

Rapport Final

Find Your Father



Matière : Développement Distribué WEB Java EE

Enseignante : Djaouida ZAOUCHE

Sommaire :

Sommaire :	1
I - Manuel d'utilisation :	2
I.1 - Prérequis :.....	2
I.2 - Manuel d'utilisation pour générer le frontend :.....	2
I.3 - Manuel d'utilisation pour générer le backend :.....	3
II - Présentation des pages :	3
II.1 - Présentation générale :.....	3
II.2 - Présentation de la page "Login" :.....	3
II.3 - Présentation de la page "User Profile":.....	4
II.4 - Présentation de la page "Family Profile":.....	5
II.5 - Présentation de la page "DashBoard":.....	5
III - Retour d'expérience :	6

Ce rapport a pour but de présenter l'état d'avancement du site "FindYourFather" qui a pour but de présenter des arbres généalogiques. Tout au long de ce rapport, nous allons vous présenter un manuel d'utilisation, une présentation des différentes pages et un retour d'expérience.

I - Manuel d'utilisation :

Avant de présenter les différentes pages, nous allons vous présenter un manuel d'utilisation pour vous permettre de compiler et lancer l'application spring boot.

I.1 - Prérequis :

Pour lancer l'application, il nous faut quelques logiciels et outils installés sur notre machine. Il nous faut évidemment un "Java Development Kit" (JDK) pour pouvoir écrire du code en Java. Il nous faut "Maven Apache" pour avoir un outil regroupant tous les outils proposés par Spring boot. Il nous faut également "Node JS" et "Angular CLI" pour toute la partie frontend. Enfin, il faut avoir installé "Postgre SQL" pour le système de base de données.

Une fois tous ces logiciels et outils installés, nous pouvons passer à l'étape suivante qui est la génération du frontend qui utilise le framework Angular pour le rendu graphique de l'application.

I.2 - Manuel d'utilisation pour générer le frontend :

Tout d'abord, et comme dit dans le "*read me*", nous devons récupérer le code source de l'application web. En outre, pour cela il faut cloner le répertoire git ou récupérer le code qui vous a été envoyé lors du rendu. Une fois le code source récupéré, il faut entrer dans le répertoire frontend et ouvrir un terminal puis effectuer la commande "*npm install*" dans ce même terminal.

Une fois cette commande exécutée, il faut effectuer la commande "*ng build*" dans le même terminal pour compiler les fichiers Typescript d'angular et donc avoir la partie frontend qui peut s'afficher proprement lors de l'ouverture de l'application web.

Maintenant que nous avons vu la manière dont on génère la partie frontend, pour qu'elle puisse ensuite être lue par spring boot, nous allons maintenant voir la façon dont on peut lancer l'application spring.

I.3 - Manuel d'utilisation pour générer le backend :

Pour lancer l'application spring boot, il faut retourner à la racine du projet et ouvrir à nouveau un terminal, puis, exécuter la commande `"mvnw clean package"`.

Enfin, pour lancer l'exécution de l'application Spring boot, nous lançons la commande `"mvnw spring-boot:run"` ou la commande `"java -jar target\find-your-father.jar"` dans le même terminal que précédemment. A partir du moment où cette commande est exécutée, l'application tourne sur le port 8080 et est accessible à l'adresse `"http://localhost:8080"` ou `"http://127.0.0.1:8080"`.

II - Présentation des pages :

II.1 - Présentation générale :

Lorsque l'on lance l'application, on arrive sur la page `"index.html"` qui est une page de présentation du site `"FindYourFather"`, de son but (comme présenté sur l'image ci-dessous). En haut de la page, nous avons le nom des quatre autres pages présentes sur le site (`"Dashboard"`, `"Family Tree"`, `"User Profile"` et `"Login"`), qui ont chacune un but particulier qui sera présenté dans la suite de ce rapport.



II.2 - Présentation de la page `"Login"` :

La page “Login” est une page, qui comme son nom l’indique, permet de se connecter en temps qu'utilisateur. Elle permet également de s'enregistrer dans la base de données en se créant un compte. De plus, dans le cas où la création d'un utilisateur utilise une personne existant déjà la base de données, en clair la personne qui essaye de créer un compte est déjà présente sur l'arbre d'un autre utilisateur, l'utilisateur peut choisir de devenir cette personne déjà présente. La page “Login” se présente de la sorte.

The screenshot shows the 'Login' and 'Create an Account' forms. The 'Login' form has fields for 'Email' and 'Password', and a 'Login' button. The 'Create an Account' form has fields for 'Email', 'Password', 'Phone Number', and 'Social Security Number'. It also includes a checkbox for 'Are you from another country than France ?' and a 'Login' button.

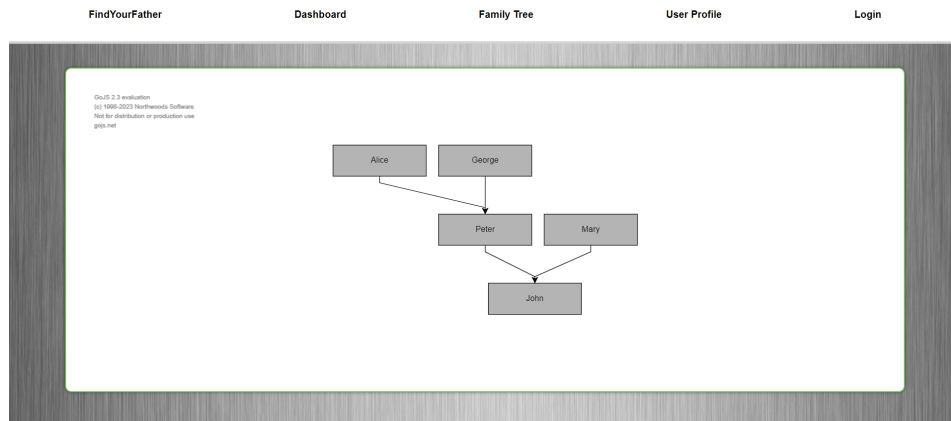
II.3 - Présentation de la page “User Profile”:

La page “User Profile” est une page qui permet l’affichage des informations du profil de l'utilisateur connecté (comme présenté ci-dessous). Grâce à cette page, un utilisateur peut également modifier certaines informations de son profil et avoir une actualisation en direct de cette modification.

The screenshot shows the 'User Profile' form, titled 'Personal Data'. It contains the following fields: 'Social Security Number' (123456789), 'Last Name' (Doe), 'First Name' (John), 'Email' (test@test.com), 'Birthdate' (01/01/1990), 'Nationality' (French), 'Gender', and 'Phone Number' (0null). There are 'Cancel' and 'Save Changes' buttons at the bottom right.

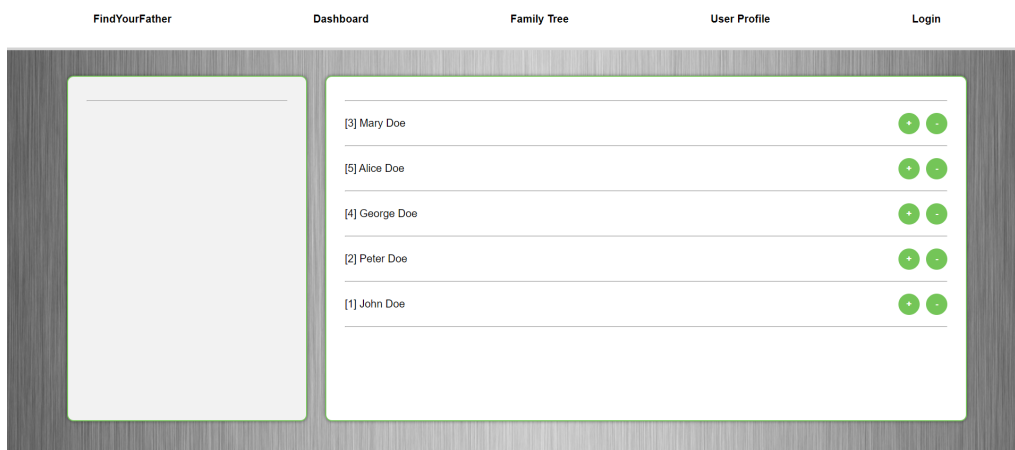
II.4 - Présentation de la page “Family Profile”:

La page “Family Profile” est une page qui permet l’affichage de l’arbre généalogique de l’utilisateur connecté sous forme graphique (via goJS, une bibliothèque Javascript d’affichage de graphes). Elle lance une recherche sur les parents (et l’ascendance) et les enfants (et la descendance) de l’utilisateur connecté et les envoie au frontend qui affiche sous forme de graphe ces informations.



II.5 - Présentation de la page “DashBoard”:

La page “DashBoard” est une page qui permet d’ajouter et de supprimer des relations dans l’arbre généalogique. Ces relations peuvent être des personnes ou des utilisateurs. Dans ce dernier cas, cela envoie une invitation à l’utilisateur avec lequel on souhaite se connecter. De plus, cette page affiche et synthétise les relations déjà existantes (comme présenté ci-dessous)



III - Retour d'expérience :

Ce projet nous a permis de comprendre comment fonctionnaient les arbres généalogiques et a mobilisé nos compétences et notre compréhension de la théorie des graphes qui sera plus pérenne dans le temps.

Toutefois, la représentation que l'on a fait dans la base de données n'était pas optimale. En outre, le fait de stocker l'identifiant de la personne parent (mère ou père) ne l'est pas. On aurait pu représenter une personne et ses parents sous forme de sous-arbres que l'on assemble les uns à la suite des autres pour former l'arbre global. Dans ce cas, nous aurions pu utiliser le pattern Composite, qui malheureusement nous a été enseigné trop tardivement pour l'utiliser dans ce projet.

Ensuite, nous n'avons pas utilisé l'API Restful pour connecter le frontend et le backend, mais nous aurions dû. En effet, cela nous aurait évité d'utiliser des requêtes "fetch" pour que le frontend puisse envoyer et recevoir des données au backend. Dans l'API Restful, ces requêtes sont automatiquement gérées et les données envoyées par le backend sont automatiquement lues par le frontend.

Puis, dans la partie backend, nous aurions pu utiliser des services qui effectuent les calculs que l'on souhaite effectuer plutôt que de les mettre directement dans les contrôleurs et ainsi encapsuler un petit peu plus le code et pouvoir effectuer des tests unitaires sur ces mêmes calculs.

Enfin, je pense que si le projet était à refaire, ce serait avec joie que nous travaillerions dessus en y appliquant les modifications que l'on a identifiées dans les paragraphes précédents.