

Documentation technique

jonathan.sands
clement.davin
ugo.chane-you-kaye
romain.chuit

26 juin 2019

Table des matières

1	Introduction	3
2	Gestion du backend	3
2.1	Gestion de la base de données	3
2.2	Gestion des paiements	4
2.3	Views, urls et models	4
3	Gestion du frontend	4
4	Conclusion	5

Le but de ce document est d'expliquer comment nous avons réalisé notre billetterie YAKAPEX.

la billetterie YAKAPEX et une billetterie sur laquelle se trouvent les événements des différentes associations de l'école EPITA.

Au travers de ce document, vous comprendrez tout ce que nous avons mis en place pour la réalisation de notre projet.

1 Introduction

Nous avons pour projet de réaliser une billetterie pour l'école EPITA, afin qu'ils puissent avoir un support sur lequel communiquer leurs événements.

Globalement il était demandé d'avoir :

- Des événements ; certains premiums, d'autres non
- Gérer les rôles de certains utilisateurs (administrateur, directeur des associations, membre des associations...)
- Acheter de façon sécurisé des billets pour des événements
- Gérer les membres des associations
- Accéder à notre propre profil
- Accéder aux événements sur lesquels nous sommes inscrits (en tant que Staff ou pour participer à l'événement)

Nous allons expliquer par quel moyen nous avons mené à bien ces tâches.

2 Gestion du backend

Pour le backend nous avons uniquement utilisé Django.

2.1 Gestion de la base de données

Du côté de la base de données, Django permet de la gérer simplement grâce à ses models et ses migrations, nous avons au début créé tous les models nécessaire à notre site en utilisant des foreignkey pour les lier entre eux, une fois cela fait, il a été facile de gérer la base de données à plusieurs, quelques modifications ont été apportés au cours du développement du site dû à des oublis de noter part, mais cela n'a jamais eu d'impact sur le temps nécessaire au développement du site complet.

2.2 Gestion des paiements

Pour la gestion des paiements, Nous avons utilisé Paypal, nous avons créé un bouton personnalisé de Paypal qui nous permet de payer le prix qu'on veut, ensuite ce prix doit être payé par l'utilisateur que ce soit par carte bancaire ou compte Paypal, une fois payé, l'utilisateur reçoit sa confirmation de paiement par mail et le tout est joué.

2.3 Views, urls et models

Pour ceux qui est des views, urls et models, nous avons essayé de répartir le travail de façon à ne pas travailler dans les mêmes sous projets, nous n'avions alors qu'à include les urls d'un sous projets à un autre pour interagir. les urls nous permettaient, en fonction de l'url actuel du site, d'appeler les views, qui appellaient elles-même les templates. Les models nous ont permit de gérer la database comme expliqué au dessus.

3 Gestion du frontend

Pour le frontend, nous avons utilisé les langages de basiques en front (HTML/CSS) avec l'aide de bootstrap pour une mise en page plus propre et du javascript pour plus d'interaction avec l'utilisateur.

Pour les choix du style, nous avons donc pris le bleu/violet comme couleurs dominantes qui seront le thème du site. Nous avons essayé de mettre les informations au plus clair possible afin que l'utilisation et la compréhension soit facile à appréhender.

De plus, nous nous sommes inspirés de quelques sites connus afin d'avoir un exemple d'expert en la matière car nous n'avions pas vraiment de formation au design d'un site.

4 Conclusion

Nous avons réalisé notre billetterie en avec de divers langages :

- Django
- CSS
- HTML
- Javascript

Grâce à tout cela nous avons pu réaliser aujourd'hui une billetterie fonctionnelle, nous permettant de gérer des données, d'acheter / vendre des billets pour des événements, de gérer des personnes sur un événement (Staff, clients, flux...).