Ergonomie & développement d’une SPA animée

Qui veut gagner rien ?

|  |  |
| --- | --- |
| **Auteur 1** | Nathanael Afangbedjee |
| **Auteur 2** | Ali Elhannouti |
| **Auteur 3** | Lucas Gregoire |
| **Auteur 4** | Rafael Millor |
| **Auteur 5** | Yohani Mwananteba |
| **Date** | 15.12.2023 |
| **Référence** | WEB2-2023-PROJECT-GROUP-11 |
| **Version** | 1.0 |

Contents

[1 Consignes et évaluations 3](#_Toc147308156)

[1.1 Consignes générales 3](#_Toc147308157)

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

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

[1.1.3 Projet 4](#_Toc147308160)

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

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

[2 Objectif du projet 11](#_Toc147308163)

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

[4 Persona 12](#_Toc147308165)

[5 Axiomes de Morville 13](#_Toc147308166)

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

[7 Besoins techniques 16](#_Toc147308168)

[7.1 Système 16](#_Toc147308169)

[7.2 Frontend 17](#_Toc147308170)

[7.3 API 17](#_Toc147308171)

[8 Choix technologiques 18](#_Toc147308172)

[8.1 Frontend 18](#_Toc147308173)

[8.2 RESTful API 18](#_Toc147308174)

[8.3 Wireframe 18](#_Toc147308175)

[9 Conception & Implémentation 19](#_Toc147308176)

[9.1 Code repositories 19](#_Toc147308177)

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

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

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

[9.5 Code réutilisé 20](#_Toc147308181)

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

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

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

[10.3 Difficultés techniques rencontrées 21](#_Toc147308185)

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

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

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

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

[12 Présentation vidéo 22](#_Toc147308190)

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

# Consignes et évaluations

## Consignes générales

### Création des groupes sur le site du cours

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

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

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

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

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

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

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

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

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

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

Veuillez identifier le membre qui créera votre équipe sur GitHub.

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

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

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

Un web repository aura été créé pour votre équipe.

#### Joindre une équipe existante

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

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

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

### Projet

Vous allez créer une SPA mettant en œuvre :

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

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

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

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

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

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

## Ergonomie : consignes techniques, timing et évaluations

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

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

# Objectif du projet

Nous avons décidé de développer un site imitant le jeu télévisé « Qui veut gagner des millions ».

Elle vise tout public qui veux tester sa culture générale. On peut sélectionner les thèmes qu’on veut. En vous connectant, vous pouvez sauvegarder votre score et un Leaderboard montrant les meilleurs scores est visible. Ajoutant un petit côté compétitif.

Le site s’appelle « Qui veut gagner rien ? » pour parodier « Qui veut gagner des millions ».

Le site permet de :

* Lancer un quizz (20 questions)
* Lancer un quizz avec seulement certains thémes
* S’inscrire (et se connecter) pour pouvoir garder votre meilleur score
* Visualiser les meilleur joueur (avec un tableau des scores)
* Ajouter des questions (si connecté)

# 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/71

# 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 avoir des messages d’erreur rapide et joli (Popup d'error lors de problème #34) nous a avons utilisé la librairie Sweeralert2 : *https://sweetalert2.github.io*

Pour faire des animations simples(Ajouter des animations #11) nous avons utilisé animejs :[*https://animejs.com*](https://animejs.com/)

## Frontend

[https://sweetalert2.github.io](https://sweetalert2.github.io/)

[*https://animejs.com*](https://animejs.com/)

## RESTful API

*https://www.npmjs.com/package/jsonwebtoken*

*https://www.npmjs.com/package/bcrypt*

## Wireframe

# Conception & Implémentation

## Code repositories

* URL pour le web repository public associé à votre projet : https://github.com/e-vinci/web2-2023-project-group\_11.git

## 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** |
| --- | --- | --- |
| **auth/register** | POST | CREATE ONE : Cree un nouvel user |
| **auth/login** | POST | READ ONE : Récupère un user |
| **auth/logout** | GET | READ ONE : Supprime les cookies de session |
| **users/** | GET | READ ALL : Récupère tous les user |
| **users/getScore ?username=value** | GET | READ ONE : Récupère le score de l’user donné |
| **users/updateBestScore** | PATCH | UPDATE ONE : change le meilleur score d’un user donné |
| **quizz/** | GET | READ ALL: Lire toutes les ressources de la collection |
| **quizz/20** | GET | READ ALL : Lire seulement 20 ressource non triée |
| **quizz/20 ?categories=value1,value2** | GET | READ ALL : Lire seulement 20 ressource triée |
| **quizz/{id}** | DELETE | DELETE ONE : Effacer la ressource identifiée |
| **posts/addQuestion** | POST | CREATE : Crée une nouvelle question |

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

## Déploiement de vos applications

* URL de votre frontend déployé : https://rafael-millor-vinci.github.io/
* URL de votre RESTful API déployée : https://rafael-millor-vinci-web2.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** |
| --- | --- | --- | --- |
|  |  |  |  |

# Analyse des résultats par le groupe

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

La totalité des objectifs fonctionnels de haute priorité ont été atteints.

Cependant, au cours du développement nous avons décider d’ajouter un timer pour répondre au question et le nombre de point dépend du timer, de même nous avons rajouter un système de « streak », où plus on répond correctement au question d’affilé plus on reçoit de points.

Ces changement ont été ajouter afin de rendre le quizz plus compétitif, car initialement le score obtenable étais entre 0 / 200 se qui limite l’aspect unique de chaque score.

Tous les objectif fonctionnel de moyenne priorité n’ont pas été atteints l’ajout ou suppression de question est possible seulement coté serveur et pas implémenté dans le site

Toutes les autre objectif ont été atteint.

## Audit ergonomique de votre projet

Notre projet étant un quizz, il est adapté à tous le monde. Comme détaillé dans nos personnas, le jeu est adressé aux personnes de toute age, sachant se servir d’un ordinateur. Les personnes mal-voyantes peuvent jouer à notre jeu.

Toutes les pages possèdent un raccourci vers la page d’accueil dans la barre de navigation.

Le site est prévu pour jouer sur PC avec un grand écran, il est responsive. Cependant, l’affichage sur des très petits écrans comme un téléphone peut ne pas être compatible.

Les boutons sont explicite et accessibles.

Les règles de GDPR sont respecté.

Notre site ne récolte pas de données autre qu’un potentiel login. Les données de nos utilisateurs sont stockées en toute sécurité sur les serveurs hébergeant le projet. Le mot de passe entré par nos utilisateurs est vérifié pour être assez résistant et est stocké de manière cryptée dans la base de données.

## Difficultés techniques rencontrées

Le développement du jeu en SPA nous a posé quelques difficultés au début mais nous avons rapidement pris un rythme et aucun problème bloquant n’a été rencontré.

La majorité des problème rencontré ont été résolu rapidement.

La plus grande difficulté a été le déploiement sur le backend. Azure app service n’ont renvoyait des messages d’erreur 2 fois sur 3.

## Conseils pour appliquer cette technologie

Le plus compliqué a été de savoir par où commencer.

Mais en suivant étape par étape les tutoriel proposé par le syllabus, d’éventuelle question a chatGPT et de s’aider d’exemple trouvé sur d’autre site de quizz.

Surtout il faut commencer par quelque choses de simple et l’améliorer ensuite.

Suivre la documentation est le meilleur moyen d’avancer et de trouver des solution.

Il ne faut pas hésiter a posé des question a d’autre camarade, aux professeur ou sur des forum.

Lien : https://e-vinci.github.io/web2/

https://quipoquiz.com/fr/index

https://chat.openai.com/

https://stackoverflow.com/

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

La collaboration entre les membres du groupe était très bonne. Nous nous somme réparti le travail et avons chacun travaillé sur différentes parties du projet (frontend, jeu,login et register, backend).

Cela nous a permis d’avancer en parallèle et de ne pas devoir attendre la partie de quelqu’un pour avancer.

Si quelqu’un rencontrait un problème nous nous aidions entre nous. Le fait de s’entraider nous a permis de ne pas rester coincé sur une tâche

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

Mieux attribué les taches et surtout créée plus de sous tâche. Commencer plus tôt certaines fonctionnalité qui nous a mis en retard sur le planning et on a du un peu rush sur les dernière semaines. L’ajout de certaine features s’est fait retardé ou même supprimer par manque de temps. Nous avons voulu ajouter des fonctionnalité trop conséquente dé le début et non commencé par simple et évolué sur base de ça.

# Présentation vidéo

Lien vers la vidéo youtube : <https://www.youtube.com/watch?v=vAUlCZ1ybEY&ab_channel=LucasGrégoire>

# 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 §12 peut être entièrement effacé du rapport que vous soumettrez sur Moodle.*