



Cahier des charges ClashCode

Table des matières

I. Cahier des charges fonctionnel	4
Objectifs.....	4
Les objectifs du site	4
Vos objectifs	4
Niveaux d'exigences	5
Description des pages.....	6
Header (★★★).....	6
Footer (★★★).....	6
Page d'accueil (★★★).....	6
Page de connexion (★★★)	7
Page d'inscription (★★★).....	7
Page de profil privée (★★★)	8
Page de profil publique.....	8
Page de classement (★★)	9
Page des défis (★★★).....	9
Page de résolution de défis	9
Page administrateur (★★★).....	10
Page des équipes (★★)	10
Fonctionnalités liées à l'application.....	11
Système d'équipes (★★).....	11
Mot de passe oubliés (★).....	12
Photo de profil (★).....	12
Photo d'équipe (★).....	12
User Stories	13
Objectif global	13
Utilisateur non connecté (Invité).....	13
Utilisateur connecté	13
Classements et communauté	14
Système d'équipes	14
Administrateur.....	15
Sécurité et maintenance.....	15

Expérience utilisateur et interface	15
II. Cahier des charges techniques	16
Interface utilisateur / Frontend	16
Serveur / Backend	16
Base de données	17
Sécurité.....	17
Tests.....	18
Tests unitaires (★★★).....	18
Tests de performance (★★)	18
Tests d'interface (★).....	18
Conteneurisation	18
Images	18
Orchestration des conteneurs	18
Configuration Docker :	18
Outil de suivi de projet	19
III. Charte graphique	20
Couleurs.....	20
Polices d'écriture	20
Logo	21
IV. Modalités des livrables, mode de communication et documentation	
21	
Mode de communication	21
Livrables.....	22
Documentation.....	22

I. Cahier des charges fonctionnel

Objectifs

Cette section présente les attentes et les finalités de l'application **ClashCode**. Elle a pour but de définir nos objectifs pour la plateforme, aussi bien du point de vue des utilisateurs que pour les développeurs.

L'enjeu est de cadrer les fonctionnalités principales, l'expérience utilisateur recherchée et la valeur ajoutée attendue, afin de fournir une vision précise et partagée de ce que devra offrir l'application.

Les objectifs du site

Le site a pour objectif principal de proposer une **plateforme d'apprentissage interactif à travers des défis de programmation**.

Il permet aux utilisateurs, débutants ou confirmés, de développer et d'améliorer leurs compétences techniques dans différents langages de programmation grâce à une approche ludique.

Le site vise à :

- **Faciliter l'apprentissage** en réalisant des exercices de programmation ludique adapté à différents niveaux et différents langages.
- **Motiver les utilisateurs** par un système de points et de classement entre les utilisateurs et groupe, favorisant la progression de chacun.
- **Évaluer les compétences** en proposant des feedbacks à leurs réalisations pour mesurer les acquis.
- **Créer une communauté d'apprenants** où les utilisateurs peuvent s'unir pour progresser et s'entraider.
- **Offrir un suivi personnalisé** grâce à l'historique des scores et à la visualisation des progrès réalisés.

En résumé, ce site sert à **apprendre en pratiquant**, il rend l'expérience d'apprentissage plus **interactive, compétitive et motivante**.

Vos objectifs

Vous devrez concevoir une application durable et facilement maintenable, offrant une interface intuitive pour les utilisateurs. Le code devra être modulaire, évolutif et l'application devra s'adapter à tous types d'appareils (responsive).

L'application devra répondre à ces différents points cruciaux :

- La sécurité
- La performance
- La qualité du code

Elle devra être développée selon les bonnes pratiques de conception logicielle et respecter les standards modernes du web.

Les objectifs techniques et qualitatifs incluent notamment :

- Une documentation claire et détaillées
- Une expérience utilisateur fluide et cohérente
- Une gestion efficace des erreurs et une interface réactive

Niveaux d'exigences

Afin de hiérarchiser les fonctionnalités demandées, un système de priorités est défini :

- **★★★ : Exigence obligatoire** = Fonctionnalité indispensable au bon fonctionnement du site. Elle doit impérativement être réalisée.
- **★★ : Exigence importante** = Fonctionnalité fortement attendue, améliorant l'expérience utilisateur. Elle doit être réalisée si possible dans les délais impartis.
- **★ : Exigence optionnelle** = Fonctionnalité supplémentaire ou "bonus". Sa réalisation est appréciée, mais non prioritaire.

Ce système permet de distinguer les éléments essentiels de ceux qui constituent des compléments.

Description des pages

Header (★★★)

Le header de l'application est présent sur toutes les pages. Il permet aux utilisateurs de **naviguer dans l'application** via des boutons qui pourront disparaître selon le contexte de l'utilisateur (ex : bouton de connexion visible si l'utilisateur n'est pas connecté).

Il comportera :

- **Le logo de l'application** : Devra être cliquable pour rediriger vers la page d'accueil.
- **Avatar utilisateur** : Menu déroulant qui permettra à l'utilisateur d'accéder à son profil privé, un bouton de déconnexion et (si administrateur) à la page d'administration de l'application.
- Ainsi que les différents boutons pour naviguer (**Inscription, Connexion, Classement, Défis, Équipes**)

Footer (★★★)

Le footer de l'application est présent sur toutes les pages. Il permet aux utilisateurs d'**accéder à des raccourcis**, tel que les **mentions légales** ou les **conditions générales d'utilisation** (il n'est pas demandé de rédiger les conditions générales d'utilisation ou des mentions légales, vous pouvez utiliser des pages place-holder [page avec un texte type *Lorem ipsum*]).

Il doit aussi inclure la mention suivante : “© ClashCode – 2026 – Tous droits réservés”

Le logo de l'application doit aussi être présent, et cliquable, comme dans le header.

Page d'accueil (★★★)

La page d'accueil est le point d'entrée principal de la plateforme. Elle doit être attractive pour les utilisateurs, claire et donner accès aux éléments essentiels du site.

Elle comportera les sections suivantes :

- **Bannière d'introduction** :
Elle contiendra une **description claire** et rapide qui donnera envie aux utilisateurs de s'inscrire via un **bouton d'appel à l'action** (CTA).
- **Statistiques et graphiques** :
Cette section contiendra des statistiques et graphiques intéressants sur les **diverses activités de l'application**, tels que le nombre de défis réalisés un jour, ou bien le langage de programmation préféré des utilisateurs.
- **Avis utilisateurs** :
L'avis des utilisateurs est crucial, c'est pourquoi une section avec **6 avis aléatoire** devra être présente pour donner plus de vie à l'application.
- **Défis populaires** :

Cela contiendra les **6 défis les plus populaires** actuellement pour donner envie aux utilisateurs de tenter ces défis.

▪ **Nouveaux défis :**

De la même façon que les défis populaires, cette section affichera **les 6 défis les plus récents**.

Lorsque l'utilisateur est connecté, cette page ne doit pas afficher la bannière d'introduction.

Page de connexion (★★★)

La page de connexion permet d'accéder à son compte, une fois connecté, elle renvoie vers la page des défis.

La page de connexion contient un formulaire avec :

- Un champ pseudonyme/courriel *
- Un champ mot de passe (avec possibilité de montrer/cacher le mot de passe) *
- Un bouton se connecter (les champs avec une étoile rouge * doivent être remplis par l'utilisateur pour que le bouton soit cliquable)

Elle doit aussi intégrer, de manière moins prioritaire :

- Un bouton de réinitialisation de mot de passe
- Une possibilité de se connecter via compte Google ou Microsoft (d'autres moyens de connexion seront valorisés)

Page d'inscription (★★★)

La page d'inscription permet à l'utilisateur de se créer un compte sur l'application.

Elle doit inclure :

- Un champ pseudonyme *
- Un champ courriel *
- Un champ mot de passe*
- Un bouton (les champs avec une étoile rouge * doivent être remplis par l'utilisateur pour que le bouton soit cliquable)

Le mot de passe doit comporter 12 caractères minimum, des variations entre les majuscules et minuscules, des chiffres et des caractères spéciaux.

Elle doit aussi intégrer, de manière moins prioritaire :

- **Champ de vérification** de mot de passe (demander à l'utilisateur de réécrire son mot de passe)
- La possibilité de s'inscrire via un compte Google ou Microsoft (d'autres moyens de connexion seront valorisés)

Page de profil privée (★★★)

Page listant les informations de l'utilisateur. Cette page doit permettre à l'utilisateur de pouvoir les modifier. Dans le cas d'un changement de mot de passe, l'envoi d'un courriel de réinitialisation serait apprécié. Voici les informations à afficher :

- **Adresse courriel**
- **Mot de passe** (Vous devrez utiliser un **placeholder**, tel que ●●●●●●●●)
- **Image de profil** (avatar) de l'utilisateur
- **Pseudonyme**
- **Description**

Vous pouvez aussi afficher des informations non modifiables par l'utilisateur, tel que :

- **Le score de l'utilisateur**

Cette page est privée pour l'utilisateur, et ne peut pas être accessible à une personne externe.

Page de profil publique (★★)

La page de profil publique permet à tout utilisateur de consulter les informations non sensibles d'un autre utilisateur.

Voici les différentes informations à afficher :

- **Le pseudo**
- **Sa photo de profil** (si l'utilisateur n'a pas défini de photo de profil, un template sera alors ajouté).
- **La description**
- **Son score ainsi que son rang actuel**
- **Les défis qu'il a réalisés** (Il sera possible de tenter un des défis de cette liste)
- **L'équipe à laquelle il appartient** (Ou un texte par défaut expliquant que le joueur n'appartient à aucune une équipe)

Le style graphique est laissé à votre imagination.

Page de classement (★★)

La page de classement permettra de visualiser les performances des utilisateurs. Elle affichera une liste dynamique d'utilisateurs selon leur score total et leur rang actuel.

Cette page affichera :

- **Mise en avant du top 3** : style graphique au choix, exemple : un podium avec la photo de profil des 3 meilleurs utilisateurs.
- **Un classement général des utilisateurs** :
 - Le **rang** (position dans le classement)
 - Le **pseudo** de l'utilisateur (avec une redirection vers son profil public)
 - Le **score total**
 - Le **nombre de défis réalisés**
- **Barre de recherche** permettant de retrouver rapidement un utilisateur à partir de son pseudonyme ou de son code ami.
- **Classement par langage** : possibilité d'afficher le classement uniquement des utilisateurs ayant réalisé des défis dans un langage spécifique (ex : JavaScript, Python...).

Sur cette page, il est possible d'ajouter le classement des équipes correspondant à la fonctionnalité "Système d'équipes".

Page des défis (★★★)

La page des défis affiche à l'utilisateur les différents défis possibles, il est possible de les trier et de filtrer selon différents paramètres, tel que :

- Le langage de programmation
- La difficulté
- La date de sortie

Cette page doit inclure un champ de recherche, permettant de trouver différents défis via une recherche textuelle.

Il y aura la possibilité d'afficher une courte description pour chaque défi avant de lancer un défi.

Page de résolution de défis (★★★)

Chaque défi aura, un nom, une catégorie, une difficulté (basé sur un niveau BUT2), une courte description, une description longue, un flag. Les catégories possibles seront les suivantes :

- Développement
- Réseau

- Pentest
- Base de données

La courte description permettra à l'utilisateur d'avoir un aperçu rapide du contenu du défi. La description longue sera disponible seulement quand le défi est lancé. On retrouvera dedans les informations nécessaires pour faire le défi.

Par mesure de sécurité, il n'y aura pas de défi avec de l'exécution côté serveur. Tous les défis sont donc à réaliser en local par l'utilisateur. Cela pourra se faire en demandant à l'utilisateur d'exécuter du code, de télécharger des ressources depuis le site, ou depuis l'extérieur, par exemple une image docker du docker hub. Il est demandé que les ressources soient clairement visibles dans la description. La solution finale de tous les défis est un flag à valider sur le site web. Vous trouverez accompagné de ce PDF, une archive avec les fichiers pour les différents défis.

Une fois un défi lancé, un chrono se lancera, il ne s'arrêtera pas à moins que l'utilisateur trouve le bon flag. En cas de difficulté, l'utilisateur aura la possibilité d'afficher un indice. Cet indice viendra à un coup, payé par le score en cas de réussite du défi.

Ce score sera sur 3 étoiles, 1 étoile sera obtenue pour chacun des points suivants :

- Le défi est résolu
- Le défi est réussi en moins de 3 essais
- Le défi est résolu en moins de x minutes (ce paramètre sera différent de défis en défis)

Lorsqu'un indice est utilisé, le score sera plafonné à 1 étoile.

Page administrateur (★★★)

La page administrateur permet de pouvoir gérer le site sur le long terme. Cette page doit pouvoir être utilisée par un profil qui n'est pas technique.

Les différentes fonctionnalités attendues de cette page sont :

- Possibilité de gérer les défis (création, mise à jour, suppression)
- Possibilité de gérer les utilisateurs (mise à jour, suppression)

Si un système de journalisation (log) a été mis en place, il serait intéressant de pouvoir visionner les logs depuis l'interface d'administration.

Page des équipes (★★)

La page des équipes est directement liée à la fonctionnalité du système d'équipes présentée plus loin dans le cahier des charges.

Elle a pour objectif de permettre aux utilisateurs de consulter, créer ou gérer des équipes de manière simple.

Elle contiendra :

- **La liste des équipes** : cette section permettra d'afficher l'ensemble des équipes (non classées) existantes sur l'application, d'indiquer celles qui disposent encore de places disponibles, et

d'accéder directement à leur page de profil. Si vous le souhaitez, il sera possible de rajouter un système de filtrage d'équipe en fonction de critères que vous pourrez déterminer.

- **Un bouton pour créer son équipe** : Cela permettra de faire apparaître un formulaire pour créer une nouvelle équipe, ce formulaire contiendra :
 - Le nom de l'équipe *
 - Sa description *
 - Photo de l'équipe (**Fonctionnalité facultative**)
- **Un bouton d'accès à son équipe** : Cela permettra d'accéder à une page dédiée à l'équipe de l'utilisateur. Celle-ci permettra de voir les différentes informations de l'équipe, tels que :
 - Son nom
 - Sa photo d'équipe (**Fonctionnalité facultative**)
 - Ses membres ainsi que leurs contributions
 - Ses points accumulés

Un utilisateur aura aussi la possibilité de **quitter son équipe**.

Un créateur d'une équipe aura la possibilité de **supprimer son équipe** s'il le souhaite. Mais il pourra aussi **modifier** le nom, la description ou bien la photo.

Fonctionnalités liées à l'application

Système d'équipes (★★)

Comme nous l'avons expliqué plus haut, afin d'innover et de se démarquer des autres plateformes de défis de programmation, nous souhaitons intégrer un système d'équipes au sein de l'application. Les utilisateurs auront la possibilité de :

- **Création et adhésion à une équipe** : créer une nouvelle équipe ou rejoindre une équipe existante.
- **Contribution au score collectif** : ajouter automatiquement leurs points individuels au score global de l'équipe lorsqu'un défi est réussi, valorisant ainsi l'effort personnel et collectif.
- **Classement des équipes** : figurer dans un tableau de classement dédié aux équipes, distinct du classement individuel, renforçant la compétition entre groupes.
- **Espace dédié à l'équipe** : accéder à une page spécifique présentant le nom, la description, le score, le rang au classement et la liste des membres avec leurs contributions respectives.
- **Personnalisation d'équipe** : configurer l'identité de l'équipe (nom, description, icône ou image représentative). Celles-ci possèdent une limite maximale de 15 membres par équipe.

À terme, ce système pourra être enrichi par des défis exclusifs aux équipes, des badges de distinction ou des outils de communication interne, afin de renforcer la cohésion et l'expérience communautaire au sein de la plateforme. Cette fonctionnalité sera développée après la mise en place des fonctionnalités principales de l'application.

Mot de passe oubliés (★)

Cette page envoie un **courriel** contenant un **code** à l'utilisateur.

Ce courriel doit être un courriel **HTML5**, et pas juste du texte. Et doit être personnalisé en suivant la charte graphique établie.

Lorsque l'utilisateur **entre le code**, il peut alors **modifier** son mot de passe.

Pour des raisons de sécurité, le code doit **être invalidé au bout de 15 minutes**, contenir un **minimum de 8 caractères** et empêcher le brut force.

Photo de profil (★)

Il serait intéressant, dans la mesure du possible et du temps restant disponible, d'intégrer une fonctionnalité permettant aux utilisateurs de **personnaliser leur profil** au sein de l'application. Cette personnalisation passerait notamment par la possibilité de choisir une photo de profil afin de rendre chaque compte unique et reconnaissable.

L'utilisateur pourrait ainsi **sélectionner une image** parmi une collection prédéfinie proposée par l'application composée, par exemple, d'illustrations ou d'avatars standards ou bien, si vous préférez, **pouvoir importer sa propre photo** depuis la galerie de son appareil.

Photo d'équipe (★)

Dans la continuité de la fonctionnalité de la photo de profil, il serait également pertinent de permettre aux utilisateurs de définir une **photo d'équipe**. Cette image servirait à représenter visuellement une équipe au sein de l'application.

L'ajout d'une photo d'équipe contribuerait à renforcer le sentiment d'appartenance et à faciliter l'identification des différents groupes, particulièrement dans les classements ou les pages de profil d'équipe.

La photo pourrait être sélectionnée parmi une **galerie d'images prédéfinies** (avatars, icônes, visuels thématiques) ou être **téléchargée directement par le créateur de l'équipe**.

User Stories

Les User Stories suivantes décrivent les principaux besoins fonctionnels de la plateforme ClashCode, classés par rôle d'utilisateur.

Objectif global

En tant qu'utilisateur de ClashCode, **je veux** utiliser une application **afin d'**apprendre la programmation de manière ludique et interactive, progresser à travers des défis de différents niveaux, et rejoindre une communauté dynamique partageant les mêmes objectifs d'apprentissage.

Utilisateur non connecté (Invité)

★★★ **En tant qu'invité, je veux** accéder à la page d'accueil **afin de** découvrir le concept de ClashCode et comprendre son fonctionnement.

★★★ **En tant qu'invité, je veux** pouvoir créer un compte **afin d'**accéder aux défis et aux classements.

★★★ **En tant qu'invité, je veux** pouvoir me connecter avec mon pseudonyme ou mon courriel **afin d'**accéder à mon profil et mes données.

★★ **En tant qu'invité, je veux** pouvoir m'inscrire via Google ou Microsoft **afin de** gagner du temps lors de la création de compte.

Utilisateur connecté

★★★ **En tant qu'utilisateur, je veux** pouvoir consulter tous les défis disponibles **afin d'en** choisir un à résoudre.

★★★ **En tant qu'utilisateur, je veux** pouvoir filtrer et trier les défis par langage, difficulté ou date **afin de** trouver ceux adaptés à mon niveau.

★★★ **En tant qu'utilisateur, je veux** pouvoir soumettre ma solution (flag) à un défi **afin de** la valider, et recevoir un score.

★★★ **En tant qu'utilisateur, je veux** voir mon score et mon rang **afin d'**évaluer ma progression.

★★ **En tant qu'utilisateur, je veux** pouvoir modifier mes informations personnelles (pseudo, description, mot de passe, photo de profil) **afin de** personnaliser mon compte.

★★ **En tant qu'utilisateur, je veux** consulter mon profil public et ceux des autres **afin de** visualiser les défis réalisés et leur progression.

★★ **En tant qu'utilisateur, je veux** rechercher un autre joueur par son pseudonyme ou son code ami **afin de** pouvoir consulter son profil.

★★ **En tant qu'utilisateur, je veux** pouvoir voir mes statistiques de progression et mes dernières réussites **afin de** suivre ma progression.

- ★ **En tant qu'utilisateur, je veux** pouvoir réinitialiser mon mot de passe si je l'ai oublié **afin de** regagner l'accès à mon compte.
- ★ **En tant qu'utilisateur, je veux** recevoir un courriel sécurisé avec un code de validation temporaire **afin de** réinitialiser mon mot de passe.

Classements et communauté

- ★★ **En tant qu'utilisateur, je veux** consulter le classement global **afin de** connaître les meilleurs joueurs de la plateforme.
- ★★ **En tant qu'utilisateur, je veux** filtrer le classement par langage de programmation **afin de** comparer mes performances dans un domaine spécifique.
- ★★ **En tant qu'utilisateur, je veux** rechercher un joueur dans le classement **afin d'**accéder à son profil public.
- ★ **En tant qu'utilisateur, je veux** que le top 3 soit mis en avant visuellement (podium) **afin de** rendre la compétition plus attractive.

Système d'équipes

- ★★ **En tant qu'utilisateur, je veux** pouvoir créer une équipe avec un nom et une description **afin de** jouer en groupe.
- ★★ **En tant qu'utilisateur, je veux** pouvoir rejoindre une équipe existante **afin de** contribuer à un score collectif.
- ★★ **En tant qu'utilisateur, je veux** voir la page de mon équipe avec la liste des membres, leur score et la progression du groupe **afin de** me comparer avec eux.
- ★★ **En tant que** créateur d'équipe, je veux pouvoir modifier ou supprimer mon équipe **afin d'en** garder la gestion.
- ★★ **En tant qu'utilisateur, je veux** voir le classement des équipes **afin de** suivre la compétition entre groupes.
- ★ **En tant qu'utilisateur, je veux** pouvoir définir une photo d'équipe **afin de** personnaliser son apparence dans le classement.

Administrateur

★★★ **En tant qu'administrateur, je veux** pouvoir accéder à une interface dédiée **afin de** gérer les utilisateurs et les défis sans connaissances techniques.

★★★ **En tant qu'administrateur, je veux** pouvoir créer, modifier ou supprimer des défis **afin d'enrichir** le contenu de la plateforme.

★★★ **En tant qu'administrateur, je veux** pouvoir modifier ou supprimer un utilisateur en cas de problème de comportement ou de compte inactif **afin de** protéger l'application.

★★ **En tant qu'administrateur, je veux** consulter les logs du site (authentifications, créations d'équipes, réussite de défis) **afin d'assurer** un suivi.

Sécurité et maintenance

★★★ **En tant qu'utilisateur, je veux** que mon mot de passe soit stocké de manière sécurisée **afin de** protéger mes données personnelles.

★★★ **En tant qu'utilisateur, je veux** que mes sessions expirent automatiquement **afin d'éviter** les connexions non sécurisées.

★★ **En tant qu'administrateur, je veux** que les variables sensibles (clés JWT, identifiants, etc.) soient stockées dans des fichiers d'environnement sécurisés **afin de** sécuriser mes données.

★ **En tant qu'utilisateur, je veux** pouvoir activer une double authentification **afin de** renforcer la sécurité de mon compte.

Expérience utilisateur et interface

★★★ **En tant qu'utilisateur, je veux** une interface claire et responsive **afin de** pouvoir naviguer facilement sur ordinateur, tablette ou smartphone.

★★ **En tant qu'utilisateur, je veux** que le site conserve un thème cohérent (couleurs, polices, logo) **afin d'offrir** une expérience homogène.

★ **En tant qu'utilisateur, je veux** pouvoir changer de thème (mode clair/sombre) **afin d'adapter** l'affichage à mes préférences.

II. Cahier des charges techniques

Cette partie a pour objectif de définir les **spécifications techniques** nécessaires à la conception et au développement de ClashCode.

Cela s'adresse à l'équipe technique en charge de la réalisation, afin de garantir une compréhension des choix technologiques, des **contraintes** et des **attentes** en matière de performance, de sécurité et de maintenabilité.

Interface utilisateur / Frontend

L'interface utilisateur constitue un élément central de notre application. Elle devra offrir une expérience **fluide, intuitive** et **moderne**, tout en garantissant une bonne maintenabilité.

Afin d'atteindre ces objectifs, nous exigeons l'utilisation d'un framework frontend récent, reconnu et adopté par une grande communauté.

Celui-ci devra permettre :

- La création d'interfaces **dynamiques** et **réactives**.
- Une séparation des **responsabilités** (composants et logique).
- Une **maintenabilité** à long terme.
- Une **compatibilité** avec les bonnes pratiques actuelles en matière de performance et d'accessibilité (responsive dans la limite du possible).

Dans ce cadre, nous proposons l'utilisation de l'un des deux frameworks suivants :



Serveur / Backend

Le backend devra être écrit en **Javascript** afin de limiter le nombre de langages dans le projet. Javascript est également un langage avec un écosystème riche, grâce à npm, et orienté pour le web. Le framework à utiliser sera **Express.js** pour sa simplicité et modularité.

Bien qu'Express.js donne beaucoup de libertés, nous voulons du code structuré et modulaire, pour que les différents éléments puissent être réutilisés.

Le système d'authentification se fera en utilisant **Passportjs**, pour donner plusieurs choix d'authentification aux utilisateurs, par exemple une authentification avec un compte Microsoft, Google ou GitHub, ... La gestion des sessions une fois authentifié sera **stateless** en utilisant des **JWT**.



Base de données

La base de données sera faite avec MySQL car c'est une base de données relationnelle qui permettra de stocker de façon organisée les données du site, comme les utilisateurs, les équipes, scores, ... La base de données devra être **normalisée 3NF**.



Sécurité

La sécurité de l'application doit être une priorité, l'application devra inclure au minimum ces différents points

- Protection contre les **injections SQL**
- Protection contre le **XSS** (Cross Site Scripting)
- Les mots de passes doivent être stocké de manière hashé, via **Argon2ID**
- **Expiration automatique** des sessions
- Si possible de la double authentification

Pour les utilisateurs, il est nécessaire de forcer l'utilisation d'un mot de passe sécurisé (12 caractères, avec lettres en majuscule et minuscules, des chiffres et des caractères spéciaux).

En complément, il sera demandé de prévoir une gestion centralisée des **variables d'environnements** sécurisée pour ne pas avoir à écrire en dur les données sensibles dans le code source.

Les éléments suivants devront être stockés dans un fichier de log :

- Événements d'authentification
- La réussite des défis des utilisateurs
- Événements autour des équipes

Tests

Tests unitaires (★★★)

Les fonctions du backend devront être couvertes par des tests unitaires :

- La totalité des routes d'API
- La logique métier (calcul des scores, validation des données, ...)

Tests de performance (★★)

Également de manière optionnelle, cela pourrait être intéressant d'effectuer des tests de montée en charge sur les endpoints principaux en vue d'une mise en production.

Ces tests permettront d'anticiper certains crashes lors de grande affluence et optimiser certaines requêtes.

Exemple d'outil pour réaliser des tests de montée en charge : **Artillery**.

Tests d'interface (★)

De manière optionnelle, les pages principales du frontend (connexion, inscription, défis, classement) pourront être testées afin de vérifier **les interactions utilisateur** et les transitions entre écrans.

Conteneurisation

Le déploiement de **ClashCode** reposera sur la technologie **Docker** pour avoir une mise en production simplifiée et reproductible sur plusieurs environnements distincts.

Images

Des images distinctes devront être créées pour les différents services de l'application :

- La base de données
- Le serveur
- L'interface utilisateur

Chaque image devra être optimisée pour réduire la taille et assurer une sécurité.

Orchestration des conteneurs

L'orchestrateur **docker compose** sera utilisé pour configurer et gérer les différents conteneurs de l'application. Cela permettra de gérer les dépendances entre services efficacement et de simplifier le déploiement.

Configuration Docker :

Conteneur Base de données :

- Image à utiliser : **mysql:8.0.43-debian**
- Port exposé : **3306** (port par défaut de MySQL)
- **Utilisation de volumes docker** pour assurer la persistance des données.

Conteneur API (Backend) :

- Image à utiliser : **Image personnalisée** incluant les dépendances nécessaires au fonctionnement de l'API. (Sur la base de l'image : **node:25-trixie-slim**)
- Port exposé : **8080**

Conteneur Interface Utilisateur (Frontend) :

- Image à utiliser : Image personnalisée incluant les dépendances nécessaires au fonctionnement du service. (Sur la base de l'image : **node:25-trixie-slim**)
- Port exposé : **3000**

Outil de suivi de projet

Le suivi du projet sera réalisé via l'outil **Notion** afin de permettre aux enseignants d'accéder facilement à votre niveau d'avancement du projet. Cet outil a été choisi, car il facilite le partage et l'organisation du travail, tout en restant simple d'utilisation.

III. Charte graphique

Couleurs

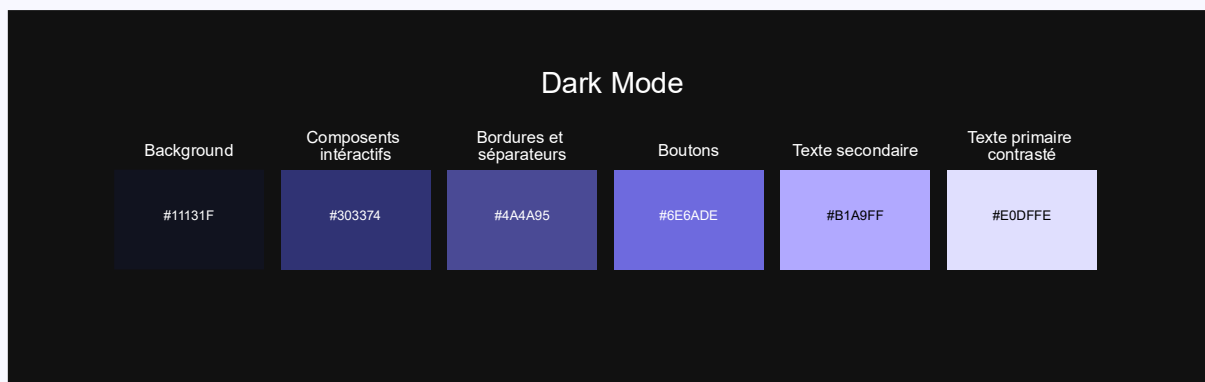
Nous souhaitons que les couleurs de l’application respectent les palettes suivantes.

Couleurs du mode jour :



#F8F8FF - #E6E7FF - #B8BAF8 - #5B5BD6 - #5753C6 - #272962

Couleurs du mode nuit :



#11131F - #303374 - #4A4A95 - #6E6ADE - #B1A9FF - #E0DFFE

Vous n’êtes **pas obligé** d’implémenter les thèmes clair/sombre. Celle-ci reste une **fonctionnalité mineure**. Cependant, si vous décidez de ne faire que le mode nuit ou jour, il devra respecter les couleurs ci-dessus. Chaque couleur correspond un composant de l’application.

Polices d’écriture

La police **Public Sans** sera utilisée pour l’ensemble des contenus textuels généraux : titres, sous-titres, paragraphes, menus et éléments d’interface. Elle garantit une lecture simple et fluide pour l’utilisateur final. [Lien Google Font](#)

La police **Fira Mono** sera réservé aux usages techniques : affichage de code, extraits informatiques, diverses références à la programmation. [Lien Google Font](#)

Logo

Ci-dessous est présent le logo de l'application en thème clair et sombre que vous devrez utiliser :



Logo pour thème clair



Logo pour thème sombre

Sans fond :



Nous pourrions vous les fournir dans le format que vous souhaitez via nos différents modes de communication.

IV. Modalités des livrables, mode de communication et documentation

Mode de communication

Après avoir pris connaissance de cahier des charges, nous attendons que vous nous contactiez par nos **adresses courriels UBS** :

- e2301317@etud.univ-ubs.fr - Paul Vrignaud
- e2202181@etud.univ-ubs.fr - Titouan Vidal
- e2301150@etud.univ-ubs.fr - Alexis Cesmat-Belliard
- e2300154@etud.univ-ubs.fr - Benjamin Trévian
- e2301875@etud.univ-ubs.fr - William Morhan
- e2303371@etud.univ-ubs.fr - François Patinec

Après cela, nous vous enverrons un lien vers notre serveur **Discord** qui permettra de communiquer avec nous plus rapidement et de répondre à vos questions si besoin.

Une fois sur le serveur discord, vous devrez mettre en place un **webhook** dans un channel donné pour nous permettre de suivre l'avancement du projet. [Tutoriel sur les webhooks Discord et GitLab.](#)

Livrables

L'accès au travail se fera via un dépôt **GitLab**, qui devra être créé sur la **forge logicielle** de l'UBS pour pouvoir être accessible aux enseignants et à notre équipe de clients.

Documentation

Afin d'avoir un maximum d'outils pour les futurs développeurs, nous exigeons un README.md à la source du projet qui expliquera les concepts clés de **ClashCode**. Nous attendons aussi un **code commenté**.

Nous demandons aussi une **documentation de chaque endpoints** sous le format **OpenAPI**. Il peut-être intéressant de créer une interface pour les visualiser via **SwaggerUI** (démonstration d'une documentation via **SwaggerUI**) ou via un service tel qu'**OpenAPIViewer**.

En plus, nous attendons un PDF expliquant le **fonctionnement du Front-end et du Back-end**. Mais aussi un PDF pour **expliquer le mode administrateur**.