

# Gestion de base de données de super-héros

---

## Présentation

---

### **Objectif du projet :**

Créer une application Python qui permet de gérer une base de données de super-héros en utilisant le fichier JSON SuperHeros.json comme source de données.

### **Nombre d'étudiants**

La totalité de la promo.

### **Répartition**

- ⇒ Étudiant N°1 : Responsable du projet
- ⇒ Étudiant N°2 et Étudiant N° 9 : Modélisation des données
- ⇒ Étudiant N°3 : Base de données
- ⇒ Étudiant N°4 : Interface ligne de commande
- ⇒ Étudiant N°5 et Étudiant N° 10 Création de vues
- ⇒ Étudiant N°6 : Gestion de l'authentification et des autorisations
- ⇒ Étudiant N°7 et Étudiant N°11 Fonctionnalité de Recherche Avancée
- ⇒ Étudiant N°8 et Étudiant N°12 : Interface Tkinter
- ⇒ Étudiant N°13 : Responsable des tests

### **Compte-rendu**

Aucun

---

## Étudiant N°1 : Responsable du projet

---

L'Étudiant N°1 assure le rôle de responsable du projet en coordonnant toutes les activités des membres de l'équipe et en veillant à ce que le projet avance de manière cohérente.

	GITGIT	
	SUPERHEROS	

### **1. Création du dépôt GitHub**

- ⇒ Créer un nouveau dépôt GitHub pour le projet.
- ⇒ Configurer les paramètres du dépôt, y compris les autorisations, les branches par défaut, etc.

### **2. Importation des données**

- ⇒ Importer le fichier SuperHeros.json dans le projet.
- ⇒ S'assurer que les images associées aux super-héros sont également importées dans le dépôt.

### **3. Mise en place des pull requests (PR)**

- ⇒ Définir une politique de gestion des PR, notamment les règles de fusion, les critères de validation et de qualité du code.
- ⇒ S'assurer que chaque tâche est assignée à une branche distincte, et que les PR sont créées à partir de ces branches.

### **4. Vérification des pull requests**

- ⇒ Examiner les PR soumises par les étudiants pour s'assurer qu'elles respectent les normes de qualité du code.
- ⇒ Effectuer des revues de code pour identifier les erreurs potentielles, les problèmes de conformité, et les améliorations possibles.

### **5. Coordination des équipes**

- ⇒ Organiser des réunions d'équipe régulières pour suivre les progrès, discuter des défis rencontrés et prendre des décisions stratégiques.
- ⇒ Assurer une communication efficace entre les membres de l'équipe.

### **6. Planification et gestion du projet**

- ⇒ Établir un plan de projet avec des échéances pour chaque étape du projet.
- ⇒ Gérer les priorités et les ressources pour s'assurer que le projet avance conformément au calendrier prévu.

### **7. Documentation**

- ⇒ Créer une documentation claire pour l'ensemble du projet, y compris une description de l'architecture, des composants, des flux de travail, etc.
- ⇒ Inclure des instructions détaillées sur la façon d'installer, de configurer et d'utiliser l'application, en mettant l'accent sur la CLI.

### **8. Tests unitaires**

- ⇒ Coordonner avec les autres étudiants pour s'assurer que les tests sont correctement effectués sur les PR soumises.

⇒ Assurer la couverture complète des tests pour les fonctionnalités de l'application.

### **9. Gestion des conflits**

- ⇒ Intervenir en cas de conflits entre les PR et aider à les résoudre de manière appropriée.
- ⇒ Fournir des conseils pour la fusion des branches lorsque cela est nécessaire.

### **10. Rapports d'avancement**

- ⇒ Fournir des rapports d'avancement réguliers à l'équipe et aux parties prenantes pour tenir tout le monde informé de l'état du projet.

---

## **Étudiant N°2 : et Étudiant N°9 Modélisation des données**

---

L'Étudiant N°2 et 9 auront en responsabilités dans le projet de modélisation des données des super-héros. Il garantit également que les données sont correctement chargées, stockées et modifiées pour le reste du projet.

### **1. Création de la classe SuperHeros (Python)**

- ⇒ Créer une classe Python nommée `SuperHero` pour représenter un super-héros.
- ⇒ Définir les attributs de la classe en fonction des données du fichier JSON (id, name, id-api, slug).

### **2. Méthodes de chargement des données**

- ⇒ Écrire une méthode dans la classe `SuperHero` pour charger les données du fichier JSON.
- ⇒ Cette méthode devrait lire le fichier SuperHeros.json, analyser son contenu JSON et créer des objets `SuperHero` à partir des données.

### **3. Méthodes de stockage des données**

- ⇒ Écrire une méthode pour stocker les objets `SuperHero` dans une structure de données appropriée, par exemple une liste.
- ⇒ Cette méthode devrait permettre l'ajout initial des super-héros à la liste.

### **4. Méthodes de modification des données**

- ⇒ Écrire des méthodes pour permettre la modification des attributs des super-héros déjà stockés.
- ⇒ Par exemple, une méthode pour mettre à jour le nom d'un super-héros existant.

### **5. Gestion des erreurs et validations**

- ⇒ Ajouter des mécanismes pour gérer les erreurs potentielles lors du chargement des données depuis le fichier JSON.
- ⇒ Mettre en place des validations pour s'assurer que les données chargées sont conformes aux attentes.

### **6. Documentation**

- ⇒ Documenter la classe `SuperHero` et ses méthodes pour expliquer leur fonctionnement et leurs paramètres.

### **7. Tests unitaires**

- ⇒ Écrire des tests unitaires pour les méthodes de la classe `SuperHero` afin de vérifier leur fonctionnement correct.
- ⇒ Inclure des cas de test pour le chargement de données, la modification des attributs, etc.

---

## **Étudiant N°3 : Mise en œuvre de la persistance des données – Modélisation des données**

---

L'Étudiant N°3 est responsable de la mise en œuvre de la persistance des données.

### **1. Gestion de la base de données**

- ⇒ Choisir une base de données adaptée pour stocker les données des super-héros, par exemple SQLite, PostgreSQL, MySQL, ou MongoDB.
- ⇒ Installer et configurer la base de données dans l'environnement de développement.

### **2. Création de modèles de données**

- ⇒ Créer des modèles de données correspondants à la structure des super-héros pour la base de données choisie.
- ⇒ Assurer la correspondance entre les attributs de la classe `SuperHero` (Étudiant N°2) et les champs de la base de données.

### **3. Mise en place des opérations CRUD**

- ⇒ Implémenter les opérations CRUD (Create, Read, Update, Delete) pour interagir avec la base de données.
- ⇒ Écrire des fonctions ou des méthodes pour ajouter, lire, mettre à jour et supprimer des super-héros dans la base de données.

#### **4. Synchronisation des données**

- ⇒ Assurer la synchronisation entre la base de données et les objets `SuperHero` créés par l'Étudiant N°2.
- ⇒ Mettre à jour la base de données lorsque de nouvelles données sont ajoutées ou lorsque des modifications sont apportées aux super-héros.

#### **5. Gestion des erreurs et des transactions**

- ⇒ Gérer les erreurs liées à la persistance des données de manière robuste, en utilisant des mécanismes de gestion des erreurs appropriés.
- ⇒ Utiliser des transactions lorsque nécessaire pour garantir la cohérence des données.

#### **6. Tests de persistance des données**

- ⇒ Écrire des tests unitaires pour vérifier que les opérations de persistance des données fonctionnent correctement.
- ⇒ S'assurer que les données sont correctement enregistrées dans la base de données et qu'elles peuvent être récupérées.

#### **7. Optimisation des requêtes**

- ⇒ Optimiser les requêtes à la base de données pour assurer des performances optimales, en utilisant des index, en évitant les requêtes coûteuses, etc.

#### **8. Collaboration avec l'équipe**

- ⇒ Travailler en étroite collaboration avec les autres membres de l'équipe pour s'assurer que la persistance des données est correctement intégrée dans l'application globale.
- ⇒ S'assurer que les données sont correctement synchronisées entre la base de données et les objets `SuperHero`.

---

## **Étudiant N°4 : Création d'une interface utilisateur en ligne de commande (CLI)**

---

L'Étudiant N°4 doit créer une interface utilisateur en ligne de commande complète et fonctionnelle pour l'application, permettant ainsi aux utilisateurs d'interagir facilement avec la base de données des super-héros

#### **1. Conception de l'interface utilisateur en ligne de commande (CLI)**

- ⇒ Concevoir une interface utilisateur en ligne de commande conviviale et intuitive pour permettre à l'utilisateur d'interagir avec la base de données des super-héros.
- ⇒ Planifier la structure du menu de commande et les options disponibles.

## **2. Création du menu principal**

- ⇒ Écrire un script Python qui affiche le menu principal de la CLI au démarrage de l'application.
- ⇒ Le menu devrait présenter les options de base, telles que l'affichage de la liste de super-héros, l'ajout, la modification, la suppression et la recherche de super-héros.

**A partir de maintenant, l'étudiant N°4 doit utiliser les fonctions écrites par les autres étudiants. Si elles ne sont pas encore écrites, il met un message indiquant le nom de la fonction puis doit se coordonner avec les autres développeurs pour les intégrer au fur et à mesure.**

## **3. Fonctionnalités d'affichage**

- ⇒ Ajouter des fonctionnalités pour afficher la liste complète des super-héros à partir de la base de données.
- ⇒ Afficher les détails d'un super-héros individuel lorsque l'utilisateur le sélectionne.

## **4. Ajout de super-héros**

- ⇒ Implémenter une fonctionnalité qui permet à l'utilisateur d'ajouter de nouveaux super-héros à la base de données en utilisant la CLI.
- ⇒ Valider les entrées de l'utilisateur pour s'assurer qu'elles sont correctes avant d'ajouter un super-héros.

## **5. Modification de super-héros**

- ⇒ Écrire des fonctions pour permettre à l'utilisateur de modifier les attributs d'un super-héros existant, comme son nom, son id-api, ou son slug.
- ⇒ Assurer la mise à jour appropriée des données dans la base de données.

## **6. Suppression de super-héros**

- ⇒ Mettre en place une fonctionnalité de suppression de super-héros qui permet à l'utilisateur de supprimer un super-héros de la base de données.
- ⇒ Confirmer la suppression avec l'utilisateur pour éviter les erreurs accidentelles.

## **7. Recherche de super-héros**

- ⇒ Créer une fonction de recherche qui permet à l'utilisateur de rechercher des super-héros par nom, id-api, ou slug.
  - ⇒ Afficher les résultats de la recherche de manière lisible.
- ⇒

## **8. Gestion des erreurs et des exceptions**

- ⇒ Gérer les erreurs et les exceptions qui peuvent se produire lors de l'interaction de l'utilisateur avec la CLI.
- ⇒ Fournir des messages d'erreur explicites et des instructions pour corriger les problèmes.

### **9. Tests de l'interface utilisateur en ligne de commande**

- ⇒ Écrire des tests unitaires pour chaque fonctionnalité de la CLI pour s'assurer qu'elle fonctionne correctement.
- ⇒ Tester différents scénarios d'utilisation.

### **11. Intégration avec les autres composants**

- ⇒ Collaborer avec les autres membres de l'équipe pour intégrer l'interface utilisateur en ligne de commande avec la modélisation des données (Étudiant N°2) et la persistance des données (Étudiant N°3).
- ⇒ Assurer que les opérations de la CLI sont correctement liées à la base de données.

---

## **Étudiant N°5 : Et Étudiant N°10 Création de vue**

---

Les Étudiants N°5 et N°12 doivent créer des vues conviviales et fonctionnelles pour l'application, permettant aux utilisateurs d'afficher, de rechercher et de parcourir les super-héros, tout en intégrant correctement les images associées

### **1. Conception des vues**

- ⇒ Concevoir les interfaces utilisateur (UI) pour afficher la liste des super-héros, les détails d'un super-héros individuel et la possibilité de rechercher des super-héros.
- ⇒ Planifier la disposition des éléments, la mise en page et l'affichage des images des super-héros.

### **2. Création de vues pour la liste des super-héros**

- ⇒ Écrire des vues qui affichent la liste complète des super-héros à partir de la base de données.
- ⇒ Utiliser une présentation claire et organisée pour afficher les super-héros, y compris leurs noms et leurs images.

### **3. Création de vues pour les détails d'un super-héros individuel**

- ⇒ Implémenter des vues détaillées qui affichent les informations complètes d'un super-héros lorsqu'il est sélectionné.
- ⇒ Inclure des détails tels que le nom, l'id-api, le slug et l'image du super-héros.

### **4. Ajout de fonctionnalités de recherche**

- ⇒ Créer une vue de recherche qui permet à l'utilisateur de rechercher des super-héros par nom, id-api, ou slug.
- ⇒ Afficher les résultats de la recherche de manière claire et facilement compréhensible.

	GITGIT	
	SUPERHEROS	

### **5. Intégration des images de super-héros**

- ⇒ Intégrer les images des super-héros dans les vues en utilisant des balises d'image HTML ou des composants d'affichage d'images appropriés.
- ⇒ S'assurer que les images sont correctement associées aux super-héros correspondants.

### **6. Style et présentation**

- ⇒ Appliquer des styles CSS pour améliorer l'apparence des vues, y compris la mise en forme des images, la gestion de la taille des polices, et la disposition générale.
- ⇒ Assurer une présentation conviviale pour l'utilisateur.

### **8. Intégration avec les autres composants**

- ⇒ Collaborer avec les autres membres de l'équipe pour intégrer les vues avec la modélisation des données (Étudiant N°2), la persistance des données (Étudiant N°3) et l'interface utilisateur en ligne de commande (Étudiant N°4).
- ⇒ Assurer que les opérations de la CLI et de la base de données sont correctement liées aux vues.

---

## **Étudiant N°6 : Gestion de l'authentification et des autorisations**

---

L'Étudiant N°6 est responsable de la gestion de l'authentification et des autorisations dans l'application permettant ainsi de contrôler l'accès des utilisateurs aux fonctionnalités de l'application en toute sécurité.

### **1. Conception du système d'authentification**

- ⇒ Concevoir le système d'authentification de l'application en déterminant les méthodes d'identification des utilisateurs (par exemple, nom d'utilisateur/mot de passe, authentification à deux facteurs, ou authentification via des fournisseurs tiers comme Google ou Facebook).
- ⇒ Planifier le flux d'inscription et de connexion.

### **2. Mise en place de l'authentification**

- ⇒ Implémenter l'authentification en utilisant un framework d'authentification adapté à votre application Python, tel que Flask-Login, Django-Auth, ou d'autres solutions similaires.
- ⇒ Assurer la sécurité en stockant les informations d'authentification de manière appropriée (en utilisant le hachage pour les mots de passe, par exemple).

### **3. Création de pages d'inscription et de connexion**

- ⇒ Créer des pages d'inscription et de connexion conviviales pour permettre aux utilisateurs de s'inscrire et de se connecter à l'application.
- ⇒ Gérer les erreurs d'inscription et de connexion de manière appropriée.

#### **4. Gestion de profil utilisateur**

- ⇒ Mettre en place la possibilité pour les utilisateurs de gérer leur profil, y compris la modification de leur mot de passe ou d'autres informations personnelles.
- ⇒ Assurer la sécurité en vérifiant l'identité de l'utilisateur lorsqu'il apporte des modifications à son profil.

#### **5. Définition des autorisations**

- ⇒ Définir des autorisations pour chaque action possible dans l'application, telles que l'ajout, la modification ou la suppression de super-héros.
- ⇒ Associer des rôles ou des groupes d'utilisateurs aux autorisations.

#### **6. Contrôle d'accès basé sur les autorisations**

- ⇒ Mettre en place un contrôle d'accès basé sur les autorisations pour s'assurer que les utilisateurs ne peuvent effectuer que les actions pour lesquelles ils ont les autorisations nécessaires.
- ⇒ Empêcher les utilisateurs non autorisés d'accéder à des fonctionnalités sensibles.

---

## **Étudiant N°7 : Et Étudiant N°11 Fonctionnalité de Recherche Avancée**

---

Les Étudiants N°7 et N°11 doivent mettre en œuvre une fonctionnalité de recherche avancée complète et efficace dans l'application, permettant aux utilisateurs de trouver rapidement les super-héros qu'ils recherchent en utilisant divers critères et options de filtrage et de tri.

#### **1. Analyse des besoins de recherche**

- ⇒ Collaborer avec l'équipe pour comprendre les besoins de recherche des utilisateurs et définir les critères de recherche prioritaires.

#### **2. Conception de l'interface utilisateur**

- ⇒ Concevoir l'interface utilisateur de la fonctionnalité de recherche en créant des maquettes ou des prototypes pour visualiser le flux de travail de recherche, les options de filtrage et de tri, et l'affichage des résultats.

#### **3. Mise en place de la recherche par différents attributs**

- ⇒ Implémenter la recherche pour permettre aux utilisateurs de rechercher des super-héros en fonction de différents attributs tels que l'id, le nom, l'id-api et le slug.
- ⇒ Assurer une correspondance insensible à la casse pour une expérience utilisateur conviviale.

#### **4. Options de filtrage**

⇒ Proposer des options de filtrage pour affiner les résultats de la recherche, par exemple, permettre aux utilisateurs de filtrer les super-héros par catégorie, par date de création, ou par d'autres critères pertinents.

#### **5. Options de tri**

⇒ Implémenter des options de tri pour permettre aux utilisateurs de trier les résultats de recherche par différents critères tels que le nom, l'id, ou d'autres attributs.

⇒ Assurer des options de tri ascendantes et descendantes.

#### **6. Intégration avec les vues existantes**

⇒ Collaborer avec les Étudiants N°5 et N°12 responsables de la création des vues pour s'assurer que la fonctionnalité de recherche est correctement intégrée dans les vues existantes.

⇒ Afficher les résultats de recherche dans une présentation conviviale.

---

## **Étudiant N°8 : Et Étudiant N°12 : Interface Tkinter**

---

Les Étudiants N°8 et N°14 pourront créer une interface graphique Tkinter conviviale et fonctionnelle pour votre application, en tenant compte des besoins des utilisateurs et de l'intégration avec les autres composants de l'application.

#### **1. Analyse des besoins de l'interface utilisateur**

⇒ Travailler en étroite collaboration avec l'équipe pour comprendre les besoins et les fonctionnalités clés que l'interface graphique devra prendre en charge.

⇒ Recueillir les exigences spécifiques des utilisateurs finaux en matière d'interface utilisateur.

#### **2. Conception de l'interface utilisateur**

⇒ Concevoir l'interface utilisateur en utilisant Tkinter en créant des maquettes ou des prototypes pour visualiser la disposition des éléments, les boutons, les zones de texte, et les composants d'affichage des données.

⇒ Planifier la navigation et l'agencement des éléments de l'interface.

#### **3. Création des fenêtres et des widgets Tkinter**

⇒ Écrire le code Python pour créer les fenêtres principales de l'application ainsi que les widgets Tkinter nécessaires, tels que des boutons, des étiquettes, des zones de texte, des listes, etc.

⇒ Organiser les widgets sur les fenêtres en fonction de la conception prévue.

#### **4. Gestion des événements Tkinter**

- ⇒ Écrire des gestionnaires d'événements Tkinter pour gérer les actions de l'utilisateur, telles que les clics de bouton, la saisie de texte, et d'autres interactions.
- ⇒ Associer des fonctions aux widgets pour répondre aux actions de l'utilisateur.

#### **5. Intégration avec les fonctionnalités existantes**

- ⇒ Collaborer avec les autres membres de l'équipe pour intégrer l'interface Tkinter avec les fonctionnalités existantes de l'application, telles que la modélisation des données, la persistance des données, la recherche avancée, etc.
- ⇒ S'assurer que les actions de l'utilisateur dans l'interface graphique déclenchent les opérations appropriées dans l'application.

#### **6. Optimisation de l'interface utilisateur**

- ⇒ Optimiser la convivialité de l'interface utilisateur en améliorant la disposition, en ajoutant des raccourcis clavier, et en minimisant la duplication d'effort pour les utilisateurs.
- ⇒ Assurer une expérience utilisateur fluide.

#### **7. Tests de l'interface Tkinter**

- ⇒ Écrire des tests pour s'assurer que l'interface Tkinter fonctionne correctement et interagit de manière fluide avec les autres composants de l'application.
- ⇒ Tester différentes interactions utilisateur.

---

## **Étudiant N°13 Et Étudiant N°14 Responsable des Tests Unitaires et d'Intégration**

---

L'Étudiant N°13 jouera un rôle essentiel dans la garantie de la qualité de l'application en s'assurant que chaque composant fonctionne correctement, que les intégrations entre les composants sont fluides et que les erreurs sont détectées et corrigées rapidement

#### **1. Analyse des exigences de test**

- ⇒ Collaborer avec l'équipe pour comprendre les exigences de test pour chaque composant de l'application, y compris la modélisation des données, la persistance des données, l'interface utilisateur, la recherche, etc.

#### **2. Conception des cas de test**

- ⇒ Concevoir des cas de test unitaires pour chaque fonction ou méthode dans le code, en couvrant différents scénarios, y compris les cas de succès et les cas d'erreur.

⇒ Concevoir des cas de test d'intégration pour vérifier que les différents composants de l'application fonctionnent correctement ensemble.

### **3. Mise en place de l'environnement de test**

⇒ Configurer un environnement de test séparé, si nécessaire, pour isoler les tests des données de production et pour garantir une reproductibilité des tests.

### **4. Écriture des tests unitaires**

⇒ Écrire des tests unitaires en utilisant des frameworks de test Python tels que unittest, pytest, ou d'autres outils de test.

⇒ Tester chaque fonction ou méthode de manière isolée en vérifiant que les résultats correspondent aux attentes.

### **5. Écriture des tests d'intégration**

⇒ Écrire des tests d'intégration pour vérifier que les différents composants de l'application fonctionnent ensemble correctement.

⇒ Tester les flux de données entre les composants pour détecter les éventuels problèmes d'intégration.

### **6. Exécution des tests**

⇒ Exécuter régulièrement les tests unitaires et d'intégration pour s'assurer que le code reste fiable au fur et à mesure des modifications.

⇒ Automatiser l'exécution des tests dans un pipeline d'intégration continue (CI) si possible.

### **7. Rapports et suivi des résultats**

⇒ Générer des rapports détaillés sur les résultats des tests, y compris les succès et les échecs.

⇒ Suivre les résultats des tests et signaler les problèmes à l'équipe de développement.

### **8. Correction des erreurs**

⇒ Collaborer avec les développeurs pour identifier et résoudre les problèmes détectés par les tests unitaires et d'intégration.

⇒ S'assurer que les erreurs sont corrigées et que les tests échouants sont mis à jour lorsque nécessaire.

### **9. Tests de régression**

⇒ Mettre en place des tests de régression pour garantir que les fonctionnalités existantes ne sont pas impactées par de nouvelles modifications.

⇒ Exécuter ces tests régulièrement pour détecter les régressions potentielles.

	GITGIT	
	SUPERHEROS	

### **10. Amélioration continue**

⇒ Participer à l'amélioration continue des tests en identifiant des moyens d'optimiser les cas de test, d'automatiser davantage de tests et d'accroître la couverture des tests.

### **11. Documentation**

⇒ Documenter les cas de test, les procédures d'exécution, les résultats et les recommandations pour l'équipe de développement.