

TP – Développeur Web et Web Mobile

Énoncé

Durée indicative : 70 h

Documents autorisés : annexes

Matériel autorisé : calculatrice, tableur, etc.



Votre examen comporte :

- ✓ Cet **énoncé** qui vous présente le sujet de l'épreuve
- ✓ Une **copie à rendre** (Excel ou Word) que vous devez télécharger, remplir informatiquement et déposer dans l'espace de dépôt prévu à cet effet.

Renommer votre copie à rendre Word ou Excel comme suit :

TP_DWWM_Nov/Déc24/Janv/Fév25_copiearendre_**NOM_Prenom**



Objectifs de l'évaluation :

L'évaluation en cours de formation que vous allez réaliser a pour vocation de figurer dans votre **livret d'évaluation**. Il sera donc remis à votre jury le jour des épreuves du titre professionnel accompagné de votre évaluation et du sujet initial.



Nous vous demandons de vous mettre en situation d'examen. Prenez le temps de lire le sujet et de manipuler les annexes afin de répondre en situation professionnelle aux questions et problématiques évoquées dans le sujet.

À vous de jouer !

Compétences du référentiel concernées

Toutes les compétences du titre professionnel « Développeur Web et Web mobile » seront analysées grâce à ce projet.

Activité – Type 1 : 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 une interface utilisateur web dynamique
- ☒ Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Activité – Type 2 : 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 ou e-commerce

Présentation de l'entreprise

Gamestore est une entreprise spécialisée dans la vente de jeu vidéo. Ils ont 5 magasins en France à Nantes, Lille, Bordeaux, Paris ainsi que Toulouse.

Ils proposent des jeux vidéo sur toutes les plateformes connues à ce jour, mais ils n'ont pas de site web. Pour le moment, seuls des prospectus sont effectués. Ce mode de fonctionnement est dépassé et Gamestore commence à perdre des clients. Ils ont besoin d'avoir accès à un outil leur donnant de la visibilité et permettant la vente en ligne des jeux vidéo.

Vous avez été diplômé par l'école Studi. De plus, l'école ayant vanté vos compétences, vous avez été pris en freelance pour réaliser cette application.

Description du projet

Gamestore souhaite améliorer sa gestion. Dans cette optique, l'entreprise a besoin d'une application lui permettant de :

- ☒ Développer sa visibilité
- ☒ Proposer de la vente en ligne (cependant, le paiement ainsi que le retrait se fera dans une agence)

Pour ce faire, José, gérant de Gamestore, a décidé de vous prendre à part afin de vous présenter tous les éléments.

US 1 : Page d'accueil

→ *Utilisateur concerné : Visiteur*

La page d'accueil doit comporter :

- ☒ Un descriptif de l'entreprise
- ☒ Une image en fond
- ☒ Les derniers jeux vidéo proposés ainsi que les promotions actuelles.

US 2 : Menu de l'application

→ *Utilisateur concerné : Visiteur*

Le menu est essentiel à l'application, il permet de faciliter la navigation et de fluidifier le trafic. Ainsi, il doit composer au minimum :

- ☒ Retour vers la page d'accueil
- ☒ Accéder à une vue globale de tous les jeux vidéo
- ☒ Connexion ou « mon espace » si l'utilisateur est connecté

US 3 : vue globale de tous les jeux vidéo

→ *Utilisateur concerné : Visiteur*

Une vue globale est nécessaire afin de proposer une interface simple et récapitulative de tous les jeux vidéo. L'enjeu est de présenter tous les jeux vidéo ; pour chacun doit apparaître son nom, son image, le PEGI, ainsi que son prix. Au clic sur le jeu vidéo, l'utilisateur est redirigé vers la page détaillée. Des filtres doivent être positionnés afin de filtrer (l'utilisation de filtre ne doit pas recharger la page) :

- ☒ Par prix
- ☒ Par genre (RPG, MMO, Action, etc.)

US 4 : vue détaillée du jeu vidéo

→ *Utilisateur concerné : Visiteur*

La vue détaillée d'un jeu vidéo permet à l'utilisateur de pouvoir avoir accès aux informations détaillées qui ne sont pas visibles dans la vue globale. Toutes les informations d'un jeu vidéo doivent être affichées. Ces informations vous seront explicitées dans une US suivante.

US 5 : ajout d'un jeu vidéo au panier

→ *Utilisateur concerné : Utilisateur*

Un visiteur peut, depuis la vue globale ou la vue détaillée, ajouter un jeu vidéo à son panier. Cependant, le clic sur le bouton est possible uniquement si le visiteur est authentifié (donc devenu utilisateur) et que le jeu est en stock !

US 6 : Panier

→ *Utilisateur concerné : Utilisateur*

Le panier est accessible depuis un bouton mis en évidence au sein de l'interface de l'utilisateur. Il devra être visible tant que le panier dispose d'au moins un article.

Un Utilisateur, au clic sur le panier, doit pouvoir visualiser les articles qui ont été positionnés.

Le panier devra comporter :

- ☒ Le libellé de chaque produit
- ☒ Le prix de chaque produit
- ☒ Le prix total
- ☒ L'agence de retrait : par défaut, il s'agit de l'agence la plus proche du client, mais cela est modifiable par lui-même.
- ☒ Date de retrait : l'utilisateur met la date qu'il souhaite, mais elle doit :
 - Être à moins de 7 jours après la commande
 - Ne doit pas être un lundi ou un dimanche (les agences sont fermées)

Une fois le panier validé, la commande est ajoutée dans l'historique des commandes de l'utilisateur et il reçoit un mail de confirmation.

US 7 : Un jeu vidéo

→ *Utilisateur concerné : Visiteur*

Un jeu vidéo est caractérisé par :

- ☒ Un libellé
- ☒ Une description
- ☒ Un PEGI
- ☒ Un genre (action, MMO, RPG, etc.)
- ☒ Un prix
- ☒ Une quantité : mise à jour par l'Admin
- ☒ Une image

US 8 : Espace Utilisateur

→ *Utilisateur concerné : Utilisateur*

Un utilisateur peut avoir accès à un bouton « Mon espace » depuis le menu principal.

Cela lui donne accès à plusieurs fonctionnalités :

- ☒ Historique des commandes avec son statut (VALIDÉ quand la commande a été confirmée, mais non encore retirée).
- ☒ Modification de ses informations personnelles.

US 9 : Création d'un compte utilisateur

→ *Utilisateur concerné : Visiteur*

Un Visiteur peut se créer un compte. Pour cela, il devra fournir :

- ☒ Un nom ainsi qu'un prénom
- ☒ Une adresse électronique et postale
- ☒ Un mot de passe

Une fois que le compte a été créé, il reçoit un mail de confirmation (ce mail ne comporte pas le mot de passe ainsi que les informations personnelles de l'utilisateur).

US 10 : mot de passe oublié

→ *Utilisateur concerné : Utilisateur*

Un utilisateur qui a oublié son mot de passe peut cliquer sur le bouton « Mot de passe oublié » en dessous des champs prévus à l'insertion du mot de passe/mail. Il devra juste fournir un mail, ensuite, il va recevoir un mail avec un lien cliquable. Ce lien devra rediriger l'utilisateur vers une interface de l'application web lui permettant de redéfinir le mot de passe.

US 11 : Espace Administrateur

→ *Utilisateur concerné : Admin*

Un Admin, une fois connecté et sur son espace, peut s'occuper :

- ☒ De la création d'un jeu vidéo
- ☒ De la gestion des stocks
 - Il peut ajouter du stock à un jeu vidéo
- ☒ Du *Dashboard* récapitulatif des ventes

- Le Dashboard est un graphique affichant les ventes de jeu vidéo par genre (en ordonnée), en fonction de la date (en abscisse)
- Pour qu'une commande soit prise en compte dans le *Dashboard*, elle doit être au statut « LIVRÉ »
- ☒ De la création d'un compte employé
 - L'admin devra fournir un mail ainsi qu'un mot de passe.

US 12 : Espace Employé

→ *Utilisateur concerné : Employé*

Un employé doit avoir son compte créé par un Admin. L'employé dispose d'un seul droit, qui est de passer les commandes du statut « VALIDÉ » à « LIVRÉ » une fois que le client est passé à l'agence pour payer et retirer sa commande.

US 13 : Calcul des ventes

→ *Utilisateur concerné : Utilisateurs*

Lorsqu'un utilisateur commande des articles et que le client passe le statut de la commande à « LIVRÉ » alors, il faut incrémenter le ou les ventes d'articles afin de les afficher plus tard. Cette donnée sera enregistrée dans une base de données NoSQL afin de fluidifier les requêtes ainsi que les informations. Vous devrez stocker : le nom de l'article, le prix ainsi que la date.

US 14 : Affichage des ventes

→ *Utilisateur concerné : Employé et Admin*

Les employés et admin veulent avoir, dans leurs espaces, une vue sur les données collectées lors des ventes. Pour ce faire, ils désirent un graphique affichant par date le nombre de ventes par article.

Au clic sur un bouton « Détails », ils peuvent avoir accès à un tableau récapitulant toutes les ventes en affichant : date, nom de l'article, nombre de ventes dans la journée, prix totaux. Également, un filtre

Stack technique

Aucune technologie n'est obligatoire pour cet ECF, à l'exception de la base de données qui doit être relationnelle.

Voici un exemple de stack technique possible :

- *Front* : HTML 5, CSS (Bootstrap), JS
- *Back-end* : PHP avec utilisation de PDO
- Base de données relationnelle : MySQL ou MariaDB
- Base de données NoSQL : MongoDB
- Déploiement : fly.io

Objectif & Livrables

L'objectif de cet ECF est que vous passiez sur chaque étape nécessaire à l'élaboration d'une solution de qualité : analyse des besoins, maquettage, intégration, développement des règles de gestion, ainsi que déploiement.

Vous devrez détailler dans votre copie les dispositions que vous avez prises notamment sur la sécurité aussi bien *front*, *back* que l'intégrité de vos données en base. De plus, vous devrez justifier tous vos choix techniques : il est très important de pouvoir les justifier et pouvoir exposer votre point de vue sur différentes technologies.

Les livrables de ce sujet sont les suivants :

- ☒ Le lien du (ou des) dépôt(s) github **PUBLIC** où sera présent le code de votre application
- ☒ Le lien de votre application déployée
- ☒ Le lien vers votre logiciel de gestion de projet (Jira, Notion, Trello, etc.)

Votre git devra comporter :

- Un fichier README.md contenant la démarche à suivre afin de déployer votre application en local
- Les bonnes pratiques de git qui doivent être appliquées
 - Une branche principale
 - Une branche de développement

- Chaque fonctionnalité sera une branche issue de la branche développement, après test, le merge sera effectué vers la branche développement.
 - Une fois que la branche développement est correctement testée, il faudra effectuer un merge vers la branche principale.
- Les fichiers de création de bases de données, mais également d'intégration de données (cela doit être un fichier sql). Une fixture ou encore une migration ne justifie pas que vous maîtrisez le SQL.
- Manuel d'utilisation en format PDF
 - Il doit présenter l'application et donner des identifiants afin de réaliser les différents parcours possibles.
- Une charte graphique en format PDF
 - Palette de couleurs utilisée ainsi que la police
 - L'export des maquettes attendues (*wireframes & mockup*)
- Une documentation traitant de votre gestion de projet
 - Explication de votre gestion de projet
- Une documentation technique de votre application
 - Réflexions initiales technologiques sur le sujet
 - Configuration de votre environnement de travail
 - Modèle conceptuel de données (ou diagramme de classe)
 - Diagramme d'utilisation ainsi que diagramme de séquence
 - Détail du déploiement de votre application.

Votre gestion de projet devra être sous la forme d'un Kanban partagé afin d'avoir une vue globale de votre réalisation. Il devra comprendre :

- Une colonne recensant toutes les fonctionnalités prévues et ordonnées par priorité
- Une colonne recensant les fonctionnalités que vous comptez faire, prévu de développement (ou dans le *sprint* si vous êtes en agile)
- Une colonne qui recense les fonctionnalités que vous déployez actuellement
- Une colonne qui énumère les fonctionnalités terminées (sur la branche de développement)
- Enfin, vous pouvez créer une dernière colonne stipulant les fonctionnalités qui ont été merge dans la branche principale.