

## Table des matières

Partie 1 - lien GitHub.....	1
Partie 2 - Documentation technique.....	1
2.1 - Liste des routes API.....	1
Authentification (/api/auth).....	1
Héros (/api/heroes).....	2
Utilisateurs (/api/users).....	2
Journalisation (/api/logs).....	2
2.2 - Schéma des collections MongoDB.....	3
Collection heros.....	3
Collection users.....	3
Collection logs.....	4
2.3 - Captures d'écran des interfaces.....	4
Tableau de Bord.....	4
Page de Détail.....	4
Interface d'ajout et de modification (similaire).....	5
Interface d'inscription et de connexion (similaire).....	5
Interface d'Administration.....	6
Partie 3 - Rapport de projet.....	6
3.1 - Présentation générale.....	6
3.2 - Architecture et technologies.....	6
Backend.....	6
Frontend.....	7
3.3 - Fonctionnalités réalisées.....	7
3.4 - Difficultés rencontrées.....	8
3.4 - Axes d'amélioration.....	8
Partie 4 - Vidéo de démonstration.....	8

## Partie 1 - lien GitHub

Voici mon lien GitHub vers le projet : <https://github.com/KyllianCel/superhero-manager>

## Partie 2 - Documentation technique

### 2.1 - Liste des routes API

Voici l'ensemble des routes API disponible via ce lien <http://localhost:5000> .

#### Authentification (/api/auth)

Méthode	Route	Description	Accès
POST	/register	Créer un nouveau compte utilisateur	Public
POST	/login	Se connecter et obtenir un token JWT	Public

## Héros (/api/heroes)

Méthode	Route	Description	Accès
GET	/	Récupérer la liste de tous les héros	Public
GET	/:id	Récupérer les détails d'un héros	Public
POST	/	Créer un héros (avec image)	Connecté
PUT	/:id	Modifier un héros (avec/sans image)	Connecté
DELETE	/:id	Supprimer un héros et son image	Admin Uniquement

## Utilisateurs (/api/users)

Méthode	Route	Description	Accès
GET	/	Lister tous les utilisateurs inscrits	Admin Uniquement

## Journalisation (/api/logs)

Méthode	Route	Description	Accès
GET	/	Voir l'historique des actions	Admin Uniquement

## 2.2 - Schéma des collections MongoDB

L'application utilise trois collections.

### Collection heros

Stocke les informations détaillées des super-héros.

```
{
  "id": Number,
  "name": String,      Nom du héros
  "slug": String,     Version URL du nom
  "univers": String,  Marvel, DC, Autre
  "images": {
    "xs": String, "sm": String, "md": String, "lg": String
  },
  "powerstats": {
    "intelligence": Number, "strength": Number, ...
  },
  "biography": {
    "fullName": String, "publisher": String, "alignment": String ...
  },
  "appearance": { ... },
  "work": { ... },
  "connections": { ... }
}
```

### Collection users

Gère les utilisateurs et leurs rôles.

```
{
  "username": String,
  "passwordHash": String, Mot de passe crypté
  "role": String,      "admin" ou "editor"
  "createdAt": Date
}
```

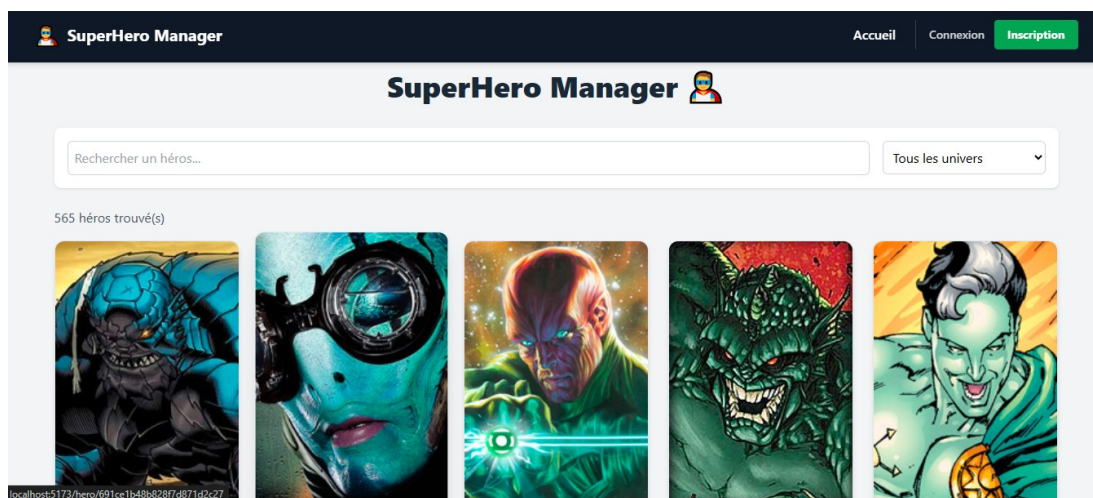
## Collection logs

Trace les actions des utilisateurs

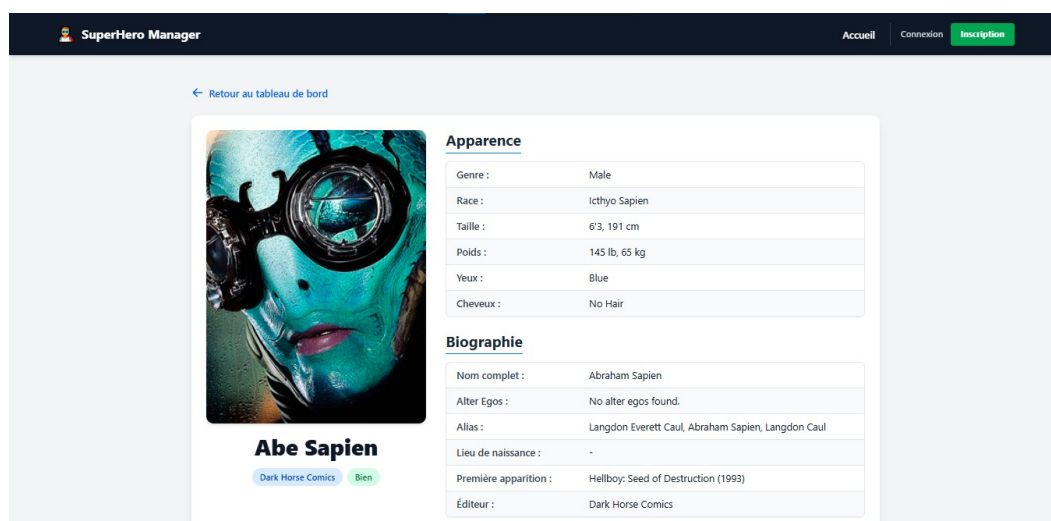
```
{
  "username": String,   Qui a fait l'action
  "action": String,    "Création", "Modification", "Suppression", ...
  "target": String,    Nom du héros concerné
  "timestamp": Date    Date de l'action
}
```

## 2.3 - Captures d'écran des interfaces

### Tableau de Bord



### Page de Détail



## Interface d'ajout et de modification (similaire)

The screenshot shows the 'Ajouter un nouveau héros' form in the SuperHero Manager application. The form is divided into three sections: 'Informations Principales', 'Biographie', and 'Apparence'. The 'Informations Principales' section includes a text input for 'Nom du Héros', a dropdown menu for 'Univers' (currently set to 'Autre'), and an 'Image' section with a 'Choisir un fichier' button and the text 'Aucun fichier choisi'. The 'Biographie' section includes text inputs for 'Nom complet (civil)' and 'Lieu de naissance'. The 'Apparence' section includes a dropdown menu for 'Genre' (currently set to 'Homme') and a text input for 'Race'. The top navigation bar includes 'Accueil', '+ Ajouter', 'Admin', 'admin ADMIN', and 'Déconnexion'.

## Interface d'inscription et de connexion (similaire)

The screenshot shows the 'Connexion' form. It features a white card with a blue 'Se connecter' button. The form includes text inputs for 'Nom d'utilisateur' and 'Mot de passe'. Below the 'Mot de passe' input, there is a link that says 'Pas de compte ? Inscrivez-vous'.

## Interface d'Administration

NOM	RÔLE	DATE D'INSCRIPTION
test	Éditeur	23/11/2025
admin2	Administrateur	23/11/2025
admin	Administrateur	23/11/2025
toto	Éditeur	18/11/2025

DATE / HEURE	UTILISATEUR	ACTION	CIBLE
23/11/2025 18:47:14	admin	Suppression	jeter
23/11/2025 18:46:54	admin	Modification	Phoenix

## Partie 3 - Rapport de projet

### 3.1 - Présentation générale

Le projet SuperHero Manager est une application web Full-Stack permettant la gestion d'un catalogue de super-héros. L'objectif principal était de concevoir une solution capable de lire, créer, modifier et supprimer des données à partir d'une interface utilisateur, tout en sécurisant l'accès via une authentification.

### 3.2 - Architecture et technologies

L'application repose sur l'architecture MERN, renforcée par l'utilisation de TypeScript.

#### Backend

- Node.js & Express : Le serveur gère les requêtes HTTP, le routage et la logique métier.
- MongoDB & Mongoose : Base de données NoSQL.
- JWT (JSON Web Token) : Utilisé pour l'authentification.
- Multer : Middleware permettant la gestion et le stockage des images uploadées par les utilisateurs.

## Frontend

- React avec vite : Framework utilisé pour créer une interface dynamique et rapide.
- Tailwind CSS : Framework CSS.
- Formik & Yup : Gestion avancée des formulaires et validation des données.

### 3.3 - Fonctionnalités réalisées

Voici les fonctionnalités réalisées durant ce TP :

#### 1. Gestion des Héros :

- Affichage sous forme de cartes réactives.
- Ajout de héros.
- Modification des données des héros.
- Suppression des héros.

#### 2. Recherche et Filtres :

- Barre de recherche dynamique par nom.
- Filtrage par Univers (Marvel, DC, Autre).

#### 3. Sécurité et Authentification :

- Inscription et Connexion sécurisées.
- Hachage des mots de passe.
- Protection des routes via Tokens JWT.

#### 4. Gestion des Rôles :

- **Visiteur** : Lecture seule.
- **Éditeur** : Ajout et Modification autorisés.
- **Administrateur** : Droit de Suppression et accès à l'interface admin.

#### 5. Journalisation (Logs) :

- Enregistrement automatique en base de données de chaque action (inscription, ajout, suppression, modification) .
- Visualisation sur l'interface admin.

### 3.4 - Difficultés rencontrées

Voici les difficultés rencontrées durant ce TP :

- problème de compilation de fiche due à des oubli de type dans les imports ou dans le code.
- corruption de mon service MongoDB suite à une mise à jour Windows , j'ai du donc refaire une installation propre de MongoDB.
- problème avec le cache navigateur pour certaines modification besoin donc de faire un rafraîchissement du cache de mon navigateur
- problème du champ image qui restait obligatoire pour la page modification de héros (Yup.mixed().optional()), j'ai du donc dans le schéma Yup ajouter .nullable() afin de ne pas rendre l'image obligatoire.

### 3.4 - Axes d'amélioration

Voici les axes d'amélioration de l'application qui pourraient être améliorés :

Par la suite, je pourrais améliorer l'interface mobile pour certains éléments (navbar, les cartes de héros, interface administrateur, ...).

je pourrais aussi ajouter que via l'interface utilisateur je puisse supprimer des utilisateurs ayant fait de mauvais changements et faire en sorte que l'administrateur remette en ordre les modifications malveillantes

## Partie 4 - Vidéo de démonstration

Voici le lien vers la vidéo de démonstration :

<https://youtu.be/AECFL4epo-c>