

Digitalyz | Software Engineering – Intern

Assignment – Problem Statement

Crestwood College is a lively place where students choose their own courses. Emily, a 2nd year student, wants to join Chorus - High and take English 10 Honors, while Liam hopes to enroll in Band - High and Drawing and Painting. Their course requests are recorded in the Lecture Requests from Students table.

The college's scheduling team, led by Mr. Thompson, must create a schedule that ensures students get their courses without conflicts. To do this, they use data from multiple sources:

- Lecturer Details → Which teacher is assigned to each course.
- Rooms Data → Which classrooms are available and their capacity.
- Courses Data → Which subjects are being offered and when.

For example, Chorus - High is taught by Professor Martinez in Room 129, and Band - High is taught by Professor Lee in Room 133. Both courses have limited slots, so careful scheduling is needed to avoid conflicts.

Long Story Short:

You have a raw data provided by Crestwood where they have:

1. Lecturer Details for available teachers
2. Rooms Details for the college
3. Course List for all the courses taught in that college
4. Requests Table, where students have given their preferences in each row

Dataset with all details: [x dataset.xlsx](#)

Milestones To Cover:

Milestone 1 (BASIC) : This will make your submission eligible

- a. The data in the excel is very raw, clean it up, with proper data structures & nomenclature and make a JSON out of all the data.
- b. Run validations on the cleaned up data & list down insights – what you inferred from the data, are there any unsolvable metrics, what all can you deduce from the data based on the RULES mentioned in RULES sheet (Read this carefully)
 - i. Write it all down in a doc
 - ii. Provide Screenshots, Code & Outputs to support your deduction claims

Milestone 2 (ADVANCED) : Solve this optimally enough and you get a sureshot interview!

Make a schedule for the college assigning the students in sections of the course they have requested: (Follow the RULES in dataset excel sheet)

1. Make a students view schedule after you're done assigning them in appropriate non-conflicting sections - Blockwise
2. Make a teachers view schedule after you're done assigning them in appropriate non-conflicting sections - Blockwise
3. Make a table for resolved & unresolved requests (percentages), Stats on the same & stats Priority wise - Required | Requested | Recommended Requests (Refer to RULES sheet for more info)

Building an algorithm is an open-ended process. You can start by running brute-force experiments to understand how the rules interact. Creating an algorithm from scratch can be challenging, but it's also rewarding when the data gradually takes shape in the way you envisioned.

Some hints which you can start on while making your algo for the **Milestone No. 2**:

- a. Keep things simple — not everything revolves around Deep Learning, write pseudo codes, run experiments with small bits & go bottom up while formulating your algorithm.

[This is what you should ideally try first in this milestone, we will be looking to understand your thought process & how you moved towards formulating your approach]

- b. Since this is a scheduling kind of problem, you can read about Google OR Tools / Gurobi, which are essentially scheduling tools built upon by the companies which lets you model your problem & feed into their model for running iterations. PuLP could also be a solid fit.

With the details, rules, and data you have so far—along with insights from data cleaning and validation—you can now model this problem using these tools and run iterations for scheduling.

[Note that if you feel these tools can help you solve this problem quickly, feel free to skip point 'a' and directly jump to 'b']

PS: For **Milestone 2**, if you've made good progress and are moving toward the goal, feel free to submit your work as it stands—we're happy to review your submissions. It doesn't have to be a fully running solution.

Tip: This is not a Leetcode problem, so don't treat it like one. Don't think of edgecases or 100% satisfaction as the first step.

If you feel that there's a breaking gap in the assignment or a question is unaddressed in the explanation, you can [click here](#) to submit your observation and we'll take action once validated.