

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 21% 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. Des projets réels en collaboration avec des OBNL locales sont disponibles à travers le service Accès Savoir de l'Université Laval. 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.

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 (gâchettes, fonctions, ou procédures) dans votre implémentation. De plus, les routines doivent faire quelque chose d'utile pour votre système, et ne peuvent pas simplement dupliquer d'autres contraintes d'intégrités ou encapsuler une seule requête ordinaire. 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; vos choix précis doivent donc découler de l'utilisation prévue de votre système par les utilisateurs. Par exemple, si votre système a une relation Utilisateurs avec le courriel comme clé primaire et que cette relation n'est utilisée que pour identifier les utilisateurs par leur courriel, il est logique et optimal de ne pas indexer d'autres attributs et de n'utiliser que l'index créé automatiquement sur la clé primaire. À l'inverse, si votre système a un catalogue de livres avec le ISBN comme clé primaire et que vous n'utilisez que cet index automatique, vous présumez que les utilisateurs ne vont chercher des livres que par ISBN et jamais par auteur, titre, genre, ou autre attribut, ce qui n'est pas réaliste et vous coutera des points.

5. 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. Ce niveau devrait également attraper des cas d'erreurs inattendus pour éviter l'échec du système (par exemple avec l'utilisation de `try-except`).

6. Implémentation de l'interface utilisateur.

Désignez le site web pour votre application. Ce site web doit permettre à l'utilisateur de demander et retrouver facilement les informations pertinentes à votre application. Elle devrait également valider les informations entrées par l'utilisateur pour éviter les erreurs de saisies ou les attaques délibérées. 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.

7. 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. En particulier, le chiffrement des mots de passe des utilisateurs (s'il y en a dans votre système) est obligatoire.

8. 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 ce passe-t-il si un utilisateur entre une chaîne de caractère dans un champ de la page web qui devrait normalement recevoir un nombre?).

9. Rédaction d'un rapport technique.

Le projet sera évalué premièrement par un rapport technique décrivant votre réalisation. Celui-ci est détaillé plus bas. Profitez-en pour trouver un nom intéressant à votre projet – si vous gagnez le Prix Pierre-Ardouin, ce sera plus intéressant de présenter votre projet par son nom que comme « le projet du cours de bases de données ».

10. 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 réaliste, 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.

Votre BD devrait avoir au moins 6 relations différentes, dont au moins 3 doivent avoir un degré de 5 attributs ou plus. 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 aucun tuple dans la relation « messages » est une tentative de contourner les exigences du projet et sera pénalisé.

Rapport et 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, excluant la page couverture et les annexes. Les critères de correction seront comme suit :

1. L'énonciation du problème et de ses exigences (1 points)
2. Le modèle entité-relation et relationnel du système (2 points)
3. Le modèle relationnel du système (1 points)
4. Les fonctionnalités du niveau serveur de BD : Création des relations (3 points)
5. Les fonctionnalités du niveau serveur de BD : Requêtes et routines (3 points)
6. Les fonctionnalités du niveau serveur de BD : Indexation et optimisation (1 points)
7. Les fonctionnalités du niveau serveur de BD : Normalisation des relations (1 points)
8. Les fonctionnalités de la logique d'affaire (2 points)
9. Les fonctionnalités de l'interface utilisateur (2 points)

10. La sécurité du système (1 points)

11. L'organisation et la gestion de l'équipe, et division des tâches (1 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.

Style et longueur du rapport

Le rapport doit être rédigé dans le même style que ce présent document (police Times New Roman 12pt, interligne 1.15, espace de 6pt entre les paragraphes, marges de 1 pouce) et avoir une limite de 10 pages. Cette limite s'applique au contenu technique, incluant les tableaux et figures. Cette limite exclue la page titre, table des matières, et les annexes. Cette limite sera strictement appliquée, et des points seront retirés pour les débordements.

Une annexe est un ajout au rapport, dont le contenu peut être utile en référence mais n'est pas strictement nécessaire pour la compréhension du rapport. En d'autres mots, votre rapport doit nécessairement présenter le contenu essentiel de votre projet, alors que le contenu offrant des détails supplémentaires intéressants mais non-essentiel et dont l'inclusion dans le rapport pourrait le rendre trop lourd, compliqué, ou long, peut aller en annexe. Par exemple, si vous avez un tableau de résultats détaillés de plusieurs pages, vous pouvez présenter un résumé ou des lignes importantes pour votre discussion dans le rapport et inclure le tableau complet en annexe. À l'inverse, une annexe n'est pas une manière de contourner la limite de pages en y déplaçant le matériel qui devrait être dans votre rapport. Par exemple, vous ne pouvez pas mettre l'ensemble de vos résultats en annexe et n'en présenter aucun dans votre rapport pour sauver de la place. Une bonne règle pour décider est de vous poser la question, « est-ce que mon rapport serait complet si j'éliminais complètement ce contenu? » Si oui, c'est du matériel pour l'annexe, et si non c'est du matériel qui va dans votre rapport.

Les explications et justifications sont à prioriser sur des extraits de code ou de pseudo-code. Ce contenu me donne une bien meilleure idée de ce que vous avez réalisé et pourquoi. Des

exemples de code ou de pseudo-code sont utiles pour illustrer une explication d'un algorithme complexe. Par contre le code doit être écrit et placé dans un cadre numéroté comme une figure, et limité aux lignes importantes pour l'explication. Un copier-coller de votre code en entier ne fait pas office d'explication et ne sera pas évalué. Une saisie d'écran du code est contre-indiquée: ça ne donne jamais un résultat visuellement plaisant et c'est souvent difficile à lire, et ce sera pénalisé.

Soumission du code

Pour la remise de ce projet, vous devez fournir une archives zip comprenant tout le code (SQL, Python, HTML, JavaScript, CSS) que vous avez écrit, et uniquement le code que vous avez écrit (excluant donc complètement les librairies externes). Ce code sera donc non-fonctionnel, mais permettra au besoin une évaluation plus approfondie de votre travail. N'oubliez pas le code SQL de création des relations et des requêtes! Le code sera utilisé en complément avec votre rapport pour évaluer l'implémentation des fonctionnalités de votre système.

Démo vidéo

Vous devez réaliser une vidéo démo de votre projet, dans laquelle vous présentez votre projet et démontrez l'utilisation de votre système. Cette vidéo doit être d'environ 15 minutes. Vous pouvez la téléverser sur le site de remise de MonPortail avec votre rapport, ou sur un site d'hébergement vidéo de votre choix en fournissant un lien avec votre rapport (n'oubliez pas d'ajuster les droits d'accès pour qu'on puisse la visionner!)

Pour l'aspect de présentation du projet, vous pouvez passer, point par point, à travers la liste d'items du rapport. Pour l'aspect utilisation du système, vous devez démontrer que toutes les fonctionnalités de votre système fonctionnent bien tant pour une utilisation normale qu'une utilisation anormale (i.e. erreurs d'entrées accidentelles ou délibérées).

Il existe une variété d'outils gratuits qui permettent d'enregistrer une saisie de votre écran durant l'utilisation de votre ordinateur avec un flux d'entrée audio et/ou vidéo. Vous pouvez utiliser celui de votre choix. Si vous désirez une suggestion, j'utilise personnellement l'outil OBS Studio.

Cette vidéo vaut 3 points dans la note du projet, en plus des points pour le rapport. Il n'est pas nécessaire que tous les membres de l'équipes soient entendus dans la vidéo.

Bonus et Pénalités

Je réserve le droit de retirer jusqu'à 20% des points en pénalité pour un rapport et/ou une vidéo de mauvaise qualité. Ceci inclut particulièrement les rapports présentant des fautes

d'orthographe et de grammaire, des figures mal préparées (ou dessinées à la main), des irrégularités de mise en page (polices, tailles de caractère, marges, etc.), et les textes incohérents, les rapports soumis en document Word. Pour la vidéo, on pénalisera un langage inapproprié ou un enregistrement de mauvaise qualité (incluant de vous filmer avec votre cellulaire tenu à la main). Dans les deux cas on pénalisera le non-respect des longueurs maximales.

À l'inverse, je peux donner des points bonus à ma discrétion pour récompenser des étudiants qui intègrent des fonctionnalités supplémentaires dans leur système (ex. : envoi de courriel de confirmation, utilisation de vrais services en ligne, importation de données de vraies BD organisationnelles, etc.). Ces fonctionnalités ne sont pas demandées dans le projet et ne sont pas nécessaires, mais l'effort supplémentaire pour ajouter une touche de réalisme au projet sera souligné.

Évaluation par les pairs

En plus du rapport d'équipe, chaque étudiant doit remettre une évaluation de ses coéquipiers à travers la section formative du site web du cours. Celle-ci doit évaluer le travail technique 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 « minimale », et donner une explication de quelques lignes pour ce choix. Comme règle du pouce, un coéquipier contribuant environ le tiers du travail de l'équipe de 3 a une contribution « satisfaisante », un coéquipier contribuant bien moins que le tiers du travail a une contribution « insatisfaisante », et un coéquipier qui traîne et ne travaille que le strict minimum lorsque ses coéquipiers le force a une contribution « minimale ». Un étudiant recevant la note de « insatisfaisante » de ses deux coéquipiers recevra une pénalité de 25% de la note, un étudiant recevant une note « insatisfaisante » et une note « minimale » recevra une pénalité de 34% de la note, et un étudiant recevant la note de « minimale » de ses deux coéquipiers recevra une pénalité de 50% de la note. 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.

Il est important de souligner que les trois niveaux de contribution supposent que l'étudiant a contribué au travail technique de l'équipe. Un étudiant qui ignore ses coéquipiers, disparaît de l'équipe et ne contribue rien au projet n'a pas une contribution « minimale » mais bien « nulle ». Si les membres de l'équipe jugent qu'un de leur partenaire est dans cette situation, ils doivent exclure son nom du rapport de projet remis, et indiquer clairement la raison dans leur évaluation des paires. L'étudiant en question aura une note de 0 au projet, reflétant le fait qu'il n'y a pas participé du tout. Soulignons que la participation au projet se doit d'être technique, et ne peut pas être limitée à la rédaction du rapport. L'évaluation du projet décrite plus haut se concentre sur les

aspects techniques du travail réalisé, et un étudiant qui n'a fait aucune contribution technique a une participation nulle et une note de 0, même s'il a contribué à la rédaction du rapport.

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, d'un étudiant du cours d'une année précédente (incluant soi-même en cas de reprise du cours), ou un travail 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.

Conseils

- Il est fortement conseillé que tous les co-équipiers travaillent sur tous les aspects du système. Ceci vous donnera la meilleure expérience d'apprentissage, et vous préparera au maximum aux autres évaluations (le questionnaire de requêtes et l'examen final).
- La description du rapport à soumettre inclut les points alloués à chaque élément du travail. Tenez-en compte pour planifier la quantité d'effort à mettre pour chaque élément du projet. Par exemple, il n'est pas normal de passer plus de temps à développer un aspect valant 1 point qu'un valant 3 points.
- La réalisation de ce projet peut facilement prendre une envergure démesurée et consommer une majorité de votre temps cette session. Ne perdez donc pas de vue que le projet vaut 21% de votre note du cours, et ne négligez pas les autres aspects du cours (ni vos autres cours). Fixez-vous des objectifs réalistes et réalisables avec un effort raisonnable. Une fois de plus, la liste de points alloués à chaque aspect du travail peut vous aider dans cette planification de votre projet.

Ressources externes

Vous connaissez peut-être une ressource externe (une plateforme 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.

Le code démo du chapitre du cours sur l'architecture trois-niveaux est disponible pour téléchargement à travers le site web du cours. Vous pouvez l'utiliser comme point de départ pour votre projet, au même titre qu'une autre ressource externe.

Il n'est pas permis d'utiliser un ORM ou un autre logiciel de création de BD ou de code SQL. Quoique ces logiciels sont utilisés et même recommandés en pratique, l'objectif du cours est l'apprentissage du SQL et de l'implémentation de bases de données, et l'utilisation d'un logiciel qui réalise automatiquement ceci pour vous va donc à l'encontre des objectifs du cours.

Vous pouvez vouloir ajouter des fonctionnalités réalistes 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 vraie API). Par contre, l'utilisation des vraies API peut valoir des points bonus, et sera un critère considéré dans la sélection d'un projet gagnant du Prix Pierre Ardouin (soulignons que ce prix n'influence en rien votre note finale).

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.

Pour ce cours, le projet sera choisi parmi ceux ayant répondu à toutes les exigences d'évaluation et ayant reçu une excellente note. Parmi ceux-ci le projet choisi sera celui qui se démarque en allant au-delà des exigences du cours, par exemple en implémentant des fonctionnalités sophistiquées d'interaction avec l'utilisateur, en appelant les APIs de vrais services en-ligne, ou en présentant un site web polis et professionnel. Ce sera un projet qui pourrait, avec un minimum de modifications, devenir un service web réel. Le rapport et la vidéo démo du projet gagnant seront diffusés sur le site du Département.