



École Nationale Supérieure
d'Informatique et d'Analyse des Systèmes

Rapport du projet JavaEE

Sujet

*Application web de suivi psychologique
des employés*

Réalisé par :

LAHLALI HATIM
MAHFOUD AYOUB
LEKMAD MOHAMED
EL KHALIFA OUSSAMA

Encadré par :

Mahmoud
ELHAMLAOUI

Année universitaire 2019–2020.

Remerciements

Tout d'abord, nous tenons à remercier toutes celles et tous ceux qui nous ont aidé à réaliser ce travail.

Nous tenons particulièrement à remercier notre professeur, Mr. ELHAMLAOUI MAHMOUD, pour son guide et ses conseils, ainsi que pour son encadrement durant toutes les phases de réalisation de ce projet.

Enfin, nos remerciements s'adressent aux professeurs et au corps administratif de l'Ecole Nationale Supérieure de l'Informatique et de l'Analyse des Systèmes-ENSIAS. Et nous espérons que le présent écrit présentera notre travail décemment et que notre étude sera à la hauteur des espérances.

Résumé

Dans le cadre de notre projet en JavaEE, nous avons réalisé une application web qui permet le suivi de l'état psychologique des employés d'une entreprise. Ce projet consiste à fournir des réponses aux différents employés. Le chef de projet ajoute un questionnaire sous forme de questions et réponses. L'ajout se fait en chargeant un fichier CSV. Puis, l'employé répond au questionnaire et envoie sa réponse.

Enfin, le chef du projet envoie des recommandations à l'utilisateur. Ce présent rapport, résumera le déroulement de toutes les étapes du projet de la conception jusqu'à la réalisation.

Absract

We created a JEE web application that allows the monitoring of the psychological state of the employees of a company.

This project consists of providing answers to the different employees. The project manager adds a test which is added by uploading a CSV file. Then, the employee answers the test and sends his response.

Finally, the project manager sends recommendations to the user. This document will summarize the progress of all steps of the project from conception to implementation.

Table des matières

Introduction générale	3
1 Analyse et conception du projet	4
1.1 Problématique du projet	4
1.2 Objectif du projet	4
1.3 Analyse des besoins	5
1.3.1 Identification des acteurs	5
1.3.2 Les besoins non-fonctionnels	5
1.4 Conception	6
1.4.1 Présentation de la méthode d'analyse	6
1.4.2 Diagramme de cas d'utilisation	6
1.4.3 Diagrammes de classe	8
2 Réalisation	9
2.1 Environnement du travail	10
2.1.1 Langage de programmation "Back end" :	10
2.1.2 Langage de programmation "Front end" :	11
2.1.3 Outils de développement	12
2.1.4 Méthode MVC	13
2.2 Les interfaces graphiques	14
2.2.1 Espace d'authentification	14
2.2.2 Espace chef de projet	16
2.2.3 Espace employé	17
2.2.4 Conclusion	19
2.3 Perspective	20
Conclusion générale	20
Netographie	21

Table des figures

1.1	Diagramme de cas d'utilisation	7
1.2	Diagramme de Classe	8
2.1	Logo Spring Boot	10
2.2	Logo MySQL	10
2.3	Logo React js	11
2.4	Logo Primefaces	11
2.5	Logo VS Code	12
2.6	Logo VS Code	12
2.7	Modèle-Vue-Contrôleur	13
2.8	Interface d'inscription	15
2.9	Interface d'authentification	16
2.10	Interface accueil chef de projet	17
2.11	Interface d'accueil employé	18
2.12	Interface questionnaire	18

Introduction générale

Dans le cadre de notre formation(Cours+TP) en JavaEE à l'ENSIAS,ce projet nous a été proposé dans le but de la réalisation d'une application web qui permet de comprendre l'état psychololgique des employés d'une entreprise à l'aide des tests psychologiques proposés par un psychologue et fournis au chef de projet.

Le présent rapport trace les différentes phases qui ont conduit à l'aboutissement de ce projet, et qu'on peut axer sur deux différents chapitres cohérents et complémentaires.

Le premier concerne la conception du projet. Tandis que le deuxième chapitre est consacré à la réalisation du projet.

Chapitre 1

Analyse et conception du projet

1.1 Problématique du projet

problématique

1.2 Objectif du projet

L'objectif de ce projet est de créer une application en javaEE pour le suivi de l'état psychologique des employés au sein d'une entreprise, suivant l'exemple que notre professeur encadrant a proposé.

1.3 Analyse des besoins

Dans cette partie, nous présenterons les besoins fonctionnels et non-fonctionnels identifiés dans ce projet.

1.3.1 Identification des acteurs

Un acteur est une personne externe ou bien un système qui interagit avec notre système étudié. Dans notre cas, on présente les acteurs suivant :

l'Employé : c'est l'utilisateur qui aura la possibilité de :

- Répondre aux tests proposés par le chef de projet.
- Consulter l'historique des examens passés.
- Consulter l'historique des recommandations proposées.

Le Chef de projet : c'est l'utilisateur qui aura la possibilité de :

- Consulter le profil d'un employé.
- Envoyer un test à tous les employés.
- Récupérer les réponses des employés.
- Envoyer une recommandation aux employés.

1.3.2 Les besoins non-fonctionnels

- **L'ergonomie** : L'application doit être simple et facile à manipuler.
- **Fiabilité** : D'une part, l'application doit fonctionner d'une façon cohérente sans erreur.
- **La Sécurité** : Les comptes utilisés par l'employé et le client doivent être sécurisés pour éviter les faux comptes et les fausses informations. Donc chaque utilisateur doit avoir une adresse email et un mot de passe qui l'identifie de plus seuls l'admin peut ajouter un employé après avoir passé l'étape de recrutement

1.4 Conception

1.4.1 Présentation de la méthode d'analyse

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

1.4.2 Diagramme de cas d'utilisation

L'objectif fondamental de cette étape est d'identifier les principaux cas d'utilisation. Dans notre application, nous avons défini trois acteurs :

- Employé :
 - Répondre aux tests proposés par le chef de projet.
 - Consulter l'historique des examens passés.
 - Consulter l'historique des recommandations proposées.
- Chef de projet :
 - Consulter le profil d'un employé.
 - Envoyer un test à tous les employés.
 - Récupérer les réponses des employés.
 - Envoyer une recommandation aux employés.

Ceci est modélisé par le diagramme de cas d'utilisation suivant :

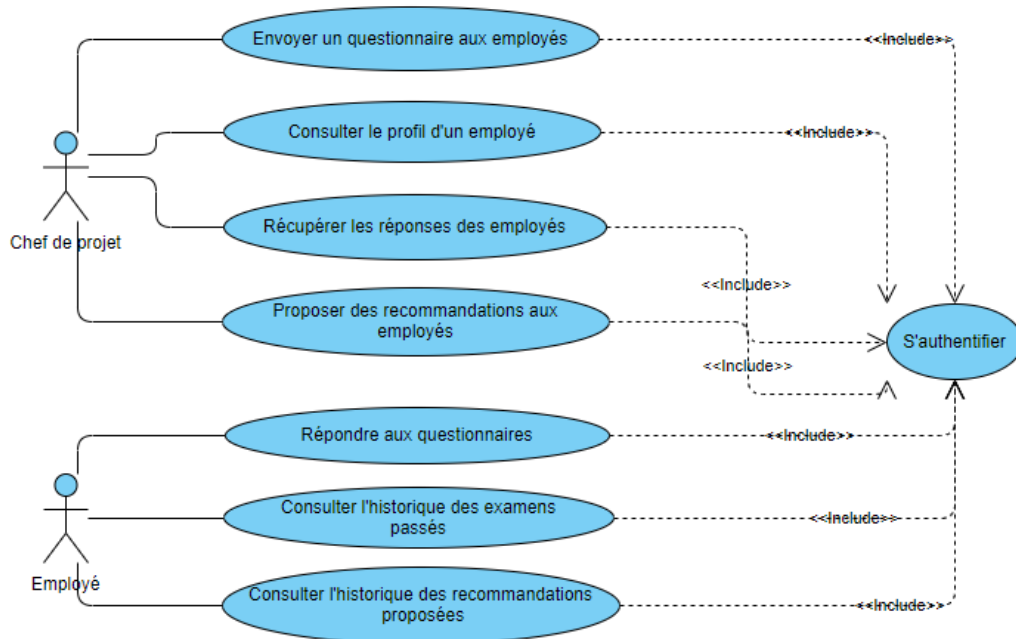


FIGURE 1.1 – Diagramme de cas d'utilisation

1.4.3 Diagrammes de classe

Il représente les classes intervenant dans le système. Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations.

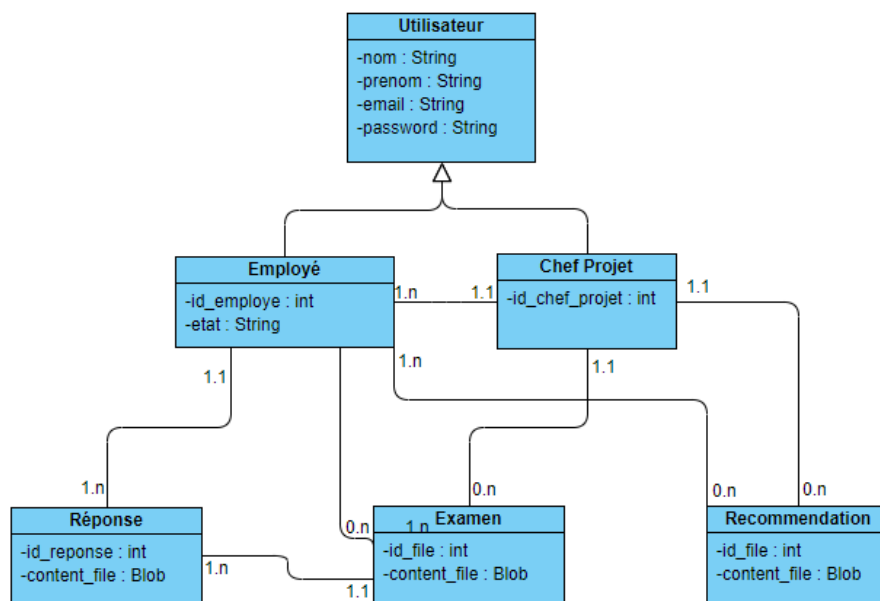


FIGURE 1.2 – Diagramme de classe

Chapitre 2

Réalisation

Après avoir élaboré la conception de notre application, nous abordons tout d'abord une étude technique où nous décrivons les ressources logicielles utilisées dans le développement de notre projet.

Nous présentons en premier lieu notre choix de l'environnement de travail, où nous spécifions l'environnement matériel et logiciel qu'on a utilisé pour réaliser notre application puis nous détaillons l'architecture, aussi nous présentons quelques interfaces réalisées pour illustrer le fonctionnement de quelques activités du système.

2.1 Environnement du travail

2.1.1 Langage de programmation "Back end" :



FIGURE 2.1 – Logo Spring Boot

Spring et Spring Boot Spring est un framework bien connu des développeurs Java pour les nombreuses fonctionnalités qu'il apporte sur les aspects web, sécurité, batch ou encore accès aux données dans le cadre du développement d'une application. Spring Boot est un projet ou un micro framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.



FIGURE 2.2 – Logo MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

2.1.2 Langage de programmation "Front end" :



FIGURE 2.3 – Logo React js

React js est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.



FIGURE 2.4 – Logo Primefaces

Primefaces est une librairie open source de composants graphiques pour les applications JSF (Java Server Faces) créée par la compagnie Turque PrimeTek. Le développement de Primefaces remonte en 2008 et la première version sera publiée Janvier 2009. Il s'agit d'un ensemble de plus de 100 composants graphiques. A part leur beau look and feel, les composants intègrent pleinement les fonctionnalités AJAX et HTML5, sont responsives et compatibles avec la plupart des navigateurs modernes.

2.1.3 Outils de développement



FIGURE 2.5 – Logo VS Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.



FIGURE 2.6 – Logo VS Code

Eclipse est un environnement de développement intégré spécialement conçu pour le langage de programmation Java. Le logiciel est entièrement gratuit, open-source, mais est également extensible. Ainsi, la partie servant à développer en Java n'est qu'une partie des plug-ins qu'utilise Eclipse. En effet, d'autres plug-ins peuvent être utilisés afin de développer tous les langages et tous les formats de fichiers supportés. Il est ainsi possible de programmer en Java, en PHP, en XML, en HTML, en C, ou encore en C++.

2.1.4 Méthode MVC

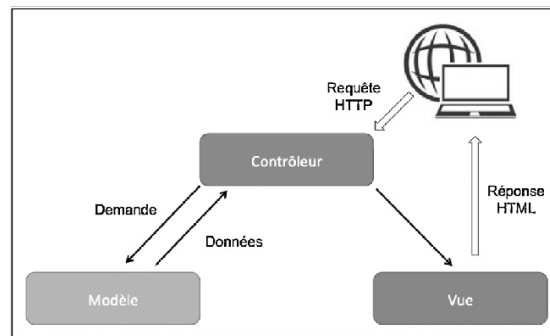


FIGURE 2.7 – Modèle-Vue-Contrôleur

Un des plus célèbres design patterns s'appelle MVC, qui signifie Modèle -Vue-Contrôleur. C'est celui que nous allons découvrir dans cette section du chapitre. Le pattern MVC permet de bien organiser son code source. Il aide à savoir quels fichiers créer, mais surtout à définir leur rôle. Son but est de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts, comme l'explique la description qui suit :

Le modèle représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données, de les convertir selon des concepts chargés de sens pour votre plateforme, tels que le traitement, la validation, l'association et beaucoup d'autres tâches concernant la manipulation des données.

La vue se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples.

Le contrôleur gère les requêtes des utilisateurs. Elle est responsable de retourner une réponse avec l'aide mutuelle des couches Model et Vue. Les Contrôleurs peuvent être imaginés comme des managers qui ont pour mission que toutes les ressources souhaitées pour accomplir une tâche soient déléguées aux travailleurs corrects. Il attend des requêtes des clients, vérifie leur validité selon l'authentification et les règles d'autorisation, délèguent les données récupérées et traitées par le Model, et sélectionne les type de présentation correctes que le client accepte, pour finalement déléguer le processus d'affichage à la couche Vue.

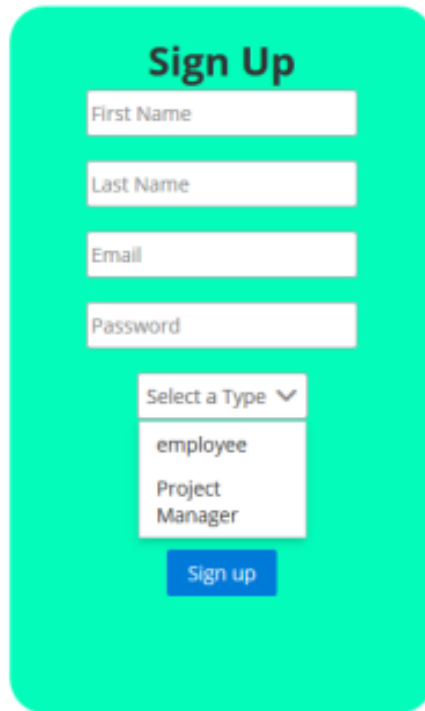
2.2 Les interfaces graphiques

Nous présentons dans cette section quelques interfaces principales de notre réalisation qui illustrent les différents cas d'utilisation.

2.2.1 Espace d'authentification

- *Page d'inscription*

On peut s'inscrire dans l'application en tant que client car l'employé est ajouté par l'admin qui gère tous les ouvriers.



The image shows a 'Sign Up' form interface. It features a light blue rounded rectangle background. At the top, the text 'Sign Up' is displayed in a bold, black, sans-serif font. Below this, there are four white input fields with thin grey borders, each containing a placeholder label: 'First Name', 'Last Name', 'Email', and 'Password'. Under the 'Password' field is a dropdown menu with a light blue background and a downward arrow icon. The dropdown menu is open, showing two options: 'employee' and 'Project Manager'. At the bottom of the form is a solid blue button with the text 'Sign up' in white.

FIGURE 2.8 – Interface d'inscription

- *Page d'authentification*

l'utilisateur s'authentifie en entrant son email et mot de passe pour rejoindre son espace personnel.

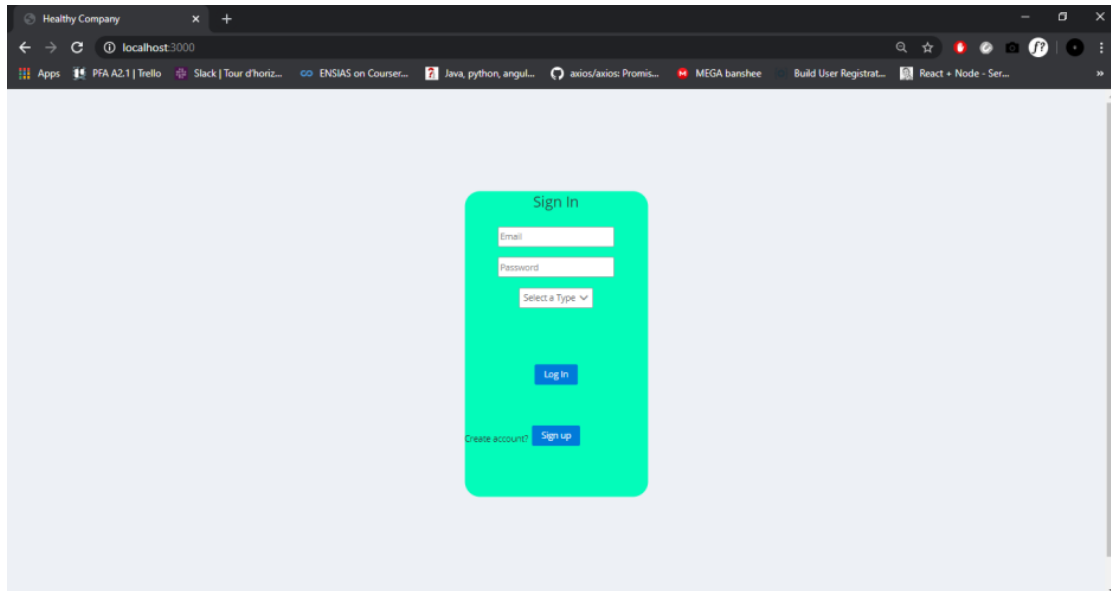


FIGURE 2.9 – Interface d'authentification

2.2.2 Espace chef de projet

- *Page d'accueil*

Le chef de projet peut consulter la liste des employés, peut leur envoyer un examen, comme il peut consulter le profil d'un employé.

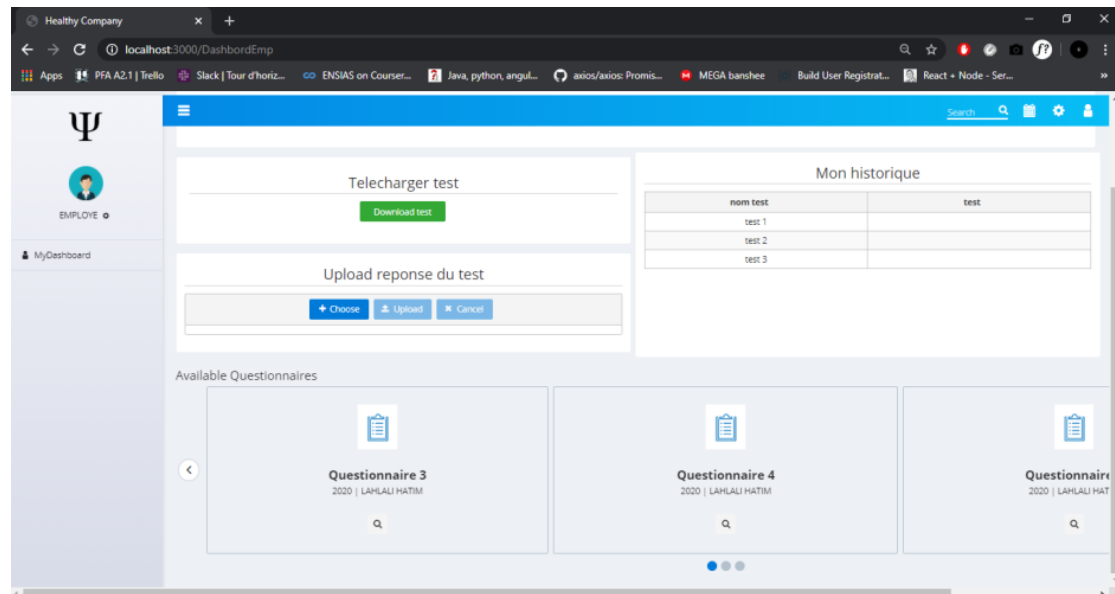


FIGURE 2.10 – Interface accueil chef de projet

2.2.3 Espace employé

- *Page accueil employé*

L'employé peut télécharger l'examen et envoyer sa réponse par la suite, et il peut consulter l'historique des examens passés et des recommandations proposées.

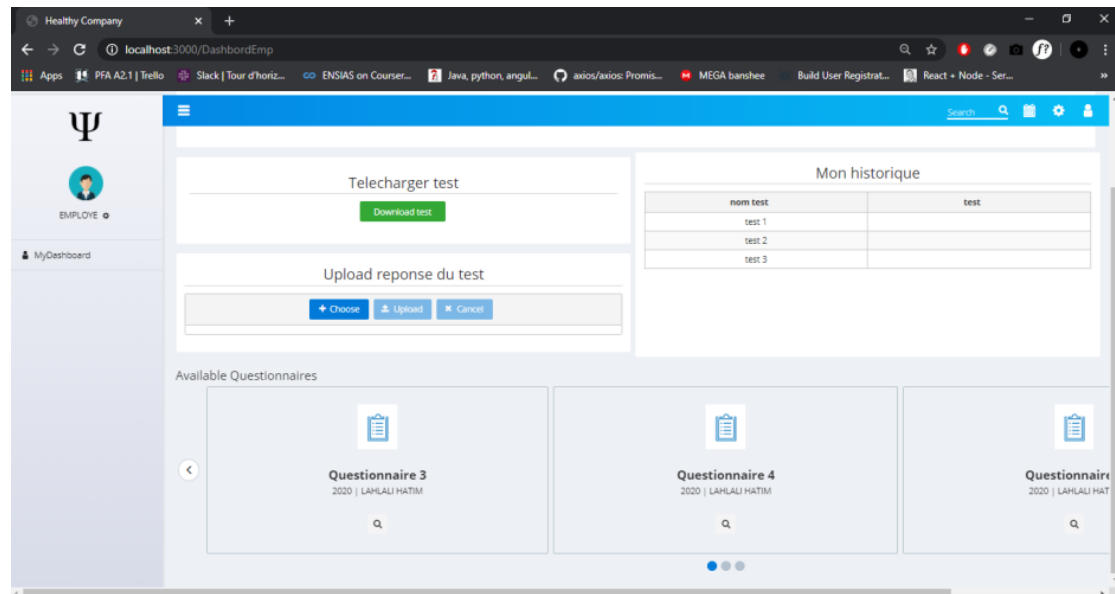


FIGURE 2.11 – Interface d'accueil employé

- *page questionnaire*

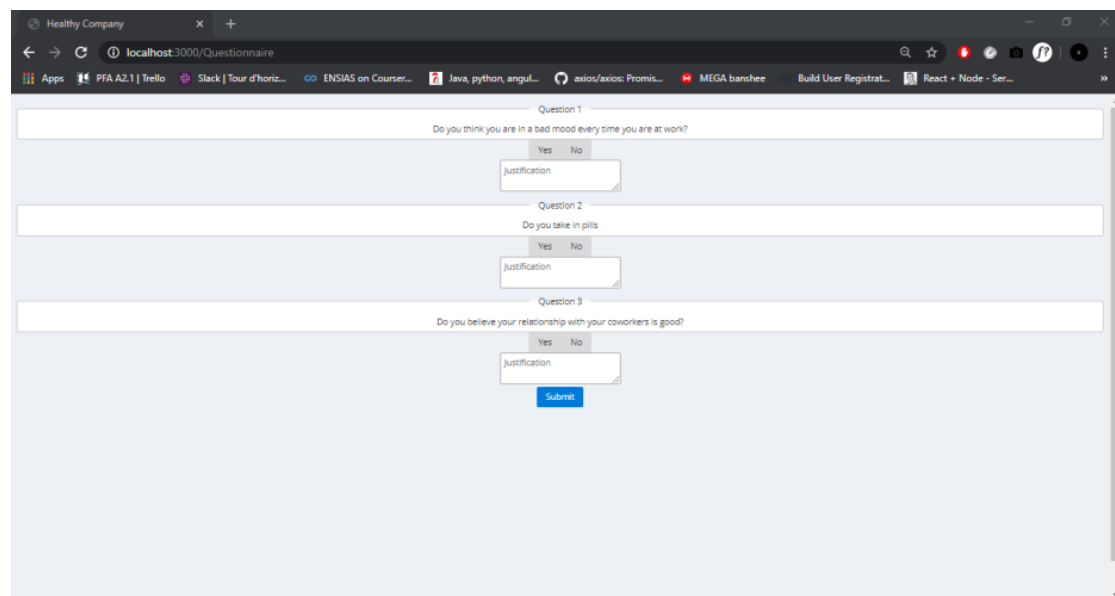


FIGURE 2.12 – Interfac questionnaire

2.2.4 Conclusion

Au cours de ce chapitre, on a présenté en détails la projection de la conception du plan théorique sur le plan pratique. On a montré les interfaces les plus significatives de notre site web.

2.3 Perspective

Conclusion générale

En substance, ce projet de fin d'année consistait à développer une application de gestion de service de proximité. Nous avons présenté tout au long de ce rapport la démarche que nous avons suivie pour mettre au point cette application.

Grâce à cette application on a travaillé les notions suivantes :

- Travailler avec de Nouvelles technologies ; 'react' , 'PrimeReact', 'Spring-Boot'.
- Transmettre les informations entre le serveur et l'application à l'aide de JSON.
- Travailler en groupe 'GIT', 'GitHub', 'Discord'.

Netographie

[1] <https://fr.reactjs.org/docs/getting-started.html>

[2] <https://primefaces.org/primereact/showcase//button>

[3] <https://medium.com/@tariqul.islam.rony/spring-boot-and-multi-stage-dockerized-image-with-mysql-in-docker-compose-part-3-2999b2bdf6aa>

[4] <https://spring.io/projects/spring-boot>

[5] <https://redux.js.org/api/api-reference>

[7] <https://stackoverflow.com/questions/28948945/org-hibernate-exception-genericjdbcexception-could-not-execute-statement-in-hib>

[8] <https://www.youtube.com/watch?v=7giZGFDGnkc>