INTERNS SUPERVISION SOLUTION (I2S)

YOUR GOAL :

Design and build a JEE Web Application that will help any teacher at EFREI Paris supervise easily the interns assigned to him/her.

REQUIREMENTS

The users of this application are the intern supervisors (on the school side)
Each tutor has a login and a password.
In your prototype be prepared to demonstrate the behavior of your solution with 2 different supervisors (of your choice).

A login screen will of course restrict the access to the application. The following input errors shall be managed with matching blocking error messages: empty field(s) and invalid credentials.

Each tutor only has access to the interns assigned to him/her.
Once logged in and for the duration of the session, the user will see the header of all his pages: [First name] [Last name] | Log out
A click on “Log out” will let the user exit the application.

Listing of all the interns of the current year.
Listing of all the interns of any past academic year (back to 2015). *Hint : see the search options below.*
Show the details of a selected intern (an example is shared in the last section of this document).
Find a way to also manage the following data :
Description of the mission, mid internship meeting info, key words, comments of the intern, comments of the supervisor title of the report, LinkedIn profile and skills.
All fields should be editable.
Search options : per year, per keyword, per name
The navigation must be intuitive.

UNLEASH YOUR CREATIVITY !... BUT SEEK MY AUTHORIZATION FIRST

You may suggest ideas for the layout and design of the pages but two conditions must be met :

a) Your idea must comply with the requirements
b) It must be practical, simple and user friendly

And you always must discuss ideas and suggestions with me first before implementing any feature not explicitly described here.

Please note that I always reward exceptional surprises in terms of architecture, elegant design, and good practices. But again, check first that I validate your ideas.

MAIN SCREEN

If the login is successful, the screen displaying the list of interns supervised by the logged user is showed.

Below is the existing Excel file for doing the same task.
Please design a Web interface that will at least help manage the same level of information.
Keep in mind that more actions should be possible in your solution.

								VISITE								
Gr	NOM	CdC	FICHE VISITE	FICHE EVAL ENTR	SONDAGE WEB	RAPPORT RENDU	SOUT.	PLANIF	FAITE	DEBUT	FIN	ENTR.	MdS	ADRESSE	NOTE TECH	NOTE COM
M2		NON	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M2		NON	OUI	OUI	OUI	OUI	NON	OUI	OUI							
M2		NON	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M2		OUI	OUI	OUI	NON	OUI	OUI	NON	NON							
M2		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M2		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M2		OUI	OUI	NON	OUI	OUI	NON	OUI	OUI							
M2		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M2		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M2		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M1		OUI	OUI	OUI	NON	OUI	OUI	OUI	OUI							
M1		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M1		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							
M1		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI							

GENERAL INSTRUCTIONS

1. Session management :
   - Manage the user's session : the "login" action opens a standard Web session (see info on the 1st page)
2. Architecture, good OO practices, code structure:
   - MVC compliant
   - Loose coupling
   - Separation of concerns
   - As many servlets as you wish
   - Entry point = servlet
   - No default package!
   - No public attributes (unless they are also final)
   - Variables : Declaration -> global / Initialization -> local

3. Documentation : useful comments in your code please !

4. Database:
   - Name : **ST2EEDB**
   - User: **adm**
   - Password: **adm**

5. Technologies and frameworks
   You may of course use all the technologies covered in class.
   Plus, you are strongly encouraged to try the following ones : JSF, EJBs (already used partially with JPA), Filters, Listeners, and annotations.
   You can also integrate other frameworks but I must approve your choice first.
   Design and architectural patterns (MVC of course included) are welcomed.

DELIVERABLES

1) A **report** with :
   a) UML class diagrams of your solution
   b) A link to your remote repositories (GitHub or GitLab or...)
   c) A representation of your data model

2) The **SQL script** to initialize your database

3) A **readme** file with eventual specific instructions about your application (which IDE you used, DB connection info, how to launch the app, tests...). Failure to provide a complete readme file = Penalties.

4) The **source code** of your application : 2 versions
   - Common expectations :
   - DBMS : Java DB or MySQL or MariaDB or PostgreSQL (in any case I expect you to document the DB used)
   - All the JSPs in /WEB-INF
   - Maven project
   - JSP (EL & JSTL)

a) Version 1 *(10 pts)*
- JDBC
- *.properties* file to store the database information

b) Version 2 *(10 pts)*
- JPA
- Quality Assurance :
  - Complete unit testing coverage using Junit & Mockito
  - Static testing using sonarqube
  - Performance testing using JMeter

c) **And… a bonus Version !** (+2 pts). This version is not mandatory and should be done only if you have finished the 2 compulsory versions.
   - RESTful Web services
   - Deployment on a Cloud platform

DEADLINE

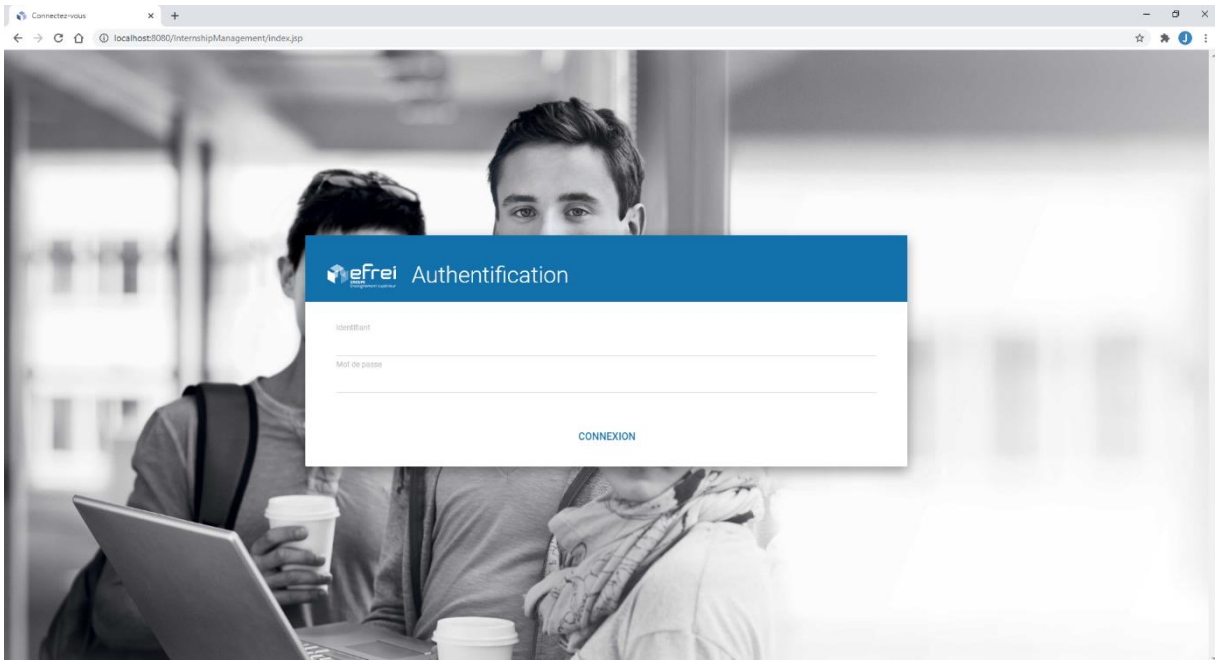
27/11/2020 @ 23:55

PRESENTATIONS !

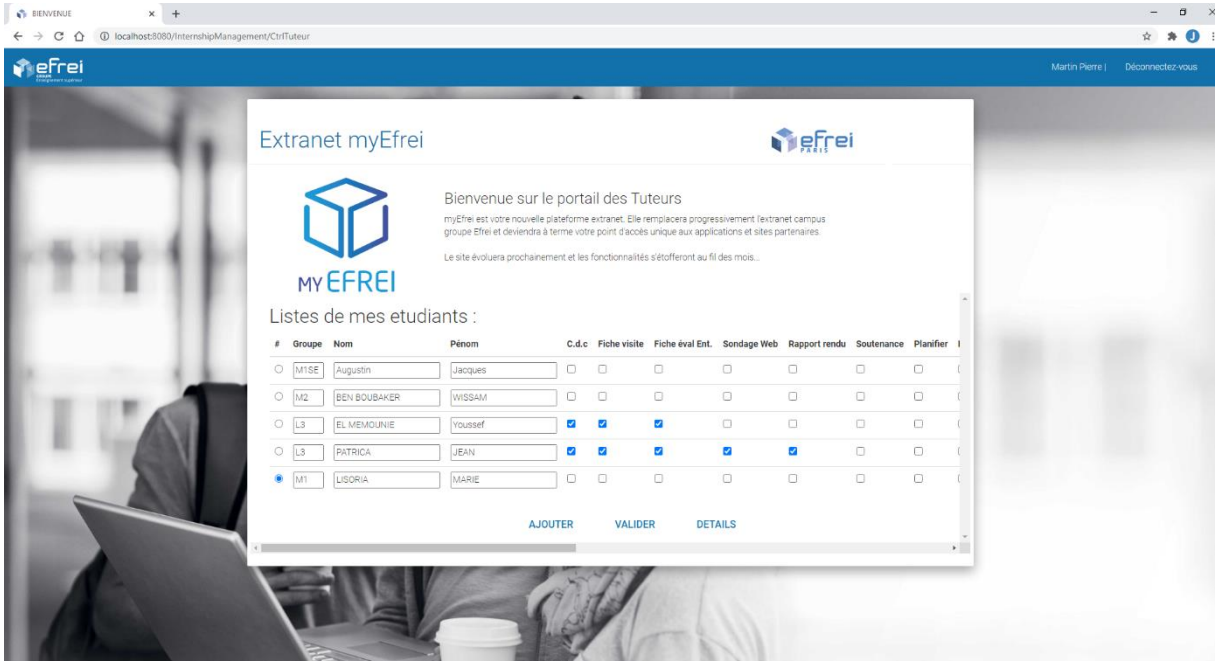
04/12/2020
30 mins for each team.

APPENDIX : FOOD FOR YOUR CREATIVITY

1. Login screen



2. Main screen



3. Details screen