AntiObliviate

Manuel technique

* **Table des matières**

1. **Introduction3**
2. **Rétrospective de l’énonce3**
   1. Acteurs4
   2. Matériels et logiciels3
   3. A rendre4
   4. Description du site web4

**3. Méthode en 6 étapes4**

* 1. S’informer4
  2. Planifier4
  3. Décider4
  4. Réaliser4
  5. Contrôler4
  6. Evaluer4

**4. Plan du site web5**

* 1. Backlog5

**5. Structure et implémentation5**

**5.1 Arborescence5**

**5.2 Base de données5**

**6. Outils6**

**6.1 Github6**

**6.2 Bootstrap Studio6**

**6.3 Visual Studio Code6**

**6.4 ChatGPT et Copilot6**

**7. Fonctionnalités majeurs6**

**7.1 Fonctionnalité 16**

**7.2 Fonctionnalité 26**

**7.3 Fonctionnalité6**

**8. Planification des tests7**

**8.1 Introduction7**

**8.2 Scénarios réussi7**

**8.3 Scénario échoué7**

**9. Conclusion7**

**9.1 Difficultés rencontrées7**

**9.2 Améliorations7**

**9.3 Bilan personnel8**

**10. Annexes8**

**10.1 Table des illustrations8**

**10.2 Sources8**

**1. Introduction**

Ce texte est la documentation technique officielle d'AntiObliviate, qui se compose de deux parties :

- La documentation technique du client : qui concerne l'application web ;

- La documentation technique du serveur : qui concerne le backend, les API utilisées et la gestion de la base de données.

AntiObliviate est une application web de suivi de séries d'animes. Les utilisateurs peuvent parcourir une vaste collection d'animes, obtenue via une API externe, puis ajouter ces animes à leur liste personnelle. Ils peuvent également tenir à jour les informations concernant les épisodes vus, ainsi que la date de leur visionnage.

Le site propose également une catégorisation des animes en listes personnalisées, permettant aux utilisateurs d'organiser leurs visionnages selon leurs préférences. L'application offre une interface conviviale et intuitive qui permet aux utilisateurs de facilement naviguer et découvrir de nouveaux animes.

Le côté serveur d'AntiObliviate est construit avec Laravel, une structure PHP robuste, et utilise une base de données SQL pour stocker les informations utilisateur. L'application récupère les données d'anime via une API(https://anime-api-dsfj.onrender.com).

L'application front-end utilise le framework Tailwind CSS pour un design responsif et moderne. Les utilisateurs peuvent accéder à l'application via n'importe quel navigateur web moderne, ce qui rend l'application accessible à une large audience.

**2. Rétrospective de l’énoncé**

**2.1 Acteurs**

**Candidat :**

* **Nom :** Péan Grégoire
* **Adresse électronique :** gregoire.pn@eduge.ch

**Maître d’apprentissage :**

* **Nom :** Christian Zanardi
* **Adresse électronique :** Christian.Zanardi@edu.ge.ch

**Experts :**

* **Nom : Brian NYDEGGER**
* **Adresse électronique : briannydegger@gmail.com**
* **Nom : Alexandre WICHT**
* **Adresse électronique : alexandre.wicht@gmail.com**

**2.2 Matériels et logiciels**

* VS Code
* Pc Portable Asus
* 1 Ecran
* System d’exploitation Win 10 Pro
* WSL2 Ubuntu
* Outils de programmation PhP / MySQL / HTML / CSS
* Framework Php : Laravel
* Outils bureautiques pour la documentation (Word / Markdown)
* Outils de versioning de code GitHub (GitHubDesktop)
* Outils de backup régulier : règles du 3-2-1 (DropBox, GDrive, Usb,)
* Journal de bord (Fichier)

**2.3 À rendre**

**Maître d’apprentissage et experts :**

* Une planification initiale dès le premier jour
* Un exemplaire PDF de son mémoire avec annexe
* Documentation technique / documentation utilisateur
* Les rapports de tests (unitaires et fonctionnels)
* Code source o Le journal des « commits » (Git Log)
* Les sources du projet en format électronique
* Le résumé du TPI
* Son journal de bord (électronique)

**2.4 Description du site web**

AntiObliviate est une plateforme de suivi de séries d'animes qui offre aux fans d'anime une expérience personnalisée et facile à utiliser. Que vous soyez un novice en matière d'anime ou un otaku chevronné, cette plateforme a été conçue pour vous aider à suivre vos séries préférées et à en découvrir de nouvelles.

En s'appuyant sur les data d’une api, AntiObliviate offre un catalogue vaste et en constante évolution d'animes. Que vous préfériez l'action, l'aventure, la romance, le surnaturel, la science-fiction ou tout autre genre, vous pouvez être sûr de trouver quelque chose qui correspond à vos goûts.

La fonctionnalité clé d'AntiObliviate est la possibilité de créer votre propre liste d'animes. Vous pouvez ajouter n'importe quel anime de notre catalogue à votre liste, marquer les épisodes que vous avez déjà vus, et même noter la date de votre visionnage. De plus, vous pouvez organiser votre liste en différentes catégories pour garder une trace de ce que vous avez regardé, de ce que vous voulez regarder et de vos animes préférés.

AntiObliviate a été développé avec un accent sur la facilité d'utilisation et l'esthétique. Avec un design moderne et épuré, l'interface est à la fois intuitive et agréable à regarder. Quel que soit l'appareil que vous utilisez, vous pouvez accéder à AntiObliviate et gérer votre liste d'animes avec facilité.

En définitive, AntiObliviate n'est pas un simple site de suivi d'animes. C'est une plateforme personnalisée conçue pour améliorer votre expérience de l'univers captivant de l'anime. Que vous préfériez revisiter vos séries favorites ou que vous soyez en constante quête de nouvelles découvertes, AntiObliviate se présente comme un partenaire indispensable dans votre exploration de l'anime.

**Pages est fonctionnalités à mettre en place**

1. Page d'accueil : Cette page affiche la liste des animes récupérés via l'API. Elle sert de point d'entrée pour les utilisateurs, leur permettant de parcourir la vaste bibliothèque d'animes disponibles et de les ajouter à leur propre liste pour un suivi futur.

2. Page de la liste d'animes : Cette page présente tous les animes que l'utilisateur a ajoutés à sa liste personnelle. L'utilisateur peut trier et filtrer cette liste en fonction de plusieurs critères, tels que le genre, l'état de visionnage (en cours, terminé, etc.), et la date d'ajout.

3. Page de détails de l'anime : Sur cette page, l'utilisateur peut consulter les détails complets d'un anime spécifique. Les informations présentées incluent le titre, la synopsis, le nombre d'épisodes, la popularité, et l'image principale. Si l'anime fait partie de la liste personnelle de l'utilisateur, ce dernier a la possibilité de modifier les informations de visionnage comme la date et le dernier épisode vu.

4. Page de connexion : Cette page permet aux utilisateurs de se connecter à leur compte AntiObliviate en fournissant leur adresse e-mail et leur mot de passe.

5. Page d'inscription : Cette page offre aux nouveaux utilisateurs la possibilité de créer un compte AntiObliviate. Pour ce faire, ils doivent fournir une adresse e-mail, un nom d'utilisateur et un mot de passe. Une fois le formulaire d'inscription soumis, l’utilisateur se retrouve sur la page d’accueil.

Ces pages et fonctionnalités constituent la structure d'AntiObliviate, facilitant le suivi et la découverte de nouveaux animes pour les utilisateurs.

**3. Méthode en 6 étapes**

**1. S'informer :** C'est la première étape où l'on fait des recherches et rassemble toutes les informations nécessaires sur le projet. Cela comprend la compréhension du domaine, l'identification des besoins des utilisateurs, la collecte des exigences, la recherche des technologies appropriées, etc.

**2. Planifier :** Une fois que toutes les informations nécessaires sont rassemblées, la prochaine étape est de planifier le projet. Cela comprend la définitiondes objectifs**,** l'élaboration d'un calendrier, la répartition des tâches, l'estimation des ressources nécessaires, etc.

3. Décider : Cette étape consiste à prendre des décisions clés qui influenceront la réalisation du projet. Cela peut inclure le choix des technologies à utiliser, la définition de l'architecture du système, la sélection des membres de l'équipe, etc.

**4. Réaliser ou Exécuter :** C'est l'étape où le travail réel commence. Les tâches planifiées sont exécutées, le code est écrit, testé et déployé. Cette étape nécessite une bonne coordination et une communication efficace entre les membres de l'équipe.

**5. Contrôler :** Pendant l'exécution du projet, il est important de contrôler régulièrement le progrès pour s'assurer que tout se passe comme prévu. Cela peut impliquer le suivi des tâches, la résolution des problèmes, l'ajustement des plans si nécessaire, etc.

**6. Evaluer :** Une fois le projet terminé, il est important de l'évaluer. Cela comprend la vérification de la satisfaction des utilisateurs, l'analyse des performances du système, l'identification des domaines d'amélioration pour les projets futurs, etc.

**4. Plan du site web(analyse fonctionnelle)**Afin de bien m’organiser, savoir ce qui est attendu et ne pas avoir de mauvaise surprise, je vais expliquer les tâches et fonctionnalités en détails et de les placer sur mes 11 jours.

**4.1 Backlog**

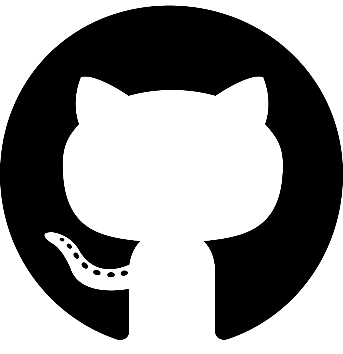
|  |  |
| --- | --- |
| **Nom** |  |
| **Description (User story)** |  |
| **Attente** |  |
| **Difficulté** |  |

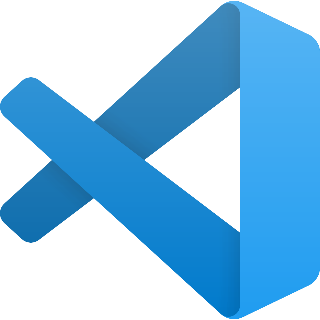
**5. Structure et implémentation**

**5.1 Arborescence**

**5.2 Base de données**

**6. Outils**

**6.1 Github**

**6.3 Visual Studio Code**

**6.4 ChatGPT et Copilot**

**7. Fonctionnalités majeures**

**7.1 [Fonctionnalité 1]**

**7.2 [Fonctionnalité 2]**

**7.3 [Fonctionnalité 3]**

**8. Planification des tests  
8.1 Introduction**

Tests fonctionnels :

Test d'inscription : Vérifier que l'utilisateur peut s'inscrire avec succès en fournissant un nom d'utilisateur, une adresse e-mail et un mot de passe valides.

Test de connexion : Vérifier que l'utilisateur peut se connecter avec succès en fournissant un nom d'utilisateur et un mot de passe valides.

Test d'affichage de la liste d'animes : Vérifier que la liste d'animes de l'utilisateur connecté s'affiche correctement avec les informations pertinentes

(titre, image, nombre d'épisodes, date de dernier visionnage et numéro du dernier épisode visionné).

Test de filtrage de la liste d'animes : Vérifier que la liste d'animes peut être filtrée avec succès en fonction des critères sélectionnés

(date d'ajout, date de visionnage du dernier épisode, catégorie).

Test de mise à jour du dernier épisode vu : Vérifier que l'utilisateur peut mettre à jour le numéro du dernier épisode vu pour un anime de sa liste avec succès.

Test de mise à jour de la date de dernier visionnage : Vérifier que l'utilisateur peut mettre à jour la date de dernier visionnage pour un anime de sa liste avec succès.

Test de changement de catégorie : Vérifier que l'utilisateur peut changer la catégorie d'un anime de sa liste avec succès.

Test de suppression d'un anime : Vérifier que l'utilisateur peut supprimer un anime de sa liste avec succès.

Test d'ajout d'un nouvel anime : Vérifier que l'utilisateur peut ajouter un nouvel anime à sa liste avec succès en utilisant l'interface de

recherche textuelle et en sélectionnant un résultat parmi les 10 renvoyés par l'API.

Tests unitaires :

Test de validation du formulaire d'inscription : Vérifier que le formulaire d'inscription valide correctement les champs

(nom d'utilisateur, adresse e-mail, mot de passe) et affiche les erreurs appropriées en cas d'entrées invalides.

Test de validation du formulaire de connexion : Vérifier que le formulaire de connexion valide correctement les champs

(nom d'utilisateur, mot de passe) et affiche les erreurs appropriées en cas d'entrées invalides.

Test de la fonction de filtrage : Vérifier que la fonction de filtrage renvoie correctement les animes filtrés en fonction des critères sélectionnés

(date d'ajout, date de visionnage du dernier épisode, catégorie).

Test de la fonction d'ajout d'un anime : Vérifier que la fonction d'ajout d'un anime ajoute correctement un nouvel anime à la liste de l'utilisateur connecté.

Test de la fonction de mise à jour du dernier épisode vu : Vérifier que la fonction de mise à jour du dernier épisode vu met à jour correctement

le numéro du dernier épisode visionné pour un anime de la liste.

Test de la fonction de mise à jour de la date de dernier visionnage : Vérifier que la fonction de mise à jour de la date de dernier

visionnage met à jour correctement la date de dernier visionnage pour un anime de la liste.

Test de la fonction de changement de catégorie : Vérifier que la fonction de changement de catégorie met à jour correctement la catégorie d'un anime de la liste. **8.2 Scénarios réussi**

|  |  |
| --- | --- |
| **Secteur** | [Quel page ?] |
| **Nom fonctionnalité** | [Nom de la fonction] |
| **Attente** | [Ce qu’elle est censée faire] |
| **Résultat** | [Ce qu’elle a fait |
| **Statut** | [Son état OK ou N.F] |

**8.3 Scénarios échoué**

|  |  |
| --- | --- |
| **Secteur** | [Quel page ?] |
| **Nom fonctionnalité** | [Nom de la fonction] |
| **Attente** | [Ce qu’elle est censée faire] |
| **Résultat** | [Ce qu’elle a fait |
| **Statut** | [Son état OK ou N.F] |

**9. Conclusion**

**9.1 Difficultés rencontrées**

[Quelles étaient les difficultés et comment je l’ai rencontré et mon sentiment]

**9.2 Améliorations**

[une listes améliorations possibles et pourquoi les ajouter]

**9.3 Bilan personnel**

[sentiment général du TPI et une revue de mon parcours du TPI]

**10. Annexes  
10.1 Table des illustrations**Image 1 [Légende de l’image] …………………………………………………………….2(page)

Image 2 [Légende de l’image] ……………………………………………………………..3(page)

Image 3 [Légende de l’image] ……………………………………………………………4(page)

**10.2 Sources**

Liste des sources utilisé pour le développement du TPI

Code source