID() value having information about the unique class id followed by the name() attribute. This attribute takes in the name of respective rooms and next is the Type() Value which means what type of class is it and then comes schedule() which stores the information about the weekly timing schedule at which the classes will be held.

Add Rooms		
🔑	id	integer
	Name	text
	Type	text
	Schedule	text

In the next table Subjects() have to be mentioned as an entity. Here id() will be the course id which is a unique value, then comes the name() which takes the name of the subjects. Next it will take hours() which will show the total number of hours each subject will have, followed by Description() which will give a brief explanation of the subject. The last attribute of this field is Instructor() which will show the available instructors for that particular course.

Add Subjects		
🔑	id	integer
	Name	integer
	hours	integer
	Description	text
	Instructor	text
	Code	text

The fourth entity of this program SLOTS() This entity will have attributes Slot_id() whill will have a unique id given to each slot,then comes the name() which stores the name of each individual slot. This is followed by hours() which will inform us about the number of hours in each of the slot. this entity will also have Subjects() as an attribute which will show which course is allotted to which slot.It will also have instructors() as one of its attributes which shows which faculty will get which slot for teaching

Add Slots		
🔑	Id	integer
	name	text
	Subjects	text
	Instructors	text
	hours	integer

4.1 Summary

In this segment, the methodology that is how the project is constructed is described along with the architectural and user interface design of the website with the help data flow diagrams.

CHAPTER 5
TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Outline

The application takes five major input; Instructors, rooms and subjects component supports addition of entry using the application and importation. The sections component allows creation of sections with special feature of sharing subject with other sections and the scenario manager component handles running configuration. There would be two types of output; using the result viewer of the application which can view the top five solutions of the last generated scenario and exporting the selected solution which will produce three comma separated values (CSV) files. Tests for reliability and performance of the program specifically the evaluation for the performance of the artificial intelligence will come from usage of application using actual records of VIT Bhopal University.

5.2 Technical Implementation

This research aims to create an artificial intelligence that can create timetable schedules. This only covers processing input data and generating human sensible results.

Computing the schedule would involve validation for each constraint. Constraints are set of rules that directs the acceptance of the output. There will be three types of constraints to be used;

5.2.1 Constraints

1. **Soft Constraints** – A set of rules that can be broken without affecting the validity of the output.
2. **Medium Constraints** – A set of rules that can be broken with an effect to the validity of the output. However, this can only be broken if the scenario is logically invalid or impossible.
3. **Hard Constraints** – A set of rules that would produce an invalid solution if broken.

Hard Constraint	Medium Constraint	Soft Constraint
Instructors teach one class at a time	Sections' subjects are placed on the schedule	Students should have only 30 minutes break for every two hours of session per day
Instructors teach at their available schedule	Sections should have at least 30 minutes vacant time between 11:00 AM to 1:00 PM for lunch break (Optional)	Subjects that are divided should follow the two defined meeting pattern which is "MWF" and "TTH"
Instructors can only take *N* amount of subjects dependent to their maximum amount of load	Sections should have at least 30 minutes vacant time for every 3 consecutive hours of session	Instructors should have normalized distributed load based on the instructor pool of subjects

Fig 5.1 : Constraints Fed to the System

Results of the application is not guaranteed to be the best possible solution for the scenario. The quality of the result relies heavily on running preference. Due to its stochastic nature, results may vary from poor to excellent. The system does not guarantee that every solution's hard constraint will be met especially when the presented scenario is logically impossible or intensely tight. This also means that when in strict scenario soft or medium constraints are subject to violation in order to meet hard constraints. Hard constraints are bound to be followed and will never be violated. Importation of values from CSV supports strict formatting shown below.

All CSV files should use line 1 of the file as file indicator. These should be one of the following "instructors", "rooms" and "subjects". The second line of the file should be used for defining table columns in same order.

5.2.2 Method of Software Development

Iterative development as a method for creating the software fits for continuously changing environment for artificial intelligence development. This enables the developer to learn every iteration and apply it to future instance. This methodology is fit for the research as the system relies on tweaking and managing constraints. Each iteration is an enhancement to the system which will repeat until the system delivers its expected result.

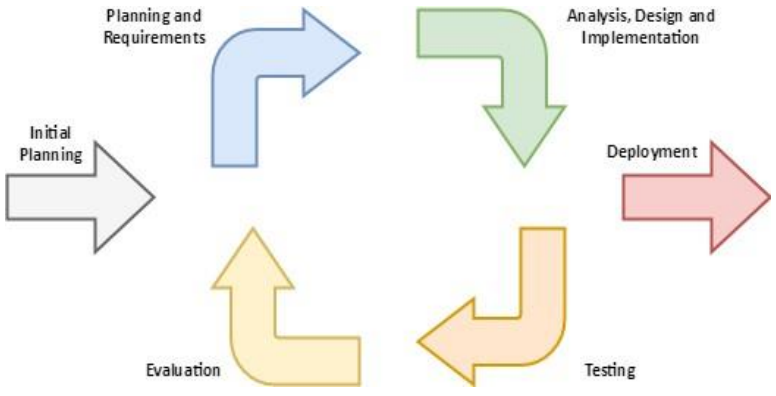
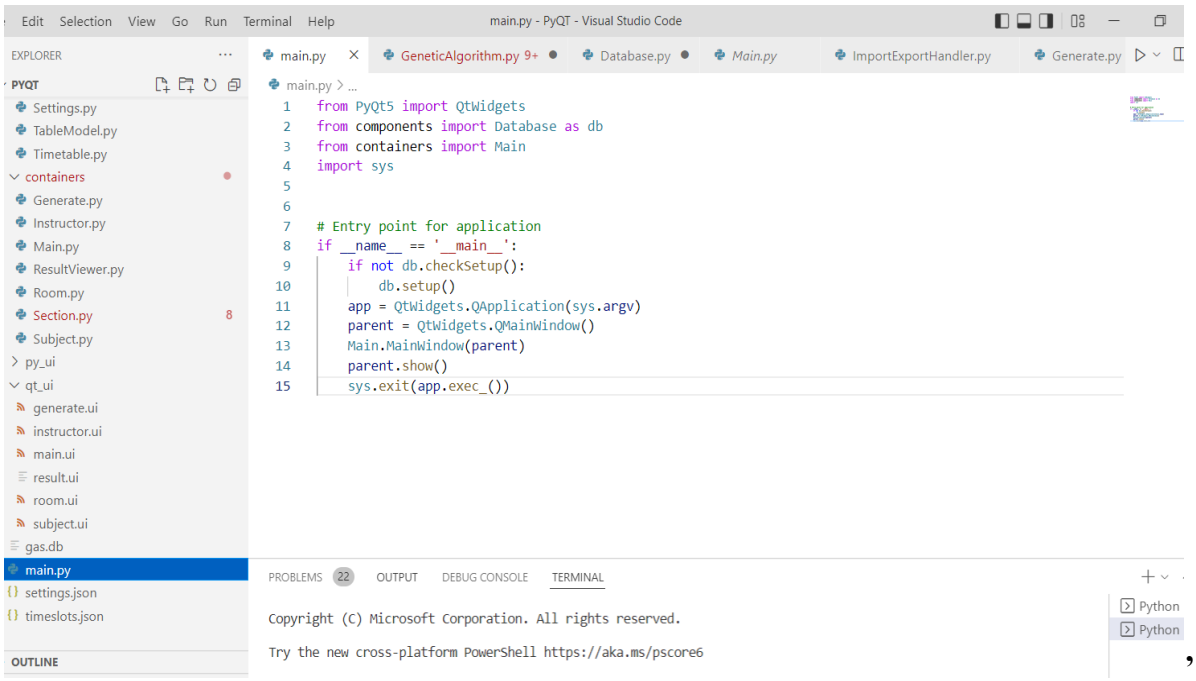


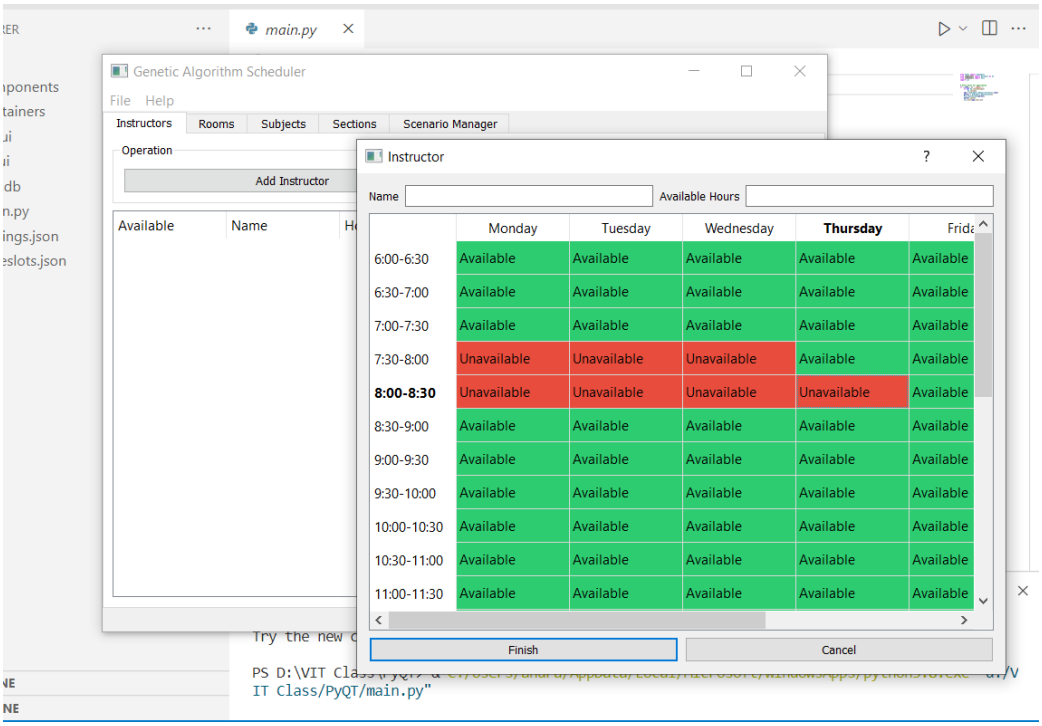
Fig 5.2 : Software Development Methodology

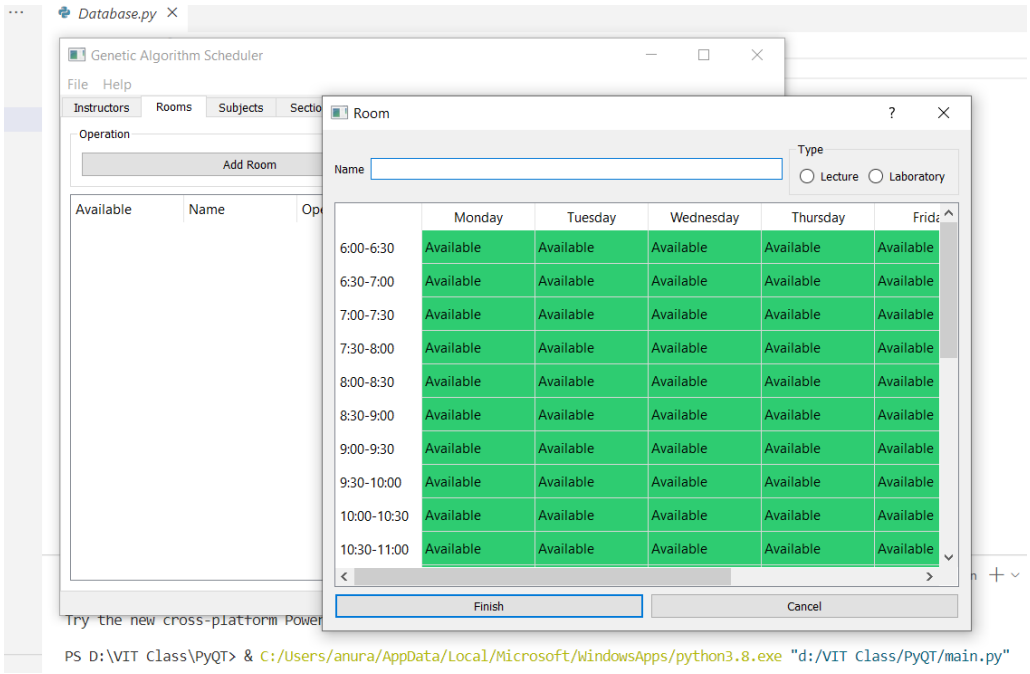
Initial planning has been done by designing system architecture, data schematic and models. Upon entering an iteration, the feature to be done is decided and planned. There are two types of iterations used; adding a feature or a model and tweaking values. Each iteration's goal is then implemented and then manually tested.

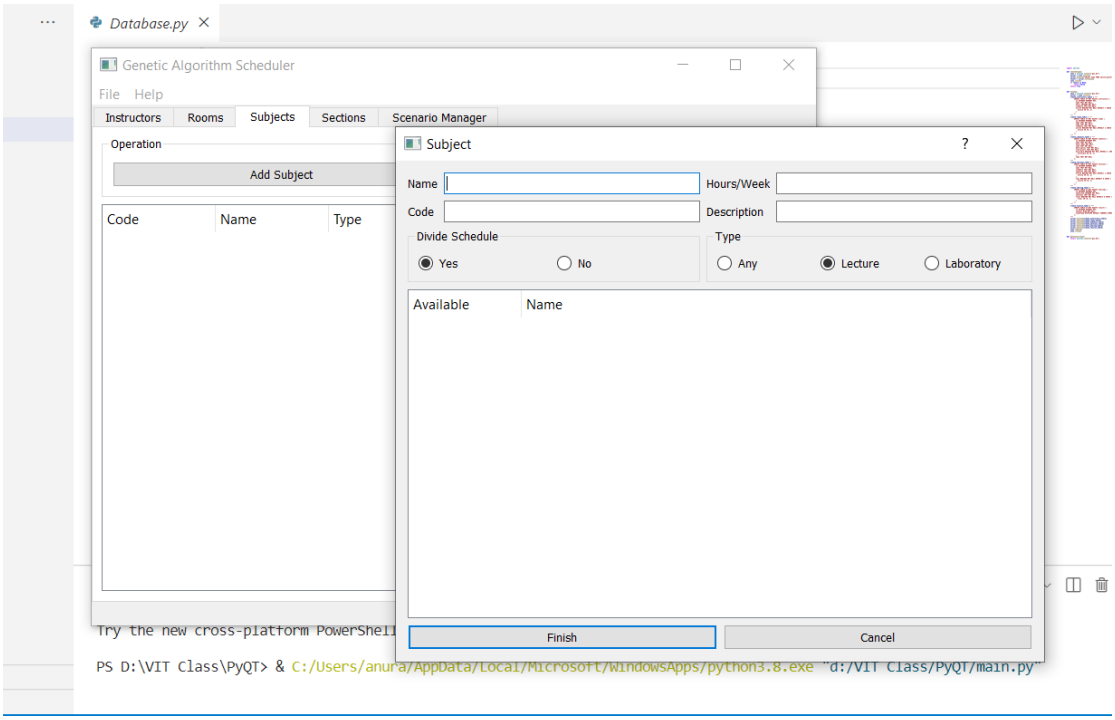
The surrogate modelling usage for this research makes use of an imaginary 1-dimensional metric basing on the evaluation matrix. The model will serve as a basis for evaluating the closeness of the solution to the ideal one. In conjunction with software evaluation methodology using the proposed steps above, data gathering would then lean towards data-centric approach. This is to provide correct conclusions towards the conclusion of the research. The proposed data to be gathered are the ones to be used by the evaluation above which will be composed of mostly result performance grade (fitness).

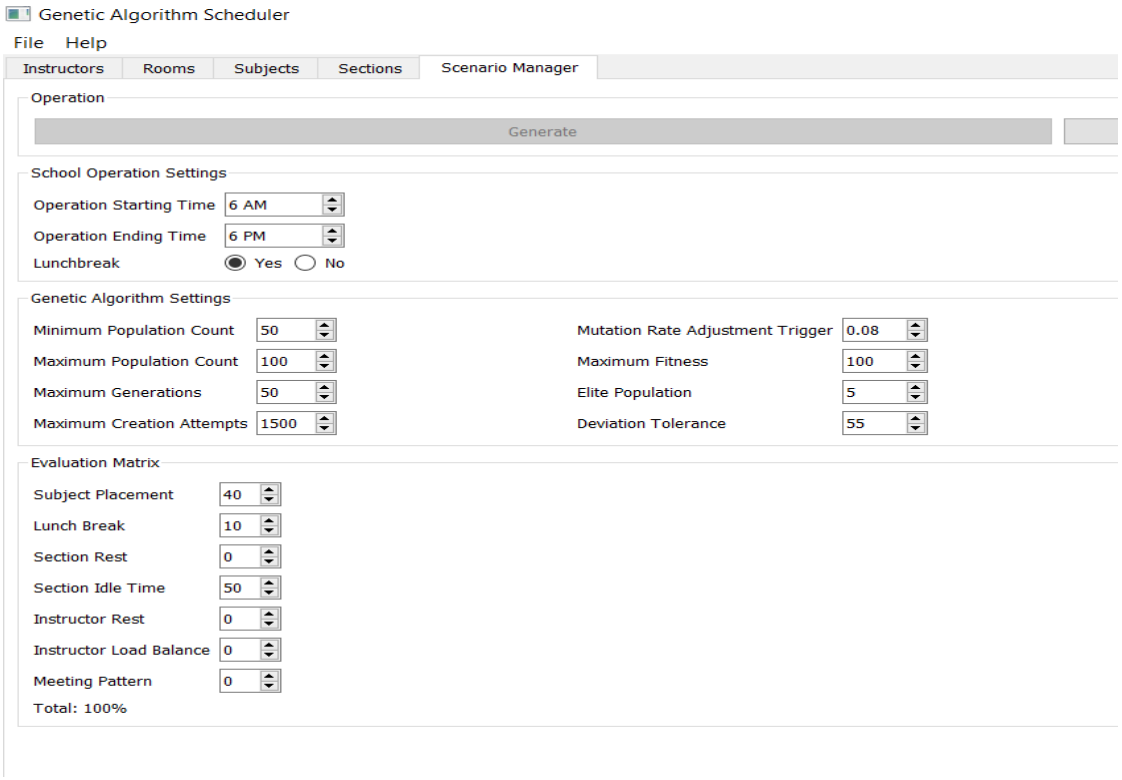
5.3 Testing :




main.py
Genetic Algorithm Scheduler
File Help
Instructors Rooms Subjects Sections Scenario Manager
Operation
Add Instructor
Available Name
Instructor
Name
Available Hours
Monday Tuesday Wednesday Thursday Friday
6:00-6:30 Available Available Available Available Available
6:30-7:00 Available Available Available Available Available
7:00-7:30 Available Available Available Available Available
7:30-8:00 Unavailable Unavailable Unavailable Available Available
8:00-8:30 Unavailable Unavailable Unavailable Unavailable Available
8:30-9:00 Available Available Available Available Available
9:00-9:30 Available Available Available Available Available
9:30-10:00 Available Available Available Available Available
10:00-10:30 Available Available Available Available Available
10:30-11:00 Available Available Available Available Available
11:00-11:30 Available Available Available Available Available
Finish
Cancel
PS D:\VIT Class\PyQT> & C:/Users/anura/AppData/Local/Microsoft/WindowsApps/python3.8.exe d:/VIT Class/PyQT/main.py"


Database.py
Genetic Algorithm Scheduler
File Help
Instructors Rooms Subjects
Operation
Add Room
Available Name
Room
Name
Type
Lecture Laboratory
Monday Tuesday Wednesday Thursday Friday
6:00-6:30 Available Available Available Available Available
6:30-7:00 Available Available Available Available Available
7:00-7:30 Available Available Available Available Available
7:30-8:00 Available Available Available Available Available
8:00-8:30 Available Available Available Available Available
8:30-9:00 Available Available Available Available Available
9:00-9:30 Available Available Available Available Available
9:30-10:00 Available Available Available Available Available
10:00-10:30 Available Available Available Available Available
10:30-11:00 Available Available Available Available Available
Finish
Cancel
PS D:\VIT Class\PyQT> & C:/Users/anura/AppData/Local/Microsoft/WindowsApps/python3.8.exe "d:/VIT Class/PyQT/main.py"


Database.py
Genetic Algorithm Scheduler
File Help
Instructors Rooms Subjects Sections Scenario Manager
Operation
Add Subject
Code Name Type
Subject
Name Hours/Week
Code Description
Divide Schedule
Yes No
Type
Any Lecture Laboratory
Available Name
Finish Cancel
PS D:\VIT Class\PyQT> & C:/Users/anura/AppData/Local/Microsoft/WindowsApps/python3.8.exe "d:/VIT Class/PyQT/main.py"


Genetic Algorithm Scheduler
File Help
Instructors Rooms Subjects Sections Scenario Manager
Operation
Generate
School Operation Settings
Operation Starting Time 6 AM
Operation Ending Time 6 PM
Lunchbreak Yes No
Genetic Algorithm Settings
Minimum Population Count 50
Maximum Population Count 100
Maximum Generations 50
Maximum Creation Attempts 1500
Mutation Rate Adjustment Trigger 0.08
Maximum Fitness 100
Elite Population 5
Deviation Tolerance 55
Evaluation Matrix
Subject Placement 40
Lunch Break 10
Section Rest 0
Section Idle Time 50
Instructor Rest 0
Instructor Load Balance 0
Meeting Pattern 0
Total: 100%

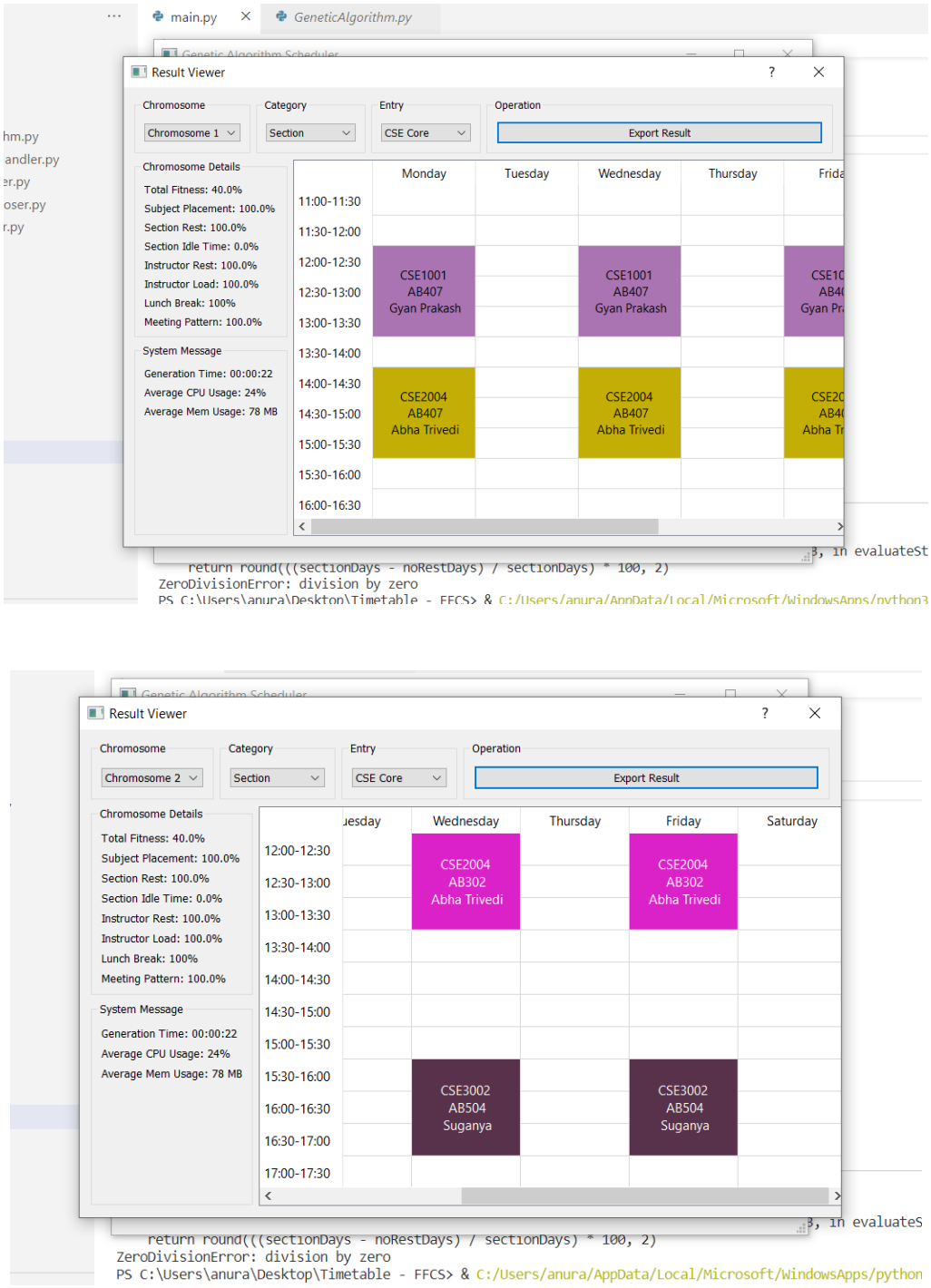


Fig 5.3 : Implementation Screenshots

CHAPTER 6
PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

In this segment, the outcome of our **" PROJECT EAZY Schedule :- FFCS Timetable Generator "** is described along with its application in real-world scenario.

6.2 Key implementations outline of the System

1) Clear & Measurable Objectives.

2) Well-structured implementation plan.

3) Relevant features for the user.

4) Simple user interface design.

The findings demonstrated that the system can provide genuine solutions that can be employed. It does not, however, enable total automation. There are still certain circumstances that would have the operator adjusting some inputs in order to generate a flawless answer.

Because solutions are generated dynamically, the system's simplicity and the inclusion of adjustable algorithm's objective and performance decreased the need for so many limitations. This allows users to quickly use and experiment with the programmer until they discover the ideal match for their situation.

The enormous number of testing permutations required to establish an appropriate evaluation for the application has proven to be a pipe dream. However, it may be argued that the system was able to create findings that, although being flawed, were still useful based on the models presented.

The system's solutions will be greatly influenced by the operating setup and

assessment matrix. If the programmer is given enough time and processing capacity, it may discover a flawless answer.

The comprehensive assessment of the system will remain difficult to solve due to the algorithm's flexibility of configuration providing a vast number of options.

6.3 Significant project outcomes

The following are the outcomes of our project:

- The algorithm provides students with a simple tool to generate the possible permutations of timetables based on their choices.
- The user-friendly algorithm GUI very well depicts the results of various timetables generated in each iteration in a easy to interpret and understandable form.
- Flexibility to the user to input data: either manually addition of the entries or choices or selecting the prefilled .csv (EXCEL) files.
- The database design and conversion at the backend works very well in terms of inputting the data through user and by creating separate files for each result generation.

6.4 Project applicability in Real-world applications

Real-time analytics show you that there is a lot of vast applications for this project concept :

- A standalone web-application version of the project can be put into use at various universities and education institutes all around, to save their time and improve their efficiency in timetable scheduling process.
- Since, it’s a metaheuristic approach, the algorithm can be put to use in other similar timetable generation problems such as Railways (train timetable), Sports tournament scheduling, etc.

6.5 Inference

Digitalization is not just a change of technology. It is part of a redefinition of how we as a species transmit knowledge, skills, and values to this generation of people.

Also, through regular updates and any future enhancements the aim will be to try to diversify the area of operation and applicability of the project. Thus, in turn not only providing a technological solution but also creating an ecosystem for technical skill development and future learning.

CHAPTER 7
CONCLUSIONS AND RECOMMENDATION

7.1 Outline

In this segment, the limitations of the project and how to overcome that through some future enhancements are mentioned.

7.2 Limitations / Constraints of the System

The results have shown that the system can provide valid solutions that can be used. However, it does not provide complete automation. There are still scenarios that would require the operator to adjust some entries to create a perfect solution.

The system was also designed to be simple and straightforward. This eliminates any confusion caused by scattered user interface controls and makes usage of the software fully utilized.

1. User Login and Admin access are still not provided, so the students have to enter their choices manually.
2. Constraints such as lunch break and free slots are sometimes not catered effectively.
3. Time Slots are in an interval of half hour each for catering lunch breaks and unavailability, this results in classes being scheduled continuously one after another.
4. The solutions that the system will provide will heavily depend on the running configuration and evaluation matrix. One may find a perfect solution if the application was given enough time and computing power. The complete evaluation for the system will remain hard to solve as the freedom for the configuration of the algorithm has provided a large number of combinations. It can also be inferred that evaluation using other methodologies will yield the same amount of result.

7.3 Future Enhancements

1. Admin access just like VTOP (edit, add, delete, courses, teachers, etc. features)
2. Improved GUI to make it look more accessible and comfortable.
3. **Analytics Integration :** With Analytics, a separate database or result section will be created to successively compare the generated outcomes in each iteration so as to compare the performance and efficiency of the algorithm. It can help to identify ideas/ways to improve the algorithm approach to better cater the needs of the users.
4. **GUI Enhancement / Application Hosting :** The project is still a proof-of-concept mode and is just an algorithmic approach. The GUI of the algorithm can be improved further and the API can be integrated to create a fully functional and responsive standalone web / application.

7.4 Inference

Technology is a tool that can help us; it cannot do the work for us, it cannot think for us, it cannot produce a better world for us. All of this is still very much in the hands and minds of humans.

As in case of any system development process where there are a number of short comings, there have been some shortcomings in the development of this system also. But, working on this project was indeed very fruitful and working on this project gave us an opportunity to explore ourselves and the new frontiers of learning which helped in being a part of the digitalization process.

Hopefully, this will provide a source of information for further research, leading to some more effective ways to proceed with the digitization process.

APPENDIX A

All Codes and related files are uploaded on google drive accessible through given link :-

https://drive.google.com/drive/folders/1HB25_1_BpSjA0z-8JEJGSaY_oTeVRQco?usp=sharing

Project deployment Github link :

https://github.com/AnuragSingh6242/Eazy-Schedule

REFERENCES

1. **D. G. Maere, (2010). "How Working Group Automated Timetabling was founded", retrieved from http://www.asap.ac.nott.ac.uk/, 2010**

2. **Schaerf, A. A Survey of Automated Timetabling. Artificial Intelligence Review 13, 87–127 (1999). https://doi.org/10.1023/A:1006576209967**

3. **Kadam, Vinod & Yadav, Samir. (2016). ACADEMIC TIMETABLE SCHEDULING : REVISITED. international journal of Research In Science & Engineering. April/2016. 417 - 423.**

4. **P. Nandal, A. Satyawali, D. Sachdeva and A. S. Tomar, "Graph Coloring based Scheduling Algorithm to automatically generate College Course Timetable," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 210-214, doi: 10.1109/Confluence51648.2021.9377151.**