

college schedule generator

Project Synopsis

Title:

Automated Timetable Generator for Teachers, Students, and Room Availability

Objective:

To develop a scheduling system that generates efficient, conflict-free timetables for teachers and students while ensuring the optimal allocation of available classrooms and labs.

Overview:

Creating timetables for colleges often involves handling multiple constraints, such as teacher availability, student course requirements, and room or lab availability. This project aims to automate the process by designing a tool that simplifies scheduling and ensures effective resource utilization.

Scope:

The system focuses on:

1. Generating schedules for teachers based on their availability and assigned courses.
 2. Providing students with a clear timetable for their registered courses.
 3. Allocating classrooms and labs according to course requirements and room availability.
-

Key Modules:

1. **Teacher Scheduling Module:**
 - Input: Teachers' availability, courses assigned, and preferred time slots.
 - Output: Individual schedules for each teacher showing assigned slots.
2. **Student Timetable Module:**
 - Input: Students' registered courses and batch details.
 - Output: Timetable for each student, ensuring no overlapping classes.
3. **Room and Lab Allocation Module:**

- Input: Classroom/lab availability, capacity, and type (lab, lecture hall, etc.).
 - Output: Allocation of rooms to courses while avoiding conflicts.
 - 4. **Conflict Resolution Module:**
 - Identifies clashes (e.g., overlapping time slots or overbooked rooms).
 - Suggests alternatives to resolve conflicts automatically.
-

Technology Stack:

- **Backend:** Python with libraries like `pandas` for data handling and `ortools` for optimization.
 - **Frontend:** HTML, CSS, JavaScript (for interactive data input and visualization).
 - **Database:** MySQL or PostgreSQL for storing input data (teachers, students, rooms, courses).
 - **Deployment:** Flask/Django for web-based systems or a standalone application using PyQt.
-

Expected Features:

1. **Teacher Timetable:**
 - Personalized schedules for teachers based on their courses and availability.
 2. **Student Timetable:**
 - Organized timetables for students showing their daily schedules.
 3. **Room and Lab Allocation:**
 - Efficiently assigns rooms/labs to courses, ensuring no overlaps.
 4. **Conflict Detection and Resolution:**
 - Automatically resolves scheduling conflicts in real time.
-

Expected Outcomes:

1. Automated generation of timetables for teachers and students.
 2. Efficient allocation of rooms and labs.
 3. Time-saving and reduced manual effort in timetable preparation.
 4. Intuitive visualization of schedules for teachers and students.
-

Future Enhancements:

1. Enable customization for special events (e.g., exams, workshops).
2. Integrate with college management systems for seamless data exchange.
3. Provide analytics for resource utilization and optimization.