v1.3 Projet Click-journeY

FILIERE préING2 • 2024-2025
AUTEURS R. GRIGNON – C. LE BRETON
E-MAILS romuald.grignon@cyu.fr – caryl.le-breton1@cyu.fr

DESCRIPTION GENERALE

- Le but de ce projet est de créer un site Web d'une agence de voyages qui propose des séjours déjà configurés au niveau des étapes.
- Cela signifie que le client qui arrive sur le site peut choisir des circuits déjà prévus et n'aura pas la possibilité de changer ces étapes (lieux géographiques). Par contre l'utilisateur pourra modifier des options à chaque étape (hébergement, restauration, activité, transport, …).
- C'est donc une agence qui propose des séjours « pseudo-sur-mesure ».
- Les utilisateurs du site peuvent s'inscrire et se connecter. Une fois inscrits, ils peuvent modifier des options de voyages et les acheter : une phase de paiement devra également être intégrée au site pour cette étape.
- 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).

- La ***phase #1*** va consister à créer la partie graphique côté client (définition du périmètre de l'agence de voyage + 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.
- La ***phase #4*** va consister à utiliser les requêtes asynchrones pour modifier les pages visibles côté client sans avoir à les recharger totalement.
- Enfin une ***soutenance orale*** (présentation fonctionnelle et technique du site Internet) terminera l'évaluation du semestre.
- 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.

Phase #1

- Cette phase concerne le côté client, avec un affichage de quelques pages en HTML + CSS
- On vous demande tout d'abord de définir le nom et le thème de votre site web (ex : ***My Trekking Planner*** pour un site de **randonnées pédestres**, ***Cold Cruise*** pour un site de **croisières en zones froides**, …). Il faut trouver un critère spécifique pour votre site : l'idée étant que chaque groupe de projet ait un thème différent. Vous verrez donc à valider ce choix au plus tôt avec votre chargé(e) de TD.
  Ensuite on vous demande de définir la charte graphique, c'est à dire la liste des couleurs (et leurs valeurs #RRGGBB) que vous allez utiliser, et pour quelle utilité (couleur des fonds, couleurs des polices, couleurs des liens, déjà visités ou non, titres, paragraphes, bordures de formulaires, taille du texte, etc…) .
  Vous pouvez vous aider du lien donné dans les ressources utiles de ce document pour choisir vos couleurs (*paletton*).
  Ces informations seront réunies dans un document de conception de votre site (au format PDF).
  Ce document pourra être amené à évoluer, donc pensez à archiver également le document d'origine (format odt, docx, …).
- Les pages à créer pour cette phase sont les suivantes :
  - Une page d'accueil par défaut avec la présentation du site
  - Une page pour la présentation du site qui reprend votre nom et votre thème + un champ de recherche rapide intégré
  - Une page spécifique pour rechercher des voyages avec plusieurs champs de filtrage (dates, lieux, options, …)
  - Une page avec un formulaire pour s'inscrire
  - Une page avec un formulaire pour se connecter lorsque l'on est déjà inscrit sur le site
  - Une page pour afficher le propre profil d'un utilisateur connecté avec des boutons pour modifier les différents champs (bouton 'modifier' ou avec un icône 'crayon', …)
  - Un page de type '*administrateur*' avec une liste d'utilisateurs inscrits et des boutons permettant de modifier une propriété de chaque utilisateur (ex : client VIP qui aurait une réduction des prix, bannissement du client qui ne pourrait plus acheter de voyages, …)
- Chaque page HTML doit être dans un fichier *.html* séparé.
- Le code CSS utilisé par toutes les pages doit être stocké dans un fichier à part et utiliser la charte graphique définie précédemment.
- Vous devez ajouter un fichier au format PDF qui comprendra le planning réalisé de votre groupe pendant la phase #1 (répartition des tâches, problèmes humains, organisationnels et techniques rencontrés, solutions apportées pour les résoudre). Ce document devra être mis à jour à chaque phase également, donc pensez à archiver le document d'origine (format odt, docx, …). Ce document sert de rapport de projet : il est différent du document de conception précédemment cité.

Phase #2

- Cette phase concerne le côté serveur, avec une gestion des données du site Internet et une génération dynamique des différentes pages.
- Dans cette phase on vous demande en premier lieu de définir avec précision le format des données que vous allez stocker dans le serveur. Cette étape est très importante pour vous permettre d'avoir une vision claire de ce que vous devez faire car il y a une multitude d'informations que vous allez devoir gérer en fonction des fonctionnalités de votre site Internet. Pour vous aider dans la conception, voici une liste non exhaustive des données potentielles à stocker :
  - ***Utilisateurs***
    - login
    - mot de passe
    - rôle (administrateur, normal, …)
    - informations (nom, pseudo, naissance, adresse, …)
    - dates (inscription, dernière connexion)
    - voyages (consultés, achetés, …)
  - ***Voyages***
    - titre
    - dates (début, fin, durée, …)
    - liste des étapes
    - spécificités du voyage
    - liste de personnes
    - prix total
    - éventuellement le statut (payé, en cours de modification)
  - ***Etapes***
    - titre
    - dates (arrivée, départ, durée
    - position géographique (position gps, nom de ville/lieu, …)
    - liste d'options par étape (activité sportive ou culturelle, hébergement, restauration, transport vers l'étape suivante, garde d'enfants, gestion du linge, …)
    - personnes supplémentaires (ex : un ami qui rejoint le groupe sur une étape et qui va partager les prestations)
  - ***Options***
    - liste de valeurs possibles
    - choix de l'utilisateur ou valeur par défaut
    - nombre de personnes pour cette option (on peut imaginer que tous les participants au voyage ne font pas forcément toutes les activités)
    - prix par personne (ou différents prix en fonction de l'age, du nb de personnes, etc.)
    - autres champs spécifiques (ex : la vitesse moyenne de déplacement pour une option de transport)
  - ***Paiement***
    - coordonnées bancaires (numéro de carte, date de validité, …)
    - lien avec l'utilisateur et le voyage choisi
    - date de transation

- On demande pour cette phase qu'il y ait au minimum ***3 utilisateurs normaux***, ainsi qu'au moins ***2 utilisateurs administrateurs*** déjà présents dans les fichiers. Ainsi il sera possible de tester la connexion sans avoir à s'inscrire.

  Il faudra également que les fichiers contiennent au minimum ***15 voyages différents*** (nombre d'étapes, options différentes possibles par étape, valeurs d'options par défaut pour chaque étape, prix, dates, …). Ainsi il sera possible de tester directement l'affichage des différents voyages, ainsi que les changements d'options, … .
  L'idée est de pouvoir visualiser des voyages avec des configurations multiples et ainsi voir toutes les fonctionnalités du site (et éventullement détecter des bugs).

- Les différentes fonctionnalités à intégrer pour cette phase sont les suivantes :
  - L'***inscription*** d'un utilisateur doit être fonctionnelle (remplissage du formulaire, vérification des informations fournies, stockage des informations du nouvel utilisateur dans les fichiers du serveur, …). Le choix de se connecter automatiquement après l'inscription ou non, est laissé libre.

  - La ***connexion*** d'un utilisateur doit être fonctionnelle via son login / mot de passe. La connexion d'un utilisateur doit être mémorisée dans une session PHP du serveur.
    Si un utilisateur connecté accède à l'URL de la page de connexion, il sera redirigé automatiquement vers sa page principale. A l'inverse, un utilisateur anonyme (non connecté) qui essaye d'afficher une page normalement protégée, sera redirigé automatiquement vers la page de connexion.

  - Si l'utilisateur qui est connecté est un ***administrateur***, il pourra accéder à la page qui affiche tous les utilisateurs du site. Si il y a trop d'utilisateurs, il faudra intégrer le moyen d'afficher les utilisateurs par page sur ***plusieurs pages***.

  - Affichage d'une ***liste limitée de voyages*** sur une page spécifique (soit la page 'voyages' soit la page d'accueil du site) pour montrer quelques voyages possibles (les plus récents, les mieux notés, une sélection aléatoire, …) : cet affichage est possible même si l'on n'est pas connecté.
  - Affichage d'une ***liste de voyage*** à la suite d'une ***recherche*** par mots clés : cet affichage peut se faire même si l'on n'est pas connecté. Il n'est pas demandé de filtrer les résultats dans cette étape. Cette fonctionnalité est laissée pour la phase #3.

- Les affichages de listes de voyages précédemment cités affichent des informations limitées des voyages (titre, date, prix, nombre d'étapes, …) peu importe la forme (mosaïque, liste, …) pour avoir une ***vue synthétique*** de ces voyages.
  Si il y a trop de voyages à afficher, permettre de les afficher sur ***plusieurs pages***, et de pouvoir naviguer entre ces pages.
  Le fait de cliquer sur l'un de ces voyages permet de passer sur une page avec une vue détaillée.

- ***Vue détaillée d'un voyage***. Dans cette vue, on aura la liste des étapes avec la possibilité de modifier les options proposées pour chaque étape (hébergement, restauration, activités, mode de transport vers la prochaine étape, nombre de personnes par activité, …).
  Le but de cette page est de pouvoir permettre à un utilisateur ***connecté*** de personnaliser à sa guise le voyage proposé.
  Une fois la personnalisation terminé, un ***bouton*** doit permettre de passer vers une nouvelle page qui **récapitule** toute la configuration du voyage.
  La modification des options ne doit pas faire apparaître en direct la modification de prix, durée, … . Cette fonctionnalité est laissée pour la phase #3.

- ***Récapitulatif d'un voyage*** personnalisé. Sur cette page doit apparaître l'entièreté des données du voyage personnalisé.
  Par conséquent, les durées, dates, prix, nb de personnes, … doivent être mis à jour en fonction des différentes options choisies par l'utilisateur.
  Un bouton permet de revenir à la page de personnalisation pour continuer à modifier le voyage (cf. vue détaillée ci-dessus).
  Un autre bouton permet de confirmer la personnalisation du voyage et de passer à une page de paiement.

- ***Page de paiement***. Sur cette page ne doit apparaitre qu'un petit récapitulatif du voyage (avec à minima, le titre du voyage, les dates de début et fin, ainsi qu'un formulaire demandant de saisir des coordonnées bancaires :
  - numéro de carte à bancaire à 16 chiffres (4 x 4)
  - nom et prénom du propriétaire de la carte
  - mois et année d'expiration
  - valeur de contrôle à 3 chiffres

  Le script en destinataire de ce formulaire (et qui vous sera fourni) simulera la vérification des coordonnées bancaires.
- Si les coordonnées sont valides, alors le script de vérification va rediriger vers un script à vous, qui va enregistrer la transaction de paiement (coordonnées bancaires, identifiants de l'utilisateur, et configuration complète du voyage) de manière à pouvoir garder une traçabilité des commandes.

- Si les coordonnées bancaires sont non valides, le script de vérification redirigera vers un autre script à vous, qui affichera un message d'erreur et permettra de revenir sur la page de paiement et/ou sur la page récapitulative du voyage pour pouvoir retenter un autre paiement ou bien revenir modifier des options du voyage (car le prix est trop élevé et même si les coordonnées bancaires sont fausses, la vérification a échoué pour cause de manque d'argent sur le compte)

- Page de ***profil utilisateur*** avec une liste des voyages déjà payés
  La liste est une liste synthétique de voyages et le fait de cliquer sur l'un d'eux renvoie vers une page avec l'affichage détaillé de ce voyage mais sans avoir la possibilité de le modifier (pensez à réutiliser les pages déjà disponibles).

- Tous les fichiers HTML doivent être renommés en PHP sauf si il n'y a rien de dynamique pour ces pages.
- Tous les scripts PHP doivent être correctement rangés dans une arborescence propre (séparer les 'vues' des scripts de code purs, créer des bibliothèques de code PHP à inclure dans vos pages, …)

- Les fichiers contenant les données doivent être dans une arborescence distincte des scripts PHP. Le format des fichiers de données est laissé libre mais, à minima, un format CSV est préconisé, et le mieux reste peut être d'utiliser le format JSON. Du moment que vous arrivez à gérer vos données, aucun format n'est imposé.

- Pour les groupes qui souhaitent utiliser des bases de données, cela reste possible mais comme ce chapitre n'est pas abordé explicitement pendant le semestre, si des problèmes surviennent en lien avec l'utilisation de la base de données, il faudra assumer le fait que l'application Web ne fonctionne pas et aucune discussion n'aura lieu concernant ce point.
  Si une base de données est utilisée, il faudra fournir dans le document de conception le schéma des tables, le type de moteur utilisé (MySQL, PostGreSQL, …), et une démarche simple pour l'installer et la configurer avant de lancer votre site. Si des erreurs sur ce dernier point apparaissent cela pourrait compromettre l'évaluation.

- Vous devez mettre à jour votre rapport de projet et relivrer au format PDF la version phase #2. Il ne s'agit pas de créer un nouveau fichier, ni de remplacer le précédent : il faut ajouter de nouveaux paragraphes à la suite de ceux existants.
  Dans ce rapport apparaitra entre autre le format des données dans vos fichiers ou votre base de données. Les explications concernant le format choisi doivent être détaillées et doivent correspondre à ce qui est réellement codé. Une mauvaise description de votre conception sur ce point sera très punitif sur l'évaluation.

Phase #3

- Cette phase met l'accent sur la partie code du front-end et notamment l'utilisation du langage JavaScript.

- La première fonctionnalité est de proposer un **changement de charte graphique** (ex : mode clair/sombre, mode contrasté, mode pour malvoyants avec des tailles de polices plus grandes, …)
  Ce changement doit se faire à l'aide d'un bouton disponible sur l'interface. L'appui sur ce bouton doit générer le chargement du nouveau fichier CSS sans recharger la page.
  Cette charte graphique doit être **ajoutée** dans le document de projet, comme la toute première charte lors de la phase #1.
  Il faut pouvoir **sauvegarder** le choix de l'utilisateur concernant le mode d'affichage dans un **cookie**. A chaque chargement de page, il faut vérifier la présence de ce cookie et sa valeur : si il n'existe pas, ou que sa valeur est incohérente, le mode choisi est le mode par défaut.

- Pour chaque **formulaire** (inscription, connexion, …) il faudra **vérifier** le contenu des champs **côté client**, et afficher un message d'erreur le cas échéant, sans recharger la page, ni envoyer le formulaire côté serveur. La requête HTTP ne doit partir que si l'ensemble du formulaire est correctement rempli.
  Il faudra également pouvoir proposer de cacher/afficher les champs de mots de passe (icône 'œil' par exemple).
  Si vous avez des champs qui sont limités en taille (mots de passe, email, pseudos, …) il faudra afficher un compteur en temps réel du nombre de caractères utilisés.

- Sur la page d'affichage du **profil complet** d'un utilisateur, les champs doivent être grisés/non éditables par défaut. En cliquant sur un bouton individuel situé près de chaque champ, ce dernier devient modifiable. Un fois la modification effectuée, un bouton permettra de valider et remettre le champ en non-modifiable. un autre bouton permettra d'annuler la modification, de remettre la valeur initiale, et désactiver l'édition du champ. Tout ceci sans effectuer de requête HTTP, ni recharger la page.
  Si une modification a été validée sur au moins un champ, un bouton soumettre apparaitra pour effectuer la modification de profil comme avant (envoi du formulaire à un script PHP).

- Sur la page **administrateur**, avec la liste des utilisateurs, lors de la mise à jour d'une propriété d'un utilisateur (qui pour le moment n'est pas censée être fonctionnelle) il faudra simuler l'attente de cette modification. C'est à dire, griser/désactiver le bouton/slider/check-box pendant quelques secondes, puis une fois passé cette durée, réactiver le composant avec la nouvelle valeur. Le changement de valeur sera réellement fait lors de la phase #4.
  Si vous avez déjà une mise a jour des données de l'utilisateur qui est foncitonnelle, attendez la fin du timer pour envoyer la requête avec les données du formulaire.

- Sur la page de **recherche** de voyages, après une recherche, des éléments sont affichés : il faut pouvoir les **trier** en fonction de **plusieurs** critères (dates, prix, durées, nombre d'étapes, …) le tout sans recharger la page ni effectuer de requête HTTP supplémentaire.

- Sur la page qui affiche un voyage en détails, et sur laquelle on permet à l'utilisateur de modifier des options, étapes, nombre de personnes, …, chaque modification ayant un impact sur le prix doit générer une modification d'affichage du prix estimé. Cette modification doit se faire sans générer de requête HTTP, ni recharger la page.
  Le prix total qui sera affiché lors de la phase de récapitulatif avant le paiement doit (calculé côté serveur) doit être cohérent avec le prix dynamique calculé en JS à chaque modification.

- Tout le code JavaScript doit se trouver dans des fichiers séparés, chargés par les différentes pages de votre site.

- En plus de la partie JS de cette phase, il faudra créer une fonctionnalité de « panier ». Cette fonctionnalité ne peut être codée que côté serveur car pour accéder aux détails d'un voyage il faut être connecté.
  On peut partir du principe que le voyage est ajouté au panier à partir du moment où l'on commence à le modifier, ou bien quand on affiche un résumé avant la phase de paiement. C'est à vous de décider du fonctionnel fourni par votre site.

- Si il vous manquait des fonctionnalités pour les phases précédentes, profitez de cette phase pour les rajouter/corriger/améliorer, une fois que les fonctionnalités précédentes de la phase #3 seront implémentées.

- N'oubliez pas de mettre à jour votre rapport de projet et relivrer au format PDF la version phase #3.

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 dvra 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 évalué sera fourni peu de temps avant le démarrage de chaque phase. Tous les critères attendus à 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 faudra effectuer régulièrement des commits de code sur votre dépôt de code.
  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 de chaque phase, un commit devra être fait avec un intitulé particulier comme ***[phase#1] DEBUT*** , ***start phase2***, ou encore ***sauvegarde avant phase3***, …, 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).
- Chaque phase sera évaluée sur un commit bien particulier : vous veillerez donc à créer un commit en fin de phase avec un message explicite du type ***[phase#1] FIN***, ou ***rendu phase2***, ou ***phase3 terminée***, … , ce message permettant de repérer votre livraison sans **aucune** ambiguité.
- Si il n'y a pas de moyen simple pour les chargé(e)s de TD de repérer votre livraison parmi tous vos commits (soit il y a plusieurs rendus avec l'intitulé de phase dans le message, soit il n'y en a aucun, soit il est impossible de différencier les commits de début et de fin), alors dans ce cas le commit avec la **date la plus proche antérieure à l'échance** 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 estimez que l'investissement des 3 membres du trinome est identique, vous donnerez des valeurs ***33%, 33%, 33%***. Si au contraire vous estimez (tous) qu'un étudiant s'est investi plus que les autres (ou moins) 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 t/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 permettra 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 site 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