



GUARDIA

CYBERSECURITY SCHOOL



CAHIER DES CHARGES

Programmation web full stack



INTRODUCTION

Nom du module : Programmation web full stack

Nom d'UE : DevSecOps

Nombre de crédits ECTS : 5

Date : 18 nov au 6 déc

Classe : GCS1

MODALITÉS

Groupes de 3 ou 4 étudiants à former avant le début du projet

CONTEXTE ET DESCRIPTION DU PROJET

CONTEXTE

Toute formation consiste en un parcours pédagogique permettant d'atteindre un objectif professionnel. Ce parcours pédagogique donne lieu à l'acquisition d'un ensemble de compétences qui doivent être mises en valeur par l'étudiant à l'issue de sa formation afin de maximiser son employabilité. Destiné à de potentiels recruteurs, l'ensemble des compétences acquises, ou plus simplement « *portfolio*

» doit, à l'ère du numérique, être accessible en ligne. Ainsi, quoi de plus évident pour un développeur web *full stack* que de concevoir et développer lui-même son cyberfolio ?

DÉTAIL DU PROJET

Objectifs

Ce projet a pour principaux objectifs d'appréhender l'architecture client-serveur et de mettre en œuvre de manière simple les langages les plus utilisés dans le développement web *full stack* :

1. Côté client, ou « *frontend* » : HTML, CSS et JavaScript.
2. Côté serveur, ou « *backend* » : PHP, MySQL et Python (ou langage C).



OBJECTIFS PÉDAGOGIQUES ET PROFESSIONNELS DU PROJET

A l'issue de ce projet, les étudiants seront capables de (Veuillez trouver les verbes actifs et mesurables dans cette [infographie](#)) :

Savoirs

- Identifier les principaux langages du web.
- Définir l'architecture client-serveur.
- Distinguer les notions de *frontend* et *backend*.
- Définir le métier de développeur web *full stack*.
- Définir la notion de *server-side scripting* (langage de *script* côté serveur).
- Définir la notion de SGBDR (Système de Gestion de Base de Données Relationnelle).
- Définir ce qu'est une API (*Application Programming Interface*).
- Reconnaître les principales balises HTML.
- Reconnaître les principales propriétés CSS.
- Définir ce qu'est le DOM (*Document Object Model*).
- Distinguer les fonctions de hachage les plus courantes.
- Identifier le premier niveau de sécurisation d'une application web.

Savoir-faire / Compétences

- Interpréter un cahier des charges.
- Lire une documentation technique.
- Présenter ses compétences en tant que développeur web *full stack*.
- Configurer un environnement de développement local.
- Utiliser un éditeur de code.
- Créer un document HTML.
- Créer une feuille de style CSS associée à un document HTML.
- Interagir avec le DOM (*Document Object Model*) en JavaScript.
- Décliner une page web pour un support mobile (version *responsive*).
- Élaborer un algorithme simple.
- Programmer en JavaScript côté client (niveau débutant).
- Programmer en PHP côté serveur (niveau débutant).
- Programmer en Python ou en langage C (niveau débutant).
- Créer une base de données MySQL.
- Exploiter une base de données grâce au langage MySQL (requêtes d'interrogation).
- Exploiter une API grâce au langage JavaScript.
- Prendre en compte les principes de sécurité élémentaires.
- Présenter son travail.

Savoir-être

- Interagir avec son *manager* de façon pertinente.
- Travailler en équipe.
- Collaborer avec efficacité.
- Planifier le bon déroulement du projet.
- Respecter les délais.
- Gérer son temps.
- S'adapter aux imprévus.
- Faire preuve de curiosité.
- Faire preuve de persévérance.
- Faire preuve d'initiative.
- Travailler en autonomie.
- Adopter un point de vue global (prise de recul).
- Adopter un point de vue critique.
- Évaluer sa propre contribution.
- Reconnaître la contribution des autres.

RESSOURCES

1. **A savoir/à apprendre** (Ce travail pourra être évalué / noté en début de projet) :

- [Développeur web - fiche métier : missions, formation, salaire...](#)
- [Portfolio de Développeur Web : 25 Exemples Pour S'inspirer](#)
- [Introduction à HTML/CSS : réaliser des pages Web | Khan Academy](#)
- [Flexbox Froggy - Un jeu pour apprendre les flexbox CSS](#)
- [Grid Garden - Un jeu pour apprendre les grid CSS](#)
- [HTML/JS : faire des pages Web interactives | Khan Academy](#)
- [Apprendre le PHP : Chapitre 1, Présentation de PHP](#)
- [Introduction au SQL : interroger et gérer des données | Khan Academy](#)
- [Apprendre Python - ackInScience — Hello World](#)

2. **A lire/à consulter** :

- [Technologies web pour développeurs | MDN](#)
- [W3Schools](#)
- [Manuel PHP - Manual](#)
- [MySQL :: MySQL 8.4 Reference Manual](#)
- [Have I Been Pwned: API v3](#)
- [Python](#)

3. **A installer/pré-requis techniques** :

- [Visual Studio Code](#)

- [MAMP](#) (ou équivalent : WampServer, XAMPP, ou UwAmp)
- [Python](#)

TRAVAIL PRÉPARATOIRE

1. Relire le support de cours du module fondamental précédent « *Introduction à la programmation & DevSecOps* ».
2. S'assurer de la disponibilité et du bon fonctionnement des logiciels nécessaires (environnement de travail).
3. Prendre en main les langages HTML, CSS, et JavaScript (Khan Academy).



ROADMAP

DESCRIPTION

Veillez trouver ci-dessous une description des livrables attendus ainsi que les dates d'échéance associées. Il est essentiel de respecter les échéances suivantes pour assurer une progression harmonieuse et structurée du projet. Chaque livrable représente une étape importante dans le processus de réalisation et permet d'évaluer l'avancement du travail. Les dates limites fixées doivent être rigoureusement respectées afin de garantir une évaluation équitable et de permettre un feedback constructif en temps opportun.

Jalon	Livrables attendus	Date limite	Moyens / formats
1	Fiche de type « persona » Langages HTML et CSS	Semaine 1 Jour 2	Fiche de type « persona » au format PDF Screenshots au format JPEG de la progression en HTML et CSS
2	Page d'accueil statique dans sa version <i>desktop</i> et <i>mobile</i>	Semaine 1 Jour 5	Présentation collective orale Codes sources HTML et CSS
3	Environnement de travail	Semaine 2 Jour 2	Screenshots au format JPEG de l'environnement de travail fonctionnel

4	Page d'accueil dynamique dans sa version <i>desktop</i> et mobile	Semaine 2 Jour 4	Présentation collective orale Codes sources HTML, CSS, et PHP
5	Page « <i>compétences</i> » dynamique dans sa version <i>desktop</i> et mobile	Semaine 3 Jour 3	Présentation collective orale Screenshot au format JPEG de la base de données MySQL Codes sources HTML, CSS, et PHP
6	Projet final Fonctionnalités complémentaires	Semaine 3 Jour 5	Présentation collective orale Démonstration de l'exploitation de l'API « <i>Have I Been Pwned</i> » Démonstration de la détection de fonction de hachage Codes sources JavaScript et Python (ou langage C)
7	Bilan du projet	Semaine 3 Jour 5	Présentation collective orale avec support de présentation simple (Google Slides ou équivalent) Difficultés rencontrées Bilan des acquis Perspectives et améliorations du cyberfolio



EVALUATION

L'évaluation est conçue pour être holistique, prenant en compte non seulement le produit final, mais aussi le processus, les compétences acquises et les attitudes démontrées tout au long du projet. Les compétences sont classées en trois catégories principales : "savoirs" (connaissances théoriques), "savoir-faire" (compétences pratiques), et "savoir-être" (compétences interpersonnelles et attitudes).

SYSTÈME DE NOTATION

Savoirs (Connaissances)

- **Compréhension théorique** : Évaluée soit en amont du projet pendant la semaine théorie soit lors de la soutenance et restitution du projet. Cela permet de mesurer la compréhension des concepts fondamentaux et des connaissances liées au projet des étudiants.

Savoir-faire (Compétences)

- **Compétences techniques et application** : Évaluées à travers la soumission finale du projet. Cela inclut la qualité, la fonctionnalité, et la précision technique du travail produit.
- **Gestion de projet** : Évaluée en fonction de l'organisation, du respect des échéances, et de l'utilisation efficace des ressources. Cela peut être évalué à travers la documentation du projet et les journaux de processus.

Savoir-être (Attitudes/Compétences interpersonnelles)

- **Travail d'équipe et collaboration** : Évalués à travers des évaluations par les pairs et les membres du groupe. Les critères incluent la communication, la coopération, et la contribution aux tâches du groupe.
- **Autonomie et initiative** : Évaluées en fonction des contributions individuelles, de la capacité à travailler de manière autonome, et de la résolution proactive des problèmes.



EVALUATION

Dans la grille d'évaluation qui suit, chaque groupe de compétences doit être évalué sur une échelle de valeurs établie de 0 à 3 :

- 0. *Non acquis*
- 1. *Partiellement acquis*
- 2. *Acquis*
- 3. *Acquis et dépassé*

#	Compétences	Niveau d'acquisition (de 0 à 3)
1 - Savoirs		
01	Identifier les principaux langages du web Définir l'architecture client-serveur Distinguer les notions de <i>frontend</i> et <i>backend</i> Définir le métier de développeur web <i>full stack</i>	
02	Définir la notion de <i>server-side scripting</i> Définir la notion de SGBDR Définir ce qu'est une API	
03	Reconnaître les principales balises HTML Reconnaître les principales propriétés CSS Définir ce qu'est le DOM	
04	Distinguer les fonctions de hachage les plus courantes Identifier le premier niveau de sécurisation d'une application web	
2 - Savoir-faire		
05	Interpréter un cahier des charges Lire une documentation technique	
06	Présenter ses compétences en tant que développeur web <i>full stack</i>	
07	Configurer un environnement de développement local	

