

| Sanjeevi    Ingénieur logiciel Détails Courrier électronique : [sanjeevimani567@gmail.com](mailto:sanjeevimani567@gmail.com)  Téléphone : +91 8973386842  Lieu : Salem Site Web et réseaux sociaux [**Site Web personnel**](https://sanjuvi.github.io/Blog/) 🌐  [Github](http://github.com/Sanjuvi/)  [LinkedIn](https://www.linkedin.com/in/sanjeevi-%E0%AE%9A%E0%AE%9E%E0%AF%8D%E0%AE%9A%E0%AF%80%E0%AE%B5%E0%AE%BF-01a7b813b/) Langues  | Tamoul | | | --- | --- | |  |  |  | Anglais | | | --- | --- | |  |  | | Profil En tant qu'ingénieur logiciel, mon objectif est d'utiliser mon expertise et mes compétences pour concevoir et développer des solutions logicielles innovantes qui répondent aux besoins des clients. Je me consacre à la création de programmes de haute qualité, évolutifs et efficaces qui optimisent les performances, améliorent l'expérience utilisateur et stimulent la croissance de l'entreprise. Passionné par la résolution de problèmes, soucieux du détail et adoptant une approche collaborative, je m'engage à obtenir des résultats exceptionnels et à dépasser les attentes. Éducation Collège gouvernemental de technologie de Coimbatore  Informatique et génie - CGPA 7.06 Projets AWS IAM CLI (gestion des identités et des accès)  Développement d'un outil CLI convivial dans Rust pour gérer la gestion des identités et des accès sur le cloud AWS.  - A utilisé des kits SDK AWS pour rationaliser les interactions avec les services AWS.  - [Lien du projet : (Fournissez un lien vers le référentiel du projet)]  - [Lien du billet de blog : (Fournissez un lien vers le billet de blog technique)] Compétences - Langages de programmation : Rust, Python, Java  - Services AWS : IAM, S3, RDS, Rekognition, Polly, Transcribe  - Bibliothèques et SDK Rust  - Développement de CLI  - Documentation technique  - Résolution de problèmes et débogage Réalisation technique  * Pendant que j'apprenais Rust, j'ai commencé à prendre conscience de l'importance du système de type statique et du risque d'une gestion manuelle explicite de la mémoire qui entraîne un danger pour la mémoire en C/C++. * L'écriture d'un code dans un paradigme différent, comme une méthode fonctionnelle ou orientée type, réduit les bogues et augmente la confiance et l'exactitude de notre intention. * [J'ai publié des articles de blog sur la [simultanéité](https://sanjuvi.github.io/Blog/posts/Concurrency-In-Rust/), l'insécurité de [la mémoire, la](https://sanjuvi.github.io/Blog/posts/Memory-Unsafety/) [gestion des erreurs](https://sanjuvi.github.io/Blog/posts/Robust-Error-Handling-Strategies/), [l'assemblage Web](https://sanjuvi.github.io/Blog/posts/Webassembly/), etc.](https://sanjuvi.github.io/Blog/) * L'un de mes articles de blog sur [les génériques dans Rust](https://sanjuvi.github.io/Blog/posts/Rust%20Generics/) a été publié dans le bulletin d'information [This week in Rust](https://www.linkedin.com/posts/sanjeevi-m-01a7b813b_one-of-my-blog-posts-was-suggested-in-this-activity-7057676034573303809-txhJ/?utm_source=share&utm_medium=member_android).   Intérêt   * Lire des articles et des articles de blog sur Rust et d'autres domaines techniques connexes. * Ne cessez jamais de vous renseigner sur la technologie et sur des sujets divers. |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |