Ergonomie & développement d’une SPA animée

**INCLURE ICI LE NOM DE VOTRE PROJET**

TimeToQuiz

|  |  |
| --- | --- |
| **Auteur 1** | Younes Benbouchta |
| **Auteur 2** | Quoc An Duong |
| **Auteur 3** | Trong Nguyen |
| **Auteur 4** | Maxime Vlaminck |
| **Auteur 5** | Ke man Yi Nghi |
| **Date** | 18.12.2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-17 |
| **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°13 : TimeToQuiz * 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, Milestones, commits…) 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 significatif 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

Notre objectif en développant notre site a été, avant tout, de créer une manière ludique d’approfondir ses connaissances en matière de culture générale. En effet, les adolescents de nos jours tendent de plus en plus à s’abrutir devant des montagnes de contenu sans fond, uniquement pour faire passer le temps et ainsi éviter l’ennui. Résultat : il y a de plus en plus d’adolescents accros à leurs smartphones et aux applications de partage de contenu telles que TikTok, Instagram, et bien d’autres... Notre projet a donc pour but d’aller chercher ces adolescents sur cette plateforme à laquelle ils sont accros pour leur proposer une manière de se détourner de ces applications nocives pour leur développement intellectuel et, ainsi, de mettre à leur disposition un site internet où l’apprentissage se fait passivement via un concept classique mais efficace : le quiz. De cette manière, au fur et à mesure des quiz complétés, l’utilisateur accumule de plus en plus de connaissances, dans un but final de répandre des connaissances au plus grand monde, que ce soit des enfants, des adolescents, des adultes ou des personnes âgées et d’en faire une norme.

# 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/41

# 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

Pour le frontend, nous avons utilisé anime.js et sweetAlert2 pour notre application. anime.js nous a permis de réaliser des animations dans notre site, le rendant plus dynamique et esthétique. Toutes nos animations, à part quelques exceptions, proviennent de la documentation de la librairie : <https://animejs.com/documentation/>.

Pour la RegisterPage, on a utilisé sweetAlert2 pour afficher de manière esthétique notre politique de confidentialité, afin de respecter les règles de GDPR (create GDPR).

Pour la QuizPage, nous avons ajouté une animation venant du site <https://tobiasahlin.com/moving-letters/#6> lors de l’affichage du titre du quiz et du message du score. Nous avons également ajouté deux animations donnant un effet fondu en noir pour enchaîner vers les questions, une animation de transition quand on passe d’une question à l’autre et une animation pour l’affichage du message après un choix (UC2 Quiz). Quelques animations de transition sont également mises lors de l’affichage du score (UC3 Quiz).

Nous avons utilisé l’animation du site <https://codepen.io/borntofrappe/pen/LwZRON> comme minuteur avec bombe (UC4 Quiz) afin de limiter le joueur à un temps imparti lors d’un choix de réponse. Cette animation a été réutilisée dans la HomePage pour soutenir la présentation du site.

Pour l’API, nous avons utilisé translate-google comme librairie. Cette librairie gratuite nous a servi à traduire d’un langage quelconque à l’anglais grâce à la détection de langue automatique. Elle a été utilisée pour la recherche d’un quiz afin de faciliter la recherche pour les personnes non anglophones (UC1 Home).

## Frontend

***Frontend***

*Librairies utilisées non vue au cours :*

*- anime.js https://animejs.com/*

*- sweetAlert2 https://sweetalert2.github.io/*

## RESTful API

***API***

*Librairies utilisées non vue au cours :*

*- translate-google https://www.npmjs.com/package/translate-google*

## Wireframe

# Conception & Implémentation

## Code repositories

* URL pour le web repository public associé à votre projet : https://github.com/e-vinci/web2-2022-project-group-13

## 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 :

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Méthode http | opération | Auths |
| /quiz | GET | READ ALL : lire toutes les ressources de type « quiz » vérifiés | Non |
| /quiz/id/ :id | GET | READ ONE : lire la ressource « quiz » vérifié | Non |
| /quiz/addQuiz | POST | CREATE ONE : créer la ressource « quiz » | JWT |
| /quiz/search | GET | SEARCH : recherche la ressource « quiz » vérifié | Non |
| /auths/register | POST | Créer une ressource de type "users" et renvoyer le username et un token JWT | Non |
| /auths/login | POST | Vérifier les credentials d'une ressource de type "users" et renvoyer le username et un token JWT si les credentials sont OK | Non |
| /admin | GET | READ ALL : lire toutes les ressources de type « quiz » non vérifiés | JWT |
| /admin/all | GET | READ ALL : lire toutes les ressources de type « quiz » | JWT |
| /admin/remove/ :id | DELETE | DELETE ONE : Effacer la ressource identifiée | JWT |
| /admin/quiz/ :id | GET | READ ONE : lire la ressource « quiz » | JWT |
| /admin/validate / :id | POST | Valide la ressource « quiz » et met sa donnée « isVerified » à true | JWT |

* Requêtes HTTP se trouvent dans : **/api/REST Client**

## Déploiement de vos applications

* URL de votre frontend déployé : https://kekmanlol.github.io/group-13-frontend-deployment-vinci/
* URL de votre RESTful API déployée : <https://timetoquiz.azurewebsites.net/>

## 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/Components/QuizPage.js | Gabriel corti | https://codepen.io/borntofrappe/pen/LwZRON | Code pour avoir un timer + animation sur la quizPage |
| /frontend/src/Components/HomePage.js | Gabriel corti | https://codepen.io/borntofrappe/pen/LwZRON | Code pour avoir une animation sur la HomePage |
| /frontend/src/Components/QuizPage.js | In « w3schools » website | https://www.w3schools.com/js/js\_random.asp | Code pour randomiser les questions et les réponses d’un quiz |
| /frontend/src/Components/QuizPage.js | Tobiasahlin website | https://tobiasahlin.com/ | Code d’animation pour le titre du quiz |

# Analyse des résultats par le groupe

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

Une fonctionnalité que nous n’avons finalement pas ajouté sur le site, est le fait de pouvoir choisir la difficulté d’un seul quiz, et de créer un quiz avec différents niveaux de difficulté (la durée d’une question se réduisait si le niveau de difficulté augmentait et les questions devenait de plus en plus dur en fonction de la difficulté). Nous estimions que pour un utilisateur, la création d’un quiz ou l’expérience du quiz serait beaucoup trop longue (30 questions au total, 10 questions par difficulté), ce qui pourrait facilement l’ennuyer et finalement laisser la création ou abandonner le quiz. Nous avions décidé de faire finalement en sorte qu’un utilisateur puisse avoir une courte, et bonne expérience de notre application. Ainsi, rendu la création d’un quiz et la durée d’un quiz moins lourde. L’utilisateur ou un administrateur pourra ainsi créer, un quiz de 10 questions uniquement en choisissant son niveau de difficulté. Le niveau difficulté changera uniquement par rapport à la difficulté des questions et non du Timer. Nous avons limité le Timer à 10 secondes pour chaque question d’un quiz. Mise à part cela, nous avions pu réaliser nos différents objectifs fonctionnels et eu le résultat qu’on voulait avoir.

## Audit ergonomique de votre projet

Les boutons principaux sont suffisamment gros pour une accessibilité facile au partie importante (quiz) et sont concentré vers le milieu de la page par rapport au menu qui sert à quitter les fonctionnalités principales du jeu.

L’interface du jeu est relativement épurée pour ne pas déconcentrer l’utilisateur et le fond est sur une couleur relativement neutre pour éviter de biaiser sa perception (perception de la couleur notamment qui peut induire sur ses choix). Les contrastes sont suffisants pour être clair.

L’utilisateur doit cependant déplacer significativement sa souris lors du choix des réponses et faire des actions inutiles (choix de réponse non modifiable mais obligation de cliquer sur suivant) mais cela est un choix actif de notre part. La raison est que notre concept d’application est orienté didactique. Ces temps de pause non nécessaire permettent à l’utilisateur d’avoir le contrôle et d’assimiler les bonnes réponses plutôt que faire un maximum de quiz en enchainant plus de questions rapidement.

Le bouton d’acceptation du RGPD n’est pas pré-coché et l’utilisateur à la possibilité de retirer son consentement.

## Difficultés techniques rencontrées

Nous avions rencontré plusieurs difficultés. Pour la HomePage, la fonctionnalité pour rechercher un quiz à eu beaucoup de problème, que ce soit en front-end ou back-end, tout d’abord pour le front-end, Quoc An Duong a eu du mal à créer un submit pour cette fonctionnalité, qui a finalement été avec un Event de type « keypress » et un « click ». la fonctionnalité a cessé de fonctionné lors de l’ajout du code de Ke Man, il a cherché pendant des heures à trouver le problème, le message d’erreur était un SyntaxError, hors, le problème se trouvait ailleurs. Il se trouvait du côté backend, sur les routes, une requête empêchait la fonctionnalité de fonctionner sans avoir le moindre détail. C’est en modifiant une certaine route, que la fonction recherche refonctionnait. Ke man et Quoc An ont eu aussi du mal à gérer l’animation du timer et aussi le timer pour la page quiz. En effet, lors de l’ajout de l’ajout de celui-ci, les boutons pour submit une réponse à une question ne fonctionnait plus, le problème a été résolu en ajoutant une nouvelle div pour l’animation de la bombe. Le timer a été fait étape par étape, ils ont d’abord créé une fonction afin de créer de compte à rebours de 10 secondes, puis ils l’ont implémenté sur la pageQuiz pour chaque question, ils ont tout d’abord eu comme idée de passer une page après les 10 secondes, mais finalement à la fin du compte à rebours, la réponse de la question est affiché après ce délais, et un boutton « next » est présent afin de passer à la question suivante. En cas de réponse trouver ou de mauvaise réponse, le timer s’arrête et la page indique si l’utilisateur a trouvé la bonne réponse ou non. Il peut ainsi aisément passer à la page suivante.

## Conseils pour appliquer cette technologie

Nous pensions que les technologies que nous avions utilisés lors de notre projet n’a pas été réellement compliqués à assimiler. Effectuer des recherches sur internet avant de se lancer sur ce projet (et en général) peut être aussi une source pour ne pas avoir de doutes sur ce que vous allez faire et avant de poser une question à quelqu’un, effectuez des recherches par vous-même, internet est grand, puissant, et donc il y a casi toutes les réponses à vos questions mais aussi de comprendre l’anglais car la plupart des réponses sont en anglais. En tant cas que futur développeur, ou informaticien en général, nous vous conseillons le forum « Stackoverflow ». Casi 90% des réponses de questions que vous vous posez ou de problèmes que vous rencontrez se retrouve sur ce site. Nous vous conseillons aussi de rejoindre les serveurs Discord officiel de certaine technologie ou des serveurs discord de programmation en général. Ce sont en générales des serveurs où des personnes s’entraident, et où l’on peut poser tout type de questions. Il y a en général beaucoup d’informaticiens ayant déjà de l’expérience dans le milieu professionnel qui peuvent vous aider. Nous pensons ainsi, que l’une des qualités d’un informaticien est le fait d’être autonome. Le fait de savoir, apprendre une nouvelle technologie, d’arriver à s’informer seul, et effectuer ses recherches en anglais, est primordial. Cela évitera de déranger une personne pour rien.

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

Lors de notre projet, étant donné, que nous nous connaissions tous, l’équipe s’est montré très coopérative, Nous avions réparti les tâches, tout le monde a pu finalement finir ce qu’il devait réaliser. Nous nous sommes toujours fait confiance. Nous avions utilisé l’application Discord pour créer un groupe et ainsi pour pouvoir nous communiquer. Si une personne avait un problème sur son code, il pouvait toujours demander à une personne sur le groupe afin de pouvoir régler son souci. Nous n’avions aussi pas eu de problème technique. Chacun n’avait pas réellement de difficulté par rapport à la matière javascript, et donc, nous avions pu faire ce projet aisément et nous avions pu facilement résoudre les problèmes de conflits sur notre code. Le point clé de notre projet était surtout la communication et la répartition de nos tâches, cela a été un point primordial pour finir le code de notre projet à temps.

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

La gestion du temps a été un point négatif de notre projet. Nous pensions s’être pris trop tard pour le projet. Nous avions dû changer de projet quasiment à la dernière minute car nous avions eu beaucoup de mal à trouver une idée de projet qui conviendrait pour tout le monde, et cela s’est répercuté sur les choix technologiques à utiliser, le choix du design de notre application, nous pensions que nous aurions pu largement améliorer la qualité de nos wireframes et avoir une meilleure interface pour l’utilisateur, et sur la répartition des tâches, nous aurions pu avoir une meilleure répartition, car finalement, certaines tâches attribuées pour une seule personne par exemple, se sont finalement réalisées à 2. Nous tâcherons de décider une idée de projet plus tôt afin de ne plus être dans la même situation qu’on a été pour ce trimestre.