**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 avec Laravel – Documentation technique**



*Cahier de charges et rapport dans le cadre du cours « Projet d’intégration de développement » pour la section « Bachelier en informatique de gestion »*

**DE VRIENDT Nicolas – MARIGLIANO Ludovic – PRZEMYSLAW Halasik (Patrick)**

*Année académique 2021 – 2022*

**Table des matières**

[Remerciements 4](#_Toc105685049)

[Glossaire 5](#_Toc105685050)

[1. Introduction 7](#_Toc105685051)

[2. Cahier de charges 8](#_Toc105685052)

[2.1. Langage de développement 8](#_Toc105685053)

[2.2. Framework 8](#_Toc105685054)

[2.3. Logiciels de développement et déploiement 9](#_Toc105685055)

[2.4. Méthode de gestion de projet 11](#_Toc105685056)

[2.5. Services à implémenter 11](#_Toc105685057)

[2.5.1. Services utilisateurs 11](#_Toc105685058)

[2.5.1.1. Consultation des spectacles 11](#_Toc105685059)

[2.5.1.2. Enregistrement 12](#_Toc105685060)

[2.5.1.3. Login 12](#_Toc105685061)

[2.5.1.4. Réservation de spectacles 12](#_Toc105685062)

[2.5.1.5. Payement de spectacles 12](#_Toc105685063)

[2.5.2. Services administrateur 12](#_Toc105685064)

[2.6. Sécurité 13](#_Toc105685065)

[2.6.1. Authentification 13](#_Toc105685066)

[2.6.2. Injections SQL 13](#_Toc105685067)

[2.6.3. Failles XSS 13](#_Toc105685068)

[2.6.4. Protection CSRF 14](#_Toc105685069)

[2.7. Webservices 15](#_Toc105685070)

[2.8. API utilisées 15](#_Toc105685071)

[2.9. Développement futur 15](#_Toc105685072)

[3. Rapport 16](#_Toc105685073)

[3.1. Implémentation de base de données (AA2) 16](#_Toc105685074)

[3.1.1. Restaurer une base de données (AA2 C1 I1) 16](#_Toc105685075)

[3.1.2. Ajouter une table (AA2 C1 I2) 17](#_Toc105685076)

[3.1.3. Insérer ou modifier des données (AA2 C1 I3) 17](#_Toc105685077)

[3.1.4. Génération d’un DUMP (AA2 C1 I4) 17](#_Toc105685078)

[3.2. Déploiement et réponses au CDC (AA3) 17](#_Toc105685079)

[3.2.1. Réalisation du travail de groupe (AA3 C1) 17](#_Toc105685080)

[3.2.1.1. Installation et configuration du framework (AA3 C1 I1) 18](#_Toc105685081)

[3.2.1.2. Outil de gestion des dépendances (AA3 C1 I2) 18](#_Toc105685082)

[3.2.1.3. Utilisation des commandes console (AA3 C1 I3) 18](#_Toc105685083)

[3.2.1.4. Système de gestion de version en ligne (AA3 C1 I4) 19](#_Toc105685084)

[3.2.1.5. Utilisation d’un moteur de templates (AA3 C1 I6) 20](#_Toc105685085)

[3.2.1.6. Hiérarchisation des templates (AA3 C1 I7) 20](#_Toc105685086)

[3.2.1.7. Personnalisation des templates (AA3 C1 I8) 20](#_Toc105685087)

[3.2.1.8. Utilisation de formats templates (AA3 C1 I9) 20](#_Toc105685088)

[3.2.1.9. Utilisation d’un ORM (AA3 C1 I10) 20](#_Toc105685089)

[3.2.1.10. Définition du mapping relationnel (AA3 C1 I11) 20](#_Toc105685090)

[3.2.1.11. Définition de la validation des champs (AA3 C1 I12) 21](#_Toc105685091)

[3.2.1.12. Définition de méthodes d’accès aux données (AA3 C1 I13) 21](#_Toc105685092)

[3.2.1.13. Définition de routes personnalisées (AA3 C1 I14) 21](#_Toc105685093)

[3.2.1.14. Génération et personnalisation de formulaire (AA3 C1 I15) 21](#_Toc105685094)

[3.2.1.15. Utilisation de documentation (AA3 C1 I16) 21](#_Toc105685095)

[3.2.1.16. Définition des migrations (AA3 C1 I17) 21](#_Toc105685096)

[3.2.1.17. Définition d’un jeu de données test (AA3 C1 I18) 22](#_Toc105685097)

[3.2.1.18. Implémentation de tests (AA3 C1 I19) 22](#_Toc105685098)

[3.2.1.19. Sécurisation de l’accès aux pages (AA3 C1 I20) 22](#_Toc105685099)

[3.2.1.20. Incorporation et utilisation d’un progiciel (AA3 C1 I21) 22](#_Toc105685100)

[3.2.2. Déploiement sur serveur (AA3 C3) 23](#_Toc105685101)

[3.2.2.1. Déploiement du site sur serveur test (AA3 C3 I1) 23](#_Toc105685102)

[3.2.2.1.1. Création du serveur MySQL sur Azure 23](#_Toc105685103)

[3.2.2.1.2. Création des bases de données de test 24](#_Toc105685104)

[3.2.2.1.3. Modification du fichier d’environnement 24](#_Toc105685105)

[3.2.2.1.4. Création de l’application web de test 25](#_Toc105685106)

[3.2.2.1.5. Installation de l’extension « Azure App Service » sur Visual Studio Code et déploiement 25](#_Toc105685107)

[3.2.2.1.6. Création de variables d’environnement sur l’application web Azure 27](#_Toc105685108)

[3.2.2.1.7. Changement de la racine de l’application web 27](#_Toc105685109)

[3.2.2.1.8. Accès à l’application de test 29](#_Toc105685110)

[3.2.2.2. Déploiement du site sur serveur de production (AA3 C1 I2) 29](#_Toc105685111)

[3.2.2.2.1. Création des bases de données de production 29](#_Toc105685112)

[3.2.2.2.2. Création de l’application web de production 29](#_Toc105685113)

[3.2.2.2.3. Accès à l’application de production 30](#_Toc105685114)

[4. Analyse 31](#_Toc105685115)

[5. Conclusion 32](#_Toc105685116)

[Bibliographie, Webographie 33](#_Toc105685117)

[Image de la page de garde 33](#_Toc105685118)

[Bibliographie des autres pages 33](#_Toc105685119)

# Remerciements

# Glossaire

**Back-End :** le terme Back-End est employé pour désigner une partie invisible aux yeux d’un utilisateur et comprend les fonctionnalités d’une application.[[1]](#footnote-1)

**DUMP :** *on appelle dump, la génération des commandes SQL destinées à reconstituer tout ou partie d'une base de données (sauvegarde logique), tandis que la "sauvegarde" d'une base de données est constituée d'une copie binaire de l'intégralité des pages de données et des transactions (sauvegarde physique).[[2]](#footnote-2)*

**Fixture :** les fixtures pour Symphony ont la même fonction qu’un seeder pour Laravel. *Les fixtures vont permettre d'insérer de fausses données en base de données.*[[3]](#footnote-3)

**Framework :** *un framework (ou infrastructure logicielle en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application. […] L'objectif du framework est de simplifier et d'uniformiser le travail des développeurs.[[4]](#footnote-4)*

**Front-End :** *le Front-End est la partie que les utilisateurs peuvent voir et avec laquelle ils peuvent interagir.[[5]](#footnote-5)*

**Générateur :** un générateur de base de données permet de générer des données de test pour remplir celle-ci lorsque le site est en phase de développement notamment.[[6]](#footnote-6)

**Open-source :** *à l'origine, l'expression « Open Source » faisait référence aux logiciels Open Source. Un logiciel Open Source est un code conçu pour être accessible au public: n'importe qui peut voir, modifier et distribuer le code à sa convenance.*[[7]](#footnote-7)

**Seeder :** un jeu de données test, ou seeder, permet d’insérer des données dans une base de données à des fins de testing de l’application et de la base de données elle-même.[[8]](#footnote-8)

# Introduction

Il est demandé dans le cadre du cours de Projet d’Intégration de Développement de réaliser une application de réservation de spectacles pour des utilisateurs selon les spécifications du chargé de cours.

Pour cela il est demandé de choisir un langage de programmation (Java, PHP ou Python), d’utiliser un Framework et dépendances associées, de suivre un roadmap selon le langage pour répondre aux critères d’évaluation fixés par le chargé de cours.

Ce travail est réalisé en groupe ce qui permet de simuler un environnement de travail réel où il est nécessaire de collaborer avec chacun pour avancer sur les tâches assignées.

Enfin, dans le cadre du cours, il est demandé de produire un cahier de charge succinct, ainsi qu’un rapport documentant le développement réalisé et répondant de manière pertinente aux questions posées par la grille d’évaluation et le roadmap.

Le présent document commencera, comme expliqué ci-dessus, par reprendre les informations essentielles d’un cahier de charges.

Le chapitre consacré au cahier de charges traitera du choix du langage de développement, du Framework, des logiciels de développement et déploiement utilisés, la méthode de gestion du projet par le groupe, des services qui seront implémentés, de la sécurité de l’application, de l’API utilisée et enfin des choix de développement futurs.

Le chapitre suivant sera consacré au rapport expliquant et répondant aux divers critères de la grille d’évaluation du chargé de cours et sera composé de deux parties principales explicitant l’implémentation de la base de données et le développement et déploiement de l’application.

Un chapitre sera consacré à l’analyse du travail (À IMPLÉMENTER POUR REMPLIR L’INTRO AVEC CE QUE CONTIENT L’ANALYSE).

Enfin une conclusion sera apportée sur base des chapitres consacrés au cahier des charges, au développement et déploiement de l’application et à l’analyse et reverra leur contenu de manière succincte.

Le lecteur est maintenant invité à commencer la lecture du document, en lui souhaitant que celle-ci lui soit instructive.

# Cahier de charges

Le chapitre relatif au cahier de charges sera subdivisé en plusieurs sections pour décrire le langage de développement utilisé, le [**Framework**](#Framework) choisi par les membres du groupe, les autres logiciels utilisés pour le développement et déploiement, la méthode de gestion de projet suivie par le groupe, les services implémentés pour les utilisateurs et administrateurs, la sécurité de l’application, l’API utilisée et le développement futur de l’application. VOIR SI ON PARLE DES WEBSERVICES

## Langage de développement

Le langage de développement utilisé est le PHP, cependant les versions ont varié quelque peu suivant que les membres du groupe travaillent depuis un ordinateur tournant sous Linux, Windows ou encore sous MacOs.

De la version 7.3.29 à la 8.1.3, toutes les parties du projet codées sur l’un ou l’autre OS n’ont rencontré aucun souci de compatibilité et c’est un des avantages de ce langage. Il est cependant à noter que d’anciennes versions d’OS ne sont pas supportées par PHP comme Windows 2008 et Vista à partir de PHP 7.2.0.[[9]](#footnote-9)

Un autre avantage de PHP est qu’il relativement aisé à apprendre, gratuit et qu’il s’inclut parfaitement dans le langage HTML. De plus son affichage est véritablement aisé grâce à des outils comme MAMP, WAMP ou encore LAMP qui sont des softwares compatibles à l’OS utilisé, quel qu’il soit, et contenant un serveur Apache, MySql et PHP.

Il est donc très simple d’afficher le résultat du travail sans devoir effectuer un hébergement ou compiler le travail effectué.

PHP est conçu pour principalement programmer du contenu WEB et est donc fortement compatible avec les divers navigateurs, les API, le XML ou JSON.

## Framework

Laravel 8.83.5 est le Framework [**open source**](#OpenSource) pour lequel le groupe a opté et qui propose un modèle de conception « modèle-vue-contrôleur ».

Certains membres avaient déjà réalisé quelques projets à l’aide de ce Framework et ont fait part aux autres de la facilité de la prise en main même si l’architecture de l’organisation des dossiers et fichiers puisse paraître assez déroutante de prime abord. Chaque membre a su cependant rapidement prendre ses marques et utiliser Laravel.

La relation entre Laravel et la base de données du projet est idéale car la facilité de migration de données est reconnue par nombre d’utilisateurs de Laravel.

De plus ce Framework est prévu pour sécuriser au mieux chaque projet comme nous le verrons ultérieurement dans le [**chapitre consacré à la sécurité**](#_Sécurité).

Pour ne prendre ici qu’un exemple, les insertions SQL sont par exemple exclues.

Laravel est, comme déjà stipulé, open source et il existe une communauté active qui propose une assistance et une aide facile à trouver sur internet, et ce, pour les débutants comme pour les professionnels.

Il est possible d’opter pour des modèles prédéfinis pour l’affichage des données.

En effet, « Blade » propose des modèles clairs, légers sans empêcher les programmeurs d’écrire leur propre code s’ils le désirent.

Laravel est un Framework progressif incluant de nombreuses fonctionnalités et une bibliothèque de modules fournie, orientée objet, permettant d’utiliser des fonctionnalités déjà toutes prêtes telle qu’un module d’authentification.[[10]](#footnote-10)

Concernant les tests unitaires Laravel n’est pas en reste, il permet de réaliser facilement des tests afin que toutes les parties de votre projet soient vérifiées avant de passer en production comme nous le verrons dans le [**chapitre consacré aux tests**](#_Implémentation_de_tests).[[11]](#footnote-11)

## Logiciels de développement et déploiement

L’avantage de travailler en équipe est d’utiliser toutes sortes d’habitudes, de softwares, de navigateurs différents. En effet, l’équipe représente en quelque sorte une frange des utilisateurs finaux.

Si dans ce cas-ci c’est principalement Visual Studio Code qui a servi d’éditeur, le groupe a utilisé durant le développement plusieurs navigateurs passant de Firefox à Edge et de Chrome à Safari, il a été possible devérifier au fur et à mesure l’affichage sur les navigateurs les plus utilisés.

« Composer » est un outil développé par la communauté PHP qui, tout comme « Bundler », créé pour Ruby ou encore « NPM » concernant NodeJs, est un gestionnaire de dépendances entre applications.et librairies.[[12]](#footnote-12)

Le framework Laravel est simple d’utilisation et permet d’implémenter rapidement et facilement des pré-fonctions telles que l’identification par exemple en allant simplement se fournir dans une bibliothèque extrêmement bien garnie.[[13]](#footnote-13)

Il est principalement orienté backend mais grâce à Blade par exemple, il est possible d’établir une partie frontend aisément.[[14]](#footnote-14)

Laravel Blade Snippets va colorer le code afin de le rendre encore plus lisible principalement pour les vues Blades.[[15]](#footnote-15)

Laravel Ide Helper est un paquet « Composer ». Grâce à lui, l’auto complétion sera étendue et encore plus simple.[[16]](#footnote-16)

PHP Intelephense permet l’auto complétion, des liens de navigation dans le code, des imports auto, etc.[[17]](#footnote-17)

Concernant le déploiement nous avons opté pour un hébergement chez OVH. L’un de nous ayant un compte mutualisé, il a pu offrir cette possibilité à l’équipe.

A COMPLETER LORS DU DEPLOIEMENT et détailler la procédure au point 9 de ce travail

## Méthode de gestion de projet

La gestion de projet s’est rapidement déroulée autour de GitHub et de manière naturelle une partie de l’équipe s’est orientée vers la partie [**Backend**](#BackEnd) du projet pendant que l’autre s’est occupée du [**Frontend**](#FrontEnd).

Une branche Master a été configurée et chacun a ensuite créé sa branche personnelle afin que chaque membre de l’équipe puisse développer à son rythme et sans risque de compromettre le travail commun.

Chaque fois qu’une partie était opérationnelle un push était réalisé sur la Master.

Une fois la base de données implémentée le groupe a principalement travaillé sur le Frontend.

De plus, GitHub propose une interface « Projects » permettant de créer des « tickets » d’intervention.

L’interface a été divisée en deux segments distincts pour les tickets Frontend et Backend pour la première partie et une seconde partie pour les autres tickets.

Cette interface est claire et divisée en trois parties : « To do », « In progress » et « Done ». Chaque élément peut être assigné à un ou plusieurs membres. Le suivi est optimalisé et le travail parfaitement réparti.

Quelques réunions se sont aussi organisées physiquement afin de partager les éléments importants, les nouveautés, les trucs et astuces découverts durant nos recherches, ainsi que des contacts téléphoniques ou via WhatsApp de manière régulière.

## Services à implémenter

Cette partie reprendra les fonctionnalités installées tant pour les utilisateurs que pour les administrateurs du site.

### Services utilisateurs

Dans cette partie il sera question des fonctionnalités qui ont été mises en place pour l’utilisateur.

#### Consultation des spectacles

La consultation des spectacles peut se faire sur la page d’accueil ou un carrousel de spectacles est affiché avec des spectacles qui sont actuellement donnés.

L’utilisateur a la possibilité de cliquer dessus pour consulter les spectacles, les lieux, les dates ou encore les artistes présents dans le spectacle.

Pour pouvoir effectuer une réservation, l’utilisateur devra être enregistré, sans quoi aucune réservation n’est possible.

#### Enregistrement

L’enregistrement permet à un utilisateur λ de pouvoir compléter un formulaire d’inscription pour pouvoir devenir un utilisateur enregistré.

Cet utilisateur reçoit un e-mail de confirmation pour pouvoir valider son compte et y accéder.

#### Login

L’utilisateur ayant un compte pourra s’authentifier via la page de login où il devra insérer son e-mail avec laquelle il s’est inscrit, ainsi que son mot de passe.

L’utilisateur aura au préalable cliqué sur le lien présent dans son e-mail d’inscription.

L’utilisateur ayant tenté de s’authentifier 10 fois sans succès sur la journée sera bloqué pour une période de 24 heures sur base de son adresse IP.

#### Réservation de spectacles

L’utilisateur enregistré aura la possibilité de réserver pour le(s) spectacle(s) au(x)quel(s) il souhaite assister. Celui-ci pourra réserver une ou plusieurs places dans la limite des places disponibles.

#### Payement de spectacles

L’utilisateur enregistré aura la possibilité d’effectuer un payement de ses réservations via divers moyens de payements (À Définir).

### Services administrateur

Le panneau administrateur permettra de naviguer entre les différents artistes, emplacements de spectacles, localisations de salles, les shows et les utilisateurs.

L’administrateur sera en mesure, via ces menus, de créer, visualiser, mettre à jour ou supprimer des enregistrements relatifs à ces différents menus.

## Sécurité

La gestion de la sécurité de l’application a lieu à plusieurs niveaux, certains de ceux-ci étant implémentés automatiquement par Laravel.

Les points suivants traiteront de la gestion de sécurité pour l’authentification, les injections SQL, les failles XSS et la protection CSRF.

### Authentification

Laravel gère lui-même l’authentification, ainsi que plus particulièrement le hashage des mots de passe via la facade « Hash » qui permet d’effectuer celui-ci via Bcrypt.[[18]](#footnote-18)

Le hashage permettra de ne pas mettre les mots de passes en clair au sein de la base de données pour assurer la sécurité et la confidentialité de ceux-ci.

L’authentification administrateur dans Laravel peut être gérée via les Middelware et en particulier celui nommé « Admin » qui permet de récupérer les informations de l’utilisateur et tester sa fonction, s’il est ou non administrateur et en fonction de cela gérer les routes exclusives aux administrateurs du site.[[19]](#footnote-19)

### Injections SQL

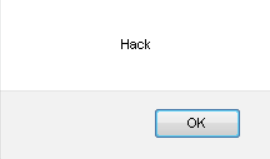
Eloquent et ici utilisé et directement implémenté par Laravel et permet de se prémunir contre toute injection SQL.

Eloquent est un ORM, c’est-à-dire un Object Relational Mapping, permettant de travailler avec les bases de données, chacune d’entre elles possédant son modèle au sein de Laravel qui permet de manipuler chacune d’entre elle.[[20]](#footnote-20)

### Failles XSS

Une faille XSS ou Cross Site Scripting est une faille de sécurité permettant d’injecter du code ou du contenu dans une page à partir d’un champ de formulaire permettant à un attaquant éventuel de rediriger un utilisateur vers une copie du site web, voler des informations relatives à des cookies ou une session, des actions au nom de l’utilisateur sur le site ayant une faille ou encore augmenter la difficulté de lecture d’une page, voire la rendre illisible.[[21]](#footnote-21)

Il est possible de vérifier que le site est ou non sensible à ce genre de failles grâce à un simple code à rentrer dans un champ formulaire (<script>alert(‘Hack’)</script> ) qui fera apparaître un message sur le site même comme suit :[[22]](#footnote-22)



Il est possible de s’en prémunir via divers moyens tels que les fonctions htmlspecialchars() permettant de convertir les caractères spéciaux, htmlentities() permettant la même chose que la précédente fonction mais filtre tous les caractères HTML et javascript ou encore strip\_tags() qui supprime toutes les balises.[[23]](#footnote-23)

Au sein de Laravel, Blade a été utilisé et permet de faire une syntaxe qui est sécurisée avec les sigles {{…}}.[[24]](#footnote-24)

### Protection CSRF

*CSRF (Cross-Site Request Forgery) est une attaque qui usurpe l'identité d'un utilisateur de confiance et envoie des commandes non désirées sur un site web. Cela peut être réalisé, par exemple, en ajoutant des paramètres malveillants dans une URL associée à un lien qui prétend aller quelque part ailleurs.*[[25]](#footnote-25)

La protection CSRF est automatiquement mise en place via Laravel.[[26]](#footnote-26)

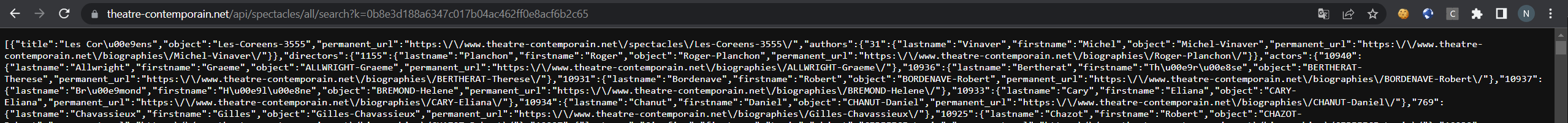
## Webservices

## API utilisées

L’API qui sera utilisée (consommée) permettra de recevoir toutes les informations par rapport à tous les spectacles du théâtre contemporain tant passés que futurs.

Le point d’entrée du site, son adresse générale, est <https://www.theatre-contemporain.net/> avec comme requête d’API « /api/spectacles/all/search » et la clé API étant la suivante : « ?k=0b8e3d188a6347c017b04ac462ff0e8acf6b2c65 » et permettant de chercher tous les spectacles (voir pertinence de la recherche pour n’avoir que les spectacles actuels-futurs).

Le rendu brut sur navigateur de la page se présente comme suit :



## Développement futur

# Rapport

Le présent chapitre consacré au rapport explicitera les méthodes utilisées pour pouvoir implémenter une base de données, la restaurer et apportera des réponses également par rapport aux demandes du cahier de charges et du déploiement du site de réservations.

## Implémentation de base de données (AA2)

Cette partie détaillera les méthodes utilisées pour ajouter une table, insérer ou modifier des données, générer un [**DUMP**](#DUMP) de la base de données et restaurer celle-ci.

### Restaurer une base de données (AA2 C1 I1)

Il est demandé que la base de données puisse être restaurée au moyen soit d’un DUMP SQL soit de [**générateurs**](#Generateur), de [**seeders**](#Seeder) ou de [**fixtures**](#Fixture).

Dans le cas d’un environnement de production, des classes de seeders (générateurs, fixtures) ont été définis pour implémenter des données test dans la base de données et sont donc automatiquement générés, permettant de restaurer à tout moment la base de données avec des données de test, rendant inutile un quelconque DUMP SQL.

Il fallait au préalable créer la table et ses champs au moyens de la commande « **php artisan make:model <NomDeLaTable>** » qui crée un modèle de table que l’on peut modifier avec les attributs des champs ($fillable si une modification est possible, $guarded s’il ne peut être modifié, etc.) et d’ensuite utiliser la commande « **php artisan make:migration <NomDeLaMigration>** » permettant la création du fichier contenant les informations de la table et qui sera par la suite modifié pour rencontrer les prérequis demandés avant de pouvoir intégrer la table dans la base de données au moyen de la commande « **php artisan migrate** ».

Les classes de seeders ont été générées via la commande « **php artisan make:seeder <NomDuSeeder>** » puis ces classes ont été modifiées pour répondre aux besoins d’insertion de chacune d’entre elles.

Après génération il était utile de rajouter au fichier « DatabaseSeeder.php » la classe de seeder à exécuter, régénérer le fichier autoloader de « Composer » via la commande « **composer dump-upload** » et ensuite la commande « **php artisan db:seed** » pour exécuter le seeder.

Dans le cas d’un déploiement en situation réelle, un DUMP de la base de données peut être nécessaire mais non indispensable si des seeders sont existants, c’est pourquoi les deux options ont été implémentées dans le cas présent.

### Ajouter une table (AA2 C1 I2)

Si une base de données peut être restaurée, c’est qu’elle a été au préalable créée. Pour un ajout de table, comme vu dans le [**chapitre relatif à la restauration de base de données**](#_Restaurer_une_base), la commande « **php artisan make:model <NomDeLaTable>** » crée le modèle de la table qui peut être modifié pour rencontrer les attributs souhaités comme par exemple $primaryKey si l’on souhaite que la clé primaire soit autre que le champs « id » qui est pris en considération par défaut par Laravel.

Ensuite un fichier de migration est créé avec la commande « **php artisan make:migration <NomDeLaMigration>** » et modifié si nécessaire pour ensuite migrer la table au moyen de la commande « **php artisan migrate** » et ensuite modifier les champs nécessaire et leur ajouter les propriétés $fillable pour les champs modifiables, $guarded dans le cas contraire, ainsi que d’ajouter la propriété public ou protected, ainsi que $primaryKey en cas de champs autre que le champ id qui est pris en considération par défaut par Laravel.

### Insérer ou modifier des données (AA2 C1 I3)

Pour insérer des données dans une table, dans le cas présents ce sont les seeders qui sont utilisés comme expliqué au point concernant la [**restauration d’une base de données**](#_Restaurer_une_base).

Pour modifier des données présentes dans une table, il est nécessaire de modifier les fichiers de seeders créés préalablement grâce à la commande « php artisan make :seeder <NomDuSeeder> » puis ajouter la classe au fichier « DatabaseSeeder.php » en cas d’une insertion nouvelle pour ensuite régénérer le fichier autoloader de « Composer » grâce à la commande « **composer dump-upload** » et ensuite ingérer les seeders dans la base de données avec la commande « **php artisan db:seed** ».

### Génération d’un DUMP (AA2 C1 I4)

## Déploiement et réponses au CDC (AA3)

### Réalisation du travail de groupe (AA3 C1)

#### Installation et configuration du framework (AA3 C1 I1)

Pour effectuer le travail, le groupe a travaillé avec l’outil Visual Studio Code qui permet d’éditer le code et fonctionne sur Windows, Linux et macOS et le développement de cet outil a été effectué par Microsoft.[[27]](#footnote-27)

Par la suite, l’installation de Laravel a dû se faire après l’installation du gestionnaire de dépendances « Composer ».

Celui-ci devait dans un premier temps être mis à jour au moyen de la commande « **composer self‑update** » et ensuite le Framework à proprement parler pouvait être installé au moyen de la commande « **composer create-project --prefer-dist laravel/laravel <NomDuProjet>** ».

Une fois installé, il est important de tester si l’installation a bien été effectuée en entrant dans le projet via le terminal et une fois entré dedans, lancer le serveur via la commande « **php artisan serve** ».

Diverses modifications pour configurer le serveur étaient nécessaire telles qu’ouvrir le fichier « .env » dans lequel la ligne APP\_NAME devait être modifiée par le nom du projet, vérifier que la ligne APP\_KEY possédait bien une entrée ou alors le cas échéant la générer via la commande « **php artisan key:generate** ».

Il fallait aussi vérifier et modifier si nécessaire le chemin d’accès à la base de données en modifiant les lignes concernant le DB\_DATABASE qui est le nom de la base de données, DB\_USERNAME pour modifier l’identifiant à la base de données et enfin DB\_PASSWORD au cas où un mot de passe était requis pour accéder à la base de données.

#### Outil de gestion des dépendances (AA3 C1 I2)

COMPOSER COMMENT INSTALLER ?

#### Utilisation des commandes console (AA3 C1 I3)

Tout au long du projet, il est demandé d’effectuer des actions au moyen de la console présente dans Visual Studio Code.

Comme nous pouvons déjà le voir dans les quelques chapitres précédents et le verrons encore dans les suivants, bon nombre de commandes ont été utilisées pour mener à bien la réalisation du projet « réservations », que ce soit au niveau du Framework en lui-même, du versioning sur GitHub, ou encore le gestionnaire de dépendances « Composer », c’est pourquoi il ne sera pas fait mention du détail dans cette partie de toutes les commandes utilisées.

#### Système de gestion de version en ligne (AA3 C1 I4)

Lors de ce projet il était demandé de faire appel à un système de versioning et celui qui a été choisi est GitHub.

GitHub est une plateforme de versioning créée en 2008 et hébergeant plus de 35 millions de projets pour 14 millions d’utilisateurs en 2016.[[28]](#footnote-28)

GitHub est composé du mot « git » *qui fait référence à un système de contrôle de version open-source* et hub pouvant désigner le réseau social qui a été construit autour du système Git ou encore à la plateforme de correspondance qui s’appelle un hub en anglais.[[29]](#footnote-29)

Cette plateforme permet donc beaucoup de fonctionnalités permettant de travailler à distance de concert avec d’autres personnes.

La méthode qui a été utilisée pour le groupe a été de conserver une branche Master qui comportait le projet et une branche pour chaque membre du groupe pour pouvoir travailler sur le code sans toucher à la Master tant que la branche personnelle n’était pas fonctionnelle. Une fois que celle-ci l’était, les modifications étaient implémentées sur la Master sur GitHub.

CONFIGURATION LARAVEL OK JUSTE AVEC LA LIGNE CI-DESSOUS ?

Dans un premier temps, il est nécessaire de relier Laravel au compte GitHub via la commande « **git config –global user.email <EMail>** ».[[30]](#footnote-30)

Pour pouvoir versionner sur GitHub, la commande « **git init** » permet de créer un nouveau dépôt Git dans le dossier du projet pour pouvoir utiliser ensuite la commande « **git add .** » permettant d’ajouter les modifications dans le dépôt pour pouvoir ensuite commenter via la commande « **git commit –m « <Commentaire>** » et finalement implémenter les modifications soit sur la branche personnelle soit sur la branche principale au moyen de la commande « **git push origin <NomDeLaBranche>** ».[[31]](#footnote-31)

Une fois cela effectué, le versionning est complet soit sur la branche personnelle ou la Master, en sachant que si un push est effectué sur la Master, il existe une protection permettant à un tiers travaillant dans le projet de valider le push avant que celui-ci ne soit définitif.

#### Utilisation d’un moteur de templates (AA3 C1 I6)

QUID VUE.JS ?

Le moteur de templates qui est utilisé pour la réalisation du travail de groupe est Blade que Laravel fournit comme fonctionnalité en termes de vue et permet d’utiliser du PHP sous une autre forme en ajoutant l’extension « .blade.php ».

DÉVELOPPER POUR ÊTRE PLUS COMPLET ?

#### Hiérarchisation des templates (AA3 C1 I7)

PLUS D’INFOS

#### Personnalisation des templates (AA3 C1 I8)

PLUS D’INFOS

#### Utilisation de formats templates (AA3 C1 I9)

PLUS D’INFOS

#### Utilisation d’un ORM (AA3 C1 I10)

Comme vu précédemment dans le [**chapitre consacré aux injections SQL**](#_Injections_SQL), l’Object Relational Mapping (ORM) qui est utilisé dans le cadre de ce projet est nommé Eloquent et permet notamment de travailler sur les bases de données au moyen de modèles (Models) prédéfinis et personnalisables au sein de Laravel.[[32]](#footnote-32)

#### Définition du mapping relationnel (AA3 C1 I11)

Un ORM ou Object Relational Mapping, ici Eloquent, se place entre le programme Laravel et la base de données relationnelle permettant par ce biais de simuler une base de données orientée objet.

AJOUTER EXPLICATION TECHNIQUE DÉFINISSANT LE MAPPING RELATIONEL POUR LE PROJET

#### Définition de la validation des champs (AA3 C1 I12)

AJOUTER EXPLICATION TECHNIQUE DÉFINISSANT LA VALIDATION DES CHAMPS POUR LE PROJET

#### Définition de méthodes d’accès aux données (AA3 C1 I13)

IDEM CHAPITRE 3 DU ROADMAP ??

#### Définition de routes personnalisées (AA3 C1 I14)

EXPLICATION TECHNIQUE

#### Génération et personnalisation de formulaire (AA3 C1 I15)

EXPLICATION TECHNIQUE

#### Utilisation de documentation (AA3 C1 I16)

Sur ce projet la documentation a été utile à de nombreux égards, car bien que chacun des membres du groupe vienne avec son propre bagage, il était essentiel de se documenter pour apprendre certaines choses que certains ou tout le monde ne connaissaient pas.

Comme il peut être constaté à la lecture du présent rapport et des sources utilisées, le groupe s’est largement documenté sur le Framework et les outils annexes utilisés tels que l’ORM, le gestionnaire de dépendances, etc.

#### Définition des migrations (AA3 C1 I17)

Comme vu dans le [**chapitre consacré à la restauration de bases de données**](#_Restaurer_une_base), lors de la création des tables de la base de données au moyen de la commande « **php artisan make:model <NomDeLaTable>** », il était nécessaire, par la suite, de migrer les tables et leur structure dans celle-ci.

Ces migrations ont été effectuées, au moyen de la commande « **php artisan make:migration <NomDeLaMigration>** » qui va générer un fichier modifiable pour chaque table afin d’intégrer les champs et leurs propriétés (noms des champs de la table, longueur de champs, etc.) dans celui-ci avant de migrer les tables dans la base de données via la commande « **php artisan migrate** » qui effectuera la migration des tables dont les informations sont présentes dans les fichiers créés précédemment.

#### Définition d’un jeu de données test (AA3 C1 I18)

Un jeu de données test, ou seeder, permet d’insérer des données initiales ou des données fictives dans la base de données à des fins de testing de l’application et de la base de données elle-même.

Dans le cas de données initiales, cela concerne des données comme par exemple des listes de codes postaux, de villes, de pays, etc.[[33]](#footnote-33)

Comme déjà expliqué dans le [**chapitre au sujet de la restauration de base de données**](#_Restaurer_une_base), la commande « **php artisan make:seeder <NomDuSeeder>** » est utilisée pour générer une classe de seeder. Ensuite les données initiales ou de test peuvent être créées en fonction de la structure de la table pour laquelle le seeder est créé.

La commande « **composer dump-upload** » permet de régénérer le fichier autoloader de « Composer » après avoir ajouté la classe de seeder au fichier « DatabaseSeeder.php ».

Enfin pour implémenter ces données dans la base de données après avoir suivi les étapes précédentes, la commande « **php artisan db:seed** » doit être utilisée.

#### Implémentation de tests (AA3 C1 I19)

PLUS D’INFOS QUAND LES TESTS SERONT IMPLÉMENTÉS (TESTS UNITAIRES ET FONCTIONNELS)

#### Sécurisation de l’accès aux pages (AA3 C1 I20)

VOIR SI IDEM QUE CHAPITRE 2.7.1. AUTHENTIFICATION ET CE QU’IL FAUT Y AJOUTER

#### Incorporation et utilisation d’un progiciel (AA3 C1 I21)

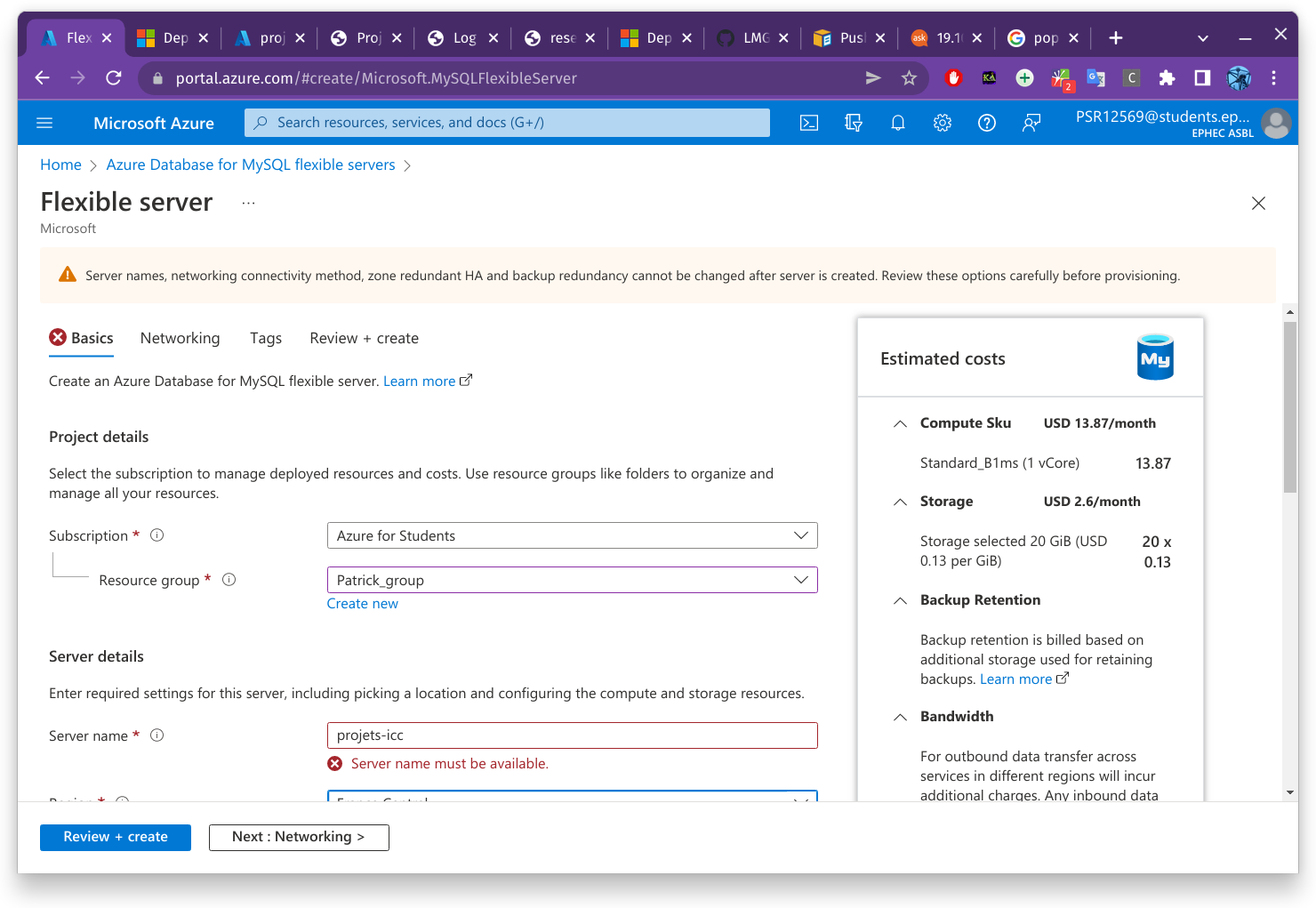
### Déploiement sur serveur (AA3 C3)

À REMPLIR LORSQUE LES POINTS 3.2.2.1. ET 3.2.2.2. SERONT REMPLIS

#### Déploiement du site sur serveur test (AA3 C3 I1)

##### Création du serveur MySQL sur Azure

Dans un premier temps, pour le déploiement, il est nécessaire de créer un serveur de base de données MySQL comme montré sur l’image suivante.



Lors de la création du serveur, il est nécessaire de rester premièrement attentif à choisir la bonne région pour le serveur. En effet, bien que la plupart des visiteurs viennent d’Europe, il est recommandé de choisir la région de base (East US).

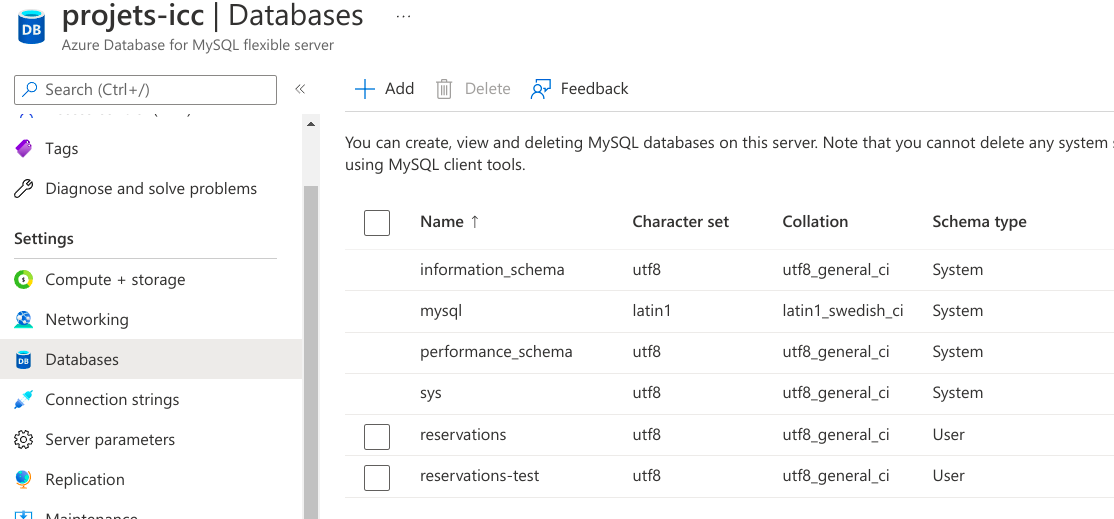
Ensuite il faut adapter celui-ci aux besoins du site, choisir le nombre de processeurs, de cœurs, la fréquence, la taille de la RAM ou du disque dur.

Il faut ensuite choisir un mot de passe pour l’administrateur du serveur SQL et, optionnellement, configurer le backup automatique.

##### Création des bases de données de test

La base de données de test « **reservations-test** »contiendra des données fictives destinées aux développeurs et testeurs de l’application.

Ces données peuvent être créées au moyen soit du DUMP de la base de données, soit au moyen des modèles de tables et des seeders créés au moyen du Framework comme nous l’avons vu précédemment.



##### Modification du fichier d’environnement

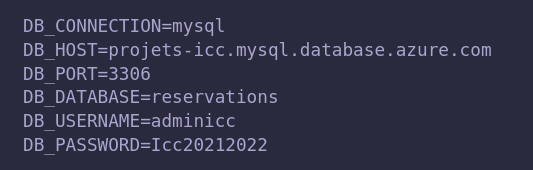
Pour pouvoir lier Laravel à la base de données Azure nouvellement créée, il faut modifier le fichier .env situé dans le dossier racine de l’application.



Cette étape réalisée, les données d’environnement suivantes doivent être modifiées pour pouvoir se connecter correctement au serveur :

* **DB\_HOST** qui est le lien du serveur MySQL sur Azure
* **DB\_DATABASE** qui est le nom de la base de données
* **DB\_USERNAME** et **DB\_PASSWORD** qui sont respectivement le nom d’utilisateur et le mot de passe de l’administrateur

Dans le cas de notre base de données, cela a été modifié comme suit :



Après cette étape, si un DUMP n’est pas utilisé, il faudra exécuter deux commandes pour effectuer une migration des tables et créer des données à partir des seeders. Ces commandes sont, comme vu précédemment, « **php artisan migrate** » et « **php artisan db:seed** ».

Il est à noter que la modification du fichier d’environnement n’est nécessaire que si l’application tourne en local. Si l’application est hébergée sur Azure, il sera nécessaire d’utiliser une autre méthode comme décrit au point relatif à l’installation de l’extension.

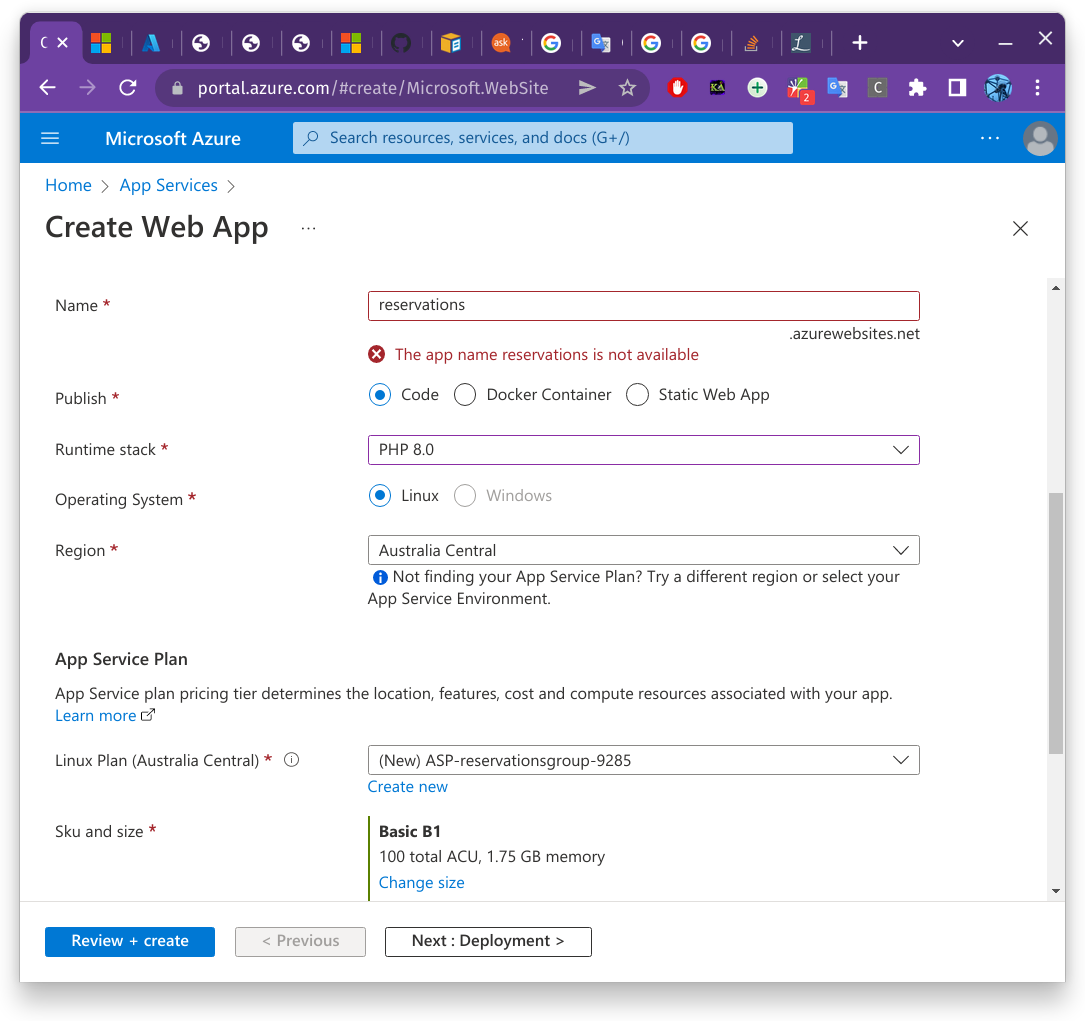
Enfin, il est également possible de créer plusieurs fichier .env au sein du dossier et adapter la commande « **php artisan serve** » afin qu’elle puisse utiliser la base de données que l’on souhaite employer comme dans les exemples suivants :

* **php artisan serve** permet de lancer l’application en local avec une base de données locale
* **php artisan serve –env=.env.test** permet de lancer l’application en local avec une base de données test sur Azure
* **php artisan serve –env=env.production** permet de lancer l’application en local avec une base de données production sur Azure

##### Création de l’application web de test

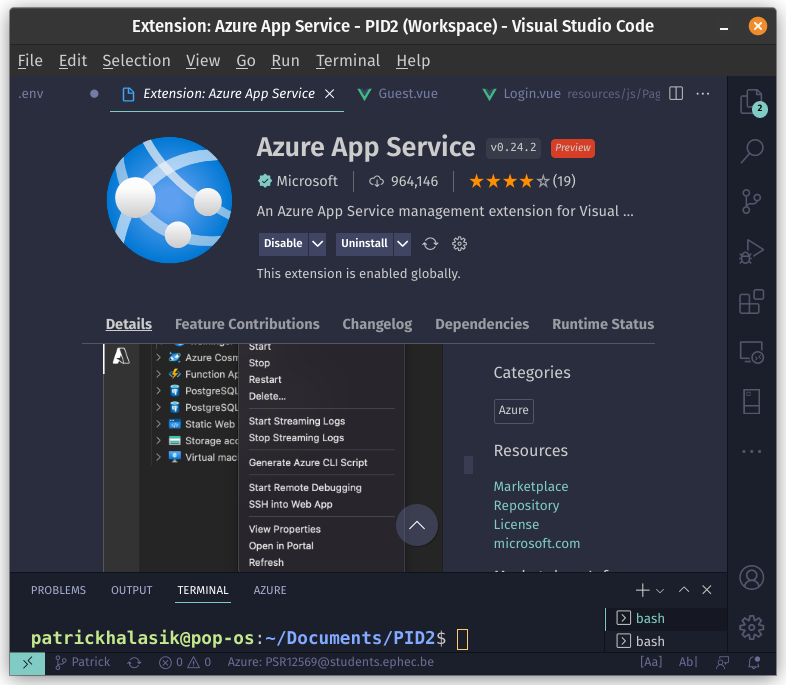
Un application de test sera créée sur Azure au moyen de PHP 8.0 et comme pour le serveur SQL, il sera important de choisir une configuration adaptée.

Dans le cas de l’application de test, la version gratuite est largement suffisante étant donné que la performance, la stabilité et la sécurité ne sont nullement la priorité d’une application de test.

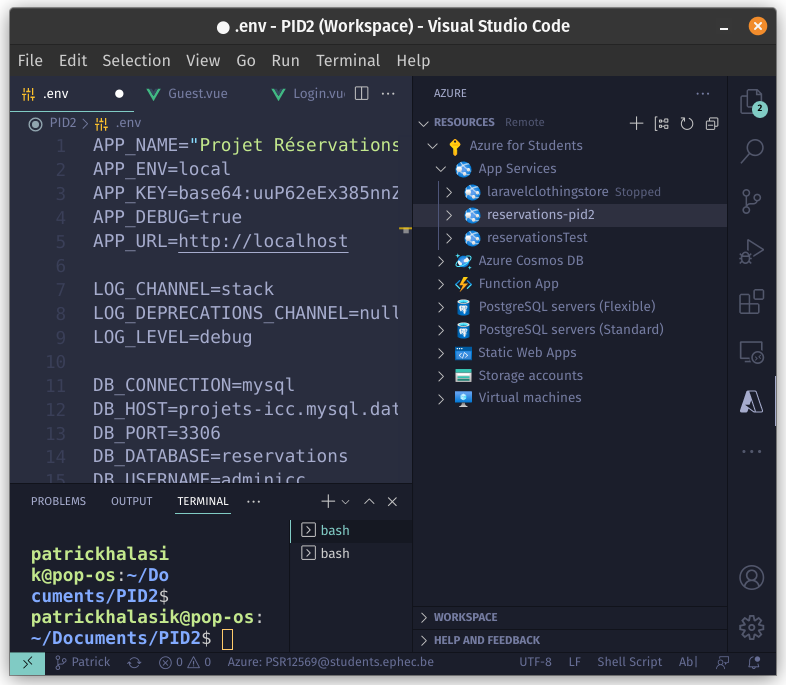


##### Installation de l’extension « Azure App Service » sur Visual Studio Code et déploiement

Pour pouvoir déployer le code vers l’application Azure, **l’extension Azure App Service** de Visual Studio Code sera utilisée comme montré sur l’image ci-dessous.

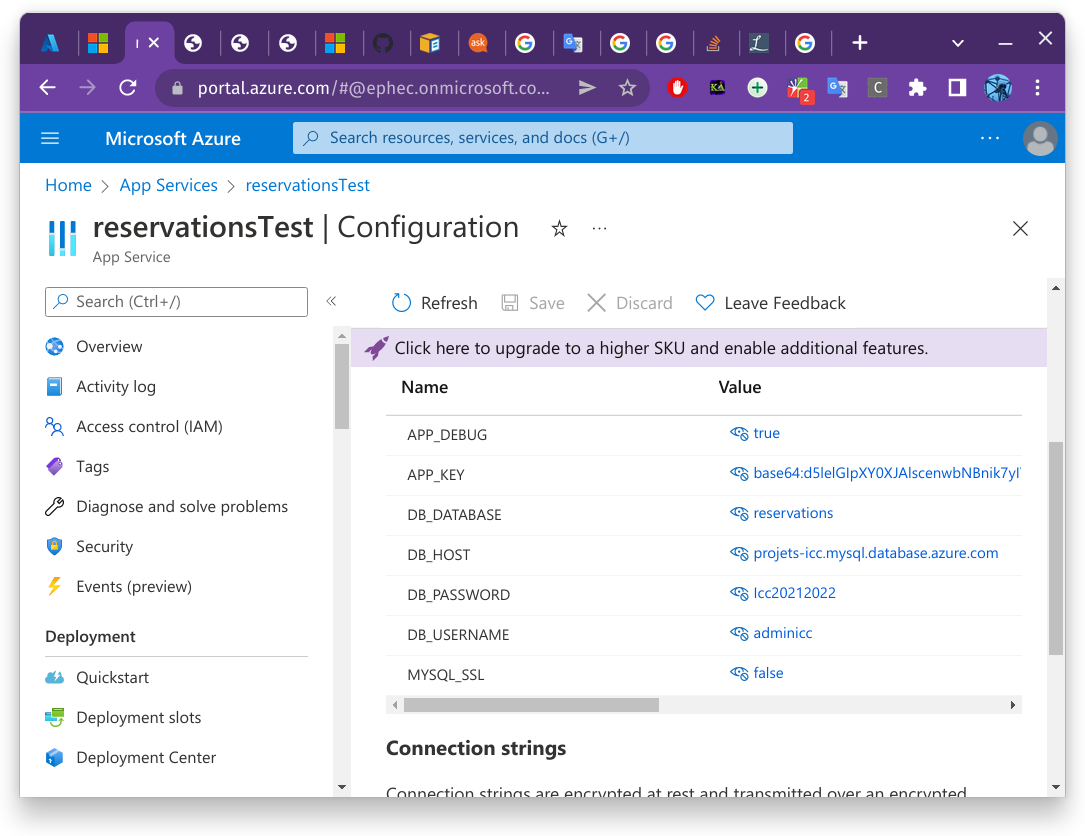


Après s’être authentifié, il faut choisir l’application où le déploiement du code doit avoir lieu et ensuite faire un clic droit et choisir « **Deploy to Web App...** »



##### Création de variables d’environnement sur l’application web Azure

Une application web Azure ne va généralement pas utiliser le fichier .env du dossier racine et **va utiliser les variables d’environnement** dans la configuration Azure en lieu et place du fichier précité.

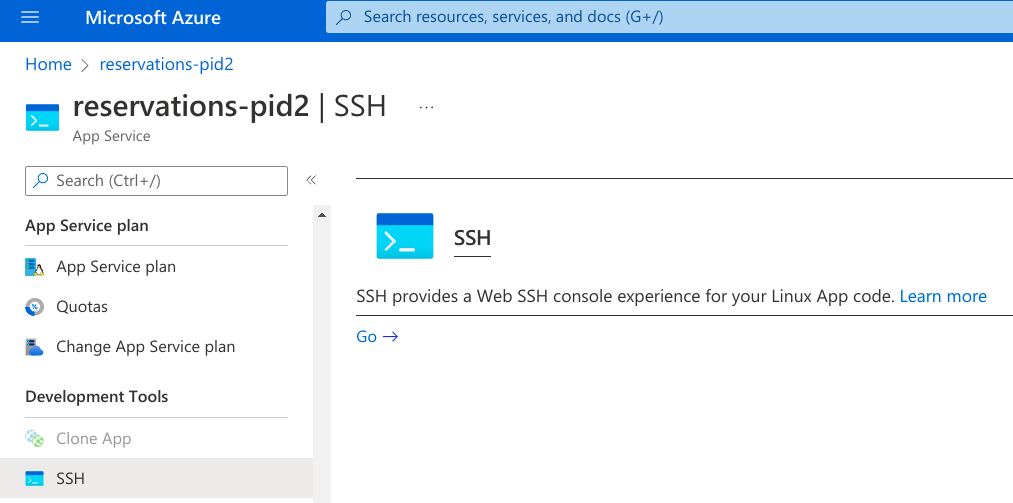


Cette configuration sera évidemment différente, selon s’il s’agisse de l’application de test ou production.

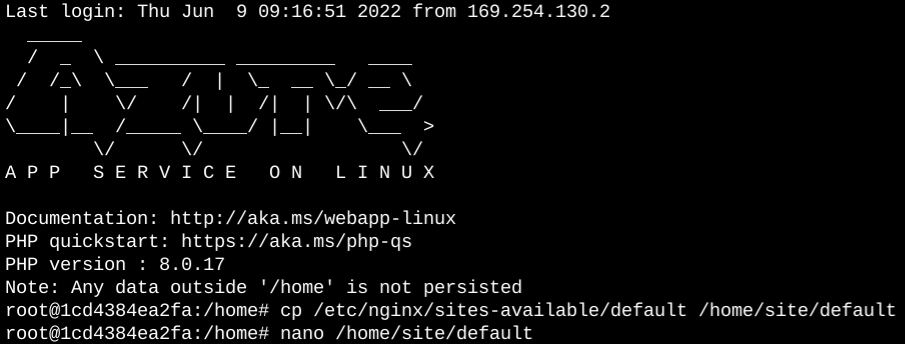
Par exemple, **APP\_DEBUG** est inutile dans un environnement de production car cette option est destinée aux développeurs de l’application et **MYSQL\_SSL** devrait être activée en production afin que l’application soit plus sécurisée

##### Changement de la racine de l’application web

Vu que l’application utilise Vue.js, il est indispensable de changer la racine de l’application afin qu’elle puisse fonctionner correctement. Pour ce faire, il faut ouvrir la console SSH de l’application web comme montré sur l’image ci-dessous.



Ensuite, il est nécessaire de copier la configuration par défaut (etc/nginx/sites-available/default) dans le dossier de l’application et il faut modifier son contenu pour qu’elle soit correcte.



Par défaut, la valeur root (racine de l’application) est home/site/wwwroot. Cependant si l’application Laravel utilise Vue.js, il est nécessaire de changer cette valeur par home/site/wwwroot/public.

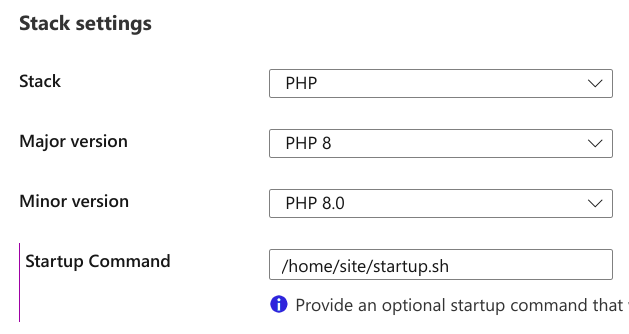
Il est à noter que cette étape n’est pas nécessaire si une application Laravel classique, avec fichiers blade, est utilisée.



Ensuite, il faut créer un fichier script startup.sh (à l’aide de la commande touch startup.sh sur Linux). Ce script va permettre d’écraser la configuration par défaut à chaque (re)démarrage de l’application.



Finalement, dans la configuration de l’application, il faut placer ce fichier script dans la commande de démarrage (Startup Command).



##### Accès à l’application de test

L’adresse de l’application de test est <http://reservationstest.azurewebsites.net>.

Le login est « admin@admin.com » et le mot de passe est « adminadmin » pour avoir accès à tout le site.

#### Déploiement du site sur serveur de production (AA3 C1 I2)

La procédure pour le déploiement du site sur serveur de production s’est déroulé de la même manière que pour le serveur test. Les points suivants aborderont les éventuelles différences par rapport au chapitre précédent.

##### Création des bases de données de production

Ce chapitre est identique à celui concernant le déploiement sur serveur de test, excepté que pour la base de données de production **reservations**, celle-ci contiendra des données réelles destinées au(x) client(s) de l’application.

##### Création de l’application web de production

Ce chapitre est identique à celui concernant la création de l’application web de test, excepté que pour l’application web de production il sera nécessaire de faire attention à la configuration et adapter le nombre de cœurs, la fréquence ainsi que la mémoire.

##### Accès à l’application de production

Ce chapitre est identique à celui concernant l’accès à l’application de test, excepté que pour accéder à celle-ci, le chemin est <http://reservations-pid2.azurewebsites.net>, le login et mot de passe restants identiques.

Également à noter qu’en situation réelle, l’application de production doit être plus sécurisée. Voici quelques étapes à faire qui sont indispensables pour une vraie application de production Azure :

* **Configuration du pare-feu du serveur SQL:** on autorise seulement l’application de test, de production, ainsi que les développeurs de l’application
* **Configuration du back-up automatique:** back-up de la base de données de production hebdomadaire
* **Configuration HTTPS/SSL:** utilisation d’un certificat SSL et hébergement du site sur le protocole HTTPS pour plus de sécurité
* **Autres:** authentification SQL via Managed Identity afin de ne plus utiliser de nom d’utilisateur et mot de passe, utilisation de services payants Azure comme les alertes de sécurité, ...

# Analyse

Lors du début de ce travail, chaque membre du groupe partait avec des bases différentes, certains avaient déjà travaillé avec Laravel, d’autres sur le PHP. L’état d’avancement des connaissances était donc de base disparate et ceux qui en avaient ont aidé les autres à acquérir le bagage nécessaire pour pouvoir développer l’application, ce qui s’est fait assez rapidement.

Ce travail aura permis également de faire un travail de groupe au moyen d’outils rencontrés dans le milieu professionnel tels que GitHub ou Laravel permettant à chacun de découvrir, en partie, la réalité du terrain.

Chaque personne dans le groupe a pu apporter sa contribution en fonction de ses forces pour permettre la réalisation de l’application dans les temps impartis.

L’entraide dans le groupe a été bénéfique pour tous afin de parvenir à résoudre les difficultés que nous avons pu rencontrer lors du travail tant en ce qui concerne les difficultés technique que celles concernant la mise à niveau des connaissances de chacun.

Grâce à ces aspects du groupe, l’entièreté a pu acquérir plus de connaissances sur la réalité du terrain, la dynamique d’un groupe et sur les outils et technologies utilisés.

Les plus grandes difficultés rencontrées dans le cadre de ce projet concernaient premièrement le temps imparti pour se familiariser avec une nouvelle technologie, celle du Framework, produire une application fonctionnelle répondant aux critères d’évaluation, ainsi que le départ de certains membres du groupe en cours de projet.

Pour pallier à ces difficultés, chaque membre a apporté son lot de connaissance aux autres comme expliqué ci-dessus pour pouvoir ensuite se focaliser sur la réalisation de l’application et concernant le départ des membres, il a été plus difficile de finaliser l’application et le groupe a dû prendre des congés supplémentaires, ce qui nous a quand même permis de voir une réalité du monde du travail, car un départ est toujours possible, qu’il soit définitif (démission, décès, etc.) ou temporaire (congé, maladie, etc.) et dans ces cas-là, les collègues restants doivent accepter une surcharge de travail et faire des heures supplémentaires et travailler plus sur le projet en cours.

Au sujet du temps imparti, il est à noter que le cours de Projet d’intégration de développement est donné à une période de l’année ou il y a beaucoup plus de charge de travail qu’en début d’année et tous les membres du groupe s’accordent pour dire qu’il serait préférable de donner ce cours dès le début de l’année et l’étaler jusque la fin à l’instar du cours de « Projet de développement Web » comme cela est de plus demandé par nombre d’étudiants au fil des années.

De plus, cela permettrait aux étudiants de découvrir l’environnement des Framework plus tôt dans l’année, ceci étant importants pour le développement de l’Épreuve Intégrée.

# Conclusion

La problématique de ce projet concernait la réalisation d’une application de gestion de réservations de spectacles à réaliser en groupe au moyen d’un Framework et il devait être répondu à divers critères d’évaluation, avec notamment la production d’un cahier de charges succinct, un rapport documentant la manière de réaliser le projet, ainsi qu’une partie d’analyse.

Le premier volet de ce rapport était consacré à remplir un cahier de charges succinct reprenant les informations essentielles sur la structure générale de l’application, sa sécurité, ses services tant pour les utilisateurs que les administrateurs, les outils de développement et déploiement utilisés, l’API consommées ainsi que le développement futur de l’application.

Le deuxième volet du présent rapport répondait aux demandes d’évaluation du chargé de cours pour permettre de visualiser les acquis du groupe dans la réalisation du travail demandé avec les explications de la méthode de travail, notamment pour implémenter, restaurer ou gérer une base de données, installer un Framework, utiliser un dépôt externe pour coordonner le travail, utiliser un ORM, définir des jeux de données test ou encore pour déployer le site.

Le dernier volet traite de l’apport de ce cours pour les membres du groupe de manière globale au moyen d’une analyse succincte permettant de se faire une idée des compétences supplémentaires qui ont pu être acquises.

En finalité de ce rapport, il a pu être mis en évidence les compétences et nouvelles connaissances acquises grâce à ce projet ainsi que la réponse apportée dans le cadre de la demande initiale de production d’une application de gestion de réservations.

Enfin dans l’analyse, il a pu être mis en évidence l’apport de tous les membres entre eux au niveau des connaissances et de l’entraide, l’apport de ce cours et l’aspect relatif à la contrainte de temps, ainsi que les solutions pour surmonter les difficultés rencontrées et plus particulièrement la suggestion d’étaler ce cours sur toute l’année à l’instar du cours de « Projet de développement Web » permettant de répartir la charge de travail de ce projet conséquent et donner aux étudiants futurs l’opportunité de découvrir les Framework plus tôt dans le but de s’y familiariser plus rapidement pour la production de l’Épreuve Intégrée.

# Bibliographie, Webographie

La présente bibliographie reprendra toutes les références évoquées dans le rapport, et ce, de manière alphabétique.

## Image de la page de garde

LARAVEL.COM. S.d. *Eloquent: Getting Started*. Site Web sur INTERNET. <https://laravel.com/docs/9.x/eloquent>. Dernière consultation le 05‑06‑2022.

## Bibliographie des autres pages

APICAL.XYZ. Dernière révision le 07-02-2019. *Les fichiers de seeds pour insérer les données initiales dans la BD*. Site Web sur INTERNET. <https://apical.xyz/fiches/insertion\_des\_donnees\_initiales/les\_fichiers\_de\_seeds\_pour\_inserer\_les\_donnees\_initiales\_dans\_la\_bd>. Dernière consultation le 05‑06‑2022.

BACK4APP.COM. S.d. *La signification du Back-End*. Site Web sur INTERNET. <<https://blog.back4app.com/fr/la-signification-du-backend-et-tout-ce-que-vous-devez-savoir-a-son-sujet/>>. Dernière consultation le 05-06-2022.

FUNINFORMATIQUE.COM. 31-05-2022. *Faille XSS, c’est quoi et comment s’en protéger ?*. Site Web sur INTERNET. <https://www.funinformatique.com/faille-xss-comment-lexploiter-et-sen-proteger/>. Dernière consultation le 05‑06‑2022.

GITHUB.COM. S.d. *Laravel IDE Helper*. Site Web sur INTERNET. <https://github.com/barryvdh/laravel-ide-helper>. Dernière consultation le 05‑06‑2022.

HOSTINGER.FR. 16-03-2022. *Les commandes GIT que vous devez absolument connaître !*. Site Web sur INTERNET. <https://www.hostinger.fr/tutoriels/commandes-git>. Dernière consultation le 05‑06‑2022.

JOURNALDUNET.FR. Mise à jour le 20-01-2019. *Framework ou infrastructure logicielle : définition et traduction*. Site Web sur INTERNET. <<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/>>. Dernière consultation le 05-06-2022.

KINSTA.COM. 24-05-2022. *Le framework PHP Laravel – la construction d’applications web pour tous*. Site Web sur INTERNET. <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/>. Dernière consultation le 05‑06‑2022.

LARAVEL.COM. S.d. *Eloquent: Getting Started*. Site Web sur INTERNET. <https://laravel.com/docs/9.x/eloquent>. Dernière consultation le 05‑06‑2022.

LARAVEL.COM. S.d. *Installation – Laravel*. Site Web sur INTERNET. <https://laravel.com/docs/4.2>. Dernière consultation le 05‑06‑2022.

LARAVEL.SILLO.ORG. 01-10-2020. *Cours Laravel 8 – les tests*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-8-les-tests/#:~:text=Laravel%20utilise%20PHPUnit%20pour%20effectuer%20les%20tests%20unitaires.,de%20bien%20isoler%20les%20tests.>. Dernière consultation le 05‑06‑2022.

LARAVEL.SILLO.ORG. 10-10-2017. *Cours Laravel 5.5 – La sécurité*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-5-5-la-securite/>. Dernière consultation le 05‑06‑2022.

MARKETPLACE.VISUALSTUDIO.COM. Dernière mise à jour le 06-03-2022. *Laravel Blade Snippets*. Site Web sur INTERNET. <https://marketplace.visualstudio.com/items?itemName=onecentlin.laravel-blade>. Dernière consultation le 05‑06‑2022.

MARKETPLACE.VISUALSTUDIO.COM. Dernière mise à jour le 22-02-2022. *PHP Intelephense*. Site Web sur INTERNET. <https://marketplace.visualstudio.com/items?itemName=onecentlin.laravel-blade>. Dernière consultation le 05‑06‑2022.

MDN. S.d. *CSRF – Glossaire MDN*. Site Web sur INTERNET. <https://developer.mozilla.org/fr/docs/Glossary/CSRF>. Dernière consultation le 05‑06‑2022.

PHP.NET. S.d. *PHP : Configuration requise pour l'installation – Manual*. Site Web sur INTERNET. <https://www.php.net/manual/fr/install.windows.requirements.php>. Dernière consultation le 05‑06‑2022.

REDHAT.COM. 24-10-2019. *L'Open Source, qu'est-ce que c'est ?*. Site Web sur INTERNET. <https://www.redhat.com/fr/topics/open-source/what-is-open-source#:~:text=%C3%80%20l'origine%2C%20l',le%20code%20%C3%A0%20sa%20convenance.>. Dernière consultation le 05‑06‑2022.

SOFT-BUILDER.COM. 2022. *Data Generator*. Site Web sur INTERNET. <https://soft-builder.com/fr/sb-data-generator/>. Dernière consultation le 05-06-2022.

WIKIPEDIA.ORG. Dernière modification le 05-05-2022. *GitHub*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/GitHub>. Dernière consultation le 05‑06‑2022.

WIKIPEDIA.ORG. Dernière modification le 07-05-2022. *Cross-site scripting*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/Cross-site\_scripting>. Dernière consultation le 05‑06‑2022.

WIKIPEDIA.ORG. Dernière modification le 10-09-2021. *Dump*. Site Web sur INTERNET. <<https://fr.wikipedia.org/wiki/Dump>>. Dernière consultation le 05-06-2022.

WIKIPEDIA.ORG. Dernière modification le 20-05-2022. *Visual Studio Code*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/Visual\_Studio\_Code>. Dernière consultation le 05‑06‑2022.

WRITECODE.FR. 01-05-2021. *Tutoriel : Les fixtures*. <https://writecode.fr/tutoriel/les-fixtures>. Dernière consultation le 05-06-2022.

1. BACK4APP.COM. S.d. *La signification du Back-End*. Site Web sur INTERNET. <<https://blog.back4app.com/fr/la-signification-du-backend-et-tout-ce-que-vous-devez-savoir-a-son-sujet/>>. Dernière consultation le 05-06-2022. [↑](#footnote-ref-1)
2. WIKIPEDIA.ORG. Dernière modification le 10-09-2021. *Dump*. Site Web sur INTERNET. <<https://fr.wikipedia.org/wiki/Dump>>. Dernière consultation le 05-06-2022. [↑](#footnote-ref-2)
3. WRITECODE.FR. 01-05-2021. *Tutoriel : Les fixtures*. <https://writecode.fr/tutoriel/les-fixtures>. Dernière consultation le 05-06-2022. [↑](#footnote-ref-3)
4. JOURNALDUNET.FR. Mise à jour le 20-01-2019. *Framework ou infrastructure logicielle : définition et traduction*. Site Web sur INTERNET. <<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/>>. Dernière consultation le 05-06-2022. [↑](#footnote-ref-4)
5. BACK4APP.COM. S.d. *La signification du Back-End*. Site Web sur INTERNET. <<https://blog.back4app.com/fr/la-signification-du-backend-et-tout-ce-que-vous-devez-savoir-a-son-sujet/>>. Dernière consultation le 05-06-2022. [↑](#footnote-ref-5)
6. SOFT-BUILDER.COM. 2022. *Data Generator*. Site Web sur INTERNET. <https://soft-builder.com/fr/sb-data-generator/>. Dernière consultation le 05-06-2022. [↑](#footnote-ref-6)
7. REDHAT.COM. 24-10-2019. *L'Open Source, qu'est-ce que c'est ?*. Site Web sur INTERNET. <https://www.redhat.com/fr/topics/open-source/what-is-open-source#:~:text=%C3%80%20l'origine%2C%20l',le%20code%20%C3%A0%20sa%20convenance.>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-7)
8. APICAL.XYZ. Dernière révision le 07-02-2019. *Les fichiers de seeds pour insérer les données initiales dans la BD*. Site Web sur INTERNET. <https://apical.xyz/fiches/insertion\_des\_donnees\_initiales/les\_fichiers\_de\_seeds\_pour\_inserer\_les\_donnees\_initiales\_dans\_la\_bd>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-8)
9. PHP.NET. S.d. *PHP : Configuration requise pour l'installation – Manual*. Site Web sur INTERNET. <https://www.php.net/manual/fr/install.windows.requirements.php>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-9)
10. KINSTA.COM. 24-05-2022. *Le framework PHP Laravel – la construction d’applications web pour tous*. Site Web sur INTERNET. <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-10)
11. LARAVEL.SILLO.ORG. 01-10-2020. *Cours Laravel 8 – les tests*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-8-les-tests/#:~:text=Laravel%20utilise%20PHPUnit%20pour%20effectuer%20les%20tests%20unitaires.,de%20bien%20isoler%20les%20tests.>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-11)
12. LARAVEL.COM. S.d. *Installation – Laravel*. Site Web sur INTERNET. <https://laravel.com/docs/4.2>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-12)
13. KINSTA.COM. 24-05-2022. *Le framework PHP Laravel – la construction d’applications web pour tous*. Site Web sur INTERNET. <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-13)
14. KINSTA.COM. 24-05-2022. *Le framework PHP Laravel – la construction d’applications web pour tous*. Site Web sur INTERNET. <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-14)
15. MARKETPLACE.VISUALSTUDIO.COM. Dernière mise à jour le 06-03-2022. *Laravel Blade Snippets*. Site Web sur INTERNET. <https://marketplace.visualstudio.com/items?itemName=onecentlin.laravel-blade>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-15)
16. GITHUB.COM. S.d. *Laravel IDE Helper*. Site Web sur INTERNET. <https://github.com/barryvdh/laravel-ide-helper>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-16)
17. MARKETPLACE.VISUALSTUDIO.COM. Dernière mise à jour le 22-02-2022. *PHP Intelephense*. Site Web sur INTERNET. <https://marketplace.visualstudio.com/items?itemName=onecentlin.laravel-blade>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-17)
18. LARAVEL.SILLO.ORG. 10-10-2017. *Cours Laravel 5.5 – La sécurité*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-5-5-la-securite/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-18)
19. LARAVEL.SILLO.ORG. 10-10-2017. *Cours Laravel 5.5 – La sécurité*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-5-5-la-securite/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-19)
20. LARAVEL.COM. S.d. *Eloquent: Getting Started*. Site Web sur INTERNET. <https://laravel.com/docs/9.x/eloquent>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-20)
21. WIKIPEDIA.ORG. Dernière modification le 07-05-2022. *Cross-site scripting*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/Cross-site\_scripting>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-21)
22. FUNINFORMATIQUE.COM. 31-05-2022. *Faille XSS, c’est quoi et comment s’en protéger ?*. Site Web sur INTERNET. <https://www.funinformatique.com/faille-xss-comment-lexploiter-et-sen-proteger/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-22)
23. FUNINFORMATIQUE.COM. 31-05-2022. *Faille XSS, c’est quoi et comment s’en protéger ?*. Site Web sur INTERNET. <https://www.funinformatique.com/faille-xss-comment-lexploiter-et-sen-proteger/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-23)
24. LARAVEL.SILLO.ORG. 10-10-2017. *Cours Laravel 5.5 – La sécurité*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-5-5-la-securite/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-24)
25. MDN. S.d. *CSRF – Glossaire MDN*. Site Web sur INTERNET. <https://developer.mozilla.org/fr/docs/Glossary/CSRF>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-25)
26. LARAVEL.SILLO.ORG. 10-10-2017. *Cours Laravel 5.5 – La sécurité*. Site Web sur INTERNET. <https://laravel.sillo.org/cours-laravel-5-5-la-securite/>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-26)
27. WIKIPEDIA.ORG. Dernière modification le 20-05-2022. *Visual Studio Code*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/Visual\_Studio\_Code>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-27)
28. WIKIPEDIA.ORG. Dernière modification le 05-05-2022. *GitHub*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/GitHub>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-28)
29. WIKIPEDIA.ORG. Dernière modification le 05-05-2022. *GitHub*. Site Web sur INTERNET. <https://fr.wikipedia.org/wiki/GitHub>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-29)
30. HOSTINGER.FR. 16-03-2022. *Les commandes GIT que vous devez absolument connaître !*. Site Web sur INTERNET. <https://www.hostinger.fr/tutoriels/commandes-git>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-30)
31. HOSTINGER.FR. 16-03-2022. *Les commandes GIT que vous devez absolument connaître !*. Site Web sur INTERNET. <https://www.hostinger.fr/tutoriels/commandes-git>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-31)
32. LARAVEL.COM. S.d. *Eloquent: Getting Started*. Site Web sur INTERNET. <https://laravel.com/docs/9.x/eloquent>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-32)
33. APICAL.XYZ. Dernière révision le 07-02-2019. *Les fichiers de seeds pour insérer les données initiales dans la BD*. Site Web sur INTERNET. <https://apical.xyz/fiches/insertion\_des\_donnees\_initiales/les\_fichiers\_de\_seeds\_pour\_inserer\_les\_donnees\_initiales\_dans\_la\_bd>. Dernière consultation le 05‑06‑2022. [↑](#footnote-ref-33)