Ergonomie & développement d’une SPA animée

**Red Devils.MEM**

|  |  |
| --- | --- |
| **Auteur 1** | Barhdadi Adam |
| **Auteur 2** | Kelmendi Leon |
| **Auteur 3** | Arslan Kubra |
| **Auteur 4** | Bekkari Ibrahim |
| **Date** | 09.12.2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-09 |
| **Version** | 1.0 |

Contents

[1 Consignes et évaluations 3](#_Toc115852290)

[1.1 Consignes générales 3](#_Toc115852291)

[1.1.1 Création des groupes sur le site du cours 3](#_Toc115852292)

[1.1.2 Création d’un groupe sur GitHub Classroom et du web repo associé 4](#_Toc115852293)

[1.1.3 Projet 5](#_Toc115852294)

[1.2 JavaScript & Node.js : consignes techniques, timing et évaluations 6](#_Toc115852295)

[1.3 Ergonomie : consignes techniques, timing et évaluations 11](#_Toc115852296)

[2 Objectif du projet 11](#_Toc115852297)

[3 Mind map du projet 12](#_Toc115852298)

[4 Persona 12](#_Toc115852299)

[5 Axiomes de Morville 13](#_Toc115852300)

[6 Planning des tâches et cas d’utilisation 13](#_Toc115852301)

[7 Besoins techniques 15](#_Toc115852302)

[7.1 Système 15](#_Toc115852303)

[7.2 Frontend 15](#_Toc115852304)

[7.3 API 16](#_Toc115852305)

[8 Choix technologiques 17](#_Toc115852306)

[8.1 Frontend 17](#_Toc115852307)

[8.2 RESTful API 17](#_Toc115852308)

[8.3 Wireframe 17](#_Toc115852309)

[9 Conception & Implémentation 17](#_Toc115852310)

[9.1 Code repositories 17](#_Toc115852311)

[9.2 Secrets éventuels pour vos API ou base de données 18](#_Toc115852312)

[9.3 Documentation de votre API 18](#_Toc115852313)

[9.4 Déploiement de vos applications 19](#_Toc115852314)

[9.5 Code réutilisé 19](#_Toc115852315)

[10 Analyse des résultats par le groupe 20](#_Toc115852316)

[10.1 Evaluation du résultat par rapport au planning des tâches et des cas d’utilisation 20](#_Toc115852317)

[10.2 Audit ergonomique de votre projet 20](#_Toc115852318)

[10.3 Difficultés techniques rencontrées 20](#_Toc115852319)

[10.4 Conseils pour appliquer cette technologie 20](#_Toc115852320)

[10.5 Quels sont les points positifs à la manière dont s’est déroulée la collaboration au sein du groupe ? 21](#_Toc115852321)

[10.6 Quels sont les points qui seraient à améliorer pour de futures collaborations ? 21](#_Toc115852322)

[11 Analyses individuelles des résultats 21](#_Toc115852323)

[12 Présentation vidéo 21](#_Toc115852324)

[13 Revues de projets par les pairs 22](#_Toc115852325)

# Consignes et évaluations

## Consignes générales

### Création des groupes sur le site du cours

Veuillez former un groupe de 4 ou 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 4 ou 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 (23/10), 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 4-5 étudiants remplis, il restera maximum 3 étudiants non liés à un projet. Si nécessaire un ou plusieurs groupes de 3 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/7av06CzK>

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**.

Graphical user interface, text, application

Description automatically generated

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

Après un refresh de la page qui suit, voilà ce qui apparait :

Graphical user interface, text, application

Description automatically generated

Un web repository a 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/7av06CzK>.

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**.

Graphical user interface, text, application, email

Description automatically generated

Si vous préférez, 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** | **Dead-line** | **Pt** | **Consignes** |
| --- | --- | --- | --- | --- | --- |
| Objectif du projet | C7) Documenter et présenter en vidéo le développement d'une SPA |  | **23/10** |  | 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 | C7) |  | **27/11** |  | 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 | C7) |  | **27/11** |  | 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 | C7) |  | **04/12** |  | Compléter le §8.  Discuter de vos choix technologiques avec un enseignant. |
| Rapports individuels d’activités | C7) | Rapports de qualité *Indicateurs : formulation de qualité, analyse de qualité, respect des consignes* | **18/12** | **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.  Tout formulaire hebdomadaire non complété amènera à une pénalité individuelle de 0.5 point. |
| Soumission du rapport de groupe | C7) | Idem | **18/12** | **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 | C7) | Vidéo de qualité  *Indicateurs : présentation du projet de qualité, analyse de qualité, respect des consignes* | **18/12** | **2** | Présenter votre projet selon les exigences du §12. |
| Soumission du code du frontend | C1) Créer une IHM interactive, moderne & esthétique  Optionnel : C4) Intégrer l'authentification, l'autorisation et les sessions d'utilisateurs au sein d'une SPA | Qualité de l’IHM produite  *Indicateurs : esthétique, fonctionnel, codage de qualité, respect des consignes, ambitieux & original* | **18/12** | **5** | Réaliser un frontend et un backend de Qualité : Code bien structuré, UI et UX de qualité  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 différent des APIs fournies dans les démos du cours de JS. |
|  | C3) Créer une SPA intégrant une IHM & un web service | Qualité de l’intégration du service web à l’IHM  *Indicateurs : fonctionnel, codage de qualité, respect des consignes* | **18/12** | **2** |
| Soumission du code du backend | C2) Créer un service web de base  Optionnel : C4) Intégrer l'authentification, l'autorisation et les sessions d'utilisateurs au sein d'une SPA | Qualité du web service produit  *Indicateurs : fonctionnel, codage de qualité, respect des consignes, ambitieux & original* | **18/12** | **4** |
| Utilisation d’une librairie pour des animations ou un jeu  Utilisation d’une librairie pour le service web | C6) Intégrer au développement d'une SPA des technologies non vues en cours | Intégration de librairies non vues en cours  *Indicateurs : utilisation d’une librairie pour l’IHM, utilisation d’une librairie pour le service web* | **18/12** | **2** |  |
| Déploiement tant de votre frontend que backend | C5) Déployer une SPA sur le cloud | Déploiement de la SPA sur le cloud  *Indicateurs : fonctionnel, performances de chargement acceptables* | **18/12** | **2** |  |
| Réaliser un minimum de 5 revues sur le site web | C8) 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, Milestone, commit…) Et Lors des sessions de cours.  Les étudiants n’ayant pas participés activement au projet recevront d’office une lourde pénalité au niveau de leurs points, Voire un 0/20.  Les étudiants n’ayant pas réalisé au moins un use case significative seront considérés inactifs. |

## 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 | **23/10** |  | 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 | **13/11** | 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 | **20/11** | 4 | Construire les wireframes détaillés de votre application. |
|  | Analyse des résultats et rapport associé | **18/12** | 2 | Auditez votre projet et vérifiez le respect des règles GDPR. |
|  | Présentation vidéo | **18/12** | 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

L'idée de base était de faire un jeu permettant à n’importe qui de découvrir ou redécouvrir la Belgique à travers un memory Game, dans lequel les cartes présenteraient différents monuments et personnalités publiques belges. L’idée nous était venue en visionnant une vidéo où la plupart des gens en dehors de l'Europe ne savaient même pas situer la Belgique sur une carte du monde. Toutefois pour susciter l’intérêt des potentiels utilisateurs, il fallait plus que ça. C’est en suivant la coupe du monde que nous avons décidé de mettre des joueurs de l’équipe de foot belge sur les cartes à retourner, afin de susciter plus d’intérêts. En effet, en cette période de CDM le globe entier est concentré sur cette compétition. Cela nous permet aussi de rendre hommage à la génération dorée des footballeurs belges qui pour la plupart, prendront leur retraite cette année.

# Mind map du projet

# Persona

# Axiomes de Morville

# Planning des tâches et cas d’utilisation

* URL vers votre GitHub Project public : <https://github.com/orgs/e-vinci/projects/47>

# 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

La première librairie que nous avons utilisée pour répondre aux besoins de plusieurs UC de notre projet est « anime.js ». En effet le principe d’une « memory game » est de retourner des cartes, pour ce faire nous avons utilisé la fonction « transform » présente dans cette librairie : <https://animejs.com/documentation/#CSStransforms.>

« Anime.js » a été utilisé dans la « home page », la « gamesolopage » ainsi que la « game1v1page ».

La deuxième librairie externe utilisée se nomme « swal js », qui présente une alternative à l’« alert box » par défaut présente sur les navigateurs web. L’utilisation de cette librairie ne répond pas directement à un besoin d’UC mais permet un rendu plus agréable pour l’utilisateur : <https://sweetalert2.github.io/.>

« Swal.js » a été utilisé dans la « home page », la « gamesolopage » ,la « game1v1page » ainsi que la page « contact view ».

## RESTful API

Pour l’api, le package utilisé, non-vu en cours, est « better-sqlite3 ». Étant donné que notre site web ne requiert pas beaucoup de données à stocker (user inscrit leur point et message), on a choisi d’utiliser une DB directement embedded dans le backend du projet, mais il nous fallait quand même pouvoir faire une base de données relationnelle avec des prepared statement. « Better-sqlite3 » nous a permis de faire cela. En créant un fichier sqlite (./models/identifier.sqlite) contenant les CREATE table et les INSERT de base il suffit ensuite de faire des requêtes directement vers ce fichier.

Lien vers la ressource principale utile au développement :

<https://www.npmjs.com/package/better-sqlite3>

## Wireframe

# Conception & Implémentation

## Code repositories

* URL pour le web repository public associé à votre projet :

<https://github.com/e-vinci/web2-2022-project-group-09>

## Secrets éventuels pour vos API ou base de données

*/*

## Documentation de votre API

* Tableaux représentant les opérations de votre API ou lien vers la documentation de votre API :

|  |  |  |  |
| --- | --- | --- | --- |
| Url | Method | Opérations | Test Rest Client |
| auth/login | POST | Permet à un user de se connecter au site web. |  |
| auth/register | POST | Permet à un nouvel utilisateur de créer un compte sur le site web. |  |
| auth/logout | GET | Permet à l’user de se déconnecter. |  |
| /message | GET | Permet de récupérer la liste des précédents messages envoyé, de l’user connecté, au admin du site. |  |
| /message/addMessageUser | POST | Permet à l’user connecté d'envoyer un message (question, suggestion ou delete my account) aux créateurs du site. |  |
| /message/addMessageVisitor | POST | Permet à l’user anonyme d'envoyer un message aux créateurs du site. |  |
| /message/id | PATCH | Permet à l’user connecté de modifier ses anciens messages. |  |
| /message/id | DELETE | Permet à l’user connecté de supprimer un ou plusieurs anciens messages. |  |
| /point | GET | Permet de récupérer les points (gagnés en jouant au jeu) de l’user connecté. |  |
| /point | POST | Permet d’ajouter des points à l’user connecté. |  |
| /point/getRanking | GET | Permet d’avoir les 10 users ayant le plus de points. |  |

## Déploiement de vos applications

* URL de votre frontend déployé :

<https://github.com/adamvinci/adamvinci.github.io>

* URL de votre RESTful API déployée : <https://github.com/adamvinci/RedDevilsMeMBackend>

## Code réutilisé

| **Chemin du fichier où se trouve le code réutilisé** | **Auteur du code source réutilisé** | **URL où le code réutilisé est disponible** | **Raison de la réutilisation du code** |
| --- | --- | --- | --- |
| /frontend/src/Pages/GameSoloPage.js  /frontend/src/stylesheets/main.css | Simo Edwin | <https://github.com/developedbyed/memory-game> | Aide pour les cartes |

# Analyse des résultats par le groupe

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

Les UC principales de notre projet étaient les suivantes :

* Faire la page d’accueil (login/register) (Adam Barhdadi).
* Faire la page d’accueil une fois connecté (Kubra Arslan & Leon Kelmendi).
* Faire la page de jeu solo (vs time) (Kubra Arslan & Leon Kelmendi).
* Faire la page de jeu 1vs1 (Adam Barhdadi).
* Faire la page de contact (Adam Barhdadi).
* Faire la page de statistiques (Ibrahim Bekkari).
* Faire la page de ranking (Ibrahim Bekkari).
* Faire l’api et les requêtes depuis le frontend vers celle-ci (Adam Barhdadi).

Grâce à une bonne organisation de chacun des membres de l’équipe, nous avons pu finir ce que nous avions prévu de faire. Quant aux UC moins urgentes, nous en avons pu en réaliser une grande partie, mais nous n’avons pas pu tout terminer. Par exemple : Animation rideau de carte dans le home page, mettre en place un « remember me ».

Quant aux UC changés en cours de projet, il n’y en a pas eu.

La seule modification par rapport à l’idée originale est le thème principal choisi du site : nous nous sommes concentrés sur le thème de l’équipe nationale de football.

## Audit ergonomique de votre projet

Pour notre site web nous avons choisi une lecture en Z. Tout ce qui est important pour jouer se trouve au centre de l’écran et vous ne devez pas cliquer sur plusieurs liens pour y accéder. Les menus se trouvent quant à eux en tête de page, leur police est suffisamment grande afin d’accrocher rapidement le regard.

Nous avons également appliqué la loi de Fitts afin d’optimaliser la durée de manipulation de l’utilisateur. Notre site possède peu de bouton et chaque bouton est accessible et intuitif. Depuis la page de login, 3 clicks vous séparent de la page de jeu !

Les couleurs sont sobres, la police des caractères est très lisible et les images utilisées sont recherchées. Tous ces éléments ont été choisis pour permettre à nos utilisateurs de jouer dans un environnement agréable.

Nous respectons également le RGPD. Vous avez le choix entre vous inscrire ou rester anonyme. Pour rappel la seule différence c’est qu’en anonyme votre score ne sera pas sauvegardé. Si vous vous inscrivez, vous devrez fournir un login en plus d’un mot de passe. Celui-ci sera bien sûr crypté dans la base de données. Par conséquent vous serez le seul à pouvoir vous connecter à votre compte.

Pour accepter les termes et conditions d’utilisation, une case est à cocher obligatoirement. Sans cela pas d’accès au jeu.

La protection de vos données nous tient à cœur : à tout moment, dans la page de contact, vous pouvez faire une demande pour pouvoir supprimer votre compte en choisissant « Delete my account ».

## Difficultés techniques rencontrées

La première difficulté rencontrée était l’initialisation de la base de données (DB), étant donné qu’aux cours, nous n’avons pas vu comment nous connecter à une DB en ligne et comment y faire des requêtes. Il a été difficile de savoir par où commencer et en voulant nous inspirer des projets de l’année passée, nous nous sommes emmêlés les pinceaux, vu que chacun faisait quelque chose de plus au moins diffèrent.

La deuxième difficulté était liée au responsive du site, vu que chaque membre du groupe travaillait sur un écran avec des résolutions différentes, lorsqu’une page s’affichait correctement sur un écran, elle pouvait être complètement désordonnée sur un autre.

La troisième difficulté rencontrée concernait le déploiement du backend en ligne, better-sqlite3 n’était pas compatible avec les azure app. Selon azure cette librairie n’était ‘pas une win32 application’, ce qui nous a contraints à retaper toute la gestion des données dans le backend (nous n’avons pas réussi à trouver de solution même en demandant directement dans le repo github de la librairie). Ce qui nous a forcés à abandonner la database embedded pour ce projet. Toujours pour le déploiement, nous nous sommes rendu compte que lorsque nous nous connections/inscrivions depuis la github page il n’y avait pas de transfert de cookie entre l’api et l’hmi pourtant ça marchait en local et avec Rest client aussi. Cela nous a donc forcés à mettre le token dans du JSON à chaque appel d’api.

## Conseils pour appliquer cette technologie

Utiliser les Single Page Application : même si les SAP permettent d’améliorer et de fluidifier l’expérience utilisateur en évitant le chargement de page à chaque action, avoir une seule page web comme accès peut entraîner certains problèmes, c’est pour cela que nous conseillons de :

* Utiliser « eslint » qui évitera des erreurs indétectables, car cachées dans le code.
* Faire le déploiement du site au début, ce qui vous évitera de tout devoir changer à la dernière minute à cause d’une erreur rencontrée pendant le déploiement
* Un des désavantages des SPA et que parfois, la page peut mettre du temps avant de charger et en sachant que la plupart des animations CSS3 peuvent être faites avec « anime.js » nous conseillons d’utiliser des librairies externes quand c’est possible.

Utiliser « better-sqlite3.js » : pour des petits sites web avec peu de données, SQLite est idéal, mais avoir le fichier de la base de données (DB) stocké directement dans le backend est contraignant pour la lecture de données donc nous conseillons soit d’ouvrir le fichier « identifier.sqlite » sur des sites web fait spécialement pour ça ou bien d’avoir la DB dans datagrip et changer le chemin des requêtes vers là où est stocké le projet datagrip.

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

Sur ces 4 semaines de projet, la collaboration au sein du groupe a permis :

* De mettre en place une discipline de travail, en voyant les autres avancer.
* De découvrir des nouvelles technologies, connues par d’autres membres.
* D’avancer plus vite, en s’entraidant lors de problèmes techniques.
* De se familiariser avec des outils permettant la collaboration en groupe.
* De s’améliorer individuellement, grâce au critique anonyme dans TEAMATES.
* De mieux comprendre des concepts : en expliquant aux collaborateurs ce qu’on fait, on a une approche extérieure et donc une meilleure analyse.

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

Pour les wireframes (maquettes) du site, 2 membres de notre groupe n’avaient pas le cours d’ergo dans leur PAE, ce qui fait que lorsque les wireframes ont été remis au prof d’ergonomie, ça a été fait sans demander l’avis des 2 autres personnes du groupe, ce qui a entrainé des changements de design en milieu de projet qui était difficile à suivre. Donc la prochaine fois il serait judicieux de choisir un design dès le départ et de s’y tenir jusqu’au bout.

Comme mentionné dans un des reports TEAMATES, pour de futures collaborations il serait judicieux d’utiliser git plus intelligemment, avec des branches et pull request, car aux heures où nous travaillions ensemble la branche main était souvent embouteillée ce qui entrainait des conflits assez conséquents qu’il a fallu gérer.

# Présentation vidéo

Lien vers la vidéo youtube : <https://youtu.be/b2DIRnuwabc>