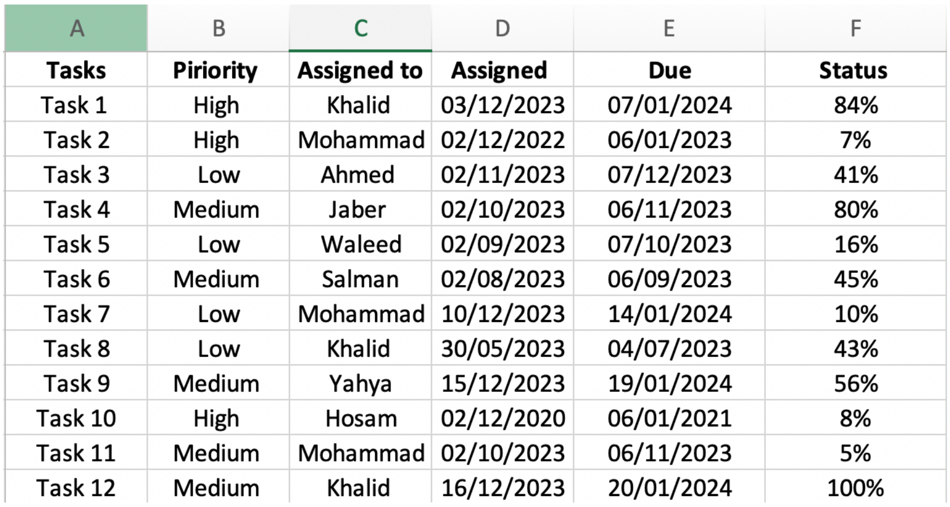
Projects management system
ICS 104 project
Section: 57,66,67,68

Project managers usually use tools such as Microsoft Project to manage their company tasks. However, Microsoft project is very tough to learn and can't do everything, moreover, it's paid. Your boss has asked if your team can save the company money and design a software that can replace Microsoft Project, with only some needed functionalities. The software can work with Excel sheets, in a way that allows the user to load the Excel sheet that has all the project tasks. Here's an example of an Excel sheet your project manager use:



A	B	C	D	E	F
Tasks	**Piriority**	**Assigned to**	**Assigned**	**Due**	**Status**
Task 1	High	Khalid	03/12/2023	07/01/2024	84%
Task 2	High	Mohammad	02/12/2022	06/01/2023	7%
Task 3	Low	Ahmed	02/11/2023	07/12/2023	41%
Task 4	Medium	Jaber	02/10/2023	06/11/2023	80%
Task 5	Low	Waleed	02/09/2023	07/10/2023	16%
Task 6	Medium	Salman	02/08/2023	06/09/2023	45%
Task 7	Low	Mohammad	10/12/2023	14/01/2024	10%
Task 8	Low	Khalid	30/05/2023	04/07/2023	43%
Task 9	Medium	Yahya	15/12/2023	19/01/2024	56%
Task 10	High	Hosam	02/12/2020	06/01/2021	8%
Task 11	Medium	Mohammad	02/10/2023	06/11/2023	5%
Task 12	Medium	Khalid	16/12/2023	20/01/2024	100%

Your Software Should Start by asking the user to enter the path of the Excel sheet, and after that, it will display the main menu.

- **<u>The main menu of our program has these options:</u>**
  1. **Find a task by name:**
     Input**:** The name of the that task
     Output: if the task is found, it will print all the information of that task in this way:

     Task: Task 1
     Priority: High
     Assigned to: Khalid
     Assigned on: 03/12/2023
     Due: 07/01/2024
     Status: 84%

if the task of the given name is not found, then print this message: "The task was not found" and return to the main menu.

3. **Add a new task:**
   First, you will ask the user to enter the following information: **Task Name**, **Priority**, the person who will handle the task (**Assigned to)**, **Date of Assigning**, and the **Due Date**. The **Status will be zero since the task is new**. Check if the task of the exact information exists in the file or not. If it exists, print this message: "This task exists"; otherwise, add the task to the file. Then return to the main menu.

4. **Modify the status of an existing task:**
   Input: **Task Name**
   Look for the task that has the exact given **Task name**. If it does exist, Enter the new **Status** from the user, and modify the task information with the new **Status**. If the task does not exist, print this message "The task doesn't exist; choose option 3 to add a new task". Then return to the main menu.

5. **Remove an existing Task:**
   Input: **Task Name**
   Look for the task that has the exact given **Task name**. If it does exist, Delete the task and its information. If the task does not exist, print this message "The task doesn't exist; choose option 3 to add a new task". Then return to the main menu.

6. **Print all the tasks that were assigned to a specific employee:**
   Input: **Name of an employee**
   Look for the tasks that were assigned to the inputted employee. If there were no tasks, then print: "No tasks were given to 'Employee name' " , otherwise, print all the tasks that were assigned to this employee and all the associated information.

7. **Print all the tasks that have the priority x:**
   Input: **The priority level**

Show all the tasks that have the priority level that was given by the user.

8. **Save the file in the same excel sheet:**
   This option saves the task in the same file. If the file is saved, print this message “The file was saved successfully”, then return to the main menu.
9. **To close the program**

Important Notes:

- You can work in groups; each group has a maximum of two students.
- Groups can be composed of any two students from any of my sections (57/66/67/68)
- To deal with Excel files, the best library to work with is: openpyxl; read the documentation of this library.
- All the dates in the excel sheets are written using DateTime object, use Datetime library to deal with them.
- Try to divide your program into small tasks, each one implemented in a separate function.
- **You should not use any other external libraries.**
- You are limited by the materials that we covered in this course, except for the above-mentioned libraries.
- Global variables are not allowed.
- Your program must not terminate because of an exception; you should handle all the possible exceptions in your program.
- Teammates should contribute equally to this project. If one teammate has less contribution, he will get a less grade.
- The submission of the project is on Saturday, the 6th of May.
- Late submissions will get some deduction.
- Only one group member needs to submit the project, submission can be done only on Blackboard.
- See the Guidelines file for more details regarding the submission

Deliverables:

Each team should submit the following in a zip file:

1. A working code that is written in a **jupyter notebook file**. The file should include the names, ids, and section numbers of team members at the beginning of the code

2. A report as a separate word file that includes the following:

- Description of how the team solved each problem.
- Description of the test cases that have been tested.
- Contribution of each team member.
- Description of the different functions with their tasks.
- Screenshots of the running code. And the test cases that have been tested

Demo and presentation:

Each group is required to present his project:

- The project demos will be scheduled in week 15 (during lab time).
- During the demo, each member is expected to run the program and perform some of its functionalities.
- Team members should be ready to answer any question about their code. So, **make sure you read and understand all parts of the code.**
- **Students who fail to appear for the project demo/presentation will get ZERO in the lab project.**
- Failing to answer questions means losing a major part of the project grade.