


Khaled BENKHALED

30,

 KhaledBenk@gmail.com

 <https://www.khaledbk.com>

 +216-50-203-360

Développeur Web

ÉDUCATION

2012 - 2016 Baccalauréat en informatique

Gatineau - Canada Université du Québec en Outaouais

2009 - 2011 Certificat en informatique de base

Monastir - Tunisie Faculté des sciences de Monastir

2005 - 2009 Baccalauréat en informatique

Kebili - Tunisie École secondaire Ibn Sina à Kebili

EXPÉRIENCES ACADÉMIQUES

2012 – 2013 UQO, Missions et tâches réalisées:

Développement Java: Formulation du problème, conception de solutions, codage de programmes en Java, maîtrise des concepts de programmation **orientée objet**.

Développement Node.js: Conception et implémentation des solutions pour des projets de commerce électronique.

2013 - 2014

Développement / Conception Web: Conception et création de sites Web utilisant **HTML5**, **CSS**, **JAVASCRIPT**, **PHP**, **ASP**, **SQL** et **WordPress** comme système de gestion de contenu. Adobe Dreamweaver, FrontPage et NetBeans: Utilisé pour coder et compiler le code lors de la réalisation de projets Web. **PhpMyAdmin** utilisé pour créer et manipuler la base de données de **MySQL**. Photoshop: utilisé pour le Web et conception graphique. Validation **W3C**. Testez les pages Web dans différents navigateurs: IE6 +, Firefox 2 +, Opera, Safari, Google.

Maîtriser la technologie du commerce électronique: Internet, réseaux, bases de données, serveurs Web, portails de marché, moteurs de recherche. Technologies de programmation Web pour le développement de sites Web transactionnels.

Administration de bases de données: Maîtriser les concepts de bases de données et de systèmes de gestion de bases de données (SGBD). Conception de bases de données relationnelles. Normalisation et administration des bases de données, y compris le traitement et l'optimisation des requêtes.

Jan. 2015

Systèmes d'exploitation *Linux, MacOS, Windows*: maîtriser les fonctionnalités de base des systèmes d'exploitation et leur implémentation . Application aux systèmes d'exploitation MacOS, Linux et Windows.

2013 – 2016

Développement C / C ++: Programmation sur un microcontrôleur et un langage assembleur. Maîtriser la programmation en langage C / C ++ des systèmes informatiques embarqués. Développement et intégration d'applications dans des environnements embarqués. Réalisation d'un projet de système embarqué (projet de synthèse «Réseau des capteurs»). Maîtrise en pratique de la programmation graphique. Techniques d'animation et synthèse d'images 2D / 3D. Concevoir et implémenter des interfaces graphiques.

Janv - Juin 2016

Projet de fin d'études: (Réseau de capteurs): Le projet consiste à mettre en place un réseau de capteurs sans fil, capable de diffuser la température de plusieurs pièces de l'UQO et de stocker les données dans un serveur local de l'université. Ma tâche était de mettre en place un serveur qui contrôlera et traitera les données des capteurs et lui-même chargé de stocker leurs données dans un serveur local à part

EXPÉRIENCES PROFESSIONNELLES

Développeur Full stack

Juin. 2017 - Décembre 2020 (Prosomo Inc)

Ma tâche principale consiste à la création d'un **SaaS** e-commerce (*Frontend*). Y compris une section administrateur (*Backend*) pour gérer, des utilisateurs, des produits, des entreprises et autres paramètres. Les technologies Web utilisés sont: HTML, CSS, JAVASCRIPT (ES6) et

METEORJS, notre framework principal de développement, utilisé pour créer des applications web et mobiles. Avec Meteor JS, j'ai créer des nouvelle sections dans la console d'administration ou dans le frontend, comme des sections de création des utilisateurs, des produits (de différentes type), et de la gestion des factures. Aussi la créations des sections dans les paramètres de l'application comme la gestion des taxes, la gestion des paramètres de la compagnie et la gestion des permissions. J'ai utilisé **REACT-ROUTER** comme une librairie de routage. Pour la création des formulaire j'ai utilisé **VAZCO UNIFORMS** cela me permet de gérer la validation de formulaire pour react (Importation des «schéma» du pont GraphQL et «thème» Uniforms-material).

REACTJS, est la librairie que j'ai utilisé pour créer des interfaces pour de l'applications Web. Aussi, la création des composants globaux et la création des pages par exemple. Au début on a commencé à créer des **class** pour chaque composants dans notre application mais depuis l'été 2020, on a commencé à créer des "functional components" et d'utiliser les hooks pour pouvoir bénéficier d'un état local et d'autres fonctionnalités de React sans avoir à écrire une classe.

REDUX, qui me permet de créer un état global partageable entre les "components". Je peux également créer des méthodes qui nous permettent d'affecter cet état global. J'avais l'habitude de créer des actions, des «fonctions» qui retournent un objet avec le type (type d'action) et les données. Ces actions indiquent quel argument il recevra et ce qu'il retournera. De plus, j'avais l'habitude de créer des réducteurs qui gèrent les actions qui modifient l'état global.

GRAPHQL (APOLLO), j'ai utilisé ce langage de requêtes pour définir les structures des données. J'avais l'habitude de créer des schémas qui représentent les données stockées ou bien à éditer (types et entrées) et des résolveurs (requêtes, mutations et abonnements) pour résoudre les requêtes et de récupérer des données en fonction de la requête client. L'utilisation quotidienne de **GraphiQL** (Playground) me permet toujours de pouvoir tester les requêtes implémentées dans l'application.

Node JS, c'est l'environnement utilisé pour exécuter nos applications. J'ai l'habitude de participer à enrichir l'API de l'application en implémentant des nouvelles fonctionnalités (requêtes par ex.).

MONGODB utilisé comme gestionnaire de bases de données orientées document. J'ai utilisé **Studio 3T** et **NoSQLBooster** en tant que système de gestion de base de données. Cela me permet d'exécuter , d'implémenter et de tester des requêtes avant de les implémenter dans le serveur après.

GIT (GITLAB), notre gestionnaire de projet principal. J'avais l'habitude de faire des "commits" chaque jour et des "push". Aussi je crée des "issues" dans les projets selon les problèmes ou les bugs trouvés dans l'application. La résolution des conflits lors des "merges" aussi était une des tâches que j'avais l'habitude de faire.

GITBOOK, l'outil que j'utilise pour écrire la documentation de l'application, est en anglais. La Documentation est obligatoire pour chaque nouvelle fonctionnalité ou section développées dans l'application.

Pour la communication dans le groupe on utilise "**Google Meet**" (pour les réunions) et aussi j'utilise "**Slack**" pour la communication quotidienne.

Développeur UI

Février. 2021 - En cours (WyPlay)

Ma tâche dans cette position consiste à maintenir le code pour fournisseurs de télévision, en utilisant Javascript(ES6) et Typescript. Dans cette position j'avais l'habitude de participer dans des "daily" vu que **SCRUM** est notre méthodologie de travail, de communiquer quotidiennement avec des équipes multidisciplinaires d'architectes, d'ingénieurs, de designers, de producteurs et de clients. Pour ces projets on utilise des variétés des framework, des langages et des plateformes, comme,

Typescript, est le langage utilisé pour coder ce genre d'applications.

Next.js, pour créer des applications web. Le but d'utiliser Next.js est d'améliorer le processus de développement, et d'améliorer l'indexation de contenu. Avec Next.js, on crée notre interface utilisateur en utilisant **React.js**.

React-Native, pour créer des applications natives qui s'exécute dans des téléviseurs, dans des STBs, (Set Top Box), ou bien dans d'autres cas, sur mobiles, quelque soit sur Android ou bien sur iOS. Les applications communiquent avec un back-end pour assurer l'authentification des utilisateurs et d'autres fonctionnalités comme la lecture des flux directs, etc.

Truffle.js, est un environnement de développement des 'blockchain' et un framework test pour Ethereum. Ce dernier utilise **Ganache** comme un simulateur de 'blockchain'. Avec truffle.js on peut toujours créer des interfaces pour l'application en utilisant **React.js**.

GraphQL, est le langage de requête utilisé pour assurer la communication entre le UI et le backend dans la plupart de nos projets. On avait toujours l'habitude de créer des modules ou des fonctions qui assurent la récupération des données.

Jest, est le framework utilisé pour faire des tests unitaires dans les différents projets. Écrire quotidiennement des tests surtout après l'implémentation des nouvelles fonctionnalités est toujours préférable dans le groupe.

Detox, -

Figma, est l'éditeur graphique et outil de prototypage qu'on se base là-dessous pour créer les interfaces fournies par les designers.

Github, est notre gestionnaire de code. J'avais l'habitude de créer des branches, des issues, des pull requests, faire des revues de code, faire des rebase, des squash et des merges, quotidiennement.

Jira, est l'outil de développement qu'on utilise quotidiennement pour travailler.

Confluence, est l'outil de wiki que j'utilise pour écrire la documentation de l'application. La documentation est obligatoire pour chaque nouvelle fonctionnalité dans les différents projets.

Pour la communication dans le groupe on utilise "**Riot**", "**Google Meet**" (pour les réunions) et aussi j'utilise "**Slack**" pour la communication quotidienne.

Développeur Back-end

Juillet. 2021 - En cours (SK-UN) Bénévole

L'objectif principal de cette expérience est d'aider les gens, dans des régions isolées à travers le monde, à accéder à certains besoins de base. Dans cette position j'avais l'habitude de créer des

Application Open Source et des APIs pour des applications web, en utilisant des technologies web comme,

GraphQL, le langage de requêtes utilisé dans le serveur avec lequel les applications communiquent.

Ce dernier utilise Utilise **Prisma**, comme un ORM (**Object-Relational-Mapper**), qui permet de construire rapidement des requêtes, et à faire moins d'erreurs.

Typescript, est le langage de programmation utilisé dans l'implémentation des fonctionnalités.

PostgreSQL, est le SGBD utilisé pour stocker et gérer les données des utilisateurs dans ces types d'applications. J'avais l'habitude de faire des requêtes et de manipuler les tables en utilisant, **PgAdmin**.

Jira, est le logiciel de gestion des tâches qu'on utilise.

Confluence, est l'outil de documentation et pour les présentation au beneficiaires

La communication avec le groupe des designers , ingénieurs et les architectes se fait à travers Microsoft **Teams**.

Gitlab, est l'outil de gestion de code qu'on utilise. J'ai l'habitude de faire des merges, des rebase, des revues de code et la création des issues et des branches.

Docker,

COMPÉTENCES SOCIALES

- Grande capacité de communication et d'intégration
- Facile à apprendre de nouvelles connaissances en particulier les langages de programmation informatique.
- Grand sens de l'organisation, des responsabilités et du professionnalisme.
- Performance en équipe ou travail individuel.

Passe-temps

Bénévolat, jeux-vidéos, voyage, sport, ...

Langues

- **Français**: lu, écrit et parlé
- **Anglais**: lu, écrit et parlé
- **Arabe**: lu, écrit et parlé (langue maternelle)

Référence

Références disponibles sur demande.

