

SORBONNE UNIVERSITÉ - MU1IN503 APS



---

## Rapport de Projet PC3R - Micro-Projet Web : Webtoon Tracker

---

Laura RATTANAVAN, 28633935  
Tracy HONG, 21314944

[https://stl.algo-prog.info/28633935/m1\\_pc3r\\_projet](https://stl.algo-prog.info/28633935/m1_pc3r_projet)

M1 Informatique, 26 mai 2024

---

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>API Web</b>	<b>2</b>
<b>3</b>	<b>Fonctionnalités de l'application</b>	<b>4</b>
<b>4</b>	<b>Cas d'utilisation classiques de l'application</b>	<b>4</b>
<b>5</b>	<b>Données stockées et mise à jour des données</b>	<b>5</b>
<b>6</b>	<b>Architecture du serveur</b>	<b>6</b>
<b>7</b>	<b>Architecture du client</b>	<b>8</b>
<b>8</b>	<b>Requêtes et réponses client-serveur</b>	<b>19</b>
<b>9</b>	<b>Schéma du projet</b>	<b>21</b>
<b>10</b>	<b>Installation</b>	<b>21</b>
<b>11</b>	<b>Utilisation</b>	<b>22</b>
<b>12</b>	<b>Conclusion</b>	<b>23</b>

## 1 Introduction

L'objectif du micro-projet PC3R est la réalisation d'une application web. Le sujet et les fonctionnalités étant libres, notre application Webtoon Tracker facilite la recherche, la découverte et le suivi de chapitres de webtoons traduits en anglais. Elle permet en outre à ses utilisateurs de créer un compte, ajouter des webtoons qui leur intéressent en favoris et suivre les derniers chapitres et chapitres suivants par le biais d'un calendrier. La langue de l'application est l'anglais.

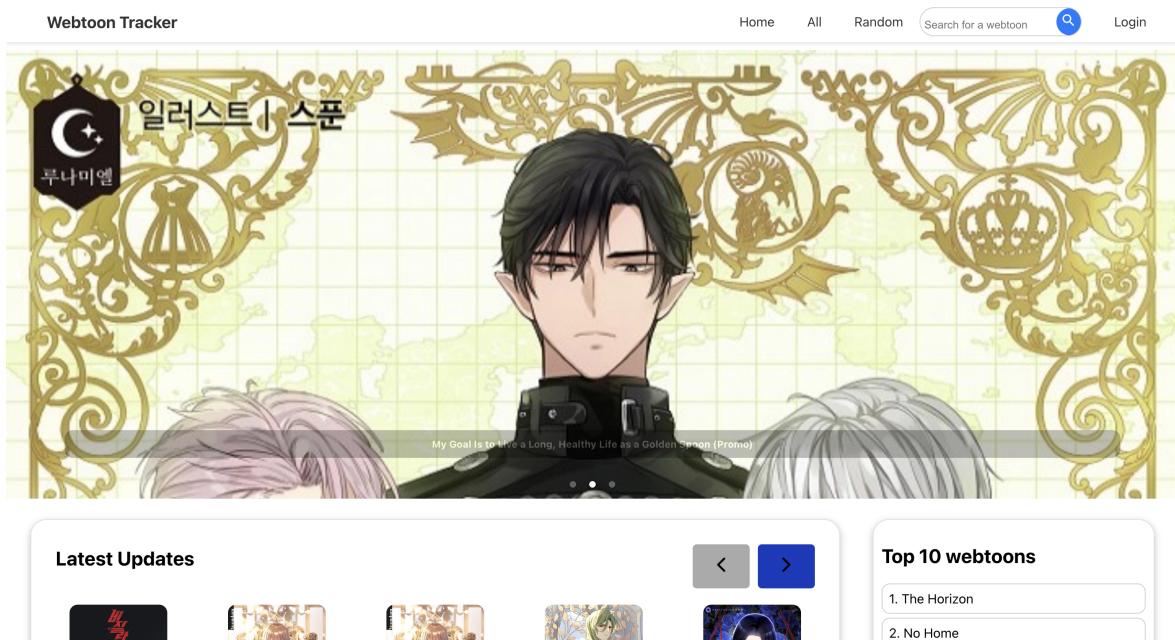


FIGURE 1 – Page d'accueil de l'application web Webtoon Tracker

## 2 API Web

Nous avons choisi d'utiliser l'API MangaDex <https://api.mangadex.org/> qui permet d'interagir avec la base de données et les fonctionnalités de MangaDex, une plateforme en ligne de lecture et partage de mangas.

Cette API est gratuite et limitée à 5 requêtes par seconde et donne accès à des données sur tout type de mangas. Étant donné que notre application se limite aux webtoons traduits en anglais, nous avons restreint les requêtes API pour ne récupérer que les données dont la langue d'origine est le coréen (ko) et la langue de traduction est l'anglais (en). De plus, le contenu est évalué comme *safe* pour filtrer les éventuels résultats indésirables. Les

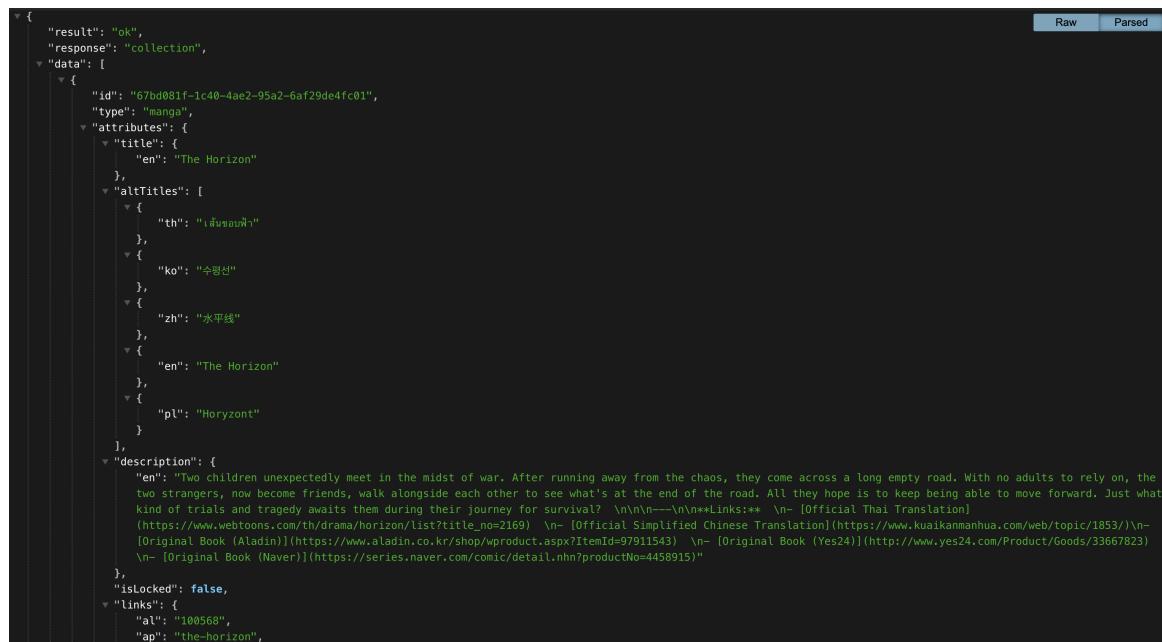
---

données sont récupérées sous format Json.

Cette API propose plusieurs types de requêtes permettant ainsi de récupérer des listes de webtoons/mangas, des informations sur un manga donné ou des chapitres, des auteurs, des images et également sur les comptes utilisateurs du site. Notre application n'utilise toutefois pas cette dernière fonctionnalité. Voici un exemple de requête API possible :

```
https://api.mangadex.org/manga?order\[rating\]=desc&originalLanguage\[\]=%ko&contentRating\[\]=%safe&limit=2
```

Les données récupérées sont entre autres une liste de 2 mangas coréens (manhwas ou webtoons) classés dans l'ordre décroissant de leur note sur MangaDex et avec un filtre de contenu *safe*. L'API fournit notamment des informations sur le titre, les titres alternatifs, la langue, la description, le dernier chapitre, la date de sortie du dernier chapitre, les tags/genres, etc. Nous manipulons ainsi beaucoup de données avec des id. Afin d'obtenir les données qui nous intéressent, nous croisons plusieurs requêtes et réponses Json pour récupérer les bonnes informations. Nous avons en effet un certain nombre de données peu utiles pour notre site.



```
{ "result": "ok", "response": "collection", "data": [ { "id": "67bd081f-1c40-4ae2-95a2-6af29de4fc01", "type": "manga", "attributes": { "title": { "en": "The Horizon", "th": "ສັບສົນທີ", "ko": "수평선", "zh": "水平线", "pl": "Horyzont" }, "description": { "en": "Two children unexpectedly meet in the midst of war. After running away from the chaos, they come across a long empty road. With no adults to rely on, the two strangers, now become friends, walk alongside each other to see what's at the end of the road. All they hope is to keep being able to move forward. Just what kind of trials and tragedy awaits them during their journey for survival? \n\n---\nLinks:**\n- [Official Thai Translation](https://www.webtoons.com/th/drama/horizon/list?title_no=2169)\n- [Official Simplified Chinese Translation](https://www.kuaikanmanhua.com/web/topic/1853/)\n- [Original Book (Aladin)](https://www.aladin.co.kr/shop/wproduct.aspx?ItemID=97911543)\n- [Original Book (Yes24)](http://www.yes24.com/Product/Goods/33667823)\n- [Original Book (Naver)](https://series.naver.com/comic/detail.nhn?productNo=4458915)" }, "isLocked": false, "links": { "al": "100568", "ap": "the-horizon" } } ] }
```

FIGURE 2 – Résultat de la requête API de l'exemple ci-dessus

Les difficultés rencontrées à cette étape ont alors été la manipulation de plusieurs types de requêtes sur cette API. Nous avons dû analyser les bonnes requêtes, jouer avec leur url et leurs paramètres pour avoir les bonnes informations, et enfin parser le Json résultat pour extraire les informations souhaitées.

---

### 3 Fonctionnalités de l'application

Les divers fonctionnalités offertes par le site sont :

1. La connexion / l'inscription / la déconnexion
2. Défilement sur la page d'accueil de 3 webtoons aléatoires favorisant la découverte de nouveaux webtoons (avec navigation possible entre les 3 images en appuyant sur les points blancs)
3. Consulter les derniers chapitres ajoutés sur la page d'accueil (affichage de 5 chapitres avec pagination pour les 5 précédents/suivants)
4. Consulter le classement des 10 webtoons les mieux notés sur la page d'accueil
5. Consulter la fiche descriptive des webtoons
6. Consulter la liste de tous les webtoons par ordre alphabétique (affichage de 10 webtoons par page)
7. Découvrir et consulter un webtoon aléatoire
8. Rechercher un webtoon avec son nom à l'aide d'une barre de recherche
9. Ajouter / retirer aux favoris des webtoons (avec un compte utilisateur connecté)
10. Suivre les chapitres des webtoons mis en favoris sur un calendrier sur la page de profil utilisateur (les 5 derniers chapitres + quelques chapitres prévus dans les semaines qui suivent)
11. Consultation des webtoons favoris à l'aide d'une page modale sur le côté droit (accès avec sous-menu / menu déroulant)
12. Remonter en haut du site avec un bouton quand l'utilisateur *scroll* en bas

---

### 4 Cas d'utilisation classiques de l'application

- Alice est une utilisatrice non connectée et sans compte. Elle attend que la page d'accueil se charge puis s'affiche et voit le défilement de 3 images de webtoon avec l'affichage de leur titre. L'image est passée au 2e sur 3 parmi les images mais elle veut revenir sur la 1e et clique sur le 1er point blanc. Elle appuie sur un des titres et affiche la page descriptive du webtoon. Elle revient sur la page d'accueil. Elle observe aussi en bas de la page les 5 dernières sorties de chapitres de webtoons. Elle clique sur la flèche droite pour regarder les 5 précédentes sorties (avant ces 5 dernières, avec la page suivante) puis elle revient en arrière (flèche gauche) pour regarder les 5 dernières. Elle clique sur l'image d'une des dernières sorties et affiche la page descriptive du webtoon associé. Elle revient sur la page d'accueil. Sur le côté en bas à droite, elle regarde quel est le top 10 des webtoons les mieux notés. Elle clique sur un des top webtoons et affiche la page descriptive du webtoon. Elle remonte en haut de la page du site et appuie sur **All** et peut consulter la liste en ordre alphabétique de tous les webtoons. Elle descend tout en bas de la page et

---

peut naviguer pour voir les 10 suivants / précédents. Elle clique sur la flèche bleue pour remonter en haut de la page. Elle appuie sur **View Details** pour consulter la fiche d'un webtoon en particulier. En haut de la page du site, elle appuie sur **Random** pour tirer un webtoon aléatoire. Elle utilise la barre de recherche en haut de la page du site pour rechercher un titre ou morceau de titre et affiche le résultat de sa recherche. Elle appuie sur **Login** en haut à droite. Elle n'a pas de compte et appuie sur **Don't have an account ? Sign up.** Elle s'inscrit et crée un compte. Un nouvel utilisateur est ajouté à la base de données du site.

- Bob est un utilisateur non connecté possédant un compte. Il arrive sur la page d'accueil et clique sur **Login** en haut à droite. Il entre ses identifiants et se connecte. Il arrive sur sa page de profil et peut consulter les 5 derniers chapitres de ses webtoons favoris, ainsi que quelques autres prédictions de chapitres de ses favoris. Il navigue sur le site et consulte une page de webtoon qui l'intéresse. Il clique sur **Add to Favorites** et ajoute un nouveau favori. Sa liste de favoris est mise à jour dans la base de données du site et il peut suivre son nouveau favori sur son calendrier. Il utilise le sous-menu à droite pour afficher sa liste de favoris. Il accède à la fiche d'un de ses favoris et clique sur **Remove from Favorites** pour retirer ce favori. La base de données est mise à jour. Il utilise le sous-menu pour se déconnecter avec **Logout**.

## 5 Données stockées et mise à jour des données

L'application web utilise MongoDB comme base de données NoSQL qui stocke les données sous forme de documents au format Bson (extension binaire de Json). Nous avons donc choisi de pas installer de base de données en local. L'API nous permet de récupérer directement et rapidement les informations.

Par ailleurs, nous avons hésité à stocker les informations des webtoons mais nous avons abandonné cette idée à cause du grand nombre de données qu'il aurait fallu stocker. Une alternative aurait été de limiter les ressources du site en stockant par exemple 1000 webtoons alors qu'il en existe des milliers sur l'API, ce qui rendrait les fonctionnalités du site moins intéressantes.

Ainsi, nous stockons uniquement les données utilisateurs : nom, email, mot de place (qui n'est toutefois pas hashé) et liste des id de webtoons favoris qui correspondent à l'id utilisé pour faire des requêtes API sur ces webtoons. À chaque ajout ou suppression d'un favori, la base de données se met à jour.

The screenshot shows the MongoDB Atlas interface. On the left, the sidebar includes sections for Overview, DEPLOYMENT, SERVICES (Database, Data Lake, Device & Edge Sync, Triggers, Data API, Data Federation, Stream Processing, Migration, SECURITY (Quickstart, Backup, Database Access, NetworkAccess, Advanced), and Goto. The main area displays the 'PROJECT 0 > DATABASES > webtoonTracker' section. The 'Collections' tab is selected, showing the 'webtoonTracker.Users' collection. It lists three documents with their IDs, names, emails, passwords, and favorite arrays. The interface includes tabs for Overview, Real Time, Metrics, Collections, Atlas Search, Performance Advisor, Online Archive, and Cmd Line Tools. Top right: VERSION 7.0.8, REGION AWS Paris (eu-west-3). Bottom right: VISUALIZE YOUR DATA, REFRESH, INSERT DOCUMENT, Reset, Apply, Options.

FIGURE 3 – Image de la base de données montrant les données utilisateurs stockées

## 6 Architecture du serveur

Au niveau du serveur, nous avons utilisé comme environnement de développement Tomcat Catalina avec intégration avec IntelliJ, ainsi que Maven pour les diverses librairies Java. Le serveur est notamment composé de Java Servlets.

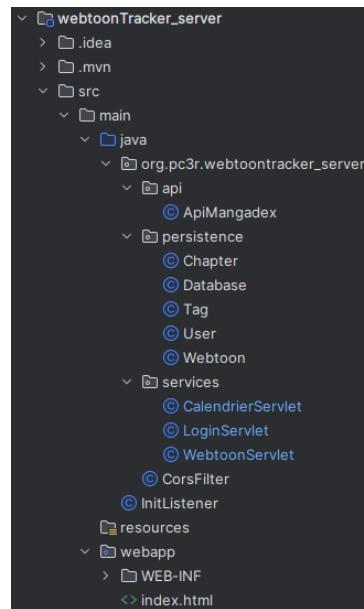


FIGURE 4 – Architecture du serveur

---

Voici une description des différents fichiers :

- listener : initialise la base
- filtre cors
- dossier api : fichier ApiMangadex pour les appels à l'API MangaDex : requête http. Nous définissons un nombre maximum de webtoons (l'API est dynamique : elle évolue) mais nous forçons le nombre maximum pour pouvoir faire nos calculs plus simplement
  - récupérer le nombre de page par rapport aux nombre de webtoons ou chapitre affiché par page sur le client (pour le client nous affichons 10 webtoons ou 5 chapitres)
  - récupérer les derniers chapitres ajoutés dans MangaDex : avec une limite
  - récupérer le top 10 des webtoons les mieux notés sur MangaDex
  - récupérer un webtoon aléatoire
  - récupérer tous les webtoons, par lots (en fonction de la page du client, par exemple 10 par 10, nous récupérons les 10 suivant avec un offset pour décaler d'un lot) : avec une limite et un offset
  - récupérer les informations d'un webtoon : avec id webtoon
  - récupérer liste après recherche d'un webtoon à partir d'un nom : avec string titre du webtoon ou une partie du titre
  - récupérer nom auteur/artiste : avec id de l'auteur/artiste
  - récupérer une liste de chapitres : avec id webtoon
  - récupérer l'image du webtoon : avec id de l'image

Chaque requête renvoie un Json. Nous avons créé des body handler qui analysent ce Json et extrait les informations utiles pour notre projet : `ofChapterList()`, `ofWebtoonList()` qui sont 2 fonctions qui vont parser le Json et retourner des listes de chapitres ou de webtoons avec les informations de la classe associée.

- dossier persistence : fichiers concernant la base de données
  - User, Webtoon, Tag, Chapter : classes Objet
  - Database : classe static singleton avec constructeur privé pour assurer que nous ayons qu'une seule base. Connexion avec la base, et opération ajout, update, delete.
    - récupérer les informations de la base pour la persistence : stocker la liste de user dans le serveur
    - vérifier si user existe dans la base pour login ou inscription
    - ajouter un user dans la base = inscription s'il n'existe pas
    - ajouter/retirer un webtoon de la liste des favoris du user
    - récupérer les favoris
- dossier service : servlet approche ressource

- 
- une servlet pour user : login en get, register en post
  - une servlet pour webtoon :
    - get pour tout type de requête vers l'API MangaDex pour récupérer des données à envoyer au client
    - post pour ajouter ou supprimer un webtoon des favoris de l'utilisateur.
- Cette servlet va traiter toutes les requêtes des clients par rapport aux webtoons pour récupérer des listes, des informations. Elle va récupérer les paramètres, données Json envoyées par le client puis appeler la méthode vers l'API MangaDex utile. en fonction des requêtes, tout le résultat de la réponse de l'api ne sera pas nécessaire pour le client alors dans ce cas, nous allons créer un Json pour retravailler le Json et extraire l'information utile.
- une servlet pour le calendrier : post seulement pour traiter les requêtes pour les event du calendrier par rapport aux derniers chapitres des webtoons favoris : avec la liste des id des webtoons fav du user

## 7 Architecture du client

Notre application client est développée en React JS. Voici un aperçu détaillé de son architecture, de ses composants et de son fonctionnement. L'utilisateur peut notamment visualiser un calendrier personnalisé basé sur sa liste de webtoons favoris.

Initialement, nous avons envisagé d'utiliser l'API de Google Calendar pour notre calendrier, mais la mise en place s'est avérée trop complexe. Nous avons donc opté pour le module "Big Calendar", qui offre des fonctionnalités similaires avec une intégration plus simple.

Nous utilisons le Context API de React pour stocker l'utilisateur et ses webtoons favoris. Le contexte inclut des fonctions de login/logout ainsi que des fonctions pour ajouter/retirer des webtoons favoris. Cela permet de mettre à jour l'état de l'application en fonction de la connexion de l'utilisateur. Certaines fonctionnalités sont masquées pour les utilisateurs non connectés.

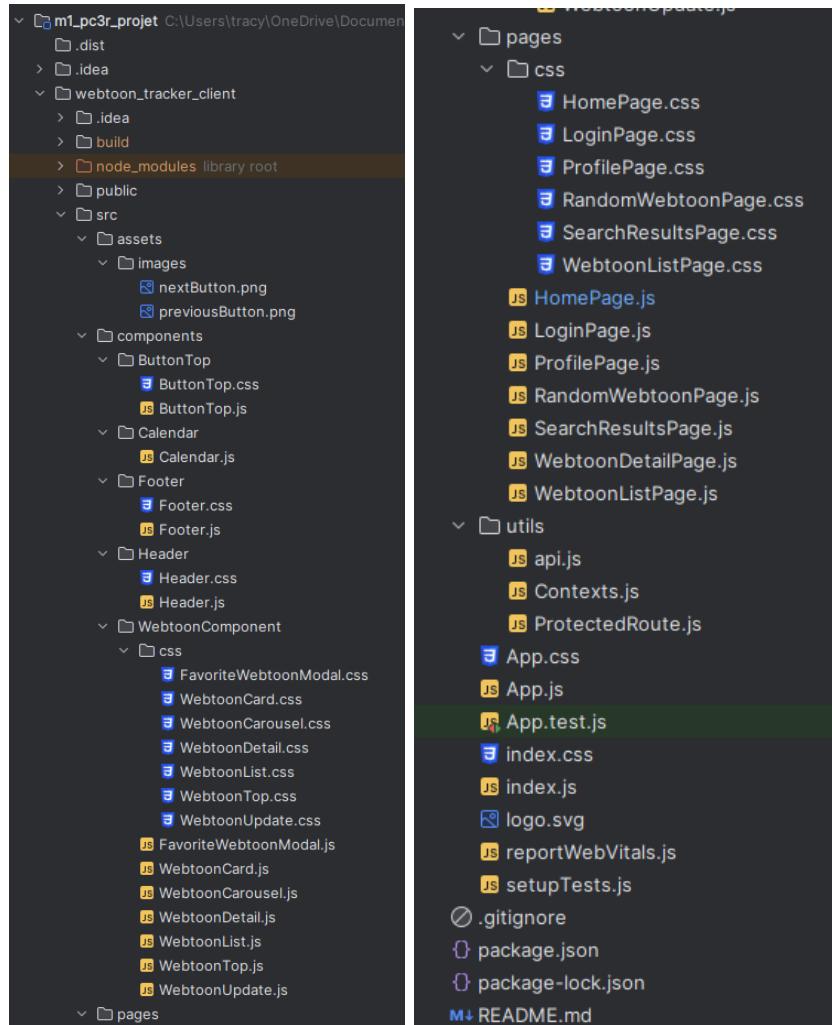


FIGURE 5 – Architecture du client

- **index.js** : Fichier principal qui lance **App.js**, le cœur de l’application.
- **App.js** :
  - Enveloppe l’application dans le contexte pour l’initialiser.
  - Gère le routage entre les pages.
  - Affiche le header et le footer sur chaque page.
- **app.css** : Fichier de styles global pour l’application.
- **assets** : Contient les images utilisées sur le site.
- **utils** :
  - **api.js** : Code factorisé pour faire des requêtes REST API vers le serveur.
  - **contexts.js** : Code pour gérer le contexte de l’application, incluant l’utilisateur et ses webtoons favoris.

- 
- **protectedRoute.js** : Gère l'accès aux pages protégées et maintient l'état de connexion de l'utilisateur même après un rafraîchissement de la page.
  - **components** : Contient des composants réutilisables pour rendre le site responsive.
    - **buttonTop** : Composant pour retourner en haut de la page avec CSS et JS associés.
    - **calendar** : Affiche le calendrier et permet de créer des événements avec des pop-ups.
    - **header** : Barre de menu en haut de chaque page, incluant des liens de navigation et des boutons de connexion/profil.
    - **footer** : Bas de page avec styles CSS et JS.
    - **webtoon** :
      - **webtoonCarousel** : Affiche un carousel responsive.
      - **webtoonDetail** : Affiche les détails d'un webtoon.
      - **webtoonCard** : Affiche les informations de base d'un webtoon avec un lien vers la page de détails.
      - **webtoonList** : Affiche une liste de webtoons.
      - **webtoonTop** : Affiche une liste de noms de webtoons avec des liens vers leurs détails.
      - **webtoonUpdate** : Affiche les mises à jour récentes des webtoons.
      - **favoriteWebtoonModal.js** : Page modale affichant les webtoons favoris, superposée sur n'importe quelle page.
  - **pages** : Contient les pages de l'application avec leurs styles associés.
    - **homePage** : Page d'accueil avec les composants **webtoonCarousel**, **webtoonTop** et **webtoonUpdate**.
    - **loginPage** : Formulaire de login/registration, redirigeant vers la page de profil après connexion.
    - **profilePage** : Page de profil avec le calendrier et la liste des webtoons favoris.
    - **randomWebtoonPage** : Affiche un webtoon aléatoire.
    - **webtoonDetailPage** : Affiche les détails d'un webtoon sélectionné.
    - **searchResultsPage** : Affiche les résultats de la recherche de webtoons.
    - **webtoonListPage** : Affiche tous les webtoons avec un système de pagination.

# Maquettes

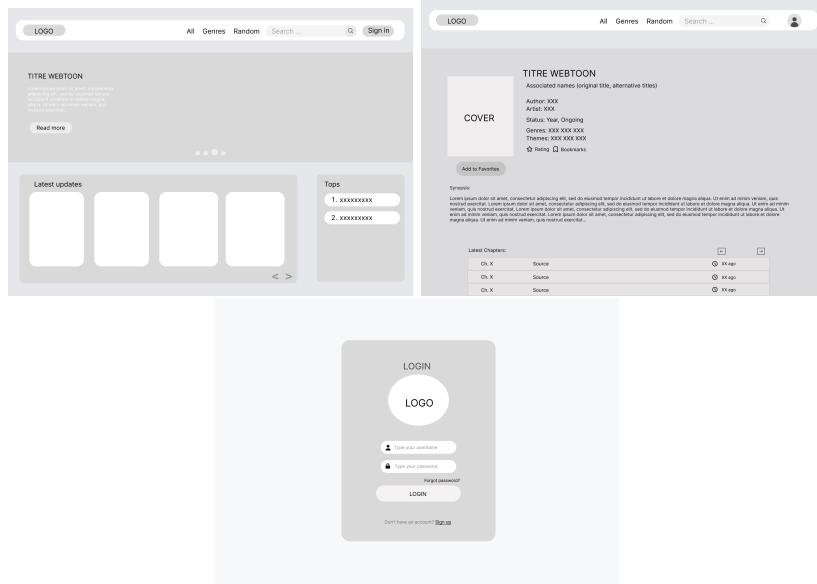


FIGURE 6 – Maquettes

## Plan du Site

Chaque page du site est structurée pour offrir une navigation cohérente et une expérience utilisateur optimale.

### Header sur chaque page

- Nom du site à gauche, redirigeant vers la page d'accueil.
- Boutons Home, All, Random et Login ou Profil, affichant les pages correspondantes.

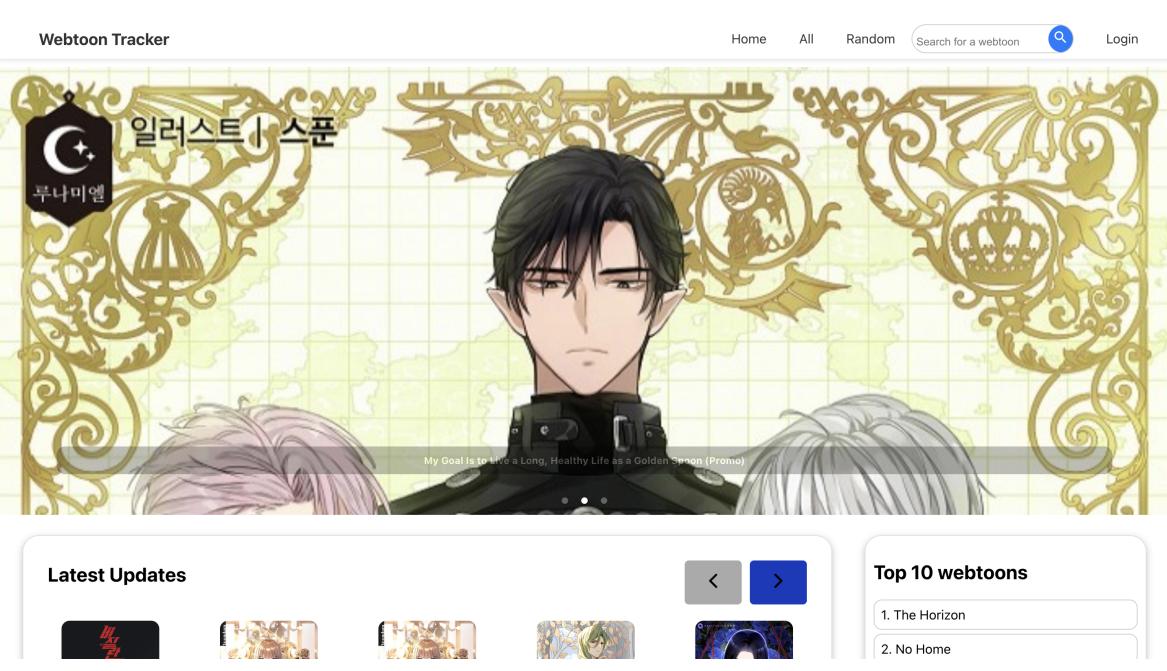


FIGURE 7 – Page d'accueil de l'application web Webtoon Tracker

## Home Page

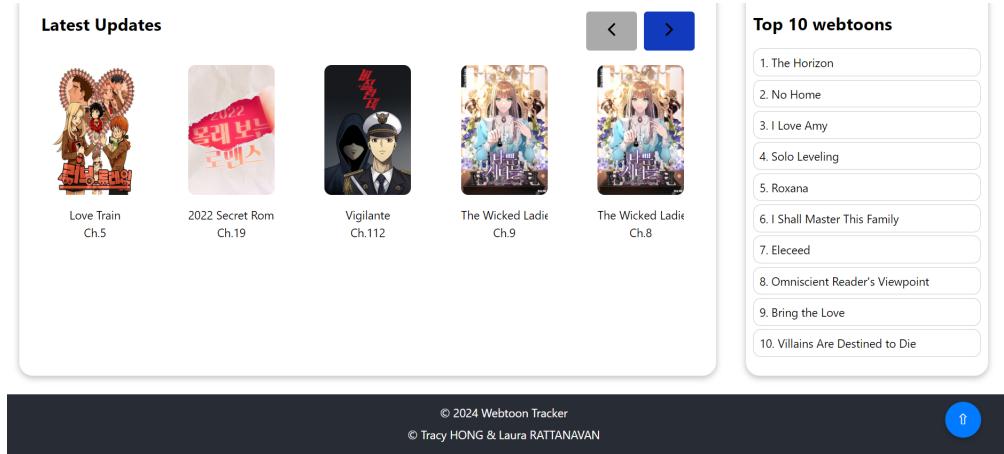


FIGURE 8 – Page d'accueil de l'application avec les listes d'updates et tops

- Carousel avec trois webtoons aléatoires.
- Last Updates pour les derniers chapitres sortis.
- Top 10 webtoons populaires.

---

## Login Page

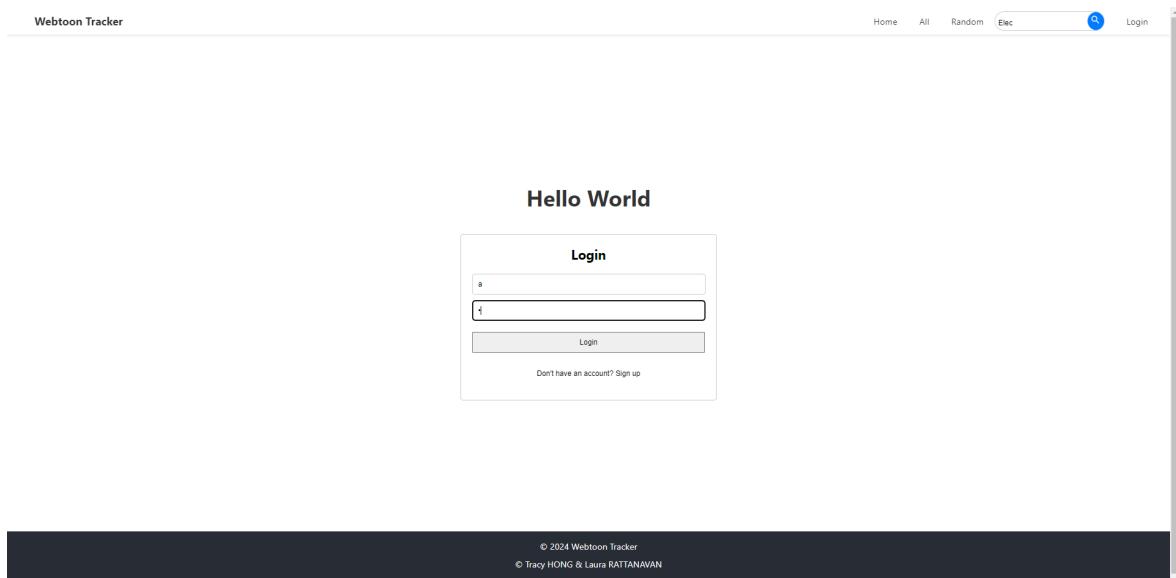


FIGURE 9 – Page profil

- Formulaire de login et d'inscription, avec alternance entre les deux via des boutons spécifiques.
- Une fois connecté, l'utilisateur est redirigé vers la page de profil et ses informations sont enregistrées dans le contexte.

## Profile Page

The screenshot shows the 'Webtoon Tracker' profile page. At the top, there's a navigation bar with links for Home, All, Random, Elec (selected), and Profile. Below the navigation is a welcome message: 'Welcome, a'. A subtitle reads 'Calendar to track all the chapters of your favorite webtoons'. The main feature is a monthly calendar for May 2024. Days are labeled from 28 to 31. Specific days are highlighted with blue boxes: May 9th ('Eleceed Ch.296'), May 10th ('Eleceed Ch.297'), May 13th ('Eleceed Ch.297'), May 14th ('The Wicked Ladies in Waiting Ch.8'), May 15th ('Eleceed Ch.298'), May 19th ('The Wicked Ladies in Waiting Ch.9'), May 20th ('The Wicked Ladies in Waiting Ch.9'), May 21st ('The Wicked Ladies in Waiting Ch.10'), May 22nd ('Eleceed Ch.299'), May 23rd ('The Wicked Ladies in Waiting Ch.11'), May 26th ('The Wicked Ladies in Waiting Ch.9'), May 27th ('The Wicked Ladies in Waiting Ch.9'), May 28th ('The Wicked Ladies in Waiting Ch.9'), May 29th ('Eleceed Ch.300'), and May 30th ('Eleceed Ch.300'). A 'Favorite Webtoons' section is also visible, featuring a thumbnail for 'The Wicked Ladies in Waiting'.

FIGURE 10 – Page profil

- Calendrier pour visualiser les mises à jour par mois, avec génération automatique des événements pour les derniers chapitres des webtoons favoris.
- Liste des webtoons favoris.
- Sous-menu pour la déconnexion ou pour visualiser les webtoons favoris.

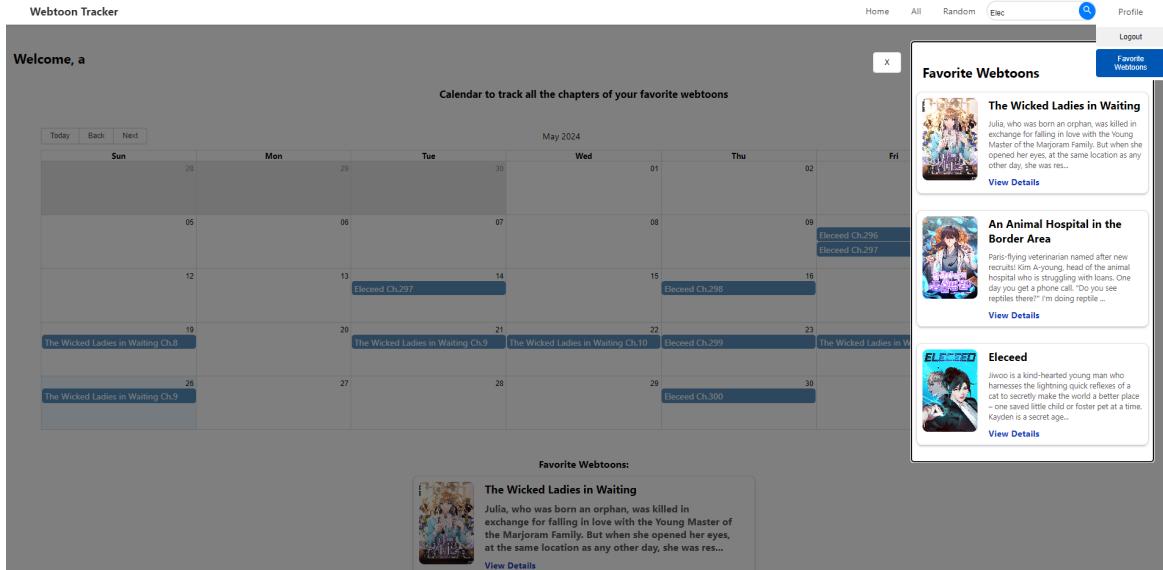


FIGURE 11 – Sous menu en pop up / page modale

## Webtoon List Page

Webtoon Tracker

Home All Random Search for a webtoon Login

## All Webtoons

**Til Debt Do Us Part**  
A man who had been assigned throughout his school days, appeared after 10 years and proposed a deal to Soo-bin, who was busy refusing many blind dates, with difficult conditions to refuse.

[View Details](#)

**(PROMO) The Returning Warrior's Alley Restaurant**  
A guard who saved the spirit world from the invasion of demons and returned to Earth after 10 years. By the way, did the returning earth suffer from the invasion of demons like the spirit world? Can G...

[View Details](#)

**[BL]Sinabro**  
When I was young, I happened to meet that child again whom I had seen at the hospital by chance. I had thought of them only as a friend, but for some reason, there was something bothering my heart...

[View Details](#)

**[Breaking News] Marriage with the Grand Duke**  
Rei Ivela became a journalist to expose corporate corruption, but in the end she reincarnated within a novel. To make matters worse, she became a bankrupt novelist who had lost everything due to he...

[View Details](#)

**#Killstagram**  
Remi Do has everything—she's super pretty, has the best brand

**>Again**  
Many generations ago, Sonogong, a 300 year old being, was the lone survivor of an attack on the village he vowed to protect. The leader of that attack only wanted Sonogong's great power for himself...  
[View Details](#)

**0 and 1**  
여기서 면역 시스템이 선 계임 속 궁한 카페티에 빙의했다. 1월자 애드워드와 대본 출동장을 해소하고 여장을 갖는 2월자로, 난 애드워드에게 살해당하기 싫다. 남의 엉에 방해하기도 싫고, 미운발로도 싫다. ‘내가 좋아하는 하정이잖아. 난 너무 당장 핑계지.’ 그레, 좋아하지 않는 게 좋을걸. 내가 고마워. 원하는 걸은 없을 테니까~!

[View Details](#)

**0-kill Assassin**  
Description not available

[View Details](#)

**0.0 MHz**  
A group of young people tormented by ghosts join hands to find a scientific explanation for their paranormal experiences – and a way out of them. In their desperate quest, they visit a haunted house...

[View Details](#)

< >

© 2024 Webtoon Tracker  
© Tracy HONG & Laura RATTANAWAN

FIGURE 12 – Pages profil

— Liste paginée de tous les webtoons, 10 par page.

## Random Webtoon Page

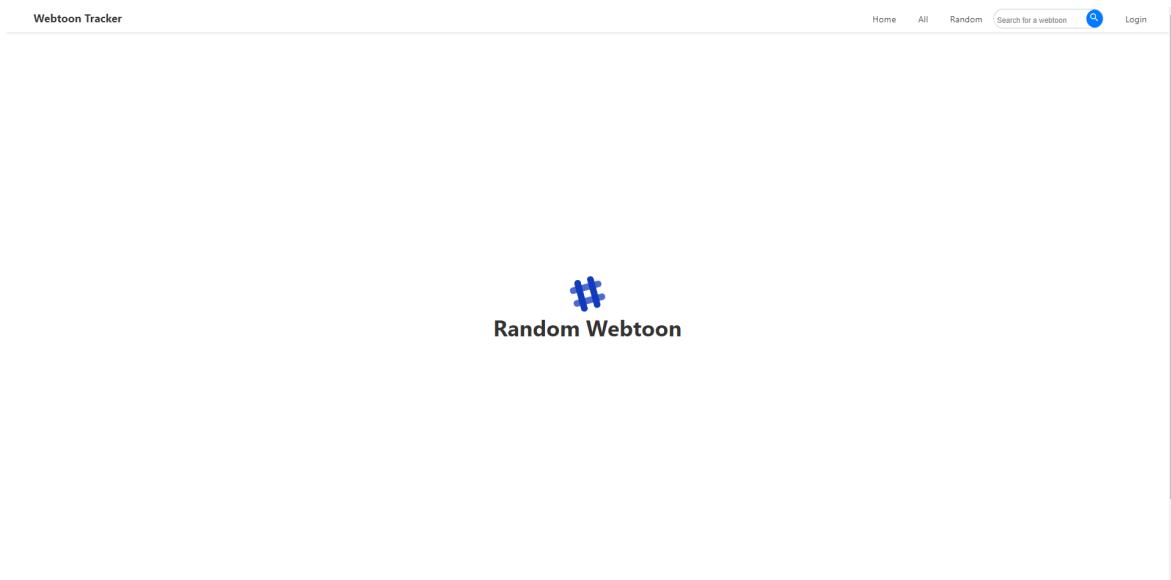


FIGURE 13 – Page en attente de chargement

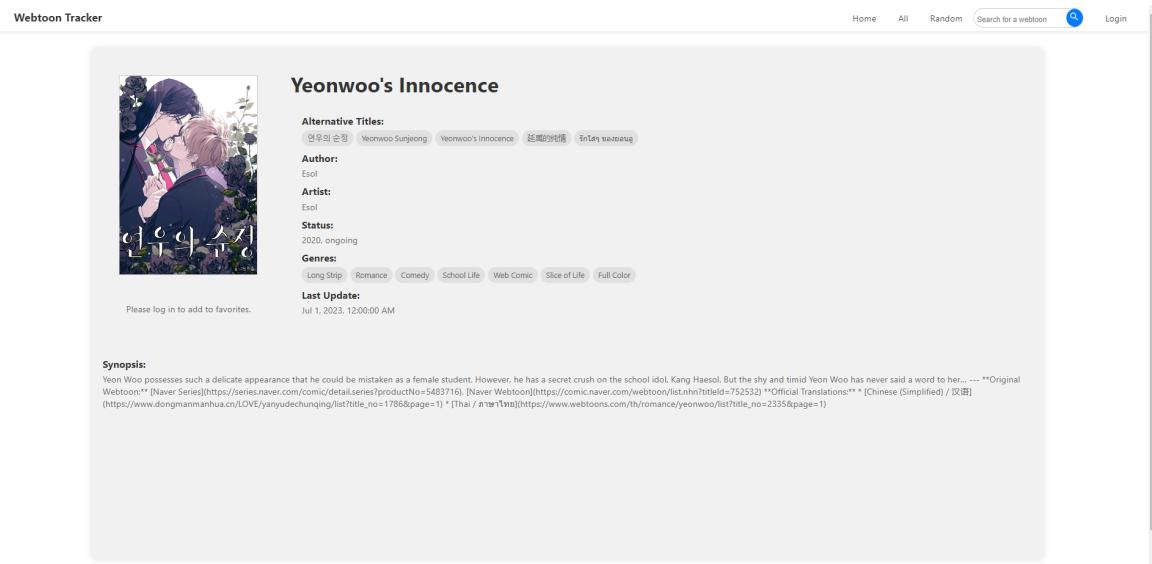


FIGURE 14 – Page d'un webtoon aléatoire

Entre les pages et l'attente de réponse du serveur, nous avons ajouté des animations de loading du module : 'react-spinners'

- Affiche les détails d'un webtoon aléatoire.

## Webtoon Detail Page

The screenshot shows a webtoon detail page. At the top, there's a navigation bar with 'Webtoon Tracker' on the left and links for 'Home', 'All', 'Random', 'Elec' (which is highlighted), and 'Profile'. Below the navigation is a search bar with a magnifying glass icon and the word 'Elec'. The main content area features a large thumbnail image of a character from the webtoon. To the right of the thumbnail, the title 'The Wicked Ladies in Waiting' is displayed in bold. Below the title, there's a section for 'Alternative Titles' with 'Bad Maids' and its Korean name. There are also sections for 'Author' (Jaya), 'Artist' (Sol), 'Status' (2024, ongoing), and 'Genres' (Historical, Long Strip, Romance, Fantasy, Supernatural, Mystery, Adaptation, Full Color). A 'Last Update' section shows 'May 24, 2024, 12:00:00 AM'. At the bottom of the page, there's a blue button labeled 'Remove from Favorites'.

FIGURE 15 – Page d'une fiche détaillée

- Affiche les détails d'un webtoon. Si l'utilisateur est connecté, un bouton pour ajouter ou retirer des favoris apparaît.

## Search Results Page

The screenshot shows a search results page for the query 'Elec'. At the top, there's a navigation bar with 'Webtoon Tracker' on the left and links for 'Home', 'All', 'Random', 'Elec' (highlighted), and 'Login'. Below the navigation is a search bar with a magnifying glass icon and the word 'Elec'. The main content area displays four search results in cards:

- Eleceed**: A card featuring a thumbnail of two characters, a brief description about a kind-hearted young man who saves little children, and a 'View Details' button.
- The King's Choice**: A card featuring a thumbnail of a king, a brief description about a one-horned king in a peaceful kingdom, and a 'View Details' button.
- Memoir of the God of War**: A card featuring a thumbnail of a warrior, a brief description about a descendant of Goryeo who remands the Central District, and a 'View Details' button.
- Haunted Electronics Shop**: A card featuring a thumbnail of a shop interior, a brief description about a mysterious horror shop run by a strange electronics shop owner, and a 'View Details' button.

At the bottom of the page, there's a dark footer bar with the text '© 2024 Webtoon Tracker' and '© Tracy HONG & Laura RATTANAVAN'.

FIGURE 16 – Page d'une recherche

- Affiche une liste de webtoons correspondant à la recherche effectuée dans la barre de recherche.

---

En suivant cette structure, l'application offre une expérience utilisateur fluide et intuitive, tout en étant facile à maintenir et à étendre.

## 8 Requêtes et réponses client-serveur

Le lien du serveur est en local :

```
http://localhost:8080/webtoonTracker_server/+suite url
```

Toute la communication se fait par JSON, ce qui signifie que toutes les réponses et données envoyées par POST au serveur sont au format JSON. Voici les url de chaque requête mentionnée précédemment : - **Login (GET)** :

```
loginServlet?name=${formData.name}&password=${formData.password}
```

- **Inscription (POST)** :

```
loginServlet
```

**Corps de la requête :**

```
JSON.stringify({  
    email: formData.email,  
    name: formData.name,  
    password: formData.password  
});
```

- **Bannière de la Page d'Accueil (GET)** :

```
webtoonServlet?get=baniere&limit=${limit}
```

- **Mise à Jour de la Page d'Accueil (GET)** :

```
webtoonServlet?get=top&limit=${limit}
```

- **Top de la Page d'Accueil (GET)** :

```
webtoonServlet?get=lastUpdates&limit=${limit}&page=${page}
```

- **Tous les Webtoons (GET)** :

```
webtoonServlet?get=all&page=${page}
```

---

- Page de Détail (GET) :

```
webtoonServlet?id=${id}
```

- Ajouter/Retirer Favori (POST) :

```
webtoonServlet
```

**Corps de la requête :**

```
JSON.stringify({  
    type: "add", or "delete",  
    userId: userData.user.id,  
    webtoonId: webtoon.id  
});
```

- Page Aléatoire (GET) :

```
webtoonServlet?get=random
```

- Recherche (GET) :

```
webtoonServlet?get=search&query=${searchQuery}
```

- Webtoon Favori (GET) :

```
webtoonServlet?id=${webtoonId}
```

- Calendrier des Événements de Chapitre (POST) :

```
calendrierServlet
```

**Corps de la requête :**

```
JSON.stringify({  
    webtoonIds: user.webtoonList  
});
```

## 9 Schéma du projet

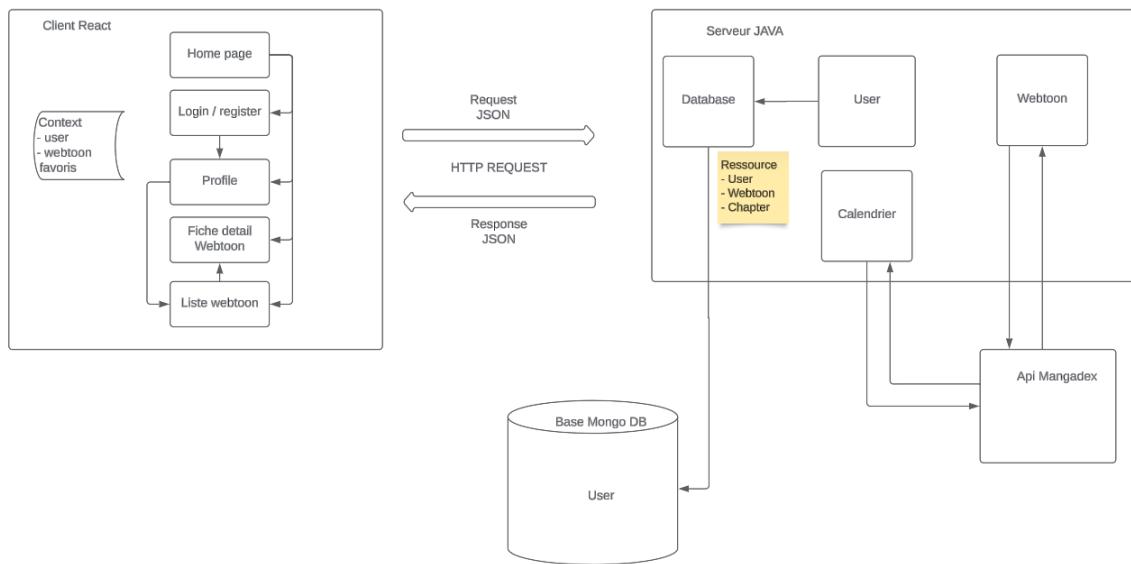


FIGURE 17 – Schéma de l’application web

## 10 Installation

Pour déployer notre application localement, vous aurez besoin de Tomcat, Maven et npm. Voici les étapes détaillées :

### Prérequis

Assurez-vous d'avoir installé Tomcat, Maven et npm sur votre machine.

### Cloner le dépôt Git

Clonez le dépôt de l'application depuis GitHub :

```
git clone https://stl.algo-prog.info/28633935/m1_pc3r_projet.git  
cd <nom_du_dossier_cloné>
```

### Installer les modules du client

Accédez au dossier du client et installez les dépendances npm :

---

```
cd client
npm install
```

## Déployer le serveur Tomcat

Assurez-vous que Tomcat est installé et configuré. Déployez votre application Java sur le serveur Tomcat en utilisant Maven ou manuellement. Cette opération ouvre une page du serveur qui nous a servi de test pour vérifier les requêtes et visualiser les réponses JSON envoyées au client.

## Vérifier et configurer le lien du serveur

Vérifiez que le serveur Tomcat est en cours d'exécution et que le lien est correct. Par défaut, le lien du serveur devrait être : [http://localhost:8080/webtoonTracker\\_server](http://localhost:8080/webtoonTracker_server). Si ce lien est différent, modifiez le lien du serveur dans le fichier `api.js` situé dans le dossier `utils` du client :

```
const BASE_URL = "http://localhost:8080/webtoonTracker_server";
```

## Lancer le client React

Accédez au dossier du client et lancez l'application React :

```
cd ../client
npm start
```

Cette commande ouvre une page web avec l'application.

En suivant ces étapes, l'application sera déployée localement.

# 11 Utilisation

Pour utiliser le site, suivez les étapes ci-dessous :

## Connexion et Inscription

Vous pouvez vous connecter avec les identifiants suivants :

- Nom d'utilisateur : a
- Mot de passe : a

Si vous n'avez pas de compte, vous pouvez vous inscrire via la page d'inscription.

- **Ajouter/Retirer un webtoon des favoris** : Vous devez être connecté pour ajouter ou retirer un webtoon de vos favoris.

- 
- **Accès aux autres fonctionnalités** : Toutes les autres fonctionnalités du site sont accessibles à tous les utilisateurs, qu'ils soient connectés ou non.

Pour plus de détails sur chaque page, consultez le plan du site.

## 12 Conclusion

Ce projet a été une expérience enrichissante et amusante. Nous avons appris énormément en créant ce site web. Étant donné que nos connaissances en React étaient limitées, nous avons souvent eu recours à ChatGPT pour nous guider sur la partie du développement en React.

Nous avons également implémenté la majorité des fonctionnalités que nous avions prévues, à l'exception de l'ajout de liste de recommandations pour les utilisateurs, de commentaires et de chapitres sur les fiches détaillées.