



Communauté Française de Belgique

Institut des Carrières Commerciales

Ville de Bruxelles

Rue de la Fontaine, 4

1000 Bruxelles

**Projet d'application de gestion de
réservations de spectacles en Python3
sous Django 3.0**

Rapport du projet d'intégration de développement

Section : Bachelier en Informatique de gestion

**MIRON Fernando, NDJONI KASONGO Cédric, NGOYI-A-YAMBU Caleb,
RAMSAMY Bryan, SOUPART Lionel, TWAHIRWA Dieudonné**

2019 – 2020

Remerciements

Pour les remerciements, moi, Cédric Ndjoni Kasongo remercie personnellement les membres de cette équipe qui ont rendu mon parcours plus agréable. Ils ont été fortement disponible, patient et à l'écoute en toute circonstance. Ils ont été justes et démocratiques dans la divergence et n'ont pas hésité à mettre la main à la patte.

Moi, Fernando Miron, je tiens à remercier toutes les personnes qui contribuent à mon succès personnel et professionnel.

Tout d'abord, je voudrais remercier mon épouse et mes filles, pour avoir beaucoup de patience; entreprendre des études supérieures lorsqu'on a une famille et un travail à temps plein très exigeant est extrêmement difficile. Sans leur support je n'y arriverais pas.

Ensuite je voudrais remercier tous les membres du groupe pour leur support et camaraderie. Même si nous nous connaissons à peine et en vue des circonstances actuelles, nous avons réussi, à distance, à créer un groupe assez uni et qui s'est entraidé à tout moment.

Enfin, une mention spéciale à Bryan Ramsamy, pour avoir accepté de prendre la responsabilité d'être le leader d'un groupe où les membres ont tous des niveaux de connaissance très différents du Framework Django, du logiciel de version git e du langage de programmation Python.

Moi, Bryan M. Ramsamy, je voudrais consacrer mes pensées au moment de la rédaction de ces lignes à ma grand-mère maternelle, que j'ai malheureusement perdue trop rapidement lorsqu'elle fût emportée par le Covid-19. Sa perte tragique, le fait que je n'ai rien pu faire et que je n'ai pas passé assez de temps avec elle par le passé m'a beaucoup touché et bien qu'il soit d'usage de remercier les personnes ayant contribués à la réalisation de ce projet, je voulais consacrer cette partie à sa mémoire.

Je voudrais également remercier mes meilleurs amis et mes frères qui me soutiennent en cette période très difficile à gérer à mon sens. J'aurais très certainement tout abandonné sans leur intervention et leur soutien. Et je les remercie également de subir mon sale caractère quotidiennement et de m'aimer pour celui-ci.

Je veux également consacrer ce passage aux membres mon groupe, en qui je n'avais pas misé gros en toute honnêteté. Je ne voyais pas d'un très bon œil le fait de devoir collaborer avec un groupe de personnes dont les connaissances en Python3 et en Django étaient proches de zéro, sachant qu'ils auraient pu m'entraîner vers le fond avec eux. Mais au lieu de cela, c'est l'inverse qui s'est produit. J'ai d'abord pu les hisser à mon niveau, puis ce sont eux qui m'ont porté encore plus haut. Bien que je regrette toujours mon passage à l'ICC, je ne regrette pas du tout d'avoir travaillé avec mes camarades et je serais d'accord de diriger un nouveau projet avec eux dans un contexte plus agréable.

Table des matières

1. Glossaire.....	5
2. Introduction	9
3. Cahier des charges	10
3.1. Le langage de programmation orientée objet : Python 3 sur Django 3.0	10
3.2. La gestion des erreurs de programmation – Techniques de débogage.....	11
3.3. Utilisation de la documentation	11
3.4. Optimisation du code, du cache et des échanges avec la base de données	12
3.4.1. Optimisation du code.....	12
3.4.2. Gestion du cache.....	12
3.4.3. Échange avec la base de données.....	13
3.5. Gestion d'un catalogue d'éléments	14
3.6. Déploiement d'un back-office d'administration.....	14
3.7. Sécurité de l'application.....	15
3.7.1. Gestion des injections SQL, du clickjacking, des failles XSS et de la vulnérabilité CSRF	15
3.7.2. Mots de passe	15
3.7.3. Accès au serveur et aux environnements de développements.....	16
3.8. Modules pull et push	16
3.8.1. API consommée (module pull).....	16
3.8.2. API produite (module push).....	17
3.8.3. Flux RSS (module push).....	17
3.8.4. Import et export en fichier CSV, XLS ou autre (module bidirectionnelle)	17
3.9. Procédure de déploiement à partir de l'archive.....	18
4. Méthodologie de développement	19
4.1. Organisation du groupe	19
4.1.1. Phase d'apprentissage	19
4.1.2. Phase d'analyse.....	19
4.1.3. Phase de développement	20
4.1.4. Phase de déploiement	21
4.2. Environnements.....	22
4.2.1. Environnements de développement	22
4.2.2. Environnement de production.....	23
4.3. Problèmes rencontrés.....	23

4.4. Solutions apportées	24
5. Analyse critique du travail réalisé	25
5.1. Points forts du projet	25
5.2. Points à améliorer	26
5.3. Développements futurs	27
5.4. Réflexion et apport individuel.....	28
5.4.1. Fernando Miron	28
5.4.2. Cédric Ndjoni Kasongo	28
5.4.3. Caleb Ngoyi-A-Yambu	29
5.4.4. Bryan Ramsamy.....	29
5.4.5. Lionel Soupart	29
5.4.6. Dieudonné Twahirwa	30
6. Conclusion	31
7. Bibliographie	32
8. Annexes.....	35
8.1. Liste exhaustive de tous les liens de l'application	35

1. Glossaire

Clickjacking (ou Détournement de clic)

Le clickjacking se produit lorsqu'un arnaqueur crée une interface web en trompe-l'œil et dérobe les clics effectués sur le site contrefait afin de les exploiter sur son site, bien réel celui-là. Les utilisateurs tombent sur le site illégitime dans lequel ils pensent remplir un champ, cliquer sur un lien ou taper leurs mots de passe pour accéder à ce qui leur est présenté.

Le clic a été détourné et utilisé pour confirmer une autre action sur un site différent ; de la même manière, ce qui a été tapé au clavier a été détourné pour récupérer le mot de passe et se connecter à des comptes à l'insu de la victime. Les applications de ce type d'activité frauduleuse sont illimitées.

La plupart des temps, ces clics ou ces frappes clavier dérobés sont acheminés vers des pages détenues par d'autres applications ou un autre domaine. Le stratagème est là devant vous et pourtant vous n'avez aucune idée qu'il est en train de s'appliquer.¹

Faible XSS ou Cross-Site Scripting

Le cross-site scripting (abrégé XSS) est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, provoquant ainsi des actions sur les navigateurs web visitant la page. Les possibilités des XSS sont très larges puisque l'attaquant peut utiliser tous les langages prises en charge par le navigateur (JavaScript, Java, Flash...) et de nouvelles possibilités sont régulièrement découvertes notamment avec l'arrivée de nouvelles technologies comme HTML5. Il est par exemple possible de rediriger vers un autre site pour de l'hameçonnage ou encore de voler la session en récupérant les cookies.²

Framework

Ensemble de composants logiciels qui permettent de créer le squelette d'un logiciel ou d'une application. Un framework est comparable à une boîte à outils dans laquelle le développeur vient chercher les composantes dont il a besoin. C'est en fait un cadre de travail qui simplifie le travail des développeurs en leur offrant une structure d'ensemble.

Les frameworks fonctionnent par langage de programmation et permettent de développer tous types de supports : sites web, jeux, applications mobiles etc.³

¹ NORTON TEAM. Mise à jour : le 26-09-2016. « Qu'est-ce que le clickjacking ? » sur *Norton Blog*. Site Web sur INTERNET. <https://fr.norton.com/norton-blog/2016/09/qu_est-ce_que_lecli.html>. Dernière consultation : le 03-06-2020.

² QUE20. Mise à jour : le 09-01-2016. « Les failles XSS » sur *Zeste de savoir*. Site Web sur INTERNET. <<https://zestedesavoir.com/articles/232/les-failles-xss/>>. Dernière consultation : le 03-06-2020.

³ C., Florian. Mise à jour : le 10-06-2019. « Qu'est-ce qu'un framework ? » sur *Wild Code School*. Site Web sur

Fonction de hashage

On nomme fonction de hachage, de l'anglais hash function (hash : pagaille, désordre, recouper et mélanger) par analogie avec la cuisine, une fonction particulière qui, à partir d'une donnée fournie en entrée, calcule une empreinte numérique servant à identifier rapidement la donnée initiale, au même titre qu'une signature pour identifier une personne. Les fonctions de hachage sont utilisées en informatique et en cryptographie notamment pour reconnaître rapidement des fichiers ou des mots de passe.⁴

Injection SQL

Une requête SQL est une demande d'action à effectuer dans une base de données, le plus souvent dans une page web qui demande un nom d'utilisateur ou un mot de passe. Mais, la plupart des sites web ne surveillant pas les entrées autres que les noms d'utilisateur et les mots de passe, un pirate informatique peut utiliser les zones d'entrée pour envoyer ses propres requêtes, c'est-à-dire injecter du SQL dans la base de données. De cette manière, les pirates informatiques peuvent créer, lire, mettre à jour, altérer ou supprimer les données stockées dans la base de données principale, généralement pour accéder à des informations sensibles comme les numéros de sécurité sociale et les données de cartes de crédit ainsi que d'autres informations financières.⁵

Look and feel (ou Aspect et Comportement)

Le *look and feel* est un ensemble de règles qui régissent la présentation visuelle ainsi que le comportement des interfaces graphiques. Les règles de présentation concernent en particulier l'usage des couleurs, la typographie, la présentation et la signification des logos et des icônes, la présentation des fenêtres — emplacement, forme et comportement des widgets — et les formes du curseur. Les règles de comportement régissent la manière dont les éléments visuels — notamment les widgets — répondent aux actions de l'utilisateur — mouvements de la souris, pression sur les boutons de la souris et du clavier. L'application de ces règles vise à faciliter l'apprentissage, améliorer la satisfaction utilisateur, apporter une identité visuelle aux produits et réduire leur coût de développement.⁶

INTERNET. <<https://www.wildcodeschool.com/fr-FR/blog/quest-ce-quun-framework>>. Dernière consultation : le 19-01-2020.

⁴ RYX. Mise à jour : le 25-10-2017. « Les fonctions de hachage cryptographiques » sur *Zeste de savoir*. Site Web sur INTERNET. <<https://zestedesavoir.com/tutoriels/1895/les-fonctions-de-hachage-cryptographiques/>>. Dernière consultation : le 04-06-2020.

⁵ AVAST ACADEMY TEAM. Mise à jour : le 13-05-2020. « Injection SQL » sur *Avast*. Site Web sur INTERNET. <<https://www.avast.com/fr-fr/c-sql-injection>>. Dernière consultation : le 03-06-2020.

⁶ JENSEN, Kara. Mise à jour : le 21-11-2013. « What is the “Look and Feel” of a Website? And Why It’s Important » sur *Bop Design*. Site Web sur INTERNET. <<https://www.bopdesign.com/bop-blog/2013/11/what-is-the-look-and-feel-of-a-website-and-why-its-important/>>. Dernière consultation : le 06-02-2020.

Mémoire cache

Une mémoire cache ou antémémoire est, en informatique, une mémoire qui enregistre temporairement des copies de données provenant d'une source, afin de diminuer le temps d'un accès ultérieur (en lecture) d'un matériel informatique (en général, un processeur) à ces données. Le principe du cache est également utilisable en écriture, et existe alors en trois modes possibles : write-through, write-back et write-around.⁷

Stack trace (ou Trace d'appels)

En informatique, une trace d'appels est la représentation des piles d'exécution à un moment donné lors de l'exécution d'un programme informatique.

Les développeurs utilisent couramment les traces d'appel lors de séances de débogage interactives ou post mortem. Les utilisateurs finaux peuvent parfois voir des traces d'appels affichées avec un message d'erreur, que l'utilisateur peut alors rapporter aux développeurs.

Une trace d'appels remonte la séquence d'appels de fonctions - jusqu'au point où la trace d'appels est générée. Dans un scénario post-mortem, cela remonte jusqu'à la fonction où l'erreur a eu lieu (mais qui ne l'a pas nécessairement provoquée). Les appels de fonctions « frères » (exécutés et terminés avant la fonction courante) n'apparaissent pas dans une trace d'appels.⁸

VPS ou Virtual Private Server (Serveur dédié virtuel)

Un serveur dédié est une méthode de partitionnement d'un serveur en plusieurs serveurs virtuels indépendants qui ont chacun les caractéristiques d'un serveur dédié, en utilisant des techniques de virtualisation. Chaque serveur peut fonctionner avec un système d'exploitation différent et redémarrer indépendamment. Dans le domaine de l'hébergement web, plusieurs dénominations recoupent le même type d'offres et donc de services. Les acronymes VPS (Virtual Private Server) et VDS (Virtual Dedicated Server) désignent le même concept, et leur usage est parfois confus.⁹

⁷ D, Roberto. Mise à jour : le 27-03-2019. « Qu'est-ce que la mémoire cache? » sur *astuce-pc.com*. Site Web sur INTERNET. <<https://astuce-pc.com/quest-ce-que-la-memoire-cache/>>. Dernière consultation : le 04-06-2020.

⁸ PARASCHIV, Eugen. Mise à jour : le 30-04-2018. « Understanding and Leveraging the Java Stack Trace » sur *Stackify*. Site Web sur INTERNET. <<https://stackify.com/java-stack-trace/>>. Dernière consultation : le 02-06-2020.

⁹ IONOS. Mise à jour : le 11-04-2017. *Qu'est-ce qu'un VPS (virtual private server) ?*. Site Web sur INTERNET. <<https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-quun-vps-virtual-private-server/>>. Dernière consultation : le 03-06-2020.

Vulnérabilité CSRF ou Cross-Site Request Forgery

En sécurité des systèmes d'information, le cross-site request forgery, abrégé CSRF (parfois prononcé sea-surf en anglais) ou XSRF, est un type de vulnérabilité des services d'authentification web.

L'objet de cette attaque est de transmettre à un utilisateur authentifié une requête HTTP falsifiée qui pointe sur une action interne au site, afin qu'il l'exécute sans en avoir conscience et en utilisant ses propres droits. L'utilisateur devient donc complice d'une attaque sans même s'en rendre compte. L'attaque étant actionnée par l'utilisateur, un grand nombre de systèmes d'authentification sont contournés.¹⁰

¹⁰ IONOS. Mise à jour : le 20-02-2020. *CSRF : le Cross Site Request Forgery*. Site Web sur INTERNET. <<https://www.ionos.fr/digitalguide/serveur/securite/cross-site-request-forgery/>>. Dernière consultation : le 03-06-2020.

2. Introduction

Au cours de la troisième et dernière année d'études en Informatique de Gestion à l'Institut des carrières commerciales (ICC), les étudiants doivent réaliser, dans le cadre du cours « Projet d'intégration de développement » un travail qui vise l'apprentissage de l'utilisation d'un Framework, la création d'une application web avec celui-ci et aussi le travail en équipe.

Dans ce contexte, notre groupe, intéressé par le Framework Django et les possibilités que celui-ci offre, a été formé. Le but de ce rapport est d'expliquer plusieurs aspects de l'organisation du groupe Django, dès sa création, jusqu'à la conclusion du projet.

Dans un premier temps, quelques aspects techniques du Framework Django et du langage de programmation Python seront expliqués. Ceux-ci visent à ce que le lecteur ait une meilleure idée du Framework, ce que celui-ci apporte et pourquoi nous l'avons choisi.

Ensuite sera décrite notre méthodologie de développement. Cette partie vise à démontrer quelles ont été les étapes que le groupe a dû franchir, quelles ont été les méthodologies de travail utilisées, nos méthodes d'organisation et les phases par lesquelles nous avons dû passer, de l'apprentissage de Django jusqu'au déploiement de l'application finie, dans un environnement de production.

Enfin, nous aborderons une analyse critique du travail effectué, les points forts du groupe, les points que le groupe considère comme pouvant être améliorés, une description de l'apport individuel de chacun de ces membres pour aboutir au travail fini ainsi qu'une conclusion.

L'application est disponible à l'adresse IP suivante : <http://80.211.219.225/show/>

Une liste exhaustive de tous les liens de l'application est disponible en fin de document (voir Chapitre 8 [Liste exhaustive de tous les liens de l'application](#)).

3. Cahier des charges

3.1. Le langage de programmation orientée objet : Python 3 sur Django 3.0

Le langage de programmation Python a l'avantage d'être à la fois facile à apprendre mais également riche en possibilités. L'avantage majeur du langage Python est que c'est un langage à interprétation dynamique. Les instructions sont immédiatement traduites en langages machines à mesure que celles-ci sont interprétées. Il n'y a donc aucune phase de compilation et les erreurs sont donc détectées dès leur exécution.¹¹

Dans le cadre de notre projet, ce sont les versions 3.6, 3.7 et 3.8 qui furent utilisées. La version 2 de Python étant désormais obsolète, l'utiliser engendrerait de nombreux problèmes de sécurité et de compatibilité.¹²

Les raisons qui nous ont poussées à opter pour ce langage de programmation plutôt qu'un autre sont nombreuses. D'abord, il est question du 3^e langage le plus populaire à l'heure actuelle. Mais contrairement au deux premiers langages (soient Java et C), Python est encore en pleine ascension là où les deux autres sont sur le déclin.¹³ Python est donc un meilleur choix pour le futur proche du développement.

Le choix proposé coté Framework n'était alors réduit qu'à Django.

Django reste cependant un bon Framework de haut niveau pour la création d'applications Web classiques. Celui-ci est très simple à utiliser et dispose d'un ORM performant, ce qui nous permet de laisser la gestion de la base de données et des instructions SQL correspondantes au Framework, limitant ainsi grandement le risque de violation d'intégrité de la base de données et d'injection SQL.¹⁴

Ajouter à cela, Django mets une très grande gamme de bibliothèques à disposition et est très modulable en ce qui concerne l'intégration de composants externes.

¹¹ SINICKI, Adam. Mise à jour : le 19-05-2020. « What is Python and how do you get started? » sur *Android Authority*. Site Web sur INTERNET. <<https://www.androidauthority.com/what-is-python-1120588/>>. Dernière consultation : le 02-06-2020.

¹² ABRAMS, Lawrence. Mise à jour : le 02-01-2020. « Python 2.7 Reaches End of Life After 20 Years of Development » sur *Bleeping Computers*. Site Web sur INTERNET. <<https://www.bleepingcomputer.com/news/software/python-27-reaches-end-of-life-after-20-years-of-development/>>. Dernière consultation : le 02-06-2020.

¹³ BITDEGREE. Mise à jour : le 24-02-2020. « Most Used Programming Languages 2020 » sur *BitDegree*. Site Web sur INTERNET. <<https://www.bitdegree.org/tutorials/most-used-programming-languages/>>. Dernière consultation : le 02-06-2020.

¹⁴ DJANGO PROJECT. 2020. « Django makes it easier to build better Web apps more quickly and with less code. » sur *Django Project*. Site Web sur INTERNET. <<https://www.djangoproject.com/>>. Dernière consultation : le 02-06-2020.

3.2. La gestion des erreurs de programmation – Techniques de débogage

Dans le cadre de notre projet de développement, nous avons intégré divers outils de débogage en plus des outils déjà disponibles par défaut dans Django 3.0.

Les bibliothèques *django-extensions* et *werkzeug*, incluant notamment les modules *iPython*, *runserver_plus*, et *debug_tt* permettent un rendu de débogage plus avancé avec un stack tracing plus explicite, ainsi qu'un terminal *iPython* dynamique permettant un accès au terminal Python 3.8 de Django avec un accès complet à toutes ses fonctions, mais aussi à la base de données.¹⁵ *iPython* permet également un gain de temps conséquent étant donné que toutes les bibliothèques, ainsi que les modèles locaux sont automatiquement importés par défaut. La bibliothèque *werkzeug* intègre *iPython* directement dans les pages d'erreur stack trace.¹⁶

Le serveur de développement avancé, utilisé dans un script de lancement bash, permet de lancer le serveur rapidement, de le relancer automatiquement à chaque changement effectué, mais aussi de relancer le serveur même en cas d'erreur critique. Dans de tels cas, le serveur ne permet pas un accès via le navigateur, mais affiche le stack trace et se relance jusqu'à ce que l'erreur soit corrigée. Le redémarrage manuel du serveur de développement n'est donc plus nécessaire.

3.3. Utilisation de la documentation

Contrairement à beaucoup de Frameworks modernes, Django a la particularité de suivre un processus de mise-à-jour organisé et complet. Cela a pour conséquence la mise-à-jour régulière et complète de sa documentation ainsi que de ses tutoriels, car la documentation officielle de Django inclut également des cours de prise en main du Framework.

Sa documentation est donc officielle et couvre absolument tous ses aspects.

L'utilisation de celle-ci a joué un rôle aussi majeur qu'incontournable. Celle-ci fût utilisée la majorité du temps afin de comprendre la sémantique de Django, la fonction de chaque méthode ainsi que la fonction des différentes bibliothèques intégrées par défaut.

Il en va de même lors de l'utilisation du Framework frontend Bootstrap 4 lors de la mise en place du look and feel de notre application de réservation.

Des recherches internet furent nécessaires pour l'utilisation de modules et de bibliothèques externes à Django. Dans ce cas, la documentation officielle correspondante fût utilisée si disponible.

¹⁵ DJANGO-EXTENSIONS. 2015. *Welcome to the django-extensions documentation!*. Site Web sur INTERNET. <<https://django-extensions.readthedocs.io/en/latest/>>. Dernière consultation : le 02-06-2020.

¹⁶ WERKZEUG. 2007. *Werkzeug*. Site Web sur INTERNET. <<https://werkzeug.palletsprojects.com/en/1.0.x/>>. Dernière consultation : le 02-06-2020.

3.4. Optimisation du code, du cache et des échanges avec la base de données

3.4.1. Optimisation du code

Le code fût écrit sur des IDE optimisés pour Django, tel que Virtual Studio Code combiné aux extensions Python et Django correspondantes, ou encore l'IDE PyCharm. L'utilisation de tels outils permet une optimisation immédiate dès que l'IDE détecte qu'une portion de code peut l'être. Le code fût également retravaillé par plusieurs membres afin de couvrir un maximum de cas de figure possible dans le délai qui nous était imposé.

3.4.2. Gestion du cache

Les portions statiques de l'application sont toutes cachées sur des délais courts. Étant donné que nous ne sommes encore qu'en environnement de développement et qu'une fois le passage en production, de légères mises-à-jour seront probablement nécessaires pour corriger quelques bugs mineurs, il est préférable de laisser le cache expirer rapidement pour éviter des résultats déplaisants.

La majorité de l'application étant dynamique et non statique, ces portions ne sont pas mises en cache.

En environnement de production, le cache est stocké en mémoire centrale du serveur en dessous d'un certain quota de données. Une fois ce quota dépassé, le cache est géré par le système de cache *memcached* installé directement sur le serveur de production.¹⁷

En environnement de développement, aucun cache n'est généré afin de prévenir des erreurs lors de la mise-à-jour de certains composants.

¹⁷ DORMANDO. 2018. « What is Memcached? » sur *Memcached*. Site Web sur INTERNET.
<<https://www.memcached.org/>>. Dernière consultation : le 06-02-2020.

3.4.3. Échange avec la base de données

Django implémente un ORM puissant, soit un Object-Relational Mapper ou un Traducteur d'Objet-Relationnel) permettant de traduire des entrées dans la base de données en objets Python 3 manipulables à souhait.¹⁸

Dans notre cas, la base de données MySQL, base de données imposée par le professeur bien qu'elle ne soit pas la mieux adaptée pour un projet Django 3.0, est connectée au Framework par le biais des modules Python 3 installée avec Django grâce à l'utilitaire d'installation pip3.¹⁹ Une fois installé et ajouté à la configuration du Framework, ces modules traduisent nos instructions Django en instructions SQL et inversement.

Aucune commande SQL n'a été entrée lors du développement du projet, rendant la base de données complètement imperméable à n'importe quelle interaction depuis l'application. Le risque d'injection SQL est ainsi complètement neutralisé sans un accès direct à l'environnement de développement ou de production, car seul des objets Python peuvent être manipulés directement.

La génération des tables et des entrées se fait via un système de migrations intégré dans le Framework, comme dans la plupart des ORM.

¹⁸ MCLAUGHLIN, Katie. Mise à jour : le 24-11-2017. « An introduction to the Django ORM » sur *OpenSource.com*. Site Web sur INTERNET. <<https://opensource.com/article/17/11/django-orm>>. Dernière consultation : le 06-02-2020.

¹⁹ DJANGO PROJECT. 2020. « Databases » sur *Django Project*. Site Web sur INTERNET. <<https://www.djangoproject.com/en/3.0/ref/databases/>>. Dernière consultation : le 02-06-2020.

3.5. Gestion d'un catalogue d'éléments

L'application se présente comme suit :

- Une page d'accueil mettant en avant 3 spectacles choisis au hasard se présentera avec le poster des spectacles en question, ainsi que leur description et un bouton permettant d'accéder à la page du spectacle (lien vers la page : <http://80.211.219.225/>).
- Une liste de tous les spectacles disponibles, trié selon leur ordre de sortie. Un champ de recherche est également disponible et permet de effectuer une recherche sur le titre des spectacles (lien vers la page : <http://80.211.219.225/show/>).
- Une pagination en bas de page afin de limiter le nombre de spectacles.
- Chaque spectacle est accompagné d'un bouton permettant d'obtenir plus d'informations sur celui-ci, ainsi que de consulter les représentations dudit spectacle (lien vers la page : <http://80.211.219.225/show/read/discours-a-la-nation/>).
- Les utilisateurs non-inscrits ne pourront pas effectuer de réservation, mais pourront consulter les spectacles, leurs représentations et les différents lieux où se dérouleront les représentations.
- Les modérateurs et les administrateurs pourront également ajouter de nouveaux spectacles, de nouvelles de représentations et de nouveaux lieux, ou modifier, voir supprimer, ceux déjà existant dans la base de données de l'application Web.
- L'application gère également l'intégration des services de partages de données brutes via des API Rest (lien vers la page : <http://80.211.219.225/api/list/>).
- L'application peut également mettre à jour les spectacles proposés depuis la lecture de données depuis une API externe de confiance. Les données peuvent être vérifiées avant d'être importées (lien vers la page : <http://80.211.219.225/api/show/pull/cleaned/>).

3.6. Déploiement d'un back-office d'administration

Une interface d'administration, uniquement accessible par les modérateurs et les administrateurs est également implémentée dans l'application. Celle-ci est implémentée par défaut par Django mais reste très basique.²⁰ Notre apport a permis de la rendre plus facile à prendre en main et de lui ajouter de nombreux modules utilitaires (lien vers la page : <http://80.211.219.225/admin/>).

Celle-ci peut gérer toutes les données stockées, à condition de disposer des autorisations nécessaires, tels que les spectacles, les représentations et les emplacements. Mais également les profils d'utilisateurs, la gestion de l'importation et de l'exportation des données sous format texte, CSV, XLS, ou autres, ainsi que l'activité des autres membres ayant accès à l'administration.

Des options d'affichage et de filtrage avancées sont également mises en place afin de rendre l'interface plus facile à prendre en main pour l'utilisateur en question.

²⁰ DJANGO PROJECT. 2020. « The Django admin site » sur *Django Project*. Site Web sur INTERNET.
<<https://www.djangoproject.com/en/3.0/ref/contrib/admin/>>. Dernière consultation : le 02-06-2020.

3.7. Sécurité de l'application

3.7.1. Gestion des injections SQL, du clickjacking, des failles XSS et de la vulnérabilité CSRF

Comme indiqué dans le chapitre consacré aux échanges entre l'application et sa base de données (voir Chapitre 3.4.3. [Échange avec la base de données](#)), l'application est à l'abri des injections SQL.

Le Framework intègre également l'intergiciel X-Frame-Option par défaut, ce qui prévient toute tentative de clickjacking.

Les formulaires Django sont également tous nettoyés après réception et validation des données. Ce nettoyage s'effectue partiellement par le Framework. Les données sont ensuite contrôlées une dernière fois avant d'être sauveées dans l'objet correspondant. Ce traitement effectué sur tous les champs rend les attaques XSS impossible.

Django 3.0 gère également les vulnérabilités CSRF par le billet d'un tag appelant les fonctions de vérifications correspondantes aux données du jeton CSRF de l'utilisateur. La seule contrainte pour cela était d'inclure ce tag dans chacun de nos formulaires. Par défaut, Django interdit les formulaires sans vérification CSRF, mais celle-ci peut être passée manuellement si besoin est.²¹

3.7.2. Mots de passe

Les mots de passes utilisateurs doivent répondre à des critères stricts de sélection. Les mots de passe trop faibles ou ne comportant pas assez de caractères, caractères spéciaux inclus, ne sont pas autorisés. De plus, chaque mot de passe est crypté par l'algorithme de hashage classique de Django. Par défaut Django utilise les dernières méthodes de hashage incluses au Framework.²²

Les mots de passes ne sont donc pas stockés en clair dans la base de données et sont donc illisibles, même pour les administrateurs. Cependant, il est toujours possible de changer ou de réinitialiser son mot de passe en cas de besoin. Cette dernière procédure passe par un mail externe envoyé avec un jeton de confirmation par Django, via un service SMTP, vers l'utilisateur en question. Celui-ci doit simplement entrer l'adresse mail utilisée lors de son inscription sur l'application et cliquer sur le lien qui lui sera envoyé que l'adresse en question.

²¹ DJANGO PROJECT. 2020. « Security in Django » sur *Django Project*. Site Web sur INTERNET.
<<https://docs.djangoproject.com/en/3.0/topics/security/>>. Dernière consultation : le 02-06-2020.

²² DJANGO PROJECT. 2020. « Password management in Django » sur *Django Project*. Site Web sur INTERNET.
<<https://docs.djangoproject.com/en/3.0/topics/auth/passwords/>>. Dernière consultation :
le 02-06-2020.

3.7.3. Accès au serveur et aux environnements de développements

L'accès au serveur se fait uniquement via une connexion SSH à un compte utilisateur configuré pour l'occasion. Tout autre mode de connexion est interdit. De plus, tous les ports à l'exception du port 22 pour les connexions SSH, et du port 80 pour les requêtes HTTP, sont fermés. Le seul moyen d'accéder au serveur autrement qu'en SSH ou en HTTP serait de se rendre physiquement sur place, dans un lieu qui nous est également inconnu.

Les informations sensibles ne sont pas codées en dur dans l'application. Celles-ci sont importées depuis un fichier d'environnement directement importé sur le serveur de production ou dans nos environnements de développement. L'application ne peut pas fonctionner sans. Il est donc impossible de lancer l'application dans un environnement non désiré à moins de radicalement changer toute la configuration de celle-ci.

3.8. Modules pull et push

3.8.1. API consommée (module pull)

L'application a été configurée pour récupérer des données automatiquement depuis une API externe. Dans le cas présent, il s'agit de l'API du Théâtre de la ville de Paris (lien vers l'API externe : <https://api.theatredelaville-paris.com/events>).

Une fois les données récupérées en format JSON, celles-ci sont transformées afin de pouvoir correspondre aux modèles de données utilisées par l'application développée par nos soins, après avoir effectué de multiples vérifications comme la gestion des doublons et des images.

Pour terminer, un modérateur ou un administrateur de l'application est en mesure de consulter la liste des spectacles obtenus depuis l'API externe directement sur notre API (lien vers la page : <http://80.211.219.225/show/pull/read/>). Depuis cette vue, celui-ci sera en mesure de mettre à jour les données de la base de données de l'application en y ajoutant les nouvelles données obtenues.

Afin de ne pas rendre notre application dépendante de l'API consommée, la mise à jour ne peut se faire qu'après confirmation d'un modérateur ou d'un administrateur. Celui-ci peut alors cliquer sur le bouton **Mise à jour des spectacles** afin d'importer les données dans notre base de données. L'application se charge de rajouter les nouveaux spectacles et de mettre à jour les spectacles importés déjà existants.

3.8.2. API produite (module push)

L'application fournit de multiples données en format JSON via l'API que nous avons développée grâce à la librairie *Django-Rest-Framework*.

Celle-ci reprend les données principales de l'application, à savoir : la liste des spectacles, la liste des emplacements et la liste des représentations (lien vers les API : <http://80.211.219.225/api/list/>). Elle reprend également les données brutes de l'API consommée ainsi que les données après traitement (voir Chapitre 3.8.1. [API consommée \(module pull\)](#))

Il est possible de filtrer directement depuis l'url sur une entrée spécifique, afin d'uniquement récolter les données relatives à un seul objet, comme un spectacle bien spécifique par exemple (lien de l'exemple : <http://80.211.219.225/api/show/?id=6>).

3.8.3. Flux RSS (module push)

Le flux RSS permet de récupérer les informations de notre site au format XML via les modules de lectures de flux disponible sur la plupart des navigateurs internet. Les informations relatives aux représentations, localisations et aux nouveaux spectacles y sont partagées (lien vers la page : <http://80.211.219.225/rss/list/>).

Concernant les spectacles, seuls les 3 derniers y sont affichés afin de tenir les internautes, inscrits au flux, informées des dernières sorties en évitant de les noyer au milieu des autres spectacles déjà existants.

3.8.4. Import et export en fichier CSV, XLS ou autre (module bidirectionnelle)

Le module *django-import-export* a permis l'intégration d'une fonctionnalité permettant d'importer et d'exporter des données dans un fichier. Les formats supportés sont : *csv*, *xls*, *xlsx*, *tsv*, *ods*, *json*, *yaml* et *html*. Lors d'une importation, le module se charge de vérifier la validité des données avant d'assigner à un objet de notre application.

Cette fonctionnalité peut cependant exporter tout type de données, y compris des données sensibles. Nous l'avons donc intégrée dans le panneau d'administration afin que celle-ci ne soit uniquement disponible par les administrateurs.

De plus, la pertinence de cette fonctionnalité vis-à-vis de l'utilisateur classique n'est pas dès lors que celui-ci a accès aux API (voir Chapitre 3.8.2. [API produite \(module push\)](#)).

3.9. Procédure de déploiement à partir de l'archive

Afin de respecter les consignes du projet, l'application fût développée dans un environnement natif utilisant un environnement virtuel Python3 lié à une base de données MySQL.

La procédure de déploiement sur un serveur linux est la suivante :

- Installation des modules principaux : *Python3*, *virtualenv*, *MySQL*, *Gunicorn* et *NGINX*.
- Création d'une base de données *MySQL* en y ajoutant les accès nécessaires pour l'application.
- Importation de l'intégralité du code source depuis un dépôt distant.
- Création de l'environnement virtuel *Python3*
- Installation des librairies *Python3* dont *Django 3.0*.
- Exécution des migrations de l'application afin de générer les tables dans la base de données
- Création d'un compte administrateur initial.
- Exécution des scripts de populations afin de peupler la base de données avec des données.
 - *Remarque : Dans un cas réel de déploiement, cette étape récupérerait les données depuis un back-up afin de récupérer le dernier état de la base de données.*
Dans le cadre du cours, aucun serveur de back-up n'a été mis à disposition, les données sont donc enregistrées en dur dans le script de population.
- Lancement de *Gunicorn* depuis le fichier service-serveur de l'application.
- Lancement du module *NGINX* depuis un fichier configuré manuellement selon les besoins de redirections ou autres.

En cas de modification au sein du projet, voici les étapes à suivre :

- Importations des modifications depuis un dépôt distant.
- Mise à jour des librairies *Python3* en cas de besoin.
- Exécution des migrations de l'application afin de mettre à jour les tables dans la base de données. Cette étape ne doit être exécutée uniquement si nécessaire, soit après modification des modèles de données Django.
- Relance des services *Gunicorn* afin d'appliquer les changements aux services-serveur.

La majorité de ces étapes sont exécutées automatiquement sur le serveur de production grâce à l'exécution de différents scripts. Certaines étapes, par mesure de sécurité, requièrent cependant une intervention manuelle.

4. Méthodologie de développement

4.1. Organisation du groupe

L'organisation de travail dans le cadre du projet a été décomposée en 4 phases décrites dans les paragraphes suivants. Toutes furent organisées et supervisées par un membre du groupe sous l'approbation des autres.

4.1.1. Phase d'apprentissage

L'organisation du cours et la répartition des groupes étant assez chaotique, nous avons commencé le projet à 3 pour ensuite agrandir notre groupe à 7 membres au bout de quelques semaines pour finalement nous retrouver à 6.

Le principal handicap de notre groupe, si ce n'est de tous les groupes, était la différence de connaissances et de compétences importantes entre nos différents membres. Nous ne débutions pas tous au même niveau et cette absence d'expérience et de leadership aurait pu nous être fatale. Les tutoriels et les roadmaps fournis dans le cadre du cours étaient incomplets, voir incorrects. Nous pouvons citer la prétendue roadmap Django qui était écrite à 70% en PHP, un langage qui n'a rien à voir, de près ou de loin, avec Python ou Django.

Nous avons donc organisé une phase d'apprentissage où les plus expérimentés d'entre nous prirent le temps de faire la leçon à ceux qui débutaient. L'idée était que les meilleurs encouragent les débutants à progresser. Chaque aspect du langage Python et du Framework Django fût expliqué et appliqué lors de nos séances d'apprentissage en commençant par les fonctionnalités les plus basiques en allant vers les plus avancées. L'idée d'origine était que chaque membre du groupe développe le projet individuellement afin de couvrir l'intégralité des fonctions de l'application. Il était hors de question de laisser l'un d'entre nous seul, à défendre et modifier un code qu'il n'a pas écrit ou qu'il n'a pas compris.

Cette phase a grandement contribué à accroître les connaissances et l'autonomie de chacun de nos membres. Elle a également permis à la majorité d'entre nous de réduire l'écart de connaissance avec ceux ayant déjà de l'expérience en la matière.

4.1.2. Phase d'analyse

Les consignes manquant parfois de précision et l'analyse n'étant pas complète voir incohérente par moments nous a poussé à revoir entièrement notre conception du projet.

Nous avons effectué des schémas et des diagrammes permettant de mieux visualiser la structure ainsi que le résultat final attendu. Nous avons également clarifié de nombreux points nébuleux issus des consignes. Mais les initiatives prises à certains moments ne furent pas toujours accueillies de manière unanime par les membres du groupe. Nous avons donc mis en place un système de vote où le souhait de la majorité l'emportait en laissant à chaque fois les différentes parties exposer leur point de vue.

Cette phase de clarification nous a permis de gagner du temps lors de la phase de développement. Le but était qu'arrivé à la phase suivante, il n'y avait plus le moindre doute sur la marche à suivre et le résultat à obtenir. Il ne fallait plus revenir en arrière et perdre du temps à clarifier ce qui ne l'était pas au départ.

4.1.3. Phase de développement

Une fois les bases de Django 3.0 acquises par l'ensemble du groupe et une fois tous les points de l'analyse clarifiés, nous avons pu nous lancer entièrement dans la phase de développement.

La première étape était la répartition des différentes tâches entre chacun de nous. Les membres étant le plus à l'aise dans un domaine étaient assignés aux tâches dudit domaine dans la mesure du possible. Il fallait également que chaque membre travaille sur toutes les fonctionnalités de l'application afin d'acquérir des connaissances dans tous les aspects du Framework. L'idée n'était donc pas de laisser quelqu'un se spécialiser dans une seule portion de Django, tout en ignorant tout des autres portions.

Chaque étape fut assignée à un responsable de tâche, soit une personne responsable de l'exécution de la tâche, de sa fonctionnalité et de ses tests éventuels. À cette personne étaient également assignés des supports, soit d'autres membres du groupe soutenant la personne responsable. Cette dernière pouvait ainsi déléguer ses tâches afin que nous puissions gagner en efficacité.

La tâche, son suivi, ce en quoi elle consiste, son responsable, ses supports, sa date limite et toutes autres informations relatives à celle-ci sont enregistrées dans un tableau d'organisation de type kanban disponible à tous.

Des réunions en téléconférence furent également organisées afin que chacun puisse faire état de son avancement dans le projet, à une fréquence d'une à trois par semaine. C'était également l'occasion de remettre en question le déroulement du projet afin de voir si on ne pouvait pas l'optimiser ou l'améliorer. Finalement, nous nous entraider et avançons efficacement les uns des autres créant un cercle vertueux où les efforts fournis par les uns poussaient les autres à en fournir davantage. Nous, qui étions de parfaits inconnus les uns des autres à l'origine, avons fini par créer des liens et surtout une solidarité résistant aux épreuves.

Le tableau suivant reprend un bref récapitulatif des tâches principales effectuées lors de la phase de développement, avec le nom de la personne responsable et de ses supports.

Tâches	Responsable	Support
Coordination du projet	Bryan	Dieudonné
Création du projet	Bryan	
Réalisation des modèles	Bryan	
Populer la base de données	Dieudonné	
Création des groupes utilisateurs	Dieudonné	Bryan
Rooters	Bryan	Lionel
Vues Statiques	Cédric	Caleb, Bryan
Vue localisation	Caleb	Cédric, Bryan
Vue profil	Dieudonné	Fernando
Vue détaillée show + représentation	Cédric	Caleb, Bryan
Vue Réservation	Cédric	Lionel, Caleb
Vue administration	Bryan	Caleb
Authentification	Fernando	Dieudonné
Permission	Dieudonné	Bryan
Cache	Bryan	
CRUD show, representation, reservation	Caleb	Cédric, Bryan
Paielement	Lionel	Bryan
API interne	Lionel	Dieudonné, Bryan
API externe	Dieudonné	Caleb, Bryan
flux RSS	Lionel	Dieudonné, Bryan
Import/export CSV	Dieudonné	Lionel, Bryan
Frontend avec Bootstrap	Cédric	Tout le monde
Déploiement	Bryan	Dieudonné, Fernando
Rapport	Lionel, Bryan	Fernando

4.1.4. Phase de déploiement

La dernière phase du projet était condensée en deux séances collectives où chaque membre aborda le travail qu'il avait effectué. Le membre en question devait enseigner aux autres ce qu'il avait réalisé, les moyens qu'il avait utilisés et comment les autres peuvent reproduire son travail.

Le but était que le membre comprenne ce qu'il a réalisé, qu'il puisse l'expliquer, que les autres puissent également comprendre les aspects sur lesquelles ils n'avaient pas ou peu travaillé et surtout qu'ils soient en mesure de modifier le code qu'ils n'avaient pas rédigé et testé eux-mêmes.

Réduire les différences de connaissances et de compétences était toujours à l'ordre du jour. En l'absence de véritable cours fourni par l'ICC, l'aspect pédagogique de ce projet était probablement celui sur lequel nous avons le plus mis l'accent.

Cette phase était précédée du déploiement de l'application sur un serveur Linux 18.04 LTS afin de rendre l'application accessible depuis internet.

4.2. Environnements

4.2.1. Environnements de développement

Chaque membre de notre groupe a développé une partie du projet sur un environnement qui lui est propre. Pour certains, l'application fût développée dans un environnement Ubuntu 18.04 ou 20.04 natifs. Pour d'autres, c'est un environnement virtuel de ces mêmes versions qui fût utilisé. L'un des nôtres a également opté pour un environnement Windows 10 Pro, au prix de certaines adaptations locales.

Pour tout projet Python3 natif, il est fortement recommandé, voir obligatoire dans certains cas, d'opter pour un environnement virtuel au niveau de l'application. Ceci afin d'installer les différents modules au sein même de l'environnement virtuel destiné à l'application sans parasiter le reste de l'environnement avec des installations et des configurations inutiles, pouvant même nuire à d'autres applications. L'utilitaire Python3 *virtualenv* fût choisi car celui-ci est aussi simple que performant.²³

Le serveur de développement intégré à Django 3.0 était ensuite utilisé pour tester la partie fonctionnelle de l'application. Les techniques avancées de développements sont toutes détaillées dans le chapitre correspondant (voir chapitre 3.2. [La gestion des erreurs de programmation – Techniques de débogage](#)).

L'utilitaire *git* a également été utilisé pour la gestion du code, de ses différentes versions et de son partage.²⁴ Chaque membre eut pour instruction claire de sauvegarder les différentes évolutions de leur code afin de pouvoir garder un historique précis ce celui-ci, d'être en mesure de voir les différents progrès de chacun et de pouvoir assembler les différentes portions de codes en résolvant le minimum de conflits.

L'utilitaire fût ensuite associé à la plateforme en ligne *GitHub*, imposée par le professeur, fût utilisée pour le stockage et le partage en ligne du code.²⁵ Celle-ci, bien qu'elle manque certaines fonctionnalités utiles, permis de nous offrir une vue globale de l'avancée du projet.²⁶ Nous avons également défini et suivi l'avancée de nos différentes tâches grâce au kanban disponible sur la plateforme GitHub. Celui-ci faisait office de point de repère principal dans le suivi de la progression du projet.

²³ VIRTUALENV. 2020. *Virtualenv*. Site Web sur INTERNET. <<https://virtualenv.pypa.io/en/stable/>> Dernière consultation : le 03-06-2020.

²⁴ SRIDHAR, Aditya. Mise à jour : le 12-08-2018. « An introduction to Git: what it is, and how to use it » sur *freeCodeCamp*. Site Web sur INTERNET. <<https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/>>. Dernière consultation : le 03-06-2020.

²⁵ BROWN, Korbin. Mise à jour : le 13-11-2019. « What Is GitHub, and What Is It Used For? » sur *How-to Geek*. Site Web sur INTERNET. <<https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>>. Dernière consultation : le 03-06-2020.

²⁶ MYSORE, Aniruddha. Mise à jour : le 13-08-2019. « 7 Best GitHub Alternatives You Should Use » sur *Beebom*. Site Web sur INTERNET. <<https://beebom.com/github-alternatives/>>. Dernière consultation : le 03-06-2020.

4.2.2. Environnement de production

Le serveur de production est en réalité un VPS tournant sur un des serveurs mis à disposition par Aruba, notre fournisseur d'hébergement web.²⁷

Le serveur tourne sous la version serveur d'Ubuntu 18.04 sur lequel fut installé Django 3.0, mais aussi les services Unicorn et Nginx offrant des services plus performants et mieux adaptés à un contexte de production. La couche Unicorn vient envelopper la couche Django pour ensuite être elle-même enveloppée par Nginx. Ces deux services serveurs sont les mieux adaptés pour faire tourner notre projet en production de manière simple et efficace.²⁸

La base de données MySQL fût également installée localement sur la machine. Ainsi, application et base de données fonctionnent de concert sur la même machine. Ce qui accroît les performances au détriment de la redondance de données et de la sécurité.

4.3. Problèmes rencontrés

Le premier problème majeur fut la différence de niveau entre les différents membres du groupe, comme évoquée dans plusieurs paragraphes de ce document. L'Institut des Carrières Commerciales n'a pas prévu le moindre cours nous permettant d'apprendre à utiliser un Framework en particulier. Le cours de Projet d'intégration de développement n'y fait pas exception étant donné que le cours se résumait à des quarts d'heures de questions-réponses avec le professeur. La plupart des membres n'étaient donc pas armés pour faire face au défi qui nous attendait.

Un autre problème de poids vu les conditions exceptionnelles dans lesquelles nous devons évoluer, à savoir les conditions de confinement strict imposées par le gouvernement à la suite de la crise du Covid-19 de 2020, était notre incapacité à nous retrouver et à faire le point ensemble.²⁹ Nous n'avions aucun contact les uns envers les autres, ce qui était d'autant plus problématique dans la mesure où nous étions encore de parfaits inconnus aux yeux de nos collègues.

Le troisième problème qui se profilait était que nous sommes tous salariés et que nous avons une vie personnelle en dehors du cadre de l'ICC. La difficulté résidait donc à pouvoir générer du temps à consacrer au projet alors que nos emplois du temps étaient déjà très remplis.

Nos consignes trop vagues, incohérentes voir contradictoires par moment jouaient également en notre défaveur. Dans l'incapacité d'interpeller directement le professeur en dehors des heures de cours et nos mails ne recevant aucune réponse, nous ne savions pas vraiment vers où nous allions. Il

²⁷ ARUBA CLOUD. 2020. *Cloud VPS*. Site Web sur INTERNET.

<<https://www.arubacloud.com/vps/virtual-private-server-range.aspx>>. Dernière consultation : le 03-06-2020.

²⁸ ELLINGWOOD, Justin. Mise à jour : le 09-07-2018. « How To Set Up Django with Postgres, Nginx, and Unicorn on Ubuntu 18.04 » sur *The DigitalOcean Community*. Site Web sur INTERNET.

<<https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-unicorn-on-ubuntu-18-04>>. Dernière consultation : le 03-06-2020.

²⁹ RAFFIN, Sophie. Mise à jour : le 13-03-2020. « Coronavirus "plus grande crise sanitaire depuis 1 siècle" : quelles étaient les autres ? » sur *Medisite*. Site Web sur INTERNET.

<<https://www.medisite.fr/pneumonie-bronchite-coronavirus-plus-grande-crise-sanitaire-depuis-1-siecle-quelles-etaient-les-autres.5560374.196469.html>>. Dernière consultation : le 03-06-2020.

était difficile de produire un résultat concret en l'absence de directive concrète et d'une analyse incorrecte et incomplète. Il nous fallu donc réadapter notre stratégie.

Finalement, le dernier problème qui se profilait à l'horizon à mesure que nous progressions dans notre travail était la défense orale individuelle du projet. Dans un contexte de confinement, n'ayant pas tous des connections internet très stable et n'étant pas tous des experts en Django 3.0, il était assez difficile de répondre aux exigences trop strictes dans un laps de temps trop court sans avoir pu acquérir suffisamment d'expérience, surtout en l'absence total d'enseignement de la part de l'ICC.

4.4. Solutions apportées

Pour pallier le manque d'expérience de certains par rapport à d'autres, des séances furent organisées où un des membres ayant le plus d'expérience, prit le temps d'enseigner aux autres les bases en leur expliquant la théorie tout en leur faisant des démonstrations au travers d'exemples. La personne enseignante partagea son écran et entra des bouts de code, des commandes ou autres afin de montrer comment le Framework était constitué, comment se servir de ses différentes fonctions et comment celles-ci fonctionnent. Ces cours remplaçaient donc l'absence de cours de l'ICC et étaient adapté au niveau de ses participants afin que personne ne soit largué.

Pour répondre au besoin de nous réunir et de faire le point régulièrement, des réunions en téléconférence furent régulièrement organisée à des jours et des heures spécifiques afin d'assurer que celles-ci ne bouleversent pas trop le planning de chacun. Les membres du groupe se réunissaient entre une et trois fois par semaine, selon les besoins. Des séances entre membres isolées pouvaient également prendre forme lorsqu'un membre demanda l'assistance d'un autre ou lorsque plusieurs membres devaient travailler à l'accomplissement de la même tâche. De manière générale, ces réunions nous ont permises de nous coordonner régulièrement et d'avancer tous ensemble afin d'éviter de laisser quelqu'un prendre trop de retard ou trop d'avance.

Grâce à la mise en place d'une organisation concrète, mais également flexible, chaque membre a pu adapter son emploi du temps afin de répondre aux demandes émanant du projet. Lorsqu'un membre était absent pour une raison quelconque, le déroulement du projet se déroulait temporairement sans lui et un autre membre se chargeait de tenir informer le membre manquant en question.

Notre stratégie face aux consignes trop vagues et incohérentes était très simple : reprendre l'analyse du projet, de ses besoins et des moyens à mettre en œuvre depuis le début. Nous avons donc refait toute la structure nous-même, en nous adaptant parfois par rapport aux consignes, pour clarifier nos attentes, les résultats que nous attendions et ce dans le but d'avoir une ligne de conduite bien plus claire et organisée (voir Chapitre 4.1.2. [Phase d'analyse](#)).

Pour finir, afin de préparer tout le monde à son examen individuel, mais aussi à notre examen en groupe, des séances récapitulatives furent organisées afin de tenir tout le monde informé du travail effectué sur les autres portions du projet (voir Chapitre 4.1.4. [Phase de déploiement](#)).

5. Analyse critique du travail réalisé

5.1. Points forts du projet

La plus grande force de notre projet a été notre travail d'équipe. Bien que la plupart d'entre nous ne se connaissaient pas ou très peu avant le début de ce projet, nous avons été capables d'apprendre à collaborer afin de délivrer dans un délai très serré une application aboutie et répondant à cahier des charges bien défini.

Ce travail d'équipe n'aurait pas été possible sans un leadership clair et une répartition des rôles bien définie. Cela a permis à chacun d'avoir une place dans ce projet et d'avoir conscience de ce qui était attendu de lui. Dès lors, chacun a pu avoir une contribution significative à ce projet et sentir son travail valorisé.

Tous les membres ont donc non seulement un sentiment de devoir accompli mais en plus la satisfaction d'avoir participé à un produit fini qui est à la hauteur de nos exigences.

Une autre force de notre projet a été l'entraide et la capacité des membres à améliorer leurs compétences en un temps très court. Certains membres n'avaient jamais programmé en python avant ce projet et d'autres n'avaient que des notions de base. Nous avons appliqué une méthode et mis en place un fonctionnement qui a permis aux membres connaissant le moins ce programme de combler leurs lacunes en un temps record, notamment grâce à l'aide des membres ayant une expertise dans ce langage.

Finalement, bien que le projet ait été divisé en différentes parties, le projet final est très bien intégré et cohérent. Il se présente comme un tout sans qu'il ne soit possible de constater que certaines parties aient été réalisées indépendamment les unes des autres.

5.2. Points à améliorer

Le premier point à améliorer n'est pas tant au niveau du projet que du cours. En effet, le projet doit être réalisé dans un délai très court à une période de l'année qui est déjà très chargée. Etaler ce cours sur toute l'année permettrait de mieux répartir la charge de travail pour les étudiants, d'autant plus que le premier quadrimestre est nettement plus léger que le second, mais également d'aboutir à un projet mieux finalisé car il laisserait plus de temps aux étudiants pour développer leurs compétences dans ce domaine.

En lien avec ce premier point, nous aurions souhaité disposer de plus de temps afin de pouvoir donner une place plus importante à la partie Javascript de notre application. Ce langage a été intégré mais seulement de façon très succincte dans notre application pour deux raisons. La première est que le Framework Django nous permet déjà de gérer une grande partie du frontend et la seconde est que nous n'avons pas disposé de suffisamment de temps pour implémenter ce langage de façon plus élaborée.

Nous avons utilisé les classes Bootstrap standard pour le design de notre application ce qui lui donne un style sobre mais efficace. Avec plus de temps, nous aurions pu concevoir une identité propre pour notre site au travers d'une charte graphique personnalisée, ce qui aurait amélioré la finition.

Finalement, la structure globale du code est fonctionnelle et couvre de nombreux cas de figure. Mais sa rédaction et son optimisation ne sont pas toujours optimales. Le standard de rédaction PEP8 imposé en langage Python n'a malheureusement pas pu être appliqué partout, étant donné que le temps nous a manqué et que la priorité était l'aspect fonctionnel du code.³⁰

³⁰ FINER, Jasmine. 2020. « How to Write Beautiful Python Code With PEP 8 » sur *Real Python*. Site Web sur INTERNET. <<https://realpython.com/python-pep8/>>. Dernière consultation : le 03-06-2020.

5.3. Développements futurs

L'usage du dépôt GitHub et l'utilisation d'une base de données de type MySQL limite le potentiel de l'application à différents aspects. Le fait que l'application utilise un environnement natif l'est également.

Le déploiement de l'application dans un environnement natif lui fait perdre en portabilité. Certaines parties de l'application devront être manuellement redéfinies afin de s'ajuster à un nouvel environnement non-Linux. Ce fût le cas pour un membre de notre groupe qui dû changer certains scripts afin que ceux-ci puissent être exécuté depuis son environnement Windows.

Conteneuriser l'application peut lui faire grandement gagner en portabilité car celle-ci peut alors être lancée plus rapidement, dans n'importe quel environnement de développement et de production, le tout de manière entièrement automatisée, avec une gestion de risques et de ressources beaucoup plus importante.

L'écriture dans une base de données MySQL n'est pas la méthode la mieux adaptée étant donné que *Django* a spécialement été conçu pour fonctionner de concert avec des bases de données *PostgreSQL*. Bien que les répercussions à l'échelle de notre projet soient faibles, dans un véritable cas de production, cela pourrait sérieusement nuire à la gestion des ressources lorsque la base de données atteindra une taille importante et que de nombreuses requêtes y seront exécutées en même temps.

Pour terminer, le dépôt GitHub ne permet pas l'intégration de mécanismes d'automatisations, tels que les *Pipelines* comme d'autres dépôts. En cas de modification du code, il faudra à chaque fois effectuer les changements manuellement même si beaucoup d'étape ont été automatisée par nos soins (voir Chapitre 3.9. [3.9. Procédure de déploiement à partir de l'archive](#)). L'utilisation d'un autre dépôt distant tel qu'*Atlassian BitBucket*, mais surtout *GitLab* permet de mettre de tels systèmes en place très rapidement, avec un gain non négligeable de temps lors de la phase de déploiement, mais surtout de maintenant.

5.4. Réflexion et apport individuel

Dans les prochains paragraphes se trouvent la réflexion de chaque membre sur son travail ainsi que sa vision de son apport au sein du projet.

5.4.1. Fernando Miron

Fernando a rejoint le groupe de Django en ayant beaucoup entendu parler de ce Framework. Il a voulu profiter cette occasion pour apprendre à comment s'en servir, pour pouvoir en faire usage lors de la réalisation de son épreuve intégrée à l'ICC.

En ce qui concerne son apport personnel, même s'il a développé une partie assez importante de l'application, à savoir la partie concernant l'authentification de ses membres, il a plus appris du groupe qu'il a apporté à celui-ci. Selon Fernando, certains membres avaient des connaissances assez avancées sur le sujet. Ses connaissances en Django 3.0 étaient donc nulles au départ, mais toujours selon lui, avoir participé à ce travail et surtout après avoir fait partie de ce groupe, qui était très organisé et motivé, de bonnes bases lui ont été acquises.

5.4.2. Cédric Ndjoni Kasongo

Cédric pense avoir apporté un certain savoir-faire dans l'intégration du design, la mise en scène et la logique de représentations des données. Ayant déjà pu faire un travail similaire en milieu professionnel, cette nouvelle expérience lui a permis d'apprendre un nouveau Framework en la présence de Django 3.0.

Le côté backend de l'application fût mieux appréhendé par celui-ci. Sortir de sa zone de confort, dans une atmosphère de travail aussi rigoureuse que vertueuse, était très intéressant. En particulier lorsque le but était l'apprentissage et la réussite d'un projet professionnel concernant une application concrète, ayant du sens et ayant bien été planifiée par le groupe. Le travail en groupe et le fait de se coordonner en télétravail dans un contexte particulier de pandémie et de confinement, avec des personnes que Cédric n'avait pour certaine jamais rencontré lui rendait l'expérience d'autant plus insolite. Cette aventure a permis pour lui d'aborder les différents aspects du code dans un contexte sans pression financière comme celle qu'il a vécu en entreprise.

L'élément le plus important retenu par Cédric restera la disponibilité des membres de son équipe. Plus particulièrement de ceux qu'il a estimé les plus expérimenté et qui ont donné leur temps pour enseigner aux autres membres.

5.4.3. Caleb Ngoyi-A-Yambu

Caleb énormément appris de ce projet. Ses connaissances sur tout ce qui concernait le mode de fonctionnement d'un Framework, notamment Django, était assez limité, ce qui n'est plus le cas aujourd'hui. Et comme l'ont souligné certains membres de l'équipe, les bases solides une fois acquises vont grandement faciliter la réalisation de l'épreuve intégrée.

Grâce à de longues heures de documentation, ce membre de l'équipe a pu s'acquitter au mieux des tâches qui lui étaient assignées.

Enfin, apprendre au sein d'une équipe sérieuse et travailleuse fût une expérience pour laquelle, Caleb pense garder un bon souvenir.

5.4.4. Bryan Ramsamy

Bryan fût le membre qui dirigea le groupe, s'occupa de la base du projet afin que tout le monde puisse évoluer depuis le même point et c'est également lui qui coordonna toutes les tâches après avoir instauré une organisation claire et simple, tendant trop dans le perfectionnisme par moment.

Développer un projet Django 3.0 n'était pas chose nouvelle pour lui, mais le faire au sein d'un environnement natif l'était. Entre autres, Bryan a pu transmettre son savoir avec une grande facilité et était agréablement surpris de voir l'enthousiasme et la détermination qu'affichaient ses collaborateurs.

Bryan a également estimé que bien qu'il a une forte conscience de sa valeur et de ses capacités de leader, il lui arrive bien trop souvent d'être trop rude, trop direct et de ne pas assez mâcher ses mots avant de les prononcer, ce qui aurait pu jouer contre lui et la dynamique du groupe si les membres ne lui avaient pas autant accordé leur confiance et leur appui.

5.4.5. Lionel Soupart

Au début de ce projet, Lionel ne disposait que de très peu de connaissance en Python et aucune de Django. Cela a donc été un réel défi pour lui que de se former afin d'avoir une contribution significative au groupe. Mais celui-ci fût en mesure de développer ses compétences suffisamment rapidement pour avoir pu assumer les tâches qui lui ont été confiées telles que la gestion des paiements.

En dehors de l'aspect programmation, sa constance additionnée à sa vision critique et constructive a beaucoup apporté au fonctionnement du groupe, surtout lorsque différents choix furent opérés.

5.4.6. Dieudonné Twahirwa

Le choix de Dieudonné fut plus facile. Il a rejoint le groupe Django étant donné que les bases du Framework lui étaient déjà acquises. Son objectif lors de la réalisation de ce projet était d'approfondir ses connaissances à un niveau plus avancé.

Lorsque les autres membres apprirent les bases de l'outil, Dieudonné y a vu une chance de revoir les bases lui aussi et de progresser avec ses partenaires. Il a non seulement pu partager ses connaissances, mais il a également beaucoup appris d'eux. Il faisait, entre autres, partie de ceux qui guidaient les membres les moins expérimentés, parfois en difficultés, vers le succès.

6. Conclusion

Au terme du projet de développement que nous avons réalisé, le sentiment général au sein de notre groupe est que « nous l'avons fait, nous avons atteint nos objectifs ».

En effet, même si beaucoup d'entre nous doutaient au début du projet que nous soyons capable d'effectuer l'ensemble du cahier des charges ou craignaient la charge de travail que cela allait représenter, la cohésion qui s'est mise en place au sein de notre groupe et l'effort continu de chacun des membres a permis de mener à bien ce projet ambitieux dans un laps de temps très réduit.

En nous basant sur le langage Python et le Framework Django, nous avons pu développer une application complète de gestion des réservations pour une salle de spectacle.

Même si elle n'est pas parfaite, nous avons tenté de développer une application répondant aux standards actuels en matière de programmation. Que ce soit en termes d'optimisation du code, de gestion du cache ou encore de sécurité, nous avons mis un point d'honneur à réaliser une application fonctionnelle que nous serions fier de pouvoir présenter à de futurs employeurs comme étant l'une de nos créations et un gage de la qualité de notre travail.

Malgré les difficultés rencontrées, telles que la charge de travail lors de ce quadrimestre, la crise du COVID-19 qui rendu les échanges plus difficiles ou encore nos vies personnelles et professionnelles qui s'ajoutaient à tout cela, nous avons réussi à donner le meilleur de nous-mêmes, tant individuellement que collectivement, pour les surmonter.

Tout cela a été possible grâce à une organisation à la fois souple dans un premier temps, au vu des niveaux de maîtrise très différents des membres du groupe, qui a mué pour devenir soutenue et structurée sur la fin afin de pouvoir tenir les délais très serrés.

Tout n'a pas été facile, tout n'a pas été parfait, tous nous n'avions pas la même vision des choses mais ensemble, nous l'avons fait.

7. Bibliographie

- ABRAMS, Lawrence. Mise à jour : le 02-01-2020. « Python 2.7 Reaches End of Life After 20 Years of Development » sur *Bleeping Computers*. Site Web sur INTERNET. <<https://www.bleepingcomputer.com/news/software/python-27-reaches-end-of-life-after-20-years-of-development/>>. Dernière consultation : le 02-06-2020.
- ARUBA CLOUD. 2020. *Cloud VPS*. Site Web sur INTERNET. <<https://www.arubacloud.com/vps/virtual-private-server-range.aspx>>. Dernière consultation : le 03-06-2020.
- AVAST ACADEMY TEAM. Mise à jour : le 13-05-2020. « Injection SQL » sur *Avast*. Site Web sur INTERNET. <<https://www.avast.com/fr-fr/c-sql-injection>>. Dernière consultation : le 03-06-2020.
- BITDEGREE. Mise à jour : le 24-02-2020. « Most Used Programming Languages 2020 » sur *BitDegree*. Site Web sur INTERNET. <<https://www.bitdegree.org/tutorials/most-used-programming-languages/>>. Dernière consultation : le 02-06-2020.
- BROWN, Korbin. Mise à jour : le 13-11-2019. « What Is GitHub, and What Is It Used For? » sur *How-to Geek*. Site Web sur INTERNET. <<https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/>>. Dernière consultation : le 03-06-2020.
- C., Florian. Mise à jour: le 10-06-2019. « Qu'est-ce qu'un framework ? » sur *Wild Code School*. Site Web sur INTERNET. <<https://www.wildcodeschool.com/fr-FR/blog/quest-ce-quun-framework>>. Dernière consultation : le 19-01-2020.
- D, Roberto. Mise à jour : le 27-03-2019. « Qu'est-ce que la mémoire cache? » sur *astuce-pc.com*. Site Web sur INTERNET. <<https://astuce-pc.com/quest-ce-que-la-memoire-cache/>>. Dernière consultation : le 04-06-2020.
- DORMANDO. 2018. « What is Memcached? » sur *Memcached*. Site Web sur INTERNET. <<https://www.memcached.org/>>. Dernière consultation : le 06-02-2020.
- DJANGO-EXTENSIONS. 2015. *Welcome to the django-extensions documentation!*. Site Web sur INTERNET. <<https://django-extensions.readthedocs.io/en/latest/>>. Dernière consultation : le 02-06-2020.
- DJANGO PROJECT. 2020. « Databases » sur *Django Project*. Site Web sur INTERNET. <<https://www.djangoproject.com/en/3.0/ref/databases/>>. Dernière consultation : le 02-06-2020.
- _____. 2020. « Django makes it easier to build better Web apps more quickly and with less code. » sur *Django Project*. Site Web sur INTERNET. <<https://www.djangoproject.com/>>. Dernière consultation : le 02-06-2020.
- _____. 2020. « Password management in Django » sur *Django Project*. Site Web sur INTERNET. <<https://docs.djangoproject.com/en/3.0/topics/auth/passwords/>>. Dernière consultation : le 02-06-2020.
- _____. 2020. « Security in Django » sur *Django Project*. Site Web sur INTERNET. <<https://docs.djangoproject.com/en/3.0/topics/security/>>. Dernière consultation : le 02-06-2020.
- _____. 2020. « The Django admin site » sur *Django Project*. Site Web sur INTERNET. <<https://www.djangoproject.com/en/3.0/ref/contrib/admin/>>. Dernière consultation : le 02-06-2020.

- ELLINGWOOD, Justin. Mise à jour : le 09-07-2018. « How To Set Up Django with Postgres, Nginx, and Unicorn on Ubuntu 18.04 » sur *The DigitalOcean Community*. Site Web sur INTERNET. <<https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-unicorn-on-ubuntu-18-04>>. Dernière consultation : le 03-06-2020.
- FINER, Jasmine. 2020. « How to Write Beautiful Python Code With PEP 8 » sur *Real Python*. Site Web sur INTERNET. <<https://realpython.com/python-pep8/>>. Dernière consultation : le 03-06-2020.
- JENSEN, Kara. Mise à jour : le 21-11-2013. « What is the "Look and Feel" of a Website? And Why It's Important » sur *Bop Design*. Site Web sur INTERNET. <<https://www.bopdesign.com/bop-blog/2013/11/what-is-the-look-and-feel-of-a-website-and-why-its-important/>>. Dernière consultation : le 06-02-2020.
- IONOS. Mise à jour : le 11-04-2017. *Qu'est-ce qu'un VPS (virtual private server) ?*. Site Web sur INTERNET. <<https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-quun-vps-virtual-private-server/>>. Dernière consultation : le 03-06-2020.
- _____. Mise à jour : le 20-02-2020. *CSRF : le Cross Site Request Forgery*. Site Web sur INTERNET. <<https://www.ionos.fr/digitalguide/serveur/securite/cross-site-request-forgery/>>. Dernière consultation : le 03-06-2020.
- MCLAUGHLIN, Katie. Mise à jour : le 24-11-2017. « An introduction to the Django ORM » sur *OpenSource.com*. Site Web sur INTERNET. <<https://opensource.com/article/17/11/django-orm>>. Dernière consultation : le 06-02-2020.
- MYSORE, Aniruddha. Mise à jour : le 13-08-2019. « 7 Best GitHub Alternatives You Should Use » sur *Beebom*. Site Web sur INTERNET. <<https://beebom.com/github-alternatives/>>. Dernière consultation : le 03-06-2020.
- NORTON TEAM. Mise à jour : le 26-09-2016. « Qu'est-ce que le clickjacking ? » sur *Norton Blog*. Site Web sur INTERNET. <<https://fr.norton.com/norton-blog/2016/09/qu-est-ce-que-lecli.html>>. Dernière consultation : le 03-06-2020.
- PARASCHIV, Eugen. Mise à jour : le 30-04-2018. « Understanding and Leveraging the Java Stack Trace » sur *Stackify*. Site Web sur INTERNET. <<https://stackify.com/java-stack-trace/>>. Dernière consultation : le 02-06-2020.
- QUE20. Mise à jour : le 09-01-2016. « Les failles XSS » sur *Zeste de savoir*. Site Web sur INTERNET. <<https://zestedesavoir.com/articles/232/les-failles-xss/>>. Dernière consultation : le 03-06-2020.
- RAFFIN, Sophie. Mise à jour : le 13-03-2020. « Coronavirus "plus grande crise sanitaire depuis 1 siècle" : quelles étaient les autres ? » sur *Medisite*. Site Web sur INTERNET. <<https://www.medisite.fr/pneumonie-bronchite-coronavirus-plus-grande-crise-sanitaire-depuis-1-siecle-quelles-etaient-les-autres.5560374.196469.html>>. Dernière consultation : le 03-06-2020.
- RYX. Mise à jour : le 25-10-2017. « Les fonctions de hachage cryptographiques » sur *Zeste de savoir*. Site Web sur INTERNET. <<https://zestedesavoir.com/tutoriels/1895/les-fonctions-de-hachage-cryptographiques/>>. Dernière consultation : le 04-06-2020.
- SINICKI, Adam. Mise à jour : le 19-05-2020. « What is Python and how do you get started? » sur *Android Authority*. Site Web sur INTERNET. <<https://www.androidauthority.com/what-is-python-1120588/>>. Dernière consultation : le 02-06-2020.
- SRIDHAR, Aditya. Mise à jour : le 12-08-2018. « An introduction to Git: what it is, and how to use it » sur *freeCodeCamp*. Site Web sur INTERNET. <<https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/>>. Dernière consultation : le 03-06-2020.

VIRTUALENV. 2020. *Virtualenv*. Site Web sur INTERNET. <<https://virtualenv.pypa.io/en/stable/>> Dernière consultation : le 03-06-2020.

WERKZEUG. 2007. *Werkzeug*. Site Web sur INTERNET. <<https://werkzeug.palletsprojects.com/en/1.0.x/>>. Dernière consultation : le 02-06-2020.

8. Annexes

8.1. Liste exhaustive de tous les liens de l'application

VUE	LIEN
ACCUEIL	http://80.211.219.225/
LISTE DES SPECTACLES	http://80.211.219.225/show/
DÉTAIL D'UN SPECTACLE	http://80.211.219.225/show/read/going-home/
RÉSERVATION	http://80.211.219.225/representation/read/53/
PAIEMENT	http://80.211.219.225/payhome/?idreservation=53
LISTE DES LIEUX	http://80.211.219.225/location/
MENU API	http://80.211.219.225/api/list/
API PULL THÉÂTRE DE PARIS	http://80.211.219.225/api/show/pull/cleaned/
API GET EXEMPLE SPECTACLES	http://80.211.219.225/api/show/
MENU FLUX RSS	http://80.211.219.225/rss/list/
FLUX RSS NOUVEAUX SPECTACLES	http://80.211.219.225/rss/show/
ADMINISTRATION	http://80.211.219.225/admin
ADMINISTRATION – GROUPES	http://80.211.219.225/adminauth/group/
ADMINISTRATION – UTILISATEURS	http://80.211.219.225/adminauth/user/
ADMINISTRATION – MODÈLE – ARTISTE	http://80.211.219.225/adminapp/artist/
ADMINISTRATION – MODÈLE – IMPORT	http://80.211.219.225/adminapp/artist/import/
ADMINISTRATION – MODÈLE – EXPORT	http://80.211.219.225/adminapp/artist/export/?
PROFIL	http://80.211.219.225/profile/#profil
CONNEXION	http://80.211.219.225/login/
DÉCONNEXION	http://80.211.219.225/logout/
MODIFICATION MOT DE PASSE	http://80.211.219.225/adminpassword_change/
RESET MOT DE PASSE	http://80.211.219.225/password/reset/