Test technique Web-atrio

Ce test se compose d'une partie théorique et d'une partie pratique.
Avant de les aborder, voici quelques remarques et consignes :

- **Le test dure 1h30**. La durée des parties est donnée à titre indicatif, pour vous fournir un repère.
  Vous êtes libre de répartir votre temps comme vous le souhaitez, mais **il est important que vous abordiez les 2 parties**, et respectiez la limite de temps.

- **La documentation est autorisée** (recherches internet, cours, ou réutilisation de projet) pour effectuer ce test.
  Je reste également à votre disposition si une clarification de l'énoncé est nécessaire, **ou pour vous débloquer** si vous voyez que vous allez perdre du temps sur un problème.
  N'hésitez pas à me solliciter : je ne pourrai pas vous aider si vous ne me le demandez pas !

- Focalisez vous sur l'essentiel.
  Gardez en tête que votre objectif principal est, dans le temps imparti, de **répondre aux questions et aux besoins demandés**, et idéalement, de finir les 2 exercices.

- Pour rendre votre travail : mettez le code source sur **github** et envoyez le lien à **recrutement@web-atrio.com** **dans la limite du temps imparti**

web·atrio

Partie théorique : XPATH (environ 20 minutes)

Cette partie est **théorique**.
L'objectif est de répondre à des questions sur le langage XPath, en se documentant si nécessaire sur internet. Connaître le langage XPath **n'est pas** un pré-requis.

Remarque *: Cette partie **ne nécessite pas d'exécuter du code**. Essayer de faire fonctionner un code est **fortement déconseillé**, le débuggage devenant rapidement une activité chronophage.*

```
<library>
	<book>
		<title>toto1</title>
		<author>titi</author>
	</book>
	<book type="doc">
		<title>toto2</title>
		<author>titi</author>
	</book>
	<book type="roman">
		<title>toto3</title>
		<author>titi</author>
	</book>
	<book type="bd">
		<title>toto4</title>
		<author>titi2</author>
	</book>
	<library>
		<book type="roman">
			<title>toto5</title>
			<author>titi</author>
		</book>
	</library>
</library>
```

Écrire les chaînes XPath permettant de :

1) Retourner tous les éléments book

2) Retourner tous les éléments title ayant comme parent un élément book avec un attribut type égal à roman

3) Retourner le nombre d'éléments book ayant un attribut type égal à bd

4) Que renvoie la requête XPath suivante : /library/library/ancestor-or-self::library

web·atrio

Partie 2 : Application web (environ 1h10 minutes)

Cette partie est **pratique**.
L'objectif est ici de livrer une application comprenant quelques fonctionnalités basiques, afin de pouvoir **effectuer une démonstration** de son fonctionnement. L'idée est d'utiliser un framework avec un **ORM.**
Si vous optez pour une architecture API REST, dans ce cas pas besoin de réaliser le front, exposer les endpoints (qui pourront être appelables avec Postman) suffira.
---

1) Écrire une classe *Personne* ayant comme attributs : un nom, prénom, date de naissance.

2) Créer une base de données correspondante

3a) Créer les endpoints qui :

- sauvegarde une nouvelle Personne.
    - **attention seule les personnes de moins de 150 ans peuvent être enregistrées sinon renvoyer une erreur**
- renvoie toutes les Personnes enregistrées par ordre alphabétique, ajouter également leur âge actuel.

4a) En option, générez une API DOC.

OU

3b) Créer une page avec un formulaire, contenant :

a. Un champ "*nom*"
b. Un champ "*prénom*"
c. Un champ "*date de naissance*"
d. Un bouton "*valider*", qui remplit la session avec les données du formulaire
    i. **attention seule les personnes de moins de 150 ans peuvent être enregistrées sinon renvoyer une erreur**

4b) Ajouter à cette page un bouton "*visualiser*", qui affiche la liste des personnes en bdd, triées alphabétiquement. Afficher également leur âge actuel.