

Table des matières

Table des matières	1
Liste des figures	2
Liste des tableaux	3
Introduction générale.....	4
Chapitre 1 : Présentation de projet	5
Introduction :.....	5
I. Etude de l'existant :.....	5
II. Présentation de notre solution	7
III. Méthodologie adoptée :.....	8
Conclusion.....	8
Chapitre 2 : Spécification de besoin.....	9
Introduction	9
I. Etude de besoin :	9
II. Les diagrammes de cas d'utilisation :	11
Conclusion.....	12
Chapitre 3 : Conception.....	13
Introduction	13
I. Architecture de la solution :	13
Conclusion.....	13
Chapitre 4 : Réalisation	14
Introduction :.....	14
I. Environnement de développement	14
II. Implémentation de l'application :.....	15
Conclusion :.....	18
Conclusion générale	19

Liste des figures

Figure 1: Meilleur joueur du moi par CocaCola	5
Figure 2: Meilleur joueur de 'League of Legends' lancé par RiotGames	6
Figure 3: Diagramme de cas d'utilisation globale	12
Figure 4: Diagramme de classes	13
Figure 5: Interface d'authentification	16
Figure 4: Interface d'accueil	16
Figure 4: Interface d'une compétition	17
Figure 4: Interface de participation	17
Figure 4: Interface mes compétitions	18

Liste des tableaux

Tableau 1: Critique de l'existant.....	7
Tableau 2: PC1	14
Tableau 3: PC2	14

Introduction générale

La compétition désigne une attitude de rivalité et de concurrence de plusieurs membres dans la recherche d'un même objectif.

Depuis le début de l'histoire, les compétitions ont été l'un des meilleures sources de défoulement et de la présentation des talents.

L'application que nous présentons dans ce rapport a pour but de structurer et organiser le processus de création, diffusion et vote dans n'importe quelle compétition.

Chapitre 1 : Présentation de projet

Introduction :

Dans le chapitre présent, nous découvrirons le cadre général de notre projet en suivant ces étapes :

- Etude de l'existant
- Méthodologie adoptée

I. Etude de l'existant :

Cette étude permet une compréhension approfondie du domaine d'étude, elle consiste à comprendre et à analyser les applications existantes, ensuite, elle comprend une comparaison entre ces applications ainsi que de décrire la solution à proposer.

1. Facebook

Dans nos jours la méthode la plus populaire utilisée par la majorité des internautes pour créer et gérer des compétitions en ligne c'est la plateforme Facebook.

Comme illustré dans la figure 1, une simple publication partagée sur une page Facebook et destinée au tous les internautes est suffisante pour créer sa propre compétition.



Figure 1: Meilleur joueur du moi par CocaCola

2. Twitter

De même, et suivant le même principe du Facebook, Twitter aussi a sa propre option de sondage qui permet de lancer un vote ouvert à tous pour exprimer leurs propres opinions.



Figure 2 Meilleur joueur de 'League of Legends' lancé par RiotGames

3. Critique de l'existant :

Nom application	Avantages	Inconvénients
Facebook	<ul style="list-style-type: none">• Facile à utiliser• un grand nombre d'utilisateurs déjà	<ul style="list-style-type: none">• Limitée• Impossible de participer par des vidéos

	existant dans la plateforme	<ul style="list-style-type: none"> • Impossible de définir une date limite • Difficile à gérer plusieurs compétitions au même temps
Twitter	<ul style="list-style-type: none"> • Facile à utiliser • Un grand nombre d'utilisateurs déjà existant dans la plateforme 	<ul style="list-style-type: none"> • Limiter à un nombre réduit de mots par poste • Impossible de voir le nombre des participants • Difficile à gérer plusieurs compétitions • Pour voir les compétitions existantes il faut consulter la page du créateurs (difficile à trouver)

Tableau 1: Critique de l'existant

II. Présentation de notre solution

Suivant les insuffisances des solutions existantes, nous proposons d'apporter une solution simple et riche qui prend en considération toutes les exigences des utilisateurs et remède aux lacunes dégagées. Ainsi, notre projet doit être conforme à l'objectif de développement d'une application conviviale et ergonomique. Ainsi, nous développons une application web nommée «Vote-ly » qui permettra de:

- À tous les utilisateurs de consulter les compétitions
- Aux utilisateurs inscrit de déposer une demande pour crée une compétition.
- À l'organisateur de gérer la compétition (consultation, ajout, modification, suppression).
- À tous les utilisateurs de consulter les Participants.
- Les utilisateurs inscrit de voter à un participant.
- À tous les utilisateurs de consulter les résultats des compétitions.
- Administrer l'application à travers un panel. (Gérer les users et les compétitions)

III. Méthodologie adoptée :

Dans cette section, nous citerons la méthodologie qui sera utilisée dans le développement de cette plateforme en justifiant la raison de notre choix.

1. Choix de la méthode :

Notre choix s'est fixé sur la méthodologie 2TUP pour les raisons suivantes : La méthodologie 2TUP autorise les changements de l'aspect fonctionnel et technique de manière incrémentale. La méthodologie 2TUP est convenable pour la programmation orientée objet.

2. Le langage de modélisation

Pour pouvoir faciliter la création de l'application, nous utiliserons le langage de modélisation UML.

Le Langage de Modélisation Unifié (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

Conclusion

Dans ce chapitre introductif, nous avons présenté le projet qui nous a été confié. Nous avons aussi présenté la méthodologie que nous allons utiliser.

Dans le chapitre suivant, nous découvrirons la phase de préparation de ce projet qui consistera en l'analyse et la spécification des besoins fonctionnels et non fonctionnels de notre application tout en élaborant les diagrammes nécessaires pour cette phase.

Chapitre 2 : Spécification de besoin

Introduction

Dans ce chapitre, il s'agit de présenter les besoins de l'application. Ces besoins vont présenter les fonctionnalités dont nous avons besoin. Il s'agit d'une étape cruciale dans la réalisation d'une application donnée. Le futur d'un logiciel dépend beaucoup de cette phase, elle nous permet d'éviter le développement d'une application non satisfaisante.

I. Etude de besoin :

Les besoins sont divisés en deux catégories, à savoir les besoins fonctionnels et les besoins non fonctionnels

1. Besoins fonctionnels

Les besoins fonctionnels d'un système décrivent à quoi sert le système à développer. Tenant compte de la nature de l'application, on distingue les besoins par acteur :

- Administrateur : l'application doit permettre à l'administrateur
 - gérer les demandes de création des compétitions.
 - gérer les utilisateurs de l'application
 - gérer les compétitions
- Internaute : l'application doit permettre au Internaute
 - Créer des comptes
 - Consulter les compétitions déjà existantes
- Utilisateurs inscrits : l'application doit permettre à l'Utilisateurs inscrits
 - Participer dans des compétitions
 - Organiser des compétitions
 - Voter pour des participants
 - Consulter des compétitions

2. Besoins non fonctionnels :

Les besoins non fonctionnels spécifient les critères de qualité concernant le comportement du système. C'est l'ensemble des contraintes internes et externes qui doivent satisfaire les besoins de l'utilisateur. Nous distinguons les besoins non fonctionnels suivants :

❖ Performance : La performance de l'application dépend en grande partie de la complexité du code implémenté, des serveurs d'application utilisés et du débit de la ligne de connexion pour le chargement initial des données. Ce temps dépend généralement de la conception de la base de données.

❖ Sécurité : Besoins de sécurité qui définissent les niveaux d'accès possibles pour les utilisateurs du système et les systèmes externes.

❖ L'ergonomie et la convivialité : L'utilisateur est en contact direct et unique avec l'interface. Pour lui, l'application est une interface à manipuler, qui revêt une grande importance dans la conception de notre application, laquelle repose sur les critères suivants :

- Clarté et bonne lisibilité des boutons, des labels et des cartes géographiques.
- Tous les boutons, labels, etc., expriment les actions à exécuter.
- Interface : ergonomique et proche de la réalité du domaine d'application.
- Navigation facile et efficace.

II. Les diagrammes de cas d'utilisation :

1. Présentation des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. On distingue 3 types d'acteurs :

- Visiteur :
 - C'est un acteur qui peut créer un compte et peut consulter les compétitions existantes
- Admin
 - C'est un acteur qui peut gérer les comptes des users et peut gerer les compétitions
- Utilisateur :
 - C'est un acteur qui peut créer, participer et voter dans des compétitions,

2. Description des cas d'utilisation

- Diagrammes des cas d'utilisations globale :

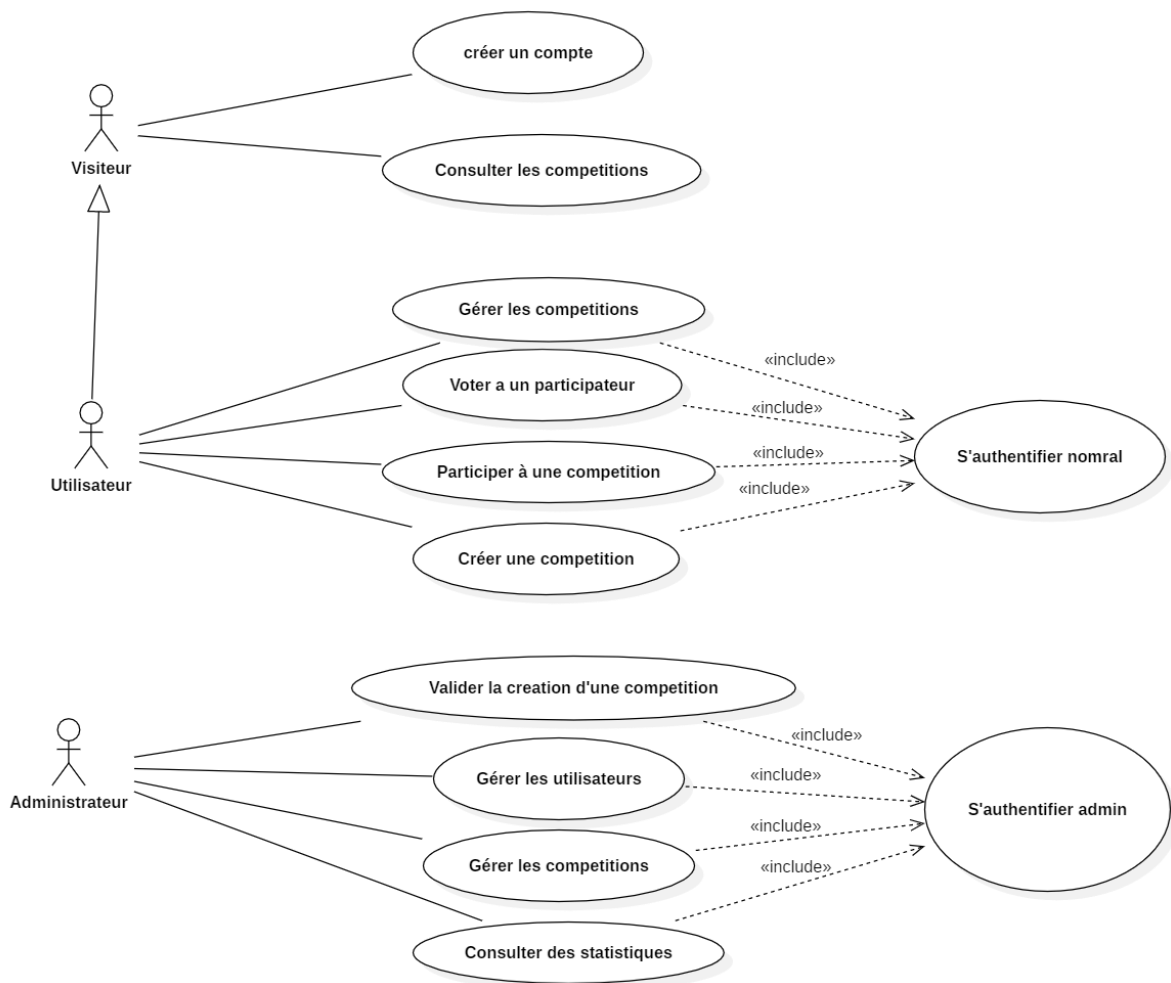


Figure 3 : Diagramme de cas d'utilisation globale

Conclusion

La spécification des besoins nous a permis d'avoir une vision plus claire du sujet et une compréhension plus profonde des tâches à réaliser. Elle mené également à prévoir les problèmes à rencontrer et chercher les solutions permettant de les contourner.

Nous avons essayé tout au long de ce chapitre de bien présenter les besoins fonctionnels et non fonctionnels du sujet et les différents scenarios d'utilisation de l'application.

Chapitre 3 : Conception

Introduction

Dans cette partie nous abordons la définition de l'architecture technique qui consiste à faire les choix de technologies et d'organisation de composants logiciels les plus adaptés aux besoins et aux contraintes de l'organisation d'accueil. Ces choix sont ensuite relayés au sein de notre projet, guidant la conception et permettant la transformation d'un modèle fonctionnel en application performante et robuste.

I. Architecture de la solution :

1. Diagramme de classe :

Le figure ci-dessous représente le diagramme de classe

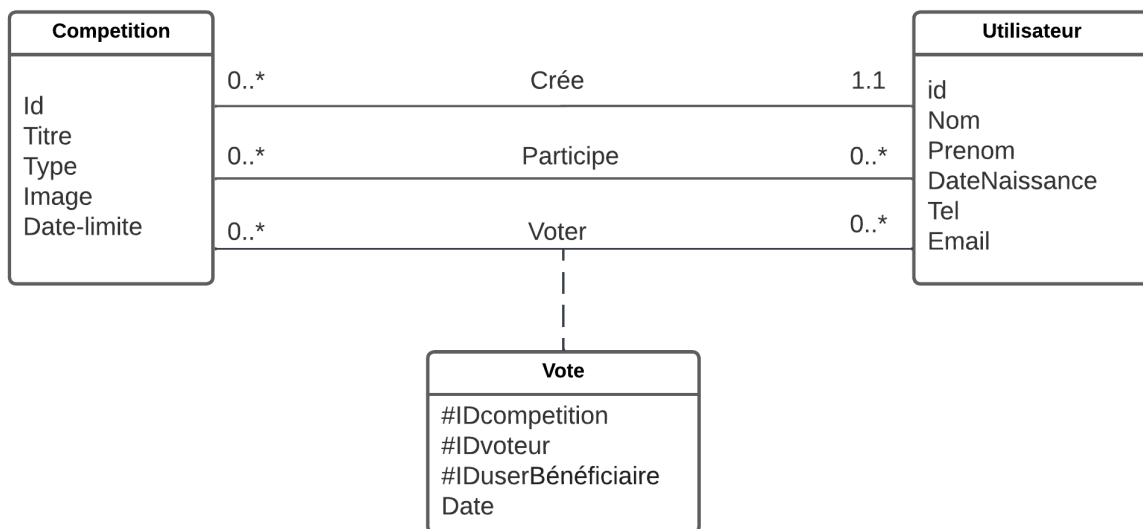


Figure 1: Diagramme de classes

Conclusion

L'activité de la conception est indispensable afin de faciliter la compréhension de notre système, qui ébauche vers l'activité réalisation et implémentation

Chapitre 4 : Réalisation

Introduction :

Après avoir achevé la partie conception de l'application, il s'agit de présenter la partie réalisation et mise en œuvre de notre travail. Ce chapitre constitue le dernier volet de ce rapport et la dernière étape du processus unifiée qui a pour objectif d'exposer le travail achevé. Dans un premier lieu, nous présentons l'environnement de travail, tout en justifiant les différents choix technologiques effectués, ensuite nous présentons les différentes interfaces de l'application.

I. Environnement de développement

1. Environnement matériel

Les caractéristiques des machines qu'on a utilisé pour la réalisation de ce projet :

- PC portable Asus Vivobook Max X540UB

Processeur	Intel Core i7-8250U
RAM	8 Go DDR4
Type de système	A base de x64
Nom de système d'exploitation	Microsoft Windows 10 professionnel

Tableau 2: PC1

- PC portable Lenovo Y520

Processeur	Intel Core i5-8250U
RAM	8 Go DDR4
Type de système	A base de x64
Nom de système d'exploitation	Microsoft Windows 10 professionnel

Tableau 3: PC2

2. Environnement logiciel :

➤ Visual studio code :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

➤ Pycharm :

PyCharm est un environnement de développement intégré utilisé pour programmer en Python.

Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django et Flask.

3. Choix des outils de développement

Notre application nécessite l'utilisation de deux technologies. Nous détaillons dans ce qui suit les différents outils de développement utilisés au cours de la réalisation de notre projet.

➤ Flask :

Flask est un petit framework web Python léger, qui fournit des outils et des fonctionnalités utiles qui facilitent la création d'applications web en Python. Il offre aux développeurs une certaine flexibilité et constitue un cadre plus accessible pour les nouveaux développeurs puisque vous pouvez construire rapidement une application web en utilisant un seul fichier Python. Flask est également extensible et ne force pas une structure de répertoire particulière ou ne nécessite pas de code standard compliqué avant de commencer.

II. Implémentation de l'application :

1. Présentation et description des interfaces

Dans cette section nous présentons quelques interfaces homme machine de la réalisation finale de l'application.

➤ Interface d'authentification (admin)

Cette figure ci-dessous représente l'interface d'authentification de l'utilisateur:

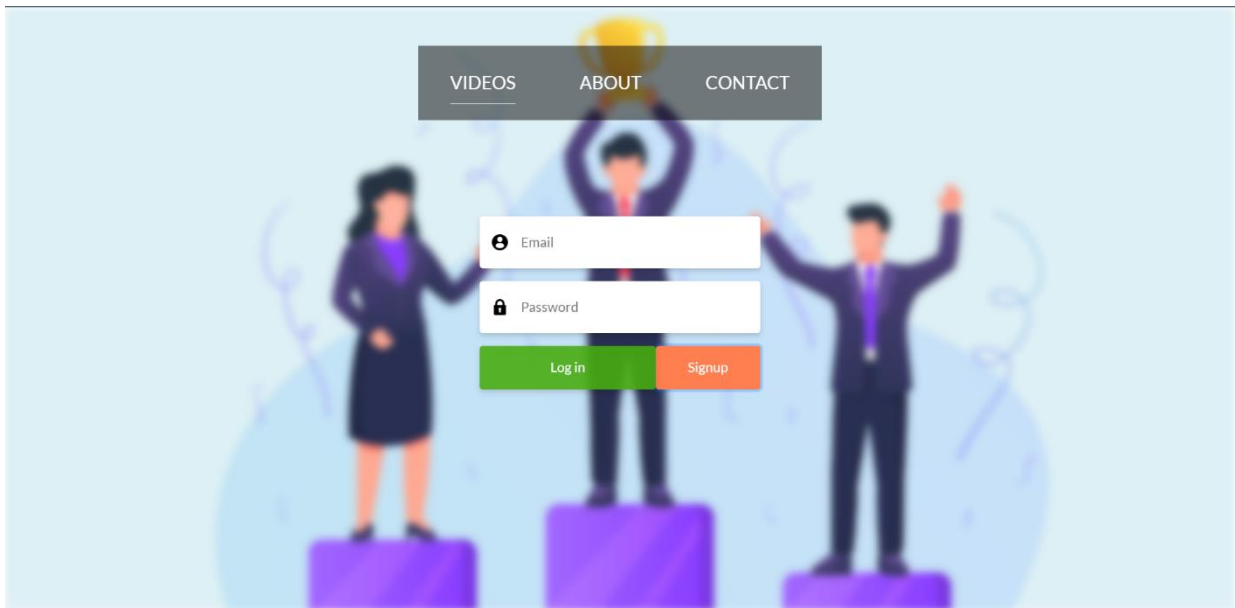


Figure 5 : l'interface d'authentification de l'utilisateur

➤ Interface d'accueil

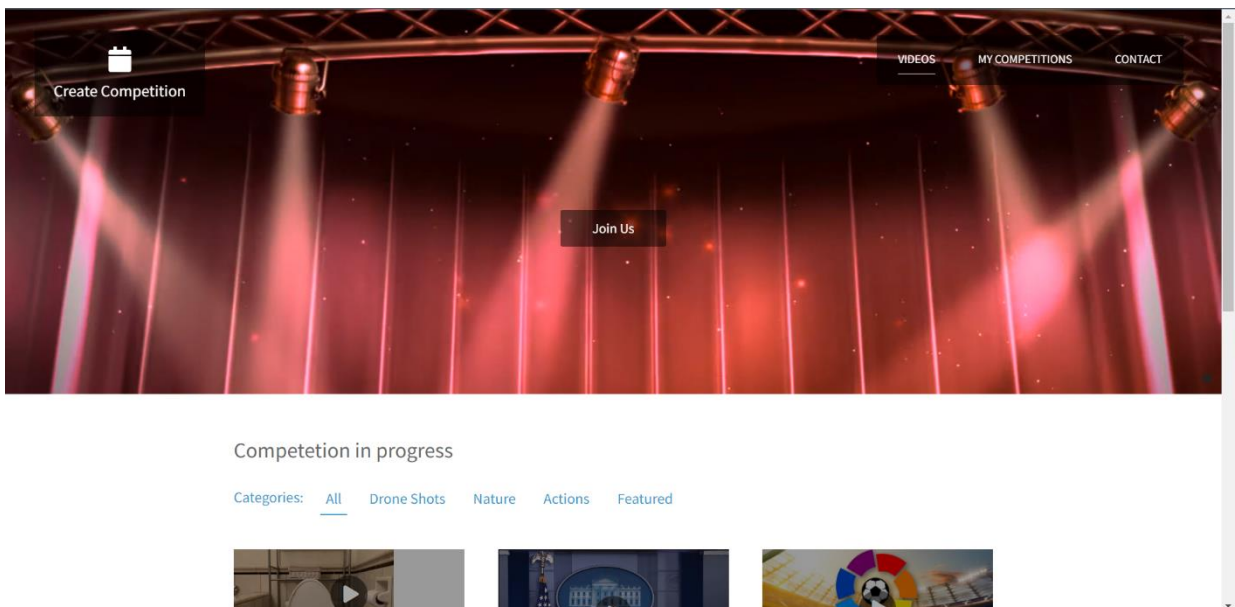


Figure 6 : Interface d'accueil

➤ Interface d'accueil

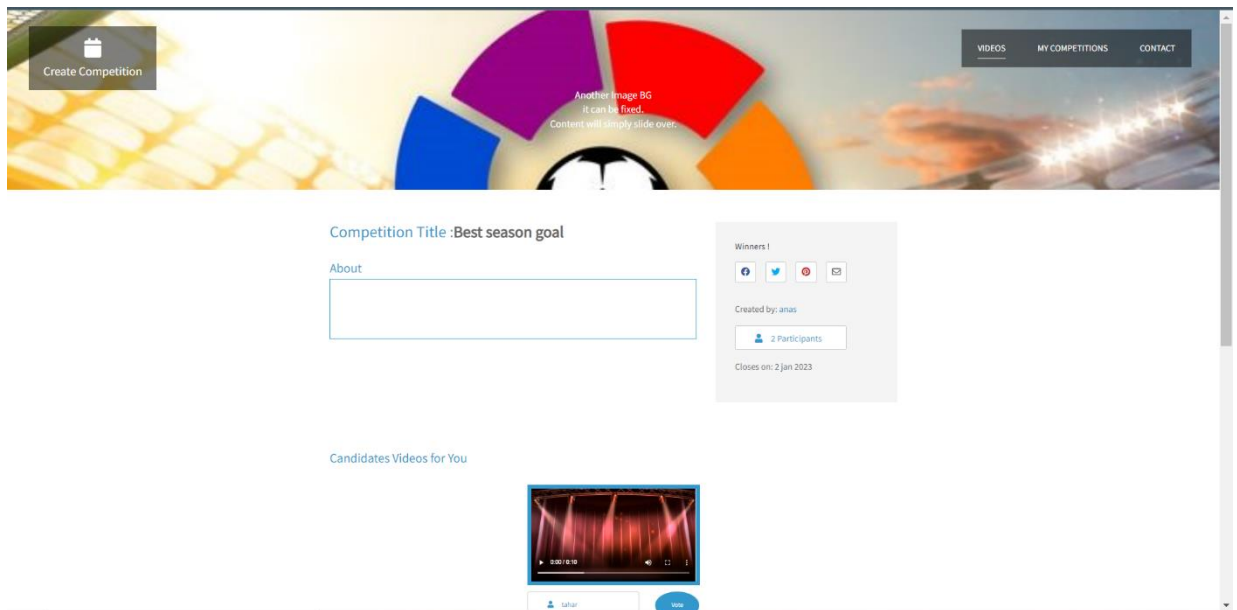


Figure 7: Interface d'une competition

➤ Interface de participation

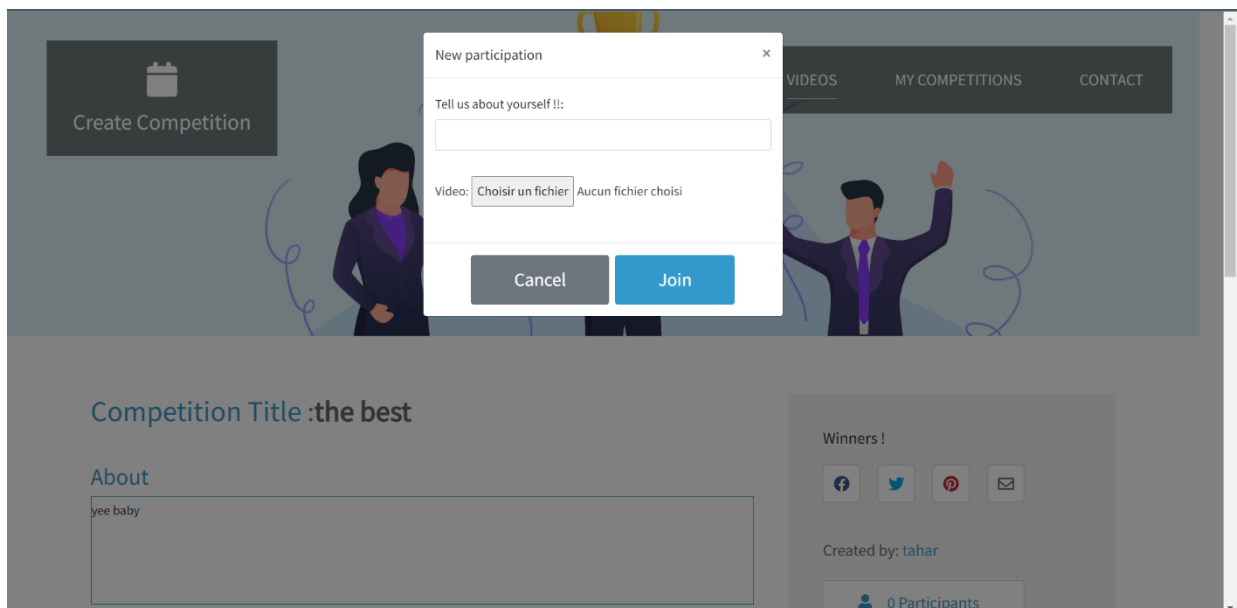


Figure 8 : Interface participation

➤ Interface mes compétitions

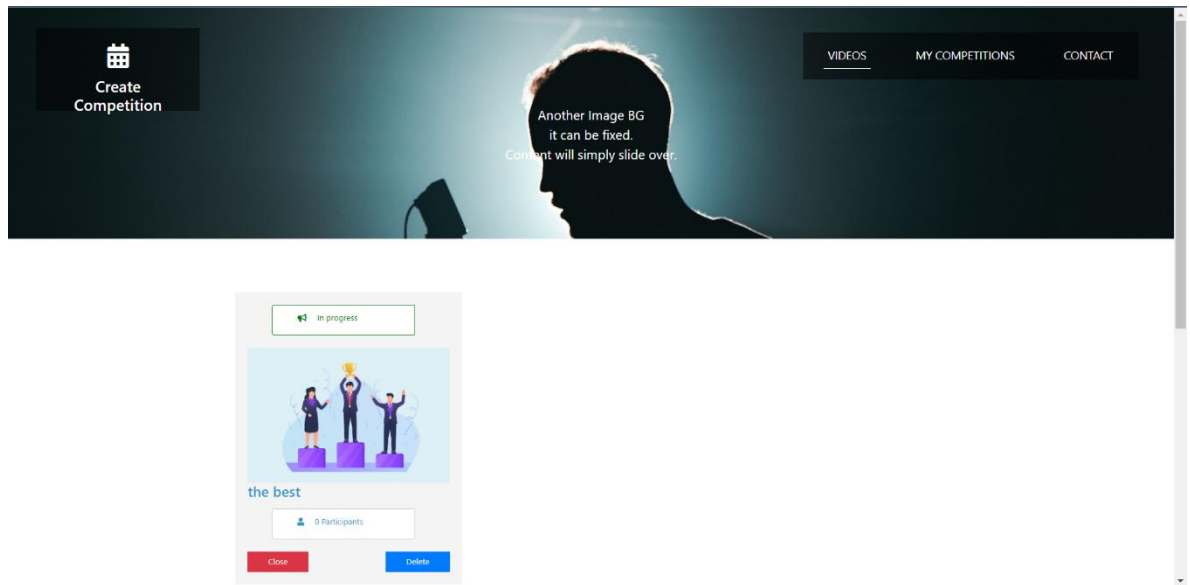


Figure 9 : Interface mes compétitions

Conclusion :

Durant ce chapitre, nous avons commencé par une description détaillée de l'environnement de travail (matérielle, logiciel) et nous avons terminé par la description de la réalisation en Montrant quelques interfaces graphiques de l'application.

Conclusion générale

Le travail s'articule autour de la conception et la réalisation d'une application qui permet de créer, participer et voter dans des compétitions.

Ce rapport décrit le bilan de notre travail, nous avons procédé dans un premier temps à une présentation générale du projet pour bien comprendre la problématique et engendrer une solution optimale qui répond aux besoins des créateurs des compétitions.

Par la suite, nous avons passé à la spécification des besoins qui nous a permis de définir les besoins fonctionnels et non fonctionnels de notre application. Ensuite, nous avons passé à la conception de l'application web en suivant la méthodologie 2TUP. Puis, nous avons passé à la réalisation.

Ce travail était très intéressant puisqu'il nous a permis de découvrir un nouveau domaine de travail et de s'éloigner des projets traditionnels. Il nous a permis d'approfondir nos connaissances dans les bonnes pratiques de programmation.

Pour conclure, notre travail ne s'arrête pas à ce stade. En effet, plusieurs fonctionnalités peuvent être ajoutées et améliorées comme le partage et l'adaptabilité avec les réseaux sociaux.

Nous pouvons citer également l'ajout d'autre langues comme le français et arabe pour que l'application vise chaque destination avec sa langue maternelle.