

DESCRIPTION D'UNE RÉALISATION PROFESSIONNELLE			N° réalisation : 1
Nom, prénom : Namias, Gabriel			N° candidat :
Épreuve ponctuelle	Contrôle en cours de formation	X	Date : ...25 / ..03.. / ..2025...
Organisation support de la réalisation professionnelle Projet – Jo Ticket			
Intitulé de la réalisation professionnelle JO Ticket			
Période de réalisation : Janvier – Mars 2025 Lieu : Esiee-IT			
Modalité : X Seul(e) En équipe			
Compétences travaillées Concevoir et développer une solution applicative Assurer la maintenance corrective ou évolutive d'une solution applicative Gérer les données			
Conditions de réalisation¹ (ressources fournies, résultats attendus) Application de réservation de ticket pour les JO -Api en Django de gestion des matchs avec une interfaces admin pour modifier les matchs Relié à une base de données -Application mobile de réservations de tickets -Application mobile pour scanner les tickets			
Description des ressources documentaires, matérielles et logicielles utilisées² Outils : Visual Studio Code, Github, WampServer Languages: Python (Django), HTML, CSS, JS, SQL			
Modalités d'accès aux productions³ et à leur documentation⁴ Github : https://github.com/esiee-it-slam-2025/Namias_Gabriel/tree/main/Projet_JoTicket			

¹ En référence aux *conditions de réalisation et ressources nécessaires* du bloc « Conception et développement d'applications » prévues dans le référentiel de certification du BTS SIO.

² Les réalisations professionnelles sont élaborées dans un environnement technologique conforme à l'annexe II.E du référentiel du BTS SIO.

³ Conformément au référentiel du BTS SIO « *Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en œuvre de ces outils et ressources. La circulaire nationale d'organisation précise les conditions matérielles de déroulement des interrogations et les pénalités à appliquer aux candidats qui ne se seraient pas munis des éléments nécessaires au déroulement de l'épreuve.* ». Les éléments peuvent être un identifiant, un mot de passe, une adresse réticulaire (URL) d'un espace de stockage et de la présentation de l'organisation du stockage.

⁴ Lien vers la documentation complète, précisant et décrivant, si cela n'a été fait au verso de la fiche, la réalisation professionnelle, par exemples service fourni par la réalisation, interfaces utilisateurs, description des classes ou de la base de données.

**Fiche descriptive de réalisation professionnelle
(verso, éventuellement pages suivantes)****Épreuve E6 - Conception et développement d'applications (option SLAM)****PROJET - JO Tickets****Compétences mobilisées**

- Utilisation d'un gestionnaire de versions pour son code
- Usage de l'HTML et du CSS pour créer des pages statiques
- Utilisation d'une base de données pour stocker les informations importantes
- Utilisation de Python avec le Framework Django pour interagir avec la BDD et proposer une interface d'administration
-

🔧 Contexte

Le Comité International Olympique fait de nouveau appel à vos talents ! En effet il a besoin d'un POC (Proof Of Concept) plus avancé pour sa compétition de football.

Il vous demande de créer une interface d'administration pour gérer les matchs de la compétition mais surtout une API utilisable depuis plusieurs applications.

La première application mobile permettra de voir la liste des matchs, se connecter, acheter des billets et générer des QR codes pour les billets achetés.

La seconde application à destination des stadiers permettra simplement de scanner le QR Code d'un billet afin d'autoriser l'entrée, et indiquer la place au supporter.

Pour cela le CIO vous fournit une ébauche de début de projet avec 3 dossiers principaux :

- /admin/ qui contient le projet Django (déjà créé)
- /mobile/ qui contiendra l'application pour les supporters
- /scanner/ qui contiendra l'application pour les stadiers

🔧 Objectifs

Créer 3 applications !

① Projet Django

Premièrement le projet Django, c'est la base. C'est lui qui accède à la base de données et fourni une API afin d'envoyer ces données aux autres applications. Il contient également une interface d'administration permettant de modifier les événements et ajouter les scores de chaque match.

Important, seul les supers utilisateurs peuvent se connecter à l'interface d'administration.

Pour la base de données, le CIO vous a fourni une base, plusieurs fichiers de Model et des données en SQL pour la garnir. Il vous faudra tout de même ajouter un model Ticket pour stocker les tickets achetés par les supporters.

② Application Mobile Supporter

L'application mobile supporter permettra de voir la liste des matchs dans leur ordre de diffusion, les supporters pourront acheter des billets pour chacun des matchs s'ils sont connectés. Ils devront donc pouvoir s'inscrire, et se connecter à l'application.

Important : L'inscription et la connexion se fait à travers l'API fourni par le projet Django. L'application mobile est un simple frontend, elle ne se connecte donc pas directement à une BDD. On enregistre donc les utilisateurs dans la BDD du projet Django.

Lors de l'achat du billet, le supporter peut choisir sa catégorie, le prix varie en fonction de la catégorie choisi. Les catégories sont les suivantes :

- Silver (100€)
- Gold (200€)
- Platinum (300€)

Le supporter peut acheter plusieurs tickets, de différentes catégories s'il le souhaite.

Une fois ses billets achetés, le supporter peut les retrouver dans son espace personnel. On regroupe les billets par match pour simplifier les choses au moment de les présenter au stade. On génère un QR Code par billet, celui-ci contient l'id du billet (un UUID), ainsi quand il sera scanné par l'application scanner, on aura immédiatement la place et le nom du supporter. Un bouton permettant de télécharger le QR Code peut être un plus.

③ Scanneur de billets

En plus de la petite application mobile, une autre page web indépendante aura pour unique but de scanner le QR Code inscrit sur les billets de la compétition. Après avoir scanné le QR Code contenant l'identifiant du billet, il faudra appeler l'**URL API** permettant de savoir si un billet est valide ou non, et en afficher les informations afin de comparer avec ce qui est imprimé sur le billet.

Pour la détection d'un QR Code à l'aide de la caméra d'un téléphone, on utilisera la bibliothèque Javascript [QR Scanner](#). Cette dernière nous renverra le contenu encodé dans le QR Code au moment où il réussira à être scanné. Pour simplifier les choses dans ce POC il est possible de simplement uploader une image du QR Code.

Schéma du projet :



