

## Generale :

### Qu'est-ce que Qualipro ?

Qualipro est un logiciel qui aide les entreprises à gérer la qualité, la sécurité, l'environnement et les risques, tout en respectant les normes comme ISO 9001, ISO 45001, etc.

### Presenter vous ?

Je m'appelle Ramzi Rihane, j'ai 26 ans et je réside à Bizerte. Je suis récemment diplômé d'un Master en Informatique, spécialité Machine Learning et Data Mining. J'ai également obtenu une licence en Informatique de Gestion, spécialité e-Business, qui est l'équivalent du développement web.

Durant mon parcours, j'ai effectué deux stages (PFE et PFA) au cours desquels j'ai commencé à utiliser les technologies Angular et .NET, ainsi que d'autres outils et frameworks. Parallèlement, j'ai réalisé plusieurs projets personnels qui m'ont permis de renforcer mes compétences techniques.

Une fois à l'aise avec ces technologies, j'ai débuté une activité en tant que freelancer, ce qui m'a permis de travailler sur des projets pour des clients en Tunisie et à l'international. Cette expérience m'a aidé à développer mes compétences relationnelles, à communiquer efficacement avec des clients, y compris ceux n'ayant aucune connaissance technique, et à gérer les deadlines avec rigueur et professionnalisme.

Pour améliorer mes capacités de raisonnement, je me suis également exercé à résoudre des problèmes algorithmiques.

Aujourd'hui, je suis à la recherche d'une opportunité professionnelle me permettant d'intégrer une équipe dynamique, afin de continuer à

apprendre, relever de nouveaux défis et contribuer activement aux projets d'entreprise.

## **HTML / CSS / JS :**

### **Q1 : Quelle est la différence entre id et class ?**

**R :** id est unique dans une page, utilisé pour cibler un seul élément. class peut être partagée par plusieurs éléments pour appliquer le même style.

### **Q2 : Qu'est-ce que le Flexbox en CSS ?**

**R :** C'est un système de mise en page en CSS qui permet d'aligner et répartir les éléments facilement, même quand la taille des éléments est inconnue.

### **Q3 : Quelle est la différence entre let, const et var en JavaScript ?**

**R :**

- var : ancienne façon, portée globale ou fonction.
- let : portée bloc, peut changer.
- const : portée bloc, ne peut pas être réassigné.

- **Q4 : C'est quoi une fonction fléchée (arrow function) ?**

**R :** Une syntaxe plus courte pour écrire des fonctions. Elle ne lie pas son propre this.

- `const add = (a, b) => a + b;`

### **Quelle est la différence entre un élément <div> et un élément <span> ?**

- **Réponse :** <div> est un élément de bloc (block-level), utilisé pour structurer des sections dans une page. <span> est un élément en ligne (inline), utilisé pour styliser une partie du texte sans affecter la structure.

### ☐ **Qu'est-ce qu'un élément sémantique en HTML ? Peux-tu donner des exemples ?**

- **Réponse :** Un élément sémantique donne un sens explicite au contenu. Exemples : <article>, <section>, <header>, <footer>, <nav>, <aside>.

### ☐ **Que signifie la balise <meta> et pourquoi est-elle importante ?**

- **Réponse :** La balise <meta> fournit des informations sur la page, comme le charset, les mots-clés ou les descriptions pour le SEO. Exemple : <meta charset="UTF-8">.

**Quelle est la différence entre les attributs id et class ?**

- **Réponse :** id identifie un élément de manière unique, tandis que class peut être utilisée pour appliquer des styles à plusieurs éléments.

**Que fait la balise <iframe> ?**

- **Réponse :** La balise <iframe> permet d'intégrer une autre page HTML dans la page actuelle.

**Qu'est-ce que le Box Model en CSS ? Peux-tu expliquer comment il fonctionne ?**

- **Réponse :** Le Box Model définit la manière dont les éléments sont dimensionnés. Il comprend les marges (margin), les bordures (border), le remplissage (padding), et le contenu (content).

**Quelle est la différence entre position: absolute; et position: relative; ?**

- **Réponse :** position: absolute; place un élément par rapport à son premier ancêtre positionné (autre que static). position: relative; déplace l'élément par rapport à sa position d'origine.

**Peux-tu expliquer les différentes méthodes de positionnement en CSS ?**

- **Réponse :** Les principales valeurs pour position sont :
  - o static (par défaut, position normale)
  - o relative (déplace par rapport à sa position d'origine)
  - o absolute (par rapport à l'élément parent positionné)
  - o fixed (par rapport à la fenêtre du navigateur)
  - o sticky (se fixe après avoir scrollé un certain point)

**Qu'est-ce qu'une pseudo-classe en CSS ? Donne des exemples.**

- **Réponse :** Une pseudo-classe permet de styliser un élément en fonction de son état. Exemples : :hover, :focus, :active.

**Qu'est-ce qu'un media query ?**

- **Réponse :** Un media query permet d'appliquer des styles CSS en fonction des caractéristiques de l'appareil (taille de l'écran, orientation, etc.). Exemple :

```
@media (max-width: 600px) {  
  .menu { display: none; }  
}
```

**Quelle est la différence entre margin et padding ?**

- **Réponse :** margin définit l'espace entre un élément et ses voisins, tandis que padding définit l'espace entre le contenu de l'élément et sa bordure.

**Comment fonctionne la propriété z-index ?**

- **Réponse :** z-index définit l'empilement des éléments. Un élément avec un z-index plus élevé sera affiché par-dessus un élément avec un z-index inférieur.

**Quelle est la différence entre var, let et const ?**

- **Réponse :**
  - o var est fonctionnel et peut être redéclaré.
  - o let est bloqué au niveau du bloc et peut être redéclaré.

- o `const` est bloqué au niveau du bloc et ne peut pas être redéclaré ni réassigné.

#### Qu'est-ce que le hoisting en JavaScript ?

- **Réponse :** Le hoisting fait référence au comportement où les déclarations de variables (avec `var`) et de fonctions sont "remontées" en haut de leur portée.

#### Quelle est la différence entre `null` et `undefined` ?

- **Réponse :**
  - o `undefined` signifie qu'une variable a été déclarée mais n'a pas encore de valeur.
  - o `null` est une valeur assignée explicitement pour indiquer "aucune valeur".

#### Qu'est-ce que le scope en JavaScript ? Quelle est la différence entre scope global et scope local ?

- **Réponse :** Le scope est la portée où une variable est accessible. Un **scope global** est accessible partout, tandis qu'un **scope local** est accessible uniquement à l'intérieur de la fonction ou du bloc où la variable a été déclarée.

#### Comment créer une fonction anonyme ?

- **Réponse :** Une fonction anonyme n'a pas de nom et est souvent utilisée comme argument. Exemple :

```
javascript
CopyEdit
let add = function(a, b) { return a + b; };
```

#### Quelle est la différence entre `==` et `===` ?

- **Réponse :** `==` compare les valeurs après conversion de type, tandis que `===` compare aussi les types de données (sans conversion).

#### Qu'est-ce qu'un callback en JavaScript ?

- **Réponse :** Un callback est une fonction passée en argument d'une autre fonction qui sera exécutée après que la fonction principale ait terminé son exécution.

#### Diff entre Sync et Async ?

##### Synchrone (synchronous)

- Les tâches sont exécutées **une par une**, dans l'ordre.
- Le programme **attend** que chaque tâche se termine **avant de passer à la suivante**.
- **Bloquant** : si une tâche prend du temps, tout le reste est bloqué.

##### Asynchrone (asynchronous)

- Le programme **lance une tâche** et **continue** sans attendre qu'elle soit terminée.
- Quand la tâche est prête, **elle revient** plus tard avec son résultat.

- Non-bloquant.

## **Angular:**

### **C'est quoi un composant dans Angular ?**

- **Réponse :** Un composant est une brique de base dans Angular qui combine un fichier HTML, un fichier CSS et un fichier TypeScript pour définir une partie de l'interface utilisateur.

### **❓ Qu'est-ce que l'injection de dépendances dans Angular ?**

- **Réponse :** L'injection de dépendances est un modèle de conception qui permet d'injecter des services dans les composants ou autres services au moment de leur création.

### **❓ Quelle est la différence entre template-driven forms et reactive forms ?**

- **Réponse :**
  - **Template-driven forms :** géré principalement dans le HTML.
  - **Reactive forms :** géré entièrement dans le code TypeScript, plus flexible et testable.

### **Explique le fonctionnement de ngOnInit() ?**

- **Réponse :** ngOnInit() est une méthode du cycle de vie d'un composant qui est appelée après la création du composant et de ses bindings.

### **Qu'est-ce qu'un service en Angular ? Pourquoi l'utiliser ?**

- **Réponse :** Un service est une classe qui permet de partager des données et de gérer la logique métier, souvent utilisée pour interagir avec une API.

## C'est quoi un module en Angular et pourquoi en avons-nous besoin ?

- **Réponse :** Un module en Angular est une unité de code qui permet de regrouper les composants, directives, pipes et services. Il aide à organiser et à gérer l'application.

## C'est quoi un pipe dans Angular ?

- **Réponse :** Un pipe permet de transformer les données dans un template Angular. Par exemple : `{{ data | date:'short' }}`.

## C'est quoi un routeur dans Angular ? Comment fonctionne-t-il ?

- **Réponse :** Le routeur dans Angular permet de gérer la navigation entre les différentes vues de l'application en fonction de l'URL. Il utilise des routes pour afficher des composants spécifiques.

## Comment Angular gère-t-il les observables avec RxJS ?

- **Réponse :** Angular utilise RxJS pour gérer les données asynchrones avec des observables. Les observables sont utilisés pour les requêtes HTTP, les événements, etc. On peut souscrire (subscribe) à un observable pour obtenir les données.

## Quelle est la différence entre Observables et Promises ?

- **Réponse :**
  - **Observable :** peut émettre plusieurs valeurs au fil du temps et peut être annulé.
  - **Promise :** émet une seule valeur une fois résolue ou rejetée.

## **SQL :**

**1. Comment faire une jointure entre deux tables ?**

```
SELECT commandes.id, utilisateurs.nom  
FROM commandes  
INNER JOIN utilisateurs ON commandes.user_id = utilisateurs.id;
```

**2. C'est quoi une agrégation (COUNT, SUM, AVG, etc.) ?**

```
SELECT COUNT(*) FROM produits WHERE prix > 50;
```

**3. Comment regrouper des données (GROUP BY) ?**

```
SELECT categorie, COUNT(*)  
FROM produits  
GROUP BY categorie;
```

**4. Qu'est-ce que HAVING ? Quelle différence avec WHERE ?**

**5. Qu'est-ce qu'un index et pourquoi l'utiliser ?**

**6. Quelle est la différence entre UNION et UNION ALL ?**

---

### **◆ Exercices classiques posés en entretien**

**1. Trouver les utilisateurs qui n'ont jamais passé de commande**

**2. Lister les produits les plus vendus**

**3.**

**4. Récupérer les 3 derniers utilisateurs inscrits**

```
SELECT * FROM utilisateurs ORDER BY date_inscription DESC LIMIT  
3;
```