



ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR
Membre de
HONORIS UNITED UNIVERSITIES

IIR

INGÉNIERIE INFORMATIQUE & RÉSEAUX

ANNÉE UNIVERSITAIRE

2024-2025

OPTION

MÉTHODES INFORMATIQUES APPLIQUÉES
À LA GESTION DES ENTREPRISES (MIAGE)

PROJET DEVOPS

THÈME

**Système de gestion de tournois e-sport avec
inscriptions et suivi des matchs.**

RÉALISÉ PAR

- Haddouch Maria
- ElOuarnazi Yassine
- Raji Ayyoub
- Bakri Adam
- Tebaa Walid

ENCADRÉ PAR

- NADIRI Abdeljalil

SOUTENU LE

Liste des figures

FIGURE 1:DIAGRAMME DE CLASSE	7
FIGURE 2:SCHEMA ARCHITECTURE SYSTEME	8
FIGURE 3:LOGIN PAGE	9
FIGURE 4:REGISTER PAGE	10
FIGURE 5:REGISTER PAGE	11
FIGURE 6:PROFILE	11
FIGURE 7:TOURNAMENT BRACKET + PREDICTION	12
FIGURE 8:DASHBOARD	12
FIGURE 9:USER MANAGER.....	13
FIGURE 10:EDIT USER.....	13
FIGURE 11: TOURNAMENT MANAGER	14
FIGURE 12:EDIT TOURNAMENT.....	14
FIGURE 13:DELETE TOURNAMENT.....	15
FIGURE 14:TEAM MANAGEMENT	15
FIGURE 15: CAOTURE 1 BACKLOG	16
FIGURE 16:CAPTURE 2 BACKLOG.....	16
FIGURE 17:SPRINT 1	16
FIGURE 18:SPRINT 2	17
FIGURE 19:SPRINT 3	17
FIGURE 20:SPRINT 4	17
FIGURE 21:FILES	18
FIGURE 22:PIPELINE.....	18
FIGURE 23:AGENT POOL	19
FIGURE 24:FICHIER AZURE-PIPELINES.YML.....	19

Table des matières

Liste des figures	2
Table des matières	3
Introduction générale.....	4
I. Spécifications fonctionnelles et techniques	5
1. Spécifications Fonctionnelles.....	5
1.1 Objectif fonctionnel principal.....	5
2.1 Fonctionnalités détaillées.....	5
2. Spécifications Techniques.....	6
II. Méthodologie adoptée	7
1. Méthode UML	7
1.1 Diagramme de classe.....	7
2. Méthodologie Agile.....	8
III. Architecture système.....	8
1. Vue d'ensemble de l'architecture	8
IV. Planification et Conception	9
1. Maquettes(extraits)	9
1.1 Côté Client	9
2.1 Côté Admin	12
2. Backlogs initials	16
3. Plan des sprints.....	16
2.1 Sprint 1.....	16
3.1 Sprint 2.....	17
4.1 Sprint 3.....	17
5.1 Sprint 4.....	17
4. Gestion du code	18
5. Intégration Continue (CI) et Déploiement Continu (CD)	18
1.1 Pipeline.....	18
2.1 Agent pool	19
3.1 Fichier azure-pipelines.yml :.....	19
Conclusion générale	20

Introduction générale

L'univers de l'e-sport a connu, ces dernières années, une expansion spectaculaire, tant sur le plan médiatique qu'économique. Ce domaine, autrefois perçu comme un simple divertissement, s'est progressivement structuré en une industrie compétitive regroupant des millions de joueurs, spectateurs, équipes professionnelles, et organisateurs d'événements à travers le monde. Dans cette dynamique de professionnalisation, la gestion des tournois devient un pilier central qui requiert des solutions technologiques robustes, intelligentes et adaptables.

L'objectif de ce projet est de concevoir et de développer une **plateforme numérique complète de gestion de tournois e-sport**, capable de répondre aux besoins actuels des différents acteurs de cette industrie. L'application visera à automatiser les étapes clés de l'organisation des compétitions (inscriptions, planification des matchs, suivi en temps réel, arbitrage, classement, etc.) tout en intégrant des outils innovants pour améliorer l'expérience utilisateur. Une attention particulière sera portée à l'**ergonomie des interfaces** (joueurs et administrateurs), à la **sécurité des données**, ainsi qu'à la **scalabilité du système**.

Un des éléments différenciateurs majeurs de ce projet réside dans l'intégration de l'**intelligence artificielle** pour la **prédiction des résultats des matchs**. Grâce à l'exploitation de bases de données issues de sources comme Riot Games API, Tracker.gg ou VLR.gg, et à l'utilisation d'algorithmes de machine learning (XGBoost, Random Forest, voire Deep Learning via TensorFlow ou PyTorch), la plateforme sera en mesure de fournir des estimations probabilistes des issues des rencontres, basées sur les performances passées des joueurs et équipes. Ces prédictions pourront être utilisées à des fins d'analyse, de création de contenu, ou même de prévention de comportements suspects.

Sur le plan technique, l'application repose sur une architecture moderne : un **backend développé avec Spring Boot**, un **frontend en React.js**, et une **base de données MySQL**. La communication temps réel entre les clients et le serveur sera assurée par un système WebSocket, garantissant des mises à jour instantanées des scores et événements. Des pipelines CI/CD assureront un déploiement continu fiable, et des tests automatisés (via Cypress) viendront garantir la stabilité du produit à chaque itération.

Le développement du projet est planifié selon une approche agile, découpée en quatre sprints, chacun ciblant des livrables précis allant de la création des maquettes à l'implémentation des fonctionnalités IA, en passant par la modélisation des données et les tests finaux.

En somme, ce projet ne se limite pas à un simple outil de gestion de tournois, mais ambitionne de proposer une **expérience immersive, intelligente et performante** au cœur de l'écosystème e-sportif, alliant technologie, interactivité et analyse prédictive.

I. Spécifications fonctionnelles et techniques

1. Spécifications Fonctionnelles

1.1 Objectif fonctionnel principal

► Créer une **plateforme web et mobile** permettant :

- L'organisation et la gestion de tournois e-sport
- Le suivi en temps réel des matchs
- L'utilisation de l'intelligence artificielle pour prédire les résultats

2.1 Fonctionnalités détaillées

Pour les Administrateurs (Interface Web uniquement)

- **Gestion des tournois :**
 - Création, modification et suppression de tournois
 - Définition des règles du tournoi : nombre d'équipes, dates, type de jeu, format de compétition
 - Génération automatique des arbres de compétition
- **Modération et arbitrage :**
 - Validation ou rejet des inscriptions des équipes
 - Gestion des conflits ou litiges entre joueurs
 - Possibilité d'annuler un match si nécessaire
- **Monitoring & Statistiques :**
 - Tableau de bord interactif avec statistiques en temps réel
 - Détection automatique d'activités suspectes avec alertes

Pour les Joueurs (Interfaces Web & Mobile)

- **Inscription et gestion d'équipe :**
 - Inscription aux tournois
 - Création d'une équipe, ajout ou suppression de membres
- **Suivi de la compétition :**
 - Consultation du calendrier des matchs
 - Réception de **notifications en temps réel**
 - Suivi en direct grâce à un **indicateur LIVE**
 - Mise à jour automatique des résultats de match
- **Profil et classement :**
 - Accès à un historique personnel des participations
 - Classement personnel mis à jour après chaque tournoi

Intelligence Artificielle (IA)

- **Sources de données :**
 - Statistiques des joueurs/équipes professionnels de *Valorant*
 - Récupération via Riot Games API, Tracker.gg, VLR.gg
 - Analyse des performances passées pour calculer des probabilités de victoire
- **Modèles d'IA utilisés :**
 - Algorithmes de machine learning : **XGBoost, Random Forest**
 - Possibilité d'intégrer des modèles plus avancés via **TensorFlow ou PyTorch**
 - Entraînement supervisé basé sur des données historiques

2. Spécifications Techniques

Composant	Technologie / Outil
Frontend Web	React.js
Frontend Mobile	React native
Backend	Spring Boot
Base de données	MySQL (via MySQL Workbench)
Authentification	JWT (JSON Web Token)
Communication temps réel	WebSocket
CI/CD	Pipeline à configurer pour intégration et déploiement
Tests automatisés	Cypress (tests E2E)
Modélisation IA	XGBoost, Random Forest, TensorFlow, PyTorch
Sources de données	Riot Games API, Tracker.gg, VLR.gg

II. Méthodologie adoptée

1. Méthode UML

→ Le langage de modélisation unifié, (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.

1.1 Diagramme de classe

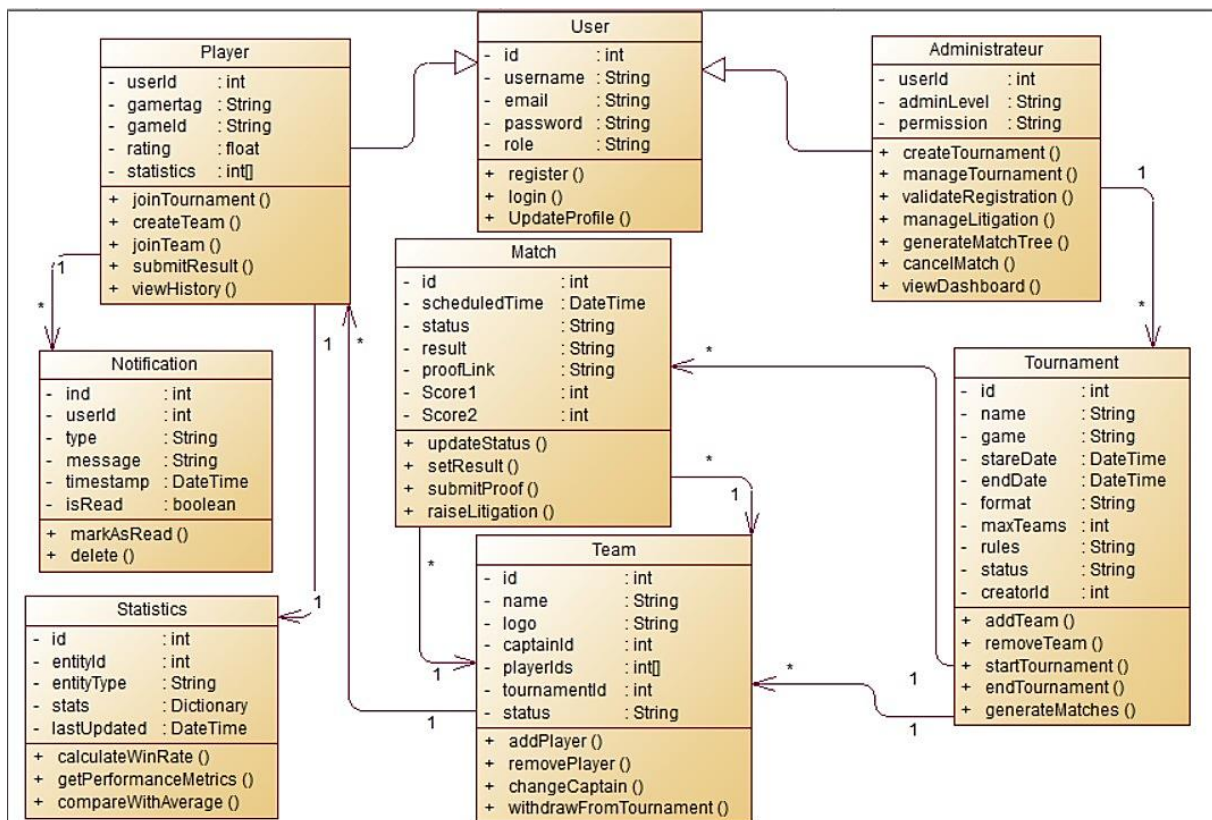


Figure 1:Diagramme de classe

→ Ce diagramme de classes modélise le système d'une plateforme de gestion de tournois de jeux vidéo. Il distingue plusieurs rôles et entités clés : **User**, **Player**, **Administrateur**, **Team**, **Match**, **Tournament**, **Notification** et **Statistics**. Chaque classe contient des attributs et méthodes correspondant à ses responsabilités. Les relations entre les classes montrent l'interconnexion entre les utilisateurs, les équipes, les matchs et les tournois. Le système gère aussi les notifications, les statistiques de performance et les litiges, offrant ainsi une structure complète pour organiser, suivre et encadrer des compétitions en ligne.

2. Méthodologie Agile

Afin d'assurer à la fois la qualité du code et la rapidité de livraison, nous avons adopté une approche DevOps agile, reposant sur les principes suivants :

- **Méthodologie Agile / Scrum** : la gestion du projet s'effectue par itérations (sprints), avec un backlog produit structuré, des user stories clairement définies et une planification centralisée dans Azure Boards.
- **Intégration Continue (CI)** : chaque commit ou push sur le dépôt déclenche automatiquement des pipelines d'intégration continue, permettant de compiler et tester le code de manière systématique.
- **Déploiement Continu (CD)** : une fois validé, le code est automatiquement déployé dans un environnement de préproduction (staging) via Azure Pipelines.
- **Infrastructure DevOps** :
 - Projet configuré dans Azure DevOps
 - Équipe projet définie avec des rôles et responsabilités attribués
 - Dépôt Git versionné pour la gestion du code source
 - Pipelines CI/CD automatisés, définis via des fichiers YAML, avec agents et pools configurés.

III. Architecture système

1. Vue d'ensemble de l'architecture

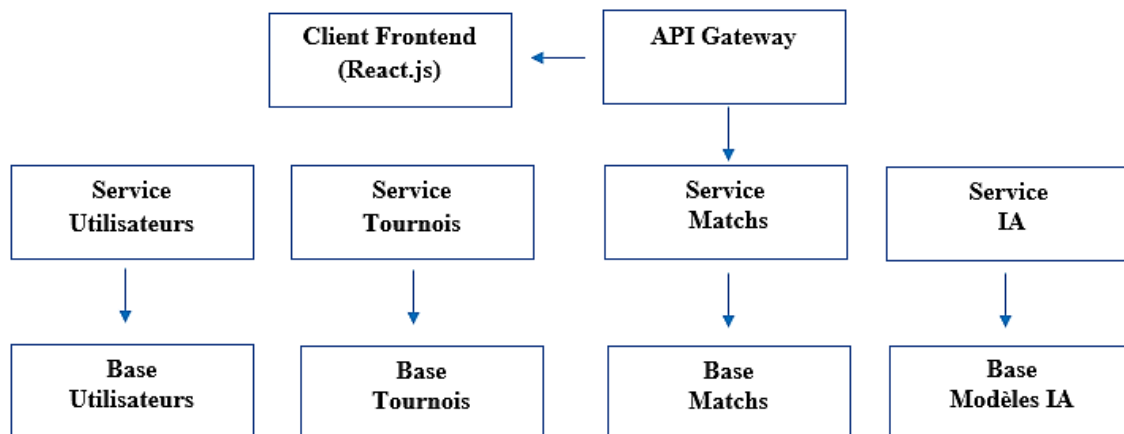


Figure 2:Schéma Architecture système

► Ce schéma représente une architecture microservices pour une application de gestion de tournois. Le frontend React.js communique avec une API Gateway, qui redirige les requêtes vers des services dédiés : utilisateurs, tournois, matchs et intelligence artificielle. Chaque service possède sa propre base de données, assurant autonomie, modularité et évolutivité du système.

IV. Planification et Conception

1. Maquettes(extraits)

1.1 Côté Client

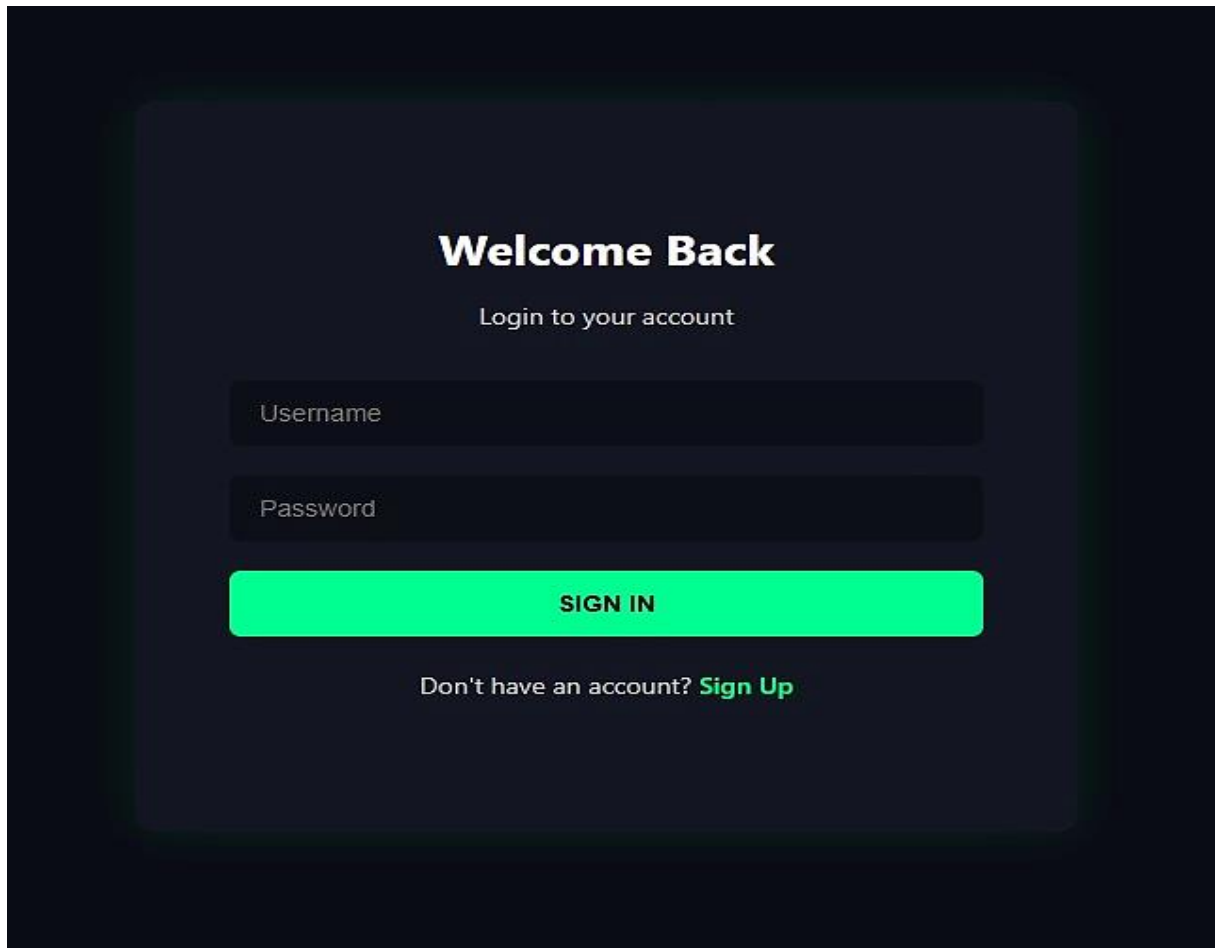
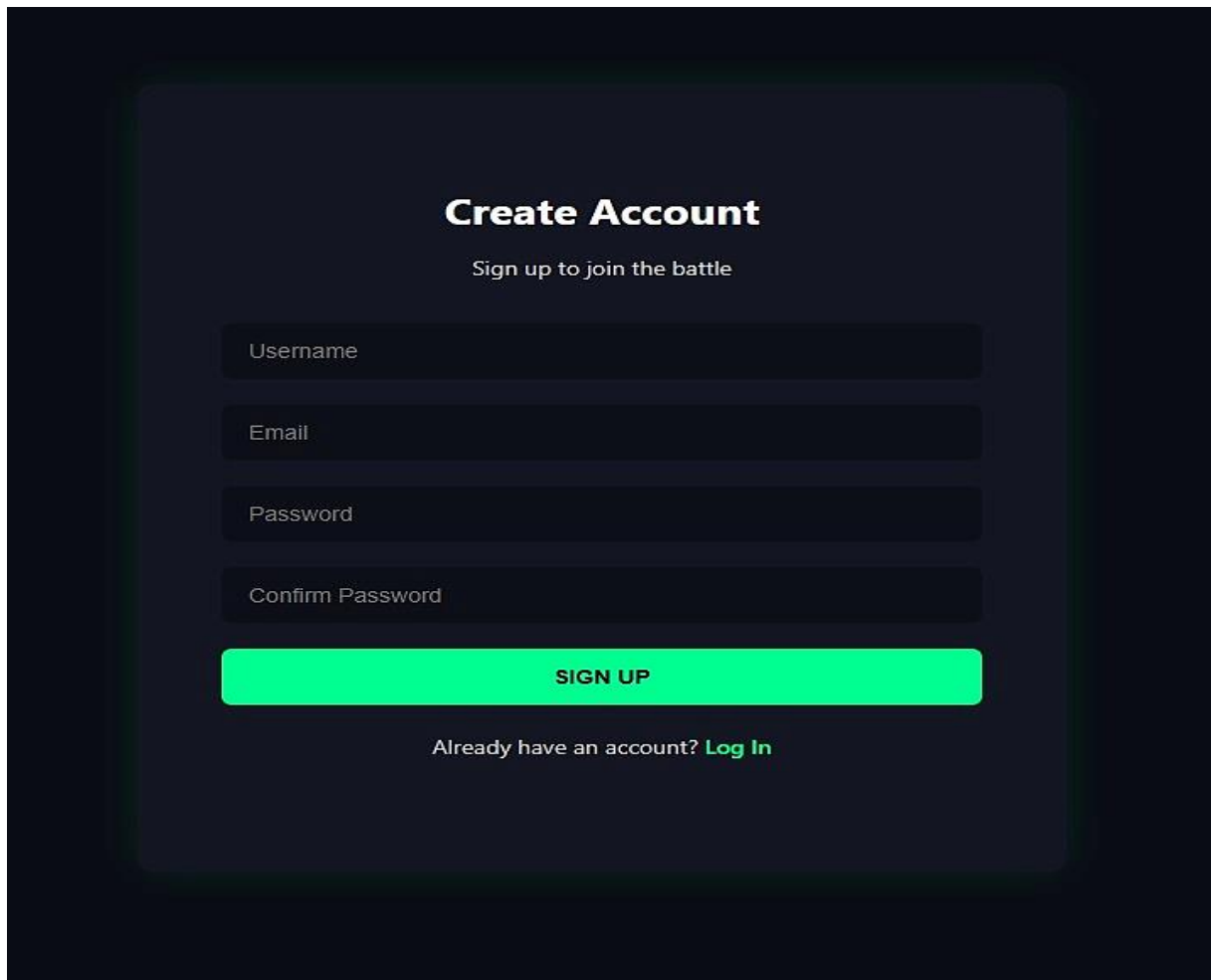


Figure 3:Login page

A registration form titled "Create Account" with a subtitle "Sign up to join the battle". It features four input fields for "Username", "Email", "Password", and "Confirm Password", followed by a red "SIGN UP" button and a link to "Log In" for existing users.

Create Account

Sign up to join the battle

Username

Email

Password

Confirm Password

SIGN UP

Already have an account? [Log In](#)

Figure 4: Register page







FILTER BY GAME ↓		FILTER BY TOURNAMENT ↓	
1		Solary BR	2600
2		LOUD Academy NA	2000
3		Liberty EU	1980
4		Supernova NA	1970
5		INTZ NA	1950

Figure 5: Register page



JohnQT

Gamertag: **adddd**

Game ID: **johnqt**

Rating: **1600**

[Edit Profile](#)

Figure 6: Profile

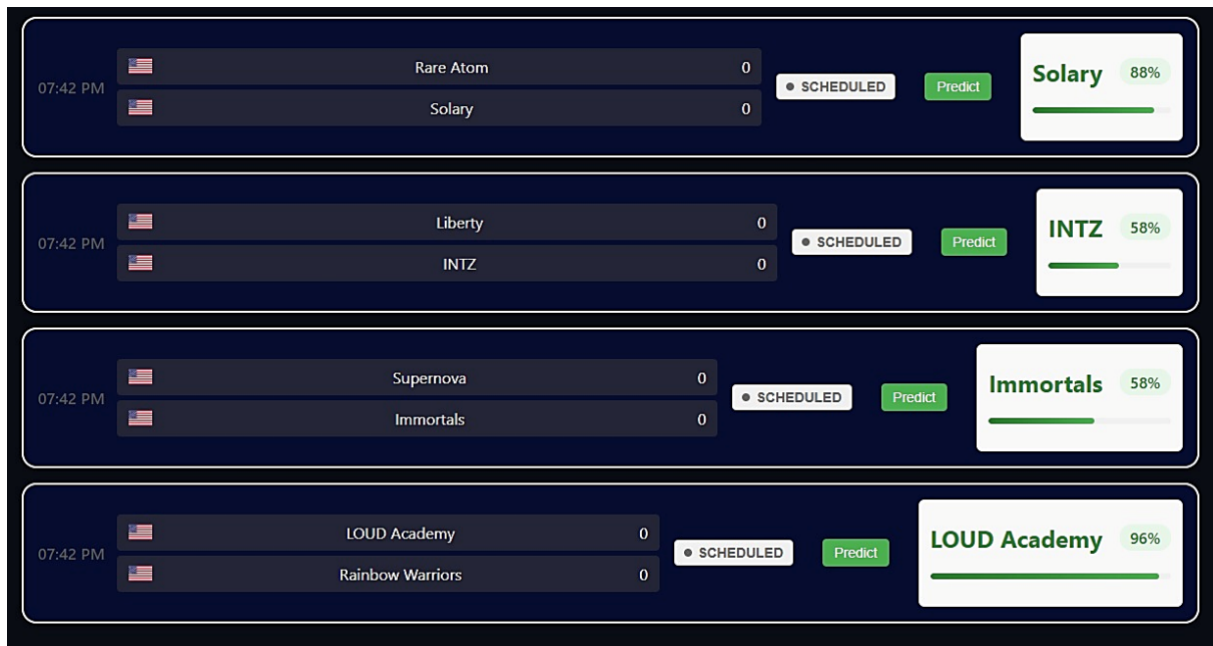


Figure 7: Tournament bracket + Prediction

2.1 Côté Admin

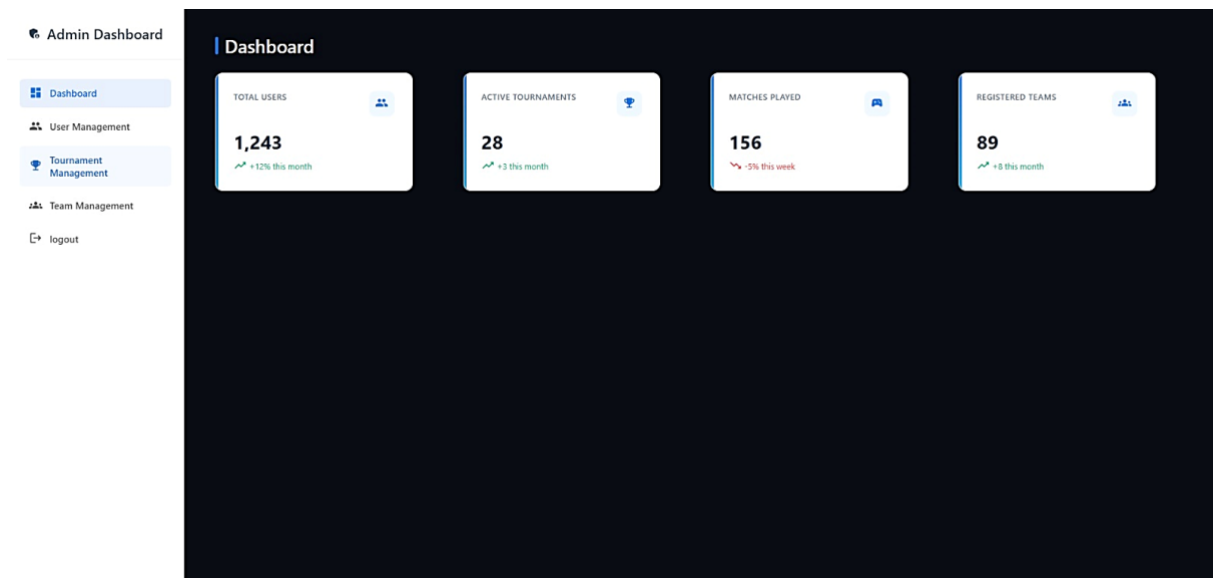


Figure 8: Dashboard

Admin Dashboard

Dashboard
User Management
Tournament Management
Team Management
Logout

User Management

Search users

ID	Username	Email	Role	Actions
1	admin	admin@example.com	ADMIN	<button>Edit</button> <button>Delete</button>
2	player1	player1@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
4	player2	player1@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
11	JohnQT	player1@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
12	coach1	coach@example.com	COACH	<button>Edit</button> <button>Delete</button>
13	zekken	player1@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
14	zellsis	player1@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
15	pppp	player1@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
16	ggg	playerpppp@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
17	dddd	playerpppp@example.com	PLAYER	<button>Edit</button> <button>Delete</button>
18	???	playernnnn@example.com	PLAYER	<button>Edit</button> <button>Delete</button>

Figure 9:User Manager

Edit User

Username

player1

Email

player1@example.com

Role

Coach

Cancel

Save Changes

Figure 10:Edit User

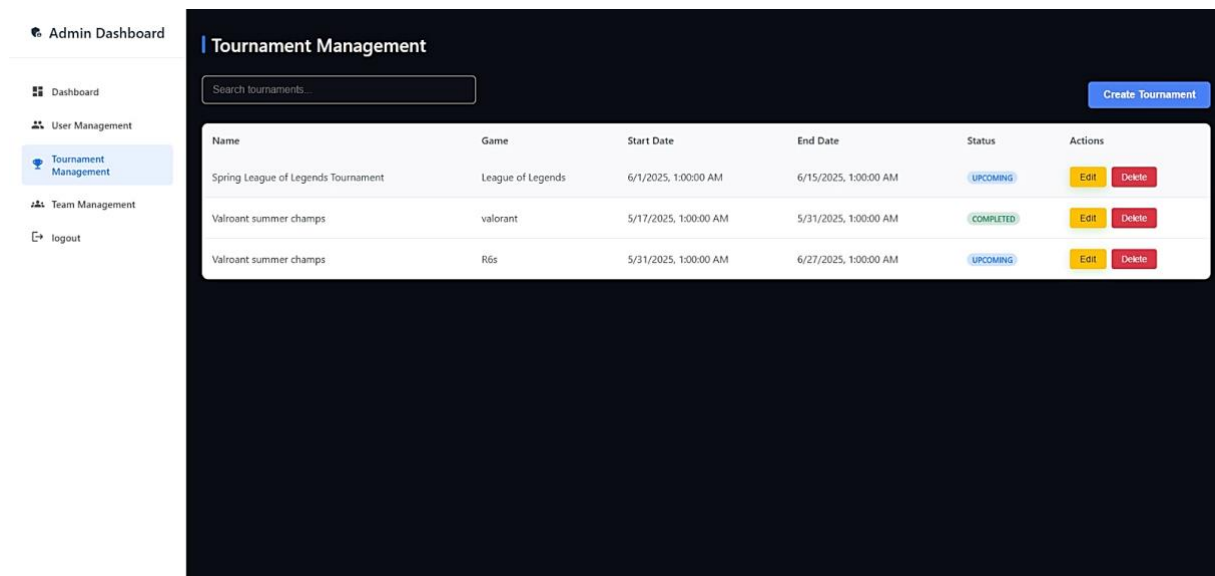


Figure 11: Tournament Manager

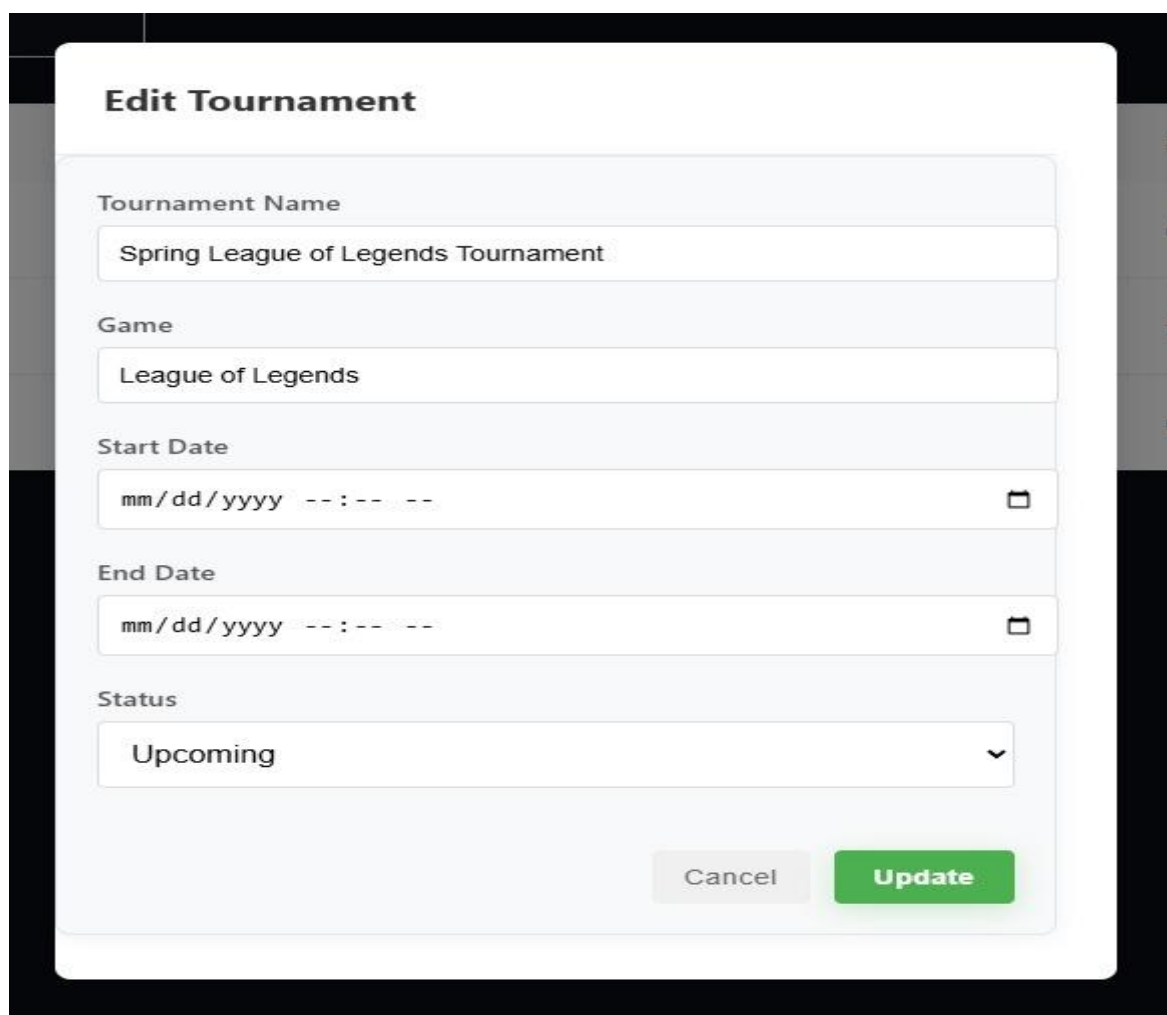


Figure 12:Edit tournament

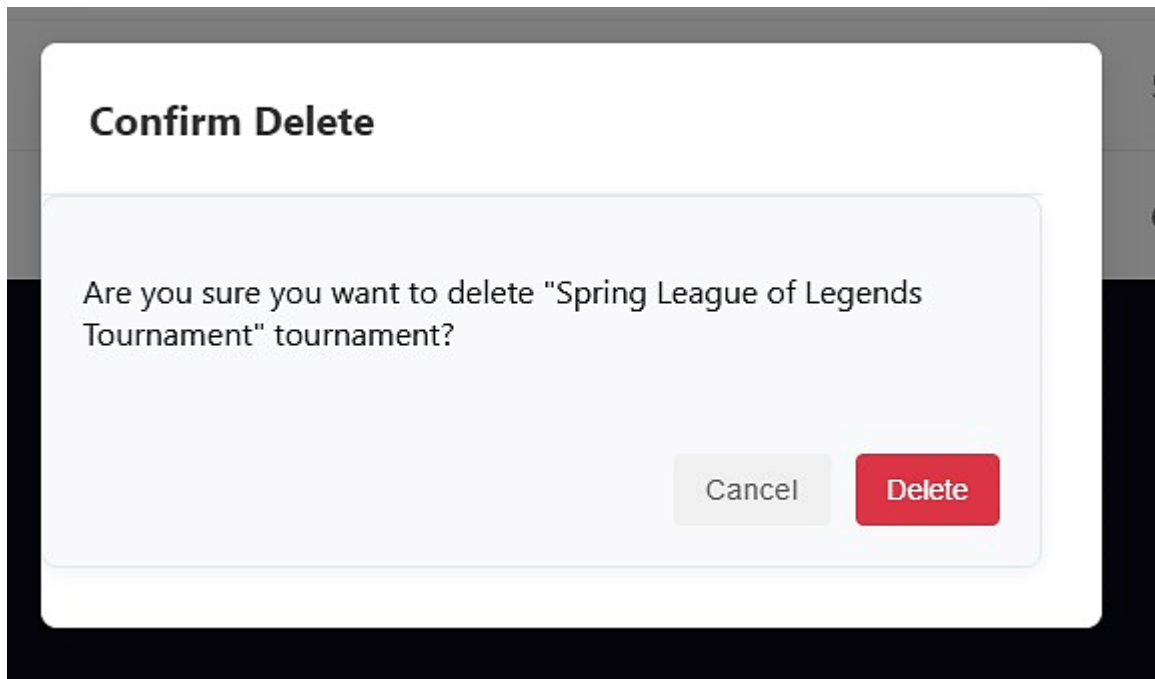


Figure 13:Delete tournament

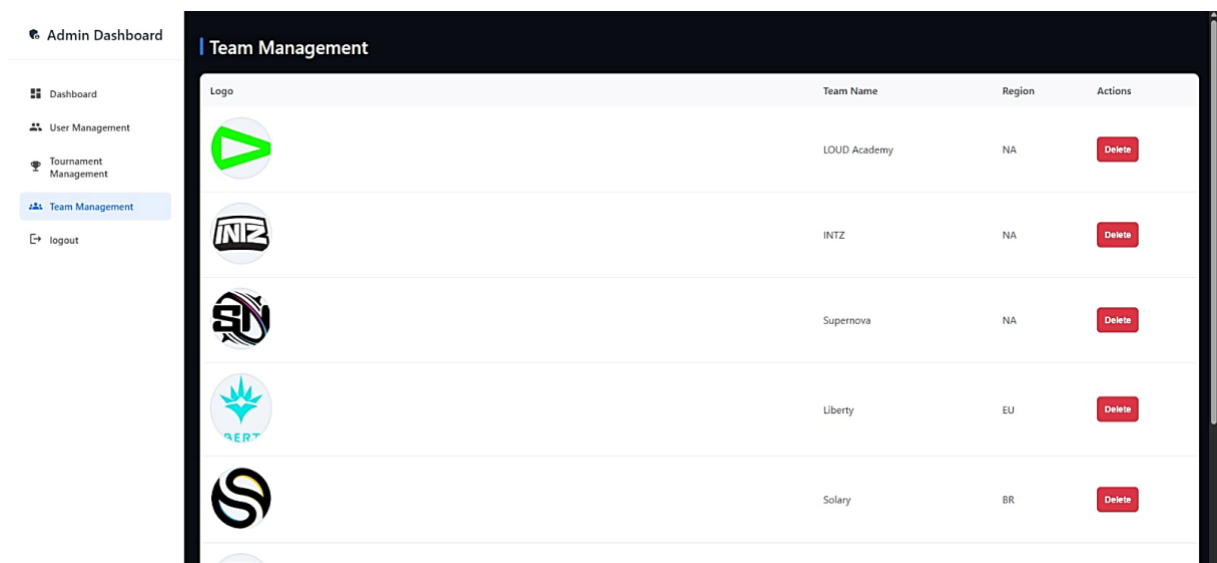
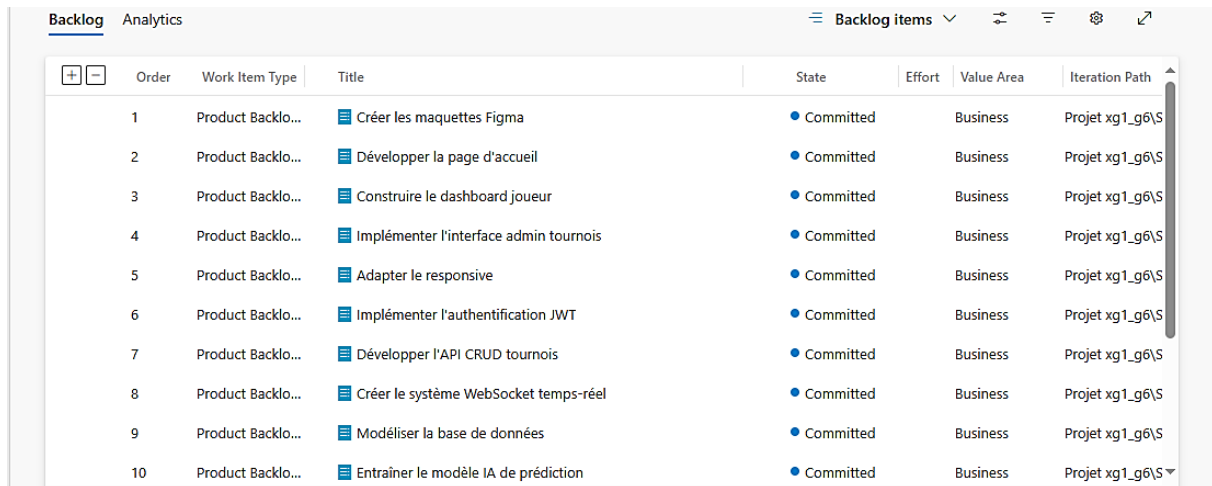


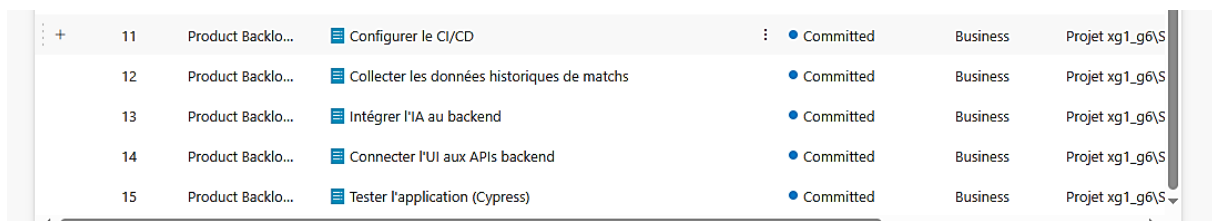
Figure 14:Team Management

2. Backlogs initials



	Order	Work Item Type	Title	State	Effort	Value Area	Iteration Path
	1	Product Backlo...	Créer les maquettes Figma	Committed		Business	Projet xg1_g6\S
	2	Product Backlo...	Développer la page d'accueil	Committed		Business	Projet xg1_g6\S
	3	Product Backlo...	Construire le dashboard joueur	Committed		Business	Projet xg1_g6\S
	4	Product Backlo...	Implémenter l'interface admin tournois	Committed		Business	Projet xg1_g6\S
	5	Product Backlo...	Adapter le responsive	Committed		Business	Projet xg1_g6\S
	6	Product Backlo...	Implémenter l'authentification JWT	Committed		Business	Projet xg1_g6\S
	7	Product Backlo...	Développer l'API CRUD tournois	Committed		Business	Projet xg1_g6\S
	8	Product Backlo...	Créer le système WebSocket temps-réel	Committed		Business	Projet xg1_g6\S
	9	Product Backlo...	Modéliser la base de données	Committed		Business	Projet xg1_g6\S
	10	Product Backlo...	Entraîner le modèle IA de prédiction	Committed		Business	Projet xg1_g6\S

Figure 15: Caoture 1 Backlog

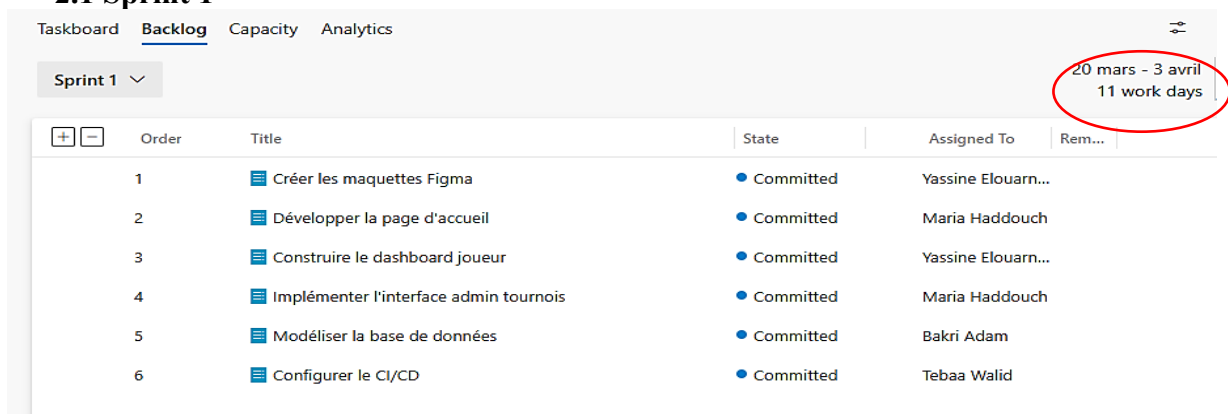


	11	Product Backlo...	Configurer le CI/CD	Committed		Business	Projet xg1_g6\S
	12	Product Backlo...	Collecter les données historiques de matchs	Committed		Business	Projet xg1_g6\S
	13	Product Backlo...	Intégrer l'IA au backend	Committed		Business	Projet xg1_g6\S
	14	Product Backlo...	Connecter l'UI aux APIs backend	Committed		Business	Projet xg1_g6\S
	15	Product Backlo...	Tester l'application (Cypress)	Committed		Business	Projet xg1_g6\S

Figure 16: Capture 2 Backlog

3. Plan des sprints

2.1 Sprint 1



	Order	Title	State	Assigned To	Rem...
	1	Créer les maquettes Figma	Committed	Yassine Elouarn...	
	2	Développer la page d'accueil	Committed	Maria Haddouch	
	3	Construire le dashboard joueur	Committed	Yassine Elouarn...	
	4	Implémenter l'interface admin tournois	Committed	Maria Haddouch	
	5	Modéliser la base de données	Committed	Bakri Adam	
	6	Configurer le CI/CD	Committed	Tebaa Walid	

Figure 17: Sprint 1

3.1 Sprint 2

Taskboard Backlog Capacity Analytics						
Sprint 2						
4 avril - 17 avril 10 work days						
+ -	Order	Title	State	Assigned To	Rem...	
	1	Adapter le responsive	Committed	Yassine Elouarn...		
	2	Implémenter l'authentification JWT	Committed	Tebaa Walid		
	3	Développer l'API CRUD tournois	Committed	Tebaa Walid		
	4	Entraîner le modèle IA de prédiction	Committed	Ayyoub Raji		
	5	Collecter les données historiques de matchs	Committed	Yassine Elouarn...		
	6	Connecter l'UI aux APIs backend	Committed	Maria Haddouch		

Figure 18:Sprint 2

4.1 Sprint 3

Taskboard Backlog Capacity Analytics						
Sprint 3						
18 avril - 1 mai 10 work days						
+ -	Order	Title	State	Assigned To	Rem...	
	1	Créer le système WebSocket temps-réel	Committed	Tebaa Walid		
	2	Intégrer l'IA au backend	Committed	Ayyoub Raji		

Figure 19:Sprint 3

5.1 Sprint 4

Taskboard Backlog Capacity Analytics						
Sprint 4						
2 mai - 9 mai 6 work days						
+ -	Order	Title	State	Assigned To	Rem...	
	1	Tester l'application (Cypress)	Committed	Ayyoub Raji		

Figure 20:Sprint 4

4. Gestion du code

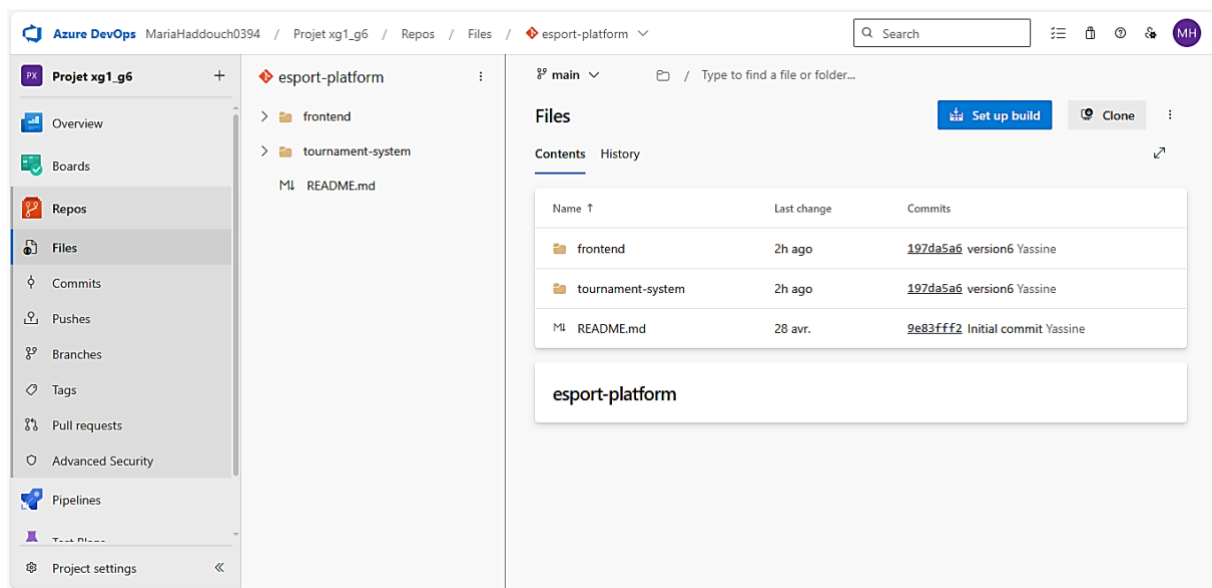


Figure 21:Files

5. Intégration Continue (CI) et Déploiement Continu (CD)

1.1 Pipeline

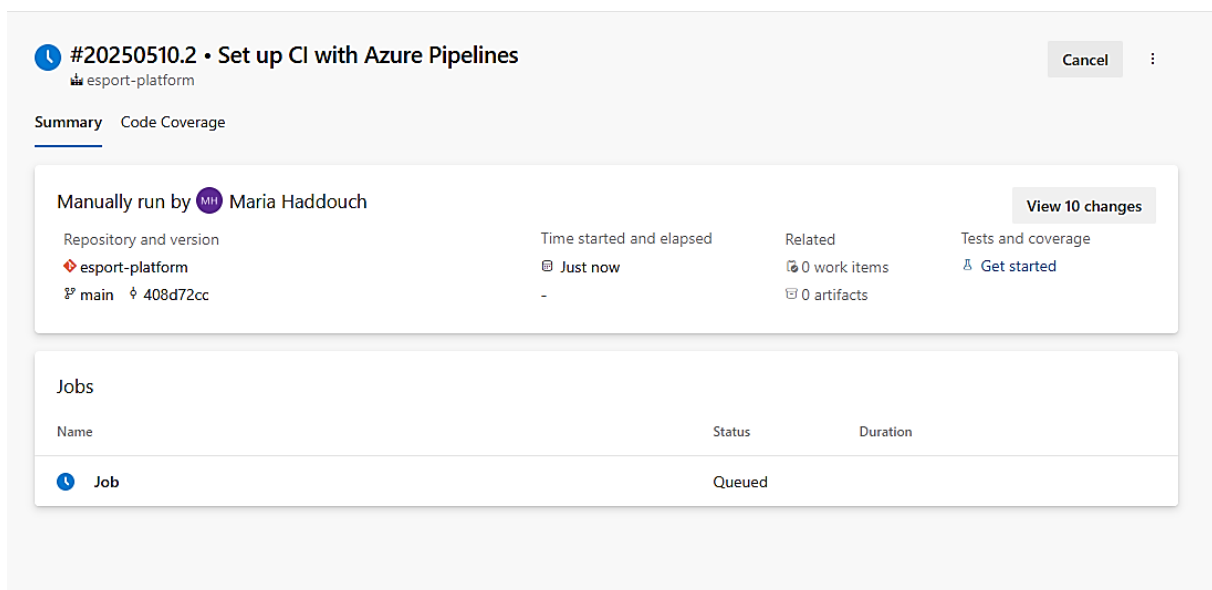


Figure 22:Pipeline

2.1 Agent pool

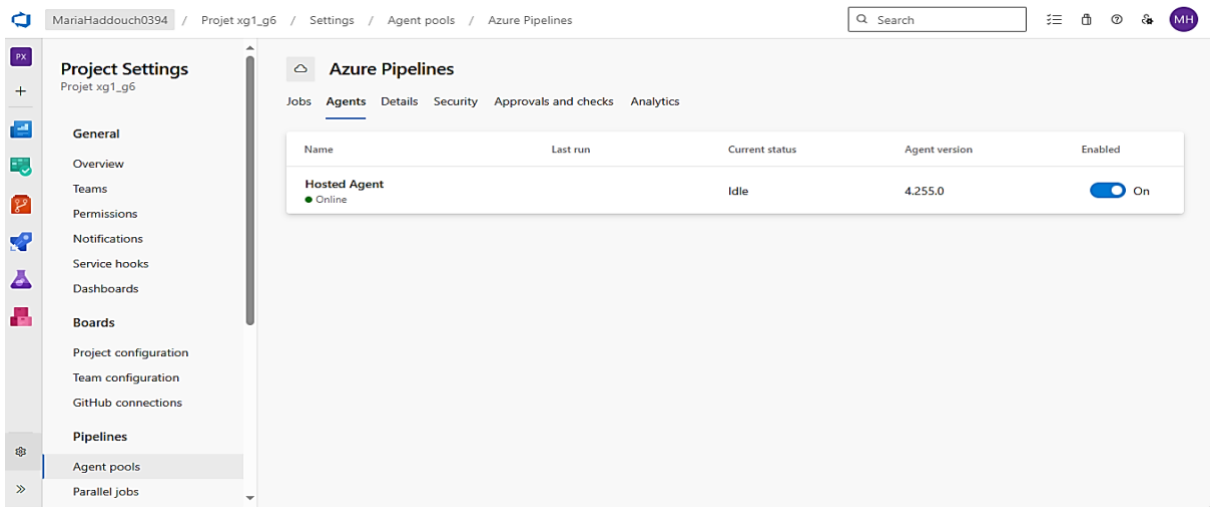


Figure 23:Agent pool

3.1 Fichier azure-pipelines.yml :

► Le fichier azure-pipelines.yml configure le pipeline d'intégration continue (CI) pour le projet student_management_system-last. Il automatise les étapes nécessaires à la construction de l'application sur Azure DevOps, telles que l'installation de Node.js en version 20.x, l'installation des dépendances via npm install, puis l'exécution de la commande npm run build. Ce pipeline est déclenché automatiquement à chaque push effectué sur la branche cible inscription-client.

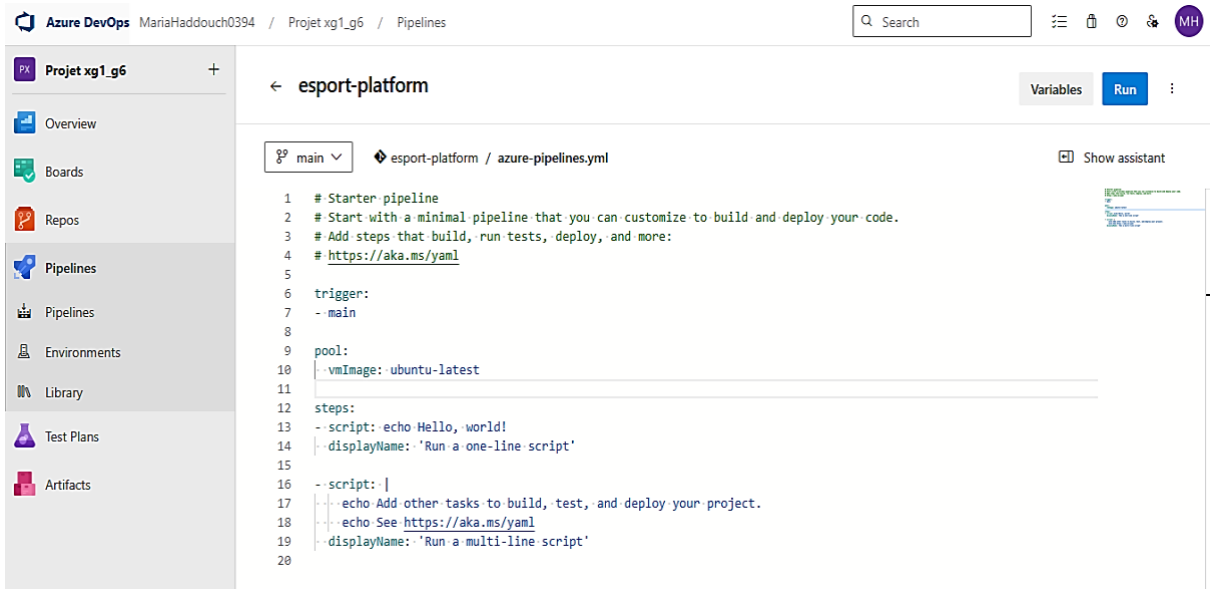


Figure 24:Fichier azure-pipelines.yml

Conclusion générale

Le développement de cette plateforme de gestion de tournois e-sport s'inscrit pleinement dans les mutations profondes que connaît l'univers du jeu vidéo compétitif. En intégrant des fonctionnalités avancées, cette solution ne se contente pas de répondre aux attentes actuelles, mais anticipe également les besoins futurs d'un secteur en constante évolution. Grâce à l'automatisation des tâches clés — telles que la gestion des inscriptions, la planification des matchs, l'arbitrage en temps réel et le classement dynamique — la plateforme apporte un gain de temps considérable et une fiabilité accrue aux organisateurs, tout en offrant une expérience fluide et engageante aux joueurs et spectateurs.

L'un des apports les plus novateurs du projet réside dans l'introduction de l'intelligence artificielle comme levier d'analyse et de valorisation des données. En exploitant des bases de données issues de sources spécialisées dans l'e-sport (Riot Games API, Tracker.gg, VLR.gg) et en appliquant des algorithmes de machine learning comme XGBoost, Random Forest ou encore des modèles de deep learning (TensorFlow, PyTorch), la plateforme est capable de générer des prédictions de résultats pertinentes et contextualisées. Ces estimations ne constituent pas seulement un atout pour le suivi des compétitions, mais ouvrent également la voie à de nouveaux usages : analyses statistiques poussées, création de contenus interactifs, prévention de comportements anormaux, etc.

Sur le plan technique, l'architecture retenue témoigne d'une volonté d'excellence : un backend robuste avec Spring Boot, un frontend moderne en React.js, une base de données fiable avec MySQL, et une communication en temps réel assurée par WebSocket pour une réactivité optimale. L'intégration de pipelines CI/CD permet de garantir un déploiement fluide et sans interruption, tandis que les tests automatisés via Cypress assurent la stabilité et la qualité du code à chaque itération. Ce socle technique solide permet d'envisager une montée en charge progressive, tout en assurant la sécurité et la maintenabilité du système.

Enfin, la méthodologie agile adoptée tout au long du projet permet une évolution incrémentale, maîtrisée et centrée sur les besoins des utilisateurs. Chaque sprint constitue une étape stratégique, allant de la conception initiale des interfaces à l'intégration des fonctionnalités d'intelligence artificielle, en passant par la modélisation des données, le prototypage et les tests finaux. Ce découpage méthodologique garantit une vision claire, une capacité d'adaptation rapide et un pilotage rigoureux du développement.

En conclusion, ce projet dépasse largement la simple réalisation d'un outil de gestion : il ambitionne de devenir une référence technologique au sein de l'écosystème e-sportif. Par sa richesse fonctionnelle, son architecture moderne, sa capacité d'adaptation et son intelligence intégrée, il propose une nouvelle manière de concevoir, de suivre et de vivre les compétitions e-sport. Il s'inscrit ainsi dans une logique d'innovation durable, au service de la performance, de l'expérience utilisateur et de la transformation digitale de l'e-sport.