



TECH



v1.0 Projet Creative-Yumland

FILIERE préING2 • 2025-2026

AUTEURS C.LE BRETON - R.GRIGNON

E-MAILS caryl.le-breton1@cyu.fr - romuald.grignon@cyu.fr

DESCRIPTION GENERALE

- Le but de ce projet est de créer le site web d'un restaurant ou d'une chaîne de restaurants. Le but final est d'avoir une application multi utilisateurs fonctionnelle pour traiter la chaîne complète d'une commande, depuis le choix du client, jusqu'à la livraison, en passant par les phases de connexion, paiement, traitement par le restaurateur et le livreur.
- La fonctionnalité de base est celle pour le client qui arrive sur le site et qui peut choisir de consulter la carte avant d'aller manger sur place ou de choisir de commander en ligne (à emporter ou en livraison). Les clients qui commandent peuvent choisir des menus déjà configurés et/ou des plats à la carte (entrées, plats, desserts, boissons...).
- Les clients peuvent s'inscrire pour avoir accès à d'autres fonctionnalités :
 - ◆ ne pas ressaisir des informations personnelles (état civil, adresse de livraison, code interphone...)
 - ◆ retrouver des commandes précédentes
 - ◆ profiter de remises spécifiques (points bonus, produits offerts, pourcentage...)
- On considérera 4 profils d'utilisateurs avec des terminaux différents
 - ◆ Le client (qui commande sur le site)
 - ◆ L'administrateur (qui peut intervenir sur les profils depuis un ordinateur, il a accès à toutes les pages du site pour vérifier les problèmes)
 - ◆ Le restaurateur (qui configure les menus déjà configurés, prépare les commandes des clients, il visualise les commandes en cours sur une tablette tactile)
 - ◆ Le livreur (qui livre les commandes, il utilise un smartphone avec un petit écran et avec une connexion Internet dont le débit n'est pas toujours optimal. En période de froid il possède de gros gants qui l'empêche de cliquer avec précision sur l'écran).

- 
- Le projet sera divisé en plusieurs phases qui seront évaluées au fur et à mesure de l'avancement du semestre. Ces phases seront des étapes fonctionnelles qui suivront le rythme des chapitres du cours d'Informatique 4 (HTML + CSS, PHP, Javascript + DOM, requêtes asynchrones, bonnes pratiques de développement Web).
 - La **phase #1** va consister à créer la partie graphique côté client (définition du périmètre du restaurant + nom du site, définition de la charte graphique, affichage statique de différentes pages du site Internet, ...)
 - La **phase #2** va consister à créer la partie côté serveur (définition du stockage des données, traitement de l'inscription et de la connexion des utilisateurs, ...)
 - La **phase #3** va consister à ajouter côté client du code Javascript pour effectuer divers traitements dynamiques sur les pages et utiliser les requêtes asynchrones pour modifier les pages visibles côté client sans avoir à les recharger totalement.
 - La **phase #4** permettra de mettre en place (si ce n'est pas encore fait) les bonnes pratiques de développement Web. Cette phase sera évaluée lors de la soutenance finale.
 - Enfin une **soutenance orale** (présentation fonctionnelle et technique du site Internet) terminera l'évaluation du semestre pendant laquelle vous devrez présenter une **fonctionnalité innovante** non demandée dans ce cahier des charges. Cette présentation sera ponctuée de scénarios d'actions à réaliser, de questions-réponses/modifications de votre code.
 - Les différents critères attendus pour chaque phase vous seront donnés dans ce document qui sera remis à jour régulièrement. Pensez donc à consulter la page de cours régulièrement.



CRITERES GLOBAUX D'EVALUATION

- La constitution des groupes devra être faite en début de semestre, et une inscription sur la page de cours sera demandée, avec une période d'inscription limitée dans le temps.
Un dépôt de code pour chaque groupe de projet devra être créé juste après la constitution des groupes, et l'URL devra être renseignée dans la page de cours (période limitée dans le temps également).
- Chaque phase du projet sera limitée en temps (voir semainier sur la page de cours) avec des rendus à dates fixées à l'avance. Le contenu à réaliser sera fourni peu de temps avant le démarrage de chaque phase. Tous les critères attendus pour chaque phase seront listés proprement dans ce document qui sera mis à jour sur la page de cours.
- Pendant toute la durée du projet (semestre) il vous faudra effectuer régulièrement des commits de code sur votre dépôt. Cette injonction est là pour vous éviter de vous retrouver dans une procédure pour plagiat contre votre gré (le fait d'effectuer des commits réguliers vous permet de montrer une antériorité sur le code). L'autre avantage est de ne pas perdre de travail si l'une de vos machines venait à tomber en panne ou si l'un(e) des membres du groupe venait à effacer par erreur du code en local avant de l'avoir sauvegardé.
Il n'y aura donc **aucune réclamation recevable** pour une perte de travail car chaque modification qui fonctionne, aussi petite soit-elle, doit être sauvegardée sur le dépôt central de code au plus tôt, et chaque membre du groupe se doit d'être à jour régulièrement également.
- Bien entendu, si le dépôt central (gitHub.com, gitLab.com, ...), ET les machines des étudiant(e)s contenant les dernières modifications synchronisées du dépôt, venaient à tomber en panne en même temps, alors nous traiterions cet état (statistiquement peu probable) comme il se doit.
- Au début et à la fin de chaque phase, un commit devra être fait avec un intitulé comme suit : **[phase#1] DEBUT , [phase#2] FIN**, afin de bien identifier l'état des lieux au démarrage de la phase associée (cela sera plus pratique pour vos chargé(e)s de TD pour comparer les modifications effectuées).
Si vous avez besoin de faire un commit supplémentaire après votre livraison, indiquez à nouveau le message **[phase#2] FIN** et vos chargé(e)s de TDs prendront le commit le plus récent contenant ce message pour l'évaluer.
Si il n'y a pas de moyen simple pour les chargé(e)s de TD de repérer votre livraison parmi tous vos commits, alors dans ce cas le commit avec la date la plus proche antérieure à l'échéance sera prise en compte pour l'évaluation sans aucun moyen pour le groupe de faire de réclamation.



- Aucun commit fourni après une date limite ne sera évalué ! Néanmoins il vous faudra tout de même faire cette livraison tardive (sûrement nécessaire au global) car le projet ne s'arrête pas après chaque phase, il progresse de manière continue tout au long du semestre. Pensez donc à bien soigner vos livraisons à chaque phase. Profitez de la période entre deux phases pour soigner les détails du site Internet et/ou rajouter ce qui n'a pas pu l'être à temps pour le livrable.
- Lors de la soutenance, nous allons vous demander en direct la répartition de l'implication de chaque membre : par exemple si vous estimatez que l'investissement des 3 membres du trinôme est identique, vous donnerez des valeurs 33%, 33%, 33%. Si au contraire vous estimatez (tous) qu'un étudiant s'est investi différemment des autres vous pourrez fournir des pourcentages qui vous paraissent plus réalistes quant à l'investissement personnel de chacun (ex : 30%, 28%, 42%).
On parle d'investissement dans le projet, ce qui n'implique pas forcément le code, mais aussi toute l'organisation, les documents, le debug, les tests, les temps de réunion (présence), et les temps passés auprès d'un membre qui travaille sur une partie pour laquelle vous avez moins de valeur ajoutée mais votre présence lui permet de rester motivé jusqu'au bout et/ou d'obtenir un avis sur des questions qui pourraient se poser. Il est donc peu probable, si vous avez travaillé classiquement dans l'équipe, que votre pourcentage personnel soit de 0 % ou de 100 %.
La valeur de ces pourcentages pourra permettre de modifier les notes à partir de la note globale du projet, c'est à dire de transférer des parties de points entre les membres du groupe de manière à individualiser les notes. Il est de la responsabilité du groupe de préparer cette répartition avant la soutenance.

RESSOURCES UTILES

GitHub

- site Web : <https://github.com/>

Couleurs de sites Internet

- Paletton: <https://paletton.com>

Format HTML

- wikipedia : https://fr.wikipedia.org/wiki/Hypertext_Markup_Language

Format CSS

- wikipedia : https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

Langage JavaScript

- wikipedia : <https://fr.wikipedia.org/wiki/JavaScript>

Langage PHP

- wikipedia : <https://fr.wikipedia.org/wiki/PHP>

Format JSON

- wikipedia : https://fr.wikipedia.org/wiki/JavaScript_Object_Notation