Timetable Generator using CSP

Student Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roll Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Computer Science and Engineering

Guru Nanak Dev University

Mini Project Report – B.Tech. (CSE) 3rd Semester

Supervisor: Dr. Harjot Kaur

Date of Submission: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Acknowledgement

I would like to express my deep gratitude to my project supervisor, Dr. Harjot Kaur, for her continuous support, guidance, and encouragement throughout this mini project. Her valuable suggestions and motivation helped me in successfully completing the work on ‘Timetable Generator using CSP’. I am also thankful to my department faculty and friends for their assistance and moral support.

# Certificate

This is to certify that the mini project entitled “Timetable Generator using CSP” has been carried out by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Roll No: \_\_\_\_\_\_\_\_\_\_\_\_) under the supervision of Dr. Harjot Kaur, Department of Computer Science and Engineering, Guru Nanak Dev University, as a part of the partial fulfillment of the requirements for the degree of Bachelor of Technology (Computer Science and Engineering), Semester III.

Supervisor: Dr. Harjot Kaur

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This project, titled 'Timetable Generator using CSP', aims to automate the generation of academic timetables by applying Constraint Satisfaction Problem (CSP) techniques. Manual timetable scheduling often leads to conflicts, inefficiencies, and human errors. This system uses Python’s constraint programming module to assign subjects, teachers, and time slots without clashes. The project’s objective is to minimize human effort, ensure optimal slot utilization, and maintain fairness across teachers. The application includes a user-friendly Graphical User Interface (GUI) built using Tkinter, allowing users to input subjects, teachers, and timings easily. Results show that the system can generate consistent, conflict-free timetables efficiently. This tool can be extended to support multi-department scheduling and integration with institutional databases.

# Introduction

In educational institutions, creating an academic timetable manually is a time-consuming and error-prone task. Administrators must ensure that no two classes or teachers overlap, and that resources are allocated efficiently. This project addresses this challenge using the concept of Constraint Satisfaction Problems (CSP). A CSP is a mathematical problem defined by a set of objects whose state must satisfy several constraints or limitations. By applying this concept to timetable generation, we can automatically assign periods to subjects and teachers while ensuring no scheduling conflicts occur.

The motivation for this project arises from the need to reduce manual workload, eliminate human error, and create a fair and efficient scheduling process. The system also serves as a learning implementation of Artificial Intelligence principles in practical software applications.

# Methodology / System Design

The proposed system utilizes the Constraint Satisfaction Problem (CSP) approach to generate feasible and conflict-free timetables. The Python ‘constraint’ library is used to define and solve constraints between subjects, teachers, and time slots. The system’s design includes three main modules: Input Module, CSP Solver, and Output Display.

The process includes the following steps:

1. 1. Input of subjects, teachers, and number of periods per week.
2. 2. Definition of days and period timings.
3. 3. Application of CSP constraints using AllDifferent and custom teacher conflict constraints.
4. 4. Solving the CSP to assign subjects to available time slots.
5. 5. Displaying the final timetable using Tkinter GUI.

A block diagram of the system is as follows:  
[Placeholder for Block Diagram – System Architecture illustrating Input, Processing, and Output Stages]

# Implementation / Experimental Setup

The implementation was carried out in Python 3.11 using the Tkinter library for GUI and the ‘python-constraint’ module for the CSP logic. The environment used was Visual Studio Code on Windows 10. The implementation involves various input fields for adding subjects, assigning teachers, and setting time slots.

Once all inputs are entered, the system applies constraint-solving algorithms to produce a valid timetable. The GUI displays the timetable in a tabular format with clearly marked periods and breaks.

[Placeholder for code and GUI screenshots here]

# Results and Discussion

The timetable generator successfully produces conflict-free schedules that assign subjects and teachers to appropriate slots without overlaps. Observations indicate that CSP-based scheduling is significantly faster and more accurate than manual methods. The graphical output enhances user interaction and helps visualize the final timetable structure.

Challenges faced during implementation included managing teacher conflicts and ensuring fairness across periods. These were overcome using custom constraint functions and systematic validation.

# Conclusion and Future Work

The project ‘Timetable Generator using CSP’ effectively demonstrates the use of constraint-based programming for solving real-world scheduling problems. It minimizes human effort, reduces conflicts, and improves efficiency in timetable creation. The objectives set for the project were successfully achieved.

Future enhancements may include integration with institutional databases, web-based access, and optimization features for multi-department or multi-class scheduling scenarios.

# References

[1] Python Constraint Library Documentation - https://pypi.org/project/python-constraint/

[2] Tkinter GUI Programming – Python.org Tutorials

[3] Artificial Intelligence: A Modern Approach – Stuart Russell and Peter Norvig

[4] ChatGPT (OpenAI) – Used for drafting and report structuring assistance