Intelligent Systems Fall 2025/2026

Project 1 (2 weeks)

Automated Timetable Generation as a Constraint Satisfaction Problem (CSP)

**Project Overview:**

Timetable generation is a classical **Constraint Satisfaction Problem (CSP)** where the goal is to assign **lectures**, **rooms**, **professors**, and **time slots** in a way that satisfies a set of **hard** and **soft constraints**.

Students will design a **dynamic timetable generator** for their **CSIT department**, using data from an **Excel sheet or a database (e.g., MySQL, SQLite)** as input. The system should automatically construct a feasible timetable that meets all defined constraints.

**Project Objectives:**

1. **Model timetable generation as a CSP problem** — defining variables, domains, and constraints.
2. **Develop a solver** that produces valid timetables satisfying all constraints.
3. **Get you initial data from CSIT time table for Level 1,2,3,4**
4. **Use a dynamic dataset** (database or Excel preferred).
5. **Provide user interfaces** to update data and regenerate timetables.
6. **Evaluate performance** (e.g., number of constraint violations, generation time).

**CSP Formulation:**

**Variables:**

Each variable represents a lecture or class session.

* Example: Lecture(CourseID, SectionID)
* Each variable’s **domain** = available time slots × available rooms × available instructors.

**Domains:**

* **Time slots:** e.g., Mon 9 - 10:30, Mon 10:45 – 12:15, ...
* **Rooms:** Lab1, Lab2, Room101, etc.
* **Instructors:** Faculty members teaching the courses.

**Constraints:**

**Hard Constraints (must not be violated):**

1. No professor can teach more than one class at the same time.
2. No room can host more than one class at the same time.
3. Each course section must have all required lectures per week.
4. Room type must match course type (lab → practical, classroom → lecture).

**Soft Constraints (preferred but flexible):**

1. Avoid gaps for students
2. Avoid early morning or late evening slots.
3. Avoid scheduling the same instructor in consecutive distant rooms.
4. Distribute classes evenly across the week.

**Dataset Description:**

Students can **generate or extract data** from the existing **CSIT department timetable and share between them to find best input format** .

**Data tables (or Excel sheets):**

1. **Courses**: CourseID, CourseName, Credits, Type (Lecture/Lab)
2. **Instructors**: InstructorID, Name, PreferredSlots, QualifiedCourses
3. **Rooms**: RoomID, Type (Lab/Lecture), Capacity
4. **TimeSlots**: Day, StartTime, EndTime
5. **Sections**: SectionID, Semester, StudentCount