COURSE SCHEDULING SYSTEM

Problem statement (one paragraph): A university needs to create a course schedule for the upcoming semester. The schedule must assign courses to specific time slots and classrooms while satisfying various constraints. These constraints include avoiding time conflicts for both students and professors, ensuring classrooms have sufficient capacity for enrolled students, respecting professors' time preferences, and adhering to department-specific requirements (e.g., certain courses must be scheduled on specific days or in particular buildings). The goal is to create an optimal schedule that satisfies all hard constraints and as many soft constraints as possible, maximizing overall satisfaction for students and faculty while efficiently utilizing available resources.

Algorithm Identified (CSP/Searching): For the course timetable scheduling problem, we will employ two search algorithms and one CSP algorithm:

1. Hill Climbing (Local Search):
   - Start with a random schedule, iteratively make small improvements.
   - Move to the neighbor with the highest improvement in satisfied constraints.
   - Use random restarts to escape local optima.
2. Depth-First Search (DFS) with Backtracking:
   - Systematically explore possible schedules, backtracking when conflicts occur.
   - Use forward checking to prune infeasible branches early.
   - Order variables by most constrained first, values by least constraining first.
3. AC-3 Algorithm (Constraint Satisfaction):
   - Use for constraint propagation to reduce variable domains based on binary constraints.
   - Apply as preprocessing and after each assignment in DFS.

References available (Papers/ links):

1. Alvarez-Valdes, R., Crespo, E., & Tamarit, J. M. (2002). Design and implementation of a course scheduling system using Tabu Search. European Journal of Operational Research, 137(3), 512-523.
2. Salamon, A. Z., Jánosi-Rancz, K. T., & Avornicului, M. C. (2015). A constraint satisfaction approach to course timetabling using genetic algorithms. Acta Universitatis Sapientiae, Informatica, 7(2), 168-186.
3. Chen, R. M., & Shih, H. F. (2013). Solving university course timetabling problems using constriction particle swarm optimization with local search. Algorithms, 6(2), 227-244.

Identified Generative AI Tool for Presentation:

The identified generative AI tool is claude AI (the react component). This component displays an overview of the generated schedule, including information such as the assigned courses, classroom capacities, and other relevant metrics. By creating this React component, the user can easily interact with and understand the course scheduling system, making it a valuable tool for presenting the solution to the problem statement.