



Dossier de Projet

Titre de Développeur Web / Web Mobile

par

Christophe Meckler

Titre RNCP de niveau III
Développeur Web / Web Mobile
Novembre 2022

Remerciements

La réalisation de ce projet a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je voudrais tout d'abord adresser toute ma reconnaissance à toute l'équipe de PopSchool, pour leur travail et leur encadrement. Je tiens à remercier en particulier Muharrem Ozipek, mon formateur principal, pour les bases qu'il m'a transmises tout au long de ma formation.

Merci à mes collègues de promotions pour l'aide, la motivation qu'ils m'ont apporté et la bonne humeur qu'ils véhiculaient.

Je remercie enfin Pôle Emploi pour son dispositif de reconversion professionnelle qui m'a permis de m'orienter vers un domaine passionnant.

Sommaire

1. Compétences couvertes par ce projet.....	5
1.1 Liste des compétences mises en œuvre dans ce projet.....	5
2. Présentation du projet.....	5
2.1 Précisions sur les technologies utilisées.....	5
2.2 Le projet.....	6
2.3 Remarques.....	6
3. Travail Préliminaire.....	7
3.1 Utilisateur : expression du besoin et formulation du cahier des charges.....	7
3.1.1 Accueil.....	7
3.1.2 Contenu.....	8
3.1.3 S'inscrire.....	9
3.1.4 Se connecter.....	9
3.1.5 Mon compte.....	10
3.1.6 Lecteur.....	11
3.1.7 Modale Erreurs et Succès.....	12
3.2 Administrateur : expression du besoin et formulation du cahier des charges.....	12
3.3 Environnement humain et technique.....	13
3.3.1 Environnement humain.....	13
3.3.2 Planification et gestion du temps.....	13
3.3.3 Environnement technique.....	13
4 Spécifications.....	15
4.1 Spécifications fonctionnelles.....	15
4.1.1 Arborescence du site.....	15
4.1.2 Liste des fonctionnalités spécifiques:.....	16
4.1.3 Remarque.....	17
4.2 Spécifications techniques.....	18
4.2.1 Technologies utilisées pour le Front-end.....	18
4.2.2 Technologies utilisées pour le Back-end.....	19
4.3 Sécurité.....	20
5. Réalisation du projet.....	21
5.1 Mise en place du projet.....	21
5.1.1 Création préliminaires.....	21
5.1.2 Création du projet.....	22
5.2 Présentation détaillée de la page «Single_player_content».....	24
5.2.1 Vérifications préliminaires.....	24
5.2.2 Suite des vérifications.....	26
5.2.3 Contenu visible de la page.....	27

6. Jeu d'essai du comportement lié à une fonctionnalité.....	31
6.1 Étapes préalables:.....	31
6.2 Test de la fonctionnalité d'édition d'un contenu.....	33
6.2.1 Validation données côté client.....	33
6.2.2 Test de la fonctionnalité d'édition d'un contenu côté serveur.....	37
7. Veille sur les questions de sécurité.....	44
8. Situation de travail ayant nécessité une recherche.....	47
8.1 Extrait du site en version originale.....	47
8.2 Traduction effectuée sans traducteur automatique.....	48
8.3 Extrait du code CSS obtenu suite à cette recherche.....	49
9. Conclusion.....	50
Annexes.....	51
Accueil Version Desktop.....	51
Accueil Version Mobile.....	52
Contenu Version Desktop.....	53
Contenu Version Mobile.....	54
Détails Cartes Contenu.....	55
S'inscrire.....	56
Se Connecter.....	56
Mon Compte.....	57
Lecteur Version Desktop.....	58
Lecteur Version Mobile.....	59
Modale de succès et d'erreurs.....	60
Script SQL.....	60

1. Compétences couvertes par ce projet

1.1 Liste des compétences mises en œuvre dans ce projet.

- Concevoir et développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité
 - Maquetter une application
 - Réaliser une interface utilisateur web statique et adaptable
 - Développer des composants d'accès aux données
 - Réaliser une interface utilisateur avec une solution de gestion de contenu
- Concevoir et développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité
 - Créer une base de données
 - Développer les composants d'accès aux données
 - Développer la partie back-end d'une application web ou web mobile
 - Élaborer et mettre en œuvre des composants dans une application de gestion de contenu

2. Présentation du projet

2.1 Précisions sur les technologies utilisées

N'ayant pas la possibilité de présenter quelque chose de substantiel, et surtout de personnel, à l'issue de mon stage en entreprise, j'ai donc pris l'initiative de réaliser un site dont je suis l'unique créateur.

En effet, il me semble beaucoup plus formateur et bien plus représentatif des compétences que je maîtrise de présenter quelque chose que j'ai construit de A à Z. De plus, je me suis imposé comme contrainte d'utiliser le plus possible les langages de base en ayant recours à des librairies uniquement lorsque cela devenait vraiment trop compliqué, ou fastidieux, à coder «à la main». Je ne me suis pas imposé cela dans le but de relever un défi mais simplement parce que je pense sincèrement :

- qu'en six mois de formation on ne maîtrise pas encore les langages de bases
- que faire appel trop facilement à «l'artillerie lourde» (que sont un peu les framework ou les librairies), au détriment des langages de bases, cela ne peut que générer des lacunes qui seront difficiles à combler par la suite
- et qu'enfin, une utilisation trop précoce de ces technologies (framework etc.) ne nous aide pas forcément à en comprendre la véritable utilité car l'on aura pas été confronté à certaines difficultés que l'on aurait rencontrées en essayant de faire les choses par soi-même.

Les technologies que j'ai utilisé pour réaliser ce projet sont donc :

- pour la partie back-end
 - PHP et SQL
 - Composer pour pouvoir utiliser la librairie PHP-FFMpeg
- pour la partie front-end
 - HTML, CSS(Sass), JavaScript
 - Node.js pour pouvoir utiliser Sass
 - les librairies Gsap et Barba.js pour les animations
 - Bootstrap pour la partie admin du site uniquement

Il va de soi, que je ne prétend pas avoir acquis une maîtrise approfondi des langages de bases au moment de la rédaction de ce rapport car, ce n'est que mon avis, une réelle aisance dans le développement web nécessite sans doute des années de pratique. Je ne suis donc pas entrain de dire que j'exclue l'utilisation de technologies plus moderne, que sont les framework etc., tant que je n'aurai pas fait le tour des langages de base. Simplement, il me paraît plus judicieux à ce stade de mon apprentissage de présenter quelque chose dont je comprend chaque éléments. Ma compréhension de l'ensemble, en utilisant des frameworks etc. (avec un résultat final certainement tout aussi fonctionnel), aurait probablement été plus vague et donc, ce projet, moins formateur à mes yeux.

2.2 Le projet

Le projet que je présente dans ce rapport est issu de ma passion pour le piano, de la rigueur nécessaire pour apprendre à en jouer et de la difficulté à trouver un site gratuit proposant des vidéos ayant pour thème principal la pédagogie musicale. Il existe bien-sûr des sites comme YouTube qui proposent des millions de vidéos sur ce thème mais la qualité n'est pas forcément au rendez-vous.

J'ai donc essayé de mettre en œuvre les compétences que j'ai acquises lors de ma formation pour créer un site qui a pour nom «Beyond The Abyss». Ce nom est une allusion au fait que souvent, lorsque l'on s'attaque à l'apprentissage d'un instrument de musique (ou de tout autre domaine), un abîme semble parfois apparaître entre notre niveau actuel et celui qu'il nous faudrait pour arriver à faire certaines choses. Ce site a donc la prétention, en toute modestie, de fournir un contenu gratuit permettant d'aller au-delà de cet abîme grâce à la mise en place d'un système d'échange collaboratif basé sur une monnaie virtuelle gérée par différentes règles sur lesquelles je reviendrai en détails plus loin dans ce rapport(p.16).

2.3 Remarques

Ce projet, bien que le résultat me satisfasse, n'a pas pour vocation à se retrouver en ligne, du moins au moment de la rédaction de ce rapport. En l'état, il n'est pour moi qu'un outil pédagogique visant à valider mes compétences au terme de la formation que j'ai suivi chez PopSchool. Les choses ne sont pas figées, peut-être qu'à l'avenir cela évoluera mais pour l'instant ce n'est pas d'actualité.

Une des raisons qui explique cela est aussi une des particularités de ce site qui est entièrement constitué de nuances de gris. Étant daltonien et le design (qui est un métier à part entière) de ce site mon œuvre, j'ai évité d'utiliser des couleurs car le résultat n'aurait sans doute pas été très heureux...

3. Travail Préliminaire

3.1 Utilisateur : expression du besoin et formulation du cahier des charges

3.1.1 Accueil

- Besoin

→ Un utilisateur web doit tout d'abord être accueilli de manière originale, comprendre quel est le thème d'un site facilement tout en suscitant son intérêt.

→ Une page d'accueil doit aussi disposer d'un menu permettant d'avoir un aperçu rapide des fonctionnalités, d'une présentation de son fonctionnement, une présentation de son auteur, d'une fonction de recherche et des mentions légales.

- Cahier des charges

→ Le titre du site aura une formulation sibylline «Beyond The Abyss» pour susciter l'intérêt et un sous titre qui dévoile le thème du site «Learning Music Videos». Trois liens ayant des noms évocateurs pour tout musicien «Tutorials», «Performances» et «Sheet Music» seront directement disponible à l'ouverture du site. En scrollant vers le bas trois cartes présenteront de manière aléatoire un contenu pour chacune de ces trois catégories (tutoriels, performances, partitions) avec sa description et un lien permettant d'en savoir plus.

→ Une barre de navigation qui disparaîtra en scrollant vers le bas (et qui réapparaît en scrollant vers le haut) pour permettre une meilleure visibilité, en particulier sur mobile contiendra un bouton qui fera apparaître un menu plein écran et une fonction de recherche s'ouvrant dans une modale.

→ Un footer contiendra le nom de l'auteur, des liens «About», «Privacy Policy» et «Terms» et des liens vers les principaux réseaux sociaux.

→ Le menu plein écran contiendra les liens vers les différentes sections du site «Login, Register, My Account etc..» ainsi que des liens vers les principaux réseaux sociaux et un lien pour contacter un administrateur s'ouvrant dans une modale par dessus le menu

→ Pour ce qui est de l'originalité visuelle, le chargement de la page d'accueil ainsi que l'ouverture et la fermeture du menu animeront l'ensemble des éléments de la page.

WireFrames Accueil:

Annexe Accueil Version Desktop (lien vers p.51)

Annexe Accueil Version Mobile (lien vers p.52)

3.1.2 Contenu

- Besoin

→ Un utilisateur arrivant sur une page présentant les contenus d'un site doit pouvoir rapidement se faire idée de ce que propose le site et pouvoir faire un choix en quelque coup d'œil.

→ Cette page doit aussi présenter les mêmes fonctionnalités au niveau de la barre de navigation et du footer et du menu. (voir 3.1.1.1 p.7)

- Cahier des charges

→ Les contenus proposés apparaîtront sous forme de carte ayant toute la même structure. Un titre, le nom du compositeur, sa catégorie (tutoriel, performance, partition), son niveau (facile, moyen, difficile, très difficile), le nombre de likes et une mention indiquant:

- «le prix» du contenu si l'utilisateur n'a pas acheté le contenu ou que celui-ci n'est pas connecté
- «gratuit» et si c'est le cas que le contenu soit accessible sans nécessité d'être connecté
- «votre contenu» si l'utilisateur est l'auteur du contenu
- «acheté» si l'utilisateur a acheté le contenu

→ Chaque carte devra lorsqu'elle est survolée (ou que l'on clique dessus quand on est sur mobile) faire apparaître deux boutons permettant de voir pour l'un tous les contenus de l'auteur du contenu concerné et pour l'autre de voir des informations sur le contenu (sa description, son prix et le solde de crédits de l'utilisateur) dans une modale contenant elle-même deux boutons l'un permettant d'acheter le contenu et l'autre permettant d'ouvrir une nouvelle modale contenant les commentaires liés à ce contenu et le nombre de likes de chaque commentaire.

WireFrames Contenu:

Annexe Contenu Version Desktop (lien vers p.53)

Annexe Contenu Version Mobile (lien vers p.54)

Annexe Détails des cartes de la page Contenu (lien vers p.55)

3.1.3 S'inscrire

- Besoin

→ Si l'utilisateur ne possède pas de compte, il faut qu'il puisse s'en créer un.

→ Cette page doit aussi présenter les mêmes fonctionnalités au niveau de la barre de navigation et du footer et du menu. (voir 3.1.1.1 p.7)

- Cahier des charges

→ Pour se créer un compte l'utilisateur devra renseigner une adresse son nom, son prénom, un email et un mot de passe qu'il devra confirmer par soucis de sécurité.

WireFrame S'inscrire:

Annexe S'inscrire Version Desktop et Mobile (lien vers p.56)

3.1.4 Se connecter

- Besoin

→ L'utilisateur doit pouvoir se connecter

→ Cette page doit aussi présenter les mêmes fonctionnalités au niveau de la barre de navigation et du footer et du menu. (voir 3.1.1.1 p.7)

- Cahier des charges

→ Pour se connecter l'utilisateur devra saisir l'email et le mot de passe qu'il a renseigné lors de la création de son compte.

WireFrame Se connecter:

Annexe Se connecter Version Desktop et Mobile (lien vers p.56)

3.1.5 Mon compte

- Besoin

→ L'utilisateur doit pouvoir avoir accès à toutes ses informations et pouvoir interagir avec elles avec des fonctionnalités appropriées, ajouter du contenu et prendre connaissance des diverses notifications qui pourront lui être adressées.

→ Cette page doit aussi présenter les mêmes fonctionnalités au niveau de la barre de navigation et du footer et du menu. (voir 3.1.1.1 p.7)

- Cahier des charges

→ Des formulaires dépliant permettront d'éditer son nom, son prénom, de changer son mot de passe et d'ajouter du contenu. La seule chose qui ne pourra être modifiée dans un souci de sécurité est l'adresse mail, il faudra faire appel à un administrateur pour la changer.

→ Trois boutons permettront soit de visualiser l'ensemble des contenus que l'on a postés, soit de visualiser l'ensemble des contenus que l'on a achetés et le dernier permettra la suppression de son compte avec au préalable l'ouverture d'une modale demandant la confirmation de la suppression (qui sera définitive et entraînera la suppression de tous les contenus de l'utilisateur et des éléments qui y sont associés comme les likes et les commentaires ainsi que toutes traces de l'utilisateur sur le site comme ses commentaires et ses likes).

→ Les notifications s'afficheront en dessous des formulaires et un bouton permettra de les effacer indépendamment l'une de l'autre avec au préalable l'apparition d'une modale demandant la confirmation de la suppression définitive

WireFrame Mon compte:

Annexe Mon Compte Version Desktop et Mobile (lien vers p.57)

3.1.6 Lecteur

- Besoin

→ L'utilisateur doit pouvoir lire la vidéo en mode plein écran ou non, avoir toutes les informations sur le contenu ainsi que les commentaires associés et pouvoir interagir avec tous ces éléments avec des fonctionnalités appropriées.

→ Cette page doit aussi présenter les mêmes fonctionnalités au niveau de la barre de navigation et du footer et du menu. (voir 3.1.1.1 p.7)

- Cahier des charges

→ La vidéo lorsqu'elle ne sera pas en mode plein écran fera apparaître son titre, son compositeur, sa catégorie et son niveau.

→ En dessous de la vidéo, une mention indiquant le nombre de likes, un bouton avec le nom de l'auteur de la vidéo permettant d'accéder à tous les contenus de cet utilisateur, la description du contenu et deux autres boutons ayant des fonctionnalités différentes suivant que l'on est l'auteur du contenu ou non:

- si l'utilisateur connecté est l'auteur du contenu l'un des ces boutons aura pour fonction de pouvoir commenter le contenu et l'autre permettra par l'apparition d'un menu dépliant d'éditer chaque élément du contenu par l'ouverture d'une modale ou de l'effacer avec au préalable l'ouverture d'une modale demandant la confirmation de la suppression(qui sera définitive et entraînera également la disparition des tous les éléments qui y sont associés comme les likes et les commentaires).

- si l'utilisateur connecté n'est pas l'auteur du contenu l'un des ces boutons permettra de liker le contenu et l'autre par l'apparition d'un menu dépliant de commenter ou de signaler le contenu

→ Encore en dessous seront affichés les commentaires, avec leur nombre de likes, le nombre de commentaires postés sur la totalité du site par l'auteur du commentaire et un bouton permettant suivant que l'on est l'auteur de commentaire ou non soit de liker soit de l'éditer le commentaire par l'ouverture d'une modale.

WireFrame Lecteur:

Annexe Lecteur Version Desktop (lien vers p.58)

Annexe Lecteur Version Mobile (lien vers p.59)

3.1.7 Modale Erreurs et Succès

- Besoin

→ L'utilisateur doit être courant de la réussite ou de l'échec de ses actions.

- Cahier des charges

→ Une modale contenant un message de succès ou d'erreur apparaîtra en fonction des actions de l'utilisateur et disparaîtra d'un clic n'importe où sur l'écran.

WireFrame modale de succès ou d'erreurs:

Annexe modale de succès ou d'erreurs (lien vers p.60)

3.2 Administrateur : expression du besoin et formulation du cahier des charges

- Besoin

→ L'administrateur doit avoir accès à l'ensemble des différents éléments de contenu du site de manière claire, ergonomique.

→ Il doit pouvoir modifier ou supprimer tous ces éléments

→ Il doit pouvoir répondre aux sollicitations des utilisateurs

- Cahier des charges

→ La partie administration dans sa partie graphique sera faite avec bootstrap afin d'être claire et tout a fait classique

→ Un menu sera disponible avec toutes les catégories qui auront chacune leur propre page

→ Chaque page affichera les éléments qui lui sont spécifiques sous forme d'un tableau regroupant le minimum d'informations nécessaires à l'identification d'un élément pour ne pas surcharger la page avec des informations inutiles

→ Des boutons permettront de voir tous les détails d'un élément tel qu'il est structuré dans la base de données, de l'éditer si cela a une utilité ou de le supprimer si cela s'avère nécessaire

3.3 Environnement humain et technique

3.3.1 Environnement humain

J'ai effectué tout mon projet depuis chez moi. Ce projet n'a pas été effectué dans le cadre d'un stage, et je ne dépendais d'aucune entreprise. Pendant mon projet, je me suis principalement aidé d'internet pour trouver des solutions aux éventuels problèmes que j'ai rencontré.

3.3.2 Planification et gestion du temps

Étant données les conditions dans lesquelles j'ai travaillé, j'ai dû m'organiser seul, et cela s'est ressenti directement sur ma planification. J'ai tenté de séparer aussi bien que possible les tâches suivant leur urgence et leur côté facultatif ou l'obligatoire. J'ai ensuite organisé mon travail en fonction des fonctionnalités du projet avec un rythme plus ou moins variables suivant les difficultés rencontrées

L'avancement du projet était principalement marqué par mes merges Git sur la branche principale, et c'est en me basant sur ceux-ci que je mesurais mon avancement .

3.3.3 Environnement technique

Plateformes

J'ai développé le projet sur Windows (10). J'ai donc eu l'occasion d'expérimenter avec les différents outils sous Windows.

Git

Afin de développer sereinement mon application, j'ai choisi d'utiliser Git, l'un des outil de versionning les plus utilisés dans le monde du développement web. Ce dernier présente de nombreux avantages qui m'ont été utiles tout au long de mon projet :

- la possibilité de revenir en arrière dans le cas où les dernières modifications ne sont pas satisfaisante
- la possibilité de développer plusieurs couches différentes de mon application en parallèle (ne pas être obligé de finir chaque fonctionnalité avant de passer à la suivante), à l'aide des branches.
- En parallèle de Git pour avoir une sauvegarde en ligne j'ai utilisé la plateforme GitHub.

Choix de l'IDE

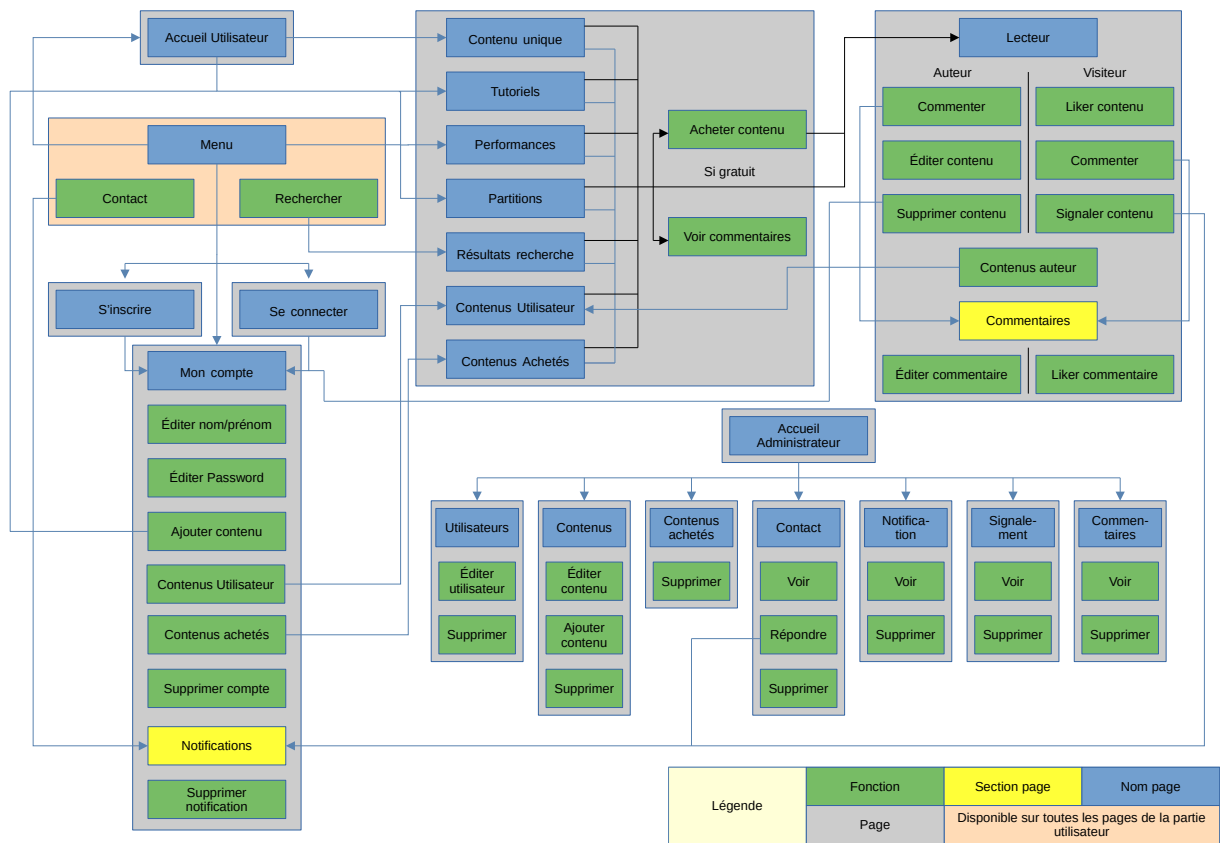
J'ai choisi l'IDE VSCode, pour les deux raisons suivantes : d'une part c'est un IDE qui est recommandé par de nombreux développeurs et, d'autre part la licence est gratuite. N'ayant pas expérimenté d'autres IDE je ne peux pas faire de comparatif mais je suis pour l'instant entièrement satisfait des fonctionnalités de bases, des nombreuses extensions disponibles et de sa stabilité sans faille.

4 Spécifications

4.1 Spécifications fonctionnelles

4.1.1 Arborescence du site

En fonction du cahier des charges et des WireFrames que j'ai présenté plus haut j'ai pu établir une arborescence du squelette du site.



Arborescence du site

4.1.2 Liste des fonctionnalités spécifiques:

Je n'apporte ici que des précisions sur les fonctionnalités qui me semble nécessaire d'être explicitées plus en détails. En effet un certain nombre de fonctionnalités sont aisément compréhensibles par leur simple appellation dans l'arborescence du site(p.15) ou les Wireframes(p.7).

- Le site est accessible gratuitement sans nécessité d'être connecté car du contenu gratuit peut être disponible.
- Une fonction de recherche avec suggestion permet de rechercher du contenu en fonction du titre, du compositeur, de la catégorie, du niveau, de sa gratuité. Toutes ces entrées sont indépendantes l'une de l'autre. La validation de la recherche sans avoir au préalable saisi au moins une entrée affichera tous les contenus du site.
- Lorsqu'un utilisateur crée un compte il dispose automatiquement de 500 Crédits.
- Lorsqu'un utilisateur ajoute du contenu il gagne 100 Crédits pour une *Performance*, 200 Crédits pour une *Partition* et 300 Crédits pour un *Tutoriel*.

Rem.: Cette classification vient de la quantité de travail à fournir pour créer chaque type de contenu. En effet, une *Performance* consiste «simplement» à jouer un morceau, une *Partition* consiste à synchroniser une partition avec sa *Performance* (donc d'avoir possiblement retranscrit le morceau si la partition n'existait pas) et un *Tutoriel* regroupe la *Performance*, la *Partition* avec en plus une approche pédagogique.

- Lors de l'ajout d'un contenu l'utilisateur fixe un prix entre 1 et 500 Crédits ou gratuit.
- Le créateur du contenu peut à tout moment changer le prix de son contenu ou sa catégorie si il s'est trompé. Le changement de catégorie entraîne une modification du nombre de crédits que celui-ci avait reçu lors de la création ou de la précédente modification (un gain pour une catégorie «supérieure» et inversement).
- Un créateur de contenu en cas de suppression d'un contenu que ce soit par lui-même ou par un administrateur voit son solde de Crédits déduit du gain au moment de la création ou de la dernière modification de catégorie. De plus, la suppression d'un contenu entraîne la suppression de tout ce qu'il lui est associé dans la base de données (commentaires, likes etc.).
- Un utilisateur qui supprime son compte entraîne la suppression de tout ce qu'il lui est associé dans la base de données.
- Un utilisateur qui a acheté un contenu dont le prix change voit son solde de crédits:
 - augmenter de la différence en cas baisse du prix.
 - inchangé si le prix augmente par rapport au prix au moment de l'achat
 - diminuer si le prix augmente de nouveau suite à une baisse dans la limite du prix au moment de l'achat.

- remboursé en totalité si le contenu vient à être supprimé ou si le créateur du contenu supprime son compte.

L'utilisateur ne peut donc jamais payer plus que son investissement au moment de l'achat mais peut être remboursé jusqu'à la totalité de son investissement si le contenu devient gratuit ou si celui-ci est supprimé.

- Un utilisateur qui achète un contenu rapporte au créateur du contenu 1/25ème (arrondi à l'entier supérieur) du prix au moment de l'achat. Donc de 1 Crédits (pour un prix d'achat de 1 Crédits) à 20 Crédits (pour un prix d'achat de 500 Crédits). *(Rem.: ces proportions n'ont rien de définitives. Elles sont essayées modifiable dans le code si elles doivent être équilibrées. Pour un site avec un nombre d'utilisateur limité comme celui-ci cela me paraît raisonnable).*

- Un utilisateur qui like un contenu rapporte au créateur du contenu 1 Crédits.

- Le signalement d'un contenu entraîne l'impossibilité pour le créateur du contenu d'éditer ou de supprimer celui-ci tant qu'un administrateur n'a pas examiné celui-ci et décidé des mesures à prendre (rappel à l'ordre pour le créateur du contenu ou de l'utilisateur signalant abusivement, modification du contenu, suppression du contenu, suppression de l'utilisateur).

4.1.3 Remarque

Ce système qui peut paraître un peu alambiqué de prime abord n'a pour but que d'inciter les utilisateurs du site à poster du contenu de la meilleure qualité possible et ce pour plusieurs raisons:

Ce système n'a pas la prétention d'être innovant ou parfait. J'ai simplement essayé d'imaginer un système qui me paraît cohérent dans le but que je me suis fixé. Et c'est aussi à vrai dire une occasion de coder des fonctions un peu moins basiques.

- Le meilleur moyen de gagner des crédits et de pouvoir acheter d'autres contenus est de poster du contenu. Le fait qu'un contenu puisse être signalé limitera les utilisateurs postant des contenus dans le seul but de gagner des crédits et de profiter du contenu des autres sans contribuer sérieusement (je rappelle que c'est un site gratuit basé sur un échange collaboratif).

- Les deux autres moyens de gagner des crédits sont d'obtenir des likes ou de vendre du contenu ce qui entraîne plusieurs choses:

- les chances de vendre un contenu augmentent avec la quantité de likes qui lui est associée
- pour avoir des likes il faut que d'autres utilisateurs aient vu et apprécié un contenu
- il n'est donc pas très judicieux de fixer lorsque l'on est un nouvel utilisateur des prix très élevés. Cela incite donc à poster dans un premier temps des contenus peu chers, voir gratuits (ce qui permet d'attirer de nouveaux membres potentiels avec des contenus que l'on peut voir sans s'inscrire).

- Le dernier point est que l'utilisateur peut (avec la possibilité que le créateur a de modifier le prix d'un contenu à sa guise) se retrouver avec un solde de crédits négatifs si il dépense les crédits qui lui ont été remboursés suite à une baisse de prix et que le créateur décide d'augmenter le prix à nouveau.

Le fait que le solde de crédits soit sujet à des fluctuations que l'on ne contrôle pas totalement (du fait qu'il existe une sorte de réserve «tampon» de crédits sujette aux variations de prix) incitera donc les utilisateurs si ils veulent pouvoir continuer à acheter des contenus qui les intéressent à poster de nouveaux contenus, car cela reste le meilleur moyen d'avoir, de manière certaine, un solde de crédits dans le positif.

4.2 Spécifications techniques

4.2.1 Technologies utilisées pour le Front-end

- HTML 5

L'HTML (HyperText Markup Language) est un langage de balisage et permet principalement de structurer le contenu d'une page web mais aussi d'y inclure des éléments tel que des images ou des formulaires.

- CSS3

CSS (Cascading Style Sheets) est un langage permet la mise en forme d'une page web. Il est utilisé pour changer les couleurs, les bordures, les polices mais également pour gérer l'adaptabilité aux différents supports, en outre c'est avec ça que l'on fait le Responsive Design.

- Node.js (pour utiliser Npm)

Node.js est un langage de programmation en Javascript orientée vers les applications réseau qui doivent pouvoir monter en charge. Parmi les modules natifs de Node.js on retrouve HTTP qui permet de développer des serveurs HTTP, il est donc possible de se passer de serveur comme Apache au moment du déploiement de son application. En outre Node.js est un langage de bas niveau qui permet l'exécution de Javascript côté serveur.

- Npm (pour utiliser Sass)

Npm est le gestionnaire de paquet officiel de Node.js. Il permet la gestion des dépendances d'une application ainsi que l'installation de paquets disponibles sur son dépôt. • Navigateur Accessible principalement depuis Chrome.

- Sass

Sass (Syntcatically Style Sheet) est un préprocesseur qui ajoute des fonctionnalités à CSS. Il permet entre autres de mieux structurer et simplifier le code et d'éviter les répétitions. Le code Sass doit être dans des fichiers ayant l'extension .scss. Les fichiers sont ensuite compilés et convertis en CSS.

- JavaScript

JavaScript est un langage de script principalement utilisé pour rendre des pages web interactives.

- GSAP

GSAP est une librairie JavaScript qui permet d'animer les éléments d'une page HTML.

- Barba.js

Barba.js est une petite bibliothèque flexible et sans dépendance qui vous aide à créer des transitions fluides et fluides entre les pages de votre site Web.

- Bootstrap

Bootstrap est un framework contenant une collection d'outil destiné à faciliter la structuration de nos page web. Il permet l'intégration d'éléments comme des boutons, des listes, des formulaires stylisés de base. Il facilite également le Responsive.

4.2.2 Technologies utilisées pour le Back-end

- PHP

PHP (Hypertext Preprocessor) plus connu sous son sigle PHP est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP. PHP est un langage impératif orienté objet. Il s'agit d'un langage de script interprété côté serveur. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.

- Composer

Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin.

- PHP -FFMPEG

Une bibliothèque orientée objet pour convertir des fichiers vidéo/audio avec Ffmpeg.

- SQL

SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données.

- MySQL

MySQL est un serveur de bases de données relationnelles développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées.

- Apache

Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde.

4.3 Sécurité

Je reviendrai sur toutes les mesures mises en œuvre pour la sécurité plus loin dans ce rapport dans la partie (situation de travail ayant nécessité une recherche). Voici néanmoins quelques éléments essentiels que nous pouvons citer dans cette partie :

→ Toutes les parties du site où des actions peuvent influencer sur le contenu du site sont protégées par l'authentification.

→ Tous les entrées utilisateurs sont vérifiées côté front et côté back et, quand elles sont affichées, les variables sont échappées avec `htmlspecialchars`.

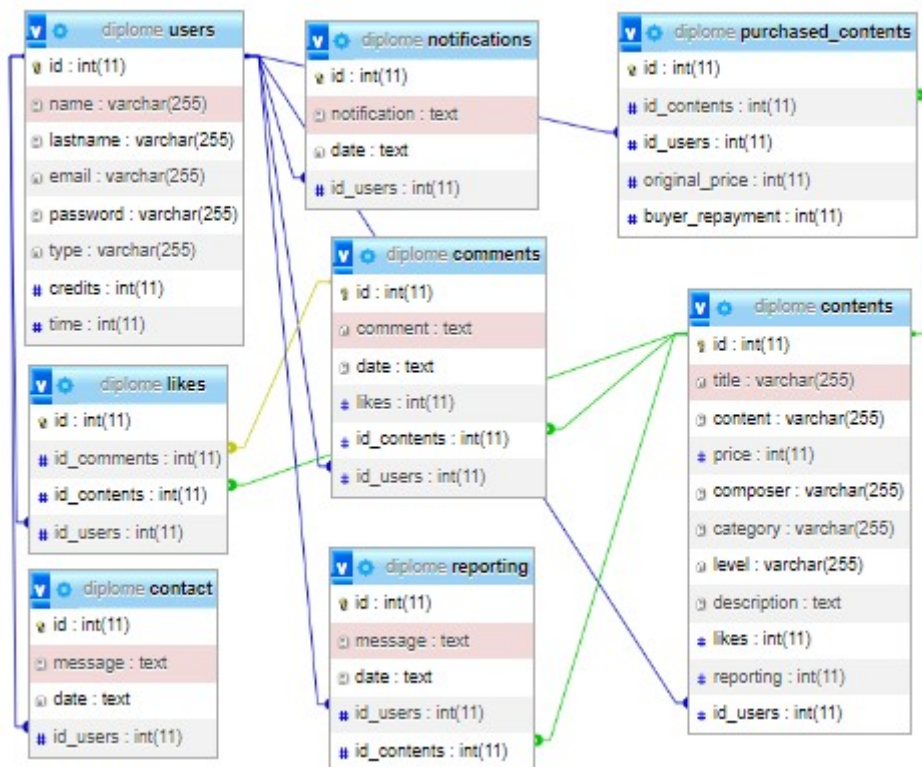
5. Réalisation du projet

5.1 Mise en place du projet

5.1.1 Création préliminaires

- La base de données

J'ai créé la base de donnée «diplome» de mon projet à partir de l'interface graphique en ligne phpMyAdmin. Je m'y suis connecté comme l'utilisateur root (administrateur), puis j'ai créé la base avec l'encodage utf8_general_ci, dont voici la représentation:



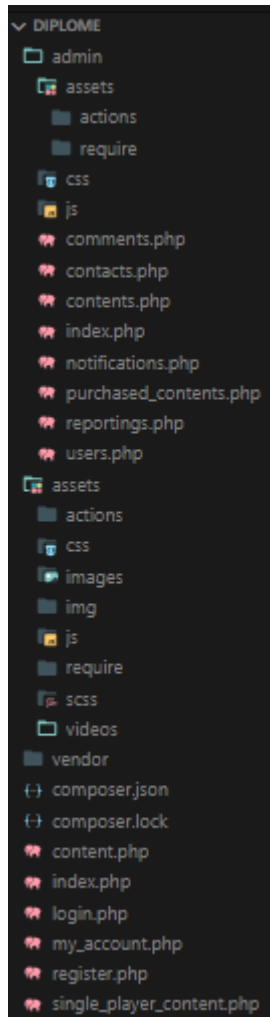
Représentation de la base de données

Script SQL en annexe (lien vers p.60)

- Le projet GitHub

J'ai créé sur GitHub un projet qui me servira de sauvegarde pendant tout l'avancement du projet. Cela permet d'être sûr de pouvoir récupérer le projet en cas de problème et d'avoir un suivi de l'avancement.

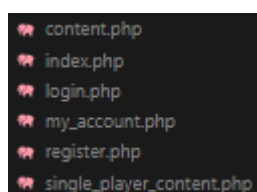
5.1.2 Création du projet



L'arborescence du projet dont qui ne sera détaillée que pour la partie utilisateur car la partie administrateur est faite selon le même schéma.

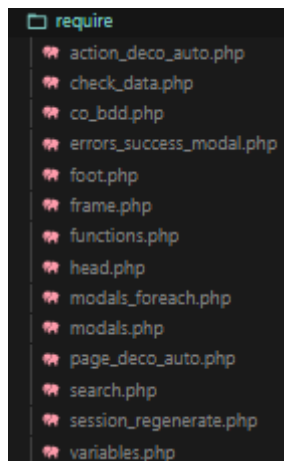
Je tiens à préciser que cette structure n'est que le fruit de mes connaissances au moment de la création de ce projet et que je ne considère en rien que cela constitue une façon optimale de faire. J'ai essayé de compartimenter les choses au maximum pour clarifier la lecture du code mais ne prétend pas respecter l'architecture MVC qui est sans doute une des manières optimale de procéder mais qui reste pour l'instant quelque chose d'assez vague à mes yeux dans son application, bien que je comprenne cette structure dans son aspect conceptuel.

- La racine du projet



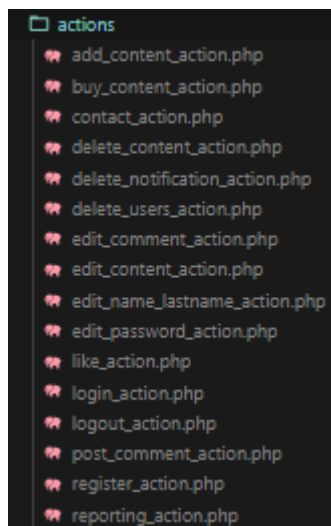
Les différentes pages composant la partie visible du projet.

- Require



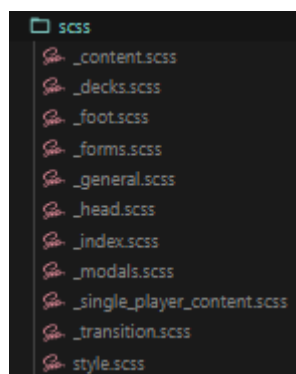
Les différents fichiers qui sont appelés dans d'autres fichiers dans le but de spécialiser et de compartimenter au maximum les différents éléments du projet.

- Actions



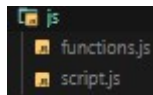
Les différents scripts que l'on rencontre sur le site et qui sont exécutées lors de la soumission d'un formulaire etc..

- Scss



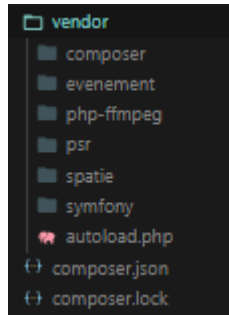
Les différents fichiers .scss qui sont compilés et convertis en CSS.

- Js



Le fichier script.js qui contient tout ce qui concerne l'animation du site et le fichier functions.js regroupant diverses fonctions.

- Vendor



Ensemble des dossiers générés par l'installation de Composer et Php-ffmpeg

5.2 Présentation détaillée de la page «Single_player_content»

Je vais maintenant dans les détails la manière dont j'ai structuré une page. Toutes les pages suivant la même logique je ne vous en présenterai qu'une. Je ne détaillerai cependant pas tous les éléments (seulement ceux qui me paraissent intéressants) pour vous épargner les répétitions.

5.2.1 Vérifications préliminaires

```

1  <?php
2  session_start();
3  require('./assets/require/check_data.php');
4
5  if (
6      isset($get_id)
7      && (isset($session_users_id) xor !isset($check_session_users_id))
8      && (isset($get_error) xor !isset($check_get_error))
9      && (isset($get_success) xor !isset($check_get_success))
10 ) {
11
12
13     $page = 'single_player';
14     require('./assets/require/co_bdd.php');
15     require('./assets/require/functions.php');
16
17     $getContentAndUserInformations = getContentAndUserInformations($bdd, $get_id);

```

Ligne 2: il peut être nécessaire d'être connecté pour aller sur cette page mais ce n'est pas obligatoire

Ligne 3: appel du fichier `check_data.php` qui sert à vérifier si les données nécessaires pour aller sur une page ont le format attendu :

Par exemple `isset($get_id)` de la ligne 6:

```
101 if (isset($_GET['id']) && !empty($_GET['id'])) {
102
103     $check_get_id = is_numeric($_GET['id'])
104         && preg_match("/^[0-9]+$/", $_GET['id']);
105
106     if ($check_get_id === true) {
107
108         $get_id = $_GET['id'];
109     }
110 }
```

Les détails sur cette manière de faire seront développés dans le chapitre sur la sécurité p.46.

Ligne 7: Ici avec la même manière de procéder que pour `$get_id` vérification qu'un utilisateur est bien connecté et que la variable a bien le format attendu ou qu'elle n'existe pas du tout. *Remarque: pour la cohérence du code toutes ces variables sont nommées avec la même logique. Ici `$session_user_id` correspond à `$_SESSION['user']['id']`. Cela permet également d'améliorer la lisibilité du code.*

Ligne 10: Si les données n'ont pas le format attendu une redirection vers la page d'accueil est effectuée.

```
237 <?php } else {
238
239     $bdd = null;
240     header('location: index.php?error=00615');
241     die();
242 } ?>
```

Une éventuelle connexion à la base de donnée est close, redirection avec une erreur «Une erreur est survenue» (toutes les erreurs ont un code unique qui affiche une modale avec le message correspondant) et le script courant est terminé.

Ligne 13: La variable `$page` sert à modifier les liens du menu en fonction de la page sur laquelle on se trouve.

Ligne 14: Appel du fichier de connexion à la base de données.

Ligne 15: Appel du fichier `function.php` contenant toutes les fonctions utilisées sur les pages:

Par exemple `getContentAndUserInformations` de la ligne 17 qui sert ici à récupérer les informations liées au contenu.

```
145 function getContentAndUserInformations(PDO $bdd, $get_id)
146 {
147     $req = $bdd->prepare('SELECT users.name, users.lastname, contents.id, contents.title,
148         contents.composer, contents.category, contents.level, contents.content, contents.price,
149         contents.description, contents.likes, contents.reporting, contents.id_users
150     FROM users
151     INNER JOIN contents
152     ON users.id = contents.id_users WHERE contents.id = :contents_id ');
153     $req->bindParam(':contents_id', $get_id, PDO::PARAM_INT);
154     $req->execute();
155     $content = $req->fetch();
156     return $content;
157 }
```

5.2.2 Suite des vérifications

```
20 if ($getContentAndUserInformations) {
21
22     require('./assets/require/variables.php');
23
24     $getComments = getComments($bdd, $get_id);
25
26
27     if (isset($session_users_id) && !empty($session_users_id)) {
28
29         $getUserInformations = getUserInformations($bdd, $session_users_id);
30
31         require('./assets/require/variables.php');
32
33         $getIdUserFromPurchasedContent = getIdUserFromPurchasedContent($bdd,
34             $getContentAndUserInformations_id);
35
36         $user_session_purchased_content = in_array($getUserInformations_id, array_column
37             ($getIdUserFromPurchasedContent, 'id_users'));
38     }
39
40     if (($getContentAndUserInformations_price > 0 && isset($session_users_id)) ||
41         $getContentAndUserInformations_price == 0) {
42
43         if ((isset($user_session_purchased_content) && $user_session_purchased_content ==
44             true) || $getContentAndUserInformations_price == 0 ||
45             $getContentAndUserInformations_id_user == $getUserInformations_id) {
46
47             require('./assets/require/head.php'); ?>
```

Ligne 20: Vérification de l'existence du contenu sinon redirection vers la page d'accueil avec une erreur «Ce contenu n'existe pas».

Ligne 22: Appel du fichier **variables.php** qui regroupe toutes les variables appelées (voir p.46).

Par exemple **getContentAndUserInformations** de la ligne 20:

```
94 if (isset($getContentAndUserInformations) && !empty($getContentAndUserInformations)) {
95
96     $getContentAndUserInformations_id = htmlspecialchars($getContentAndUserInformations['id']);
97     $getContentAndUserInformations_title = htmlspecialchars($getContentAndUserInformations['title']);
98     $getContentAndUserInformations_composer = htmlspecialchars($getContentAndUserInformations['composer']);
99     $getContentAndUserInformations_category = htmlspecialchars($getContentAndUserInformations['category']);
100     $getContentAndUserInformations_level = htmlspecialchars($getContentAndUserInformations['level']);
101     $getContentAndUserInformations_video = htmlspecialchars($getContentAndUserInformations['content']);
102     $getContentAndUserInformations_price = htmlspecialchars($getContentAndUserInformations['price']);
103     $getContentAndUserInformations_description = str_replace('<br />', ' ', nl2br(htmlspecialchars
104         ($getContentAndUserInformations['description'])));
105     $getContentAndUserInformations_likes = htmlspecialchars($getContentAndUserInformations['likes']);
106     $getContentAndUserInformations_id_user = htmlspecialchars($getContentAndUserInformations['id_users']);
107     $getContentAndUserInformations_author_name = htmlspecialchars($getContentAndUserInformations['name']);
108     $getContentAndUserInformations_author_lastname = htmlspecialchars($getContentAndUserInformations
109         ['lastname']);
110 }
```

Dans ce fichier les différents traitements nécessaires pour afficher des informations dans une page sont appliqués. *Remarque: les variables sont toutes nommées avec la même logique. Ici on sait par exemple que `$getContentAndUserInformations_id` est issu de la variable `$getContentAndUserInformations` qui est elle-même générée par la fonction `getContentAndUserInformations()`. Cela permet donc une nouvelle fois d'alléger le code et de savoir dès que l'on rencontre une variable dans une page de savoir d'où elle vient.*

Ligne 33: Récupération de l'**id** de tous les utilisateurs ayant acheté ce contenu.

Ligne 35: L'**id** de l'utilisateur connecté fait-elle partie des **id** récupéré à la ligne 33.

Ligne 39: Si le contenu n'est pas gratuit il faut être connecté sinon redirection vers page d'accueil avec une erreur «Ce contenu n'est pas gratuit. Connectez-vous ou inscrivez-vous. Vous obtiendrez 500 Crédits».

Ligne 41: Si l'utilisateur connecté a acheté le contenu ou si il est gratuit ou si l'utilisateur connecté est l'auteur du contenu, visionnage autorisé sinon redirection vers page d'accueil avec une erreur «Vous n'avez pas acheté ce contenu».

Ligne 43: Début du chargement de la page avec l'appel de **head.php** qui contient hormis le **!doctype** etc. l'appel de deux fichiers nécessaire à la sécurité du site **page_deco_auto.php** qui déconnecte l'utilisateur automatiquement au bout de 15 min en cas d'inactivité et **session_regenerate.php** qui régénère l'identifiant de session toutes les 5 min. Les détails concernant ces fichiers seront développés dans le chapitre sur la sécurité.

5.2.3 Contenu visible de la page

```
46 <main class="autoAlpha" data-barba="wrapper">
47   <div class="min-height" data-barba="container"
48     data-barba-namespace="single_player_content-section">
49
50     <div class="movie-card">
51       <div class="single_player_container">
52
53         <div class="hero">
54
55           <video src="./assets/videos/<?= $getContentAndUserInformations_video ?>"
56             controls controlsList="nodownload"></video>
57
58           <div class="details">
59             <h1 class="title1"><?= $getContentAndUserInformations_title ?></h1>
60             <h2 class="title2"><?= $getContentAndUserInformations_composer ?></h2>
61             <h2 class="title3"><?= $getContentAndUserInformations_category ?> <?=
62               $getContentAndUserInformations_level ?></h2>
63           </div>
64         </div>
65
66         <div class="player_bottom_bar">
67           <span class="likes"><i class="far fa-thumbs-up"> <?=
68             $getContentAndUserInformations_likes ?></i></span>
```

Affichage de la vidéo et de différentes informations. Rien de particulier si ce n'est la présence des data-barba «wrapper» et «container» nécessaires aux transitions entre les pages.

```

76      <?php if (isset($getUserInformations_id) && !empty($getUserInformations_id))
77      {
78
79          if ($getUserInformations_id != $getContentAndUserInformations_id_user) { ?>
80
81              <button class="dropbtn gray" id="like_content_button"
82                  onfocus="javascript:modal('like_content')">Like</button>
83
84              <div class="dropdown">
85                  <button class="drop">Comment/Report</button>
86                  <div class="dropdown-content">
87                      <button class="button_drop gray" id="comment_button"
88                          onfocus="javascript:modal('comment')">Comment</button>
89                      <button class="button_drop gray" id="report_button"
90                          onfocus="javascript:modal('report')">Report this Content</button>
91                  </div>
92              </div>
93
94      <?php } else { ?>
95
96          <button class="dropbtn gray" id="comment_button"
97              onfocus="javascript:modal('comment')">Comment</button>
98
99          <div class="dropdown">
100              <button class="drop">Edit/Delete</button>
101              <div class="dropdown-content">
102                  <button class="button_drop gray" id="edit_button"
103                      onfocus="javascript:modal('edit')">Edit Content</button>
104                  <button class="button_drop gray" id="delete_content_button"
105                      onfocus="javascript:modal('delete_content')">Delete Content</button>
106              </div>
107          </div>
108
109      <?php } ?>
110
111      <?php } ?>

```

Ligne 76: Si l'utilisateur est connecté

Ligne 79: Si l'utilisateur connecté n'est pas l'auteur du contenu affichage des boutons «Like», «Comment» et «Report this content» ouvrant chacun des modales.

Ligne 93: Si l'utilisateur connecté est l'auteur du contenu affichage des boutons «Comment», «Edit content» et «Delete content» ouvrant chacun des modales.

La modale liée au bouton «Edit content» sera développée dans le chapitre «Présentation d'un jeu d'essai» p.31

```

112
113
114
115
116
117
118
119
120
121
122
123

```

```

</div>

<div class="description">
  <div class="column1 gray">
    <a class="link_page" href="content.php?id=<?
    $getContentAndUserInformations_id ?>&name=visitor&category=user_content"><?
    = $getContentAndUserInformations_author_name . " " .
    $getContentAndUserInformations_author_lastname ?></a>
  </div>

  <div class="column2">
    <p><? = $getContentAndUserInformations_description ?></p>
  </div>
</div>

```

Affichage du nom de l'auteur dans un lien conduisant vers la page des contenus de cet utilisateur et affichage de la description du contenu.

```

125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147

```

```

<?php foreach ($getComments as $getComment) {

    require('./assets/require/variables.php');

    $getNumbersOfcomments = getNumbersOfcomments($bdd, $getComment_user_id);

    require('./assets/require/variables.php');

    require('./assets/require/modals_foreach.php'); ?>

    <div class='deck'>
      <div class='single_player_card'>

        <div class='cardHeader'>
          <span class='cardHeader_account'><? = $getComment_user_name . " " .
          $getComment_user_lastname ?></span>
          <span class='cardHeader_date'><? = $getComment_date ?></span>
        </div>

        <div class='cardBody'>

          <p class='cardText'><? = $getComment_text ?></p>

```

Ligne 126: Affichage de tous les commentaires liés au contenu, récupérés par la fonction `getComments()` ligne 24: (voir plus dans chapitre sur la sécurité p.46).

```

157
158
159
160
161
162
163
164
165
166
167
168
169
170

```

```

function getComments(PDO $bdd, $get_id)
{
    $req = $bdd->prepare('SELECT users.name, users.lastname, comments.comment, comments.id,
    comments.id_users, comments.date, comments.likes
    FROM comments
    INNER JOIN contents
    ON comments.id_contents = contents.id
    INNER JOIN users
    ON comments.id_users = users.id
    WHERE comments.id_contents = :contents_id ORDER BY comments.date ASC');
    $req->bindParam(':contents_id', $get_id, PDO::PARAM_INT);
    $req->execute();
    $comments = $req->fetchAll();
    return $comments;
}

```


Ligne 130: Récupération du nombres de commentaires postés par l’auteur du commentaire.

Ligne 134: Appel d’un fichier contenant les modales destinés à être appelées dans un **foreach**.

```
149 <section class='cardStats'>
150
151 <span class='cardStats_stat cardStats_stat-comments'><?=
152 $getNumbersOfcomments_from_user ?> <i class='far fa-comment
153 fa-fw'></i></span>
154
155 <span class='cardStats_stat cardStats_stat-likes'><?=
156 $getComment_likes ?> <i class="far fa-thumbs-up"></i></span>
157
158 </section>
159 </div>
160
161 <?php if (isset($session_users_id) && ($session_users_id !=
162 $getComment_user_id)) { ?>
163
164 <button class="dropbtn gray" id="like_comment_button"><?=
165 $getComment_id ?>" onfocus="javascript: modalForeach('like_comment',
166 '<?= $getComment_id ?>')>Like</button>
167
168 <?php } else if (isset($session_users_id) && ($session_users_id ==
169 $getComment_user_id)) { ?>
170
171 <button class="dropbtn gray" id="edit_comment_button"><?=
172 $getComment_id ?>" onfocus="javascript: modalForeach('edit_comment',
173 '<?= $getComment_id ?>')>Edit</button>
174
175 <?php } ?>
176
177 </div>
178 </div>
179 </div>
180 </main>
181
182 <?php require('./assets/require/footer.php'); ?>
```

Affiche la partie statistique de la carte de commentaire (le nombre de likes et le nombres de commentaires postés par l’auteur) ainsi qu’un bouton «Edit» pour éditer le commentaire ou un bouton «like» pour le liker suivant que l’on est l’auteur ou non du commentaire. Ces deux boutons ouvrent chacun une modale.

Ligne 186: Appel du footer.

La page se termine ensuite par les différentes redirections qui ont été mentionnées précédemment .

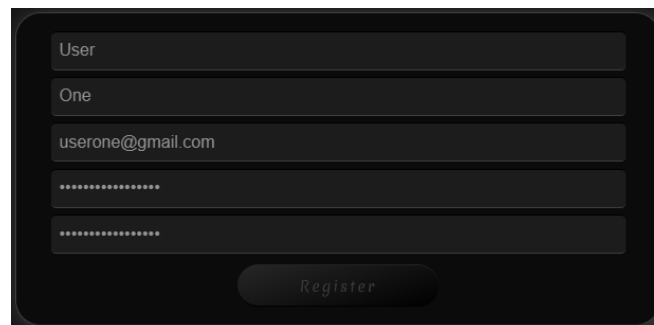
6. Jeu d'essai du comportement lié à une fonctionnalité

Pour ce jeu d'essai j'ai choisi de décrire les tests réalisés sur le script destiné à éditer un contenu.

6.1 Étapes préalables:

- Création de deux utilisateurs User One et User Two

→ Données en entrée: remplissage des formulaires d'inscription

A registration form with a dark theme. It contains five input fields: 'User' (with 'One' entered), 'email' (with 'userone@gmail.com' entered), and two password fields (both with masked characters '*****'). A 'Register' button is at the bottom.

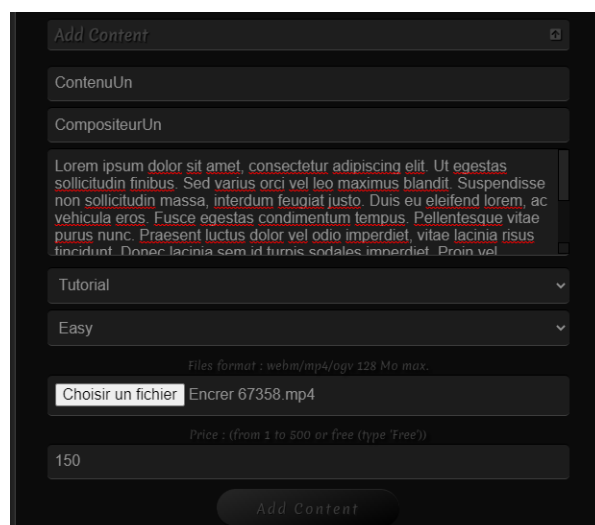
→ Données attendues: créations des deux utilisateurs avec 500 Crédits chacun

→ Données obtenues:

id	name	lastname	email	password	type	credits	time
2	User	One	userone@gmail.com	\$2y\$10\$BMzTaCZE6coyf0OPf0y6OzsJau80fUUypsdUZLdglx...	user	500	1665127895
3	User	Two	usertwo@gmail.com	\$2y\$10\$Mw10OEEv4i9cwzXQHVR53emJrS2JDF7IXU6jLcNI6KC...	user	500	1665127961

- User One crée un contenu

→ Données en entrée: remplissage formulaire d'ajout de contenu

An 'Add Content' form with a dark theme. It includes fields for 'ContenuUn', 'CompositeurUn', a large text area with placeholder text, 'Tutorial' (dropdown), 'Easy' (dropdown), a file selection button labeled 'Choisir un fichier' with 'Encreur 67358.mp4' selected, and a price field with '150' entered. A note specifies 'Files format : webm/mp4/ogv 128 Mo max.' and another note says 'Price : (from 1 to 500 or free (type "Free"))'. An 'Add Content' button is at the bottom.

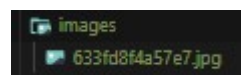
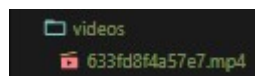
→ Données attendues:

- création d'un contenu à 150 Crédits
- gain de 300 Crédits (gain pour l'ajout d'un tutoriel) pour User One
- upload de la video dans le dossier «videos»
- extraction d'une image de la video dans le dossier «images»

→ Données obtenues:

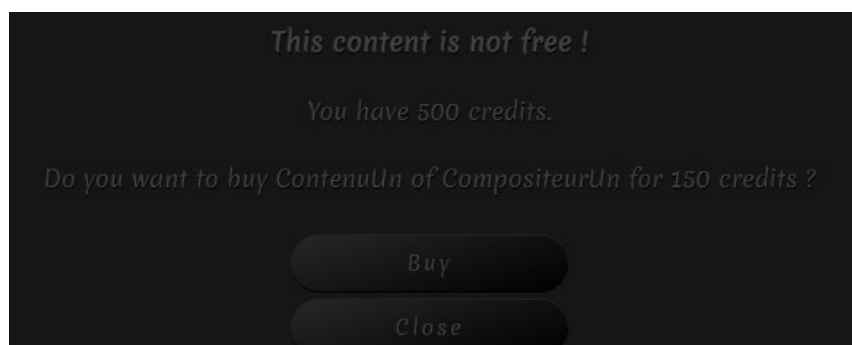
id	title	content	price	composer	category	level	description	likes	reporting	id_users
1	ContenuUn	633fd8f4a57e7.mp4	150	CompositeurUn	tutorial	easy	Lorem ipsum dolor sit amet, consectetur adipiscing...	0	0	2

id	name	lastname	email	password	type	credits	time
2	User	One	userone@gmail.com	\$2y\$10\$BMzTaCZEr6coyf0OPf0y6OzsJau80fUUypsdUZLdglx...	user	800	1665129582



- User Two achète ce contenu:

→ Données en entrée: achat du contenu



→ Données attendues:

- débit de 150 Crédits pour User Two et gain de 150/25 Crédits (arrondi à l'entier supérieur) donc 6 crédits pour User One

- ajout d'User Two dans la table des utilisateurs ayant acheté un contenu avec le prix d'achat original

→ Données obtenues:

id	name	lastname	email	password	type	credits	time
2	User	One	userone@gmail.com	\$2y\$10\$BMzTaCZE6coyf0OPf0y6OzsJau80fUUypsdUZLdglx...	user	806	1665129582
3	User	Two	usertwo@gmail.com	\$2y\$10\$Mw1OOEEv4i9cwzXQHVRs3emJrS2JDF7IXU6jLcNI6KC...	user	350	1665130670

id	id_contents	id_users	original_price	buyer_repayment
2	1	3	150	0

6.2 Test de la fonctionnalité d'édition d'un contenu

Maintenant que les choses sont mises en place je vais réaliser un jeu d'essai sur le script d'édition d'un contenu `edit_content.php` ainsi que sur les scripts annexes qui y sont liées:

6.2.1 Validation données côté client

- Je vais tout d'abord tester la fonctionnalité de validation des données utilisateur côté client avec Javascript;

→ Données en entrée: je tente d'éditer le titre du contenu et je tape un caractère qui n'est pas accepté («_» par exemple).

Remarque: le choix des caractères autorisés ou non a été établi pour être le plus restrictif possible aussi bien dans un souci de sécurité que pour la cohérence esthétique des différents textes que l'on peut rencontrer sur le site. En effet, c'est un choix qui ne relève que de ma sensibilité, mais je n'ai jamais goûté un nom d'utilisateur du type «User_158:-)» par exemple. Le choix du nom étant par ailleurs libre (la seule donnée saisie par l'utilisateur devant être unique est l'adresse mail) je ne vois pas l'intérêt d'autoriser des dénominations de ce genre.

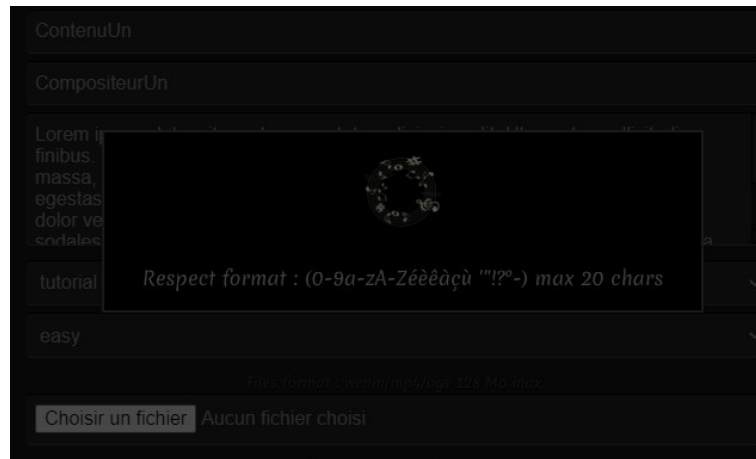
Il en va de même pour les textes qui ne peuvent contenir que les signes que l'on rencontre couramment dans un texte.

La seule entrée pour laquelle je n'ai pas restreint les caractères autorisés en fonction de mes affinités esthétiques est l'entrée destinée à l'adresse mail qui est validée grâce à une Regex (du côté client) qui est censé correspondre au format de 99 % environ des adresses mail pouvant exister et par `FILTER_VALIDATE_EMAIL` de PHP (côté serveur).*

* `^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$`

→ Données attendues: effacement du caractère qui n'est pas accepté, apparition d'une modale d'erreur spécifiant les caractères autorisés et repositionnement du curseur de l'<input> à l'endroit où l'on tentait d'éditer le texte.

→ Données obtenues:



Voici les détails de l'apparition de cette modale:

```
<label for="single_player_edit_title"></label>
<input type="text" class="inputbox" value="<?=$getContentAndUserInformations_title ?>" placeholder="<?=$getContentAndUserInformations_title ?>" id="single_player_edit_title" name="title" onkeyup="javascript:input(this,'single_player_edit_title',21,'input_title');" />
```

Au relâchement d'une touche l'<input> appelle la fonction javascript `input()`:(voir p.46)

```
16 let position;
17 let old_position;
Complexity is 14 You must be kidding
18 function input(string, id, maxLength, index) {
Complexity is 4 Everything is cool!
19   function position_event(id) {
20     document.getElementById(id).addEventListener("keydown", function () {
21       position = this.selectionStart;
22       return position;
23     });
24
25     old_position = position;
26     return old_position;
27   }
28   position_event(id);
29
30   let array_regex = {
31     input_name_lastname: /^(?!s*$)[a-zA-Zéèëââçù '"]+$/,
32     input_password: /^[0-9a-zA-Z]+$/,
33     input_title: /^[0-9a-zA-Zéèëââçù '!"?°-]+$/,
34     input_composer: /^[0-9a-zA-Zéèëââçù -]+$/,
35     input_description: /^[\\s\\r\\n0-9a-zA-Zéèëââçù# ()'".!?,;:°-]+$/,
36     input_contact: /^[\\s\\r\\n0-9a-zA-Zéèëââçù# ()'".!?,;:°-]+$/,
37     input_price: /^[1-9][1-9][0-9][1-4][0-9][0-9][500|[Free]]+$/,
38   };
39   let regex = array_regex[index];
```

```

40
41     if (
42         (string.value.match(regex) && string.value.length < maxLength) ||
43         event.key === "Tab" ||
44         event.key === "Backspace" ||
45         event.key === "Enter"
46     ) {
47         return true;
48     } else {
49         string_value = document.getElementById(id).value;
50         document.getElementById(id).value = string_value.replace(event.key, "");
51
52         document.getElementById(id).focus();
53         document.getElementById(id).setSelectionRange(old_position, old_position);
54
55         let input_modal = document.getElementById("input_modal");
56         let message_regex = document.getElementById(index);
57
58         input_modal.style.display = "flex";
59         message_regex.style.display = "flex";
60
61         document.onclick = function () {
62             input_modal.style.display = "none";
63             message_regex.style.display = "none";
64         };
65
66         document.onkeydown = function () {
67             input_modal.style.display = "none";
68             message_regex.style.display = "none";
69         };
70     }
71 }

```

Ligne 18: Cette fonction prend quatre paramètres:

- le texte concerné
- l'id de l'<input> visée par la fonction
- la longueur du texte maximum
- l'index de la Regex utilisée pour cette entrée

Ligne 19: Fonction servant à récupérer la position du curseur dans l'<input> à l'enfoncement d'une touche

Ligne 30: Array comprenant les différentes Regex associées aux entrées correspondantes

Ligne 39: Variable contenant la Regex correspondant à l'entrée concernée

Ligne 42: Condition vérifiant si le caractère entré est autorisé ou non et si la longueur de l'entrée est inférieur à la longueur maximum autorisée.

Lignes 43/44/45: Conditions servant à pouvoir naviguer dans les formulaires avec la touche TAB, effacer des caractères et faire des retours à la ligne sans générer l'apparition de la modale d'erreur et l'effacement de caractères.

Lignes 49/50: Récupération de la valeur de l'entrée et effacement du dernier caractère tapé si celui n'est pas autorisé.

Lignes 52/53: Repositionnement du curseur, après l'effacement d'un caractère, à l'endroit de l'édition avant l'entrée d'un caractère non autorisé

Lignes 58/59: Affichage de la modale d'erreur et de son message d'erreur correspondant en fonction des éléments contenu dans les variables des lignes 55/56

```
<div id="input_modal" class="modal_messages">
  <div class="modal_content">
    <div class="modal_form">
      <div class="modal_form_content messages">
        <form class="form_action">

          <div class="messages_logo">
            
          </div>

          <br>

          <div id="input_price">Price : from 1 to 500 or free (type 'Free')</div>
          <div id="input_contact">Respect format : (0-9a-zA-Zèèèàçü# ()'".!?,;:-) max 10000 chars </div>
          <div id="input_description">Respect format : 0-9a-zA-Zèèèàçü# ()'".!?,;:-) max 1500 chars </div>
          <div id="input_composer">Respect format : (0-9a-zA-Zèèèàçü -) max 20 chars </div>
          <div id="input_title">Respect format : (0-9a-zA-Zèèèàçü '!"?)-) max 20 chars </div>
          <div id="input_password">Respect format : (0-9a-zA-Z) min 2 max 40 chars</div>
          <div id="input_name_lastname">Respect format : (a-zA-Zèèèàçü '"- ) min 1 max 10 chars</div>
          <div id="input_content">Respect files format : (0-9a-zA-Zèèèàçü# ()'\!?,;:-) (.) (webm/mp4/ogv) and 128 Mo max</div>

        </form>
      </div>
    </div>
  </div>
</div>
```

Lignes 61/68: La modale d'erreur disparaît si l'on clique n'importe où sur l'écran ou si l'on tape un nouveau caractère. Cela permet de taper sans se soucier de cette modale qui effacera les caractères non autorisé sans intervention de la part de l'utilisateur. Son apparition et sa disparition sont donc relativement transparente.

Ce système est mis en place sur toutes les entrées utilisateur du côté client et du côté serveur. Des variantes existent pour l'entrée email comme je l'ai expliqué plus haut et une autre variante pour l'upload des vidéos qui nécessite des vérifications supplémentaires.

Ce système faisant parti des mesures de sécurité mises en place dans le projet, il sera, comme je l'ai dit dans le chapitre 5.2.1 p.24(en référence cette fois là à l'équivalent de ce système mis en place côté serveur), développé dans le chapitre sur la sécurité p.46

6.2.2 Test de la fonctionnalité d'édition d'un contenu côté serveur

- Modification de la description du contenu en y intégrant volontairement des espaces et des sauts de lignes inutiles

→ Données en entrée

```

Lorem ipsum dolor      sit amet, consectetur adipiscing elit.

Ut egestas  sollicitudin      finibus. Sed varius orci vel leo  maximus
blandit. Suspendisse non sollicitudin massa, interdum feugiat justo.

Duis eu eleifend lorem, ac vehicula eros. Fusce      egestas condimentum
tempus. Pellentesque vitae purus nunc. Praesent luctus dolor vel odio imperdiet,
vitae lacinia risus tincidunt. Donec lacinia      sem id turpis sodales imperdiet.
Proin vel porttitor ante. Etiam efficitur,      massa et suscipit porta, nibh nibh varius
justo, vitae convallis mi ligula id odio. Maecenas accumsan libero ipsum, in cursus
augue dignissim vel. Aliquam erat volutpat. Nulla mollis mauris urna, et tincidunt
lacus dictum at. Vivamus.
```

→ Données attendues

- Mise en forme du texte avec suppression des espaces et des sauts de lignes en trop pour uniformiser l'esthétique du site.

→ Données obtenues

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Ut egestas sollicitudin finibus. Sed varius orci vel leo maximus blandit. Suspendisse
non sollicitudin massa, interdum feugiat justo.

Duis eu eleifend lorem, ac vehicula eros. Fusce egestas condimentum tempus.
Pellentesque vitae purus nunc. Praesent luctus dolor vel odio imperdiet, vitae lacinia
risus tincidunt. Donec lacinia sem id turpis sodales imperdiet. Proin vel porttitor
ante. Etiam efficitur, massa et suscipit porta, nibh nibh varius justo, vitae convallis
mi ligula id odio. Maecenas accumsan libero ipsum, in cursus augue dignissim vel.
Aliquam erat volutpat. Nulla mollis mauris urna, et tincidunt lacus dictum at.
Vivamus.
```

Voici les détails de cette fonctionnalité contenu dans le fichier **check_data.php**: (voir p.46)

```

340 if (isset($_POST['description']) && !empty($_POST['description'])) {
341
342     $check_post_description = is_string($_POST['description'])
343         && preg_match("/^[\\s0-9a-zA-Zéèëääçü# ()'\".!?,:;-]+$/", $_POST['description'])
344         && strlen($_POST['description']) <= 1502;
345
346     if ($check_post_description === true) {
347
348         $sanitize_one = preg_replace('/ +/', " ", $_POST['description']);
349         $sanitize_two = preg_replace('/^ /', "", $sanitize_one);
350         $sanitize_three = preg_replace('/ $/', "", $sanitize_two);
351
352         $regex = '/(\\r\\n|\\n|\\t|\\r){3,}/';
353         $replacement = "\\r\\n\\r\\n";
354
355         $post_description = preg_replace($regex, $replacement, $sanitize_three);
356     }
357 }
```

- **Modification de catégorie du contenu**

→ Données en entrée

- modification de la catégorie de Tutoriel vers Performance

→ Données attendues

- débit de 200 Crédits pour User One

→ Données obtenues

id	name	lastname	email	password	type	credits	time
2	User	One	userone@gmail.com	\$2y\$10\$BMzTaCZE6coyf0OPf0y6OzsJau80fUUypsdUZLdglx...	user	606	1665585028

- **Modification du prix du contenu**

→ Données en entrée

- modification du prix de 150 Crédits à 100 Crédits

→ Données attendues

- gain de 50 Crédits pour User Two

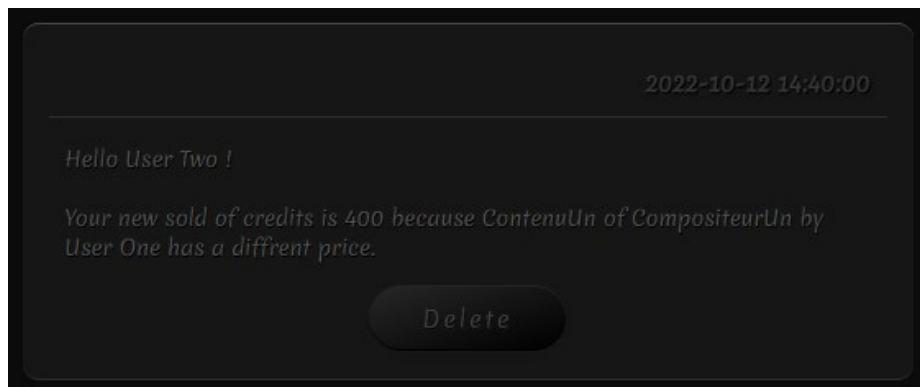
- mise à jour de la colonne remboursement à l'acheteur de la table des utilisateurs ayant acheté du contenu

- envoi d'une notification à User Two

→ Données obtenues

id	name	lastname	email	password	type	credits	time
▼ 1							
3	User	Two	usertwo@gmail.com	\$2y\$10\$Mw10OEEv4i9cwzXQHVRs3emJrS2JDF7IXU6jLcNI6KC...	user	400	1665130670

id	id_contents	id_users	original_price	buyer_repayment
2	1	3	150	50



Voici les détails de cette fonctionnalité:

```

79      $req = $bdd->prepare('SELECT purchased_contents.id_contents, purchased_contents.id_users,
80      purchased_contents.original_price, purchased_contents.buyer_repayment ,users.credits,
81      users.name, users.lastname
82      FROM purchased_contents
83      INNER JOIN users
84      ON purchased_contents.id_users = users.id
85      WHERE purchased_contents.id_contents = :id_contents');
86      $req->bindParam(':id_contents', $post_id, PDO::PARAM_INT);
87      $req->execute();
88      $repayment_informations = $req->fetchAll();
89
90      $new_price = $post_price;
91
92      if ($post_price == 'Free') {
93          $new_price = 0;
94      }

```

Ligne 79: récupération de différentes données

Ligne 89 à 92: le nouveau prix est égal au nouveau prix fixé par l'auteur du contenu. *Remarque: pour rendre le contenu gratuit il est nécessaire de taper «Free» dans l'input du formulaire car lorsque que l'on entre zéro \$post_price a une valeur = NULL. Il aurait été possible d'utiliser la variable globale \$_POST['price'] pour éviter ce problème mais procéder de cette manière permet de s'assurer que l'utilisateur rendant son contenu gratuit en a bien conscience.*

```

95      if ($content_informations['price'] != $new_price) {
96
97          foreach ($repayment_informations as $repayment_informations_foreach_buyer) {
98
99              $original_price = $repayment_informations_foreach_buyer['original_price'];
100             $buyer_repayment = $repayment_informations_foreach_buyer['buyer_repayment'];
101             $old_buyer_repayment = $buyer_repayment;

```

Ligne 95: cette condition permet de ne pas exécuter une partie du code inutile quand l'auteur du contenu ne change pas le prix.

Ligne 99 à 1001: stockage de différentes valeurs dans des variables


```

104         if ($original_price > $new_price) {
105             $buyer_repayment = 0;
106             $buyer_repayment = $original_price - $new_price;
107
108             $new_sold_of_credits = $repayment_informations_foreach_buyer['credits'] +=
109             $buyer_repayment - $old_buyer_repayment;
110
111             $req = $bdd->prepare('UPDATE users SET credits = :credits WHERE id = :id');
112             $req->bindParam(':credits', $new_sold_of_credits, PDO::PARAM_INT);
113             $req->bindParam(':id', $repayment_informations_foreach_buyer['id_users'],
114             PDO::PARAM_INT);
115             $req->execute();
116         } else if ($original_price == $new_price) {
117             $new_sold_of_credits = $repayment_informations_foreach_buyer['credits'] -=
118             $buyer_repayment;
119
120             $req = $bdd->prepare('UPDATE users SET credits = :credits WHERE id = :id');
121             $req->bindParam(':credits', $new_sold_of_credits, PDO::PARAM_INT);
122             $req->bindParam(':id', $repayment_informations_foreach_buyer['id_users'],
123             PDO::PARAM_INT);
124             $req->execute();
125
126             $buyer_repayment = 0;
127         }
128
129         $req = $bdd->prepare('UPDATE purchased_contents SET buyer_repayment =
130         :buyer_repayment WHERE id_users = :id_users AND id_contents = :id_contents');
131         $req->bindParam(':buyer_repayment', $buyer_repayment, PDO::PARAM_INT);
132         $req->bindParam(':id_users', $repayment_informations_foreach_buyer['id_users'],
133         PDO::PARAM_INT);
134         $req->bindParam(':id_contents', $post_id, PDO::PARAM_INT);
135         $req->execute();

```

Ici, différents calculs avec les valeurs récupérées plus haut et mise à jour des crédits des acheteurs ainsi que de la colonne remboursement de la table des utilisateurs ayant achetés un contenu.

```

135     $date = date('Y-m-d H:i:s');
136
137     if ($new_price == 0) {
138         $notification = 'Hello ' . $repayment_informations_foreach_buyer['name'] . '
139         ' . $repayment_informations_foreach_buyer['lastname'] . ' !
140
141         Your new sold of credits is ' . $new_sold_of_credits . ' because ' .
142         $content_informations['title'] . ' of ' . $content_informations['composer'] . '
143         by ' . $content_informations['name'] . ' ' . $content_informations['lastname'] .
144         ' is now Free.
145
146         You have been reimbursed ' . $buyer_repayment . ' $';
147
148         $req = $bdd->prepare('INSERT INTO notifications (notification, date,
149         id_users) VALUES (:notification, :date, :id_users) ');
150         $req->bindParam(':notification', $notification, PDO::PARAM_STR);
151         $req->bindParam(':date', $date, PDO::PARAM_STR);
152         $req->bindParam(':id_users', $repayment_informations_foreach_buyer
153         ['id_users'], PDO::PARAM_INT);
154         $req->execute();
155     } else {
156         $notification = 'Hello ' . $repayment_informations_foreach_buyer['name'] . '
157         ' . $repayment_informations_foreach_buyer['lastname'] . ' !
158
159         Your new sold of credits is ' . $new_sold_of_credits . ' because ' .
160         $content_informations['title'] . ' of ' . $content_informations['composer'] . '
161         by ' . $content_informations['name'] . ' ' . $content_informations['lastname'] .
162         ' has a different price.';
163
164         $req = $bdd->prepare('INSERT INTO notifications (notification, date,
165         id_users) VALUES (:notification, :date, :id_users) ');
166         $req->bindParam(':notification', $notification, PDO::PARAM_STR);
167         $req->bindParam(':date', $date, PDO::PARAM_STR);
168         $req->bindParam(':id_users', $repayment_informations_foreach_buyer
169         ['id_users'], PDO::PARAM_INT);
170         $req->execute();
171     }
172 }
173 }

```

Ici, envoi des notifications aux utilisateurs ayant acheté le contenu.

- Upload d'une nouvelle video

→ Données en entrée

- upload d'une nouvelle video

→ Données attendues

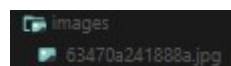
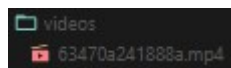
- mise jour de la table contenu avec le nouveau nom de la vidéo

- upload de la nouvelle vidéo dans le dossier «vidéos» et suppression de l'ancienne

- extraction d'une nouvelle image, stockage dans le dossier «images» et suppression de l'ancienne

→ Données obtenues

id	title	content	price	composer	category	level	description	likes	reporting	id_users
1	ContenuUn	63470a241888a.mp4	100	CompositeurUn	performance	easy	Lorem ipsum dolor sit amet, consectetur adipiscing...	0	0	2



Voici les détails de cette fonctionnalité:

```

419 if (isset($_FILES) && !empty($_FILES)) {
420     if (array_key_exists('content', $_FILES)) {
421
422         $explode_files_name = explode('.', $_FILES['content']['name']);
423         $explode_files_tmp_name = explode('.', $_FILES['content']['tmp_name']);
424
425         if (
426             count($explode_files_name) == 2
427             && count($explode_files_tmp_name) == 2
428         ) {
429
430             $allowed_extensions = ["webm", "mp4", "ogv"];
431             $allowed_mime_types = ["video/webm", "video/mp4", "video/ogv"];
432
433             $check_files_name = is_string($explode_files_name['0'])
434                 && preg_match("/^[0-9a-zA-Zèëääçû# ()'! ,;'-]+$/", $explode_files_name['0']);

```

```

436     $check_files_extension = in_array(strtolower(end($explode_files_name)),
437     $allowed_extensions);
438
439     $check_files_mime_type = in_array((strtolower(fininfo_file(fininfo_open
440     (FILEINFO_MIME_TYPE), $_FILES['content']['tmp_name']))), $allowed_mime_types);
441
442     if (
443         $check_files_name === true
444         && $check_files_extension === true
445         && $check_files_mime_type === true
446     ) {
447
448         $files_content_name = $_FILES['content']['name'];
449         $files_content_type = $_FILES['content']['type'];
450         $files_content_tmp_name = $_FILES['content']['tmp_name'];
451         $files_content_error = $_FILES['content']['error'];
452         $files_content_size = $_FILES['content']['size'];
453     }
454 }

```

Vérification du nom, de l'extension et du type mime du fichier ainsi que de l'absence d'une double extension. Cette vérification contenu dans le fichier `check_data.php` sera également développé dans la chapitre sur la sécurité p.46

```

165     if (isset($files_content_name)) {
166
167         if ($files_content_error == 0) {
168
169             if ($files_content_size <= 128000000) {
170
171                 $content = uniqid() . '.' . pathinfo($files_content_name, PATHINFO_EXTENSION);
172                 move_uploaded_file($files_content_tmp_name, '../videos/' . $content);
173             } else {
174
175                 if ($get_type == 'admin') {
176
177                     $bdd = null;
178                     header('location: ../../admin/contents.php?error=017129');
179                     die();
180                 } else {
181
182                     $bdd = null;
183                     header('location: ../../single_player_content.php?id=' . $post_id . '&
184                     error=017154');
185                     die();
186                 }
187             }
188         } else {
189
190             if ($get_type == 'admin') {
191
192                 $bdd = null;
193                 header('location: ../../admin/contents.php?error=017130');
194                 die();
195             } else {
196
197                 $bdd = null;
198                 header('location: ../../single_player_content.php?id=' . $post_id . '&
199                 error=017155');
200                 die();
201             }
202         }
203     }

```

Ensuite dans le fichier `edit_content.php`

Lignes 165 à 169: il est vérifié que le fichier à bien passé les vérifications de `check_data.php`, l'absence d'erreur et que sa taille est inférieure à celle qui est spécifiée.

Lignes 171/172: renommage du fichier avec un identifiant unique et upload du fichier dans le dossier «videos».

Lignes 175 à 198: redirections suivant que l'on est administrateur ou non.

```
204     if (!isset($content) && empty($content)) {
205
206
207         $req = $bdd->prepare('UPDATE contents SET title = :title ,composer = :composer, level
        = :level, category = :category, price = :price, description = :description, reporting
        = :reporting WHERE id = :id');
208         $req->bindParam(':title', $post_title, PDO::PARAM_STR);
209         $req->bindParam(':composer', $post_composer, PDO::PARAM_STR);
210         $req->bindParam(':level', $post_level, PDO::PARAM_STR);
211         $req->bindParam(':category', $post_category, PDO::PARAM_STR);
212         $req->bindParam(':price', $new_price, PDO::PARAM_INT);
213         $req->bindParam(':description', $post_description, PDO::PARAM_STR);
214         $req->bindParam(':reporting', $_POST['reporting'], PDO::PARAM_INT);
215         $req->bindParam(':id', $post_id, PDO::PARAM_INT);
216         $req->execute();
217     } else {
218
219         $req = $bdd->prepare('SELECT * FROM contents WHERE id = :id');
220         $req->bindParam(':id', $post_id, PDO::PARAM_INT);
221         var_dump($req);
222         $req->execute();
223         $old_content = $req->fetch();
224
225         unlink('../videos/' . $old_content['content']);
226
227         $image = explode('.', $old_content['content']);
228
229         unlink('../images/' . $image[0] . '.jpg');
230
231         require('../require/frame.php');
232
233         $req = $bdd->prepare('UPDATE contents SET title = :title ,composer = :composer, level
        = :level, category = :category, price = :price, description = :description, reporting
        = :reporting, content = :content WHERE id = :id');
234         $req->bindParam(':title', $post_title, PDO::PARAM_STR);
235         $req->bindParam(':composer', $post_composer, PDO::PARAM_STR);
236         $req->bindParam(':level', $post_level, PDO::PARAM_STR);
237         $req->bindParam(':category', $post_category, PDO::PARAM_STR);
238         $req->bindParam(':price', $new_price, PDO::PARAM_INT);
239         $req->bindParam(':description', $post_description, PDO::PARAM_STR);
240         $req->bindParam(':reporting', $_POST['reporting'], PDO::PARAM_INT);
241         $req->bindParam(':content', $content);
242         $req->bindParam(':id', $post_id, PDO::PARAM_INT);
243         $req->execute();
244     }
```

Ligne 204: une nouvelle vidéo a-t-elle été postée ou non ? Si ce n'est pas le cas exécution des lignes 207 à 216.

sinon

Ligne 219 à 223: récupération du nom de l'ancienne vidéo.

Ligne 225: suppression de l'ancienne vidéo du dossier «vidéo»

Ligne 227/229: remplacement de l'extension de la vidéo par l'extension .jpg et suppression de l'image correspondante du dossier «images».

Ligne 231: appel du fichier **frame.php**

```
1 <?php
2 require('../vendor/autoload.php');
3
4 if (isset($content) && !empty($content)) {
5
6     $frame = explode('.', $content);
7     $video_file = '../assets/videos/' . $content;
8     $frame_second = 1;
9     $ffmpeg = FFMpeg\FFMpeg::create();
10    $video = $ffmpeg->open(dirname("../Diplome") . '/' . $video_file);
11    $video
12        ->frame(FFMpeg\Coordinate\TimeCode::fromSeconds($frame_second))
13        ->save('../assets/images/' . $frame[0] . '.jpg');
14 }
```

Utilisation de la librairie FFMpeg pour extraire une image de la vidéo.

Ligne 233 à 243: Mise à jour de la table des contenus.

7.Veille sur les questions de sécurité

Tout au long de ce projet, j'ai essayé de mettre en oeuvre de bonnes pratiques de sécurité, et d'effectuer une veille sur les éventuelles vulnérabilités de sécurité de mon projet. Voici une liste des principaux points que j'ai mis en place grâce à cette veille :

→ en premier lieu, on peut citer la différenciation, au niveau de la base de donnée, des simples utilisateurs, des administrateurs

id	name	lastname	email	password	type	credits	time
2	User	One	userone@gmail.com	\$2y\$10\$BMzTaCZEr6coyf0OPf0y6OzsJau80fUUypsdUZLdglx...	admin	606	1665599972
3	User	Two	usertwo@gmail.com	\$2y\$10\$Mw1OOEEv4i9cwzXQHVRs3emJrS2JDF7IXU6jLcNI6KC...	user	400	1665585804

→ les mots de passe sont hashés en base de données

```
$password = password_hash($post_password, PASSWORD_BCRYPT);
```

→ l'identifiant de session est renouvelé toutes les 5 minutes (une fréquence de renouvellement plus élevée peut entraîner des problèmes sur mobile liés à une plus grande instabilité des réseaux) pour prévenir la fixation de session et dans une certaine mesure les attaques CSRF (dans une certaine mesure car je ne régénère l'identifiant que toutes les 5 minutes pour la raison susmentionnée).

```
1 <?php
2 if (isset($session_users_id)) {
3
4     $old_session = session_id();
5
6     $req = $bdd->prepare('SELECT time FROM users WHERE id= :session_users_id');
7     $req->bindParam(':session_users_id', $session_users_id);
8     $req->execute();
9     $time = $req->fetch();
10
11     $time = intval(implode($time));
12
13     if ($time + 300 - time() <= 0) {
14
15         $time = time();
16
17         $req = $bdd->prepare('UPDATE users SET time = :time WHERE id= :session_users_id');
18         $req->bindParam(':time', $time);
19         $req->bindParam(':session_users_id', $session_users_id);
20         $req->execute();
21
22         session_regenerate_id();
23         $new_session = session_id();
24
25         if (isset($new_session)) {
26
27             session_write_close();
28             session_id($new_session);
29         } else {
30
31             session_write_close();
32             session_id($old_session);
33         }
34     }
35 }
```

→ déconnexion automatique au bout de 15 min pour la prévention de fixation de session également

```
3 if (isset($session_users_id)) {
4
5     $req = $bdd->prepare('SELECT time FROM users WHERE id= :session_users_id');
6     $req->bindParam(':session_users_id', $session_users_id);
7     $req->execute();
8     $time = $req->fetch();
9
10     $time = intval(implode($time));
11
12     if ($time + 900 - time() <= 0) {
13
14         unset($_SESSION['users']);
15         session_destroy();
16         $bdd = null;
17         header('location: index.php?success=007246');
18         die();
19     }
20 }
```

→ pour la prévention des failles XSS toutes les variables appelées sont entourées par **htmlspecialchars** comme vous pouvez le voir avec cet exemple p.26 et sont contenu dans un seul fichier **variables.php** pour alléger la lecture du code et faciliter la modification. *Remarque: comme vous pourrez le constater dans le point suivant, les données en entrée ne sont pas nettoyées en échappant ou en supprimant des caractères car des attaquants savent contourner cette protection. La meilleure façon de se protéger des failles XSS est donc de convertir les caractères spéciaux en entité html en utilisant htmlspecialchars.*

→ les entrées utilisateur sont filtrées de la manière la plus restrictive possible du côté client (voir p.34) et du côté serveur comme par exemple p.25 ou p.37. Ceci vient donc en complément du point précédant pour la prévention des failles XSS et du point suivant pour la prévention des injections SQL (bien que la validation des entrées utilisateur ne soit pas une solution directe à l'injection SQL, elle aide à éviter que des données utilisateur malveillantes ne soient interprétées par la base de données).

Cette façon de procéder permet de s'assurer que les données ont bien le format attendu, de ne pas dénaturer des données inoffensives par erreur en supprimant des caractères ou de ne pas augmenter la place prise par des données en échappant des caractères préalablement à leur entrée dans la base.

Comme précisé à la p.41 pour l'upload d'un fichier une vérification d'une éventuelle double extension est réalisée pour éviter qu'un attaquant puisse transmettre un fichier avec un nom du genre **filename.php.mp4** qui pourrait s'exécuter sur le serveur comme un fichier php et engendrer toute une somme de désagréments.

→ pour la prévention des injections SQL toutes les interactions avec la base de données sont faites avec des requêtes préparées utilisant PDO (PHP data objects) avec des paramètres liés (bindParam) comme ici par exemple p.29.

8. Situation de travail ayant nécessité une recherche

Lors de la création de la partie front-end du site, j'avais dans l'idée d'animer le maximum d'éléments de manière un peu originale dont les boutons. Les boutons que j'avais mis en place présentaient un dégradé et lorsque je les survolais j'inversais le dégradé. Simplement cela ne me plaisait pas que le changement de dégradé soit instantané, j'ai donc essayé d'appliquer une transition mais je me suis aperçu que cela ne fonctionnait pas. Je suis donc aller chercher une solution à mon problème sur internet et j'ai trouvé une solution sur un site anglophone.

8.1 Extrait du site en version originale

In, CSS, you can't transition a background gradient. It sure would be nice if you could:

```
.gradient {  
  background-image: linear-gradient(  
    to right,  
    hsl(211, 100%, 50%),  
    hsl(179, 100%, 30%)  
  );  
  transition: background-image 0.5s linear;  
}  
  
.gradient:hover {  
  background-image: linear-gradient(  
    to bottom,  
    hsl(344, 100%, 50%),  
    hsl(31, 100%, 40%)  
  );  
}
```

But, no. It jumps from one gradient to the other immediately, with no smooth transition between the two. So let's hack it!

We can achieve this effect with the help of a pseudo-element and an opacity transform instead.

First, apply one gradient to the element. Then, position its pseudo-element to fill the element and apply the second gradient to that. To transition between the two gradients, transition the opacity of the pseudo-element.


```

.gradient {
  position: relative;
  background-image: linear-gradient(
    to right,
    hsl(211, 100%, 50%),
    hsl(179, 100%, 30%)
  );
  z-index: 1;
}

.gradient::before {
  position: absolute;
  content: "";
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  background-image: linear-gradient(
    to bottom,
    hsl(344, 100%, 50%),
    hsl(31, 100%, 40%)
  );
  z-index: -1;
  transition: opacity 0.5s linear;
  opacity: 0;
}

.hover::before {
  opacity: 1;
}

```

The pseudo-element is initially hidden via `opacity: 0`. On hover, that transitions to `opacity: 1`. This produces the illusion of the main gradient transitioning to the pseudo-element's gradient. It also takes a little bit of `z-index` work to ensure the pseudo-element stays positioned behind the content of the main element.

This takes a fair bit of code, unfortunately. But if you need this effect, this is the best (and only) way I've found so far to accomplish it.

8.2 Traduction effectuée sans traducteur automatique

En css, vous ne pouvez pas effectuer une transition de dégradé. Ce serait certainement une bonne chose:

Mais non. Cela saute d'un dégradé à l'autre immédiatement sans transition fluide entre les deux. Alors trichons!

Nous pouvons obtenir cet effet grâce à un pseudo-élément et une opacité à la place.

Premièrement, appliquez un dégradé à l'élément. Ensuite, positionnez son pseudo-élément de manière à ce qu'il remplisse l'élément et appliquez un second dégradé à celui-ci. Pour effectuer une transition entre les deux dégradés, utilisez l'opacité du pseudo élément.

Le pseudo-élément est initialement caché via une opacité à 0. Au survol, cela transitionne vers une opacité de 1. Ceci produit l'illusion que le dégradé principal transitionne vers le dégradé du pseudo-élément. Cela nécessite également un petit peu de z-index pour s'assurer que le pseudo-élément reste positionné derrière le contenu de l'élément principal.

Cela demande un peu de code, malheureusement. Mais si vous voulez cet effet, c'est la meilleure(et unique) manière que j'ai trouvé jusqu'à maintenant pour accomplir cela.

8.3 Extrait du code CSS obtenu suite à cette recherche

```
.button_drop {
  background: linear-gradient(162.5deg, □#272727 0%, □#000000 100%);
  position: relative;
  z-index: 1;
  padding: 11px;
  height: 58px;
  border: none;
  border-top: 1.5px solid □rgb(64, 64, 64);
  border-bottom: 1.5px solid □rgb(0, 0, 0);
  color: □rgb(66, 66, 66);
  text-shadow: 1px 1px 1px □black, 1px 1px 1px □#343434;
  font-family: var(--primary-font);
  font-weight: 400;
  font-size: 0.7rem;
  min-width: 75px;
  letter-spacing: 0.11rem;
  outline: none;
  transition: color 1s;
  width: 100%;
  cursor: pointer;
  &.gray {
    &:before {
      background-image: linear-gradient(162.5deg, □#000000 0%, □#272727 100%);
      padding: 11px;
      position: absolute;
      content: "";
      top: 0;
      right: 0;
      bottom: 0;
      left: 0;
      z-index: -1;
      transition: opacity 0.7s linear;
      opacity: 0;
    }
  }
  &.gray {
    &:hover {
      color: □#000000;
      text-shadow: 1px 1px 1px □#343434, 1px 1px 1px □rgb(66, 66, 66);
    }
  }
  &.gray:hover::before {
    opacity: 1;
  }
}
```

9. Conclusion

Tout d'abord, je voudrais préciser que ce projet, au moment de la rédaction de ce rapport, a été pour moi l'occasion de mettre en œuvre les connaissances que j'ai acquises tout au long de ma formation et m'a permis d'acquérir, de consolider, mes compétences avec les langages «vanilla».

Cependant j'ai bien conscience que ce n'est que le début d'un apprentissage passionnant, qui n'aura sans doute jamais de fin. Le secteur du numérique étant en constante évolution, il faut savoir mettre à jour ses acquis et l'apprentissage perpétuel fait donc partie intégrante du métier de développeur web.

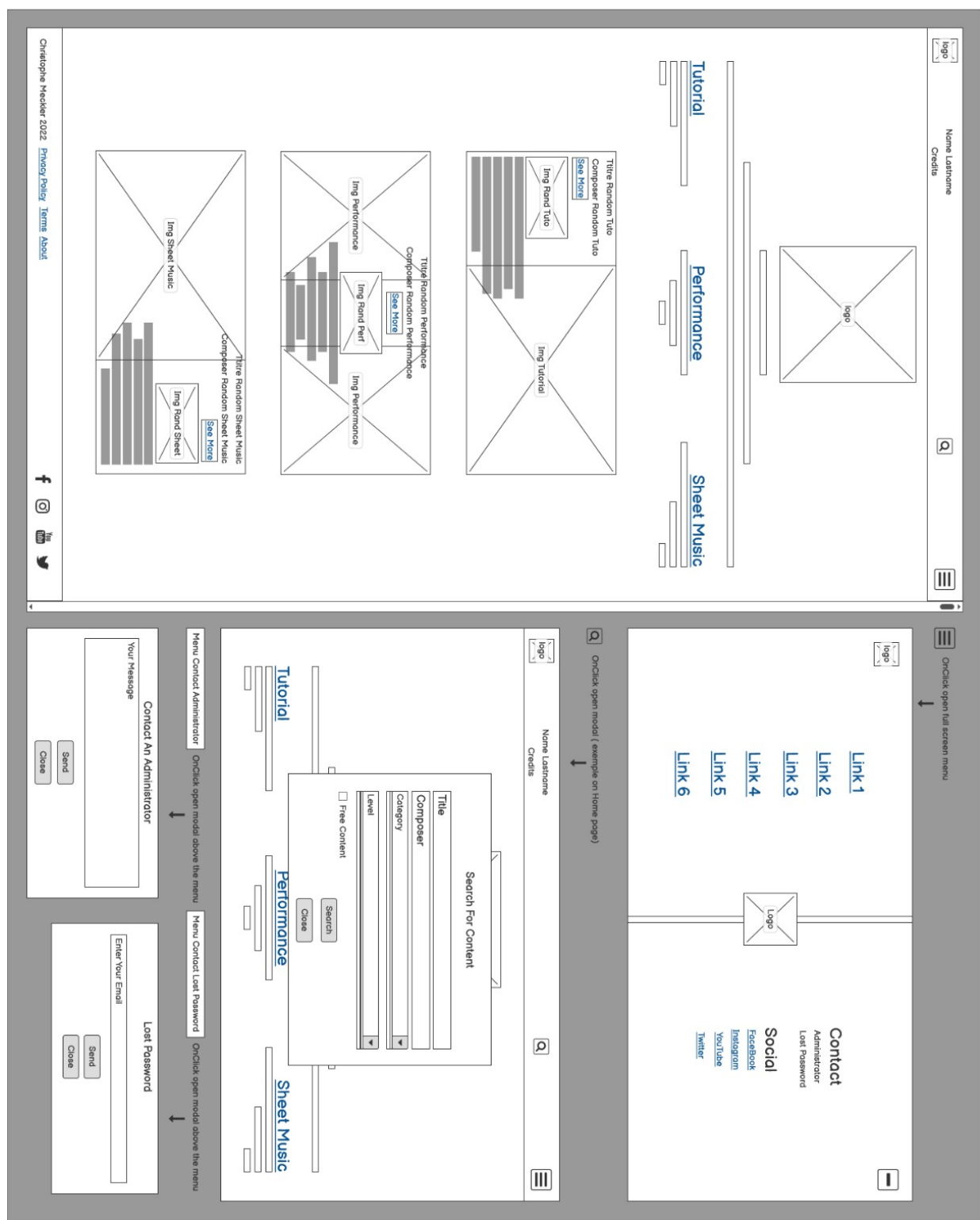
Ce qui m'amène à un point capital dont j'ai pu me rendre compte lors du développement de ce projet. C'est que, plus que de se reposer sur les connaissances que l'on possède à un moment donné, ce qui importe avant tout c'est notre capacité à être autonome, de savoir aller rechercher les informations dont on a besoin sur internet et de savoir les mettre en pratique. Car de ce que j'ai pu constater, bien que certaines choses soient relativement codifiées, il n'y a pas vraiment de solution «clé en main» dans l'univers du développement. Il a toujours des réglages à effectuer pour qu'une solution (ou une piste du moins), trouvé sur un forum etc., soit fonctionnelle dans le cas qui nous concerne.

Suite à la rédaction de ce rapport et dans les semaines qui suivront cet examen, j'ai l'intention de refaire toute la partie back-end de ce projet avec le framework Symfony afin d'obtenir un résultat au moins équivalent à ce que je vous présente dans ce rapport ce qui me permettra de mettre en perspective les avantages et les inconvénients des deux approches.

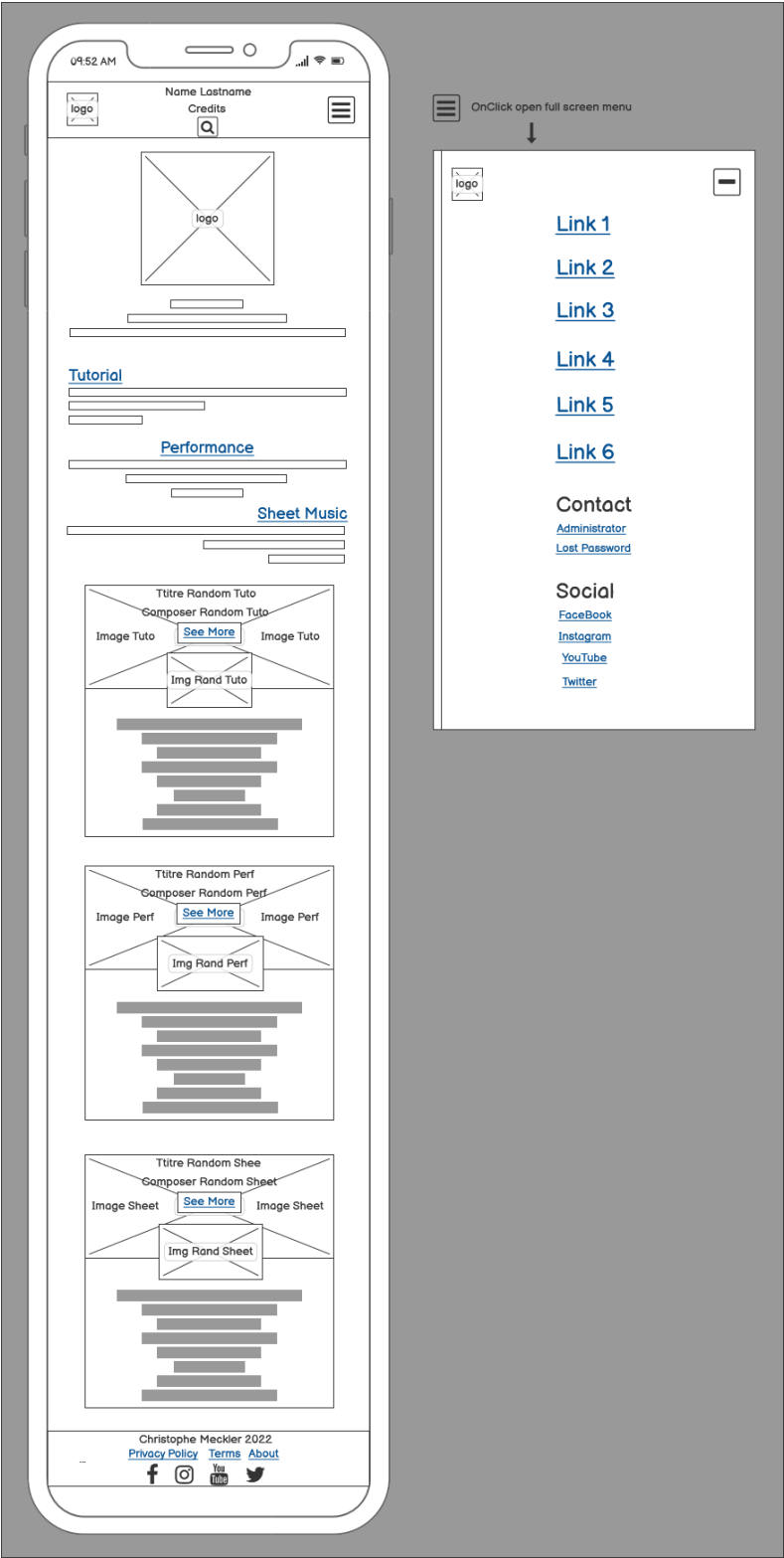
Enfin, ce projet a confirmé que le développement web est une voie professionnelle dans laquelle j'ai envie de persévérer et que l'endurance nécessaire à l'exercice de ce métier correspond à ma manière d'appréhender les choses.

Annexes

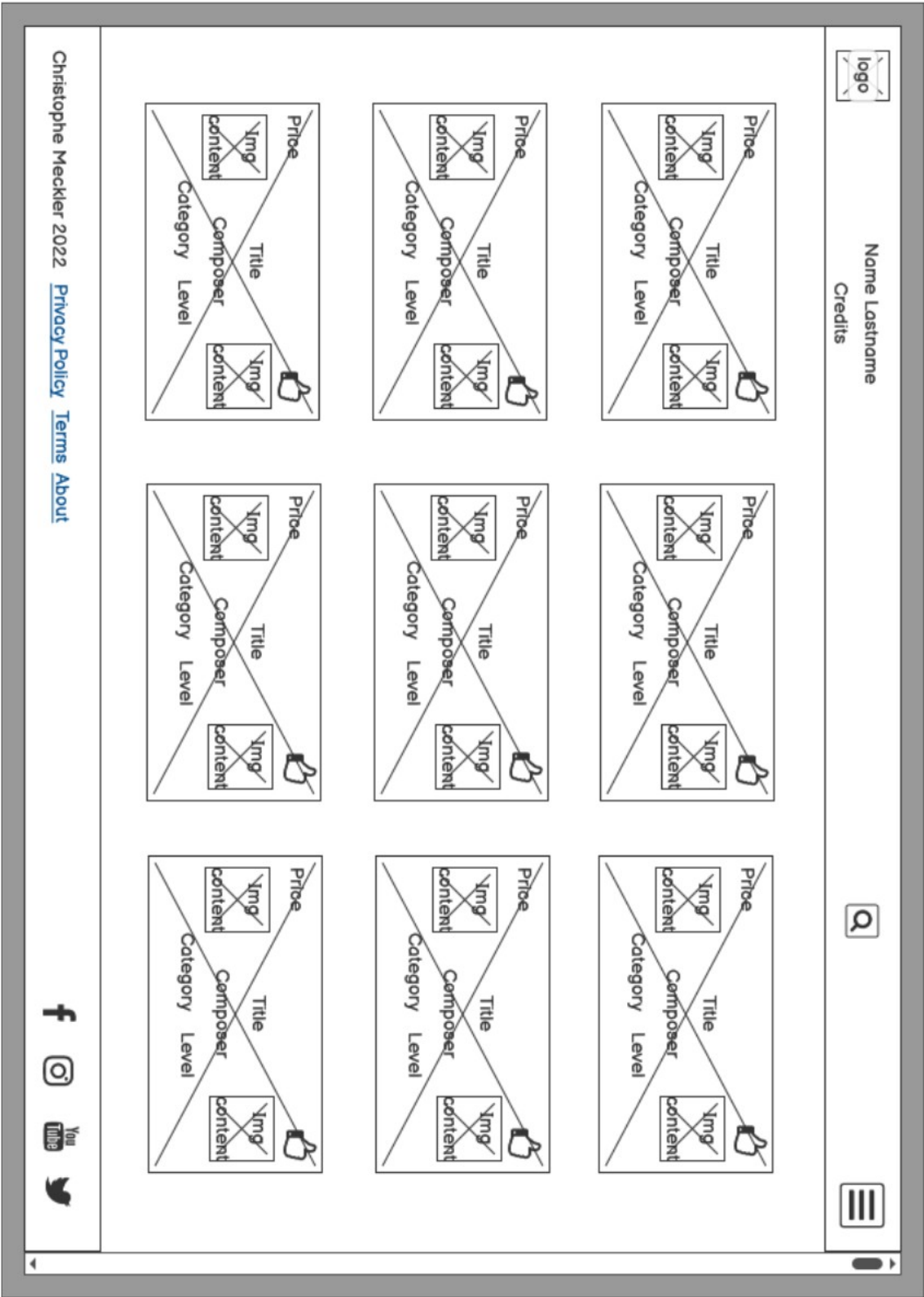
Accueil Version Desktop



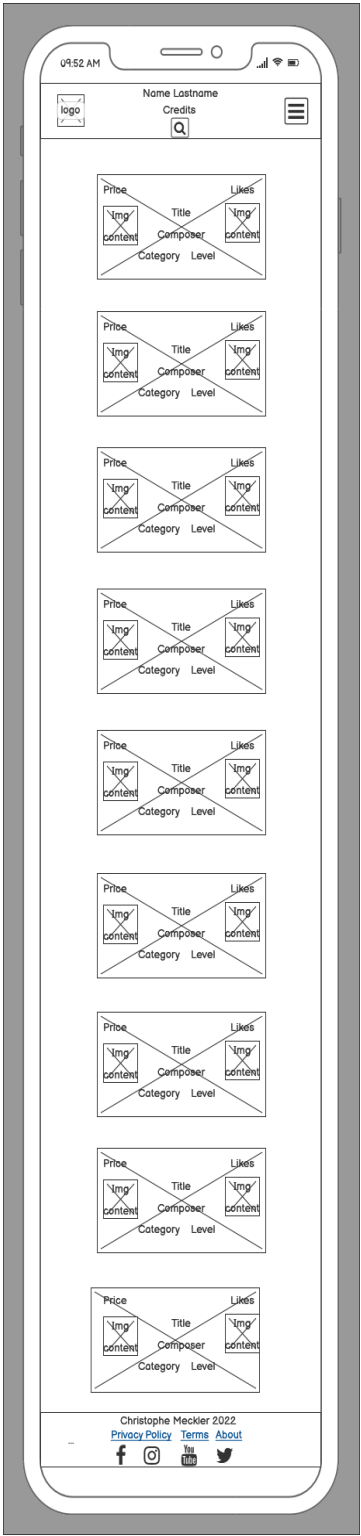
Retour p.7



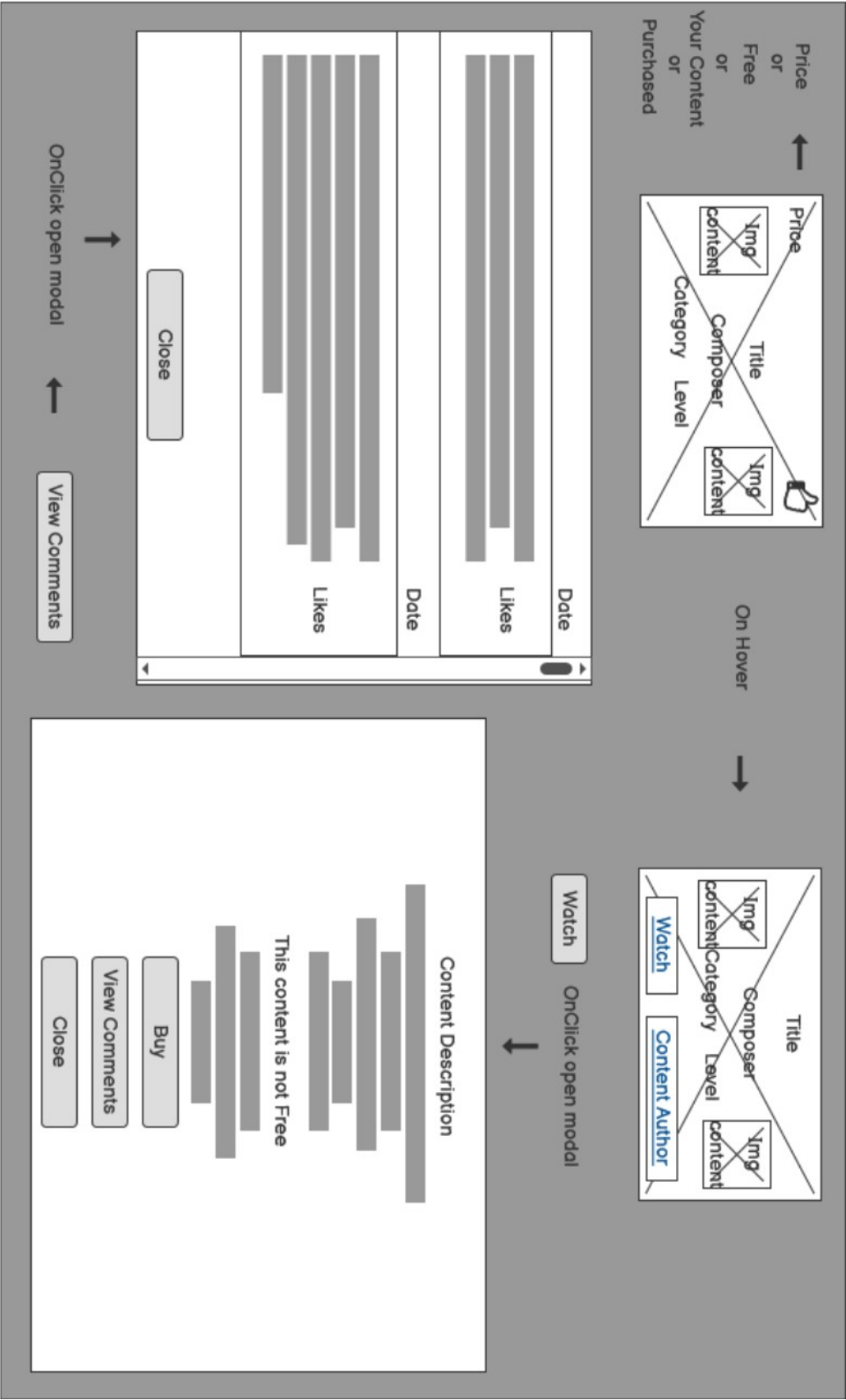
retour p.7



Retour p.8

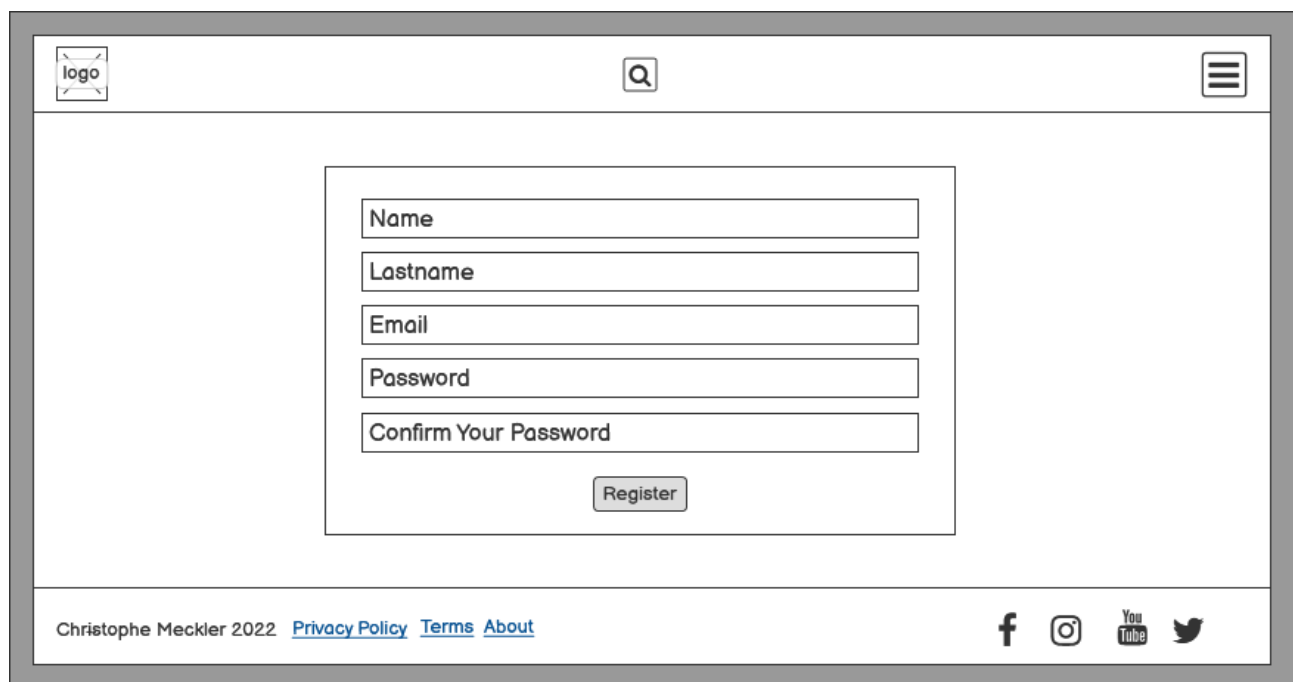


Retour p.8



Retourn p.8

S'inscrire



A registration form layout within a browser window. The header contains a 'logo' placeholder, a search icon, and a hamburger menu icon. The form itself is a centered box with five input fields: 'Name', 'Lastname', 'Email', 'Password', and 'Confirm Your Password'. Below these fields is a 'Register' button. The footer of the browser window shows the text 'Christophe Meckler 2022' followed by links for 'Privacy Policy', 'Terms', and 'About'. On the right side of the footer are social media icons for Facebook, Instagram, YouTube, and Twitter.

logo

Search

Name

Lastname

Email

Password

Confirm Your Password

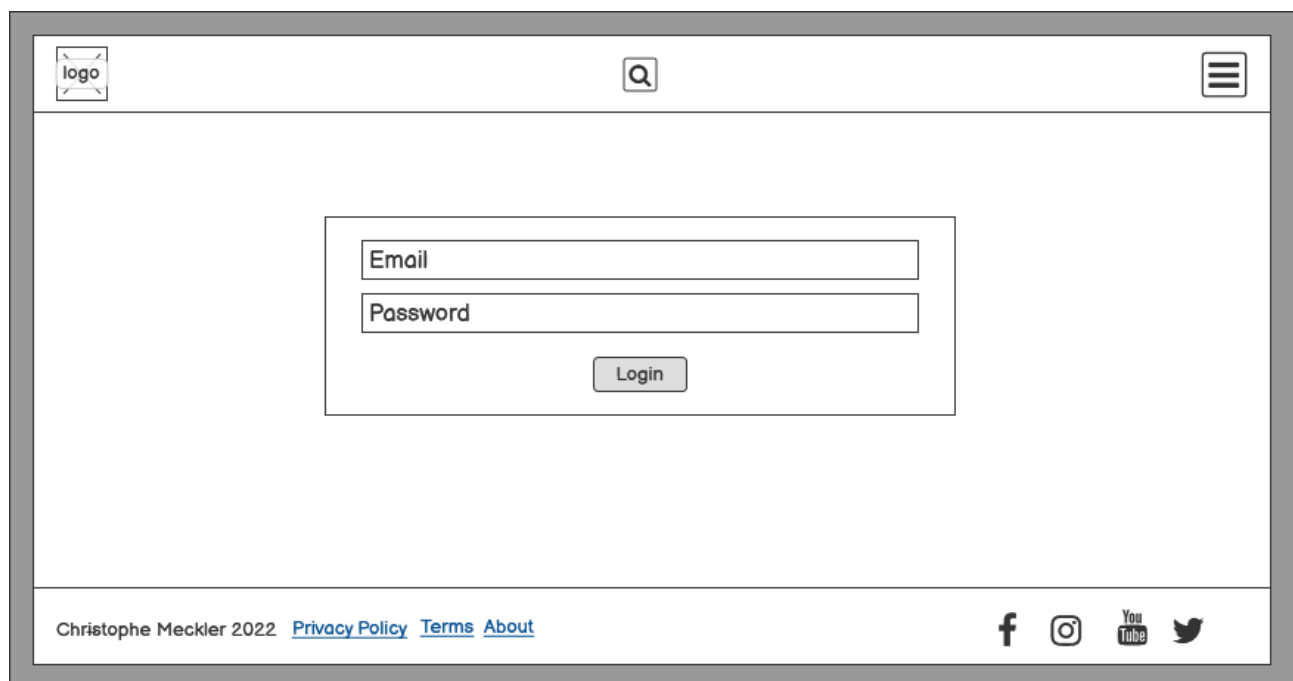
Register

Christophe Meckler 2022 [Privacy Policy](#) [Terms](#) [About](#)

f i YouTube

Retour p.9

Se Connecter



A login form layout within a browser window. The header is identical to the registration form, with a 'logo' placeholder, a search icon, and a hamburger menu icon. The form is a centered box with two input fields: 'Email' and 'Password'. Below these fields is a 'Login' button. The footer of the browser window is identical to the registration form, showing 'Christophe Meckler 2022' with links for 'Privacy Policy', 'Terms', and 'About', and social media icons for Facebook, Instagram, YouTube, and Twitter.

logo

Search

Email

Password

Login

Christophe Meckler 2022 [Privacy Policy](#) [Terms](#) [About](#)

f i YouTube

Retourn p.9

Sommaire p.3

56/65

logo

Name Lastname

Credits

🔍

☰

Email : email@email.com

Edit Your Name/LastName

Edit Your password

Add Content

Your Content

Purchased Content

Delete My Account

Notifications :

Date

Delete

Date

Delete

Date

Delete

Christophe Macker 2022

[Privacy Policy](#)

[Terms](#)

[About](#)

f

📷

YouTube

🐦

Delete

or

Delete My Account

OnClick open modal →

Logo

Are You Sure ?

Delete

Close

Unfold →

Email : email@email.com

Edit Your Name/LastName

Name

Lastname

Add content

Edit Your password

Old Password

New Password

Confirm New Password

Add content

Add Content

Title

Composer

Description

Category

Level

Upload file

no files

Price

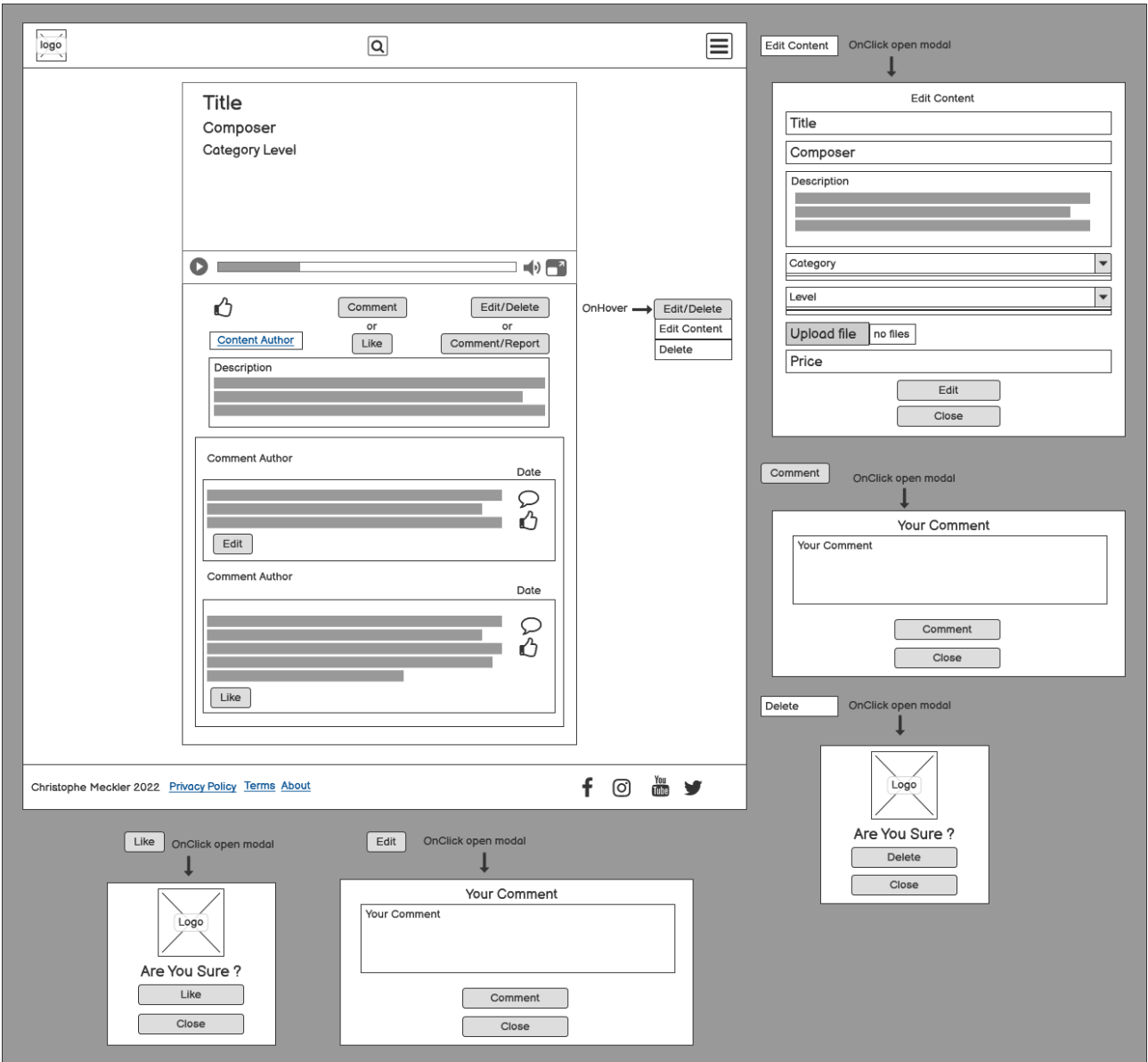
Add content

Your Content

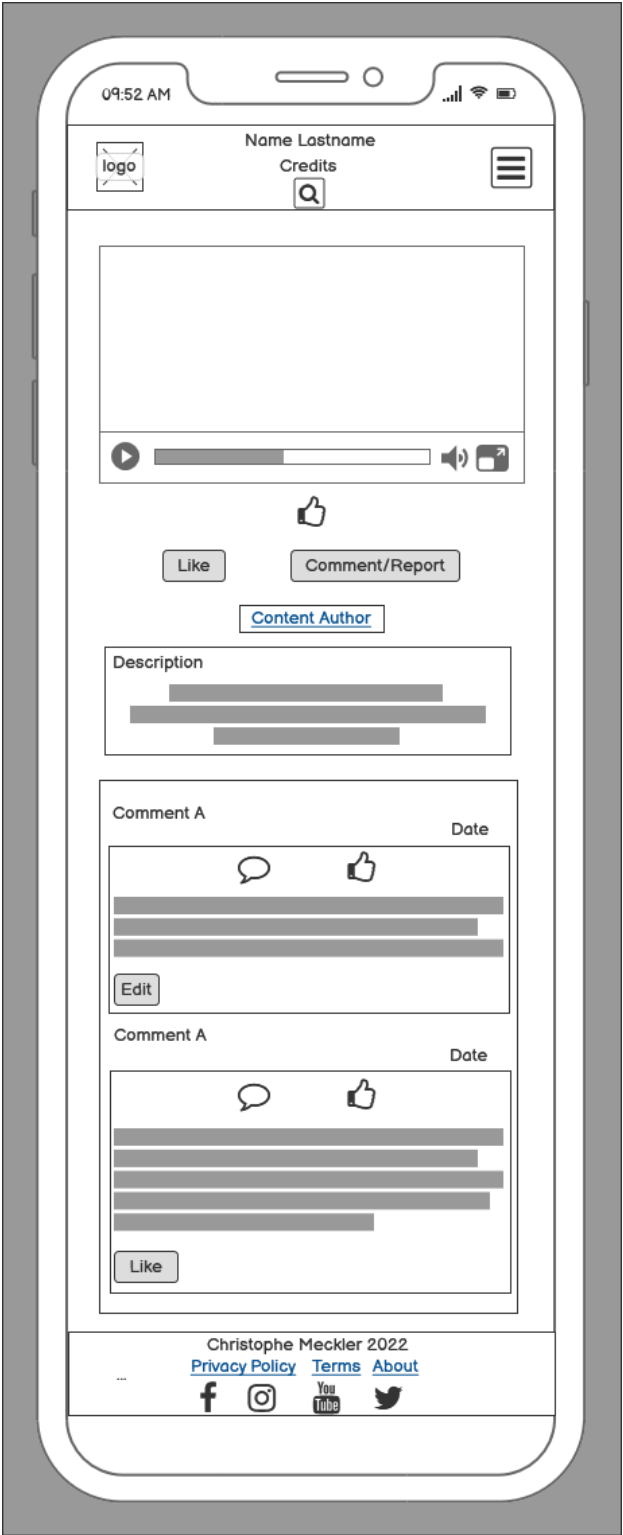
Purchased Content

Delete My Account

Retour p.10

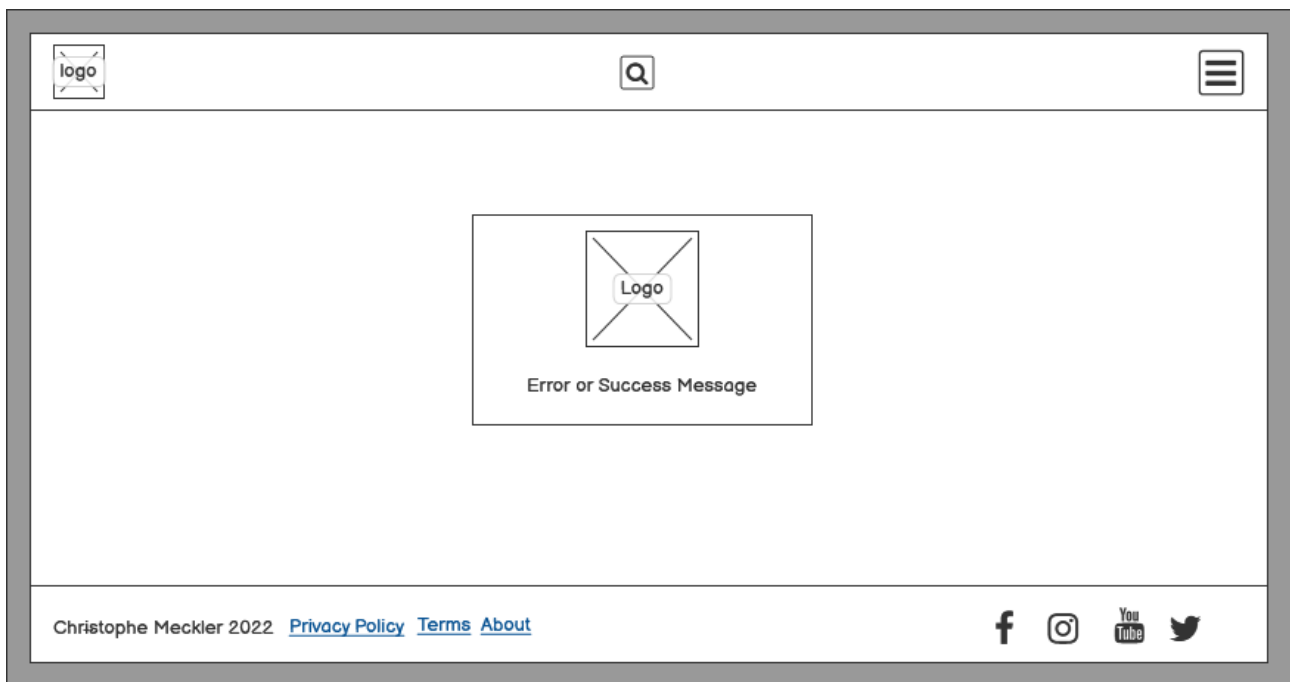


Retour p.11



Retour p.11

Modale de succès et d'erreurs



Retourn p.12

Script SQL

```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Hôte : 127.0.0.1:3306
-- Généré le : Lun. 03 oct. 2022 à 14:08
-- Version du serveur : 5.7.36
-- Version de PHP : 8.1.10

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données : `diplome`
```

```

--

-- -----

--
-- Structure de la table `comments`
--

DROP TABLE IF EXISTS `comments`;
CREATE TABLE IF NOT EXISTS `comments` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `comment` text NOT NULL,
  `date` text NOT NULL,
  `likes` int(11) NOT NULL DEFAULT '0',
  `id_contents` int(11) NOT NULL,
  `id_users` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `comments_id_contents_FK` (`id_contents`),
  KEY `comments_id_users_FK` (`id_users`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table `contact`
--

DROP TABLE IF EXISTS `contact`;
CREATE TABLE IF NOT EXISTS `contact` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `message` text NOT NULL,
  `date` text NOT NULL,
  `id_users` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `contact_id_users_FK` (`id_users`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;

-- -----

--
-- Structure de la table `contents`
--

DROP TABLE IF EXISTS `contents`;
CREATE TABLE IF NOT EXISTS `contents` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(255) NOT NULL,
  `content` varchar(255) NOT NULL,
  `price` int(11) NOT NULL,

```

```

`composer` varchar(255) NOT NULL,
`category` varchar(255) NOT NULL,
`level` varchar(255) NOT NULL,
`description` text NOT NULL,
`likes` int(11) NOT NULL DEFAULT '0',
`reporting` int(11) NOT NULL DEFAULT '0',
`id_users` int(11) NOT NULL,
PRIMARY KEY (`id`),
KEY `contents_id_users_FK` (`id_users`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;

```

```

--
-- Structure de la table `likes`
--

```

```

DROP TABLE IF EXISTS `likes`;
CREATE TABLE IF NOT EXISTS `likes` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_comments` int(11) DEFAULT NULL,
  `id_contents` int(11) DEFAULT NULL,
  `id_users` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `likes_id_users_FK` (`id_users`),
  KEY `likes_id_comments_FK` (`id_comments`),
  KEY `likes_id_contents_FK` (`id_contents`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8;

```

```

--
-- Structure de la table `notifications`
--

```

```

DROP TABLE IF EXISTS `notifications`;
CREATE TABLE IF NOT EXISTS `notifications` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `notification` text NOT NULL,
  `date` text NOT NULL,
  `id_users` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `notifications_id_users_FK` (`id_users`)
) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=utf8;

```

```

--
-- Structure de la table `purchased_contents`

```

```
--

DROP TABLE IF EXISTS `purchased_contents`;
CREATE TABLE IF NOT EXISTS `purchased_contents` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_contents` int(11) NOT NULL,
  `id_users` int(11) NOT NULL,
  `original_price` int(11) NOT NULL,
  `buyer_repayment` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  KEY `purchased_contents_id_contents_FK` (`id_contents`),
  KEY `purchased_contents_id_users_FK` (`id_users`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```

```
-- -----

--
-- Structure de la table `reporting`
--
```

```
DROP TABLE IF EXISTS `reporting`;
CREATE TABLE IF NOT EXISTS `reporting` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `message` text NOT NULL,
  `date` text NOT NULL,
  `id_users` int(11) NOT NULL,
  `id_contents` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `reporting_id_users_FK` (`id_users`),
  KEY `reporting_id_contents_FK` (`id_contents`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

```
-- -----

--
-- Structure de la table `users`
--
```

```
DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `lastname` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `type` varchar(255) NOT NULL DEFAULT 'user',
  `credits` int(11) NOT NULL DEFAULT '500',
  `time` int(11) NOT NULL,
  PRIMARY KEY (`id`)
```

```

) ENGINE=InnoDB AUTO_INCREMENT=39 DEFAULT CHARSET=utf8;

--
-- Contraintes pour les tables déchargées
--

--
-- Contraintes pour la table `comments`
--
ALTER TABLE `comments`
  ADD CONSTRAINT `comments_id_contents_FK` FOREIGN KEY (`id_contents`) REFERENCES `contents` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `comments_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES `users` (`id`) ON DELETE CASCADE;

--
-- Contraintes pour la table `contact`
--
ALTER TABLE `contact`
  ADD CONSTRAINT `contact_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES `users` (`id`)
ON DELETE CASCADE;

--
-- Contraintes pour la table `contents`
--
ALTER TABLE `contents`
  ADD CONSTRAINT `contents_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES `users` (`id`) ON DELETE CASCADE;

--
-- Contraintes pour la table `likes`
--
ALTER TABLE `likes`
  ADD CONSTRAINT `likes_id_comments_FK` FOREIGN KEY (`id_comments`) REFERENCES `comments` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `likes_id_contents_FK` FOREIGN KEY (`id_contents`) REFERENCES `contents` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `likes_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES `users` (`id`)
ON DELETE CASCADE;

--
-- Contraintes pour la table `notifications`
--
ALTER TABLE `notifications`
  ADD CONSTRAINT `notifications_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES `users` (`id`) ON DELETE CASCADE;

--
-- Contraintes pour la table `purchased_contents`

```



```
--
ALTER TABLE `purchased_contents`
  ADD CONSTRAINT `purchased_contents_id_contents_FK` FOREIGN KEY (`id_contents`)
REFERENCES `contents` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `purchased_contents_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES
`users` (`id`) ON DELETE CASCADE;

--
-- Contraintes pour la table `reporting`
--
ALTER TABLE `reporting`
  ADD CONSTRAINT `reporting_id_contents_FK` FOREIGN KEY (`id_contents`) REFERENCES
`contents` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `reporting_id_users_FK` FOREIGN KEY (`id_users`) REFERENCES `users`
(`id`) ON DELETE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Retour p.21