

GLO-2005: Projet de cours

Description

Pour ce projet, vous devrez réaliser une application web suivant une architecture à trois niveaux et utilisant les technologies vues en classe (HTML, CSS, JavaScript, Python, Flask, et MySQL). Ce projet est à réaliser en équipes de 3, et compte pour 30% de votre semestre. De plus, l'équipe produisant le meilleur projet sera considérée pour le prix Pierre Ardouin (voir plus bas).

Vous êtes libre des aspects créatifs de votre application, incluant le sujet, les données à gérer, et l'apparence du site. Des exemples d'applications possibles incluent :

- Un magasin en-ligne, incluant un catalogue de produits, des comptes pour les clients, et des opérations de magasinage. Des exemples réels incluent Amazon.ca et eBay.ca.
- Un répertoire d'information partagé, incluant des pages documentant chaque instance, des pages personnelles pour les contributeurs, et des opérations d'édition. Des exemples réels incluent IMDb et Wikipédia.
- Une communauté virtuelle, incluant des profils individuels pour les membres, un historique des activités, et des opérations de contacts interpersonnels. Des exemples réels incluent Facebook et Twitter.
- Un service en-ligne, incluant un répertoire d'information, des comptes utilisateurs, et des opérations de requêtes et de suivi. Des exemples réels incluent l'Université Laval et la Bibliothèque de Québec.

Ces projets peuvent être ludiques et utiliser des données fictives mais réalistes (voir plus bas). Alternativement, si un projet pertinent réel est disponible à travers une compagnie ou un groupe de recherche, vous pouvez l'utiliser. En cas de doutes sur la validité du projet que vous désirez faire, consultez le professeur afin d'éviter les mauvaises surprises.

Réalisation

La réalisation d'un tel système devra passer par la série d'étapes suivante :

1. Énonciation du problème et des exigences.

Décrivez quelle application vous avez choisi d'implémenter, et dans quel contexte elle doit opérer. Étant donné le contexte, il devra exister des exigences auxquelles le système devra se soumettre; listez-les. Par exemple, si vous choisissez de développer un catalogue de magasin en-ligne, le contexte sera de savoir si le magasin en question est une boutique spécialisée ou un grande-surface. Une exigence, dans le cas d'une boutique spécialisée, sera d'avoir des fiches très détaillées pour chaque produit

afin de satisfaire une clientèle plus informée. Dans le cas d'un magasin grande-surface, une exigence sera de permettre des fiches pour une grande variété de produits parfois très différents les uns des autres.

2. Spécifications du système et des responsabilités des trois niveaux.

Décrivez les fonctionnalités nécessaires pour votre système (par exemple : créer un compte client, naviguer dans le catalogue de produits, acheter et faire livrer des produits, mettre à jour le catalogue). Puis définissez les fonctions qui seront réalisées par chacun des trois niveaux, et les informations qui seront transférées d'un niveau à l'autre. Une attention particulière devra être portée à décider si chaque fonctionnalité doit être réalisée par du code JavaScript du côté client, du code Python dans le serveur d'application, ou des requêtes, contraintes et routines SQL dans la base de données.

3. Modélisation des données et de la BD.

Créez le modèle entité-relation du système. Celui-ci devra découler logiquement de votre application, de ses contraintes, et de ses fonctionnalités. Puis, créez le modèle relationnel du système.

4. Implémentation de la base de données et des requêtes, et optimisation.

Créez la base de données dans MySQL et peuplez-la de données (voir plus bas). Définissez les contraintes et les requêtes nécessaires pour les fonctionnalités de ce niveau. Vous devez obligatoirement utiliser des contraintes d'intégrité et des routines dans votre implémentation.

5. Implémentation de la base de données et des requêtes, et optimisation.

Définissez des index sur les données et normalisez les relations pour optimiser l'exécution des requêtes. Assurez-vous que votre système fonctionne de manière optimale.

6. Implémentation de la logique d'affaire.

Créez les fonctions Python et Flask pour implémenter les fonctionnalités de ce niveau. Afin d'assurer une bonne performance, ce niveau devrait valider les informations provenant et allant aux deux autres niveaux, et minimiser les communications inutiles entre les niveaux. N'oubliez pas de commenter votre code de manière appropriée.

7. Implémentation de l'interface utilisateur.

Désignez le site web pour votre application. L'apparence du site est libre à vous; cependant, cette apparence doit être cohérente d'une page à l'autre à l'intérieur du site, et donc utiliser des CSS. De plus, toutes les informations nécessaires à votre application étant donné ses exigences doivent se retrouver sur le site et être organisées de manière logique. Par exemple, pour un catalogue en-ligne, on devrait retrouver une seule page par produit affichant tous les détails du produit en commençant par le nom, la disponibilité, et le prix (les trois informations plus importantes pour les utilisateurs), puis en allant aux détails plus fins; par contre

l'apparence et l'organisation des informations dans la page (en liste, en onglets, en sous-titres, etc.) est libre à vous. Créez les fonctions JavaScript pour implémenter les fonctionnalités de ce niveau. N'oubliez pas de commenter votre code de manière appropriée.

8. Sécurité du système.

Assurez-vous que votre système soit sécurisé, et qu'il n'y ait pas de risques de perte ou de vol des données.

9. Tests du système.

Assurez-vous que le système fonctionne bien, que les trois niveaux sont bien intégrés et communiquent bien ensemble. Assurez-vous que les cas d'erreurs sont bien gérés (par exemple, que se passe-t-il si un utilisateur entre une chaîne de caractère dans un champ de la page web qui devrait normalement recevoir un nombre?).

10. Rédaction d'un rapport technique.

Le projet sera évalué premièrement par un rapport technique décrivant votre réalisation. Celui-ci est détaillé plus bas.

11. Gestion de l'équipe et division du travail.

Tout au cours du projet, vous devrez diviser les tâches entre les membres de l'équipe de manière équitable et efficace. Vous devrez organiser des rencontres régulières afin que chaque membre de l'équipe soit au courant du progrès global du projet. Une attention particulière devra être portée à vous assurer que les interactions entre les différentes parties du projet, développées par différentes personnes, soient bien planifiées et implémentées.

Données

Dans le cadre de ce projet, il est acceptable d'utiliser des données fictives pour peupler votre base de données. Un script simple peut facilement générer et emmagasiner dans la BD une bonne quantité de données avec des valeurs aléatoires. Les valeurs ont besoin d'être réalistes, sans pour autant nécessiter être réelles et exactes. Par exemple, pour un catalogue en-ligne, les prix des produits doivent couvrir une gamme de valeurs allant de quelques dollars à quelques milliers de dollars, mais que la valeur exacte d'une voiture affichée soit 35 000\$ ou 3.50\$ ne fait aucune différence. De même, les messages échangés entre des utilisateurs sur un réseau social doivent varier de quelques mots à quelques paragraphes de longueur, mais que le contenu des messages soit une vraie correspondance ou un *lorem ipsum* n'importe pas.

La cardinalité des relations devrait être d'au moins une centaine de tuples par relation, à moins que le contexte et les exigences de votre projet n'exigent le contraire pour une raison réaliste, et que d'autres aspects du projet compensent pour cela. Par exemple, pour un catalogue de maisons en ligne (comme DuProprio), il est raisonnable que la relation « acheter » contienne peu de tuples, étant donné que la majorité des utilisateurs achètent 0 ou 1 maison. Par contre, les relations « utilisateurs » et « maisons » devraient avoir une centaine de tuples chacune, et le

degré de la relation « maisons » (le nombre d'attributs pour décrire complètement les maisons) sera très élevée. Contre-exemple : de designer un réseau social pour ermites introvertis afin de justifier par le contexte qu'il n'y aura aucuns tuples dans la relation « messages » est une tentative de contourner les exigences du projet et sera fortement pénalisé.

Correction

Le rapport technique devra documenter clairement tout le travail que vous avez fait à chaque étape de la réalisation du projet, tant les aspects techniques du système que les réflexions ayant mené à sa création. Il y a un rapport à remettre par équipe, en utilisant le site web du cours. Ce rapport devrait être d'environ 10 pages. Les critères de correction seront comme suit :

1. L'énonciation du problème et de ses exigences (2 points)
2. Le modèle entité-relation du système (3 points)
3. Le modèle relationnel du système (3 points)
4. L'implémentation et les fonctionnalités du niveau serveur de BD (4 points)
5. L'indexation des données, normalisation des relations, et l'optimisation des requêtes (4 points)
6. L'implémentation et les fonctionnalités de la logique d'affaire (3 points)
7. L'implémentation et les fonctionnalités de l'interface utilisateur (3 points)
8. La sécurité du système (2 points)
9. L'organisation et la gestion de l'équipe, et division des tâches (1 points)
10. Une revue du code (5 points)
11. Pénalité pour la qualité et le français (-3 points)
12. Pénalité selon l'évaluation des pairs (-6 points)

Un aspect important du rapport est de justifier vos décisions de design du système. En effet, chaque décision devrait découler logiquement des conséquences des étapes précédentes. Par exemple, le modèle entité-relation découle de l'énonciation et de la spécification du problème, le modèle relationnel découle de la spécification et du modèle entité-relation, l'implémentation des requêtes découle de la spécification du problème et du modèle relationnel, et l'optimisation découle des requêtes et du problème. Pour un contre-exemple plus concret, si votre système est pour une université et que vous prévoyez que les requêtes les plus communes seront pour des cours à suivre, de déployer un index sur les cotes finales des étudiants serait une décision très étrange et difficile à justifier, et donc qui vous coûtera des points (même si l'index est créé correctement dans le langage SQL). Une bonne manière suggérée pour structurer votre présentation dans le rapport est donc de justifier un besoin ou une étape de design, d'illustrer dans une figure le code ou le modèle résultant, et d'expliquer dans un paragraphe de texte la logique et le fonctionnement du contenu de la figure.

La revue de code sera faite à partir du code de votre projet. Vous devez donc inclure un lien à votre dépôt Git dans votre rapport. Si votre code nécessite des instructions spéciales pour être exécuté, incluez-les avec votre soumission. Également, prévoyez une manière simple pour que le correcteur puisse obtenir et charger vos données, et exécuter votre code.

Notez également que jusqu'à 10% des points peuvent être enlevés en pénalité pour un rapport de mauvaise qualité. Ceci inclut particulièrement les fautes d'orthographe et de grammaire, les figures mal préparées (ou dessinées à la main), les rapports écrits à la main, les irrégularités de polices et tailles de caractère, et les textes incohérents.

En plus du rapport, chaque étudiant doit remettre une évaluation de ses coéquipiers à travers le site web du cours. Celle-ci doit évaluer le travail et la contribution à l'équipe des coéquipiers. Cette évaluation est fournie séparément du rapport, de manière confidentielle, en utilisant la soumission d'évaluations formatives sur le site web du cours. Pour chacun de ses coéquipiers, chaque étudiant doit évaluer sa contribution comme « satisfaisante », « insatisfaisante », ou « nulle », et donner une explication de quelques lignes pour ce choix. Un étudiant recevant la note de « insatisfaisante » de ses deux coéquipiers recevra une pénalité de 3 points (10% de la note), un étudiant recevant la note de « insuffisant » d'un coéquipier et « nulle » de l'autre recevra une pénalité de 4.5 points (15% de la note), et un étudiant recevant la note de « nulle » de ses deux coéquipiers recevra une pénalité de 6 points (20% de la note). Un étudiant ne soumettant pas l'évaluation de ses coéquipiers recevra également une pénalité de 3 points. Finalement, l'enseignant se réserve le droit de substituer son propre jugement à celui des pairs advenant que leurs évaluations ne semblent pas conformes aux faits.

Plagiat

Le plagiat est une offense académique sérieuse. Tout étudiant qui tente de soumettre un travail qui n'est pas le sien sera pénalisé. Ceci inclut de copier le travail ou rapport d'un autre étudiant du cours, d'un étudiant d'un autre cours ou d'une année précédente, ou un système trouvé ailleurs sur internet. Un étudiant coupable de plagiat recevra automatiquement la note de zéro pour le projet entier et s'exposera à d'autres sanctions telles que décidées par l'Université.

Ressources externes

Vous connaissez peut-être une ressource externe (une infrastructure de développement JavaScript ou un modèle standard CSS ou un serveur d'application C++, etc.) que vous aimeriez utiliser dans votre projet. Ceci est acceptable, étant donné certaines conditions. Premièrement, gardez en tête le règlement sur le plagiat : toute tentative de remettre un travail que vous n'avez pas fait comme si c'était le vôtre est une infraction académique majeure. En d'autres mots, votre rapport doit indiquer clairement quelle partie du projet vous avez réalisé et quelle partie a été empruntée d'une autre source, et la division entre les deux ne doit laisser aucune place à

l'ambiguïté. Deuxièmement, les points du projet sont donnés pour le travail que vous avez réalisé, et non pour l'intégration du travail de tiers. En d'autres mots, si vous utilisez une ressource externe pour réaliser un des objectifs évalués du projet sans contribution de votre part, vous aurez une note de zéro sur cet objectif, comme si vous ne l'aviez pas fait du tout. De même, aucuns points bonus ne sont donnés pour l'intégration de ressources externes au projet.

Vous pouvez également vouloir ajouter des fonctionnalités réaliste à votre site en appelant une API d'un service externe (ex. : compléter le site de votre boutique en ligne en appelant PayPal pour faire des paiements). Cette touche de réalisme n'est pas obligatoire, et n'est pas comptabilisée dans les critères de correction. Vous pouvez donc remplacer ces APIs par des étapes-bidons dans votre code pour les simuler (ex. : une fonction PayPal dans votre code qui accepte toutes les transactions sans contacter la vrai API). Par contre, l'utilisation des vraies API sera un critère considéré dans la sélection d'un projet gagnant du Prix Pierre Ardouin (ce qui n'influence en rien votre note finale).

Soumission du code

Pour la remise de ce projet, vous devez fournir une archives zip comprenant les éléments suivants :

- Un répertoire nommé app contenant:
- Le code source de votre application Flask,
- Un fichier dockerfile permettant de builder le conteneur de votre application Flask,
- Le répertoire .git associé à votre application Flask,
- Un répertoire nommé bd_init contenant le script sql permettant de créer votre base de données (avec les tables et les tuples),
- Un fichier docker-compose à la racine de l'archive permettant de builder et déployer votre projet.

Si votre projet contient d'autres dépendances (à un framework front-end javascript par exemple), ces dernières doivent être utilisées via un conteneur Docker et buildés lors du déploiement du projet.

Votre archive zip ne devrait pas contenir autre chose que du code que vous avez vous-même écrit.

Prix Pierre Ardouin

Depuis l'automne 2013, le Département d'informatique et de génie logiciel a mis en place un concours récompensant l'équipe qui aura produit le meilleur TP/projet dans le cadre d'un cours. Ces travaux de session ont l'envergure d'un mini-projet qui est admissible par rapport aux normes fixées par le Département. À la suite des évaluations des travaux, l'enseignant du cours détermine l'équipe gagnante; chaque membre de l'équipe gagnante reçoit alors une bourse de 50\$ ainsi qu'une attestation remises par le Département.

De plus, le Département d'informatique et de génie logiciel a mis en place une bourse Élite, appelée bourse « Pierre Ardouin », qui vise à récompenser le meilleur projet de session, tous cours confondus. Deux principaux critères guident le choix des évaluateurs dans l'identification du lauréat : l'excellence du travail (par rapport à ce qui est demandé dans l'énoncé) et l'aspect créativité/innovation. Il est actuellement prévu une bourse de 200\$ pour récompenser chaque membre de l'équipe « élite » gagnante (pour un maximum de 1000\$ pour toute l'équipe). Aussi, le Département veille à publier l'information sur un site Web dédié : <http://www.ift.ulaval.ca/vie-etudiante/prix-pierre-ardouin>.

À la deuxième moitié du mois de mai de chaque année universitaire, le Département organise une cérémonie pour honorer les finalistes et le lauréat du prix « Pierre Ardouin » des sessions d'automne et d'hiver, et leur remettre une attestation.

