

Aaron Muganda

azmuganda@gmail.com | (613) 304 – 1227 | Gatineau, Québec, J8Y 4B1 | [LinkedIn](#) | [Portfolio](#) | [Github](#)

Profil

- B.Sc.A. en génie logiciel (COOP), option gestion et entrepreneuriat en ingénierie
- Expérience en développement logiciel avec C, C++, Java, Kotlin, Python
- Expérience en développement web avec CSS, HTML5, JavaScript et TypeScript
- Expérience dans le développement d'applications Android en Java avec Android Studio et IntelliJ IDEA
- Expérience en développement Front-end avec Angular, Bootstrap, Tailwind CSS, Next.js, React.js et Vue.js
- Expérience en développement Back-end avec Java, Kotlin, Spring Boot, Express.js, Nest.js et Node.js
- Expérience avec les API REST
- Expérience avec le langage de requête SQL
- Expérience avec les SGBD relationnelles (MySQL, PostgreSQL) et les SGBD non relationnelles (MongoDB, Firebase)
- Expérience avec les outils de test d'API Insomnia et Postman
- Expérience avec Docker et Kubernetes pour la conteneurisation et le déploiement
- Expérience avec RabbitMQ pour la messagerie et les files d'attente
- Expérience avec les IDEs Visual Studio, VS Code, Android Studio, CLion, Code::Blocks, Eclipse, IntelliJ IDEA, WebStorm et Xcode
- Expérience des protocoles HTTP et HTTPS pour le développement et le test d'applications web
- Maîtrise des tests unitaires et d'intégration avec JUnit
- Utilisation de Cucumber pour l'automatisation de scénarios de tests en BDD (Behavior-Driven Development)
- Rédaction et gestion de configurations avec YAML
- Documentation et conception d'API avec Swagger/OpenAPI
- Mise en place de mécanismes de sécurité incluant JWT (JSON Web Tokens) pour l'authentification et l'autorisation
- Utilisation de cURL pour tester, consommer et déboguer des API
- Expérience en automatisation des tests et intégration continue (CI/CD)
- Expérience en tests & en assurance qualité logicielle
- Expérience avec les méthodologies Agile et Cascade (Waterfall)
- Expérience avec les cadres AgilePM, Scrum, Kanban, Scrumban
- Expérience avec les approches DevOps et DevSecOps
- Expérience avec les outils Asana, Azure DevOps, Confluence, Jira, Planner, Microsoft Project et Wrike
- Excellent communicateur à l'oral et à l'écrit en français et en anglais

Études

Baccalauréat ès Sciences appliquées en Génie logiciel (COOP), option en Gestion et entrepreneuriat en ingénierie

Université d'Ottawa, Ottawa, Ontario

Septembre 2021 – Juin 2026

- Bourse au mérite : Automne 2024
- Palmarès du doyen : Automne 2024

Compétences techniques & Outils

- Outils de gestion de projet : Asana, Azure DevOps, Confluence, Jira, Planner, Microsoft Project et Wrike
- IDE : Visual Studio, VS Code, Android Studio, CLion, Code::Blocks, Eclipse, IntelliJ IDEA, WebStorm et Xcode
- Développement logiciel : C, C++, Java, Python
- Développement web : CSS, HTML, JavaScript et TypeScript
- Développement Front-end : Angular, Bootstrap, Tailwind CSS, Next.js, React.js et Vue.js
- Développement Back-end : Java, Kotlin, Spring Boot, Express.js, Nest.js et Node.js
- Paradigmes : Programmation orientée objet, conception orientée objet, principes SOLID, design patterns, Domain Driven Design et Test Driven Design
- Langage de requête : SQL
- Bases de données : SQL (MySQL, PostgreSQL), NoSQL (MongoDB, Firebase)
- Tests & Qualité : JUnit, Cucumber, tests unitaires, tests d'acceptation, tests utilisateurs
- API & Intégration : Swagger, YAML, cURL
- Protocoles : HTTP/HTTPS
- Sécurité : Authentification & autorisation, JWT, tokenisation
- Conteneurisation & Orchestration : Docker et Kubernetes
- Outils de test et d'intégration d'API : Postman et Insomnia
- Messagerie et files d'attente : RabbitMQ
- Microsoft Office 365 & Microsoft Power Plateforme
- Contrôle de version : Git / GitHub
- Documentation : Confluence

Projets

Mealer App | Java, Firebase, Android Studio, GitHub

- Configurer le projet Android Studio avec Firebase
- Implémenter les fonctionnalités de découverte des repas en affichant les données récupérées depuis Firebase
- Ajouter des validations pour gérer les divisions par zéro et retourner une réponse appropriée en cas d'erreur
- Ajouter un système de commande permettant aux utilisateurs de sélectionner et commander des repas
- Tester l'application pour valider les fonctionnalités

[Web Services Data Rest](#) | **Kotlin, Spring Boot, Angular, HTML, CSS, JavaScript, TypeScript, MySQL, VS Code, Postman, GitHub**

- Configurer un projet Spring Boot avec Spring HATEOAS, Spring Data JPA, MySQL, et SpringDoc pour OpenAPI
- Créer l'entité Author et ses relations
- Implémenter des repositories pour les opérations CRUD sur les entités
- Développer un contrôleur REST pour gérer les opérations CRUD sur les livres et auteurs
- Ajouter des hyperliens dans les réponses API à l'aide de HATEOAS
- Configurer le module HttpClient pour communiquer avec l'API REST
- Créer un service Angular pour gérer les appels API (ajout, récupération, mise à jour des livres et auteurs)
- Ajouter une option de menu « Authors » à la barre de navigation
- Créer un formulaire pour rechercher un auteur par ID et afficher ses informations
- Implémenter des fonctionnalités pour associer des auteurs aux livres

[Web Services GraphQL](#) | **Kotlin, Spring Boot, Angular, HTML, CSS, TypeScript, NoSQL, MongoDB, VS Code, Postman, GitHub**

- Ajouter un schéma GraphQL (employee.graphqls) avec des requêtes et mutations pour gérer les employés
- Implémenter les résolveurs (EmployeesResolver) pour traiter les requêtes et mutations du serveur GraphQL
- Configurer MongoDB pour stocker les données et valider le serveur avec GraphiQL
- Mettre à jour le service Angular pour interagir avec l'API GraphQL (requêtes et mutations)
- Tester et valider la communication entre le serveur GraphQL et l'application Angular
- Modifier le formulaire d'entrée de données d'employés pour consommer les mutations de l'API
- Intégrer Apollo Angular pour gérer les requêtes et mutations vers le serveur GraphQL

[Console Based Card Game](#) | **C++, Visual Studio, VS Code, GitHub**

- Créer les classes principales du jeu (Card, Deck, DiscardPile, Hand, Chain, TradeArea, Player, Table, CardFactory)
- Gérer les exceptions pour les types illégaux (IllegalType) et les fonds insuffisants (NotEnoughCoins)
- Implémenter la logique du jeu dans une boucle principale en suivant le pseudo-code fourni
- Implémenter une fonctionnalité de sauvegarde et de recharge de l'état du jeu pour permettre une pause

[Web Services RESTful](#) | **Kotlin, Spring Boot, VS Code, Postman, GitHub**

- Configurer un projet Spring Boot pour implémenter un service web RESTful de calculatrice
- Créer des endpoints REST pour les opérations arithmétiques de base (addition, soustraction, multiplication, division)
- Implémenter chaque opération en tant que requête GET avec une route dédiée (ex. : /calculator/add/{num1}/{num2})
- Gérer les cas particuliers, comme la division par zéro, en renvoyant des réponses appropriées
- Générer la documentation OpenAPI pour décrire les endpoints et faciliter leur utilisation par les clients
- Tester les endpoints avec Postman pour valider les résultats et le bon fonctionnement du service
- Publier le projet sur GitHub pour assurer la gestion de version et la collaboration

[Calculator Angular](#) | **Angular, HTML, CSS, TypeScript, VS Code, GitHub**

- Développer une application Angular de calculatrice permettant à l'utilisateur de saisir deux nombres et de sélectionner une opération arithmétique (addition, soustraction, multiplication, division)
- Implémenter les composants Angular nécessaires pour gérer l'interface et la logique métier
- Utiliser TypeScript pour la gestion des calculs et le traitement des données saisies par l'utilisateur
- Afficher le résultat de manière dynamique dans une interface conviviale
- Réaliser des tests unitaires avec Jasmine pour valider les fonctionnalités et assurer la fiabilité du code
- Publier le projet sur GitHub pour le suivi et la gestion de versions

[Calculator Spring](#) | **Kotlin, Spring Boot, HTML, CSS, VS Code, GitHub**

- Re-développer la calculatrice en tant qu'application Spring MVC avec Thymeleaf pour le rendu dynamique des pages HTML
- Implémenter un contrôleur Spring pour traiter les opérations arithmétiques côté serveur
- Gérer l'entrée des données utilisateur via un formulaire HTML connecté au back-end
- Utiliser Thymeleaf pour afficher les résultats des calculs sur une page web dynamique
- Configurer et exécuter l'application avec Spring Boot pour simplifier le déploiement et la maintenance
- Héberger le projet sur GitHub pour assurer le suivi de développement et la collaboration

[Angular Web Form](#) | **Angular, HTML, CSS, TypeScript, VS Code, GitHub**

- Concevoir une application Angular avec un formulaire réactif pour collecter les informations utilisateur (prénom, nom, téléphone, email)
- Mettre en place une validation des champs avec affichage de messages d'erreurs en cas de saisies invalides
- Développer une navigation permettant de rediriger l'utilisateur vers une nouvelle page après la soumission réussie du formulaire
- Afficher les données saisies dans un tableau dynamique sur la nouvelle page
- Utiliser TypeScript et Angular pour gérer les données et assurer la fiabilité du formulaire
- Versionner et publier le projet sur GitHub

[Shopping List](#) | **Angular, HTML, CSS, TypeScript, VS Code, GitHub**

- Développer une application Angular permettant la création et la gestion d'une liste de courses
- Implémenter un composant pour la saisie des articles et l'ajout à la liste via un bouton « Add »
- Créer un second composant pour afficher la liste des articles ajoutés avec un bouton « Delete » associé à chaque élément
- Gérer la suppression dynamique des éléments pour maintenir une liste à jour
- Utiliser TypeScript pour structurer la logique applicative et manipuler les données
- Mettre en ligne et versionner le projet sur GitHub

[e-Hotels](#) | **HTML, CSS, JavaScript, Node.js, PostgreSQL/MySQL, SQL, GitHub**

- Concevoir et développer une plateforme web centralisée pour permettre à cinq chaînes hôtelières d'Amérique du Nord de collaborer et de partager leurs disponibilités
- Créer une interface utilisateur en HTML, CSS et JavaScript permettant aux clients de consulter les chambres disponibles en temps réel
- Implémenter un back-end en Node.js pour gérer la logique métier et la communication entre la base de données et l'interface utilisateur
- Développer une base de données relationnelle (PostgreSQL/MySQL) pour stocker les informations sur les hôtels, les chambres et les réservations
- Mettre en place des requêtes SQL pour gérer les opérations de consultation, d'insertion et de mise à jour des réservations
- Implémenter un système de réservation permettant aux clients de sélectionner une chambre et de finaliser une réservation en ligne
- Gérer la synchronisation des données entre plusieurs chaînes hôtelières afin d'assurer la cohérence et la disponibilité en temps réel
- Héberger et gérer le code source sur GitHub pour assurer le suivi du développement et la collaboration en équipe