Ergonomie & développement d’une SPA animée

**Boonder**

|  |  |
| --- | --- |
| **Auteur 1** | Kornel AMBROZEWSKI |
| **Auteur 2** | Gabriel CROKAERT |
| **Auteur 3** | Guillaume KENNES |
| **Auteur 4** | Mona NAIT MAZI |
| **Auteur 5** | Chuqi ZHANG |
| **Date** | 17.12.2023 |
| **Référence** | WEB2-2023-PROJECT-GROUP-06 |
| **Version** | 1.0 |

Contents

[1 Consignes et évaluations 3](#_Toc147308156)

[1.1 Consignes générales 3](#_Toc147308157)

[1.1.1 Création des groupes sur le site du cours 3](#_Toc147308158)

[1.1.2 Création d’un groupe sur GitHub Classroom et du web repo associé 4](#_Toc147308159)

[1.1.3 Projet 4](#_Toc147308160)

[1.2 JavaScript & Node.js : consignes techniques, timing et évaluations 6](#_Toc147308161)

[1.3 Ergonomie : consignes techniques, timing et évaluations 11](#_Toc147308162)

[2 Objectif du projet 11](#_Toc147308163)

[3 Mind map du projet 12](#_Toc147308164)

[4 Persona 12](#_Toc147308165)

[5 Axiomes de Morville 13](#_Toc147308166)

[6 Planning des tâches et cas d’utilisation 13](#_Toc147308167)

[7 Besoins techniques 16](#_Toc147308168)

[7.1 Système 16](#_Toc147308169)

[7.2 Frontend 17](#_Toc147308170)

[7.3 API 17](#_Toc147308171)

[8 Choix technologiques 18](#_Toc147308172)

[8.1 Frontend 18](#_Toc147308173)

[8.2 RESTful API 18](#_Toc147308174)

[8.3 Wireframe 18](#_Toc147308175)

[9 Conception & Implémentation 19](#_Toc147308176)

[9.1 Code repositories 19](#_Toc147308177)

[9.2 Secrets éventuels pour vos API ou base de données 19](#_Toc147308178)

[9.3 Documentation de votre API 20](#_Toc147308179)

[9.4 Déploiement de vos applications 20](#_Toc147308180)

[9.5 Code réutilisé 20](#_Toc147308181)

[10 Analyse des résultats par le groupe 21](#_Toc147308182)

[10.1 Évaluation du résultat par rapport au planning des tâches et des cas d’utilisation 21](#_Toc147308183)

[10.2 Audit ergonomique de votre projet 21](#_Toc147308184)

[10.3 Difficultés techniques rencontrées 21](#_Toc147308185)

[10.4 Conseils pour appliquer cette technologie 22](#_Toc147308186)

[10.5 Quels sont les points positifs à la manière dont s’est déroulée la collaboration au sein du groupe ? 22](#_Toc147308187)

[10.6 Quels sont les points qui seraient à améliorer pour de futures collaborations ? 22](#_Toc147308188)

[11 Analyses individuelles des résultats 22](#_Toc147308189)

[12 Présentation vidéo 22](#_Toc147308190)

[13 Revues de projets par les pairs 24](#_Toc147308191)

# Consignes et évaluations

## Consignes générales

### Création des groupes sur le site du cours

Veuillez former un groupe de 5 étudiants sur le site associé au cours : <https://e-vinci.github.io/web2>. Pour ce faire, veuillez-vous authentifier en cliquant sur l’icône . Rendez-vous sur l’onglet **Projets** (<https://e-vinci.github.io/web2/project-page>). Il est recommandé que l’attribution des **groupes** se fasse par **discussions** entre les **étudiants**. Lorsque 5 étudiants ont **un intérêt commun** pour un **projet**, ils s’inscrivent au sein d’un groupe en cliquant sur l’icône .

Pour aider à la création de groupes, il est aussi possible de vous inscrire :

* **à un groupe vide**. Cela permettra à tous d’identifier les partenaires potentiels.
* **à un groupe où il y a déjà un ou plusieurs étudiants**. Dans ce cas, veuillez-vous entretenir avec ces potentiels partenaires sur le **sujet de votre projet**.

Si nécessaire, vous pouvez vous désinscrire d’un groupe où vous n’avez pas trouvé de sujet commun dans le but de rejoindre un autre groupe. Il suffit de cliquer sur l’icône .

A la date ultime de création de groupe (Séance 9 du cours de JS, 16/10 ou 17/10 selon la série), pour les étudiants toujours en recherche de partenaires, nous faciliterons (ou imposerons si nécessaire) la création des groupes, mais pas des sujets de projet.

Une fois tous les groupes de 5 étudiants remplis, il restera maximum 4 étudiants non liés à un projet. Si nécessaire un ou plusieurs groupes de 4 étudiants seront créés.

### Création d’un groupe sur GitHub Classroom et du web repo associé

Pour chaque groupe de projet, vous allez héritez d’un web repository contenant un boilerplate via GitHub classroom.

**Veuillez passer à cette étape qu’une fois votre groupe déjà finalisé sur le site du cours.**

#### Création de l’équipe associée à un projet

Veuillez identifier le membre qui créera votre équipe sur GitHub.

Ce membre accédera à l’assignement via : <https://classroom.github.com/a/zJz7A4kY>

Ce membre devra créer une équipe reprenant le numéro de projet donné sur <https://e-vinci.github.io/web2/project-page> : si le nom de projet indiqué est **Projet N°4 : …** , il créera une équipe portant le nom **group-04** puis cliquera sur **Create team**.

Ce membre devra encore cliquer par la suite sur **Accept this assignment**.

Un web repository aura été créé pour votre équipe.

#### Joindre une équipe existante

Une fois l’équipe d’un projet créée, les autres membres accéderont aussi à l’assignement via : <https://classroom.github.com/a/zJz7A4kY>.

Ces membres joindront l’équipe existante en cliquant sur **Join** au sein de la bonne équipe. Par exemple, pour les membres du **Projet N°4**, ils cliqueront sur **Join** dans l’équipe **group-04**.

Si vous le souhaitez, vous pouvez visualiser cette vidéo qui montre [comment Joindre un GitHub Classroom Group Assignment](https://youtu.be/3TIVVCkne_0).

### Projet

Vous allez créer une SPA mettant en œuvre :

* Des sujets et technologies qui vous tiennent à cœur ;
* Une RESTful API tournant sous Node.js & Express ;
* Un frontend animé ;
* Un frontend consommant votre RESTful API et éventuellement des APIs tierces ;
* Au moins une librairie JS non vue en cours pour le frontend (anime.js ou phaser.io sont autorisées) ainsi qu’une librairie non vue pour l’API.

Pour votre frontend animé, l’animation peut être 2D, 3D, sous forme de jeux ou de simples effets visuels…

Au niveau de la présentation de votre projet, veillez à :

* Prendre en compte l’expérience utilisateur dès le début
* Optimiser le choix de vos technologies en fonction de l’expérience utilisateur
* Appliquez un maximum de théorème psycho-marketing
* Respectez les règles de Usability et auditez votre projet
* Respectez le GDPR

## 

## JavaScript & Node.js : consignes techniques, timing et évaluations

| **Tâche** | **Compétences** | **Critères** | **Deadline** | **Pt** | **Consignes** |
| --- | --- | --- | --- | --- | --- |
| Objectif du projet | C6) Documenter et présenter en vidéo le développement d'une SPA |  | **Séance 9 : 16/10 ou 17/10** |  | Donnez un nom à votre projet et décrire l’objectif de votre projet au §2 de ce document ainsi que sur <https://e-vinci.github.io/web2/project-page>, complétez :   * Le nom du projet : Projet N°X : Nom de votre projet * le champs « Description ».   Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d’aller plus loin dans votre projet. |
| Planning des tâches et cas d’utilisation | C6) |  | **Séance 12 : 24/10 ou 27/10** |  | Décrire le planning des tâches et cas d’utilisation selon les instructions données au §6.  Présenter votre planning à un enseignant, afin qu’il puisse vous aider à bien prioriser les tâches. |
| Indiquer l’URL de votre code repository | C6) |  | **Séance 12 : 24/10 ou 27/10** |  | Votre code doit être accessible par tout le monde via un web repository public qui vous sera assigné par GitHub Classroom. Cela permettra notamment aux enseignants de suivre vos avancées tout au long de votre projet. Veuillez indiquer votre URL sur <https://e-vinci.github.io/web2/project-page>.  Plus d’information aux §1.1.2 et §9.1. |
| Choix technologiques | C6) |  | **Séance 15 : 13/11 ou 14/11** |  | Compléter le §8.  Discuter de vos choix technologiques avec un enseignant. |
| Rapports individuels d’activités | C6) | Rapports de qualité *Indicateurs : formulation de qualité, analyse de qualité, respect des consignes* | **12/11**  **19/11**  **26/11**  **3/12**  **10/12**  **17/12** | **1 solo** | Des sessions individuelles de feedback sont organisées via TEAMMATES permettant à chacun de répondre à des questions dont les réponses sont confidentielles ou anonymisées au sein d’un groupe. Des e-mails seront envoyés vous invitant à compléter un formulaire hebdomadaire, à compléter pendant le WE.  À partir de la 2ème soumission, tout formulaire hebdomadaire non complété amènera à une pénalité individuelle de 0.5 point.  Si vous manquez deux soumissions, vous aurez l’obligation de montrer que vous êtes actif sur le projet sous risque d’être écarté du projet. |
| Soumission du rapport de groupe | C6) | Idem | **17/12** | **1** | Compléter le §10 ainsi que tous les paragraphes qui n’auraient pas été finalisés de ce document.  Soumettre ce document, via Moodle (un devoir sera créé) ainsi que dans le répertoire **/report** de votre repo.  Effacer toutes les consignes mises *en grisé* dans ce document avant de soumettre ce rapport sur Moodle. |
| Soumission de la vidéo | C6) | Vidéo de qualité  *Indicateurs : présentation du projet de qualité, analyse de qualité, respect des consignes* | **17/12** | **2** | Présenter votre projet selon les exigences du §12. |
| Soumission du code du frontend | C2) Création d'IHM pour SPA  inclus :  C5 : Intégrer au développement d'une SPA des technologie non vues en cours  inclus si nécessaire :  C3) Sécurisation de SPA | Qualité de l’IHM produite  *Indicateurs : esthétique, fonctionnel, codage de qualité, respect des consignes, ambitieux & original, utilisation d’une librairie pour l’IHM non vue en cours* | **17/12** | **8** | Réaliser un frontend et un backend de Qualité : Code bien structuré, UI et UX de qualité, API bien documentée (documentation des opérations de votre API, requêtes permettant de tester votre API…).  Être ambitieux et original.  Démontrer une appropriation personnelle du code (via commentaires dans le code, discussion lors des cours...).  Respecter les spécifications techniques décrites dans ce document.  Déployer votre frontend et votre backend chez un provider gratuit.  NB : votre RESTful API doit être un minimum différente des APIs fournies dans les démos du cours de JS. |
| Soumission du code du backend | C1 : Création de services web  inclus :  C5 : Intégrer au développement d'une SPA des technologie non vues en cours  inclus si nécessaire :  C3 : Sécurisation de SPA | Qualité du web service produit  *Indicateurs : fonctionnel, codage de qualité, respect des consignes, ambitieux & original, utilisation d’une librairie pour le service web non vue en cours* | **17/12** | **5** |
| Déploiement tant de votre frontend que backend | C4) Déploiement d'applications web | Déploiement de la SPA sur le cloud  *Indicateurs : fonctionnel, performances de chargement acceptables* | **17/12** | **2** |  |
| Réaliser un minimum de 5 revues sur le site web | C7) Analyser le développement de SPA faites par des pairs | Revues de projets compréhensibles & constructives  *Indicateur : présence d’un minimum de 5 revues* | **Avant examen de 1ère session** | **1**  **solo** | Via <https://e-vinci.github.io/web2/my-reviews-page>, vous devez revoir les vidéos de présentation de 5 groupes (sauf le vôtre), exécuter leurs applications, et fournir votre critique de chacun de ces projets. Vous pourrez fournir la critique d’autant de projets que vous le souhaitez. Plus d’info sur la revue de projet au §13. |
|  | **TOTAL POINTS** |  |  | **20** | Il est à noter que des membres d’un même groupe pourront être côtés différemment en fonction de leur engagement sur le projet. L’engagement d’un étudiant est visible via les rapports individuels d’activités (outil TEAMMATES)via GitHub (GitHub Project, Issues, Milestones, commits…) et lors des sessions de cours.  Les étudiants non actifs risquent d’être écarté du projet, spécialement s’ils ne soumettent pas leurs rapports individuels.  Les étudiants n’ayant pas réalisé au moins un use case significatif seront considérés inactifs.  Les étudiants n’ayant pas participés significativement au projet recevront d’office une lourde pénalité au niveau de leurs points, voire un 0/20. |

## Ergonomie : consignes techniques, timing et évaluations

Les « deadlines » données ci-dessous sont les dates où au plus tard l’avancement des tâches doivent être présentables à un enseignant pendant le cours.

| **Compétence** | **Tâches** | **Deadline** | **Points** | **Consigne** |
| --- | --- | --- | --- | --- |
| Reporting & présentation | Objectif du projet | **17/10** |  | Décrire l’objectif de votre projet au §2 de ce document.  Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d’aller plus loin dans votre projet. |
| Conception | Définir la vision marketing | **5/11** | 6 | Décrire le Mind map du projet. Créer le persona de (s) l’utilisateur (s) ciblé (s) par le projet. Répondre aux axiomes de Morville. |
| Analyse d’applications web | Architecture UX | **5/11** | 4 | Construire les wireframes détaillés de votre application. |
|  | Analyse des résultats et rapport associé | **17/12** | 2 | Auditez votre projet et vérifiez le respect des règles GDPR. |
|  | Présentation vidéo | **17/12** | 8 | Présenter votre projet en intégrant l’expérience utilisateur. |
|  | **TOTAL** |  | **20** | Il est à noter que différents membres d’un groupe pourront être côtés différemment en fonction de leur engagement sur le projet visible lors des sessions de cours. |

# Objectif du projet

Notre projet s’appelle Boonder. Il s’agit d’une application de rencontres entre livres et humains. Son principe est le suivant : un utilisateur (internaute ayant créé un compte sur le site) se voit proposer différents livres, qu’il est libre de « like » ou non ; selon son choix, le livre sera considéré comme un match et ajouté à la liste de recommandations personnalisées de l’utilisateur, ou simplement mis de côté, et on passera au suivant. Dans le but d’affiner la sélection de livres proposés à l’utilisateur, ces derniers possèdent des scores dans trois catégories : « sluttiness », « fluffiness » et « darkness ». L’utilisateur lui-même est invité à remplir un quizz à son inscription, dont le résultat déterminera quelle catégorie de livres est la plus pertinente pour lui (pour davantage d’explications sur ces catégories et ce qu’elles représentent, la page About us de Boonder est à votre disposition).

Le but de Boonder est simplement de fournir des idées de nouvelles lectures à ses utilisateurs, et ce d’une façon amusante et inhabituelle. Aucun type d’organisation n’est particulièrement pris en charge par notre application ; elle s’adresse à des particuliers, même si, évidemment, toute entreprise ou association serait libre de l’utiliser.

Ce projet nous tient à cœur car il s’agit d’une idée un peu déjantée qui nous est venue lors d’une discussion entre amis à l’école. Le fait de pouvoir réellement l’implémenter en un site web est donc pour nous une grande victoire. De plus, nous sommes nombreux dans le groupe à apprécier la lecture, aussi le sujet central de Boonder nous plaît-il beaucoup.

# Mind map du projet

Voir***/ergonomics/marketing-view***sur notre dépôt Git ; le document concerné s’intitule mindmap.pdf

# Persona

Voir***/ergonomics/marketing-view***sur notre dépôt Git ; les documents concernés s’intitulent persona1.docx et persona2.docx

# Axiomes de Morville

Voir***/ergonomics/marketing-view***sur notre dépôt Git ; le document concerné s’intitule axiomes\_morville.docx

# Planning des tâches et cas d’utilisation

Nous avons créé différentes Issues sur notre projet GitHub, de différents niveaux de priorités. Chacune a été assignée à un des membres de notre équipe.

Voici l’URL vers notre GitHub Project public : <https://github.com/e-vinci/web2-2023-project-group_06>

Il est possible, dans l’onglet Issues, de voir quelles ont été nos différentes tâches et à quel membre de l’équipe nous avons attribué chacune d’entre elles.

Pour plus de clarté, nous listons également ci-dessous les fonctionnalités importantes, associées au membre de l’équipe qui s’est principalement chargé de leur développement et, pour chacune d’entre elles, d’une brève explication :

1. Créer un profil – Guillaume

*Page « sign up »*

1. Affichage des infos des livres – Chuqi

*Sur la page d’accueil, permet de visualiser les profils des livres*

1. Possibilité de like/dislike sur un livre – Chuqi

*Sur la page d’accueil, boutons permettant à l’utilisateur de marquer ses préférences*

1. Se connecter – Guillaume

*Page « log in »*

1. Gestion des livres – Gabriel

*Création des profils des livres en rassemblant des informations les concernant*

1. Quizz pour déterminer un match – Mona

*Qcm permettant de cataloguer l’utilisateur et ainsi d’affiner les propositions qui lui seront faites*

1. Consulter liste des matchs – Gabriel

*Page permettant à l’utilisateur de voir les livres qui lui sont recommandés*

1. Animations – Mona

*Apparition d’un cœur lorsqu’un utilisateur like un livre*

A noter que ceci correspond à ce qui est indiqué sur GitHub, mais en pratique, les points 5 et 8 ont été interchangés (Mona s’est chargée de la gestion des livres et Gabriel de l’animation).

De plus, Kornel a contribué aux points 1, 2, 4 et 7, les fonctionnalités qui lui avaient été initialement assignées ayant été par la suite abandonnées par l’équipe.

# Besoins techniques

## Système

TRS01 : Vous devez développer une Single Page Application (SPA) à l’aide de JS et Node.js.

TRS02 : Votre RESTful API doit être indépendant de votre frontend ; vous aurez donc deux applications distinctes, une pour le frontend et l’autre pour la RESTful API.

TRS03 : Vous devez utiliser GitHub sur votre projet afin de gérer le développement de chacun des membres d’une équipe.

*Nous vous recommandons d’appliquer un workflow vu dans votre cours de DevOps : pour chaque cas d’utilisation / feature que vous développez, essayez de créer une branche correspondante. De plus, il serait intéressant que vous mettiez en œuvre des revues de code au sein de votre projet via des Pull Request sur Github.*

## Frontend

TRF01 : Votre frontend doit utiliser Webpack en tant que package bundler.

TRF02 : Le frontend, développé en HTML / CSS (bootstrap ou autre) / JavaScript, doit consommer au moins une de vos RESTful API.

*Votre frontend peut consommer des API externes, des APIs que vous n’avez pas développées vous-même (e.g. API de youtube, de google maps…)*

TRF03 : Votre frontend doit mettre en œuvre une librairie JS externe, ou l’API Canvas, afin de réaliser une animation.

*L’animation peut prendre la forme d’une animation 2D, 3D ou d’un jeu vidéo.   
Attention à ne pas juste offrir une minuscule animation à l’aide d’une librairie ne demandant aucun code JS, comme certaines librairies mettant tout en œuvre à l’aide de CSS.*

TRF04 : Votre frontend doit mettre en œuvre au minimum une librairie JS non vue en cours.

*Anime.js est autorisé pour votre animation.*

TRF05 : Votre frontend doit respecter les droits d’auteurs, que ça soit pour les éventuels sons, images, vidéos, librairies et morceaux de codes utilisés. Cela est de votre responsabilité et non pas de celle de vos enseignants.

TRF06 : Vous devez déployer votre frontend sur GitHub Pages ou d’autres providers gratuits supportant votre application.

## API

TRA01 : Vous devez créer une RESTful API afin d’offrir des opérations sur des ressources utiles à votre projet.

*La RESTful API ne peut pas être uniquement un « copier/coller » de ressources offertes dans le cours (notamment les ressources users et auths). Vous pouvez utiliser les ressources offertes dans le cours, mais vous devez y apporter des ajouts significatifs.*

TRA02 : Votre RESTful API doit mettre en œuvre au minimum un package non vu en cours.

TRA03 : Vous devez documenter les opérations de votre API conformément aux conventions REST.   
*Vous pouvez documenter votre API soit sous forme de tableau, comme vu dans le cours, soit à l’aide d’outils tel que Swagger.*

TRA04 : Les tests de votre API, les requêtes HTTP, doivent être données au sein de votre projet. Pour chaque opération de votre API, il doit exister au minimum une requête HTTP associée.

TRA05 : Votre API doit respecter les droits d’auteurs, que ça soit pour les éventuelles librairies utilisées, les morceaux de code, les sons, images, vidéos… Cela est de votre responsabilité et non pas de celle de vos enseignants.

TRA06 : Vous devez déployer votre backend sur Azure ou d’autres providers gratuits supportant votre application.

# Choix technologiques

## Frontend

Pour les animations, nous avons utilisé la librairie anime.js.

Voici le lien vers le site : [*https://animejs.com*](https://animejs.com)

## RESTful API

Nous avons utilisé les packages suivants dans notre API :

* BCrypt : pour crypter les mots de passe des utilisateurs dans la base de données. Voici le lien : <https://www.npmjs.com/package/bcryptjs>
* Pg : pour l’utilisation de notre base de données PostgreSQL : <https://www.npmjs.com/package/pg>
* Multer : cela nous a été utile pour les redirections vers d’autres pages du site. Voici le lien : <https://www.npmjs.com/package/multer>

## Wireframe

Voir***/ergonomics/wireframes***sur notre dépôt Git ; le document concerné s’intitule Boonder\_wireframe.pptx

# Conception & Implémentation

## Code repositories

Voici l’URL vers notre GitHub Project public : <https://github.com/e-vinci/web2-2023-project-group_06>

## Secrets éventuels pour vos API ou base de données

La clé secrète que nous avons utilisée est visible dans ce fichier : \web2-2023-project-group\_06\api\routes\auths.js

## Documentation de votre API

Tableaux représentant les opérations de notre API :

|  |  |  |
| --- | --- | --- |
| **URI** | **Méthode HTTP** | **Opération** |
| /books | GET | READ ALL BOOKS : lit et affiche les informations de tous les livres de la base de données |
| /books/add | GET | ADD BOOK : ajoute un livre à la base de données |
| /books/ :id | GET | READ ONE BOOK : affiche les informations du livre dont l’id est passé en paramètre |
| /listUsers | GET | GET LOG : affiche tous les utilisateurs |
| /listUsers/ :id | GET | READ ONE USER : affiche uniquement l’utilisateur dont l’id est passé en paramètre |
| /login | POST | LOGIN USER : affiche les données de l’utilisateur qui s’est connecté |
| /match/ :id | GET | READ MATCHES : affiche tous les livres avec lesquels l’utilisateur dont l’id est donné en paramètre a matché |
| /profilePage | POST | UPDATE PROFILE : permet de créer un nouvel utilisateur avec les informations qu’il a encodées, ou de modifier un profil existant |
| /quizz | GET | SET USER TYPE : enregistre le type et le score de l’utilisateur en fonction de ses résultats au quizz |
| /swipe | GET | GET IMAGE : affiche l’image d’un livre afin que l’utilisateur puisse swiper |
| /swipe | POST | MATCH : ajoute le livre à la liste des matches de l’utilisateur s’il y a eu match |
| /uploadProfilePicture | POST | SAVE PROFILE PICTURE : enregistre la nouvelle photo de profil dans la base de données |
| /createUser | POST | CREATE USER : ajoute un nouvel utilisateur à la base de données s’il n’existe pas déjà |

Les requêtes HTTP se trouvent dans : **/api/REST Client**

## Déploiement de vos applications

URL de notre frontend déployé : <https://chuqi-zhang-vinci.github.io/>

URL de notre RESTful API déployée : <https://boonder.azurewebsites.net/>

## Code réutilisé

## Nous n’avons pas réutilisé directement de code provenant d’un tiers, hormis le boilerplate du projet, qui nous était déjà fourni par les professeurs.

Notre seule source extérieure de code est chatGPT : <https://chat.openai.com/>

Nous l’avons essentiellement utilisé pour résoudre des problèmes précis que nous rencontrions lors du développement ; par exemple, le calcul du score du quizz et la sauvegarde du résultat ou la gestion des token.

# Analyse des résultats par le groupe

## Évaluation du résultat par rapport au planning des tâches et des cas d’utilisation

Malheureusement, nous avons manqué de temps pour implémenter certaines des fonctionnalités que nous aurions aimé mettre en place. Ainsi, nous avons dû abandonner tout ce qui concerne l’ajout de livres par les utilisateurs (issues 11 à 14 dans Git : « client propose un nouveau livre », « admin voir proposition », « admin valider propositions », « admin refuser les propositions ») et le filtre (issue 10 : « filtrer par critères »). Nous n’avons donc pas réussi à atteindre les objectifs que nous nous étions fixés initialement, mais avons quand même accompli tout ce qui était nécessaire pour que le site soit fonctionnel.

De plus, nous n’avons finalement pas suivi la répartition des tâches que nous avions prévue à la base. En effet, de part l’abandon du développement des fonctionnalités citées ci-dessus, certaines tâches ont été réassignées. Nous avons également dû tenir compte des capacités de chacun ; en effet, certaines personnes ont eu besoin d’aide pour terminer leur part du travail.

## Audit ergonomique de votre projet

Selon le modèle VPTCS :

* Visibilité : ce critère n’est pas très pertinent dans le cadre de ce projet ; nous avons simplement déployé notre frontend et notre API (voir point 9.4 de ce rapport)
* Perception : d’un point de vue ergonomique, Boonder a comme point fort son utilisation du swipe, avec lequel de nombreux utilisateurs sont familiers et qui s’inscrit dans la lignée ergonomique des applications de rencontres. La présence d’images pour représenter chaque livre rend aussi l’expérience d’utilisateur plus agréable. La navigation sur le site est également assez aisée. Par contre, il est vrai que nous n’avons pas eu beaucoup de temps à consacrer au Web Design, aussi y-a-t-il des améliorations à faire dans ce domaine.
* Technique : le site fonctionne correctement. Du point de vue de la sécurité (GDPR), nous n’avons pas mis grand-chose en place. Les mots de passe sont cryptés dans la base de données, mais nous n’avons rien implémenté en ce qui concerne la gestion des cookies et les préférences des utilisateurs à ce niveau.
* Contenus : les résumés des livres présentés sont originaux, ainsi que le ton général du site, notamment sur la page « About us ». La rédaction a été faite en anglais pour une plus grande accessibilité.
* Services : ce critère n’est pas très pertinent dans le cadre de ce projet.

En résumé, notre site manque d’ergonomie pour certains points importants, tels que la sécurité et l’aspect visuel. Cependant, d’autres points (l’originalité du contenu, sa qualité et la navigation) compensent cela.

## Difficultés techniques rencontrées

Nous avons rencontré les difficultés suivantes lors de la programmation de notre site web :

* Difficultés à utiliser les token ; nous avons réglé ce problème en révisant en détails le cours de JavaScript et avec le soutien de chatGPT.
* Implémentation du quizz : nous avons eu du mal à calculer le score de l’utilisateur à partir de ses choix dans le formulaire « radio ». Ce problème a également été réglé grâce au cours (un exemple similaire nous a servi de modèle) et chatGPT.
* Base de données : cela n’a pas été évident pour nous de faire en sorte qu’elle soit accessible pour tout le monde, en ligne. Nous avons pour cela utilisé le site ElephantSQL : <https://www.elephantsql.com/>
* Lien entre frontend et API : il nous a fallu du temps pour réussir à implémenter cela correctement. Heureusement, le cours nous a permis de comprendre l’utilisation de fetch.
* Swipe : nous avons rencontré quelques problèmes techniques pour cette fonctionnalité, notamment le fait que le swipe sur un élément ajoutait également l’élément suivant à la liste des livres likés. Malheureusement nous n’avons pas réussi à résoudre ce problème.

## Conseils pour appliquer cette technologie

Comme précisé dans le paragraphe précédent, l’utilisation et l’accessibilité à la base de données ont été un défi pour nous. Nous avons donc utilisé ElephantSQL, dont le lien est disponible au point 10.3 de ce rapport.

Nous conseillons également la librairie anime.js pour la réalisation d’animations simples. Elle a été assez facile à prendre en main et nous a permis d’atteindre nos objectifs en termes d’animation pour Boonder.

Pour les images libres de droit de nombreuses couvertures de livres, nous pouvons recommander le site suivant : <https://openlibrary.org/books> ; bien que la plupart des images que nous y avons trouvé soient des photos prises par des particuliers de leur exemplaire du livre en question, elles ont l’avantage d’être libres de droits et ne sont pas trop moches.

## Quels sont les points positifs à la manière dont s’est déroulée la collaboration au sein du groupe ?

Nous avons eu une bonne communication entre les membres du groupe, et bien que tout le monde n’ait pas été capable de contribuer de la même manière au développement du projet, chacun a fait de son mieux. Les tâches ont été plutôt bien réparties selon les points forts de chacun (par exemple, Mona n’étant pas très douée en JavaScript, elle a contribué davantage à d’autres aspects du travail, comme la recherche des informations concernant les livres, la mise en place du quizz et la rédaction du présent rapport). De plus, certains des membres se connaissaient déjà depuis l’année dernière, mais ce n’était pas le cas de tout le monde, aussi nous semble-t-il pertinent de mentionner que ce projet a aussi été l’occasion pour nous de créer de nouvelles amitiés.

## Quels sont les points qui seraient à améliorer pour de futures collaborations ?

Tout d’abord, bien que, comme mentionné au point précédent, la communication au sein du groupe était bonne, il serait profitable de mettre au clair plus vite ce que chacun a en tête en termes d’attentes concernant le projet. En effet, nous nous sommes rendu compte seulement à un stade assez avancé du développement que nous n’avions pas tous les mêmes ambitions concernant Boonder, et que les attentes peut-être trop élevées de certains membres avaient pu ralentir leur travail, et par conséquent l’intégralité du développement du projet.

Un autre point intéressant à améliorer serait notre organisation personnelle en tant que groupe. En effet, certains membres (comme Chuqi) avaient une très bonne organisation individuelle, et tiraient donc le groupe vers l’avant. Cependant, d’autres membres du groupe avaient plus de mal à se fixer des deadlines personnels, et ont donc procrastiné un peu trop pour le bien commun. Heureusement, ce souci a pu être réglé vers la fin du déroulement du projet, simplement en exprimant des dates limites, autres que celles fournies par les professeurs, pour avoir fini certaines tâches spécifiques. Cela a beaucoup aidé certaines personnes à finir leur travail dans un délai plus serein.

# Présentation vidéo

Lien vers la vidéo YouTube : <https://youtu.be/BvFzAIrE18I>