

Projet d'algorithmique et programmation assisté par IA

François Delbot

February 15, 2024

1 Le travail demandé

1.1 Objectifs

Ce projet est conçu pour équiper les participants avec des compétences diverses et essentielles requises dans le développement logiciel moderne. Les objectifs spécifiques comprennent :

1. **Compréhension approfondie des langages de programmation :** Manipuler des structures de données avancées, utilisation de différents langages de programmation, communication entre différentes parties d'un projet.
2. **Adoption de pratiques de développement collaboratif :** Utiliser GitHub pour encourager la révision de code, le partage de connaissances, et une méthodologie de construction collective, essentielle pour le développement collaboratif moderne.
3. **Développement de nouvelles compétences :** Créer un serveur web, utilisation d'outils collaboratifs, de structures de données avancées, apprentissage d'un nouveau langage de programmation, utilisation et implémentation d'un algorithme approprié ... Tout cela est rendu beaucoup plus simple grâce aux outils comme ChatGPT ou Bard. Vous devez utiliser ces outils dans la réalisation de votre projet.
4. **Pratique de la gestion d'APIs :** Développer et consommer des APIs, en mettant l'accent sur le format JSON pour l'échange de données, standard de l'industrie pour la communication inter-systèmes.
5. **Préparation pour l'innovation continue :** Établir une base solide dans diverses compétences clés, préparant les participants à s'adapter et à innover dans leurs futurs projets technologiques.
6. **Travail collaboratif :** Travailler en équipe est une tâche ardue. Cela permettra de développer vos compétences en communication, planification, gestion des conflits etc.

1.2 Consignes du Projet

Afin d'assurer une expérience d'apprentissage cohérente et productive, les participants doivent respecter les consignes suivantes pour le déroulement du projet :

- **Travail en trinôme :** Les étudiants doivent travailler en groupes de trois. Cette structure favorise la collaboration, permet un échange d'idées diverses et encourage la résolution de problèmes en équipe. Chaque membre du groupe est attendu pour contribuer activement au projet.
- **Travail régulier :** Les groupes doivent s'engager dans un travail régulier tout au long de la durée du projet. Il est recommandé d'organiser des réunions hebdomadaires pour discuter de la progression, planifier les étapes suivantes et résoudre les problèmes rencontrés. Cette régularité est essentielle pour éviter les complications de dernière minute et garantir un développement fluide du projet.
- **Soumission du projet :** Votre travail doit être concrétisé par un rapport exhaustif au format PDF, rédigé avec précision et clarté pour refléter la rigueur et l'implication dans votre projet. Ce document, structuré et cohérent, devra comprendre les éléments suivants :

- *Introduction* : Présentez les ambitions du projet, la stratégie mise en œuvre et une synthèse des résultats atteints.
- *Corps du rapport* : Élaborez sur les étapes clés de la mise en œuvre, les obstacles surmontés, les résolutions prises, ainsi que la répartition des tâches au sein de l'équipe.
- *Conclusion* : Récapitulez les apprentissages, mettez en avant les réussites, discutez les éventuelles contraintes de votre méthodologie, et envisagez les améliorations futures.
- *Annexes* : Incluez tous les codes sources, données, et autres supports exploités ou élaborés durant votre projet.

Il est impératif d'utiliser le modèle LaTeX fourni sur la plateforme *coursenligne* pour assurer l'uniformité et le professionnalisme des soumissions. Les rapports qui ne respectent pas ces consignes ou qui sont soumis dans un format différent seront malheureusement écartés. N'oubliez pas de soumettre également les fichiers sources LaTeX de votre travail.

- **Évaluation des pairs** : À la fin du projet, chaque étudiant sera tenu de soumettre une évaluation confidentielle de la contribution de ses coéquipiers. Cela permet une évaluation équitable et encourage la participation active de tous les membres de l'équipe.
- **Date limite** : Tous les matériaux du projet, y compris le rapport final et le code source, doivent être soumis avant le 15 janvier 2024. Aucun retard ne sera accepté.

2 Listes des sujets

2.1 Titre du projet : Manipulation de l'Arbre de Syntaxe Abstrait (AST)

2.1.1 Description

Ce projet vise à approfondir la compréhension des étudiants des concepts de langages de programmation, de compilation et d'interprétation en manipulant directement les arbres de syntaxe abstraite (AST). Les étudiants travailleront sur des fragments de code, générant des AST à partir de ceux-ci et effectuant diverses transformations. Cela pourrait inclure la réorganisation des structures de contrôle, l'optimisation des expressions, la refonte des déclarations de variables, etc., tout en garantissant que la logique du programme d'origine reste inchangée.

2.1.2 Objectifs

Familiariser les étudiants avec le concept et la structure des AST et leur rôle dans la compilation et l'exécution des programmes. Développer des compétences en manipulation d'AST, y compris la génération, la transformation et l'optimisation des AST. Encourager la réflexion sur les différentes façons dont le même code peut être représenté et exécuté, ainsi que sur les implications en termes de performance et de lisibilité.

2.1.3 Fonctionnalités clés

Le projet comprend plusieurs fonctionnalités essentielles conçues pour offrir une expérience interactive et instructive en manipulant des AST. Ces fonctionnalités incluent :

- **Générateur d'AST** : Construit un AST à partir du code source fourni, permettant une analyse structurale détaillée.
- **Visualiseur d'AST** : Offre une représentation graphique de l'AST, facilitant la compréhension de la structure du code source.
- **Éditeur d'AST** : Autorise les modifications directes sur les éléments de l'AST, avec une mise à jour synchrone dans le code source correspondant.
- **Compareur de code** : Confronte le code source pré- et post-modification pour mettre en évidence les répercussions des ajustements apportés à l'AST.
- **Benchmark** : Évalue et compare les performances du code avant et après les transformations, illustrant l'efficacité des changements.

2.1.4 Technologies proposées

Pour la mise en œuvre des fonctionnalités ci-dessus, nous suggérons l'utilisation des technologies suivantes, choisies pour leur efficacité et leur pertinence dans le traitement des AST :

- **Langage de programmation** : Python, privilégié pour sa syntaxe intuitive et ses bibliothèques spécialisées dans la manipulation des AST (telles que `ast` et `gast`).
- **Environnement de développement** : Un IDE performant comme PyCharm ou VSCode, complété par des extensions facilitant le travail avec les AST.
- **Visualisation** : Utilisation de Firefox pour une visualisation des structures JSON représentant les AST.
- **Contrôle de version** : Git, essentiel pour une gestion efficace des versions et une collaboration fluide au sein de l'équipe.
- **Benchmarking** : Recours à des outils intégrés ou à des bibliothèques externes dédiées pour quantifier les conséquences des modifications apportées au code.

2.2 Titre du projet : Développement de Programmes Assisté par une IA Locale

2.2.1 Description

Ce projet invite les étudiants à élaborer une plateforme qui intègre une intelligence artificielle (IA) dans le processus de développement de logiciels. En exploitant une IA déployée localement, la plateforme sera capable de proposer des recommandations de codage, d'optimiser le code existant, et de faciliter diverses tâches de programmation. L'initiative explore le potentiel de l'IA pour améliorer l'efficacité et la résolution de problèmes dans le développement de logiciels.

2.2.2 Objectifs

- **Intégration de l'IA** : Fusionner une intelligence artificielle dans l'environnement de développement pour améliorer l'expérience de codage.
- **Assistance au codage** : Mettre en œuvre l'IA pour l'analyse, la suggestion, et la génération automatisée de segments de code.
- **Interaction homme-machine** : Approfondir la compréhension des dynamiques interactives entre les développeurs et l'IA dans un contexte de programmation.
- **Évaluation de l'efficacité** : Mesurer l'impact de l'IA sur la rapidité, l'efficacité et la qualité du processus de développement de logiciels.

2.2.3 Fonctionnalités clés

- **Interface de communication** : Système interactif permettant des requêtes en langage naturel et générant des réponses cohérentes et pertinentes de la part de l'IA.
- **Analyse de code** : Capacité de l'IA à examiner le code, à identifier les domaines d'amélioration et à proposer des recommandations optimisées.
- **Génération de code** : Fonctionnalité où l'IA, en se basant sur les demandes de l'utilisateur, génère des fragments de code pour des fonctions spécifiques.
- **Débogage assisté** : Assistance de l'IA dans la détection et la résolution des erreurs de code, améliorant ainsi l'efficacité du processus de débogage.
- **Évaluation de performance** : Outil comparatif pour analyser l'efficacité du développement avec l'assistance de l'IA par rapport aux méthodes traditionnelles.

2.2.4 Technologies proposées

- **IA Locale** : Implémentation d'un modèle d'IA avancé, tel que GPT-3 ou GPT-4, pour les interactions en langage naturel et l'analyse de code. Pour une installation locale, suivez les instructions sur [ce guide](#).
- **Langage de Programmation** : Libre choix entre Python, Java, ou tout autre langage adapté au développement de l'interface utilisateur et à l'intégration de l'IA.
- **Environnement de Développement** : Un IDE ou un éditeur de code qui permet l'intégration fluide avec les outils d'IA.
- **API et Bibliothèques** : Intégration d'APIs spécifiques pour l'IA et utilisation de bibliothèques pour le traitement du langage naturel, si nécessaire. Pour plus d'informations sur l'utilisation de GPT-3 ou GPT-4 pour diverses applications, visitez [ce dépôt](#).

2.3 Titre du projet : Analyseur de Complexité de Code en C

2.3.1 Description

Dans ce projet, les étudiants créeront un outil d'analyse pour évaluer la complexité des programmes écrits en C. Cet outil extraira et analysera divers métriques pour déterminer la complexité du code, comme la complexité cyclomatique, le nombre de lignes de code, le degré de nidification des structures de contrôle, et plus encore. L'objectif est de sensibiliser les étudiants à l'importance de la lisibilité, de la maintenabilité, et de l'efficacité du code en démontrant comment la complexité influence ces facteurs.

2.3.2 Objectifs

Comprendre et appliquer les concepts fondamentaux de la complexité du code dans la programmation. Développer un outil capable d'analyser le code C et d'en extraire des métriques clés liées à la complexité. Interpréter les résultats de l'analyse pour proposer des recommandations concrètes visant à améliorer la qualité du code. Sensibiliser à l'importance des pratiques de codage propres et efficaces.

2.3.3 Fonctionnalités clés

Analyse de la complexité cyclomatique : Évaluer le nombre de chemins d'exécution à travers le code. Compteur de lignes de code : Calculer le nombre de lignes de code, en excluant les commentaires et les espaces blancs. Analyse de la profondeur de nidification : Déterminer à quel point les structures de contrôle sont imbriquées. Rapports détaillés : Générer des rapports qui récapitulent les métriques de complexité, avec éventuellement un système de notation ou de classement. Suggestions d'amélioration : Sur la base des analyses, l'outil devrait pouvoir proposer des recommandations pour améliorer la qualité et la lisibilité du code.

2.3.4 Technologies proposées

Langage de programmation : C pour l'analyse du code source et, si nécessaire, un langage de script tel que Python pour la partie analyse et génération de rapports. Analyseur de code : Utilisation de bibliothèques existantes ou de frameworks (si disponible et applicable) pour aider à extraire des métriques du code C. Environnement de développement : IDE approprié avec support pour C et tout autre langage utilisé (comme VSCode, Eclipse, etc.). Visualisation de données : (optionnel) Outils ou bibliothèques pour créer des visualisations graphiques des résultats d'analyse, facilitant l'interprétation.

2.4 Titre du projet : Visualisation de tri d’algorithmes

2.4.1 Description

Dans ce projet, les étudiants construiront une application visuelle qui illustrera comment différents algorithmes de tri fonctionnent en temps réel. L’application devra être capable de prendre une série de données et d’appliquer plusieurs types de tris (comme le tri à bulles, le tri par sélection, le tri par insertion, le tri rapide, etc.), tout en affichant graphiquement le processus de tri étape par étape. Ce projet aidera les étudiants à comprendre les mécanismes internes des algorithmes de tri, ainsi que leur complexité temporelle respective.

2.4.2 Objectifs

Approfondir la compréhension des étudiants des algorithmes de tri et de la complexité algorithmique. Développer des compétences en programmation graphique et en visualisation de données. Comparer et contraster l’efficacité des différents algorithmes de tri en fonction de divers ensembles de données. Encourager la pensée critique sur la performance algorithmique et l’optimisation du code.

2.4.3 Fonctionnalités clés

Sélection d’Algorithmes : Permettre aux utilisateurs de choisir parmi divers algorithmes de tri pour tester. Contrôles de Visualisation : Inclure des fonctionnalités pour contrôler la visualisation (par exemple, démarrer, arrêter, mettre en pause, réinitialiser). Jeux de Données Variables : Permettre la manipulation de la taille de l’ensemble de données et des valeurs pour tester les performances des algorithmes. Indicateurs de Performance : Afficher les métriques pertinentes liées à la performance de l’algorithme (par exemple, le temps d’exécution, le nombre de comparaisons ou d’échanges, etc.). Narration Graphique : Créer des animations détaillées montrant les mouvements d’éléments, les comparaisons, etc., pour expliquer le fonctionnement interne de chaque algorithme.

2.4.4 Technologies proposées

Langage de programmation : JavaScript (avec des bibliothèques de visualisation comme D3.js) pour un frontend interactif, ou Python (avec Pygame ou Matplotlib) pour une approche basée sur un script. Environnement de développement : Un IDE qui supporte le langage choisi, comme VSCode, PyCharm, ou un éditeur de texte avec les plugins nécessaires. Framework UI (facultatif) : Des bibliothèques comme React (pour les applications web) ou Tkinter (pour Python) si une interface utilisateur est nécessaire au-delà de la visualisation. Contrôle de version : Git, pour la gestion du code source et la collaboration entre les membres de l’équipe.

2.5 Titre du projet : Créateur d’emploi du temps

2.5.1 Description

Dans ce projet, les étudiants sont chargés de concevoir un système intelligent de création d’emploi du temps qui prend en compte diverses contraintes et préférences et génère des emplois du temps optimisés pour les étudiants, les enseignants, ou les employés dans un contexte organisationnel. Le système doit être capable de gérer des entrées complexes, d’offrir des solutions flexibles et d’adapter les emplois du temps en fonction des changements et des demandes spécifiques des utilisateurs.

2.5.2 Objectifs

Concevoir et implémenter un algorithme robuste de création d’emploi du temps qui tient compte des contraintes de temps, des ressources disponibles (par exemple, salles de classe, équipements) et des préférences individuelles. Créer une interface utilisateur intuitive où les utilisateurs peuvent entrer leurs contraintes et préférences et recevoir un emploi du temps personnalisé. Intégrer une fonction de modification qui permet aux utilisateurs d’apporter des modifications en temps réel, avec un recalcul automatique de l’emploi du temps. Assurer que le système produit des résultats non seulement viables mais aussi optimaux, en utilisant des principes d’optimisation.

2.5.3 Fonctionnalités clés

Interface de saisie : Un formulaire détaillé pour recueillir toutes les informations nécessaires concernant les préférences des utilisateurs et les ressources disponibles. Algorithme de planification : Un cœur algorithmique qui crée des emplois du temps en tenant compte de toutes les contraintes et optimise selon les critères spécifiés. Visualisation de l’emploi du temps : Un affichage graphique de l’emploi du temps, permettant aux utilisateurs de voir leur semaine à un niveau macroscopique. Fonctionnalité de modification en direct : Permettre des ajustements en temps réel avec une mise à jour instantanée de l’emploi du temps. Feedback et ajustement : Les utilisateurs peuvent laisser des commentaires sur les emplois du temps proposés, qui seront utilisés pour affiner les algorithmes de planification.

2.5.4 Technologies proposées

Langage de programmation : JavaScript (pour une application web interactive) ou Python (pour sa simplicité et ses bibliothèques puissantes comme NumPy pour les calculs complexes). Framework Front-end : React ou Angular pour développer une interface utilisateur dynamique et réactive. Framework Back-end : Node.js ou Flask pour gérer les requêtes du serveur, le traitement des données, et l’interaction avec la base de données. Base de données : SQL ou MongoDB pour stocker et récupérer toutes les informations nécessaires, des données des utilisateurs aux ressources disponibles. Outils d’optimisation : Bibliothèques spécialisées ou intégrées, selon le langage de programmation choisi, pour aider à trouver des solutions optimales aux problèmes de planification complexes.

2.6 Titre du projet : Générateur de cartes de monde fictif

2.6.1 Description

Le projet consiste à créer un générateur de cartes de mondes fictifs, permettant aux utilisateurs de générer dynamiquement des cartes de territoires inexplorés pour des jeux, des récits de fantasy, ou des scénarios hypothétiques. En utilisant des algorithmes de génération procédurale, les étudiants construiront un outil capable de créer des topographies uniques, comprenant diverses caractéristiques géographiques, telles que des montagnes, des rivières, des forêts, des déserts, des villages, et des villes, le tout avec des options de personnalisation pour les utilisateurs.

2.6.2 Objectifs

Explorer et appliquer les principes de la génération procédurale pour créer des paysages géographiques variés. Développer une interface utilisateur interactive permettant la personnalisation des caractéristiques géographiques et culturelles des mondes générés. Intégrer les concepts de bases de données pour stocker et récupérer des modèles de mondes, permettant aux utilisateurs de sauvegarder et de revisiter les mondes créés. Encourager l'expérimentation avec différents algorithmes pour observer les variations dans les résultats de la génération de monde.

2.6.3 Fonctionnalités clés

Création procédurale : Génération de terrains diversifiés avec différents biomes et structures géographiques. Interface utilisateur : Une interface conviviale pour ajuster les paramètres de génération de monde, comme le climat, le relief, et la densité de la population. Visualisation de carte : Outils de visualisation en 2D ou 3D présentant les topographies générées de manière interactive. Sauvegarde et chargement : Fonctionnalités pour enregistrer des cartes générées et les recharger pour une utilisation future. Exportation : Capacité d'exporter des cartes dans différents formats pour utilisation dans des logiciels de conception graphique ou des plateformes de jeux.

2.6.4 Technologies proposées

Langage de programmation : Python ou JavaScript, populaires pour leur flexibilité et leur support pour la génération graphique et procédurale. Frameworks graphiques : Three.js pour la visualisation 3D (si JavaScript est choisi), ou des bibliothèques comme matplotlib ou Pygame pour Python. Technologies web (optionnelles) : HTML/CSS/JavaScript pour créer une interface utilisateur web si le projet est destiné à être une application web. Base de données : SQLite pour un stockage léger ou MongoDB pour des solutions basées sur des documents, pour enregistrer et gérer des modèles de monde. Contrôle de version : Git, pour la collaboration et le suivi des modifications du projet.

2.7 Titre du projet : Application de chat en réseau avec système de cryptage

2.7.1 Description

Dans ce projet, les étudiants sont chargés de créer une application de chat fonctionnelle qui permet la communication en temps réel entre plusieurs utilisateurs via un réseau. L'objectif principal est de mettre en œuvre un système de cryptage pour sécuriser les messages transitant sur le réseau, assurant la confidentialité et l'intégrité des données. Les étudiants exploreront les concepts fondamentaux des réseaux, de la cryptographie, et de la sécurité informatique.

2.7.2 Objectifs

Comprendre et appliquer les principes de base des communications réseau et de la programmation socket. Mettre en œuvre des stratégies de cryptographie pour sécuriser les communications. Développer une interface utilisateur graphique (GUI) pour faciliter la communication entre utilisateurs. Promouvoir la compréhension de l'importance de la sécurité des données et de la gestion des risques dans les applications en réseau.

2.7.3 Fonctionnalités clés

Communication en temps réel : Les utilisateurs peuvent envoyer et recevoir des messages instantanément. Cryptage des messages : Les messages sont chiffrés avant l'envoi et déchiffrés après la réception. Gestion des utilisateurs : Capacité de gérer les utilisateurs, y compris la création de nouveaux comptes et la gestion des utilisateurs connectés. Interface utilisateur intuitive : Une GUI propre et réactive qui facilite la navigation, l'envoi de messages, et la visualisation des utilisateurs en ligne. Logs de chat sécurisés : Les historiques de chat sont sauvegardés de manière sécurisée et peuvent être consultés ou supprimés par l'utilisateur.

2.7.4 Technologies proposées

Langage de programmation : Java (pour sa gestion intégrée des sockets et ses bibliothèques de sécurité), Python (avec des bibliothèques comme Tkinter pour la GUI et PyCrypto pour le cryptage), ou une autre langue préférée par les étudiants. Environnement de développement : Environnement qui supporte le développement d'applications en réseau et la cryptographie, comme IntelliJ IDEA pour Java ou PyCharm pour Python. Protocoles de sécurité et cryptographie : Utilisation de TLS/SSL pour les communications sécurisées, et des bibliothèques de cryptographie intégrées pour le cryptage des messages. Système de gestion de version : Git pour le suivi des versions et la collaboration en équipe.

2.8 Titre du projet : Logiciel de compression de données

2.8.1 Description

Dans ce projet, les étudiants sont chargés de concevoir et d’implémenter un logiciel de compression de données simple mais efficace. L’objectif est de créer un programme capable de prendre des fichiers en entrée, d’appliquer une méthode de compression pour réduire la taille du fichier, et ensuite d’être capable de décompresser le fichier à son état original. Ce projet permettra aux étudiants de comprendre les algorithmes de compression, les applications pratiques de la théorie de l’information, et l’importance de la gestion de l’espace de stockage dans l’informatique moderne.

2.8.2 Objectifs

Explorer et comprendre les différentes méthodes de compression de données, y compris la compression sans perte et avec perte. Développer des compétences en implémentant des algorithmes de compression et en travaillant avec des structures de données binaires. Analyser l’efficacité de diverses techniques de compression en termes de taux de compression, de vitesse et de fidélité de la récupération des données. Sensibiliser à l’importance de la compression de données dans différents domaines, comme le stockage, la transmission de données, et les applications web.

2.8.3 Fonctionnalités clés

Interface utilisateur : Une interface simple pour que les utilisateurs puissent uploader des fichiers et choisir le type de compression qu’ils souhaitent appliquer. Compression et décompression : Le cœur du logiciel capable de compresser les fichiers entrants et de les décompresser retour à leur état original. Aperçu des statistiques : Affichage des statistiques telles que la taille originale, la taille après compression, et le taux de compression. Support de multiples formats : Capacité de gérer différents types de fichiers (texte, images, audio, etc.) et d’appliquer la compression pertinente. Rapport de performance : Fonctionnalité permettant de générer des rapports sur le processus de compression, incluant le temps pris, l’efficacité, et toute perte de données si applicable.

2.8.4 Technologies proposées

Langage de programmation : Python ou C++, connus pour leurs performances en matière de manipulation de données et d’implémentation d’algorithmes. Environnement de développement : Un IDE adapté comme Visual Studio Code, Eclipse, ou PyCharm. Bibliothèques de compression : Utilisation de bibliothèques existantes pour certaines fonctionnalités de compression peut être autorisée (par exemple, zlib, libpng, etc.). Interface utilisateur : Des frameworks pour développer l’interface utilisateur, tels que Qt (pour C++) ou Tkinter (pour Python), selon le langage choisi. Contrôle de version : Git, pour suivre l’évolution du projet et faciliter la collaboration si le projet est réalisé en groupe.

3 Structure Générale des Projets

Tous les projets dans ce programme nécessitent l’adhésion à certaines pratiques standard et l’utilisation d’outils communs. Ces exigences assurent une expérience uniforme dans le développement du projet et facilitent la collaboration et l’évaluation.

3.1 Utilisation de GitHub

- **Pourquoi GitHub ?** GitHub est une plateforme de développement collaboratif qui permet aux développeurs de travailler ensemble sur des projets. En utilisant GitHub, vous pouvez collaborer avec d’autres, gérer les versions de votre projet et maintenir un historique de tous les changements.
- **Ce que vous devez faire :** Assurez-vous que votre projet est entièrement accessible sur GitHub, ce qui inclut non seulement le code, mais aussi la documentation et les scripts de configuration. Familiarisez-vous avec les commandes Git de base, la gestion des branches, les pull requests et les merges. Ce savoir-faire est crucial pour la collaboration en équipe et la gestion efficace des modifications du code.

- **Exemple :**

```
# Cloner le dépôt
git clone https://github.com/utilisateur/le_nom_du_projet.git

# Ajouter tous les fichiers modifiés à l'index
git add .

# Commiter vos changements
git commit -m "Ajout de nouvelles fonctionnalités"

# Pousser vos modifications vers le dépôt distant
git push origin main
```

3.2 Serveur Web avec Flask

- **Pourquoi Flask ?** Flask est un framework web léger pour Python. Il est conçu pour faciliter le démarrage rapide d'une application web, tout en garantissant que l'application reste performante avec l'ajout de nouvelles fonctionnalités.
- **Ce que vous devez faire :** Si votre projet implique le développement d'une application web, utilisez Flask en raison de sa simplicité et de sa flexibilité. Assurez-vous de suivre les meilleures pratiques pour la configuration, la sécurité et l'optimisation des performances.
- **Exemple :**

```
# Installation de Flask
pip install Flask

# Exemple d'application Flask
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Bonjour, monde !'

if __name__ == '__main__':
    app.run(debug=True)
```

3.3 Communication en JSON

- **Pourquoi JSON ?** JSON (JavaScript Object Notation) est un format léger d'échange de données. Facile pour les humains à lire et à écrire et pour les machines à analyser et à générer, JSON est idéal pour la communication entre services web et clients.
- **Ce que vous devez faire :** Toutes les interactions de votre projet qui nécessitent l'échange de données doivent se faire en JSON. Cela concerne la communication entre le client et le serveur, ou entre différentes parties de votre application (par exemple, les services back-end).
- **Exemple :**

```
from flask import Flask, jsonify

app = Flask(__name__)
```

```

@app.route('/api/data', methods=['GET'])
def get_data():
    example_data = {
        'id': 1,
        'name': "Exemple"
    }
    return jsonify(example_data)

if __name__ == '__main__':
    app.run(debug=True)

```

3.4 Génération d'AST via Clang

- **Pourquoi Clang ?** Clang est un compilateur pour le langage de programmation C, offrant des performances de pointe et une utilisation mémoire réduite. L'utilisation de Clang pour la génération d'AST assure une analyse précise et une représentation détaillée du code source.
- **Ce que vous devez faire :** Si votre projet implique l'analyse de code source en C, utilisez Clang pour générer un AST (Abstract Syntax Tree). L'AST est crucial pour comprendre la structure du code et est utilisé dans de nombreuses applications telles que l'analyse statique, la refactoring, la navigation dans le code, etc.
- **Exemple :**

```

# Générer un AST pour un fichier source
clang -Xclang -ast-dump -fsyntax-only example.c

```