

Plan de cours

Programme informatique

TITRE DU COURS	PROGRAMMATION WEB – SERVEUR II
NUMÉRO DU COURS	420-W56-GG
PONDÉRATION	<p>2 - 4 - 3</p> <p>→ heures de travail personnel par semaine</p> <p>→ heures de laboratoire par semaine</p> <p>→ heures de cours théorique par semaine</p>
COMPÉTENCE	00SU
ENSEIGNANT	MOHAMED AIROUCHE (LOCAL 449)
TÉLÉPHONE	514-626-2666 (POSTE 5613)
COURRIEL	m.airouche@cgodin.qc.ca
HORAIRE	<p>LUNDI: 09H00 À 12H00 AU 444 (À DISTANCE)</p> <p>MERCREDI : 08H00 À 11H00 AU 444 (À DISTANCE)</p>
DISPONIBILITÉ	RÉFÉREZ-VOUS À OMNIVOX, SUR LE BABILLARD DU DÉPARTEMENT

Compétences à acquérir dans ce cours

Compétence Gérald-Godin	3. Collaborer à l'analyse et à la conception de produits logiciels.		
Compétence ministérielle	00SU	Effectuer le développement d'applications Web transactionnelles.	Atteinte partielle
Éléments de compétence	1. Analyser le projet de développement de l'application. 2. Préparer l'environnement de développement informatique. 3. Préparer la base de données. 5. Programmer la logique applicative du côté serveur. 6. Programmer la logique applicative du côté client.		

Schémas des préalables

Session I	Session II	Session III	Session IV	Session V	Session VI
			420-W46-GG 420-B44-GG	→ 420-W56-GG →	420-B68-GG 420-A66-GG 420-A63-GG 420-P63-GG

Brève description du cours

Ce quatrième cours de programmation Web approfondit les concepts de programmation Web du côté serveur. Les applications sont développées en utilisant la technologie *.NET* (*ASP.NET* et/ou *ASP.NET Core*) avec C# comme langage de programmation. Les applications s'exécutent sur un serveur IIS ou autre et exploitent le gestionnaire de bases de données *Microsoft SQL Server*.

Contribution du cours dans le programme

Ce cours prépare l'étudiant à la réalisation d'un projet d'envergure dans le cadre du cours *Projet commerce électronique* (420-B68-GG).

Objectif terminal

Comprendre et appliquer les concepts de la programmation Web modulaire du côté serveur en utilisant la technologie *.NET* (*ASP.NET* et/ou *ASP.NET Core*).

Apprentissages et éléments de contenus essentiels

Élément de compétence et Critères de performance	Objectif d'apprentissage / Contenu essentiel	Poids
<p>00SU Effectuer le développement d'applications Web transactionnelles.</p> <p>1. Analyser le projet de développement de l'application.</p> <ul style="list-style-type: none"> Analyse juste des documents de conception. Détermination correcte des tâches à effectuer. 	<ul style="list-style-type: none"> À partir d'un énoncé détaillé, être capable de concevoir et structurer une application basée sur une architecture modulaire afin de répondre aux exigences fonctionnelles et techniques (problèmes de performance, de maintenance, de sécurité et de persistance des données). Développer un prototype de l'application pour approbation par le client. 	10%
<p>2. Préparer l'environnement de développement informatique.</p> <ul style="list-style-type: none"> Installation correcte de la plateforme de développement Web et du système de gestion de base de données de développement. Installation correcte des logiciels et des bibliothèques. Configuration appropriée du système de gestion de versions. Importation correcte du code source. 	<ul style="list-style-type: none"> Installation de la version adéquate de <i>.NET SDK (Software Development Kit)</i> pour la création d'applications Web (<i>ASP.NET</i> et/ou <i>ASP.NET Core</i>). Utilisation de <i>Visual Studio</i> et/ou <i>Visual Studio Code</i>. Utilisation de <i>SQL management Studio</i>. Installation des <i>packages</i> via le gestionnaire <i>NuGet</i>. Installation d'un outil de débogage des services Web comme <i>Postman</i> ou <i>Fiddler</i>. Utilisation d'un gestionnaire de version local (ex. <i>Git</i>) et d'hébergeurs sur l'infonuagique (Ex. <i>GitHub</i>, <i>BitBucket</i>...) pour la gestion du code source. 	5%
<p>3. Préparer la base de données.</p> <ul style="list-style-type: none"> Création ou adaptation correctes de la base de données. Insertion correcte des données initiales ou des données de tests. Respect du modèle de données. 	<ul style="list-style-type: none"> Accès à la base de données <i>SQL-Server</i> avec l'utilisation appropriée de connecteur <i>ADO.NET</i>. Création des modèles et accès aux données via l'<i>ORM Entity Framework (Code first, Model first, Database First)</i>. Présentation des commandes permettant de synchroniser les entités avec la base de données (fonction de migration). 	15%
<p>5. Programmer la logique applicative du côté serveur.</p> <ul style="list-style-type: none"> Programmation ou intégration correctes de mécanismes d'authentification et d'autorisation. Programmation correcte des interactions entre l'interface Web et l'utilisatrice ou l'utilisateur. 	<ul style="list-style-type: none"> Les différentes approches client-serveur. Architecture modulaire d'une application Web. Les différentes logiques de programmation en <i>ASP.NET</i>. Structure d'une application utilisant l'architecture modulaire et multicouche avec un modèle <i>MVC (Modèle – Vue - Contrôleur)</i>. Routage, navigation et les filtres. Liaison de données (<i>Data binding</i>). Validation des entrées d'utilisateur du côté serveur. 	60%

<ul style="list-style-type: none"> • Choix approprié des clauses, des opérateurs, des commandes ou des paramètres dans les requêtes à la base de données. • Manipulation correcte des données de la base de données. • Utilisation appropriée des services d'échange de données. • Application correcte des techniques d'internationalisation. • Application rigoureuse des techniques de programmation sécurisée. 	<ul style="list-style-type: none"> • Utilisation de concept de couplage faible pour les objets et les classes de services pour faciliter l'isolation des parties d'une application à des fins de test ou de remplacement. • Utilisation de la technique de l'injection de dépendances. • Utilisation des intergiciels (<i>middlewares</i>). • Utilisation appropriée des services d'échange de données (<i>ADO.NET, DbContext, LINQ, Entity Framework</i>). • Utilisation appropriée des services <i>Web API</i> pour l'échange de données. • Utilisation d'<i>Identity Framework</i> pour la sécurité, l'authentification et l'autorisation. • Déploiement extensible(<i>scalable</i>) de projets sur serveur (<i>IIS, Azure</i> ou autre). 	
<p>6. Programmer la logique applicative du côté client.</p> <ul style="list-style-type: none"> • Manipulation adéquate des objets du modèle DOM. • Programmation appropriée d'appels asynchrones. 	<ul style="list-style-type: none"> • Séparer les logiques de développement d'une application en <i>ASP.NET</i> (<i>MVC</i> ou <i>Web API</i> du côté serveur et utilisation d'un <i>Framework</i> coté client comme <i>Blazor, React.js</i> ou <i>Angular.js</i>.) • Utilisation des <i>Frameworks</i> comme <i>JQuery</i> pour les appels asynchrones. • Ajout de fonctionnalités Web en temps réel aux applications. (La fonctionnalité Web en temps réel permettant au code côté serveur de transmettre instantanément du contenu au client.) 	10%

Déroulement du cours

- ⊙ Le cours est donné à raison de deux blocs de 3 heures par semaine.
- ⊙ C'est par l'intermédiaire de cours théoriques, laboratoires et exercices que les concepts importants sont présentés et par des projets et examens qu'ils sont évalués pour savoir s'ils sont bien assimilés.
- ⊙ Le cours se divise en deux grandes parties interreliées :
 1. Étude des concepts de la programmation Web modulaire de côté serveur en utilisant la technologie *.NET* (*ASP.NET* et/ou *ASP.NET Core*) avec le langage de programmation *C#*.
 2. Développement d'une application basée sur la technologie *.NET* (*ASP.NET* ou *ASP.NET Core*) dans le cadre d'un projet de fin de session.
 - Dans un premier temps, chaque équipe soumet un prototype de l'application à développer à partir du résultat de l'analyse élaborée dans le cadre du cours *Analyse et modélisation (420-A56-GG)*.
 - Dans un deuxième temps, l'application est développée en *C#* en utilisant la technologie *.NET*.

Évaluation des apprentissages

☉ L'évaluation formative prendra la forme d'exercices théoriques et pratiques qui ont pour objectif de valider la maîtrise des connaissances acquises à ce jour.

☉ L'évaluation sommative prendra la forme suivante :

- Exercices et laboratoires 7 %
- Deux projets de programmation (individuels) 8 %
- Un projet de fin de session (en équipe) 35 %
- 50 %

- Deux examens (25 % chacun) 50 %
- (Référez-vous à la P.D.E.A entreposée sur P pour la politique concernant les seuils.) 100 %

La précision et la qualité du français utilisé sont encouragées en tout temps. Cependant, pour les cours de programmation, la qualité du français fera partie des critères de correction principalement pour les interfaces utilisateurs (la partie visible par l'utilisateur d'un programme en exécution) des travaux effectués.

Détails sur l'évaluation finale	
Pondération	35 %
Contexte de réalisation	<ul style="list-style-type: none">• Projet de session étalé sur 4-5 semaines• En collaboration avec les personnes participant au projet.• À partir d'une analyse des besoins.• À partir de réseaux internes et mondiaux.• À partir d'une station de travail et des outils de développement appropriés.• Dans des environnements comprenant des logiciels de bases de données et des systèmes de gestion de bases de données.• À l'aide des outils de référence en ligne disponibles et de toute autre documentation.• À partir des exigences de l'entreprise et des standards de l'informatique.

Critères de performance	<ol style="list-style-type: none"> 1. Produire le prototype de présentation. <ul style="list-style-type: none"> • Exploitation correcte des possibilités des outils de développement. • Établissement approprié des caractéristiques de l'exploration. • Représentation graphique correcte des interfaces Web. • Détermination des caractéristiques globales des interfaces Web. • Représentation de l'interface de l'application conformément aux exigences de l'ergonomie et de l'esthétisme. • Présentation du prototype pour approbation aux personnes responsables. • Création appropriée de l'aide en ligne. 3. Préparer la base de données. <ul style="list-style-type: none"> • Création ou adaptation correctes de la base de données. • Insertion correcte des données initiales ou des données de tests. • Respect du modèle de données. 5. Programmer la logique applicative du côté serveur. <ul style="list-style-type: none"> • Utilisation d'une architecture modulaire pour faciliter l'isolation des parties d'une application à des fins de test ou de remplacement. • Codification des fonctions et des modules de l'application conformément aux exigences de l'entreprise. • Programmation correcte des interactions entre l'interface Web et l'utilisatrice ou l'utilisateur. • Manipulation correcte des données de la base de données. • Création et modification appropriées de la base de données. • Codification correcte des requêtes d'accès à la base de données. • Implémentation du module pour la sécurité de l'application Web. • Vérification rigoureuse du fonctionnement de chacun des programmes et des modules de l'application dans l'environnement de développement. • Validation et optimisation des performances de l'application. • Production complète et archivage de toute l'information relative aux programmes. 6. Programmer la logique applicative du côté client. <ul style="list-style-type: none"> • Programmation correcte de l'interface Web du côté client et des interactions entre l'interface Web et l'utilisatrice ou l'utilisateur.
--------------------------------	---

Exigences de travail

- ⊙ La pondération du cours est de 2-4-3. Ceci signifie qu'il y a en moyenne chaque semaine 2 heures de théorie, 4 heures de laboratoire et 3 heures de travail personnel.
- ⊙ Le temps de travail personnel sert à préparer, compléter et étudier les laboratoires et les exercices qui n'auraient pas été complétés à la fin de la période allouée. Il pourra également servir à la réalisation des différents projets. Ce temps peut varier d'un étudiant à l'autre et d'un moment de la session à l'autre.
- ⊙ La façon la plus efficace de progresser en programmation est sûrement de mettre un effort constant dans la réalisation du travail à effectuer. La qualité des apprentissages réalisés sera supérieure s'ils se déroulent sur une période prolongée.
- ⊙ Les étudiants qui auraient plus de facilité à compléter les laboratoires et qui n'auraient pas besoin de toutes les heures de laboratoires prévues à l'horaire seront invités à poursuivre autrement leur apprentissage durant ces périodes. Cela pourra prendre par exemple l'une des formes suivantes :
 - Travailler sur le projet de programmation courant;
 - Réaliser des activités d'enrichissement proposées par l'enseignant;
 - Agir comme personne-ressource auprès de collègues moins avancés, ce qui favorise l'intégration des apprentissages.
- ⊙ **La présence aux cours théoriques et laboratoires est obligatoire.**

Démarche pédagogique

- ⊙ Dans ce quatrième cours de programmation Web, les travaux sont individuels, sauf pour le projet final.
- ⊙ Comme indiqué précédemment, chaque période de cours est divisée en deux parties : une partie théorique où les concepts sont présentés alors que la partie pratique permet de les expérimenter.
- ⊙ Le laboratoire demeure formatif au sens où l'étudiant a droit à l'erreur et peut présenter une solution autant de fois qu'il le désire, jusqu'à ce qu'il ait atteint les objectifs. L'activité peut parfois prendre la forme de mini-projets s'étalant sur plus d'une période.
- ⊙ Les exercices comportent des éléments formatifs évalués par les étudiants.

Outils de travail

Aucun volume obligatoire. (La documentation sera accessible via des sites Web et LÉA et/ou OneDrive.)

Bibliographie

Au début de la session, l'étudiant sera invité à s'abonner aux services de la *BAnQ* (*Bibliothèque et Archives nationales du Québec*) pour avoir accès au *Safari Books Online* contenant de nombreux ouvrages sur *ASP.NET* et *C#*.

Pour s'inscrire aux services de la *BAnQ* :

<http://www.banq.qc.ca/accueil/>

Pour accéder directement à la librairie (noter qu'il y aura redirection automatique vers la page d'authentification si vous n'êtes pas connectés) :

<http://proquestcombo.safaribooksonline.com.res.banq.qc.ca/>

Quelques manuels disponibles via la librairie SAFARI BOOKS ONLINE

MacDonald, Matthew, Beginning ASP.NET 4 in C# 2010, Apress, 2010
ISBN-978-1-4302-2608-6

<http://proquestcombo.safaribooksonline.com.res.banq.qc.ca/9781430226086>

GRIFFITHS Ian, ADAMS Matthew, LIBERTY Jesse, Programming C# 4.0, O'Reilly Media, Inc, 2010, ISBN-978-0-596-15983-2

<http://proquestcombo.safaribooksonline.com.res.banq.qc.ca/9781449392192>

LIBERTY Jesse, MAHARRY Dan, HURWITZ Dan, Programming ASP.NET 3.5 Fourth Edition, O'Reilly Media, Inc., 2008, ISBN-978-0-596-52956-7

<http://proquestcombo.safaribooksonline.com.res.banq.qc.ca/9780596156657>

FOSTER, Robert, ASP.NET 3.5 AJAX Unleashed, Sams, 2008, ISBN-978-0-672-32973-9
<http://proquestcombo.safaribooksonline.com.res.banq.qc.ca/9780768680539>

Livre récent disponible à la bibliothèque de cégep G. G.

Andrés Talavera et Christophe Gigax, ASP.NET Core MVC - Maîtrisez ce Framework Web puissant, ouvert et multiplateforme, Éditions ENI, décembre 2016, ISBN-978-2-7460-9922-7. TK5105.8885A26G542019

Ce plan de cours peut être amendé à tout moment durant la session. Les étudiants seront avisés en classe des amendements apportés, qu'ils pourront reporter sur leur copie du plan de cours.

Extraits de la Politique institutionnelle d'évaluation des apprentissages (PIEA)	Mise en application en discipline Informatique dans le programme Techniques de l'informatique
<p>4.3 Évaluation</p> <p><i>Évaluation sommative</i></p> <p>L'ensemble des évaluations sommatives atteste progressivement l'atteinte de l'objectif terminal ou de la compétence visée.</p> <p>Dans chaque cours, la pondération des travaux, des examens ou des autres activités d'évaluation sommative est équilibrée en fonction de la valeur relative accordée aux principaux objets d'apprentissage.</p> <p>Par ailleurs, la correction d'une évaluation sommative en cours de session comporte des commentaires permettant à l'étudiant de prendre connaissance de ses erreurs et de ses lacunes. Un enseignant peut conserver des copies d'examen à condition qu'il les ait présentées aux étudiants et que ceux-ci puissent les consulter au besoin. Les copies corrigées doivent être conservées jusqu'à la date de fin de la procédure de révision de note prévue au calendrier scolaire.</p>	<p>La règle suivante vise à préserver l'équilibre de la contribution des divers types d'évaluations sommatives dans l'attestation de l'atteinte de l'objectif terminal des cours d'informatique.</p> <ol style="list-style-type: none"> 1. Un seuil minimum de 50 % du total alloué aux examens est requis pour réussir le cours. 2. À moins d'avis contraire dans le plan de cours, un seuil minimum de 50 % du total alloué aux travaux pratiques et aux laboratoires est également requis pour réussir le cours. 3. Un étudiant qui ne respecte pas cette règle ne peut obtenir plus de 55 % comme note finale du cours.
<p>4.5 Qualité du français</p> <p>∴ Pour tous les autres cours, chacune des fautes de français réduit la note finale, et cela, jusqu'à concurrence de 10 % de la note finale. Le barème de correction liée à la langue est fixé par les disciplines concernées.</p> <p>Avant de remettre à l'étudiant son évaluation formative ou sommative, l'enseignant souligne les fautes de français afin que l'étudiant puisse repérer ses faiblesses et améliorer la qualité de son français.</p> <p>Quand les fautes de français entravent trop la compréhension du texte, l'enseignant peut refuser le travail et demande alors à l'étudiant de corriger ce travail avant de le remettre à nouveau. L'étudiant est alors passible d'une pénalité de retard.</p>	

<p>4.6 Remise des travaux et présentation</p> <p>Tout retard dans la remise d'un travail doit être justifié auprès de l'enseignant. S'il n'est pas justifié et appuyé par une pièce justificative administrative appropriée comme un billet médical, il entraîne une pénalité de 10 % par jour franc de retard, et ce, incluant les journées de congé. Par ailleurs, aucun travail ne sera accepté après cinq jours ou après la remise des travaux corrigés par l'enseignant.</p> <p>L'étudiant doit présenter ses travaux écrits en conformité avec les normes de présentation prescrites par les différents programmes ou regroupements. Quand un travail est négligé, mal écrit ou quand il ne respecte pas ces normes de présentation, l'enseignant refuse de recevoir le travail et demande à l'étudiant de corriger son travail avant de le remettre à nouveau. L'étudiant est alors passible d'une pénalité de retard.</p>	<p>Lorsque pour un travail donné, le fait de remettre en retard irait à l'encontre des objectifs visés ou accorderait un avantage indu au retardataire, des consignes particulières peuvent être ajoutés dans l'énoncé du travail.</p>
<p>4.7 Présence aux cours</p> <p>Un étudiant peut se voir refuser l'accès au cours s'il perturbe le déroulement du cours, par exemple par un manque de ponctualité ou par un comportement inapproprié. Il conserve toutefois le droit de se présenter aux évaluations sommatives.</p> <p>L'enseignant a des exigences supplémentaires pour la présence aux cours dans les trois cas suivants :</p> <ol style="list-style-type: none"> 1. lorsque l'étudiant est sous contrat de réussite pour une session donnée; 2. lorsque l'étudiant doit apprendre la manipulation adéquate et sécuritaire du matériel ou de l'équipement relié à l'apprentissage dans les cours; 3. lorsque l'étudiant doit prendre part à des activités qui ont une incidence sur un travail d'équipe. 	<p>Au-delà de l'équivalent de 2 semaines d'absences, l'élève sera convoqué à un comité départemental afin d'évaluer la situation et prendre les mesures nécessaires pour favoriser la réussite.</p>
<p>4.8 Reprise d'une activité d'évaluation à la suite d'une absence</p> <p>La reprise d'une évaluation à la suite d'une absence n'est pas automatique. L'absence à une évaluation sommative en cours de session doit être justifiée auprès de l'enseignant au plus tard dans les cinq jours ouvrables qui suivent la date de l'activité d'évaluation.</p> <p>Dans le cas où l'étudiant justifie son absence par une pièce justificative administrative appropriée, comme un billet médical, l'enseignant décide si l'étudiant participe à une activité de reprise ou si la pondération des notes de l'étudiant est changée.</p> <p>Si l'absence n'est pas justifiée, l'étudiant obtient la note zéro « 0 » pour cette évaluation.</p> <p>En cas de désaccord avec la décision, l'étudiant peut utiliser la procédure de traitement des plaintes étudiantes du Collège.</p>	

4.9 Plagiat et manquement à l'honnêteté intellectuelle

Définition

Tout plagiat, toute tentative de plagiat, ou tout manquement à l'honnêteté intellectuelle, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation, constitue une infraction (...).

Sanction

- ∴ L'étudiant responsable de plagiat à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation sommative se voit attribuer par l'enseignant la note zéro « 0 » pour cet examen ou ce travail. Aucune reprise n'est possible dans ce cas.
- ∴ Par ailleurs, un formulaire de « déclaration de plagiat », signé par l'enseignant et le coordonnateur, est remis à l'API et déposé au dossier scolaire de l'étudiant pour la durée de ses études au Collège. Une copie de ce formulaire est envoyée à l'étudiant.
- ∴ Dans le cas d'une récidive, le cas est rapporté au Directeur des études qui rencontre l'étudiant pour l'aviser que ce dernier obtient la note zéro « 0 » pour le cours concerné.
- ∴ Dans le cas d'une évaluation à dimension formative, un avertissement est adressé à l'étudiant.

4.10 Révision de note

4.10.1 Révision en cours de session

- ∴ L'étudiant doit en faire la demande à son enseignant au plus tard dans les trois jours ouvrables suivant la remise de l'évaluation.
- ∴ L'enseignant révisé la copie dans les cinq (5) jours ouvrables suivant la demande.

En cas de désaccord avec la décision, l'étudiant utilise *la Procédure en cas de litige dans le cadre de la relation pédagogique entre étudiants et enseignants*.

4.10.2 Révision en fin de session - Objet de la révision

L'étudiant ne peut faire réviser, en fin de session, que l'épreuve terminale, le travail de fin de session ou le calcul de la note de session.