

Calendar Management (Assignment 2)

Description

You're working at a school and are tasked with developing a simple calendar app to give the administrative personnel an easier overview of the teachers' schedule. Sometimes a teacher or a room is double booked, and our program should catch this. They've given you a csv file they've pulled from their administrative system.

Before Task

- Read the section, "Stuff to keep in mind".
- Look through the input.csv and see whether the data is usable for your scenario. The administration doesn't always ensure clean data.
- Feel free to clean up the data, but if so should be done programmatically. You will get more of this type of file in the future, so a quick script to clean up the data or making it a part of an integrated API for your upcoming system is also a possibility.
- Determine whether you want the data to be stored in memory("Simple") or in a database("Complex").

Task

- Import input.csv dataset and analyze potential conflicts.
- Create an application with multiple views/pages (for each iteration).
- **HINT:** The word "replicate" will be used in the different iterations, but be smart about it and write reusable code instead 😊

Iteration 1

- Make a list of all the conflicts.(Potentially print out all items and color only the conflicting ones)

Iteration 2

- Create a calendar and insert all items in their respective date column.
- Color conflicting items.
- To not overwhelm the user, you can only show 5-7 days in your calendar at a time. This means pagination must be implemented. Both forward and backward.
- **OPTIONAL :** After reloading the page stay at the last viewed date and don't get forced back to current date.

Iteration 3

- Replicate the calendar from Iteration 2.
- Make every item editable(and delete-able in case of duplicates). When clicking on the item you can edit any attribute you wish.
- After saving, the calendar should update and the edited item should now be updated.
- If the update fixed the conflict, the color of the item should be reset; to show it's not conflicting anymore.

Iteration 4

- Replicate the calendar from Iteration 2 or 3.
- Add a drag n' drop functionality to the calendar.
- When dragged and dropped into another column the date should change to reflect the date in the new column.

Stuff to keep in mind

- There are no time component in the dataset. Therefore assume all the classes in a particular day are being held at the exact same time. If you feel like it, you can update the dataset to contain a time component.
- Setup your code in a repo, and create a new branch before starting so that it's easier to review the code with a PR.
- The application needs to have a visual component. Whether it's written as a web application, mobile application, desktop application or console application(Is it even possible?) doesn't matter. Furthermore the assignment can be solved with any desired language, library and framework.