

## Chapitre I - Preamble

---

Ce projet consiste à faire quelque chose que vous n'avez jamais fait auparavant. Rappelez-vous le début de votre parcours en informatique. Regardez-vous maintenant. Il est temps de briller !

## Chapitre II - Points Essentiels

Ce projet est un défi complexe qui exige de prendre des décisions tout en respectant les contraintes spécifiées. Vous avez une certaine flexibilité pour implémenter certains modules, à vous de décider dans les limites du sujet. Toutes vos décisions doivent être justifiables.

Si vous estimez nécessaire d'utiliser nginx pour configurer votre site web, cela ne pose aucun problème. Mais posez-vous d'abord la question suivante : est-ce vraiment indispensable ? Puis-je m'en passer ? De même, face à une bibliothèque qui pourrait vous aider, il est crucial de comprendre si elle remplira les tâches demandées. Il ne s'agit pas de refaire des sous-couches sans intérêt, mais de rendre les fonctionnalités proposées opérationnelles.

Il est essentiel de comprendre que vous rencontrerez des décisions qui susciteront des doutes quant à l'implémentation de certaines fonctionnalités. Il est fortement recommandé de bien comprendre les exigences du projet dès le début. Une fois que vous aurez saisi ce qui doit être accompli, il sera nécessaire de rester dans le cadre du projet. Quand nous mentionnons une technologie imposée, cela signifie explicitement que tout ce qui est officiellement lié au framework/langage demandé est autorisé.

Cependant, nous soulignons que lorsque vous souhaitez implémenter un module, toutes les restrictions s'appliquent à ce module. Par exemple, si vous voulez réaliser le projet avec le module Backend tel qu'indiqué dans le sujet, vous ne pouvez plus utiliser le langage par défaut et devez adapter votre projet en conséquence. Si vous souhaitez toujours créer un backend en utilisant le langage/framework par défaut, c'est aussi possible, mais ce module ne sera pas considéré comme valide.

Avant de conclure, il est important de noter que certains modules dépendent fortement d'autres de manière intentionnelle.

Vos choix sont significatifs et doivent être justifiés lors de votre évaluation. Soyez prudent.

Prenez le temps de réfléchir à la conception de votre application en fonction de vos choix avant de plonger dans le code – c'est essentiel.

Amusez-vous bien ! 😊

## Chapitre III - Partie Obligatoire

Ce projet consiste à créer un site web pour le grand concours de Pong !

- L'utilisation de bibliothèques ou d'outils fournissant une solution immédiate complète pour une fonctionnalité ou un module global est interdite.
- Toute directive explicite concernant l'utilisation (peut, doit, ne peut pas) d'une bibliothèque ou d'un outil tiers doit être respectée.
- L'utilisation d'une petite bibliothèque ou d'un outil qui résout une tâche simple et unique, représentant un sous-composant d'une fonctionnalité ou d'un module global, est autorisée.
- Lors de l'évaluation, l'équipe devra justifier l'utilisation de toute bibliothèque ou outil qui n'est pas explicitement approuvé par le sujet.
- L'évaluateur devra définir si l'utilisation d'une bibliothèque ou d'un outil spécifique est légitime (et autorisée) ou si elle résout presque toute une fonctionnalité ou un module (et est donc interdite).

### III.1 Aperçu

Grâce à votre site web, les utilisateurs joueront à Pong les uns contre les autres. Vous devez fournir une interface utilisateur agréable et des jeux en ligne multijoueurs en temps réel.

- Votre projet doit respecter les directives suivantes comme exigences minimales, représentant une petite part de la note finale.
- La seconde partie du sujet propose des modules supplémentaires qui peuvent remplacer ou compléter les règles ci-dessous.

Dans ce sujet, certains mots sont surlignés en vert. Ils représentent des choix de technologies qui évolueront avec le temps. Portez une attention particulière à la version du sujet.

### III.2 Exigences Techniques Minimales

Votre projet doit respecter les règles suivantes :

1. **Développement du site** : Libre choix de développement avec ou sans backend.
  - Si vous choisissez d'inclure un backend, il doit être écrit en pur Ruby. Cependant, cette exigence peut être remplacée par le module Framework.
  - Si votre backend ou framework utilise une base de données, vous devez suivre les contraintes du module Base de Données.
2. **Frontend** : Développé en pur JavaScript vanilla. Cette contrainte peut être modifiée par le module FrontEnd.
3. **Application monopage** : Le site doit être une application monopage, permettant d'utiliser les boutons Retour et Avancer du navigateur.
4. **Compatibilité avec Chrome** : Le site doit être compatible avec la dernière version stable de Google Chrome.
5. **Gestion des erreurs et des avertissements** : L'utilisateur ne doit rencontrer aucune erreur non gérée ni avertissement en naviguant sur le site.

6. **Commandes Docker** : Tout doit être lancé avec une seule ligne de commande pour exécuter un conteneur autonome fourni par Docker, par exemple :
- ```
docker-compose up --build.
```
- Si vous utilisez Docker, les fichiers runtime de Docker doivent être situés dans `/goinfre` ou `/sgoinfre`.
  - Vous ne pouvez pas utiliser de volumes de type "bind-mount" entre l'hôte et le conteneur si des UID non root sont utilisés dans le conteneur. Des solutions alternatives existent : Docker dans une VM, reconstruire votre conteneur après chaque changement, créer votre propre image Docker avec root comme UID unique.

### III.3 Jeu

Le but principal de ce site est de jouer à Pong contre d'autres joueurs.

1. **Jeu multijoueur en ligne** : Les utilisateurs doivent pouvoir participer à un jeu de Pong en temps réel contre un autre joueur, directement sur le site. Les deux joueurs utiliseront le même clavier. Le module Joueurs Distants peut améliorer cette fonctionnalité pour inclure des joueurs distants.
2. **Tournoi** : Les utilisateurs doivent pouvoir organiser un tournoi impliquant plusieurs joueurs jouant à tour de rôle les uns contre les autres, avec un affichage clair des participants et de l'ordre des matchs.
3. **Inscription** : Au début d'un tournoi, chaque joueur doit entrer un pseudo. Les pseudos seront réinitialisés pour chaque nouveau tournoi. Cette exigence peut être modifiée par le module Gestion Utilisateur Standard.
4. **Système de matchmaking** : Le système de tournoi doit organiser le matchmaking entre les participants et annoncer le prochain match.
5. **Règles de jeu uniformes** : Tous les joueurs doivent avoir les mêmes règles de jeu, y compris la vitesse de la raquette, et l'IA (si utilisée) doit être à la même vitesse qu'un joueur régulier.
6. **Développement du jeu** : Le jeu doit être conforme aux contraintes de frontend par défaut ou peut être modifié par les modules FrontEnd ou Graphiques.

### III.4 Préoccupations de Sécurité

Pour créer un site web fonctionnel de base, voici quelques mesures de sécurité que vous devez implémenter :

- **Hachage des mots de passe** : Tout mot de passe stocké dans votre base de données, si applicable, doit être haché.
- **Protection contre les injections SQL et XSS** : Votre site web doit être sécurisé contre les injections SQL et les attaques XSS.
- **Connexion HTTPS** : Si vous avez un backend ou toute autre fonctionnalité nécessitant une connexion, une connexion HTTPS est obligatoire pour tous les aspects (utiliser wss au lieu de ws...).
- **Validation des formulaires et entrées utilisateur** : Une forme de validation doit être mise en place pour les formulaires et toute entrée utilisateur, que ce soit dans la page de base si aucun backend n'est utilisé ou côté serveur si un backend est en place.
- **Sécurité API et authentification** : Que vous choisissiez ou non d'implémenter le module de sécurité JWT avec 2FA, la sécurité de votre site reste une priorité. Si vous optez pour une API, assurez-vous que les routes sont protégées. Même sans JWT, la sécurisation des routes reste essentielle.

**Conseil** : Utilisez un algorithme de hachage robuste pour les mots de passe. Pour des raisons évidentes de sécurité, toute information sensible (identifiants, clés API, variables d'environnement, etc.) doit être stockée localement dans un fichier `.env` et ignorée par Git. Les informations d'identification stockées publiquement entraîneront l'échec du projet.

---

## Chapitre IV - Modules

Félicitations, vous avez atteint 25 % du projet ! Avec un site web fonctionnel de base en place, la prochaine étape consiste à choisir des modules pour améliorer davantage votre site. Pour atteindre 100 % de réalisation, un minimum de 7 modules principaux est requis.

Il est essentiel de lire chaque module attentivement, car certains modules peuvent nécessiter des modifications dans la structure de base de votre site. Nous vous recommandons vivement de lire tout le sujet pour bien comprendre les interactions entre les modules.

### IV.1 Vue d'Ensemble des Modules

Vous trouverez ici une liste de modules majeurs et mineurs organisés par domaine :

- **Web :**
  - **Module majeur :** Utiliser un framework pour le backend (Django).
  - **Module mineur :** Utiliser un framework ou un toolkit pour le frontend (Bootstrap).
  - **Module mineur :** Utiliser une base de données pour le backend (PostgreSQL).
  - **Module majeur :** Stocker le score d'un tournoi sur une blockchain (Ethereum).
- **Gestion des Utilisateurs :**
  - **Module majeur :** Gestion standard des utilisateurs (inscription, authentification, profils).
  - **Module majeur :** Authentification à distance (OAuth 2.0 avec l'école 42).
- **Gameplay et Expérience Utilisateur :**
  - **Module majeur :** Joueurs distants (jeu en réseau pour deux joueurs sur des ordinateurs différents).
  - **Module majeur :** Multijoueurs (plus de deux joueurs dans le même jeu).
  - **Module majeur :** Ajouter un autre jeu avec historique utilisateur et matchmaking.
  - **Module mineur :** Options de personnalisation du jeu.
  - **Module majeur :** Chat en direct.
- **IA et Algorithmes :**
  - **Module majeur :** Introduction d'un adversaire IA.
  - **Module mineur :** Tableaux de statistiques pour les utilisateurs et le jeu.

- **Cybersécurité :**
    - **Module majeur :** Mise en place d'un WAF/ModSecurity avec une configuration renforcée et gestion des secrets avec HashiCorp Vault.
    - **Module mineur :** Options de conformité RGPD avec anonymisation des utilisateurs et suppression des comptes.
    - **Module majeur :** Authentification à deux facteurs (2FA) et tokens JWT.
  - **DevOps :**
    - **Module majeur :** Infrastructure de gestion des logs avec la pile ELK (Elasticsearch, Logstash, Kibana).
    - **Module mineur :** Système de monitoring (Prometheus et Grafana).
    - **Module majeur :** Conception du backend en microservices.
  - **Graphiques :**
    - **Module majeur :** Techniques 3D avancées avec ThreeJS/WebGL.
  - **Accessibilité :**
    - **Module mineur :** Support sur tous les appareils.
    - **Module mineur :** Compatibilité avec plusieurs navigateurs.
    - **Module mineur :** Support multilingue.
    - **Module mineur :** Accessibilité pour les malvoyants.
    - **Module mineur :** Rendu côté serveur (SSR).
  - **Pong Côté Serveur :**
    - **Module majeur :** Remplacer le Pong de base par un Pong côté serveur avec une API.
    - **Module majeur :** Permettre de jouer à Pong via une interface en ligne de commande (CLI) contre des utilisateurs web grâce à l'intégration API.
- 

## IV.2 Web

Ces modules permettent d'intégrer des fonctionnalités web avancées dans votre projet Pong :

- **Utiliser un framework pour le backend :** Obligatoirement avec Django, ce module valide l'utilisation d'un framework si vous suivez cette contrainte.

- **Utiliser Bootstrap pour le frontend** : Ce module valide l'utilisation d'un toolkit frontend.
- **Utiliser PostgreSQL pour la base de données** : Obligatoire pour garantir la cohérence des données.
- **Stockage des scores sur la blockchain** : Intégration d'Ethereum avec des contrats intelligents en Solidity pour stocker les scores en toute sécurité.

### IV.3 Gestion des Utilisateurs

Ces modules abordent des aspects essentiels de la gestion des utilisateurs et du contrôle d'accès :

- **Gestion standard des utilisateurs** : Inscription, connexion, mise à jour des informations, gestion d'amis, historique de match.
- **Authentification à distance** : Système d'authentification OAuth 2.0 avec l'école 42.

### IV.4 Gameplay et Expérience Utilisateur

Ces modules visent à enrichir le gameplay de votre projet :

- **Joueurs distants** : Permettre à deux joueurs éloignés de jouer ensemble.
- **Multijoueurs** : Plus de deux joueurs par jeu.
- **Ajouter un autre jeu avec historique et matchmaking.**
- **Personnalisation des jeux** : Options de personnalisation du gameplay.
- **Chat en direct** : Permettre la messagerie instantanée entre les utilisateurs.

### IV.5 Intelligence Artificielle et Algorithmes

Ces modules introduisent des éléments basés sur les données dans le projet, avec un module majeur qui introduit un adversaire IA pour enrichir le gameplay, et un module mineur axé sur les tableaux de statistiques utilisateur et de jeu.

- **Module Majeur : Introduction d'un Adversaire IA**
  - Objectif : Ajouter un joueur IA pour une expérience de jeu plus stimulante.
  - Contraintes :
    - L'algorithme A\* n'est pas autorisé.
    - L'IA doit simuler un comportement humain, ce qui implique de simuler les entrées clavier.
    - L'IA ne peut rafraîchir sa vue du jeu qu'une fois par seconde.
  - L'IA doit utiliser des power-ups si le module d'options de personnalisation de jeu est implémenté.



- Vous devez expliquer en détail le fonctionnement de votre IA lors de l'évaluation.
  - **Module Mineur : Tableaux de Statistiques Utilisateur et Jeu**
    - Objectif : Offrir des tableaux de statistiques permettant aux utilisateurs de suivre leurs performances et d'afficher les résultats des sessions de jeu.
    - Fonctionnalités :
      - Tableau de bord pour chaque utilisateur avec ses statistiques de jeu.
      - Affichage de données de session de jeu sous forme de graphiques et tableaux.
      - Les utilisateurs peuvent explorer leur historique de jeu et leurs performances.
- 

## IV.6 Cybersécurité

Ces modules de cybersécurité visent à renforcer la sécurité du projet en implémentant des solutions de protection robustes et en respectant les normes de protection des données.

- **Module Majeur : Mise en Place d'un WAF/ModSecurity et HashiCorp Vault pour la Gestion des Secrets**
    - Objectif : Déployer un pare-feu d'application web (WAF) et utiliser HashiCorp Vault pour sécuriser les informations sensibles.
    - Fonctionnalités :
      - Configurer et déployer un WAF avec ModSecurity.
      - Utiliser HashiCorp Vault pour stocker les clés API, identifiants et autres données sensibles.
  - **Module Mineur : Options de Conformité RGPD avec Anonymisation des Données et Suppression de Compte**
    - Objectif : Ajouter des fonctionnalités conformes au RGPD pour la protection de la vie privée des utilisateurs.
    - Fonctionnalités :
      - Anonymisation des données personnelles sur demande.
      - Gestion locale des données pour que les utilisateurs puissent consulter, éditer ou supprimer leurs informations personnelles.
      - Possibilité de supprimer définitivement un compte et toutes les données associées.
  - **Module Majeur : Authentification à Deux Facteurs (2FA) et JWT**
    - Objectif : Renforcer la sécurité des comptes utilisateurs avec une authentification forte.
    - Fonctionnalités :
      - Implémentation d'une authentification à deux facteurs (2FA) pour renforcer la sécurité des utilisateurs.
      - Utilisation de JSON Web Tokens (JWT) pour gérer les sessions et autoriser l'accès aux ressources en toute sécurité.
-

## IV.7 DevOps

Ces modules DevOps se concentrent sur l'amélioration de l'infrastructure et de l'architecture du projet, avec des modules pour la gestion des journaux, la surveillance du système, et la conception en microservices.

- **Module Majeur : Infrastructure de Gestion des Journaux avec la Pile ELK**
    - Objectif : Mettre en place un système de gestion et d'analyse des journaux avec Elasticsearch, Logstash, et Kibana.
    - Fonctionnalités :
      - Déploiement d'Elasticsearch pour le stockage et l'indexation des journaux.
      - Configuration de Logstash pour la collecte et le traitement des données de journaux.
      - Mise en place de Kibana pour visualiser les données et créer des tableaux de bord.
  - **Module Mineur : Système de Surveillance avec Prometheus et Grafana**
    - Objectif : Configurer un système de surveillance avec Prometheus et Grafana.
    - Fonctionnalités :
      - Déploiement de Prometheus pour la collecte de métriques et la surveillance des composants du système.
      - Visualisation des métriques en temps réel avec Grafana et création de règles d'alerte.
  - **Module Majeur : Conception du Backend en Microservices**
    - Objectif : Concevoir une architecture backend en microservices pour plus de flexibilité et de scalabilité.
    - Fonctionnalités :
      - Diviser le backend en microservices indépendants, chacun responsable d'une fonction spécifique.
      - Communication entre microservices via des API REST ou des files de messages.
- 

## IV.8 Gaming

Ces modules visent à enrichir l'aspect ludique du projet, avec un module majeur ajoutant un nouveau jeu et un module mineur offrant des options de personnalisation.

- **Module Majeur : Ajouter un Autre Jeu avec Historique Utilisateur et Matchmaking**
  - Objectif : Introduire un nouveau jeu sur la plateforme avec suivi de l'historique utilisateur et système de matchmaking.
  - Fonctionnalités :
    - Suivi de l'historique de jeu pour chaque utilisateur.
    - Matchmaking pour trouver des adversaires de niveau équitable.
    - Mise à jour régulière et optimisation du jeu pour une expérience utilisateur agréable.

- **Module Mineur : Options de Personnalisation du Jeu**
    - Objectif : Fournir des options de personnalisation pour améliorer l'expérience de jeu.
    - Fonctionnalités :
      - Ajout de fonctionnalités de personnalisation telles que des power-ups, attaques, ou différentes cartes.
      - Interface utilisateur pour ajuster les paramètres de jeu.
- 

## IV.9 Graphiques

- **Module Majeur : Techniques 3D Avancées**
    - Objectif : Utiliser des techniques 3D avancées avec ThreeJS/WebGL pour améliorer l'expérience visuelle du jeu Pong.
    - Fonctionnalités :
      - Création d'effets visuels immersifs pour une meilleure expérience de jeu.
      - Intégration avec ThreeJS/WebGL pour des graphismes 3D avancés.
- 

## IV.10 Accessibilité

Ces modules sont axés sur l'amélioration de l'accessibilité du site web, en assurant une compatibilité avec divers appareils, navigateurs, langues et en fournissant des fonctionnalités pour les utilisateurs malvoyants.

- **Module Mineur : Support sur Tous les Appareils**
    - Assurer une expérience utilisateur cohérente sur les ordinateurs, tablettes et smartphones.
  - **Module Mineur : Compatibilité Multi-Navigateurs**
    - Extension de la compatibilité de l'application pour un navigateur supplémentaire.
  - **Module Mineur : Support Multilingue**
    - Ajouter le support de plusieurs langues avec une interface de sélection de langue pour une expérience utilisateur personnalisée.
  - **Module Mineur : Accessibilité pour les Utilisateurs Malvoyants**
    - Support des lecteurs d'écran, texte alternatif pour les images, schéma de couleurs contrasté, et navigation au clavier.
  - **Module Mineur : Intégration du Rendu Côté Serveur (SSR)**
    - Amélioration de la performance du site avec SSR pour un chargement plus rapide et une meilleure expérience utilisateur.
- 

## IV.11 Pong Côté Serveur

- **Module Majeur : Remplacer Pong Basique par Pong Côté Serveur avec une API**

- Objectif : Développer un Pong côté serveur avec gestion des scores, mouvements de balle et interactions.
  - Intégration d'une API pour permettre l'accès au jeu via une interface web et une ligne de commande (CLI).
  - **Module Majeur : Pong via CLI contre les Utilisateurs Web avec Intégration API**
    - Développer une interface CLI permettant aux utilisateurs de jouer à Pong contre des utilisateurs utilisant la version web.
- 

## Chapitre V - Partie Bonus

La partie bonus consiste à intégrer davantage de modules pour obtenir des points supplémentaires. Cependant, elle ne sera évaluée que si la partie obligatoire est parfaite.

- **5 points** pour chaque module mineur ajouté.
- **10 points** pour chaque module majeur ajouté.

## Chapitre VI - Soumission et Évaluation par les Pairs

Soumettez votre travail dans votre dépôt Git habituel. Seul le contenu du dépôt sera évalué lors de la soutenance.

**Rappel** : Suivez les consignes concernant l'utilisation de bibliothèques ou d'outils, et justifiez leur utilisation lors de l'évaluation.