Ergonomie & développement d’une SPA animée

**Double Or Nothing**



|  |  |
| --- | --- |
| **Auteur 1** | Brandon VAN BELLINGHEN |
| **Auteur 2** | Emilien DIERYCK |
| **Auteur 3** | Loic THOMAS |
| **Auteur 4** | Tanguy VANDERVELDEN |
| **Auteur 5** |  |
| **Date** | 18.12.2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-06 |
| **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 amenera 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 différente 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

Dans cette partie, nous allons parler des objectifs de notre projet.

Notre objectif principal de notre projet est très simple. L’objectif est de créer un site de casino fonctionnelle et gratuit à tous. Mais pour cela, nous devons réaliser plusieurs petits objectifs pour que le site soit fonctionnel.

Maintenant je vais parler des petits objectifs à réaliser pour réussi l’objectif principal :

* Création de la base de données.
* Création des méthodes pour créer un compte et se connecter
* Création des frontend de tous les jeux
* Création des méthodes pour que les jeux fonctionnent.
* Régler les problemes

Les personnes prisent en charge par notre site sont les personnes fans des jeux de casinos qui auront accès à plusieurs jeux sans payer quelques choses ou télécharger plusieurs applications. Et aussi nous visons les gens qui veulent découvrir les jeux d’argent sans dépenser de l’argent directement. Bien sur pour respecter les lois. Tous utilisateurs doivent avoir plus de 21 ans sur notre site.

Comme nous sommes des fans de jeux de casino comme le Blackjack, le poker et autre. Nous sommes tous motivé à faire ce site et donner toutes notre énergie pour qu’il fonctionne à 100%.

# Mind map du projet

*Veuillez créer le Mind map de votre application en allant le plus en détails. N’oubliez pas de regrouper en catégories pertinentes comme vu en cours.*

*Veuillez soumettre votre mind map au sein du répertoire* ***/ergonomics/marketing-view*** *de votre repo.*

# Persona

*Veuillez créer au moins 2 personas de votre audience cible. Le but étant de représenter au mieux les personnes qui seront le plus engagés par votre application.*

*Veuillez soumettre vos persona au sein du répertoire* ***/ergonomics/marketing-view*** *de votre repo.*

# Axiomes de Morville

*Veuillez décrire chaque axiome de Morville de manière assez complète vis-à-vis de votre application. Ne pas donner de réponse basique mais allez plus loin dans les besoins.*

*Exemple : App de la ville de Bruxelles*

* *Valable : “parce que les gens en ont besoin et ça les aidera”*

*--> Une application aide toujours mais en quoi cela rajoute de la plus-value ?*

*--> En quoi l’application va*

*réellement changer l’expérience utilisateur déjà existante pour ce type de projet ? Et si cela ne change pas, il faudra aussi le justifier. Ne pas recréer la roue n’est pas un défaut en soi, ça s’appelle de l’efficience.*

*Veuillez soumettre vos axiomes au sein du répertoire* ***/ergonomics/marketing-view*** *de votre repo.*

# Planning des tâches et cas d’utilisation

*Au sein du repository GitHub de votre projet, vous devez créer un* ***Project*** *pour planifier les tâches, allouer les responsables, documenter vos avancées, visualiser vos* ***Milestones****...*

*Votre projet doit être* ***public*** *et doit suivre le template :* ***New project****,* ***Team backlog****.*

*Vous pouvez supprimer la colonne* ***New****.*

*Voici une vidéo expliquant la mise en place de votre GithHub Project :* [*Partie1 : Créer et configurer un GitHub Project*](https://youtu.be/y2e_ItjAZ2I)*.*

*Voici le workflow que nous souhaitons vous voir appliquer sur GitHub Project :*

* *Veuillez commencer votre projet en identifiant toutes les tâches principales à réaliser sur votre projet, principalement en soignant l'identification des use cases. Pour ce faire, vous pouvez visualiser la vidéo :* [*Partie 2 : planifier des tâches via GitHub Project*](https://youtu.be/hz0P-PfYqgo)*.*
* *Chaque****tâche****doit être couverte par une****draft Issue****au sein de GithHub que vous devrez convertir plus tard en****Issue****. Veuillez allouer une****Priority****à chaque****Issue****(ou****draft Issue****), ainsi qu'une****Size****.*
* *Lors de l'identification des tâches, les****Issues****associées se trouvent dans la colonne****Backlog****.*
* *Un****cas d'utilisation****doit être couvert par au moins une****Issue****avec un label nommé****enhancement****.*
* *Chaque****Issue****doit être associée à au moins un****Assignee****.*
* *S'il y a plusieurs****Assignees****associés à une****Issue****, celle-ci devra être découpée en suffisamment de tâches pour qu'il y ait un seul****Assignee****par tâche. Pour la découpe d’une Issue en tâches, la création de Label, et la gestion de Milestones, vous pouvez visualiser la vidéo :* [*Partie 3 : gestion approfondie des tâches via GitHub Project*](https://youtu.be/F8lPoL8qiZA)*.*
* *Dans un premier temps, une****Issue****associée à plusieurs****Assignees****peut simplement identifier les tâches associées au sein d'une****task list****. Plus tard, ces tâches devront être converties en nouvelles****Issues****associées à un seul****Assignee****.*
* *Lorsque vous travailler sur une****Issue****:*
  + *elle doit se trouver dans la colonne****In progress****ou****In review****si vous pensez avoir terminé mais que vous attendez le feedback d'un membre de votre projet.*
  + *pour chaque avancée significative sur une****Issue****, vous devez indiquer un commentaire via****Comment****résumant le travail effectué.*
* *Lorsque vous considérez qu'une****Issue****est terminée, faites la passer dans la colonne****Done****, indiquez un message via****Comment****résumant le travail effectué et cliquez sur****Close issue****.*
* *Pour facilement voir le pourcentage de progrès dans la fermeture des****Issues****qui vous sont associées, vous devez créer une****Milestone****par membre de projet et associer cette****Milestone****à toutes les****Issues****où le membre de projet est le seul****Assignee****.*

*Pour visualiser les tâches d’un seul membre de projet, il est possible de faire, via GitHub Project, autant de* ***view*** *qu’il y a de membres de projet, ou une seule vue pour chaque utilisateur connecté. Pour obtenir une vue pour un membre de projet, il suffit de cliquer sur la* ***view****, puis* ***filter****, puis d’écrire :* ***assignee:*** *et d’indiquer le username d’un membre du projet. Il est aussi possible de créer une vue des tâches de l’utilisateur connecté, qui pourra être réutilisée par tous les membres du groupe, en indiquant :* ***assignee:@me*** *pour le* ***filter****.*

*Pour votre développement, n’hésitez pas à être ambitieux, tout en restant réaliste. Comment faire ? Nous vous recommandons de spécifier les cas d’utilisations qui* ***doivent*** *être implémentés par une priorité «****Haute****». Ceux qui* ***pourraient*** *être implémentés, mais qui ne sont donc pas indispensables à l’application de base, vous devriez les catégoriser selon une priorité « Moyenne » ou « Basse ».*

*Afin de vous aider dans la création de votre planning au sein de GitHub Project, nous vous proposons une liste de base d’****Issues*** *à prendre en compte**:*

* *Create marketing vision (documentation)*
* *Create wireframes (documentation)*
* *Identify list of use cases (documentation)*
* *UC1 : ... (enhancement)*
  + *UC1-frontend : ... (enhancement)*
  + *UC2-api : ... (enhancement)*
    - *Create API HTTP requests (tests)*
* *UC2 : ... (enhancement)*
* *UC3 : ... (enhancement)*
* *UC...*
* *Document API (documentation)*
* *Deploy frontend (deployment)*
* *Deploy API (deployment)*
* *Create Project Report (documentation)*
* *Create Video (documentation)*

*Notons que les rapports individuels d’activités que vous devrez compléter chaque semaine, suite à invitation par e-mail, ne sont pas à reprendre dans votre planning de tâches.*

*Attention que chaque membre de projet doit avoir au moins un use case significatif pour être considéré actif.*

*Veuillez indiquer l'URL vers votre* ***Project*** *public sous GitHub ici :*

* URL vers votre GitHub Project public : <https://github.com/orgs/e-vinci/projects/24>.

# 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

Dans cette partie, nous allons parler des choix technologiques :

Dans notre code, nous avons utiliser que seule librairies JS que nous n’avons pas vu pendant les heures de cours.

La librairie que nous avons utilisée est Jquery. Jquery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web. Nous avons pris Jquery pour faire notre menu carrousel qui se trouve sur notre Homepage (le cas d’utilisateur lier à cette librairie est Frontend Home Page #12). Mais pour utiliser cette librairie, vous devez d’abord l’installer dans votre projet avec la commande ‘’*npm i jquery’’ quand jquery est installé.*

*Après d’avoir installé Jquery dans votre projet, vous devez alors créer un fichier qui s’appelle jquery.js pour mettre intérieur les codes de base* de jquery et après ça vous pouvez l’appeler dans n’importe quels fichiers de votre projet mais il ne faut pas oublier d’import dans vos fichiers avec la commande import ‘’jquery from Jquery’’ car sinon le code jquery ne sera jamais reconnu.

Mais si vous avez des problèmes, vous pouvez aller voir sur le site de Jquery (<https://jquery.com/>.).

Mais il ne faut pas oublier que Jquery est une librairie JS qui est utilisé pour les animations.

## Frontend

*Pour le frontend, voici des exemples de librairies open source qui pourraient être choisies, en fonction de votre objectif :*

* *pour la 2D :* [*https://animejs.com*](https://animejs.com)
* *pour la 3D :* [*https://threejs.org*](https://threejs.org)
* *pour les jeux :* [*https://phaser.io*](https://phaser.io)

## RESTful API

Pour notre api, nous avons utilisé une base de donnée pour nous faciliter le travail pour gérer les jetons dans notre application.

## Wireframe

*Veuillez créer sur PowerPoint, des wireframes de votre application. Il en faut au minimum 3.*

*Exemple d’un jeu : il y aura au minimum, l’accueil inscription, le jeu, une page de résultat, page contact développeur, etc.*

*Veuillez soumettre vos wireframes au sein du répertoire* ***/ergonomics/wireframes*** *de votre repo.*

# Conception & Implémentation

## Code repositories

*Veuillez indiquer l’URL de votre web repository public, générée par GitHub Classroom, sur* *https://e-vinci.github.io/web2/project-page. Pour ce faire, veuillez au moins modifier le champs « Repo frontend » (et éventuellement « Repo backend »). L’URL du web repository de votre groupe devrait correspondre à qqch du style « https://github.com/e-vinci/web2-2022-project-group-04 » si vous êtes membres du Projet N°4.*

*Veuillez aussi indiquer cette URL ci-dessous.*

* URL pour le web repository public associé à votre projet : <https://github.com/e-vinci/web2-2022-project-group-06>.

## Secrets éventuels pour vos API ou base de données

*Si vous utilisez une base de données ou des API nécessitant des secrets, il est important de ne pas rendre public vos secrets. Dans ce cas :*

*- Votre application doit être sur le cloud pour que les autres étudiants puissent la revoir ; les étudiants ne pourront donc pas exécuter l'API localement. Veuillez clairement indiquer dans le README de votre projet si l’application ne peut pas être exécutée localement sans les secrets et veuillez indiquer l'URL tant de votre frontend que de votre API au sein de ce README.*

*- Vous devez mettre à disposition tous ces secrets (fichiers de configuration) à disposition de vos enseignants lors de la soumission de ce rapport. Le devoir Moodle vous permettra d’inclure les fichiers nécessaires.*

*- Pour la création de votre éventuelle DB, si elle ne se fait pas automatiquement lors du démarrage de votre API, vous devez offrir un script et le mettre au sein de votre projet. Dans ce cas, la procédure pour créer la DB doit être documenté au sein du README de votre projet.*

## Documentation de votre API

|  |  |  |
| --- | --- | --- |
| login | Connecter sur notre site | users.js |
| register | Créer un compte sur notre site | users.js |
| getchips | Récupérer les jetons | users.js |
| leaderboad | Récupère des valeurs pour faire le leaderboard | users.js |
|  |  |  |

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

*Veuillez aussi documenter les tests de votre API : les requêtes http doivent être données au sein de votre projet. REST Client devrait être utilisé, mais si vous préférez un autre client léger, vous devez vous mettre d’accord au sein de votre équipe de projet. Veuillez indiquer où se trouvent les requêtes HTTP si ça n’est pas dans le répertoire* ***/api/REST Client****.*

*Rappel : il ne peut pas y avoir d'opération de votre API sans au minimum une requête HTTP associée.*

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

## Déploiement de vos applications

*Veuillez indiquer l’URL de votre frontend déployés sur* *https://e-vinci.github.io/web2/project-page. Pour ce faire, veuillez modifier le champs « URL du site ».*

*Veuillez aussi indiquer ci-dessous deux URLs, comme par exemple https://e-vinci.github.io/wowapp :*

* URL de votre frontend déployé : <https://e-vinci.github.io/DoubleOrNothing-Frontend/>
* URL de votre RESTful API déployée : <https://doubleornothing.azurewebsites.net/>

## Code réutilisé

*Il est important que vous citiez les parties de code que vous avez réutilisées, du code issu d’un tiers, au sein de votre code source. Pour ce faire, dans votre code source, utilisez des commentaires, en complétant au minimum les informations associées à l’auteur et à l’endroit où le code est disponible (URL). Voici le format que pourrait prendre votre commentaire:*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*    Title: <title of program/source code>  
\*    Author: <author(s) names>  
\*    Date: <date>  
\*    Code version: <code version>  
\*    Availability: <where it's located, URL>  
\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*Veuillez résumer tous les codes sources utilisés dans votre code, au sein de ce tableau :*

| **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** |
| --- | --- | --- | --- |
| *e.g. /webApp/src/index.js* | *Dogan Erisen* | *https://github.com/Azure-Samples/active-directory-b2c-javascript-msal-singlepageapp* | *Code pour recevoir un access token via Azure AD* |
|  |  |  |  |
|  |  |  |  |

# Analyse des résultats par le groupe

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

Nous avons pu atteindre 85% du projet. Les plus grosses parties du projet a été réalise sans problèmes mais certains nous encore des problèmes.

Comme :

* Le leaderboard qui est non fonctionnelle sinon créer des erreurs
* Des problèmes pour lier notre base de données à notre code javascript
* Des petits problèmes lors du déploiement

(Mais résolu pour nous)

Sinon nous avons bien réussi tous nos jeux sans problèmes nous avons la plus grosse totalité de notre projet sinon nous aurions plus de temps. Le projet aura pu être fini à 100% avec 1 semaine ou même 2 le projet aurait été fini.

Bien sûr, nous n’avons pas dévie de nos objectifs tous les objectifs que nous avons mis à été fait

## Audit ergonomique de votre projet

Notre projet respect parfaitement toutes les règles vues en ergonomie web.

Notre site respect bien les règles GDPR et passe en-dessous sous aucune ligne des règles.

Nous avons bien fait attention à tous pour respecter à 100% les règles du GDPR ou autres.

Plusieurs personne de notre a bien sur tous regarde avec tous ce qu’on a vu en ergonomie web.

## 

## Difficultés techniques rencontrées

Nous avons eu beaucoup de problèmes quand on a déployé le code du frontend et backend donc on a galéré pendant plusieurs jours et heures.

Un autre problème est que on a eu beaucoup de galère au début avec Jquery car on avait oublié de mettre des dossiers importants pour l’utiliser donc on a perdu beaucoup de temps là-dessus.

Un autre est que on avait beaucoup de conflit dans le nom de fonction car certains noms étaient les mêmes.

On a aussi beaucoup galéré dans les imports car on ne réussissait pas toujours avec nos imports.

On a aussi beaucoup de problèmes quand on devait lier notre base donnée à notre site car il ne trouvait pas quelquefois nos appels qu’on fessait.

Nous avons parfois créé des problèmes sur notre site quand on fessait un push sur github.

On a eu des pro

## Conseils pour appliquer cette technologie :

Comme savoir qu’on devait répondre à teammates tous les Week end car certains membres de notre groupe push souvent le dimanche donc ce prenait des notes négatives par les autres car pour eux. Ces personnes ne la travaillaient pas alors qu’il pushait que le dimanche soir.

Maintenant je vais donner quelques conseils aux gens pour qu’il évite nos problèmes qu’on rencontrait.

Le premier, nous conseillons de choisir quelqu’un comme chef pour donner les choses à faire au gens pour que le projet avance vite, sans perdre de temps.

Le second, nous conseillons de ne pas perdre sur des choses bêtes comme le désigne, le logo et autres.

Maintenant nous allons donner liens utiles pour les gens qui veulent commencer des projets.

Le premier est codepen qui permet de trouver des liens de codes publics pour les gens qui sont bloquer dans certains codes. ( <https://codepen.io/trending>.)

Le second est free frontend qui peux utile pour trouve des codes frontend public pour tous les gens. Des codes en css, html, Bootstrap, javascript et etc.( <https://freefrontend.com/>.).

Et le dernier est pixabay qui peut être utile pour trouver images gratuites pour son site. (<https://pixabay.com/fr/>.)

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

Nous avons eu de la chance d’avoir une personne qui a entre guillemets mener l’équipe en donnant les choses à faire au gens qui nous as beaucoup aidé.

Nous n’avons pas perdu du temps le choix de chose stupide comme dans le logo, le désigne et autre, nous avons vite choisi ce qui a aidé à avancer vite dans le projet.

Une bonne entente. Nous sommes à l’écoute de chaque membre du groupe et nous étions toujours là pour aider les autres.

Nous sommes une bonne équipe de travailleur qui ne perdais pas son temps dans des choses inutiles.

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

Nous n’avons pas des millions problèmes dans notre groupe. Mais nous avons les problèmes de base dans un groupe.  
un des problèmes, c’est la communication, on parlait très peu hors des cours donc le projet a eu du mal à avancer.

Un autre problème est que nous avons visé trop grand comme projet et on a beaucoup galérer pour certaines choses.

Un des autres problèmes est que nous avons perdu un binôme qui avait des taches à faire et qui n’a rien fait donc qui nous a peu aider dans ce projet.

Un autre problème la pression du temps car le temps du projet a vite avance et les autres projets d’autre cours ne nous a pas aidé

# Présentation vidéo

*Voici les exigences associées à votre présentation vidéo :*

* *Elle doit viser une durée de 5 minutes, et ne peut pas dépasser 10 minutes.*
* *Elle doit être visible sous youtube par n’importe qui possédant son URL. Sa visibilité doit donc être en ‘’Unlisted" ou "Public", mais pas « Private » !*
* *Elle se basera principalement :*
  + *sur la présentation de votre application web : exécution, en live, de votre API et du frontend ;*
  + *la présentation de l’expérience utilisateur ;*
* *Elle pourra aussi se baser sur d’autre(s) point(s) éventuel(s) vous permettant de vendre au mieux votre travail.*
* *Votre présentation devra être bien visible et audible.*
* *Il serait bien que celle-ci soit bien structurée, notamment via l’affichage éventuel de titres.*
* *Vous pouvez ajouter une bande son, et des images, mais seulement si celles-ci respectent les droits d’auteurs.*
* *Vous veillerez à ce que, dans la description de votre vidéo, vous fournissiez les liens vers le web repository associé à votre projet ainsi que l’URL vers le frontend déployé.  
  L’idée est que si un projet intéresse des visiteurs de votre repo, ils aient accès à tout ce qui est nécessaire pour bien le comprendre, voire pour le réutiliser, sous réserve de bien citer vos ressources.*

*En plus de ces exigences, la présentation vidéo a pour but de vendre un projet qui vous tient à cœur. Il est possible que si le résultat soit accrocheur, les enseignants demandent votre autorisation afin de rendre votre projet public. Avec votre autorisation, nous pourrions notamment présenter votre projet lors de salons d’étudiants, soit via votre vidéo, ou directement en exécutant votre application déployée sur le cloud.*

*Le site* [*https://e-vinci.github.io/web2/*](https://e-vinci.github.io/web2/) *présentera les projets qui auront été sélectionnés pour être publics. De plus, vous pourriez utiliser vos projets comme portfolio pour vos futurs employeurs.*

*Pour créer votre vidéo, avant de la mettre sous youtube, veillez à ce que celle-ci soit bien visible et bien audible. Nous vous recommandons :*

* *de la réaliser au format 1920 X 1080*
* *d’utiliser un logiciel gratuit pour la réaliser. Voici ceux que nous pouvons vous conseiller :*
  + [*https://obsproject.com/*](https://obsproject.com/)*: logiciel open source demandant un temps d’adaptation, mais permettant de faire énormément*
  + [*https://www.loom.com/*](https://www.loom.com/)*: logiciel pouvant être utilisé gratuitement sous réserve d’accepter un logo. Très facile d’utilisation.*
  + [*https://screencast-o-matic.com/*](https://screencast-o-matic.com/)*: logiciel pouvant être utilisé gratuitement sous réserve d’accepter un logo. Très facile d’utilisation.*

*Veuillez indiquer le lien vers la vidéo youtube que vous avez créée sur le site* [*https://e-vinci.github.io/web2/project-page*](https://e-vinci.github.io/web2/project-page)*. Pour ce faire, veuillez modifier le champ « Vidéo de présentation ».*

*De plus, veuillez indiquer ci-dessous ce lien :*

Lien vers la vidéo youtube : <https://www.youtube.com/watch?v=YEUy0jG4v_I>.