Ergonomie & développement d’une SPA animée

**Incognito**

|  |  |
| --- | --- |
| **Auteur 1** | Gabriel De Herdt |
| **Auteur 2** | Quentin Garwig |
| **Auteur 3** | Mohamed Senhaji |
| **Date** | 18.12.2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-11 |
| **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. |

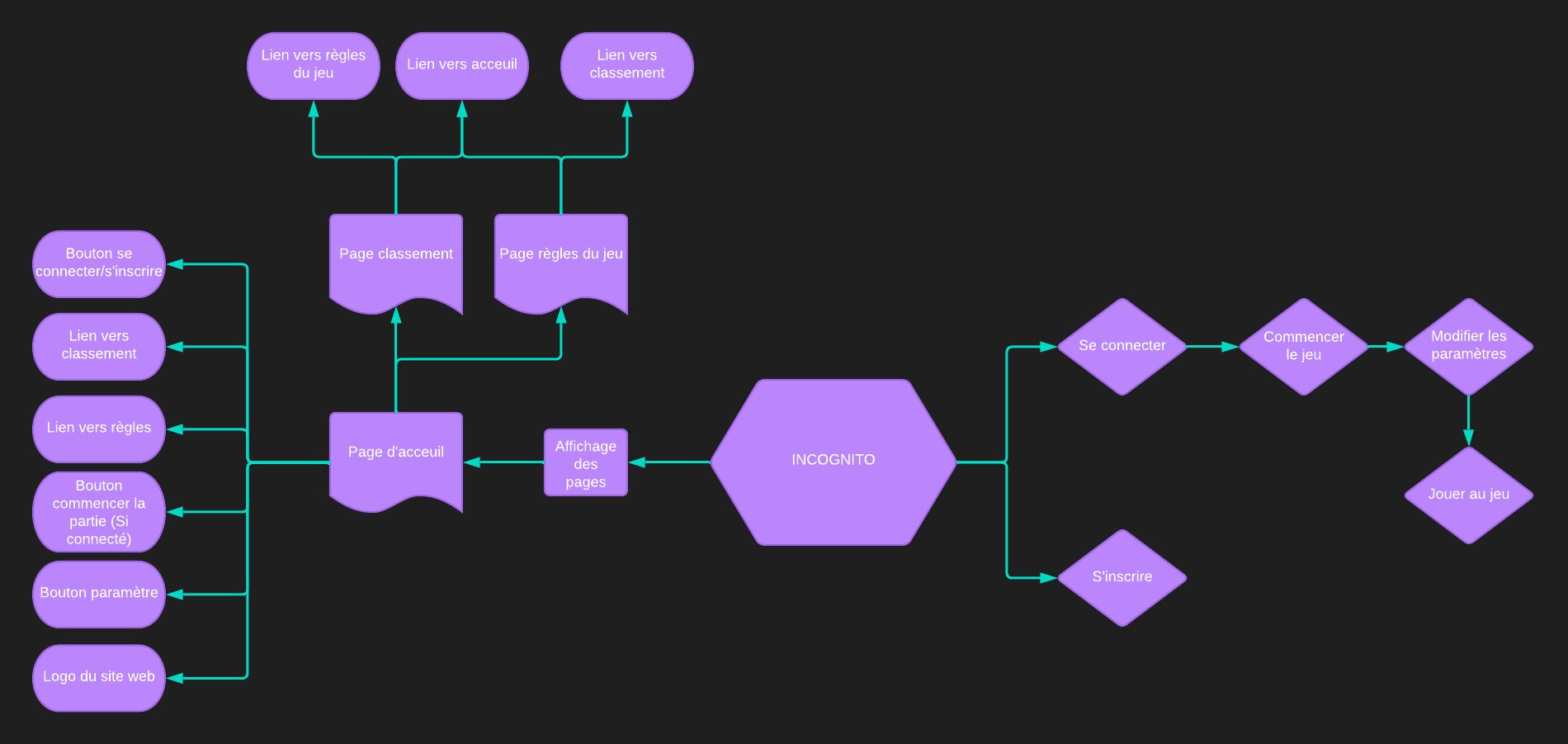
# Objectif du projet

Notre projet se base sur un jeu de sociéte qui se joue avec des cartes spéciales. Nous voulions donc l’adapter pour qu’il puisse se jouer en ligne sur un seul ordinateur. Afin que l’on puisse y jouer entre amis, au cours d’une après-midi ou en soirée. Il est donc fait principalement pour des adolescents mais est bien évidemment jouable par des familles et donc enfants.

Nous aimons bien ce jeu et ne l’avons pas toujours sur nous lors de nos déplacement. Il est donc plus simple de l’avoir sur internet et pouvoir y jouer n’importe quand et n’importe quand tant qu’une connexion internet est disponible.

*Veuillez aussi donner un nom à votre projet que vous compléterez sur* [*https://e-vinci.github.io/web2/project-page*](https://e-vinci.github.io/web2/project-page) *ainsi que sur la première page de ce rapport.*

# Mind map du projet



# 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.*

*Une image contenant texte

Description générée automatiquement*

**

# Axiomes de Morville

*Une image contenant table

Description générée automatiquement*

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

# Planning des tâches et cas d’utilisation

*.*

* URL vers votre GitHub Project public : https://github.com/orgs/e-vinci/projects/45

# 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

*Principalement pour la ou les librairies JS non vue en cours que vous avez ou allez appliquer, veuillez décrire la technologie choisie pour répondre à un ou plusieurs cas d’utilisation identifié(s) au §6. Ce paragraphe est à rédiger sur environ 20 lignes. Veuillez décrire les liens vers le ou les sites à utiliser (ou utilisés), les ressources principales utiles au développement.*

## 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

*Random-int est un package permettant de générer un nombre aleatoire. Je l’utilise donc pour récupérer un mot aleatoire en créant une partie.*

[*https://www.npmjs.com/package/random-int*](https://www.npmjs.com/package/random-int)

*https://www.npmjs.com/package/random-number*

## Wireframe

# Conception & Implémentation

## Code repositories

* URL pour le web repository public associé à votre projet : https://e-vinci.github.io/web2-2022-project-group-11/party

## 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

*Veuillez documenter les opérations de votre API, soit à l'aide de tableaux à donner ci-dessous, comme ceux vu dans le cours, soit en référençant la documentation qui aurait été générée à l'aide d'outils (Swagger par exemple). Votre fichier README de votre projet doit indiquer l'endroit où se situe la documentation de votre API.*

* 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/web2-2022-project-group-11/

* URL de votre RESTful API déployée : https://incognito-ipl-api.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

Gabriel : Dans le gros du projet, on a pu atteindre tous les objectifs qu’on s’était fixé. Malheureusement, il nous manque quelques fonctionnalités en plus qu’on aurait aimé ajouter. Faute au manque de temps, on a dû les laisser de coter.

Mohammed : Moi j’estime que dans l’ensemble du projet, on a atteint ce qu’on s’était fixés à faire de base mais comme Gabriel a dit, plus le projet avançait, plus on avait l’ambition d’ajouter une tonne de fonctionnalités, finalement manque de temps mais ce n’est qu’une partie minime du gros travail qu’on a effectué. Donc je dirais plutôt qu’on a même dépassé nos espérances et qu’on est plus que satisfait du site.

Quentin : Je pense la même chose qu’eux, et j’ajouterai juste que nous avons fini notre projet un peu en dernière minute mais que si nous avions voulu nous contenter du minimum, on aurait fini vraiment à l’avance, peut-être qu’on a été un peu trop gourmands mais finalement ça a payé et on est super fiers de notre jeu. En termes des objectifs non atteints, on aurait aimé rajouter un peu plus d’animations js et quelques fonds sonores aussi.

## Audit ergonomique de votre projet

*Veuillez utiliser les outils vus en cours pour analyser l’ergonomie de votre projet et si les règles de GDPR sont respectées. Créer ensuite ci-dessous un rapport à rédiger d’environ 10 lignes.*

L’ergonomie du projet est excellente dans sa globalité, en effet, l’UX est respectée dans toutes ses catégories :

* Utilité : le jeu est utile dans le sens ou il apporte une bonne distraction et amusement aux utilisateurs. De plus il est inspiré d’un jeu existant populaire et est donc très addictif.
* Utilisable : avec seulement 4 icones dans la navbar, des couleurs sublimes, des animations CSS, un design épuré et l’absence de déchets, le site sera très facile d’utilisation pour toute tranche d’âge.
* Désirable : Comme cité dans le point précédent, le site est très attractif et sait attirer l’œil grâce a son design épuré et responsive. Il est clair qu’il donne envie de jouer au jeu !
* Accessible : nous n’avons pas fait énormément en termes d’accessibilité pour les personnes déficientes mais vu que c’est un jeu de société, elles sauront se faire guider, de plus en revenant au point deux, avec sa facilité d’utilisation, elles auront tout de même un peu plus facile.
* Crédible : Le site respecte bien les derniers standards du W3C dans sa globalité.
* Trouvable : le site est trouvable grâce a son lien github, après, il est plus destiné aux élèves de l’IPL qui pourront avoir un accès au repo.

Le site respecte dans sa globalité aussi les règles du GDPR et ne force pas les utilisateurs à confier leurs données. Mais nous constatons quelques failles dans la protection des données.

## Difficultés techniques rencontrées

Gabriel : J’ai rencontré quelques problèmes qui m’ont mis du temps à résoudre. Notamment avec le querySelectorAll() d’une classe, Il m’a fallu du temps pour comprendre comment les connecter à un addEventListener(‘click…). EsLint m’a beaucoup aidé mais m’a apporté des problèmes aussi, des bugs dans mon code qui sans ESLINT n’ont pas lieu d’être.

Mohammed : J’ai rencontré des problèmes avec le main.innerHTML, en effet lorsque je voulais lancer une exception dans un form, celle-ci se lançait directement dans le main j’ai pu finalement régler le problème en créant un : ‘main = <div exemple etc….’ et donc je faisais un inner directement dans la div. J’ai aussi eu des problèmes avec le déploiement du front mais j’ai su ajouter le fichier deploy-frontend.yml.

Quentin : J’ai rencontré des problèmes avec ESLINT, même avec l’extension installée, ça ne me voulait pas afficher les erreurs. Également lorsque j’ai voulu importer des nouveaux packages en backend, le déploiement en back n’a pas été facile l’url du back me menait vers une page et j’avais une erreur que j’ai eu du mal à régler.

## Conseils pour appliquer cette technologie

Gabriel : Ça parait logique mais google est vraiment d’une grande d’aide, si on bloque, le problème a souvent déjà été resolu par qlq d’autre qui pourra vous aider sur un forum.

Je me suis servi de pas mal de generator pour le css (Style box-shadow, clip-path, gradient(),…)

Mohammed : Je leur conseillerais de bien lire les javadoc, de ne pas hésiter à chercher sur internet sur google ou youtube si ça ne fonctionne pas. Ne pas hésiter à poser la question sur StackOverflow, en géneral les gens répondent vite. De bien suivre les étapes lors d’un déploiement ou d’une installation parce qu’il peut manquer une seule étape et tout crash. J’aurais aussi aimé être plus régulier car « try-hard » en dernière minute c’est ultra-fatiguant et ça mène à de nombreux autres problèmes.

Gabriel : Si je peux conseiller des futurs développeurs front, back ou full, je leur dirais déjà d’être propre dans leur code. Un code mal indenté est la cause de beaucoup d’erreurs car on s’y retrouve mal. Toujours dans « l’ergonomie » du code, de bien nommer ses variables, on peut vite oublier à quoi elles servent ou se tromper dans le code. Et de bien commenter, si jamais tu retournes sur le code 1 semaine après, tu sais ce que tu as fait. J’aurais aimé savoir tout ça avant de commencer car 50% de mes problèmes sont issus de là.

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

Gabriel : On est un groupe qui s’entend très bien, c’est le plus gros point positif. Pas de soucis au niveau de la cohésion du groupe.

Mohammed : On se connaissait déjà avant la formation du groupe, donc on connaissait déjà un peu les facilités et difficultés de chacun dans le coding, ça nous a permis de mieux se répartir les taches et puis d’être plus à l’aise en travail de groupe.

Quentin : Je dirais que notre plus grosse force a été le groupe Discord. En effet, on postait tous nos problèmes là-bas directement et puis les sessions appel pour travailler ensemble ont été une grosse force de motivation. Les membres du projet sont tous sympas, et c’est une grosse satisfaction et un énorme rapprochement que de monter un projet tous ensemble.

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

Gabriel : Se voir plus souvent dans une classe pour améliorer la collaboration.

Mohammed : J’aurais aimé venir plus souvent en classe, je pense avoir pu pénaliser mon équipe dans quelques cas même si mon travail a toujours été fait, ne serait ce que dans la collaboration, car même si Discord c’est bien, c’est mieux de se voir en vrai pour travailler.

Gabriel : Se répartir les tâches plus tôt, car si chacun avance tout seul il se peut qu’il y ait plus de travail pour une personne. Et donc pour cela bien discuter du projet et de sa structure, de ses algorithmes, pour évaluer la difficulté de chaque partie et d’avoir un travail équitable.

# Analyses individuelles des résultats

*Les analyses individuelles sont à réaliser via les sessions de feedback qui seront à soumettre individuellement à la fin de chaque semaine de cours, pendant le WE, via l’outil TEAMMATES (*[*https://teammatesv4.appspot.com*](https://teammatesv4.appspot.com)*).*

*Vous recevrez des e-mails vous invitant à compléter votre feedback hebdomadaire sur le projet. Attention aux pénalités si vous ne complétez pas votre feedback.*

*NB : Ce §11 peut être entièrement effacé du rapport que vous soumettrez sur Moodle.*

# Présentation vidéo

Lien vers la vidéo youtube : https://youtu.be/HScQy6vUyOU

# Revues de projets par les pairs

*Une fois la deadline de soumission des projets atteinte, la saison de revues des projets sera ouverte !*

*Le but ?*

* *Participer à la revue bienveillante, aux critiques constructives, de sites web faits par vos pairs ;*
* *Identifier les projets qui plaisent, notamment afin d’améliorer leur visibilité !*

*Pour vos revues de projets, voici les règles :*

* *5 revues sont attribuées automatiquement à chaque membre d’un projet, ainsi qu’un coup de cœur lorsque vous accédez à* [*https://e-vinci.github.io/web2/my-reviews-page*](https://e-vinci.github.io/web2/my-reviews-page)*.*
* *Toute revue doit comprendre au moins 1 point fort identifié et 1 point d’amélioration ; vous pouvez baser ces points suite au visionnage de la vidéo uniquement, mais nous vous recommandons de le faire après avoir exécuté l'application associée au projet ; vous pouvez bien sûr aussi accéder au code de l’application pour votre revue.*
* *Libre à vous de vous attacher au design, au gameplay, à l'ergonomie, au code, à la vidéo ou tout autre aspect dans votre revue. Chaque revue sera affichée – ainsi que votre nom – dans le détails d’une revue. Soyez donc bienveillants et constructifs dans votre analyse critique ; )*
* *Les résultats des revues ne sont accessibles qu’à vos pairs ! Votre analyse critique n’est donc pas publique, seuls les membres d’un projet de Web2 y ont accès, uniquement pour les membres ayant réalisé au moins 5 revues. Les projets sont listés par nombre de coups de cœur reçus, par nombre de revues faites, puis simplement par ordre alphabétique.*
* *Une revue soumise peut  être modifiée. Nous vous conseillons, avant de donner un coup de cœur, d’avoir jeté un œil à tous les projets qui vous intéressent ; un coup de cœur donné ne peut pas être retiré ; )  
  Par contre, vous pouvez faire vos revues tranquillement, puis mettre à jour celles-ci plus tard pour attribuer vos coups de cœur.*
* *Après avoir effectué vos 5 revues attribuées, vous obtenez :*
  + *un deuxième coup de cœur à offrir,*
  + *l’accès aux résultats généraux des revues et aux détails de chacune des revues.*
  + *Le pouvoir de revoir n’importe quel projet non encore revu.*

*Comme l’hébergement gratuit d’API devient de plus en plus compliqué, nous souhaitons garantir que les 3 projets les plus aimés puissent bénéficier d’un hosting offert et géré par la HE Vinci.*

*NB : Ce §13 peut être entièrement effacé du rapport que vous soumettrez sur Moodle.*