

Ergonomie & développement d'une SPA animée

SocialSync

Auteur 1	Efe ALAGOZ
Auteur 2	Nicolas HEYMANS
Auteur 3	Filippo FAIELLA
Auteur 4	Mykhailo KRETSU
Auteur 5	Gauthier COLLARD
Date	17.12.2023
Référence	WEB2-2023-PROJECT-GROUP-16
Version	1.0

Contents



1	Consignes et évaluations	3
1.1	Consignes générales	3
1.1.1	Création des groupes sur le site du cours	3
1.1.2	Création d'un groupe sur GitHub Classroom et du web repo associé	4
1.1.3	Projet	4
1.2	JavaScript & Node.js : consignes techniques, timing et évaluations	6
1.3	Ergonomie : consignes techniques, timing et évaluations	11
2	Objectif du projet	12
3	Mind map du projet	12
4	Persona	12
5	Axiomes de Morville	12
6	Planning des tâches et cas d'utilisation	13
7	Besoins techniques	16
7.1	Système	16
7.2	Frontend	16
7.3	API	17
8	Choix technologiques	17
8.1	Frontend	18
8.2	RESTful API	18
8.3	Wireframe	18
9	Conception & Implémentation	18
9.1	Code repositories	18
9.2	Secrets éventuels pour vos API ou base de données	19
9.3	Documentation de votre API	19
9.4	Déploiement de vos applications	19
9.5	Code réutilisé	20
10	Analyse des résultats par le groupe	20

- 10.1 Évaluation du résultat par rapport au planning des tâches et des cas d'utilisation²⁰
- 10.2 **Audit ergonomique de votre projet**²¹
- 10.3 Difficultés techniques rencontrées²¹
- 10.4 Conseils pour appliquer cette technologie²²
- 10.5 Quels sont les points positifs à la manière dont s'est déroulée la collaboration au sein du groupe ?²³
- 10.6 Quels sont les points qui seraient à améliorer pour de futures collaborations ?²³
- 11 Analyses individuelles des résultats **Erreur ! Signet non défini.**
- 12 Présentation vidéo²³
- 13 Revues de projets par les pairs²⁵

1 Consignes et évaluations


1.1 Consignes générales

1.1.1 Création des groupes sur le site du cours

Veillez 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.

1.1.2 Création d'un groupe sur GitHub Classroom et du web repo associé

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

Veillez passer à cette étape qu'une fois votre groupe déjà finalisé sur le site du cours.

1.1.2.1 Création de l'équipe associée à un projet

Veillez 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.

1.1.2.2 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](#).

1.1.3 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

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

Tâche	Compétences	Critères	Deadlin e	Pt	Consignes
Objectif du projet	C6) Documenter et présenter en vidéo le développement d'une SPA		Séance 9 : 16/10 ou 17/10		<p>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 :</p> <ul style="list-style-type: none"> - Le nom du projet : Projet N°X : Nom de votre projet - le champs « Description ». <p>Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d'aller plus loin dans votre projet.</p>
Planning des tâches et cas d'utilisation	C6)		Séance 12 : 24/10 ou 27/10		<p>Décrire le planning des tâches et cas d'utilisation selon les instructions données au §6.</p> <p>Présenter votre planning à un enseignant, afin qu'il puisse vous aider à bien prioriser les tâches.</p>
Indiquer l'URL de votre code repository	C6)		Séance 12 : 24/10 ou 27/10		<p>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.</p> <p>Plus d'information aux §1.1.2 et §9.1.</p>

Tâche	Compétences	Critères	Deadline	Pt	Consignes
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 <i>Indicateurs : formulation de qualité, analyse de qualité, respect des consignes</i>	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 2 ^{ème} 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 <i>en grisé</i> dans ce document avant de soumettre ce rapport sur

Tâche	Compétences	Critères	Deadline	Pt	Consignes
					Moodle.
Soumission de la vidéo	C6)	Vidéo de qualité <i>Indicateurs : présentation du projet de qualité, analyse de qualité, respect des consignes</i>	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 <i>Indicateurs : esthétique, fonctionnel, codage de qualité, respect des consignes, ambitieux & original, utilisation d'une librairie pour l'IHM non vue en cours</i>	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...).
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 <i>Indicateurs : fonctionnel, codage de qualité, respect des consignes, ambitieux & original, utilisation d'une librairie pour le service web non vue en cours</i>	17/12	5	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.
Déploiement tant	C4) Déploiement	Déploiement de la SPA sur le	17/12	2	

Tâche	Compétences	Critères	Deadline	Pt	Consignes
de votre frontend que backend	d'applications web	cloud <i>Indicateurs : fonctionnel, performances de chargement acceptables</i>			
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 <i>Indicateur : présence d'un minimum de 5 revues</i>	Avant examen de 1^{ère} 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 cotés 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 participé significativement au projet recevront d'office une

Tâche	Compétences	Critères	Deadline	Pt	Consignes
					lourde pénalité au niveau de leurs points, voire un 0/20.

1.3 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		<p>Décrire l'objectif de votre projet au §2 de ce document.</p> <p>Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d'aller plus loin dans votre projet.</p>
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 cotés différemment en fonction de leur engagement sur le projet visible lors des sessions de cours.

2 Objectif du projet

Le projet que je souhaite développer est une plateforme en ligne qui offre la possibilité d'utiliser les réseaux sociaux de manière anonyme. Cette plateforme permettra aux utilisateurs de suivre et de consulter le contenu de différentes plateformes de réseaux sociaux sans avoir à créer de compte sur ces dernières ou à devoir suivre les utilisateurs directement sur ces plateformes.

Cette application s'adresse à toute personne souhaitant naviguer sur les réseaux sociaux de manière discrète, préservant ainsi son anonymat tout en restant connectée aux contenus et aux actualités partagés par des individus ou des organisations sur des plateformes telles que X, sans la nécessité de s'engager dans la création de comptes ou de divulguer des informations personnelles.

Le concept de cette plateforme prend à cœur la protection de la vie privée et la liberté d'explorer le contenu des réseaux sociaux sans les contraintes habituelles. Cela m'importe car je crois fermement en la préservation de la confidentialité en ligne et en l'autonomie des individus quant à la manière dont ils souhaitent interagir avec les médias sociaux. La possibilité d'accéder aux actualités et aux publications sans compromettre son anonymat permettrait aux utilisateurs de se sentir plus en contrôle de leur expérience en ligne, favorisant ainsi une utilisation plus libre et plus éthique des réseaux sociaux.

3 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.*

4 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.*

5 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'efficacité.

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

6 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**...

Veuillez regarder la vidéo qui suit pour mettre en place votre GitHub Project. Cette année, si vous souhaitez accélérer la création de votre GitHub Project, vous pouvez sélectionner comme organization templates : **web2-project-backlog-template** au lieu de **Team backlog** comme montré dans la vidéo.

The screenshot shows the GitHub Project creation interface. On the left, under 'Organization templates', the 'web2-project-backlog-template' is selected. The main area displays the details for this template:

- Project name:** web2-2023-project-group-00
- Views:** 4 (Backlog, By priority, By size, View 4)
- Fields:** 11 (Title, Assignees, Status, Labels, Linked pull requests, Reviewers, Repository, Milestone, Priority, Size, Scope)
- Workflows:** 2 (Item closed, Pull request merged)
- Insights:** 1 (Priority)

A green 'Create' button is visible at the bottom right.

Si vous préférez faire exactement comme dans la vidéo et créer manuellement toutes les tâches, vous utiliserez le template : **Team backlog**.

Voici la vidéo expliquant la mise en place de votre GitHub Project : [Partie 1 : Créer et configurer un GitHub Project](#).

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](#).
NB : si vous avez accéléré la création de votre GitHub Project en sélectionnant le template **web2-project-backlog-template**, reste intéressante pour voir comment créer des drafts issues, convertir des draft issues en issues, ajouter des Labels aux issues...
- Chaque **tâche** doit être couverte par une **draft Issue** au sein de GitHub 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](#).
- 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 travaillez 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 (c'est les draft issues qui sont offertes dans le **web2-project-backlog-template**) :

- Create marketing vision (documentation)
- Create wireframes (documentation)
- Identify list of use cases (documentation)
- UC1 : ... (enhancement)
 - o UC1-frontend : ... (enhancement)
 - o 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.

- URL vers votre GitHub Project public :
<https://github.com/e-vinci/web2-2023-project-group-16>

7 Besoins techniques

*La spécification technique des besoins vous est donnée
N'hésitez pas à la compléter s'il manque qqch d'important.*

7.1 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.

7.2 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.

7.3 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.

8 Choix technologiques

TailwindCSS : Nous avons opté pour l'utilisation de Tailwind pour nous permettre de créer des interfaces modernes, rapides et user-friendly. De plus, Tailwind nous facilite aussi la tâche en générant du CSS complexe en n'écrivant que quelques mot-clefs.

TailwindAnimated : Nous avons opté pour l'utilisation de TailwindAnimated car la librairie nous

permet, comme TailwindCSS, de très facilement implémenter des animations, de plus la librairie possède un outil en ligne très intuitif et qui a une bonne prise en main pour créer des animations. DaisyUI : Nous avons opté pour l'utilisation de DaisyUI car c'est une librairie de composants de TailwindCSS qui nous propose des exemples pré-faits sur lesquels on a pu s'inspirer pour la conception de notre site web.

PocketBase : Nous avons opté initialement pour PocketBase car c'est un logiciel de création de base de données intuitif inspiré de Firebase et Supabase, qui ont fait leurs preuves, le logiciel est simple d'utilisation et possède une interface graphique qui aide à la gestion de bases de données, suite à des difficultés rencontrées lors de son utilisation, notamment dans l'accès de plusieurs éléments situé dans plusieurs table (car PocketBase < v20.0 ne prend pas des requêtes mais des fonctions), donc nous avons décidé de le remplacer par PostgreSQL pour nous faciliter la tâche car c'est une technologie que nous avons vu en cours.

PostgreSQL : PostgreSQL est un système de gestion de bases de données relationnelles et objets dont nous avons fait l'utilisation en cours. Suite aux problèmes rencontrés lors de l'utilisation de PocketBase, nous sommes passés à PostgreSQL qui s'est avéré plus intuitif pour nous.

8.1 Frontend

- TailwindCSS : <https://tailwindcss.com/>
- TailwindAnimated : <https://www.tailwindcss-animated.com/>
- DaisyUI : <https://daisyui.com/>

8.2 RESTful API

- node-postgres: <https://node-postgres.com/>

8.3 Wireframe

Veillez 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.

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

9 Conception & Implémentation

9.1 Code repositories

Repo github généré en cours : <https://github.com/e-vinci/web2-2023-project-group-16>

Repo github du frontend : <https://github.com/Brulans4/SocialSync>

Repo github du backend : https://github.com/Brulans4/SocialSync_backend

- URL pour le web repository public associé à votre projet :
<https://github.com/e-vinci/web2-2023-project-group-16>

9.2 Secrets éventuels pour vos API ou base de données

9.3 Documentation de votre API

URI	Méthode	Contenu
/auths/register	POST	Renvoie "OK" si l'utilisateur est inscrit
/auths/login	POST	Renvoie le username et son token
/dbUtils/	GET	Renvoie la liste des influenceurs
/dbUtils:id	GET	Renvoie les informations d'un seul influenceur
/dbUtils2/	GET	Renvoie la liste des plateformes
/users/subscribe	POST	Abonne l'utilisateur à un influenceur
/users/unsubscribe	POST	Désabonne l'utilisateur d'un influenceur
/users/	GET	Renvoie la liste des abonnements
/users/deleteUser	POST	Supprime toutes les informations d'un utilisateur et l'utilisateur lui-même.

9.4 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 ».

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

- URL de votre frontend déployé : <https://brulans4.github.io/SocialSync/>
- URL de votre RESTful API déployée : <https://socialsync.azurewebsites.net/>

9.5 Code réutilisé

Veillez 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
/api/models/*	Raphael Baroni	https://e-vinci.github.io/web2/	Code vu en cours.
/frontend/pages/*	DaisyUI	https://daisyui.com/components/	Création du css du frontend

10 Analyse des résultats par le groupe

10.1 Évaluation du résultat par rapport au planning des tâches et des cas d'utilisation

Veillez décrire si vous avez pu atteindre vos objectifs fonctionnels et autres tâches. Si vous avez dévié des objectifs fonctionnels identifiés initialement, veuillez décrire les raisons de ce changement.

Ce paragraphe est à rédiger sur environ 15 lignes.

Nous avons initialement pas mal d'idées au cours du projet mais surtout au début de celui-ci concernant l'affichage des feeds et posts de réseaux sociaux mais nous nous sommes rendu compte que nous étions beaucoup plus limités que nous le pensions à ce niveau-là. Nous pensions au début faire du « scraping » qui est un processus automatisé d'extraction d'informations ou de données depuis différentes sources en ligne mais nous nous sommes très vite rendu compte que c'était illégal. Nous avons donc du changer de direction pour remplir notre base de données et nous sommes donc directement tournés vers l'outil d'incorporation X (Twitter)

afin d'afficher toutes les publications d'un compte. Il était également prévu d'afficher des posts de différents réseaux sociaux sur la page principale afin de diversifier l'affichage mais cette fonctionnalité s'est également avérée plus compliquée que prévu à concrétiser. Même si nous en sommes donc réduits à un seul réseau social (X), le site reste dynamique et prêt à en accueillir d'autres.

Bien que les premières idées aient évolué, la suite du projet s'est déroulée avec une meilleure clarté et fluidité étant donné que nous connaissions les limitations auxquelles nous faisons face.

10.2 Audit ergonomique de votre projet

Au moment de l'inscription d'un utilisateur sur notre plateforme, nous avons intégré une case spécifique qui requiert l'acceptation explicite du Règlement Général sur la Protection des Données (RGPD). Cette démarche vise à garantir que chaque individu inscrit comprend et consent à la manière dont ses données seront collectées et traitées. De plus, chaque utilisateur a la possibilité de consulter à tout moment les informations personnelles collectées en accédant à la section "Compte". Cette transparence permet à chacun de vérifier l'exactitude et l'utilisation de ses données conformément aux directives du RGPD.

En conformité avec la réglementation, nous offrons également la possibilité à chaque utilisateur de supprimer son compte du site. Ce processus inclut la suppression définitive des données personnelles de notre base de données, comme l'exige la loi en matière de protection des données.

10.3 Difficultés techniques rencontrées

*Indiquez ici les difficultés techniques que vous auriez rencontrées au cours de votre projet.
Ce paragraphe est à rédiger sur environ 15 lignes.*

Nous avons rencontré pas mal de difficultés et de limitations notamment dues à l'inactivité de deux membres au sein de notre équipe malgré leur réactivité lorsque nous discutons. Malgré les efforts et l'engagement des autres membres, l'absence de contribution de ces deux personnes a restreint notre capacité à avancer efficacement et leur non-participation a généré des retards considérables dans la réalisation des objectifs fixés. Le projet rencontrait déjà des retards et nous pensions qu'en distribuant les tâches restantes, nous pourrions être certains de finaliser le site comme nous le désirions, mais même cela n'a pas été suffisant à ce que nous soyons tous actifs en étant proches de la date de remise limite.

Un problème avec la base de données est également survenu puisque nous utilisions initialement PocketBase pour manier notre base de donnée facilement. Même si nous pensions cette tâche facilitée, nous avons rencontré des limitations concernant l'utilisation de cette application ce qui nous a obligés à tout convertir et créer notre propre base de données. Réaliser de simples

requêtes et créer des « views » devenaient de véritables corvées avec l'application utilisée en premier lieu.

Ces combinaisons de difficultés humaines et techniques nous ont vraiment freinés dans l'accomplissement des objectifs initiaux, limitant non seulement notre capacité à répartir les tâches équitablement mais aussi le temps disponible pour réaliser le reste du projet.

10.4 Conseils pour appliquer cette technologie

Veillez résumer tout ce qui serait intéressant pour quelqu'un qui souhaiterait appliquer le même genre de technologie, sur base de ces questions :

Qu'est-ce que vous auriez aimé savoir avant de démarrer ?

Des conseils pour éviter certains problèmes rencontrés ?

Des liens vers les sites les plus intéressants ?

Etc.

Ce paragraphe est à rédiger sur environ 15 lignes.

Utiliser des technologies compatibles et adaptables pour chacune des parties du projet, permettant une intégration fluide et une collaboration efficace entre elles. De plus, il est important de maintenir une documentation claire pour faciliter la compréhension et la maintenance à long terme de ton projet. Enfin, adopte une approche sécurisée en appliquant des bonnes pratiques pour éviter les vulnérabilités, tout en restant flexible pour ajuster l'approche en fonction des besoins changeants du projet.

Avant de démarrer avec une base de données SQL, j'aurais aimé mieux comprendre les différentes options disponibles en termes de SGBD (Système de Gestion de Base de Données) et leurs cas d'utilisation spécifiques. Comprendre les avantages et inconvénients de chaque système aurait permis de choisir le plus adapté au projet.

En ce qui concerne Tailwind CSS, avoir une meilleure connaissance des classes de base et de la logique derrière son approche utility-first aurait été bénéfique. Comprendre comment organiser efficacement les classes et les composants pour éviter la duplication de code aurait simplifié le processus de stylisation.

Pour éviter des problèmes avec la base de données SQL, je recommande de bien planifier la structure de la base dès le départ. Définir les relations entre les tables, les index et les contraintes de manière claire et précise peut éviter des problèmes de performances et de cohérence des données plus tard.

En ce qui concerne Tailwind CSS, il est essentiel de comprendre comment personnaliser et étendre les classes de manière efficace pour répondre aux besoins spécifiques du projet sans compromettre la maintenabilité du code.

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

Veillez donner des arguments justifiant votre réponse. Cette réponse est à rédiger sur environ 10 lignes.

Avant que nous ne soyons plus que trois dans le projet, l'entente était bonne et nous arrivions à nous répartir les tâches de manière assez équitable. Malgré le rush nécessaire en fin de projet nous avons réussi à nous réattribuer les tâches entre nous de manière très efficace afin de réaliser la version la plus simple du projet que nous avons envisagé. Chaque membre a également apporté ses compétences uniques, ce qui permet de combiner des capacités diverses pour résoudre des problèmes de manière plus complète. Travailler en équipe a aussi favorisé la capacité à s'adapter aux changements qu'il y a pu avoir. Nous avons donc pu ajuster nos efforts et nos priorités en fonction des besoins changeants du projet ou des retours mutuels.

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

Veillez donner des arguments justifiant votre réponse. Cette réponse est à rédiger sur environ 10 lignes.

L'amélioration de l'organisation et de la communication représente clairement des aspects cruciaux pour de prochains projets puisque nous étions peu souvent au courant quant à l'investissement de certains de nos collaborateurs. Mieux connaître les outils et technologies que nous utilisions avant de commencer le projet avec aurait pu nous aider à commencer directement sur de bonnes bases et ne pas perdre de temps à complètement réadapter le code. Savoir combien nous étions à réellement travailler dès le début nous aurait aussi permis de travailler plus efficacement ainsi que de mieux se répartir la totalité du travail. Mettre en place des sessions de feedback régulières pour évaluer les progrès, identifier les problèmes et ajuster les stratégies si nécessaire aurait également été un plus.

Finalement, encourager la documentation continue des décisions prises, des changements effectués et des solutions adoptées pour assurer une continuité et faciliter l'apprentissage futur pourrait tout autant être bénéfique à l'avenir.

11 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 :
 - o sur la présentation de votre application web : exécution, en live, de votre API et du frontend ;
 - o 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/> 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/> : logiciel open source demandant un temps d'adaptation, mais permettant de faire énormément
 - <https://www.loom.com/> : logiciel pouvant être utilisé gratuitement sous réserve d'accepter un logo. Très facile d'utilisation.
 - <https://screencast-o-matic.com/> : logiciel pouvant être utilisé gratuitement sous réserve d'accepter un logo. Très facile d'utilisation.

Veillez indiquer le lien vers la vidéo youtube que vous avez créée sur le site <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=OZbkiw81H4A>

12 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>.*
- *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.