

RÉFÉRENTIEL D'ACTIVITÉS	RÉFÉRENTIEL DE COMPÉTENCES	RÉFÉRENTIEL D'ÉVALUATION	
		MODALITÉS D'ÉVALUATION	CRITÈRES D'ÉVALUATION
BLOC 1 : Développement Front End de sites et applications web.			
<p>Activité 1 : Traduction de la maquette en code interprétables par les différents navigateurs.</p> <ul style="list-style-type: none"> - Intégration Web - responsive design - Normes et accessibilité - Standardisation du code - Référencement naturel 	<p>C1.a : Utiliser les langages de balisage html et css (avec et sans framework) pour intégrer les maquettes graphiques en utilisant un éditeur de code.</p> <p>C1.b : Produire l'encodage en tenant compte des différentes contraintes imposées par les différentes résolutions d'écrans et les navigateurs afin d'obtenir un affichage optimisé pour smartphones, tablettes et desktop.</p>	<p>Mise en situation professionnelle : projet :</p> <p>Le candidat réalise l'intégration complète du site internet ou de l'application en utilisant les dernières versions des langages de balisage HTML et CSS. Il développe en utilisant le JavaScript les animations et différentes interactions utilisateur demandées dans le cahier des charges.</p> <p>Il réalise le déploiement du site internet ou de l'application sur le serveur.</p> <p>Le candidat présente le résultat de son travail, le code source et argumente son travail effectué concernant le référencement naturel et le respect des normes d'accessibilité (Critères RGAA) auprès des jurys de professionnels.</p>	<p>Cr 1.a.1 : L'intégration est conforme à la maquette</p> <p>Cr 1.a.2 : Le code respecte les normes W3C et les normes d'accessibilité.</p> <p>Cr 1.a.3 Le code passe avec succès les tests du validateur.</p> <p>Cr 1.a.4 : Le code est commenté et correctement indenté</p> <p>Cr 1 a 5 Les balises sémantiques sont utilisées à bon escient.</p> <p>Cr 1.b.1 : Le codage de l'application s'adapte correctement aux différentes résolutions d'écran</p> <p>Cr 1.b.2: Les propriétés utilisées sont compatibles avec les différents navigateurs</p> <p>Cr 1.b.3 : En cas d'incompatibilité du navigateur d'une propriété, le</p>

	<p>C1.c : <i>Considérer la diversité des publics, notamment en situation de handicap, afin d'améliorer l'expérience utilisateur, en mobilisant les différentes normes d'accessibilité. (Ex: RGAA)</i></p> <p>C1.d : Travailler sur une logique d'intégration réutilisable, organisée et synthétique, à des fins de lisibilité et de performances, en utilisant un système de "classes" génériques et fonctionnelles.</p>	<p>Éléments fournis :</p> <ul style="list-style-type: none"> - Les maquettes à intégrer - Le cahier des charges - Les éléments graphiques non optimisés à intégrer - un espace sur le serveur pour le déploiement. <p>Éléments attendus :</p> <ul style="list-style-type: none"> - Déploiement complet et fonctionnel du site internet ou de l'application sur le serveur. 	<p>candidat apporte une correction ou utilise une alternative, en s'appuyant sur la documentation.</p> <p>Cr 1.c.1: les attributs des éléments visuels sont correctement renseignés pour les logiciels de lecture d'écran.</p> <p>Cr 1.c.2 : Une police spécifique pour les personnes dyslexiques est prévue et intégrée.(Opendys)</p> <p>Cr 1.c.3 : Les informations importantes ne sont pas uniquement transmises par un code couleur mais sont textuellement exprimées.</p> <p>Cr 1.c.4 : L'utilisateur peut naviguer, accéder aux fonctionnalités et au contenu en utilisant le clavier.</p> <p>Cr 1.d.1: Le nommage des classes CSS est pertinent et propose une approche flexible,réutilisable.</p> <p>Cr 1.d.2: Le code css est organisé et commenté,</p> <p>Cr 1.d.3 Les classes sont regroupées par thématiques</p>
--	--	---	--

	<p>C1.e : Travailler le référencement naturel d'un site internet tout au long de sa construction pour optimiser son positionnement en appliquant les recommandations pour les différents moteurs de recherche.(hiérarchisation des contenus, sémantique, poids des pages...)</p>		<p>Cr 1.d.4 : Le code css produit est synthétique et ne présente pas de répétitions.</p> <p>Cr 1.e.1 : Les textes sont hiérarchisés et correctement titrés.</p> <p>Cr 1.e.2 Les expressions clés sont mises en exergue.</p> <p>Cr 1.e.3 Le balisage d'enrichissement de contenu est compris via le schema.org.</p> <p>Cr 1.e.4 La sémantique des balises est respectée (article, aside, nav)</p> <p>Cr 1.e.5 : Les balises meta sont uniques sur chaque page et contiennent un nombre de caractères optimisé.</p> <p>Cr 1.e.6 Les pages canoniques sont renseignées.</p> <p>Cr 1.e.7 Les attributs alternatifs des images sont présents ainsi que les titres des liens.</p> <p>Cr 1.e.8 : Les temps de chargement des pages sont</p>
--	--	--	--

			<p>optimisés (poids des images, utilisation des sprites ...).</p> <p>Cr 1.e.9 Le favicon est intégré.</p> <p>Cr 1.e.10 : la navigation entre les différentes pages du site est implémentée.</p> <p>Cr 1 .e.11 Les ancrs sont utilisées pour la navigation au sein d'une même page.</p>
<p>Activité 2 : Développement de fonctionnalités front end (navigateur)</p> <ul style="list-style-type: none"> - Interactions et animations JavaScript - Validation de données - Fonctionnalités asynchrones - Implémentation de librairies 	<p>C 2.a : Enrichir l'interface de l'application ou du site internet en utilisant le langage JavaScript afin d'apporter à la partie front end de l'interactivité et des animations.</p>		<p>Cr 2.a.1 : Les syntaxes modernes (ES5, ES6 et supérieures) et les fonctions natives du langage sont acquises.</p> <p>Cr 2.a.2 : La manipulation des éléments du document (DOM) en termes de contenu comme de style est maîtrisée.</p> <p>Cr 2.a.3 : Les animations JavaScript développées permettent une meilleure expérience utilisateur.</p> <p>Cr 2.a.4 Les animations sont fonctionnelles et leurs comportements sont gérés sur les différents navigateurs.</p>

	<p>C 2.b : Valider les saisies utilisateurs dans les formulaires afin de vérifier l'échange de données entre l'utilisateur et le serveur à l'aide de méthodes de validation adaptées. (ex : expressions régulières)</p> <p>C 2.c : Développer des fonctionnalités mettant en jeu des requêtes asynchrones avec le serveur afin d'apporter de la fluidité et de l'interactivité sans rechargement de page en utilisant des API.</p>		<p>Cr 2.a.5 : Le code est développé en utilisant la programmation procédurale, fonctionnelle ou orientée objet, et la programmation événementielle.</p> <p>Cr 2.b.1 : Les données saisies par les utilisateurs dans les espaces interactifs sont contrôlées pendant la saisie en temps réel.</p> <p>Cr 2.b.2 : Les méthodes de contrôle mises en œuvre sont cohérentes en fonction de la nature des données à traiter.</p> <p>Cr 2.b.3 : L'envoi des informations au serveur n'est effectif que lorsque les données correspondent au format attendu. Le cas échéant, des messages préviennent l'utilisateur des erreurs de saisie à corriger.</p> <p>Cr 2.c.1 : Les développements des requêtes asynchrones sont fonctionnelles et correctement mises en œuvre.</p> <p>Cr 2.c.2 : Les requêtes HTTP asynchrones n'exposent pas de données sensibles ou personnelles.</p>
--	--	--	--

	<p>C 2.d : Optimiser les temps de développement en utilisant des ressources externes (bibliothèques JavaScript) pour résoudre des problématiques de développement complexes.</p>		<p>2.c.3 : Les réponses renvoyées par le serveur sont traitées et utilisées.</p> <p>Cr 2.c.4 Dans le cas d'un renvoi d'erreurs, celles-ci sont traitées de manière à ne pas interrompre l'exécution du script.</p> <p>Cr 2.d.1 : Les bibliothèques répondent à une problématique spécifique.</p> <p>Cr 2.d.2 : La bibliothèque est correctement implémentée d'après les recommandations d'utilisation de sa documentation.</p> <p>Cr 2.d.3 : Le candidat peut clairement expliquer le fonctionnement global de la bibliothèque et son utilisation.</p>
Bloc 2 : Développement back end d'applications Web			
<p>Activité 3 : Data : analyse, modélisation, et traitement des données.</p> <p>- Modélisation des données</p>	<p><i>C 3.a : Synthétiser les données utiles à l'application en analysant le cahier des charges afin de formaliser le modèle de données</i></p>	<p>Mise en situation professionnelle : projet :</p> <p>Le candidat conceptualise et développe from scratch (Depuis</p>	<p>Cr 3.a.4 : Les données nécessaires à l'application sont correctement identifiées</p> <p>Cr 3.a.2 : Les données sont retranscrites sur un schéma</p>

<ul style="list-style-type: none"> - Construction de bases de données - Exploitation des bases de données - Cadre légal et protection des données 	<p>C 3.b : Construire la base de données à l'aide d'un outil d'administration de base de données, afin de permettre la bonne circulation des données nécessaires au fonctionnement de l'application.</p> <p>C 3.c : Interroger la base de données par l'intermédiaire d'un langage de requêtes (SQL) pour permettre la manipulation et l'exploitation des données par l'application.</p>	<p>une page blanche sans code préconstruit) l'application web spécifique demandée et sa base de données, à l'aide d'un langage de programmation serveur.</p> <p>L'application est développée en paradigme objet, incluant l'héritage et en utilisant une architecture MVC.</p> <p>L'application est déployée sur le serveur mis à disposition.</p> <p>Le candidat présente son travail aux jurys. Il argumente son modèle de données et ses schémas conceptuels. Le candidat doit être en mesure de modifier son code en direct selon les demandes imprévues des membres du jury.</p> <p>Eléments fournis :</p> <ul style="list-style-type: none"> - cahier des fonctionnalités - un accès au serveur - un accès à une base de données 	<p>décrivant les différentes tables et les relations entre elles.</p> <p>Cr 3.a.3 : Le candidat exploite dans son modèle de données des informations externes provenant d'une api.</p> <p>Cr 3.b.1 : Le nommage des tables et des champs est cohérent avec la typologie des données.</p> <p>Cr 3.b.2: le type des champs est choisi en adéquation avec la nature des données (varchar, boolean, integer...)</p> <p>Cr 3.b.3 : La mise en relation des tables est correctement effectuée.</p> <p>Cr 3.c.1 : Le candidat effectue les principales opérations de manipulation des données (lister, ajouter, modifier, supprimer)</p> <p>Cr 3.c.2 : Le candidat affine ses requêtes en utilisant des systèmes de tri et de filtres</p> <p>Cr 3.c.3 : Les requêtes sont optimisées par l'utilisation de clés étrangères et de liaisons de</p>
--	--	---	--

	<p><i>C 3.d Respecter le cadre légal en utilisant les normes imposées par le règlement général sur la protection des données (RGPD) afin de garantir l'intégrité des utilisateurs et la protection des données</i></p>	<p>Éléments demandés :</p> <ul style="list-style-type: none"> - Les schémas conceptuels et physiques du modèle de données - Les schémas fonctionnels de l'application - La base de donnée de l'application - L'application fonctionnelle déployée sur le serveur 	<p>tables.</p> <p>Cr 3.d.1 : le candidat a identifié, avec le client, les données sensibles et réglementées qui doivent bénéficier d'un traitement spécifique.</p> <p>Cr 3.d.2 : L'application informe l'utilisateur du stockage, de l'utilisation et du cadre de partage de ses données personnelles.</p> <p>Cr 3.d.3 : L'utilisateur dispose d'un droit de consultation, modification et de suppression de ses données personnelles.</p> <p>Cr 3.d.4 : Les données sensibles sont protégées.</p>
--	--	--	--

<p>Activité 4 : Développement de fonctionnalités back end (serveur)</p> <ul style="list-style-type: none"> - Conceptualisation d'applications - Programmation côté serveur - Programmation orientée objet - Architecture M-V-C - Sécurité - Travail en équipe et versionning 	<p>C 4.a : <i>Conceptualiser l'application, formaliser son schéma fonctionnel, à partir du cahier des charges fourni et des échanges avec le client, afin d'optimiser la charge serveur et les temps de réponse.</i></p> <p>C 4.b : Développer une application en utilisant un langage de programmation adapté afin d'en construire l'architecture et les fonctionnalités côté serveur.</p>		<p>Cr 4.a. 1 : Le candidat a posé les bonnes questions au client dans sa démarche de compréhension du fonctionnement de l'application à développer.</p> <p>Cr 4.a.2 Le candidat est force de proposition lors de ses échanges.</p> <p>Cr 4.a.3: Toutes les fonctionnalités nécessaires au bon fonctionnement de l'application sont correctement listées et détaillées.</p> <p>Cr 4.a.4 : Le schéma fonctionnel décrit en détail l'enchaînement des vues en fonction des différentes actions et interactions.</p> <p>Cr 4.b.1 : La syntaxe et les fonctions natives du langage sont acquises.</p> <p>Cr 4.b.2 Le code est indenté, les commentaires aident à la compréhension du code.</p> <p>Cr 4.b.3 : Les dossiers et fichiers du projet sont organisés.</p> <p>Cr 4.b.4 Les conventions de nommage sont respectées pour</p>
---	---	--	--

	<p>C 4.c : Développer une application en utilisant la programmation orientée objet et les héritages afin de produire un code réutilisable et structuré</p> <p>C 4.d : Développer une application à l'aide d'une architecture Modèle-Vue-Contrôleur (MVC) afin d'assurer un code facile à maintenir et évolutif.</p> <p>C 4.e Identifier un utilisateur et délimiter ses champs d'action dans le but de sécuriser</p>		<p>l'ensemble du code.</p> <p>Cr 4.b.5 : Les limites du code sont connues,</p> <p>Cr 4.b.6 Les erreurs de codage sont traitées.</p> <p>Cr 4.c.1 : La portée des attributs et des méthodes est cohérente.</p> <p>Cr 4.c.2 : Le code implémente des classes génériques et l'héritage est correctement mis en place.</p> <p>Cr 4.c.3 : Les classes sont implémentées en utilisant les namespaces et chargées par l'intermédiaire d'un autoloader, a défaut elles sont chargées manuellement dans un fichier de configuration.</p> <p>Cr 4.d.1 : Le modèle gère les interactions avec la base de données.</p> <p>Cr 4.d.2 : Les contrôleurs implémentent la logique et préparent les variables nécessaires au rendu de la "vue".</p> <p>Cr 4.d.3 : La "vue" reçoit et</p>
--	--	--	---

	<p>l'application par l'attribution de rôles spécifiques.</p> <p>C 4.f Travailler en équipe en utilisant des outils de collaboration et de gestion des versions afin de construire une application efficacement au sein d'une équipe de développeurs en entreprise.</p>		<p>permet l'affichage des données transmises par le controller et remplit son rôle principal d'affichage.</p> <p>Cr 4.e.1 : Le programme protège l'intégrité des données en empêchant toute injection d'éléments pouvant les compromettre.</p> <p>Cr 4.e.2 : Un utilisateur s'authentifie par l'intermédiaire d'un identifiant unique et d'un mot de passe L'utilisation d'un système de session, de token, ou équivalent permet d'identifier l'utilisateur connecté</p> <p>Cr 4.e.3 : L'implémentation dans le programme de différents rôles permet une délimitation des actions possibles et permissions pour chaque type d'utilisateur. (administrateur, auteur ...)</p> <p>Cr 4.f.1 : <i>Le candidat mobilise et transmet son savoir, son savoir-faire et ses méthodes. Il participe activement à la collaboration</i></p> <p>Cr 4.f.2 : L'utilisation de l'outil de</p>
--	--	--	---

	<p>C 4.g : Préparer l' application pour la livraison en s'assurant de sa conformité à la demande du client et son bon fonctionnement.</p>		<p>travail collaboratif est maîtrisée. (Ex:Gitlab)</p> <p><i>Cr 4.f.3 : le candidat sait auto évaluer et mesurer la compatibilité de son code avant de le soumettre comme contribution au projet.</i></p> <p><i>Cr 4.f.4 Le candidat peut clairement rendre compte de sa participation individuelle au travail collectif .</i></p> <p>Cr 4.g.1 : Le candidat s'assure de la conformité des fonctionnalités attendues par le cahier des charges et celles déployées.</p> <p>Cr 4.g.2 : Des tests unitaires sont réalisés et validés</p> <p>Cr 4.g.3 : L'application mise en ligne est exempte de bugs et fonctionnelle.</p> <p>Cr 4.g.4 : L'application est testée en production et ne montre pas d'erreurs ou d'effets de bords pouvant nuire à son utilisation.</p>
--	---	--	--

Bloc 3 : Développement avancé : Comprendre et utiliser une architecture de type framework (option)			
<p>Activité 5 : Développement d'applications à l'aide de framework (Framework front-end ou framework back-end)</p> <ul style="list-style-type: none"> - Appropriation d'un nouveau cadre de travail - Configuration d'un framework et installation de dépendances - Développement et mise en production 	<p>C 5.a : <i>Confronter ses connaissances en développement avec la documentation d'un framework dans le but de s'approprier l'architecture et les fonctionnalités d'un nouveau cadre de travail.</i></p> <p>C 5.b : Configurer le framework en utilisant un gestionnaire de dépendances afin de préparer l'environnement de développement de l'application.</p>	<p>Mise en situation professionnelle : projet:</p> <p>Le candidat conceptualise une application dotée de fonctionnalités avancées à partir d'un cahier des charges et un cahier des fonctionnalités fournis.</p> <p>Il construit son environnement de développement en installant un framework et les dépendances inhérentes aux fonctionnalités demandées.</p> <p>Le candidat développe l'intégralité de l'application avancée demandée dans l'environnement du framework</p> <p>Lors de la soutenance auprès du jury de professionnels, le</p>	<p>Cr 5.a.1 : Le candidat possède des bases solides dans les langages et technologies utilisés par le framework.</p> <p>Cr 5.a.2 : <i>Le candidat connaît la documentation du framework et est en mesure d'en expliquer les spécificités.</i></p> <p>Cr 5.a.3 : Le candidat maîtrise la syntaxe, l'architecture et les fonctionnalités du framework</p> <p>Cr 5.b.1 : Le choix des dépendances installées est cohérent avec le projet à réaliser.</p> <p>Cr 5.b.2 : La compatibilité de la version du framework et de ses dépendances est respectée.</p> <p>Cr 5.b.3 : Les variables</p>

	<p>C 5.c : Développer une application évolutive avec un framework afin de disposer d'un environnement de développement standardisé pour l'ensemble du projet.</p>	<p>candidat présente son application fonctionnelle et déployée sur un serveur. Il argumente :</p> <ul style="list-style-type: none"> - le fonctionnement global du framework, ses spécificités et son architecture - les choix des dépendances installées - Sa démarche pour solutionner les problématiques rencontrées <p><i>Le jury demande au candidat des modifications ou ajouts de code pour solutionner une problématique inattendue dans l'instant.</i></p> <p>Eléments fournis :</p> <ul style="list-style-type: none"> - Un cahier des charges - Un cahier des fonctionnalités - Un accès au serveur, une base de données ou API <p>Eléments demandés :</p> <ul style="list-style-type: none"> - Tous documents appuyant 	<p>d'environnement sont correctement renseignées.</p> <p>Cr 5.c.1 : Le candidat est en mesure d'adapter les solutions existantes du framework pour résoudre des dysfonctionnements</p> <p>Cr 5.c.2 : Les erreurs de développement sont identifiées,</p> <p>Cr 5.c.3 les outils de débogage sont maîtrisés.</p> <p>Cr 5.c.4 : L'application livrée est fonctionnelle</p>
--	--	---	---

		<p>la conceptualisation et l'argumentation</p> <ul style="list-style-type: none"> - l'application livrée fonctionnelle 	
Bloc 4: Design d'interfaces utilisateur (option 2)			
<p>Activité 6 : Réalisation des maquettes d'interface (site web/ applications)</p> <ul style="list-style-type: none"> - Analyse d'une demande client - Schématisation de l'interface utilisateur - Conception graphique - Prototypage multi support 	<p><i>C 6.a : Comprendre une problématique client et son cahier des charges au moyen d'une grille d'analyse dans le but de concevoir des interfaces adaptées à l'identité du client et de la cible.</i></p> <p>C 6.b : Préparer la maquette et ses différents composants au moyen d'un schéma (Wireframe) regroupant tous les éléments et leurs positionnement , afin de proposer une expérience utilisateur simple, intuitive, et facilement accessible.</p>	<p>Mise en situation professionnelle : projet :</p> <p>Le candidat réalise en autonomie les maquettes d'interfaces de site web ou d'application, adaptées aux différents périphériques en suivant et analysant le cahier des charges fourni.</p> <p>Durant cette période, le candidat doit être force de proposition auprès du client représenté par la personne supervisant l'examen.</p> <p>Le candidat présente ensuite à l'oral les maquettes aux jurys et argumente ses choix graphiques, ainsi que sa réflexion en termes d'interface / expérience utilisateur.</p>	<p>Cr 6.a.1 : Le périmètre métier et la cible sont correctement identifiés.</p> <p>Cr 6.a.2 : L'analyse du marché et de la concurrence est complète et structurée.</p> <p>Cr 6.a.3 : les choix graphiques reflètent l'identité du client et de la cible, ils sont argumentés de façon claire et professionnelle.</p> <p>Cr 6.b.1: La hiérarchisation des contenus textuels met en avant les produits, services et fonctionnalités attendus par le client.</p> <p>Cr 6.b.2 : Prise en compte des critères d'accessibilité : contraste des couleurs, choix et tailles des</p>

	<p>C 6.c : Concevoir des maquettes en utilisant un logiciel professionnel de création graphique afin de présenter au client un prototype d'interface pour son application ou site internet.</p> <p>C 6.d : Faire évoluer le contenu des maquettes afin de l'adapter aux différents périphériques et résolutions en construisant un prototype pour smartphone, tablette et desktop.</p>	<p>Documents attendus :</p> <ul style="list-style-type: none"> - maquettes - Liste des propositions validées par le client. - tous supports permettant d'appuyer l'argumentation - Sources d'inspiration, <ul style="list-style-type: none"> - analyse de la concurrence - Schéma de conception (wireframe) - Fichier validant les choix colorimétriques, typographiques ...(Brand Board) - Prototypage des différentes interactions des éléments interactifs - Export des éléments graphiques optimisés en vue de l'intégration (assets) 	<p>polices de caractères.</p> <p>Cr 6.b.3 : le positionnement et la mise en valeur des composants participent à une bonne expérience utilisateur.</p> <p>Cr 6.c.1: Les calques de la maquette sont organisés en dossiers / fichiers correctement hiérarchisés.</p> <p>6.c.2 : Les prototypages d'événements et d'interactions avec l'interface sont correctement réalisés</p> <p>Cr 6.c.3 : découpage et exportation des éléments graphiques optimisés pour l'affichage.</p> <p>Cr 6.d.1: la hiérarchisation du contenu est adaptée à l'affichage sous différents périphériques.</p> <p>Cr 6.d.2 Les caractéristiques des composants (taille, positionnement, et structure) s'adaptent aux différents formats d'affichage.</p> <p>Cr 6.d.3 : Les choix</p>
--	--	---	--

			ergonomiques effectués permettent un confort de navigation optimal sur les petites résolutions.
Bloc 5: Utiliser la méthodologie DevOps pour automatiser, conteneuriser et déployer une application en continu. (option 3)			
<p>Activité 7 : Automatisation les différentes étapes tout au long cycle de vie d'une application.</p> <ul style="list-style-type: none"> - Identification des processus à automatiser - Programmation de scripts d'automatisation - Conteneurisation de processus et de services. - Orchestration 	<p>C 7.a : Identifier les points d'automatisation possibles en collaborant avec les développeurs durant les phases du cycle de vie du projet afin de faciliter le développement, le déploiement et l'exploitation de l'application.</p> <p>C 7.b : Programmer les actions en utilisant un langage de script afin de construire le processus d'automatisation.</p>	<p>Mise en situation professionnelle : le candidat reçoit le code source d'une application déjà développée. Il en audite le code afin d'en dégager les adaptations en termes d'architecture et de sécurité nécessaires. Il en identifie les points automatisables et en construit des processus d'automatisation aussi bien du côté du développement que du déploiement.</p> <p>Il crée l'environnement conteneurisé nécessaire au fonctionnement de l'application, construit les images avec les services nécessaires, empile les différents services afin d'obtenir une stack fonctionnelle.</p> <p>La stack est testée puis déployée sur un serveur.</p> <p>Le candidat présente auprès du</p>	<p>Cr 7.a.1 : Le candidat a bien analysé les contraintes en termes d'infrastructure et de sécurité</p> <p>Cr 7.a.2 : le candidat propose un ensemble de solutions pertinentes pour automatiser tout ou partie de l'ensemble du processus</p> <p><i>Cr 7.a.3 : Le candidat prend en compte les interactions avec les activités connexes, autant sur la partie développement que sur la partie de l'infrastructure.</i></p> <p>Cr 7.b.1 : Le candidat maîtrise la syntaxe d'un langage de script</p> <p>Cr 7.b.2 : L'automatisation est fonctionnelle et fiabilisée.</p> <p>Cr 7.b.3 : Le candidat planifie des tâches répétitives (planificateur de tâche, cron tab)</p>

	<p>C 7.c : Créer un environnement de développement indépendant à l'aide d'un système d'exploitation pour conteneur dans une machine virtuelle. (exemple : Docker) afin de s'affranchir des contraintes d'infrastructure.</p> <p>C 7.d : Assurer un déploiement continu de l'application en utilisant une plateforme qui permet d'automatiser et de tester le déploiement et la mise à jour. (exemple : Github Action)</p>	<p>jury le déploiement de l'application et explique les procédés qu'il a mis en place pour automatiser, sécuriser et déployer la stack de l'application demandée.</p> <p>Eléments fournis :</p> <ul style="list-style-type: none"> - Un sujet d'exercice sous forme de demande client - une ou plusieurs applications selon la demande du client - un accès à un serveur hôte <p>Eléments demandés :</p> <ul style="list-style-type: none"> - l'application automatisée, conteneurisée et déployée. - tous supports permettant d'appuyer l'argumentation 	<p>Cr 7.c.1 : la machine virtuelle créée par le candidat est configurée et opérationnelle.</p> <p>Cr 7.c.2 : Le système d'exploitation pour conteneur est installé dans la machine d'hébergement virtuelle.</p> <p>Cr 7.c.3 : L'application complète est correctement conteneurisée avec les services et les dépendances nécessaires au fonctionnement de l'application.</p> <p>Cr 7.c.4 : Le fichier de configuration est renseigné et permet de lancer la stack applicative complète avec une seule ligne commande.</p> <p>Cr 7.d.1 : L'architecture serveur est mise en place et fonctionnelle</p> <p>Cr 7.d.2 : L'application est testée avant déploiement.</p> <p>Cr 7.d.3 : L'intégration et le déploiement continus sont testés et l'application est livrée.</p>
--	---	---	---

Validation de la certification

Pour valider un bloc, l'évaluation des acquis doit être supérieure ou égale à 50 %.

Pour valider le titre développeur Web, Il faut obtenir une note supérieure ou égale à 50%

Le calcul de la moyenne des notes est pondéré de la manière suivante:

Contrôle continu: 30%

Stage évalué en entreprise: 20%

Examens évalués par les jurys: 50%

Le titre est obtenu en validant le tronc commun (bloc 1 + bloc 2) , un bloc optionnel au choix ainsi qu'une période de stage en entreprise.

Développeur web option UX-UI designer : tronc commun + bloc 4 + stage

Développeur web option framework (front-end ou back-end) : tronc commun + bloc 3 + stage

Développeur web option DevOps : tronc commun + bloc 5 + stage

Le titre de Développeur Web peut être obtenu selon les modalités suivantes:

1: Le candidat suit la formation complète:

Il doit alors obtenir une note supérieure ou égale à 50% dans tous les blocs pour valider le titre, et 50% en moyenne globale (contrôle continu + stage + examen)

2: Le candidat veut effectuer une validation des acquis d'expérience:

Le candidat peut justifier de son expérience sur 1, 2 ou la totalité des blocs (Voir le processus d'accès par la V.A.E.)

3: Le candidat souhaite valider un bloc uniquement: Chaque bloc peut être validé indépendamment des autres blocs. Néanmoins pour valider la certification dans son ensemble, le candidat devra valider le tronc commun , un bloc optionnel ainsi que d'effectuer une période de stage en entreprise

LEXIQUE

HTML : (Hyper Text Markup Language) langage de balisage utilisé pour décrire la structure et le contenu sémantique d'une page web

CSS : (Cascading Style Sheets) langage décrivant la mise en forme d'un document HTML

JAVASCRIPT : Langage de programmation utilisable dans un navigateur

BRAND BOARD : Proposition cohérente d'une identité graphique montrant les choix colorimétriques, typographiques et les éléments graphiques du design choisi.

WIREFRAME : trame générale schématique de l'agencement des blocs d'une maquette

ES5 / ES6 : (Ecma Script) Normes syntaxiques et standards des langages de scripts. Actuellement les navigateurs supportent le javascript jusqu'au standard ES6.

FRONT END : (coté client) désigne le programme exécuté dans le navigateur dont le code source est visible publiquement

BACK END : (côté serveur) désigne le programme exécuté sur le serveur dont le code source est invisible dans le navigateur.

DOM : (Document Object Model) Interprétation sous forme d'un objet manipulable par javascript d'une page web.

HTTP : Hyper Text Transfert Protocol, Protocole de communication entre le client et le serveur utilisé dans le web.

FRAMEWORK : Cadre de travail, désigne un ensemble d'outils interdépendants utilisés pour créer rapidement et facilement des applications.

MVC : Model View Controller, Désigne un patron de conception d'une architecture de code.

DEVOPS : pratique technique visant à l'unification du développement logiciel et de l'administration des infrastructures informatiques

RGPD : Règlement général sur la protection des données. Cadre réglementaire relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données