

# Programmation Web

## Projet

Le but de ce projet est de réaliser un site de livraison de pizzas à domicile. Le site proposera à ses clients plusieurs sortes de produits :

1. Des entrées (Salade Caesar, Chicken Wings, ...) dont certaines avec une sauce au choix (par exemple, pour les Chicken Wings : sauce Barbecue, sauce Curry, Ketchup, ...).
2. Des boissons (Sprite, Coca, Coca-Light, ..., en 25cl, 33cl, 50cl, 1l, 1.5l, ...).
3. Des pizzas, de différentes tailles (Medium, Large, XLarge, ...) :
  - (a) des pizzas prédéfinies (Margherita, Napolitaine, Calzone, ...),
  - (b) des pizzas composables par le client à partir d'une liste d'ingrédients (thon, olive, anchois, pepperoni, ...) : six ingrédients maximum, les trois premiers gratuits, les autres en supplément.
4. Des menus de prix fixe, composés d'une ou plusieurs entrées, boissons et pizzas choisies par le client - par exemple :
  - (a) Extra Menu (1 entrée, 1 pizza Médium, 2 boissons d'au plus 33cl),
  - (b) Mega Menu (2 entrées, 2 pizzas Médium, 1 boisson d'au moins 1l),
  - (c) Giga Menu (2 entrées, 2 pizzas Médium, 1 boisson d'au moins 1l),
  - (d) etc.

## 1 Design et fonctionnalités du site

Le site doit être consultable aussi bien sur un téléphone portable ou sur une tablette que sur un écran large, avec le même confort d'usage (l'agencement des différents éléments d'une page doit donc s'ajuster en fonction de la taille de l'écran). Il doit être sur un nombre le plus réduit possible de pages – idéalement, une unique page – dont les contenus varieront dynamiquement. L'esthétique du site (couleurs, images, logos, décorations, animations, effets spéciaux) est totalement libre, mais elle doit être uniforme, et suffisamment riche visuellement.

La liste minimale de fonctionnalités décrite ci-dessous est imposée, mais elle n'est pas exhaustive - toute extension de cette liste est la bienvenue.

Lorsque le client consulte le site, il est initialement accueilli par une présentation de l'ensemble des menus. Le client doit pouvoir examiner l'ensemble des produits de chaque sorte (entrées, boissons, etc.), et passer librement d'une sorte à l'autre.

### 1.1 Ajout de produits au panier

Tout produit présenté au client sera accompagné de son nom, de son prix (pour la pizza composable, précisez "À partir de ..."), d'une image de ce produit, et d'un bouton permettant au client d'ajouter ce produit à son panier, initialement vide :

- La présentation d’une entrée avec sauce sera complétée par un menu déroulant permettant de choisir sa sauce.
- La présentation d’une pizza prédéfinie sera complétée par un menu déroulant permettant de choisir sa taille – l’affichage du prix de la pizza sera mis à jour à chaque changement de taille.
- Un clic sur la présentation de la pizza composable mènera à une présentation de l’ensemble des tailles de pizzas possibles, avec le prix de chaque taille.

Le choix d’une taille de pizza composable mènera le client à une présentation de l’ensemble de ses ingrédients possibles. Le client devra pouvoir sélectionner au plus 6 de ces ingrédients - tant qu’il n’en aura pas sélectionné au moins 3, les nouveaux ingrédients seront étiquetés par “gratuit”. À partir de 3 ingrédients sélectionnés, les nouveaux ingrédients seront marqués par un même prix de supplément (e.g. 1.5 euros). Le prix de la pizza composée sera visualisable en permanence, et mis à jour en permanence. Si le client compose une pizza identique à l’une des pizzas prédéfinies, leurs prix doivent être les mêmes.

Le client doit pouvoir sélectionner plusieurs fois le même ingrédient (double, triple, quadruple merguez), retirer un ingrédient sélectionné (avec dans ce cas une mise à jour de l’étiquetage des nouveaux ingrédients, gratuit ou non), finaliser l’ajout de sa pizza au panier, ou encore, annuler la composition de sa pizza et revenir à l’écran de présentation des pizzas.

- La présentation d’une pizza prédéfinie sera complétée par un bouton “Personnaliser”. Ce bouton permettra d’accéder à un choix de taille et d’ingrédients similaire à celui d’une pizza personnalisée, mais avec les ingrédients spécifique à cette pizza déjà sélectionnés.

Le montant total du panier du client sera visualisable en permanence, et mis à jour en continu au fil des ajouts.

## 1.2 Gestion du panier et finalisation de la commande

Le client doit pouvoir accéder au contenu courant de son panier, supprimer de ce panier les éléments de son choix, et revenir à l’une des présentations permettant d’ajouter des produits à son panier. Il peut aussi choisir de finaliser sa commande. Dans ce cas, le site lui demandera de remplir un formulaire précisant son nom, son prénom, son adresse, son ou ses codes d’entrée d’immeuble (optionnels), un numéro de portable et une adresse email, ainsi que l’heure de livraison souhaitée. Une fois ces données correctement entrées, la commande du client sera enregistrée, et le panier du client vidé.

## 1.3 Connection des livreurs

Les livreurs du site se connectent au site sur l’URI `/livraison`. Lorsqu’un livreur est connecté au site, le site lui indique, si elle existe, le détail d’une unique commande dont il a la charge (données du client, nom et prix de chaque article, prix total) : il s’agit de la commande non encore livrée d’heure de livraison la plus ancienne, et non encore prise en charge par un autre livreur. Le livreur peut à tout moment indiquer au site que cette commande a été effectivement livrée, le site lui indiquant alors dans ce cas, si elle existe, la commande suivante à prendre en charge.

## 2 Implémentation

Le serveur du site sera implémenté en faisant un usage libre mais approprié des *seuls* outils suivants : Bootstrap, jQuery, CSS, Node, les modules `express`, `ejs` et `pg` – PostgreSQL, pour la gestion d’une base de données des produits des clients, des commandes, des livreurs et de toute autre information utile au fonctionnement du site<sup>1</sup> (aucun ingrédient, entrée, sauce, composition de pizza prédéfinie, prix, données d’un client ou d’une commande, etc., ne doit apparaître en dur dans le code : ces informations seront toutes stockées dans la base de données).

Votre base de données devra être intégralement initialisée par un script `.sql` (et non par le code Javascript). Une fois initialisée (y compris avec plusieurs clients déjà enregistrés), la base doit permettre une démonstration immédiate et convaincante.

Noter que la qualité d’écriture du code sera l’un des critères de l’évaluation : factorisation maximale (*aucun* copier-coller), généricité et efficacité, découpage en fonctions (ou encore mieux, en modules) mise en page (sur 80 colonnes, jamais plus), commentaires, etc.

## 3 Modalités

**Composition des groupes.** Le projet doit être réalisé en binôme, sauf cas d’impossibilité majeure (groupe en nombre impair, conditions d’études particulières, etc.) - nous vous invitons dans ce cas à en discuter le plus tôt possible avec votre enseignant. La répartition des tâches au sein d’un groupe doit être raisonnablement équilibrée. En cas de déséquilibre avéré, les notes finales pourront être individualisées.

**Usage de Gitlab.** Votre projet doit être réalisé à l’aide d’un dépôt sur le Gitlab de l’université<sup>2</sup>. Ce dépôt doit être privé, créé par l’un des deux membres d’un binôme, l’autre membre du binôme étant invité comme développeur. Vous donnerez accès (au moins comme reporter) à ce dépôt à chacun de vos enseignants (et bien sûr à aucun autre utilisateur). La régularité et l’équilibrage des commits seront pris en compte dans l’évaluation.

**Individualité des projets.** De manière évidente, votre code doit être **strictement personnel**. Nous pouvons tolérer l’adaptation d’exemples repris sur la documentation des outils utilisés, mais en aucun cas le partage de code entre groupes – tous les rendus seront injectés dans un détecteur de plagiat insensible aux renommages ou aux changements de l’ordre des définitions<sup>3</sup>.

La qualité d’écriture du code sera l’un des critères de l’évaluation : factorisation maximale (*aucun* copier-coller), efficacité, découpage en fonctions (ou encore mieux, en modules) mise en page (sur 80 colonnes, jamais plus), commentaires, etc.

---

1. Ce projet n’est pas un projet de base de données. Une base de donnée basique mais fonctionnelle suffit.

2. [https://gaufre.informatique.univ-paris-diderot.fr/users/sign\\_in](https://gaufre.informatique.univ-paris-diderot.fr/users/sign_in). Utilisez le même identifiant et le même mot de passe que sur les machines de l’UFR.

3. La soutenance de projet est un examen comme un autre. Le plagiat en projet constitue une fraude aux examens, passible au pire de lourdes sanctions disciplinaires – ce cas s’est produit plus d’une fois dans cette UFR.

**Rendus des projets.** Le rendu sera une archive (*tar.gz* ou *zip*) sur moodle contenant l'intégralité des fichiers nécessaires au déploiement du site. En supposant postgresql installé et lancé, ce déploiement ne devra nécessiter que :

- la décompression de l'archive,
- l'exécution d'un script d'initialisation de la base de données nommé `init.sql`
- la commande `nodejs main.js`.