Python Web – Final Project

You need to build a web site for A Factory management :

The web site composed from a Flask based server, a MongoDB DB, and HTML pages in the front (you can use native JS or jQuery or any other library)

- The Factory has employees
- Each employee belongs to a department
- Each employee works several shifts.
- Each shift compose from one or more employees
- ONLY registered users can work (log in to) with the web site

Each user has :

- ID (PK, ObjectId)
- Full Name (string)
- Num Of Actions (int)

Each department has

- ID (PK, ObjectId)
- Name (string)
- Manager (reference to emloyee id)

Each employee has:

- ID (PK, Objectid)
- First Name (string)

- Last Name (string)
- Start Work Year (int)
- DepartmentID (reference to department id)

Each Shift has

- ID (PK, ObjectId)
- Date (date time)
- Starting Hour (int)
- Ending Hour (int)

<u>The Web site has the following pages:</u>

Login Page
- Only registered users can work on the system (They must log-in). Use JWT based authentication. Each User login with his username & email – Those credentials are verified against the https://jsonplaceholder.typicode.com/users REST API
- Once a user logged-in , his Full Name will be shown in EVERY page
- Every Page also has a "Log-Out" link which redirect the user back to the Log-In Page
- Each user has a limited actions per day. Each action a user request (like getting the list of employees, create a new department etc..) will reduce from his credit (actions). When he reached to the maximum actions allowed, he will be redirected out of the system for that day.

"Employees" Page

This page has:
- A Table with all Employees data. Each row presents : his FullName, Department, and a list of All his shifts (Date & Time) .
  - o Each Employee Full name is a link. A click on it will redirect to "Edit Employee" Page with all his personal data in a form to be edited.
  - o Each Department name is a link. A click on it will redirect to "Edit Department" Page with all the department data in a form to be edited.
- A button called "New Employee". A click on it will Redirect to "Add Employee" Page
- A drop down with all "Departments". Picking an item in that drop down will filter the employees table and presents only the employees in that chosen department.

"Edit Employee" Page

This page has:
- A Form with the selected employee data to be edited.
- An "Update" button to save the update data in the server
- A "Delete" button to delete all employee's data (and his related data, like shifts) from the server.
- A table with all his Shifts
- An option to register him to an EXISTING shift

"New Employee" Page

This page has:
- An empty Form for creating a new Employee (without shifts) including his department
- A "Save" button for saving the data in the server

- A "Cancel" button for redirecting back to the "Employees" page.

"Department" Page

This page has:
- A Table with all Departments data. Each row presents : Department name, Department manager name, and a list of Employees names work in that department.
  - ○ Each Employee Full name is a link. A click on it will redirect to "Edit Employee" Page with all his personal data in a form to be edited.
  - ○ Each Department name is a link. A click on it will redirect to "Edit Department" Page with all the department data in a form to be edited.
- A button called "New Department". A click on it will Redirect to "Add Department" Page

"Edit Department" Page

This page has:
- A Form with The selected department data to be edited.
- An "Update" button to save the update data in the server
- A "Delete" button to delete all departments data from the server (including all department's employees data)
- A Dropdown with all employees that NOT belong to that department, and a "Add" button. A click on "Add" button will allocate the selected employee to that department (instead of his previous one)

"New Department" Page
This page has:
- An empty Form for creating a new Department (without Employees)
- A "Save" button for saving the data in the server
- A "Cancel" button for redirecting back to the "Departments" page.

"Shifts" Page
This page has an option for creating new shifts, and change existing shifts, and allocate employees to a given shift. Shifts Can NOT be deleted !!

"Users" Page
This page has a table with all users data (can NOT be changed or deleted). Each row presents the user name. maximum actions allowed, and the CURRENT actions allowed today.

System Users:

- Only registered users allow to Login to the web site. ALL users are pre-declared in advanced in a proper DB collection.
- Each "Action" user takes (like watching the "Employees" page, adding department, delete employee ,search by department etc) is logged in a JSON file at the server as follows :
  -

```json
{
    "actions" : [
        {"id":1, "maxActions" : 5, "date" : "1/03/2022", "actionAllowd" : 3 },
        {"id":1, "maxActions" : 5, "date" : "1/03/2022", "actionAllowd" : 2 },
        {"id":1, "maxActions" : 5, "date" : "2/03/2022", "actionAllowd" : 4 },
        {"id":2, "maxActions" : 7, "date" : "2/03/2022", "actionAllowd" : 6 },
        {"id":2, "maxActions" : 7, "date" : "2/03/2022", "actionAllowd" : 5 }
    ]
}
```

- A user can NOT take more actions per day than allowed to. When he reached the maximum actions allowed – He's been logged out until the following day!

Project Delivery:

The project will be delivered (with no due date limitations) by sending a link to your repository (git, google drive..) to the following email address :

Extreme70labs@gmail.com with the subject : "[Your name] – Python Final Project".

Good Luck !!!