

PROJET 4 : SOLUTION

LEKISHVILI RATI



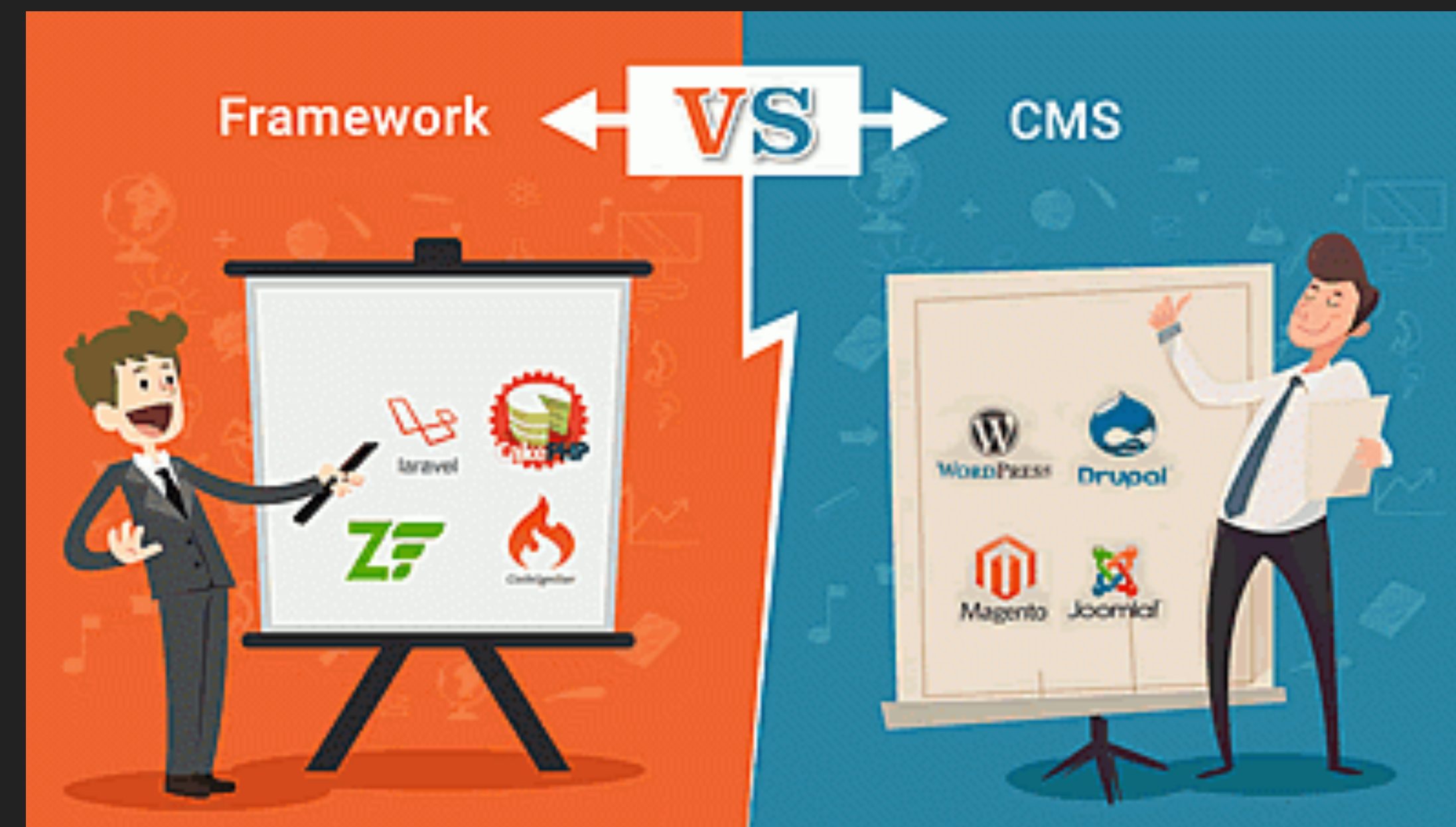
SOLUTION TECHNIQUE & SOLUTION FONCTIONNELLE



COMMENT CONSTRUIRE UNE APPLICATION WEB ?

Pour construire une application web, les principales méthodes sont :

- ▶ Le développement « **From scratch** » qui se traduit par « À partir de rien ».
- ▶ A l'aide d'un « **Framework** » ou infrastructure logicielle.
- ▶ En utilisant un « **CMS** » Content **M**anagement **S**ystem, qui se traduit en français par « Système de Gestion de Contenu » SGC.



DÉVELOPPEMENT « FROM SCRATCH »

* Le développement « from scratch » consiste à partir d'une feuille blanche pour créer l'application. Cela implique la conception de tous les composants de l'application par l'équipe de développement.

AVANTAGES

- ▶ Une application sur mesure.
- ▶ Sécurité. Ne ressemblant à aucune autre application il sera beaucoup plus compliqué à pirater.
- ▶ Une application très évolutive, offrant une personnalisation à souhait.

INCONVÉNIENTS

- ▶ Temps de développement et coût.
- ▶ Risque de « réinventer la roue », re-développer des fonctions existantes.

Personnalisation	★★★★★
Evolutivité	★★★★★
Sécurité	★★★★★
Coût / Temps	★☆☆☆☆

FRAMEWORK

* Un « framework » ou infrastructure logiciel en français, est un ensemble d’outils et de composants à la base d’une application. Les frameworks aident les développeurs à concevoir des applications grand public ou pour les entreprises.

AVANTAGES

- ▶ Une application évolutive et personnalisable.
- ▶ Sécurité.
- ▶ Temps et coût de développement.

INCONVÉNIENTS

- ▶ Plus restreint que le développement « from scratch », car plus encadré.

Personnalisation	★★★★★
Evolutivité	★★★★★
Sécurité	★★★★★
Coût	★★★★☆

CMS E-COMMERCE

* Les CMS sont une famille d'applications permettant la mise à jour de sites web dynamiques à des utilisateurs qui ne maîtrisent pas les langages de programmation pour gérer un site web.

AVANTAGES

- ▶ Temps et coût de développement.
- ▶ Vous avez la possibilité de gérer votre application web une fois celle-ci créée.
- ▶ Des fonctionnalités à portée de main, pour votre application.

INCONVÉNIENTS

- ▶ Moins évolutif et performant qu'un framework.
- ▶ Sécurité. Les CMS sont connus dans leurs modes de fonctionnement, donc plus simple à pirater.

Personnalisation	★ ★ ☆ ☆ ☆
Evolutivité	★ ★ ☆ ☆ ☆
Sécurité	★ ★ ★ ☆ ☆
Coût	★ ★ ★ ★ ★

QUELLE SOLUTION CHOISIR ?

FROM SCRATCH

Personnalisation	★★★★★★
Evolutivité	★★★★★★
Sécurité	★★★★★★
Coût / Temps	★☆☆☆☆

FRAMEWORK

Personnalisation	★★★★★
Evolutivité	★★★★★
Sécurité	★★★★★
Coût	★★★★☆

CMS

Personnalisation	★★★☆☆
Evolutivité	★★★☆☆
Sécurité	★★★★☆
Coût	★★★★★

Demandes d’OC Pizza à prendre en compte pour la conception de son application web :

- **Performance.**
- **Sécurité.**
- **Personnalisation.** Vous n’avez pas trouvé de système correspondant à vos besoins sur le marché des logiciels.
- Réalisable dans un **délai de 6 mois.**

RECOMMANDATION : « FRAMEWORK »

Pourquoi opter pour le développement à l'aide d'un framework ?

- **Le délai.** En six mois le développement à l'aide d'un framework est bien plus envisageable comparé au développement from scratch.
- **Une application personnalisée.** Si vous n'avez pas trouvé de logiciel correspondant à vos besoins, nous pouvons vous en faire un sur mesure, de préférence à l'aide d'un framework donnant beaucoup plus de liberté que les CMS.
- **Sécurité.** Un framework vous offrira une meilleure protection que les CMS contre un éventuel tentative de piratage, qui pouvant affecter votre organisation et mettre en danger l'activité du groupe entier.

QUEL FRAMEWORK CHOISIR ?

- * Nous vous conseillons le framework open source du langage Python: **Django**

Pourquoi Python et Django ?

- ▶ Python occupe la **première place** du classement des « langages les plus populaires pour le développement de sites et applications web » selon l'IEEE (Institute of Electrical and Electronics Engineers) la plus grande association mondiale de professionnels techniques.

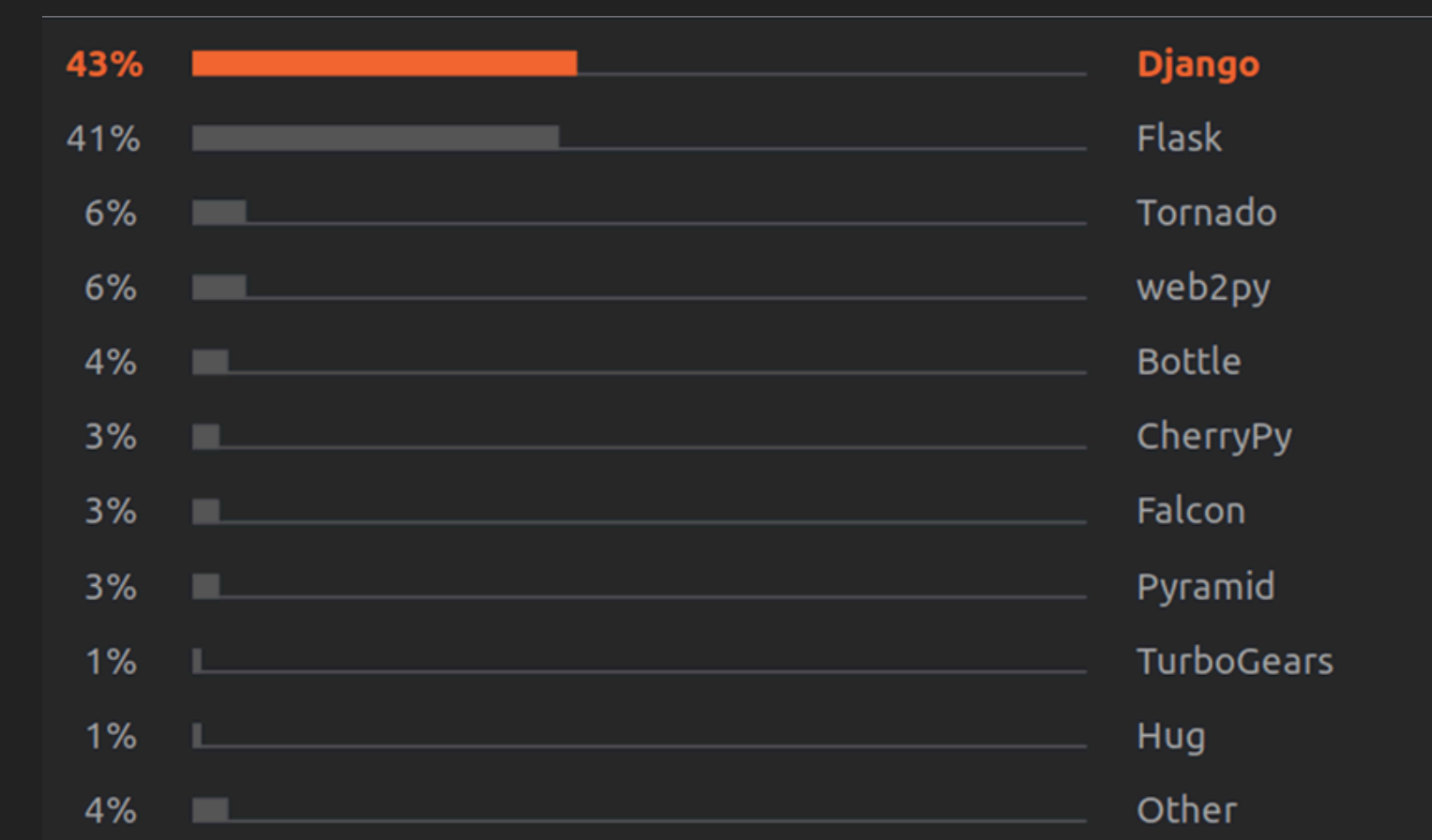
Source (2019) : <https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019>

- ▶ Django est le framework web **le plus populaire** du langage python devançant de 2 points « Flask » (43% contre 41%). L'avantage majeur de Django est qu'il est moins permissif que Flask, les applications web produites sont donc généralement plus **sécurisées**.

Source (2020) : <https://steelkiwi.com/blog/top-10-python-web-frameworks-to-learn/>

- * Pour rappel, la popularité d'un langage ou d'un framework est un point positif pour plusieurs raisons, deux sont évoquées ci-dessous :

- Témoigne de la **confiance** des développeurs qui les utilisent.
- Regroupe une **communauté active**, apportant des solutions rapidement aux problèmes rencontrés.



Source (2020) : <https://steelkiwi.com/blog/top-10-python-web-frameworks-to-learn/>

PAIEMENT EN LIGNE

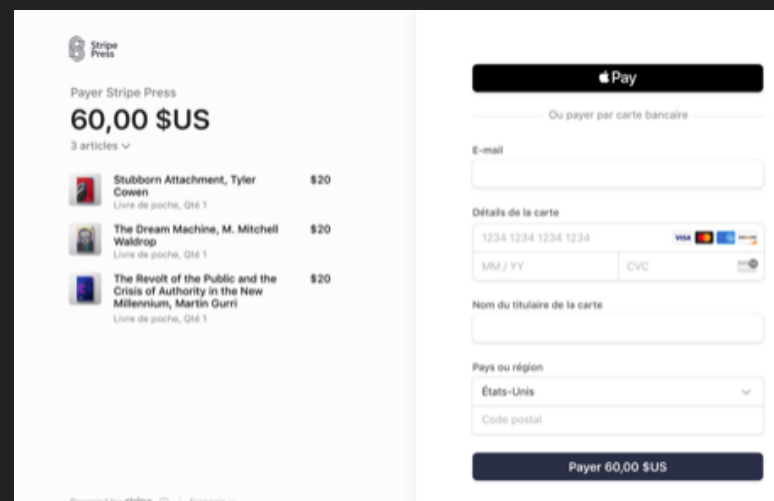
- * En 2020, la liste des acteurs proposant des solutions de paiement en ligne est longue. Choisir un acteur reconnu et de confiance est un choix très important, car cela peut freiner un utilisateur à effectuer le règlement en ligne.
- * À la suite de nos recherches, « **Stripe** » est l'acteur susceptible de répondre au mieux à vos attentes et aux attentes des consommateurs.

Pourquoi Stripe ?



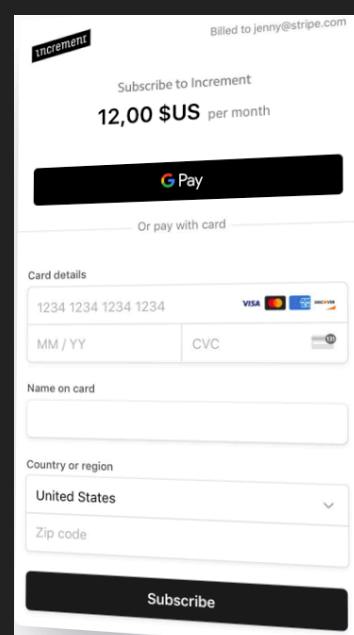
Stripe bénéficie d'une très bonne **image de marque**. Ils occupent la deuxième place sur le marché mondial des technologies spécialisées dans les paiement en ligne, devancé par PayPal.

Source (2020) : <https://kinsta.com/fr/blog/stripe-vs-paypal/#what-stripe-paypal-do>



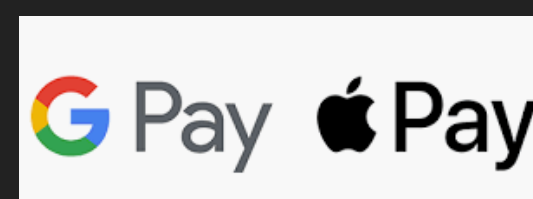
Stripe offre **une meilleure expérience** de paiement que PayPal, plus simple et nécessitant moins de clique, offrant ainsi une meilleure expérience utilisateur.

Source (2020) : <https://kinsta.com/fr/blog/stripe-vs-paypal/#what-stripe-paypal-do>



Optimiser pour les **appareils mobiles**. L'application propose de façon **dynamique** des portefeuilles mobiles tels qu'**Apple Pay** et **Google Pay**.

Source (2020) : <https://stripe.com/fr-br/payments/checkout>



source : wikipédia



OBJECTIFS

* Objectifs du dossier de spécifications fonctionnelles :

- ▶ Analyser les besoins du client.
- ▶ Identifier les acteurs primaires et secondaires qui interagiront avec le système.
- ▶ Utiliser les méthodologies existantes pour identifier les fonctionnalités dont les utilisateurs auront besoin pour se servir de l'application.
- ▶ Détailler les fonctionnalités qui composeront l'application.
- ▶ Echanger sur les fonctionnalités afin de répondre au mieux aux besoins du client.

MÉTHODOLOGIES

* Les méthodologies utilisées ont été les suivantes :

- UML (Unified Modeling Language) ou Langage de Modélisation Unifié
 - ▶ Diagramme de cas d'utilisation
 - ▶ Description textuelle d'un cas d'utilisation
 - ▶ Diagramme de paquets
 - ▶ Diagramme de séquence
 - ▶ Diagramme d'activité
- Impact mapping ou Diagramme d'impact
- Persona :

« Une personne fictive dotée d'attributs et de caractéristiques sociales et psychologiques et qui représente un groupe cible ».

source : wikipédia