



المملكة المغربية

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Direction Recherche et Ingénierie De Formation

Centre de Développement des Compétences en Digital,

Intelligence Artificielle, Audiovisuel et Cinéma



Secteur Digital & Intelligence Artificielle

Développement Digital - Option

Web Full Stack

Programme de formation

Version 2.1

27/07/2022

Version expérimentale
En cours de validation

Développement Digital - Option Web Full Stack

Développement de l'offre de formation
dans les métiers du Digital et de l'Intelligence Artificielle

Ce document est la propriété de l'OFPPT. Il ne peut être utilisé, reproduit ou communiqué à des tiers sans l'autorisation préalable écrite de l'Office.

Version 2.1 (27/07/2022)

Ce document est la propriété de l'Office de la Formation Professionnelle et de la Promotion du Travail. Il ne peut être utilisé, reproduit ou communiqué à des tiers sans l'autorisation préalable écrite de l'Office.

Équipe de production

Coordination

Mohamed Slimani	Directeur CDC Digital, IA, Audiovisuel et Cinéma
-----------------	--

Conception et rédaction

Asmae Youala	Formatrice à ISTA ADARISSA – FES
Imane Bourous	Formatrice à ISTA HAY RIAD – RABAT
Kawtar Harmouchi	Formatrice au CFMOTI – CASABLANCA
Ahmed Naim	Formateur à ISTA CITE DE L’AIR - EL JADIDA
Abdeljalil El Mouden	Développeur full stack Angular à Grupo Avalon
Oussama Haij	Formateur à CMC NADOR
Mohamed Goumih	Formateur à CMC AGADIR
Widad Jakjoud	Formatrice à ISTA TIC - SAFI
Mohamed Chakouj	Formateur à ISTA INEZGANE
Noureddine Ababar	Formateur à CF TIZNIT - TAFRAOUT
Abdelmounim Bendaoud	Formateur à ISGI LAAYOUNE
Ali Id Mansour	Formateur à ISTA SYBA – MARRAKECH
Mariam Mahdaoui	Formatrice à ISTA GESTION SIDI MOUMEN - CASABLANCA
Mohamed Chchab	Formateur à ISTA BENGUERIR
Ilyass Salamat	Ingénieur Concepteur Développeur à SQLI

Validation

Entité	Prénom, Nom, Fonction
DRIF	<ul style="list-style-type: none"> Assia Mohsine - Cheffe de division Coordination des CDC Nezha Rhazouani - Cheffe de division Modes et Méthodes
CDC Digital et IA	<ul style="list-style-type: none"> Mohamed Slimani - Directeur Abdelhak Rahmani - Formateur animateur Driss MHAMDI - Formateur animateur Ghizlane El Khattabi - Formatrice animatrice Soukaina Laouija - Formatrice animatrice

Suivi des versions

Version	Auteur	Date	Modifications
1.0	Asmae Youala	10/05/2022	Consolidation des éléments de contenus proposés par les membres de la commission de conception et de rédaction
1.1	Asmae Youala	16/05/2022	Intégration de la compétence C1
2.0	Abdelhak Rahmani Driss Mhamdi Ghizlane El Khattabi Soukaina Laouija	20/07/2022	Mise à jour des contenus des compétences en prenant compte les remarques de la commission des formateurs, professionnels et formateurs animateurs
2.1	Abdelhak Rahmani Ghizlane El Khattabi Soukaina Laouija Driss Mhamdi	27/07/2022	Mise à jour des compétences de la partie "tronc commun" à la suite de l'exploitation des questionnaires administrés à l'ensemble des formateurs du développement digital, ainsi que les remarques de la commission des formateurs, professionnels et formateurs animateurs

Remerciements

La production du présent document a été possible grâce à la collaboration et à la participation de nombreuses personnes, que ce soient des professionnels du métier ou des formateurs.

Il y a lieu de souligner la qualité des renseignements fournis par les personnes consultées à titre de professionnels du domaine du Développement Digital et de remercier particulièrement celles qui ont généreusement accepté de partager leur expérience à l'égard du métier de technicien spécialisé en Développement Digital - Option Web Full Stack et celles qui ont participé à la validation de ce Projet de formation.

Table des matières

Remerciements.....	5
Acronymes	7
Présentation du programme de formation	9
Conditions d'accès au programme de formation.....	10
Buts du programme de formation.....	13
Matrice des compétences	15
Phases d'acquisition d'une compétence	17
Rôle du formateur en Approche Par Compétences	18
Logigramme des compétences.....	20
Glossaire	21
Fiches prescrites et suggestions pédagogiques.....	24
Compétence 1 : « Se situer au regard du métier et de la démarche de formation »	24
Compétence 2 : « Acquérir les bases de l'algorithmique »	28
Compétence 3 : « Programmer en Orienté Objet »	34
Compétence 4 : « Développer des sites Web statiques »	42
Compétence 5 : « Programmer en Javascript»	49
Compétence 6 : « Manipuler des bases de données ».....	56
Compétence 7 : « Développer des sites web dynamiques ».....	61
Compétence 8 : « S'initier à la sécurité des systèmes d'information »	67
Compétence 9 : « Préparer un projet web».....	72
Compétence 10 : « Adopter l'approche agile ».....	82
Compétence 11 : « Gérer les données »	94
Compétence 12 : « Développer en front-end»	102
Compétence 13 : « Développer en back-end»	112
Compétence 14 : « Créer une application Cloud native».....	123
Compétence 15 : « Réaliser un projet de synthèse ».....	130
Compétence 16 : « S'intégrer en milieu professionnel»	137

Acronymes

ESN	Entreprise de services du numérique
SSII	Société de services en ingénierie informatique
APC	Approche Par Compétence
RM	Référentiel Métier
IA	Intelligence Artificielle
RGPD	Règlement général sur la protection des données
IFML	Interaction Flow Modeling Language
WebML	Web Modeling Language
UML	Unified Modeling Language
SQL	Structured Query Language
PL/SQL	Procedural Language / Structured Query Language
SQL/PSM	Structured Query Language / Persistent Stored Modules
SGBD	Système de Gestion de Bases de Données
JSON	JavaScript Objet Notation
XML	Extensible Markup Language
CSV	Comma-separated values
CRUD	(Create, Read, Update, Delete)
SPA	Single Page Application
PWA	Progressive Web Application
API	Application Programming Interface (Interface de programmation d'application)
TDD	Test-Driven Development
BDD	Behavior-Driven Development
DevSecOps	Development -Security- Operations
CI/CD	Intégration Continue / Livraison Continue
SOAP	Simple Object Access Protocol

REST	Representational State Transfer
GraphQL	Graph Query Language
IAAS	Infrastructure-as-a-Service
PAAS	Platform-as-a-service
SAAS	Software as a Service
AST	Analyse de Situation de Travail
POO	Programmation Orientée Objet
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	PHP HyperText Preprocessor
GANTT	Generalized Activity Normalization Time Table
DOM	Document Object Model
MERISE	Méthode d'analyse, de conception, de développement et de gestion de projet informatique
MVC	Model View Controller
MVVM	Model-View View-Model
VIPER	View Interface Presenter Entity Router
HTTP	HyperText Transfer Protocol
SSL	Secure Socket Layer
OAUTH	Open Authorization Protocol
YAML	Yet Another Markup Language

Présentation du programme de formation

Le programme de formation de la filière Développement Digital - Option Web Full Stack s'inscrit dans les orientations retenues par le Département de la Formation Professionnelle, concernant la formation professionnelle. Il a été conçu suivant le Cadre méthodologique d'élaboration et de mise en œuvre des programmes de formation selon l'Approche Par Compétences qui exige, notamment, la participation conjointe des milieux du travail et de la formation.

Le programme de formation est défini par compétences, formulé par objectifs et structuré en modules. Il est conçu selon une approche globale qui tient compte à la fois de facteurs tels les besoins de formation, la situation de travail, les finalités, les buts ainsi que les stratégies et les moyens pour atteindre les objectifs.

Dans le programme de formation, on énonce et structure les compétences que le participant doit acquérir pour obtenir son Diplôme de technicien spécialisé en Développement Digital - Option Web Full Stack. Ce programme de formation doit servir de référence pour la planification de la formation et de l'apprentissage ainsi que pour la préparation du matériel didactique et du matériel d'évaluation.

La durée du programme de formation est de 1530 heures ; de ce nombre, 940 heures sont consacrées à l'acquisition de compétences spécifiques et 590 heures à l'acquisition de compétences transversales. Le programme de formation est divisé en 16 modules dont la durée varie de 15 à 120 heures. Cette durée comprend le temps requis pour l'évaluation des apprentissages aux fins de l'obtention de L'attestation de Poursuite de la Formation.

Ce programme de formation a été élaboré en 2022.

NB : L'usage du masculin, dans le présent document, n'a pour but que d'alléger le texte pour en faciliter la lecture.

Conditions d'accès au programme de formation

Code : DIA_DEVOWFS_TS

« Développement Digital - Option Web Full Stack »

Année d'approbation : 2021

Type de sanction : Technicien spécialisé

Nombre d'unités : 102 (unité de formation : 15h)

Nombre de modules : 16

Durée totale : 1530

Pour être admis au programme « Développement Digital - Option Web Full Stack », il suffit de satisfaire aux conditions suivantes :

- Être titulaire du baccalauréat au moins ou équivalent, muni d'une copie conforme du relevé des notes ;
- Avoir au plus 30 ans, au 1er septembre de l'année scolaire en cours, justifié par un extrait d'acte de naissance original ou une copie conforme de la Carte Nationale d'Identité ;
- Avoir au plus 33 ans, au 1er septembre de l'année scolaire en cours, pour les bénéficiaires du Système des Passerelles ;
- Les candidats ayant un baccalauréat de branche scientifique ou technique industrielle.

Le concours, pour ceux qui sont admissibles, est composé d'une entrevue permettant de vérifier :

- L'intérêt du candidat pour le métier ;
- La motivation du candidat pour la formation ;
- La capacité du candidat à comprendre et à parler le français

PREMIERE PARTIE

Tableau synthèse du programme de formation

Nombre de modules : 16

Codes du programme : DIA_DEV_TS (Tronc Commun)

DIA_DEVOWFS_TS (Option)

Durée en heures : 1530

Niveau : TS

Valeur en unités : 102

Code	Num	Compétence	Durée (heures)
DIA_DEV_TS - 01	1	Se situer au regard du métier et de la démarche de formation	15h
DIA_DEV_TS - 02	2	Acquérir les bases de l'algorithmique	120h
DIA_DEV_TS - 03	3	Programmer en orienté objet	120h
DIA_DEV_TS - 04	4	Développer des sites web statiques	110h
DIA_DEV_TS - 05	5	Programmer en JavaScript	110h
DIA_DEV_TS - 06	6	Manipuler des bases de données	100h
DIA_DEV_TS - 07	7	Développer des sites web dynamiques	120h
DIA_DEV_TS - 08	8	S'initier à la sécurité des systèmes d'information	45h
DIA_DEVOWFS_TS - 09	9	Préparer un projet web	60h
DIA_DEVOWFS_TS - 10	10	Adopter l'approche agile	100h
DIA_DEVOWFS_TS - 11	11	Gérer les données	100h
DIA_DEVOWFS_TS - 12	12	Développer en front-end	100h
DIA_DEVOWFS_TS - 13	13	Développer en back-end	120h
DIA_DEVOWFS_TS - 14	14	Créer une application Cloud native	90h
DIA_DEVOWFS_TS - 15	15	Réaliser un projet de synthèse	60h
DIA_DEVOWFS_TS - 16	16	S'intégrer en milieu professionnel	160h

Buts du programme de formation

Le programme « Développement Digital - Option Web Full Stack » vise à former des personnes aptes à effectuer une analyse fonctionnelle et technique du besoin client, développer des fonctionnalités, tester et apporter des correctifs, mettre en œuvre des actions d'améliorations et en assurer le suivi tout en garantissant le respect des consignes de sécurité et des exigences de productivité et de qualité.

Le programme prépare également les participants à assumer le bon fonctionnement de l'application dont ils auront la responsabilité. Le participant exerce son métier dans différents types de structures et dans le cadre d'environnements et de projets variés.

Le « Développement Digital - Option Web Full Stack » est exercé dans des entreprises du secteur digital (ESN / SSII ...). Toutefois, sa formation doit pouvoir l'amener à travailler dans d'autres secteurs d'activités à savoir : l'administration, l'industrie ...

La polyvalence des participants est assurée par le développement de compétences transversales. Ce sont celles qui concernent l'hygiène, santé et sécurité en milieu de travail, l'utilisation d'un poste de travail informatique, la gestion des projets, la gestion de temps et des priorités, la résolution de problèmes, la gestion des approvisionnements, le soft skills ainsi que la communication.

La maîtrise des tâches professionnelles liées au métier est quant à elle assurée par l'acquisition des compétences managériales propres au métier. Ce sont celles qui concernent la gestion d'équipe, la mise en œuvre de système de maintenance, la gestion des projets ainsi que la mise en œuvre des actions d'améliorations continues.

Conformément aux buts généraux de la formation professionnelle, le programme « Développement Digital - Option Web Full Stack » vise à :

1. Rendre la personne efficace dans l'exercice d'une profession ou d'un métier), soit :

- Lui permettre de jouer les rôles, d'exercer les fonctions et d'exécuter des tâches et des activités associées à une profession ou un métier ;
- Lui permettre d'évoluer adéquatement dans un milieu de travail ;

- Lui permettre de développer des habiletés intellectuelles et techniques qui entraînent des choix judicieux ;
- Lui permettre de développer une préoccupation constante de la santé et de la sécurité au travail.

2. Assurer l'intégration de la personne à la vie professionnelle, soit :

- Lui permettre de connaître le marché du travail en général ;
- Lui permettre de connaître le contexte particulier de la profession choisie.

3. Favoriser l'évolution de la personne et l'approfondissement de savoirs professionnels, soit :

- Lui permettre de développer son autonomie et sa capacité d'apprendre ainsi que d'acquérir des méthodes de travail ;
- Lui permettre de comprendre les principes sous-jacents aux techniques et aux technologies utilisées ;
- Lui permettre de développer sa faculté d'expression, sa créativité, son sens de l'initiative et son esprit d'entreprise ;
- Lui permettre d'adopter des attitudes essentielles à son succès professionnel, de développer son sens des responsabilités et de viser l'excellence.

4. Favoriser la mobilité professionnelle de la personne, soit :

- Lui permettre d'adopter une attitude positive à l'égard des changements ;
- Lui permettre de se donner des moyens pour gérer sa carrière.

Matrice des compétences

La matrice des compétences met en évidence les compétences générales (portent sur des activités communes à plusieurs tâches ou à plusieurs situations), les compétences spécifiques (portent sur des tâches et des activités directement liées au métier ou à la profession) ainsi que les grandes étapes du processus de travail.

Le tableau est à trois entrées permettant de voir les liens qui unissent les éléments placés à l'horizontale (compétences transversales) et ceux placés à la verticale (compétences spécifiques). Le symbole Δ montre qu'il existe une relation entre une compétence spécifique et une étape du processus de travail.

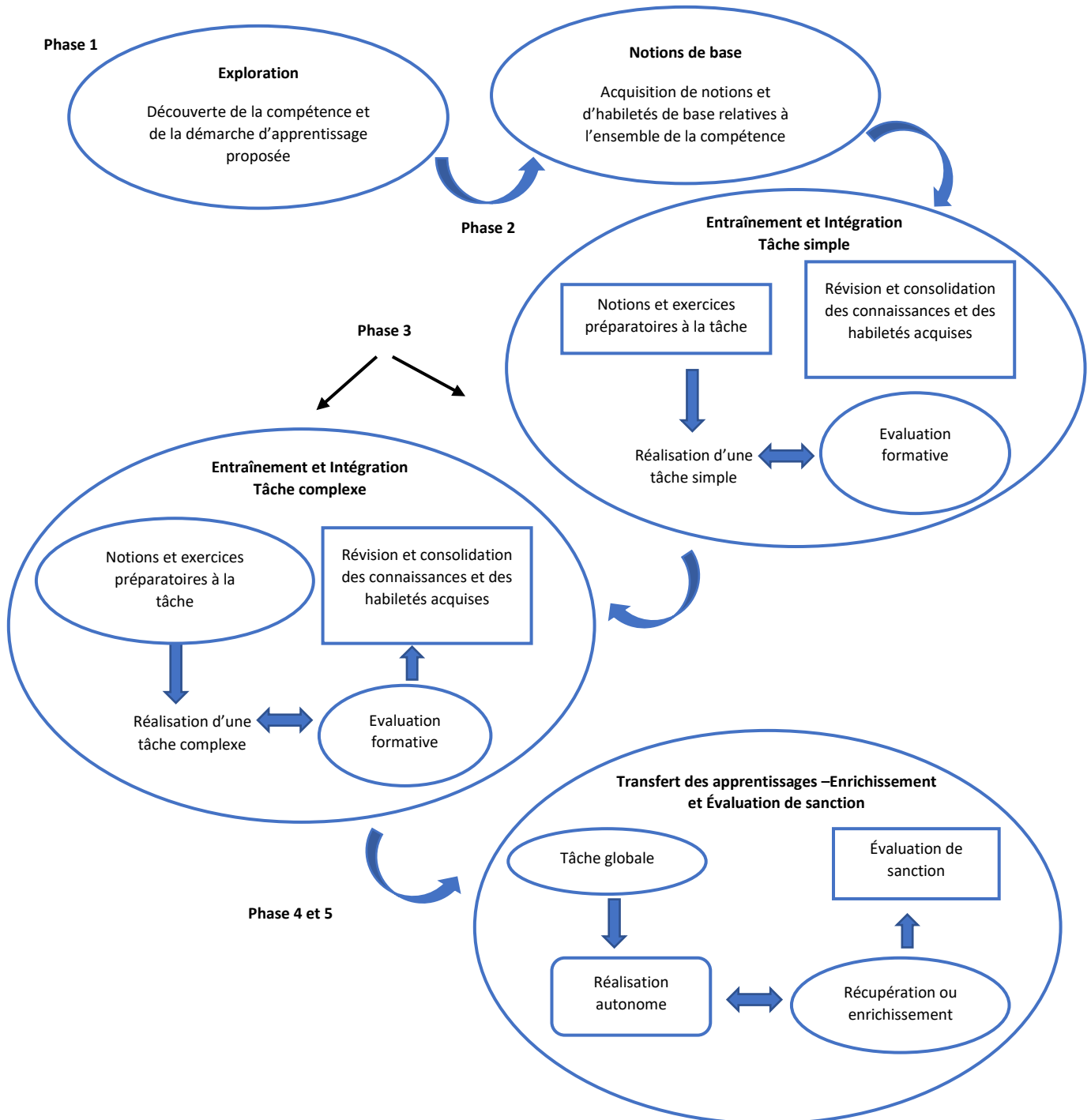
Le symbole \bigcirc marque quant à lui un rapport entre une compétence transversale et une compétence spécifique. Des symboles noirs \bullet \blacktriangle indiquent en plus que l'on tient compte de ces liens dans la formulation d'objectifs visant l'acquisition des compétences spécifiques.

La logique qui a présidé à la conception de la matrice influe sur la séquence d'enseignement de ces modules. De façon générale, on prend en considération une certaine progression dans la complexité des apprentissages et le développement de l'autonomie du participant. De ce fait, les compétences spécifiques sont présentées dans l'ordre à privilégier pour l'enseignement et servent de point de départ pour l'agencement de l'ensemble des modules. Certains deviennent ainsi préalables à d'autres ou doivent être vus en parallèle.

Développement Digital - Option Web Full Stack		DUREE	PROCESSUS (grandes étapes)					COMPÉTENCES TRANSVERSALES							TOTAUX	
MATRICE DES OBJETS DE FORMATION			Contribuer à l' analyse fonctionnelle et technique du besoin client	Développer les fonctionnalités demandées	Livrer des productions clean et sans bug	Fournir le support technique au client	Assurer la maintenance	Se situer au regard du métier et de la démarche de formation	Acquérir les bases de l'algorithmique	Programmer en orienté objet	Développer des sites web statiques	S' initier à la sécurité des systèmes d'information	Préparer un projet web	Adopter l' approche agile	NOMBRE DE COMPÉTENCES	DUREE DE LA FORMATION
N°	NUMEROS							1	2	3	4	8	9	10	7	
	Durée	h						15	120	120	110	45	60	120		590
5	Programmer en JavaScript	110	△	▲	▲		△	○	●	○	●	○	○	○		
6	Manipuler des bases de données	100	▲	△	△			○	○			○	○	○		
7	Développer des sites web dynamiques	120	△	▲	▲		△	○	●	●	●	○	○	○		
11	Gérer les données	90	△	▲	△			○	●			○	○	○		
12	Développer en front-end	90	△	▲	▲		△	○	●	○	●	○	○	○		
13	Développer en back-end	120	△	▲	▲		△	○	●	●	●	●	●	●		
14	Créer une application Cloud native	90	△	▲	▲		△	○	●	●	●	○	●	●		
15	Réaliser un projet de synthèse	60	▲	▲	▲	△	△	○	●	○	●	○	●	●		
16	S'intégrer en milieu professionnel	160	△	△	△	△	△	○	●	○	●	○	●	○		
NOMBRE DE COMPETENCES		9													16	
DUREE DE LA FORMATION		940														1530

Phases d'acquisition d'une compétence

Pour favoriser l'atteinte des objectifs, il est suggéré de structurer les apprentissages de façon progressive, c'est-à-dire d'avoir recours à des activités d'apprentissage, d'évaluation formative, d'enseignement correctif ou d'enrichissement, selon le cas. Le processus d'acquisition de compétences est illustré par les schémas ci-dessous.



Rôle du formateur en Approche Par Compétences

Le formateur doit adapter son enseignement en tenant compte :

- d'une approche intégrée des objets de formation ;
- du rythme individuel et de la façon d'apprendre des participants ;
- d'une responsabilité accrue des participants au regard de leurs apprentissages ;
- du vécu professionnel des participants.

Pour exercer pleinement leur rôle, le formateur doit :

- planifier et organiser leur enseignement ;
- informer les participants ;
- effectuer de l'animation pédagogique ;
- évaluer les apprentissages.

Planification et organisation de l'enseignement

Cette fonction consiste tout d'abord à situer les modules dont il a la responsabilité et ensuite, à l'aide du logigramme de la séquence d'enseignement :

- ajouter ou ajuster, au besoin, les phases préalables et les éléments du contenu ;
- prévoir et produire des activités propres à ces modules ;
- coordonner des activités d'apprentissage pour les participants ;
- répartir les postes de travail et le matériel nécessaire ;
- agencer et élaborer des activités d'apprentissage, d'évaluation, d'enseignement correctif et d'enrichissement.

Information au participant

Cette autre fonction consiste à :

- situer les participants par rapport à l'ensemble du programme et, aussi, par rapport au module en cours ;
- fournir aux participants les données utiles à une compréhension suffisante des tâches reliées au métier ;
- faire ressortir l'importance et la pertinence des apprentissages à réaliser.

Note : Il revient à chaque formateur de situer les participants par rapport à l'ensemble de leur formation et de les stimuler dans leurs apprentissages et de leur fournir, au début de chaque cours et de chaque activité importante, les données nécessaires à ces fins.

Animation pédagogique

Le formateur doit :

- guider les apprentissages par un rappel des objectifs, par la détermination des phases préalables et par la formulation d'indications sur les activités à réaliser ;
- créer un climat de confiance reposant sur le respect des personnes et de leur autonomie, ainsi que sur la clarification des enjeux réels ;
- maintenir l'intérêt des participants tout au long de leur cheminement par des propositions d'activités intéressantes et diversifiées, par un dosage judicieux du niveau de difficulté, par l'utilisation d'approches à caractère pratique et par une ouverture aux préoccupations personnelles des participants ;
- encadrer les activités d'apprentissage par l'implantation d'un système souple et efficace de suivi des participants , par une assistance particulière aux participants en difficulté et par une direction adéquate des participants vers des activités d'apprentissage, d'évaluation, d'enseignement correctif et d'enrichissement ;
- fournir des explications claires et justes au groupe et à chaque participant.

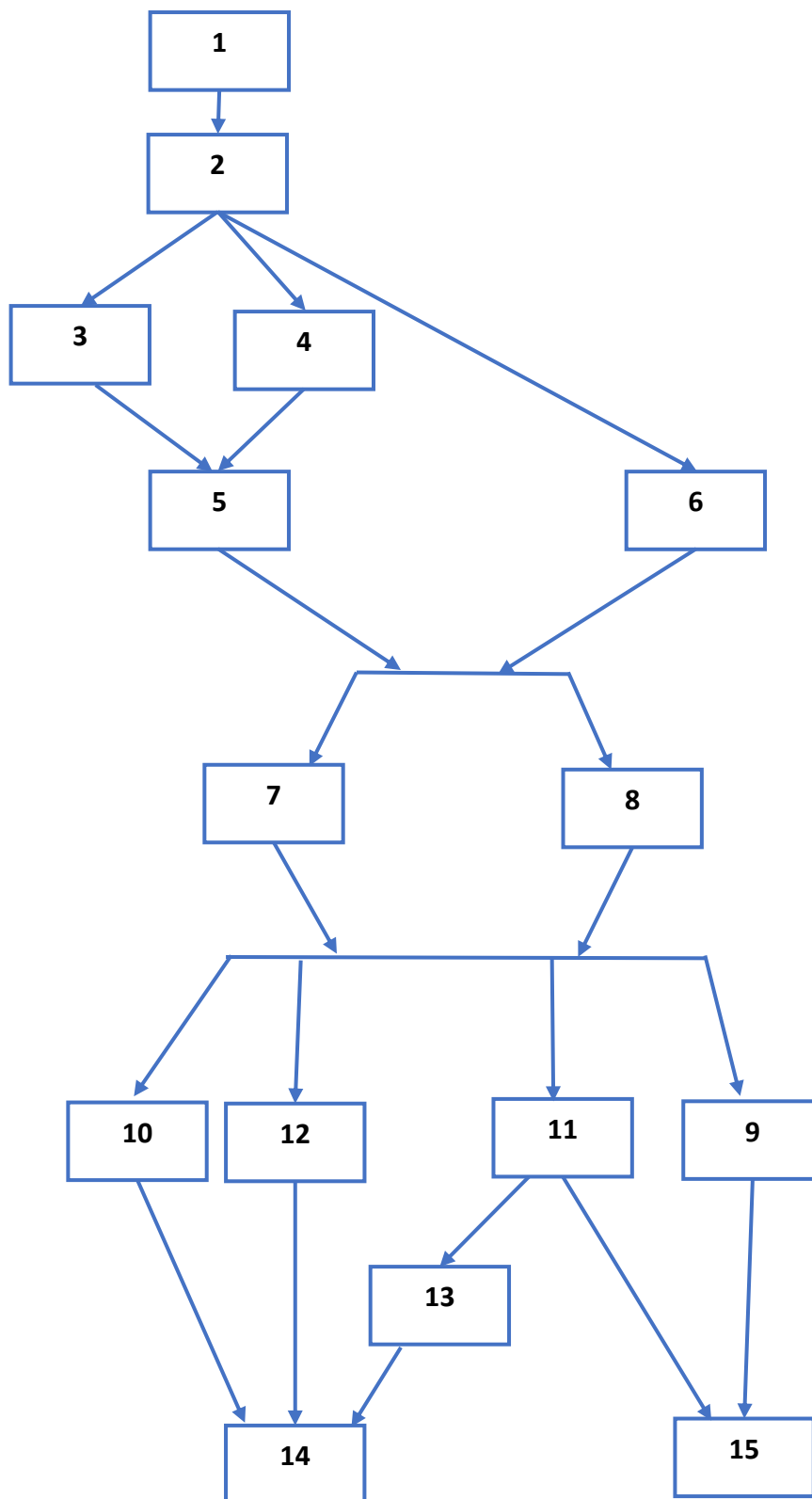
Évaluation des compétences

Le formateur assure le suivi mentionné précédemment :

- en produisant et en utilisant des instruments d'évaluation formative afin de guider le participant dans son cheminement et lui fournir des informations de façon continue sur l'état de son cheminement;
- en administrant les épreuves aux fins de certification;
- en fournissant les résultats de l'évaluation de certification à la personne responsable dans le centre de formation.

Logigramme des compétences

Filière : «Développement Digital - Option Web Full Stack »



Glossaire

Programme de formation professionnelle

Un programme est un ensemble cohérent de compétences à acquérir, formulé en termes d'objectifs et découpé en modules. Il décrit les apprentissages attendus du stagiaire en fonction d'une performance déterminée. Ses objectifs et son contenu sont obligatoires.

Buts de la formation

Les buts de la formation sont les énoncés des intentions éducatives retenues pour le programme. Il s'agit d'une adaptation des buts généraux de la formation professionnelle pour un programme de formation donné.

Compétence

Une compétence est un pouvoir d'agir, de réussir et de progresser qui permet de réaliser adéquatement des tâches ou des activités et qui se fonde sur un ensemble intégré de connaissances, d'habiletés, d'attitudes et de comportements. Les compétences sont de deux types :

- Les compétences spécifiques qui portent sur des tâches types du métier ou de la fonction de travail et qui rendent la personne apte à assurer avec efficacité la production d'un bien ou d'un service.
- Les compétences transversales qui portent sur une activité de travail ou de vie professionnelle qui déborde du champ spécifique des tâches du métier lui-même ; ces compétences peuvent être transférables à plusieurs activités de travail.

Objectifs généraux

Les objectifs généraux servent à catégoriser les compétences à faire acquérir par le stagiaire. Ils servent à orienter et à regrouper les objectifs opérationnels.

Objectifs opérationnels

L'objectif opérationnel est défini en fonction d'un comportement relativement fermé et décrit les actions et les résultats attendus du stagiaire. Il comprend cinq composantes :

- Le comportement attendu qui présente la compétence.
- Les conditions d'évaluation qui renseignent sur les conditions qui prévalent au moment de l'évaluation de sanction : contexte, matériel, etc.
- Les précisions sur le comportement attendu qui décrivent des éléments essentiels à la compréhension de la compétence.

- Les critères particuliers de performance qui définissent des exigences à respecter et accompagnent chacune des précisions sur le comportement. Ils permettent également de porter un jugement rigoureux sur l'atteinte de la compétence.
- Les critères généraux de performance qui définissent des exigences liées à l'accomplissement d'une tâche ou d'une activité et donnent des indications sur le niveau de performance recherché ou sur la qualité globale d'un produit ou d'un service. Ils sont également rattachés à l'ensemble ou à plusieurs précisions sur le comportement attendu.

Module de formation

Subdivision autonome d'un programme de formation professionnelle formant en soi un tout cohérent et signifiant.

Unité

Étalon servant à exprimer la valeur de chacun des modules d'un programme de formation en attribuant à ces composantes un certain nombre de points pouvant s'accumuler pour l'obtention d'un diplôme ; l'unité correspond à 15 heures de formation.

DEUXIEME PARTIE

Fiches prescrites et suggestions pédagogiques

Fiche prescrite

Compétence 1 : « Se situer au regard du métier et de la démarche de formation »	
Code de la compétence : DIA_DEV_TS-01	Durée : 15 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et/ou en groupe À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales De consignes De spécifications fonctionnelles De base documentaire À l'aide : <ul style="list-style-type: none"> D'internet Du réseau professionnel 	<ul style="list-style-type: none"> Identification précise des différents métiers Utilisation judicieuse des outils de recherche d'emploi Suivi d'une démarche adéquate pour la connaissance du marché du travail Connaissance approfondie du cadre de formation proposé
Éléments de la compétence	Critères particuliers de performance
A. Découvrir les métiers du Développement Digital	<ul style="list-style-type: none"> Compréhension générale des métiers liés au Développement Identification globale des différents métiers du Développement Recueil ciblé des compétences mobilisées
B. Identifier les modalités de formation	<ul style="list-style-type: none"> Cartographie des compétences de l'année 1 Cartographie des compétences de l'année de spécialisation Usage des ressources de formation
C. Identifier les compétences relatives aux options	<ul style="list-style-type: none"> Connaissance des compétences spécifiques Connaissance des perspectives de travail

Suggestions Pédagogiques

Compétence 1 :	« Se situer au regard du métier et de la démarche de formation »	Code : DIA_DEV_TS-01
DURÉE : « 15 h »	Compétences Préalables : Aucune compétence préalable nécessaire	
Type de compétences : Transversale	Compétences en parallèles : Compétence à effectuer seule	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Découvrir les spécificités des métiers du Développement	A.1 – S'informer sur le marché du travail	<ul style="list-style-type: none"> Identification des secteurs d'activité liés au Développement Représentation du contexte de travail selon les secteurs des entreprises 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des ressources fournis par le formateur (polycop, documents, cours) Quizz sur le secteur d'activité du digital au Maroc 	50%
	A.2 – Se renseigner sur les compétences métier	<ul style="list-style-type: none"> Listing des comportements et postures professionnelles à adopter Reconnaissances des missions et tâches à effectuer dans les métiers du Développement 		

			<ul style="list-style-type: none"> Vidéos témoignages et interviews de professionnels qui décrivent leur entreprise et leur métier 	
B. Identifier les modalités de formation	B.1 – Comprendre les objectifs de la formation	<ul style="list-style-type: none"> Connaissance des objectifs à atteindre au terme de la formation Utilisation des ressources pédagogiques Recueil sur les différents modes d'évaluation durant le parcours de formation 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports pédagogiques mis à disposition durant la formation sur la plateforme d'apprentissage Investigations sur les secteurs d'activités auprès de professionnels et entreprises Création d'un compte sur un réseau professionnel 	30 %
	B.2 - Situer les compétences de la formation	<ul style="list-style-type: none"> Visualisation sous forme de cartographie des compétences de l'année 1 Visualisation sous forme de cartographie des compétences de l'année de spécialisation 		
	B.3 - Projeter sa formation dans le milieu de travail	<ul style="list-style-type: none"> Utilisation des réseaux professionnels Compréhension des marchés de l'emploi (ouvert, cache) Observation du milieu de travail 		

C. Identifier les compétences relatives aux options	C.1 Identifier les compétences relatives au développement web full stack	<ul style="list-style-type: none"> • Présentation du métier • Enumération des compétences relatives au métier • Perspectives professionnelles 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports pédagogiques mis à disposition durant la formation sur la plateforme d'apprentissage • Investigations sur les métiers du développement digital 	20%
	C.2 Identifier les compétences relatives aux applications mobiles	<ul style="list-style-type: none"> • Présentation du métier • Enumération des compétences relatives au métier • Perspectives professionnelles 		
	C.3 Identifier les compétences relatives au développement RV/RA	<ul style="list-style-type: none"> • Présentation du métier • Enumération des compétences relatives au métier • Perspectives professionnelles 		

Fiche prescrite

Compétence 2 : « Acquérir les bases de l'algorithmique »	
Code de la compétence : DIA_DEV_TS-02	Durée : 120 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et / ou en groupe À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales De consignes De spécifications fonctionnelles De spécifications techniques De cahier des charges À l'aide : <ul style="list-style-type: none"> D'un logiciel de PAO D'un éditeur de texte De logiciels : D'exécution d'algorithmes (Algobox, Visual Code, Larp...) D'environnement intégré de développement IDE (VS Code, ...) 	<ul style="list-style-type: none"> Identification précise des données d'entrée, de sorties et des traitements d'un problème réel Ciblage correct des différents types de traitements et leurs applications Définition claire d'un algorithme Bonne lecture et compréhension d'un algorithme Traduction adéquate d'un algorithme en Python Création des scripts fonctionnels en Python Choix adéquat des bibliothèques standards Déploiement correct des solutions Python
Éléments de la compétence	Critères particuliers de performance
A. Modéliser un problème	<ul style="list-style-type: none"> Modélisation adéquate de l'esprit logique Structuration correcte de sa pensée Décomposition logique d'un problème en tâches simples
B. Formuler un traitement	<ul style="list-style-type: none"> Manipulation correcte des structures logiques Ecriture correcte d'un algorithme Manipulation correcte des variables Maîtrise de la définition des traitements

	<p>alternatifs</p> <ul style="list-style-type: none">• Maîtrise de la définition des traitements répétitifs• Maîtrise de l'utilisation des fonctions• Manipulation correcte des tableaux
C. Programmer en Python	<ul style="list-style-type: none">• Traduction structurée des algorithmes en Python• Suivi et visualisation claire de l'exécution d'un algorithme codé en Python• Maîtrise de l'utilisation des fonctions Python• Utilisation adéquate des structures de données• Manipulation correcte des fichiers• Management des bibliothèques
D. Déployer la solution Python	<ul style="list-style-type: none">• Gestion adaptée des erreurs en Python• Déploiement correct d'une solution Python

Suggestions Pédagogiques

Compétence 2 :	« Acquérir les bases de l'algorithmique »	Code : DIA_DEV_TS-02
DURÉE : « 120 h »	Compétences Préalables : Compétence 1	
Type de compétences : Transversale	Compétences en parallèles : Aucune	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Modéliser un problème	A.1 – Analyser un problème	<ul style="list-style-type: none"> Définition du problème (Contexte, Entrées/Sorties, traitements) Types de traitement des données 	<ul style="list-style-type: none"> À l'aide des supports fournis par le formateur (polycop, documents, cours) Exercice d'application sur l'analyse d'un problème Exercice pratique sur les approches d'analyses des traitements sur un problème (ascendante / descendante) 	10 %
	A.2 – Identifier les approches d'analyse d'un problème	<ul style="list-style-type: none"> Approche descendante Approche ascendante 		
B. Formuler un traitement	B.1 - Reconnaître la structure d'un algorithme	<ul style="list-style-type: none"> Définition d'un algorithme Objets informatiques (variable, constante, type) 	<ul style="list-style-type: none"> Exercice d'application sur les variables, constantes, affectations Exercice d'application sur les entrées 	45 %

		<ul style="list-style-type: none"> Structure d'un algorithme 	sorties d'un problème donné	
	B.2 – Connaître les bases	<ul style="list-style-type: none"> Traitement séquentiel (affectation, lecture et écriture) Traitement alternatif (conditions) Traitement itératif (boucles) 	<ul style="list-style-type: none"> Exercice d'application sur le déroulement séquentiel d'un algorithme sur un problème donné Exercice d'application sur les traitements alternatifs et itératifs 	
	B.3 Structurer un algorithme	<ul style="list-style-type: none"> Fonctions Portée des variables 	<ul style="list-style-type: none"> Exercice d'application sur la manipulation des tableaux Exercice d'application sur la manipulation des fonctions 	
	B.4 - Structurer les données	<ul style="list-style-type: none"> Chaînes de caractères Différents types de tableaux 	<ul style="list-style-type: none"> Quiz sur les fonctions 	

C. Programmer en Python	C.1 - Transformer une suite d'étapes algorithmique en une suite d'instructions Python	<ul style="list-style-type: none"> • Critères de choix d'un langage de programmation • Bloc d'instruction • Conversion de l'algorithme en Python • Optimisation du code (Bonnes pratiques de codage, commentaires...) 	<ul style="list-style-type: none"> • À l'aide des supports fournis par le formateur (polycop, documents, cours) • Travaux pratiques sur ordinateur sur la traduction d'un algorithme en python • Exercices sur la manipulation des fichiers • Quiz sur les fichiers 	40 %
	C.2 - Manipuler les données	<ul style="list-style-type: none"> • Manipulation des fonctions/lambda • Listes, tuples, dictionnaires, ensembles (set) • Fichiers de données (textes, csv...) • Bibliothèques standards (math, chaines de caractères, rand...) 	<ul style="list-style-type: none"> • Exercices sur les structures de données • Apprentissage par projet 	
D. Déployer la solution Python	D.1 - Déboguer le code Python	<ul style="list-style-type: none"> • Débogage • Gestion des erreurs (compilation, syntaxe) • Outils de suivi et de Visualisation de l'exécution d'un code Python 	<ul style="list-style-type: none"> • Exercices d'application sur le débogage du code • Exercices d'application sur le déploiement et la création des fichiers d'installation des 	5%

	D.2 - Déployer une solution Python	<ul style="list-style-type: none">• Outils de déploiement de solution Python• Création de fichiers d'installation de solution Python• Documentation du programme	solutions python	
--	------------------------------------	--	------------------	--

Fiche prescrite

Compétence 3 : « Programmer en Orienté Objet »	
Code de la compétence : DIA_DEV_TS-03	Durée : 120 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et / ou en groupe À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales De consignes De spécifications fonctionnelles De spécifications techniques De cahier des charges À l'aide : <ul style="list-style-type: none"> Éditeur de texte ou IDE (VS Code...) Langage Python Terminal ou Console DOS 	<ul style="list-style-type: none"> Maîtrise de la notion de « objet/classe » en programmation Différenciation claire entre la programmation Orientée Objet et la Programmation structurée Implémentation adéquate des principes et des règles de base de la POO utilisées dans les différentes plateformes Application conforme d'une solution de qualité <ul style="list-style-type: none"> Réponse aux exigences fonctionnelles demandées Élaboration d'un programme fiable, optimisé, lisible et facile d'utilisation Productions de solutions extensibles et maintenables
Éléments de la compétence	Critères particuliers de performance
A. Appréhender le paradigme de la POO	<ul style="list-style-type: none"> Définition exacte d'un objet Modélisation et création exactes d'une classe Exploration étoffée des paradigmes de programmation Bonne exploration des concepts de base de l'orienté objet Mise en œuvre précise de l'encapsulation
B. Connaître les principaux piliers de la POO	<ul style="list-style-type: none"> Mise en œuvre précise de l'abstraction Application adéquate du polymorphisme Adaptation conforme de l'héritage Bonne manipulation des interfaces

C. Coder des solutions orientées objet	<ul style="list-style-type: none">• Communication simplifiée entre objets• Manipulation correcte des données (listes, tuples, fichiers, ...)• Manipulation correcte des expressions régulières• Gestion optimale des exceptions
D. Manipuler les modules et les bibliothèques	<ul style="list-style-type: none">• Implémentation correcte des bibliothèques• Intégration optimale des bibliothèques des interfaces graphiques• Manipulation correcte des interfaces graphiques• Management des modules

Suggestions Pédagogiques

Compétence 3 :	« Programmer en Orienté Objet »	Code : DIA_DEV_TS-03
DURÉE : « 120 h »	Compétences Préalables : Compétence 1/2	
Type de compétences : Transversale	Compétences en parallèles : Compétences 4	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Appréhender le paradigme de la POO	A.1. Introduire la POO	<ul style="list-style-type: none"> Introduction à la programmation Orientée Objet Brève historique de l'évolution des langages de programmation Orientée Objet Connaissance des avantages de la POO par rapport aux autres paradigmes 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours) Quizz MindMap Projet de synthèse 	5%

	A.2. Définir une classe	<ul style="list-style-type: none"> • Définition d'une classe • Modélisation d'une classe • Composantes d'une classe <ul style="list-style-type: none"> - Attributs - Types de données - Visibilité - Constructeurs : rôles et types - Destructeurs - Attributs de classe 	<ul style="list-style-type: none"> - Rédaction de besoin - Analyse d'une situation - Proposition de la démarche de la solution 	
	A.3 – Connaître l'encapsulation	<ul style="list-style-type: none"> • Principe de l'encapsulation • Modificateurs et accesseurs (getters, setters, ...) 		
	A.4. Créer un objet	<ul style="list-style-type: none"> • Définition d'un objet • Instanciation • Destruction explicite d'un objet 		
	A.5. Manipuler les méthodes	<ul style="list-style-type: none"> • Définition d'une méthode • Visibilité d'une méthode • Paramètres d'une méthode 		

		<ul style="list-style-type: none"> • Appel d'une méthode • Méthodes de classe • Fonctions imbriquées 		
B. Connaître les principaux piliers de la POO	B.1 – Définir l'héritage	<ul style="list-style-type: none"> • Principe de l'héritage • Types d'héritage • Surcharge des constructeurs et chaînage • Visibilité des attributs et des méthodes de la classe fille 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours) • Travaux pratiques sur papier/logiciels/... • Projet de synthèse <ul style="list-style-type: none"> - Finalisation de la conception du projet • QCM 	15 %
	B.2 - Définir le polymorphisme	<ul style="list-style-type: none"> • Principe du polymorphisme • Redéfinition des méthodes • Surcharge des méthodes • Surcharge des opérateurs 		
	B.3 – Caractériser l'abstraction	<ul style="list-style-type: none"> • Principes • Classes abstraites • Méthodes abstraites 		

	B.4 – Manipuler les interfaces	<ul style="list-style-type: none"> Définition des interfaces Utilité des interfaces Implémentation des interfaces 		
C. Coder des solutions orientées objet	C.1 – Coder une solution orientée objet	<ul style="list-style-type: none"> Création d'un package Codage d'une classe Intégration des concepts POO <ul style="list-style-type: none"> Encapsulation Héritage Polymorphisme Abstraction Interfaces 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours) Travaux pratiques sur ordinateur Utilisation d'un compilateur Projet de synthèse <ul style="list-style-type: none"> Codage Test 	45%
	C.2 - Manipuler les données	<ul style="list-style-type: none"> Collections Listes Fichiers 		
	C.3 – Utiliser les expressions régulières	<ul style="list-style-type: none"> Création des expressions régulières Manipulation des expressions régulières 		

	C.4 – Administrer les exceptions	<ul style="list-style-type: none"> Types d’erreur et expérimentation Types des exceptions Gestion des exceptions <ul style="list-style-type: none"> - Création/Déclenchement - Analyse de la traçabilité 		
D. Manipuler les modules et les bibliothèques	D.1 – Manipuler les modules	<ul style="list-style-type: none"> Création des modules Importation des modules 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l’aide des supports fournis par le formateur (polycop, documents, cours) Travaux pratiques sur ordinateur Utilisation d’un compilateur Projet de synthèse <ul style="list-style-type: none"> - Codage des modules - Intégration des bibliothèques standards - Intégration d’une bibliothèque graphique - Manipulation d’interface graphique - Déploiement et documentation 	35 %
	D.2 – Manipuler les bibliothèques	<ul style="list-style-type: none"> Installation des bibliothèques externes (pip) Bibliothèques standards Bibliothèques graphiques (Tkinter, Tix, wxPython, pyqt, ...) Création des bibliothèques Importation des bibliothèques 		
	D.3 – Réaliser des interfaces graphiques	<ul style="list-style-type: none"> Installation de bibliothèque Création d’éléments graphiques (widgets) : <ul style="list-style-type: none"> ○ Fenêtre ○ Bouton 		

		<ul style="list-style-type: none"> ○ Label ○ Zone de saisie ○ Case à cocher ○ Bouton radio ○ Listes ○ Canvas ○ Scale ○ Frames ○ PanedWindow ○ Spinbox ○ Alertes (MessageBox) ○ Menus • Manipulation des méthodes/événements et options des widgets 		
--	--	---	--	--

Fiche prescrite

Compétence 4 : « Développer des sites Web statiques »	
Code de la compétence : DIA_DEV_TS-04	Durée : 110 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et / ou en groupe À partir : <ul style="list-style-type: none"> De consignes De spécifications fonctionnelles De spécifications techniques D'un cahier des charges À l'aide : <ul style="list-style-type: none"> D'éditeur de texte IDE (Vs Code) Framework (Bootstrap 5) Serveur (local, en ligne) 	<ul style="list-style-type: none"> Compréhension facile de HTML et CSS3 Bonne maîtrise des outils pour la conception et la réalisation des pages web Utilisation aisée de Bootstrap Réalisation facile d'une page web Responsive avec Bootstrap
Éléments de la compétence	Critères particuliers de performance
A. Créer une page web en HTML	<ul style="list-style-type: none"> Manipulation adaptée des outils de création d'une page web Connaissance juste des éléments de base d'une page web
B. Implémenter une page web statique	<ul style="list-style-type: none"> Connaissance exacte du fonctionnement d'une page web statique Réalisation correcte d'un site web statique Intégration appropriée d'un formulaire dans une page web

C. Mettre en forme une page web avec les feuilles de style CSS	<ul style="list-style-type: none">• Bonne maîtrise du CSS pour la mise en forme d'une page web• Utilisation correcte des positionnements pour l'organisation d'une page web avec CSS• Ajout approprié des animations• Capacité d'adaptation des templates HTML/CSS à un site web statique
D. Maîtriser Bootstrap	<ul style="list-style-type: none">• Bonne maîtrise des différentes composantes de Bootstrap• Production adaptée d'une page web avec Bootstrap• Bonne maîtrise des classes CSS de Bootstrap• Production aisée d'une page web responsive
E. Héberger un site	<ul style="list-style-type: none">• Hébergement correct d'un site web statique en intranet• Hébergement correct d'un site web statique en ligne• Maîtrise de transfert de fichiers (FTP)

Suggestions Pédagogiques

Compétence 4 :	« Développer des sites Web statiques »	Code : DIA_DEV_TS-04
DURÉE : « 110 h »	Compétences Préalables : Compétence 1/2	
Type de compétences : Spécifique	Compétences en parallèles : Compétence 3	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Créer une page web en HTML	A.1 - Introduction au langage HTML	<ul style="list-style-type: none"> Historique de l'HTML Norme W3C Propriétés des navigateurs 	<ul style="list-style-type: none"> Seul ou en groupe Cours et énoncés des travaux pratiques fournis par le formateur 	20 %
	A.2 - Utiliser l'environnement de développement pour produire du HTML	<ul style="list-style-type: none"> Utilisation de Visual Studio Code Installation des extensions VS Code 	<ul style="list-style-type: none"> Travaux pratiques sur la manipulation de Visual Studio Code 	
	A.3 - Définir les éléments basiques d'une page HTML	<ul style="list-style-type: none"> Norme W3C Squelette d'une page HTML Balises de bases (contenu) 	<ul style="list-style-type: none"> Travaux pratiques sur la réalisation d'une page web statique simple Projet de synthèse 	

		<ul style="list-style-type: none"> • Couleurs • Listes en HTML • Types de chemin d'accès • Liens en HTML • Balises Multimédia (Images, Audio, Vidéo) • Tableaux avec du HTML • Balises de structuration (Header, NAV, Article, Section, Footer, Div, SPAN) 	<ul style="list-style-type: none"> - Analyse de la conception d'un site web existant - Structuration des pages HTML - Intégration du contenu Multimédia - Navigation entre les pages 	
B. Implémenter une page web statique	B.1 - Réaliser une page web statique	<ul style="list-style-type: none"> • Fonctionnement d'une page web statique • Réalisation d'un site web statique avec HTML5 	<ul style="list-style-type: none"> • Seul ou en groupe • Cours et énoncés des travaux pratiques fournis par le formateur • Travaux pratiques sur la réalisation des pages web avec HTML5 • Projet de synthèse - Ajout des formulaires aux pages web 	20 %
	B.2 - Intégrer des formulaires dans une page web	<ul style="list-style-type: none"> • Balise <form> • Champs de saisie (Input, TextArea) • Balises de choix (RadioButton, CheckBox, Select, • Balises HTML5 pour les formulaires - Validation automatique - Boutons 		

C. Mettre en forme une page web avec les feuilles de style CSS	C.1 - Introduire le CSS	<ul style="list-style-type: none"> • Présentation du CSS • Codification des couleurs • Unités de mesure • Positions (center, left, right) • Fonts • Types d'intégration du CSS • Sélecteurs simples (element, class, id) • Sélecteurs complexes • Pseudo classes 	<ul style="list-style-type: none"> • Seul ou en groupe • Cours et énoncés des travaux pratiques fournis par le formateur • Travaux pratiques sur la réalisation d'un formulaire avec HTML5 • Projet de synthèse 	30%
	C.2 - Utiliser les propriétés CSS	<ul style="list-style-type: none"> • Typographie • Bordures et ombres • Marges et padding • Images • Couleurs de fond • Backgroud 	<ul style="list-style-type: none"> - Mise en forme des pages avec CSS - Création des animations avec CSS 	
	C.3 - Manipuler le positionnement	<ul style="list-style-type: none"> • Types de positionnement (relatif, absolu, float, ...) • Utilisation des blocs Flex (FlexBox) 		
	C.4 - Adapter une page web au dispositif d'affichage	<ul style="list-style-type: none"> • Introduction au Responsive Design • Media Queries 		

	C.5 - Créer des animations	<ul style="list-style-type: none"> Manipulation des propriétés d'animation Types d'animation (Transition, Transformation, ...) Utilisation des keyFrames 		
	C.6 - Adapter des templates HTML/CSS avec un site Web	<ul style="list-style-type: none"> Analyse de la structure d'une template Adaptation du contenu d'un site selon la charte 		
D. Maîtriser Bootstrap	D.1 - Intégrer Bootstrap	<ul style="list-style-type: none"> Introduction au Bootstrap (versions, avantages, ...) Système de Grid 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur Travaux pratiques sur la découverte de Bootstrap Apprentissage par projet (réalisation d'un site web avec Bootstrap) 	25 %
	D.2 - Maîtriser les classes CSS de base	<ul style="list-style-type: none"> Typographie et liens Tableaux Éléments de formulaires et contrôles supportés Boutons 		
	D.3 – Utiliser les composants Bootstrap	<ul style="list-style-type: none"> Barres de navigation Menus déroulants Pagination, Badges et Alertes Barres de progression Cartes (Cards) 		

E. Héberger un site	E.1 – Déployer en intranet	<ul style="list-style-type: none"> • Définition d'un réseau local • Installation d'un serveur Web • Protocoles de transfert de fichiers • Création des comptes FTP • Transfert des fichiers • Mise à jour du contenu 	<ul style="list-style-type: none"> • Seul ou en groupe • Cours et énoncés des travaux pratiques fournis par le formateur • Travaux pratiques sur l'hébergement d'un site web statique 	5%
	E.2 – Déployer en ligne	<ul style="list-style-type: none"> • Critères de choix d'un hébergeur • Hébergement (gratuit, payant) 	<ul style="list-style-type: none"> • Projet de synthèse - Hébergement du site web 	

Fiche prescrite

Compétence 5 : « Programmer en Javascript »	
Code de la compétence : DIA_DEV_TS-05	Durée : 110 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et / ou en groupe À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales De spécifications fonctionnelles De spécifications techniques D'un cahier des charges À l'aide : <ul style="list-style-type: none"> D'environnement intégré de développement IDE (VS Code) Des librairies JS Bootstrap 5 	<ul style="list-style-type: none"> Ciblage clair de la nature et du besoin de JavaScript Identification précise des données d'entrée, de traitement et de sortie Application adéquate des différents traitements possibles en code JavaScript Écriture, lecture et manipulation facile du code en JavaScript Utilisation adaptée des librairies des bibliothèques JQuery et AJAX
Éléments de la compétence	Critères particuliers de performance
A. Définir le rôle de JavaScript dans le Développement	<ul style="list-style-type: none"> Reconnaissance judicieuse de la différence entre un langage compilé et un langage interprété Bonne maîtrise de l'aspect dynamique offert par JavaScript dans le développement web. Utilisation aisée des outils et des environnements de travail
B. Acquérir les fondamentaux de JavaScript	<ul style="list-style-type: none"> Manipulation correcte de la syntaxe de Javascript Utilisation correcte des structures logiques Reconnaissance précise des tâches répétitives et des tâches conditionnelles Bonne maîtrise des notions de fonctions et de

	<p>méthodes</p> <ul style="list-style-type: none">• Bonne manipulation des objets en Javascript
C. Manipuler les éléments d'une page avec DOM	<ul style="list-style-type: none">• Bonne maîtrise de l'interaction avec le DOM• Manipulation correcte des éléments HTML (Création, modification, suppression)
D. Gérer les événements utilisateur	<ul style="list-style-type: none">• Bonne gestion de l'interactivité• Bonne gestion des éléments d'un formulaire
E. Manipuler jQuery	<ul style="list-style-type: none">• Utilisation correcte de la bibliothèque JQuery• Compréhension de la bibliothèque AJAX

Suggestions Pédagogiques

Compétence 5 :	« Programmer en Javascript »	Code : DIA_DEV_TS-05
DURÉE : « 110 h »	Compétences Préalables : Compétences 1/2/3/4	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 6	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Définir le rôle de JavaScript dans le Développement	A.1 - Comparer un langage de script avec un langage compilé	<ul style="list-style-type: none"> Langage compilé Langage de script (javascript) 	<ul style="list-style-type: none"> Seul ou en groupe Quiz sur les langages compilés et les langages de script Quiz sur le fonctionnement d'une architecture client/serveur Exercices d'application sur l'utilisation de l'environnement de développement VS 	10 %
	A.2 - Comprendre l'architecture client/serveur	<ul style="list-style-type: none"> Composition d'une architecture client/serveur Fonctionnement d'un système client/serveur pour le cas d'une architecture Web 		

	A.3 – Découvrir l'écosystème de développement	<ul style="list-style-type: none"> • Environnements de développement • Découverte des librairies appropriés (jQuery, React, Vue JS, Angular, ...) 	Code <ul style="list-style-type: none"> • Quiz sur les librairies JavaScript 	
B. Acquérir les fondamentaux de JavaScript	B.1 – Maîtriser la syntaxe JavaScript et ses notions fondamentales	<ul style="list-style-type: none"> • Notions de variables et de données • Expressions et opérateurs • Notions de lecture et d'écriture • Types primitifs et objets de base 	<ul style="list-style-type: none"> • Seul ou en groupe • Exercices d'application sur les notions fondamentales de Javascript • Cours et polycop fournis par le formateur • Travaux pratiques sur les fonctions • Exercices d'application sur la création de tableaux et d'objets • Projet de synthèse <ul style="list-style-type: none"> - Etude du projet 	40 %
	B.2 – Maîtriser les structures de contrôle	<ul style="list-style-type: none"> • Structures alternatives • Structures itératives 		
	B.3 - Utiliser des fonctions	<ul style="list-style-type: none"> • Fonctions • Expressions lambdas • Gestion des exceptions 		
	B.4 – Manipuler les objets	<ul style="list-style-type: none"> • Création d'objet • Manipulation d'objet • Manipulation des objets natifs - Tableaux (insertion, suppression, modification, balayage des éléments, tri, recherche, ...) 		

		<ul style="list-style-type: none"> - Date - Math - RegEx • Manipulation JSON 		
C. Manipuler les éléments d'une page avec DOM	C.1 – Comprendre l'arbre DOM, les nœuds parents et enfants	<ul style="list-style-type: none"> • Arbre DOM • Objet Document • Navigation dans le DOM (parentNode, childNodes, ...) 	<ul style="list-style-type: none"> • Seul ou en groupe • Quiz sur l'arbre DOM • Exercice d'application sur les modes d'accès aux éléments • Travaux pratiques sur ordinateur pour la manipulation des éléments HTML de la page • Projet de synthèse - Manipulation du DOM 	25%
	C.2. – Connaître les bases de la manipulation du DOM en JavaScript	<ul style="list-style-type: none"> • Sélecteurs (simples, multiples...) • Modes d'Accès aux éléments 		
	C.3 - Manipuler les éléments HTML	<ul style="list-style-type: none"> • Manipulation des éléments (Création, modification, suppression) • Mise à jour des styles, attributs et classes • Création DOMMenuObject 		

D. Gérer les événements utilisateur	D.1 – Comprendre la notion d'évènement pour gérer l'interactivité	<ul style="list-style-type: none"> • Définition d'un évènement • Méthode addEventListener • MouseEvents • Interaction avec le clavier 	<ul style="list-style-type: none"> • Seul et/ou en groupe • Exercice d'application sur la création d'évènements • Travaux pratiques sur ordinateur pour la gestion des éléments d'un formulaire • Projet de synthèse <ul style="list-style-type: none"> - Utilisation des formulaires - Ajout de l'interactivité des formulaires - Manipulation de la mise en forme avec JavaScript 	10 %
	D.2 – Gérer les éléments d'un formulaire	<ul style="list-style-type: none"> • Soumission d'un formulaire • Interruption d'un formulaire • Validation d'un formulaire 		
E. Manipuler jQuery	E.1 - Découvrir jQuery	<ul style="list-style-type: none"> • Fonctions essentielles et chaînage • Comportement des liens • Association d'évènements et déclenchement • Intégration de plugins existants • Utilisation de plugins existants 	<ul style="list-style-type: none"> • Seul et/ou en groupe • Travaux pratiques sur ordinateur pour l'utilisation de jQuery • Travaux pratiques sur ordinateur pour la découverte d'Express JS • Projet de synthèse 	15%



	E.2 – Découvrir AJAX	<ul style="list-style-type: none">• Introduction à AJAX• Fonctionnement d'AJAX	- Intégration des librairies	
--	----------------------	---	------------------------------	--

Fiche prescrite

Compétence 6 : « Manipuler des bases de données »	
Code de la compétence : DIA_DEV_TS-06	Durée : 100 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et / ou en groupe À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales De spécifications fonctionnelles De spécifications techniques D'un cahier des charges À l'aide : <ul style="list-style-type: none"> Editeur de texte Environnement de développement (MySQL Workbench, AnalyseSI, Power AMC, Mocodo...) 	<ul style="list-style-type: none"> Bonne modélisation des données Maitrise de gestion d'une base de données (création, mise à jour et interrogation) Administration appropriée de la base de données
Éléments de la compétence	Critères particuliers de performance
A. Concevoir une base de données	<ul style="list-style-type: none"> Capacité d'analyse d'un cahier de charges Construction correcte du graphe de dépendances fonctionnelles Bonne connaissance des règles de passage du graphe au Modèle Conceptuel de Données Bonne maitrise des formes normales Capacité de passage du MCD au MLD Normalisé
B. Préparer l'environnement	<ul style="list-style-type: none"> Bonne utilisation de l'outil de modélisation Capacité d'Installation d'un serveur MySQL Bonne configuration des services MySQL

C. Manipuler les données	<ul style="list-style-type: none">• Maîtrise des étapes de création d'une base de données• Bonne manipulation des différents types de requêtes SQL• Utilisation adéquate des jointures• Capacité d'importation et exportation des données• Bonne gestion des privilèges de base
--------------------------	---

Suggestions Pédagogiques

Compétence 6 :	« Manipuler des bases de données »	Code : DIA_DEV_TS-06
DURÉE : « 100 h »	Compétences Préalables : Compétences 1/2	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 5	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A– Concevoir une base de données	A.1 - Analyser le cahier des charges	<ul style="list-style-type: none"> Lecture d'un cahier des charges Description des limites du projet Analyse des données et des traitements de la situation présentée 	<ul style="list-style-type: none"> Seul et/ou en groupe Exercices d'application sur la modélisation d'une base de données 	40 %
	A.2 - Modéliser les données	<ul style="list-style-type: none"> Contraintes déduites des règles de gestion Dictionnaire de données Construction du graphe de dépendances fonctionnelles Règles de passage du graphe au Modèle Conceptuel de Données Construction du Modèle Conceptuel de Données 	<ul style="list-style-type: none"> Projet de synthèse <ul style="list-style-type: none"> - Modèle MCD - Modèle MLD 	

	A.3 - Normaliser les données	<ul style="list-style-type: none"> Formes Normales Règles de passage du MCD au MLD Normalisé 		
B. Préparer l'environnement	B.1 - Exploiter un outil de modélisation	<ul style="list-style-type: none"> Procédure d'installation d'un outil de modélisation Utilisation de l'outil de modélisation 	<ul style="list-style-type: none"> Seul et/ou en groupe Exercices d'exploitation de l'outil de modélisation Exercices d'exploitation de l'environnement 	5 %
	B.2 – Préparer le serveur MySQL	<ul style="list-style-type: none"> Installation de Workbench Management des services MySQL Configuration des ports MySQL 		
C. Manipuler les données	C.1 - Créer une base de données	<ul style="list-style-type: none"> Création d'une base de données Choix de moteur Création des tables Contraintes d'intégrité sur les tables (intégrité référentielle, contrainte de domaine CHECK) Manipulation d'objet table (DROP, ALTER) Création des colonnes Typage des colonnes Intégration des contraintes d'intégrité sur les colonnes (Primary key, Foreign key, not null, RegEx P, ...) 	<ul style="list-style-type: none"> Seul et/ou en groupe Exercices d'application sur la création et la manipulation des bases de données Exercices sur l'administration de la base de données Projet de synthèse <ul style="list-style-type: none"> Générer la base de données sur la base du MLD 	55%

	C.2 - Réaliser des requêtes SQL	<ul style="list-style-type: none"> • Requêtes CRUD <ul style="list-style-type: none"> - INSERT - UPDATE - DELETE • Requêtes de sélection • Expressions du SGBD • Fonctions d'agrégation du SGBD • Sous requêtes • Requêtes de l'union • Jointures <ul style="list-style-type: none"> - Equi-jointures, - Jointures externes (right, left) - Auto-jointures 	<ul style="list-style-type: none"> - Insertion/Importation des données - Manipulation des données - Exportation de la base de données - Gestion des privilèges 	
	C.3 – Administrer une base de données	<ul style="list-style-type: none"> • Backup • Importation • Exportation • Commandes de création des comptes utilisateurs • Commandes de gestion des privilèges de base 		

Fiche prescrite

Compétence 7 : « Développer des sites web dynamiques »	
Code de la compétence : DIA_DEV_TS-07	Durée : 120 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et/ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De base documentaire • À l'aide : <ul style="list-style-type: none"> - Editeur de texte - Du réseau professionnel - VS Code - XAMP 	<ul style="list-style-type: none"> • Maîtrise de l'architecture client/serveur • Maîtrise de l'environnement de développement PHP/MySQL • Bonne maîtrise du langage PHP • Bonne manipulation des données • Sécurisation des données • Connaissance de l'architecture MVC
Éléments de la compétence	Critères particuliers de performance
A. Introduire la notion client/serveur	<ul style="list-style-type: none"> • Idée claire sur les types de serveurs web • Bonne connaissance de l'architecture client/serveur • Maîtrise de l'environnement de développement
B. Programmer en PHP	<ul style="list-style-type: none"> • Bonne maîtrise du langage PHP • Bonne manipulation des tableaux • Bonne manipulation des fichiers • Maîtrise du concept de l'orienté objet en PHP • Manipulation adéquate des méthodes magiques en PHP

C. Manipuler les données	<ul style="list-style-type: none">• Interrogation correcte d'une base de données à travers un formulaire• Bonne manipulation des données CRUD• Utilisation correcte des sessions et des cookies• Sécurisation des données
D. Découvrir l'architecture MVC	<ul style="list-style-type: none">• Connaissance du modèle MVC avec ses différentes couches• Idée claire sur les Web services

Suggestions Pédagogiques

Compétence 7 :	« Développer des sites web dynamiques »	Code : DIA_DEV_TS-07
DURÉE : « 120 h »	Compétences Préalables : Compétences 1/2/3/4/5/6	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 8	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Introduire la notion client/serveur	A.1 – Approfondir la notion client/serveur	<ul style="list-style-type: none"> Architectures 2-tiers Principe Client/serveur Types de serveurs web 	<ul style="list-style-type: none"> Seul et/ou en groupe Ateliers des installations et configuration <ul style="list-style-type: none"> Serveurs web Ports et Services web Extensions VS Code Activation des extensions PHP (PDO, ...) Quiz 	10%
	A.2 – Introduire le langage PHP	<ul style="list-style-type: none"> Définition du PHP Généralités sur les fonctionnalités du PHP 		
	A.3 - Préparer l'environnement de développement	<ul style="list-style-type: none"> Choix des serveurs web Installation du serveur (Apache) Configuration du serveur (ports, services, ...) Installation des extensions nécessaires (VS Code) 		

B. Programmer en PHP	B.1 – Maitriser le langage PHP	<ul style="list-style-type: none"> • Structure générale d'un script PHP • Manipulation des Variables/constantes/Affectation • Manipulation des types de données • Instructions de sortie • Contrôles de flux et boucles • Formulaires simples • Transmission de variables (GET, POST) • Variables d'environnement (\$HTTP_HOST, \$SERVER_ADDR, \$SERVER_NAME, ...) • Redirection entre pages • Fonctions sur les chaines de caractères et les dates 	<ul style="list-style-type: none"> • Seul et/ou en groupe • Exercices d'application en PHP • Projet de synthèse <ul style="list-style-type: none"> - Création des scripts - Création des classes - Instanciation des objets - Création des formulaires - Traitement des données d'un formulaire 	45%
	B.2 – Traiter les données en PHP	<ul style="list-style-type: none"> • Traitement des tableaux (simple, Associatif) • Manipulation de fichier (chargement, Suppression, téléchargement) 		
	B.3 - Utiliser l'orientée objet en PHP	<ul style="list-style-type: none"> • Intérêt de programmer en Orienté Objet en PHP • Application des concepts de base de la programmation orientée objet en PHP 		

		<ul style="list-style-type: none"> Utilisation des méthodes magiques en PHP (<code>__construct()</code>, <code>__destruct()</code>, <code>__call()</code>, <code>__unset()</code>, <code>__sleep()</code>, <code>__wakeup()</code>, <code>__serialize()</code>, <code>__toString()</code>, ...) 		
C. Manipuler les données	C.1 - Ecrire des scripts d'accès aux données	<ul style="list-style-type: none"> Connexion à une base de données MySQL avec PDO Interrogation d'une base de données à travers un formulaire Récupération des résultats Manipulation des données CRUD 	<ul style="list-style-type: none"> Seul et/ou en groupe Exercices d'application en PHP et gestion de base de données Projet de synthèse <ul style="list-style-type: none"> Création de la base de données MySQL Connexion à la base de données Gestion des données (CRUD) Sécurisation des données 	35%
	C.2 – Sécuriser les données	<ul style="list-style-type: none"> Utilisation des sessions et des cookies Sécurisation des données (cryptage, injections SQL, ...) 		
D. Découvrir l'architecture MVC	D.1 - Comprendre l'organisation globale de l'architecture MVC	<ul style="list-style-type: none"> Présentation de l'MVC (Modèles, Vues, Contrôleurs) Intérêt du modèle MVC Rôles de chacune des 3 entités du modèle MVC Interactions entre les 3 entités du modèle 	<ul style="list-style-type: none"> Seul et/ou en groupe Exercices d'application <ul style="list-style-type: none"> Architecture MVC Web services 	10%



		MVC		
	D.2 – Découvrir les Web services et les API REST	<ul style="list-style-type: none">• Définition des Web services• Présentation des protocoles SOAP et REST		

Compétence 8 : « S'initier à la sécurité des systèmes d'information »	
Code de la compétence : DIA_DEV_TS-08	Durée : 45 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et/ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De base documentaire • À l'aide : <ul style="list-style-type: none"> - Editeur de texte 	<ul style="list-style-type: none"> • Bonne maîtrise de la notion de sécurité informatique • Bonne compréhension des types d'attaques des systèmes informatiques • Bonne connaissance des techniques de protection • Disposition de la confidentialité des données clients • Disposition de la protection des données utilisateurs • Bonne sécurisation d'un service Web • Bonne utilisation des firewalls
Éléments de la compétence	Critères particuliers de performance
A. Introduire la sécurité informatique	<ul style="list-style-type: none"> • Maîtrise de la notion de sécurité informatique • Bonne connaissance des types de virus • Bonne appréhension des droits d'auteur (Copyright) • Bonne compréhension des types d'attaques des systèmes informatiques • Bonne connaissance des techniques de protection • Bonne exploitation des Antivirus • Bonne utilisation des firewalls

B. Assurer la confidentialité des données	<ul style="list-style-type: none">• Idée claire sur les données privées• Bonne connaissance des lois sur la protection des données personnelles• Maîtrise des règles de protection des données utilisateurs• Bonne connaissance des droits d'accès des utilisateurs d'une application• Compréhension des méthodes de cryptage des données
C. Protéger les applications Web	<ul style="list-style-type: none">• Bonne gestion de la politique d'utilisation des mots de passe• Gestion appropriée des rôles d'utilisateurs• Connaissances en types de chiffrements• Compréhension du fonctionnement d'un firewall dans la protection d'application HTTP

Suggestions Pédagogiques

Compétence 8 :	« S’initier à la sécurité des systèmes d’information »	Code : DIA_DEV_TS-08
DURÉE : « 45 h »	Compétences Préalables : Compétences 1/2/3/4/5/6	
Type de compétences : Transversale	Compétences en parallèles : Compétences 7	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D’APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Introduire la sécurité informatique	A.1 – Introduire la notion de sécurité	<ul style="list-style-type: none"> • Introduction à la notion de sécurité informatique • Introduction aux types de failles et attaques • Types de virus • Notion de Copyright 	<ul style="list-style-type: none"> • Seul • Quiz sur les notions de base de la sécurité informatique 	20 %
	A.2 – Comprendre les types d’attaques des systèmes informatiques	<ul style="list-style-type: none"> • Vulnérabilités des applications Web • Attaques " Cross Site Scripting " ou XSS • Attaques sur les sessions (cookie poisoning, session hijacking, ...). • Exploitation de vulnérabilités sur le frontal HTTP (ver Nimda, faille Unicode, ...) • Attaques sur les configurations standards 		

		(Default Password, Directory Transversal, ...)		
		<ul style="list-style-type: none"> • Attaques d'hameçonnage (fishing) • Attaques DDOS 		
	A.3 – Se prémunir d'une quelconque tentative de piratage	<ul style="list-style-type: none"> • Techniques de protection • Notion des Antivirus • Notion des firewalls 		
B. Assurer la confidentialité des données	B.1 – Confidentialiser les données clients	<ul style="list-style-type: none"> • Définition des données privées • Initiation à la protection des données <ul style="list-style-type: none"> - Loi 09-08 (CNDP) - Loi 50.20 (CNDP) - Droits d'auteur (Copyright) • Mécanismes de protection des données privées 	<ul style="list-style-type: none"> • Seul • Quiz sur la confidentialité des données 	40 %
	B.2 – Protéger les données utilisateurs	<ul style="list-style-type: none"> • Obfuscation du code source • Règles de protection des données utilisateurs • Droits d'accès des utilisateurs d'une application • Accès sécurisé aux données de l'application • Cryptage des données 		

C. Protéger les applications Web	D.1 - Sécuriser un service	<ul style="list-style-type: none"> • Gestion de la politique d'utilisation des mots de passe • Gestion des rôles d'utilisateurs • Chiffrements symétriques et asymétrique • Notion de private/public Key 	<ul style="list-style-type: none"> • Seul • Quiz sur la protection des applications Web 	40 %
	D.2 – Utiliser les firewalls	<ul style="list-style-type: none"> • Fonctionnement d'un firewall réseau dans la protection d'application HTTP • Principe du développement sécurisé • Firewall "applicatif " 		

Compétence 9 : « Préparer un projet web »	
Code de la compétence : DIA_DEVOWFS_TS-09	Durée : 60 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et / ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De spécifications techniques - De cahier des charges • À l'aide : <ul style="list-style-type: none"> - D'un logiciel de modélisation UML STAR UML - De logiciels : <ul style="list-style-type: none"> ▪ Applications d'UI/UX adobe \ Figma 	<ul style="list-style-type: none"> • Bonne compréhension du cycle de vie d'un projet web ; • Bonne maîtrise de la modélisation d'un projet web en utilisant les différents diagrammes UML ; • Création d'une maquette fonctionnelle pour un projet Web ; • Préparation adéquate de l'environnement pour le développement Web.
Éléments de la compétence	Critères particuliers de performance
A. Modéliser un projet web	<ul style="list-style-type: none"> • Bonne appréhension du cycle de vie d'un projet web ; • Maîtrise du diagramme de cas d'utilisation ; • Bonne saisie du diagramme de classes ; • Utilisation adéquate du diagramme de séquence ; • Bonne utilisation d'un outil de modélisation des diagrammes UML.
B. Représenter la vue dynamique d'un système	<ul style="list-style-type: none"> • Bonne description des changements d'états d'objets à l'aide d'un diagramme d'état-transition ; • Bonne description du comportement d'un système à l'aide d'un diagramme d'activités.
C. Créer une maquette pour le développement web	<ul style="list-style-type: none"> • Bonne compréhension des principes de la conception de l'expérience et l'interface utilisateur ;

	<ul style="list-style-type: none">• Structuration appropriée d'un wireframe sur papier ;• Représentation correcte des wireframes et maquettes en utilisant les fonctionnalités basiques de Figma.
D. Préparer l'environnement de développement web	<ul style="list-style-type: none">• Bonne compréhension d'un Environnement de développement intégré (IDE) ;• Bonne connaissance des différents langages Web ;• Description générale des bibliothèques Web ;• Bonne connaissance des serveurs de bases de données ;• Bonne distinction des différents serveurs web ;• Choix adéquat des Frameworks de développements Frontend et Backend.

Suggestions Pédagogiques

Compétence 9 :	Préparer un projet web	Code : DIA_DEVOWFS_TS-09
DURÉE : « 60 h »	Compétences Préalables : Compétences 1/2/3/7	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 10/11/12/13/14	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Modéliser un projet web	A.1- Appréhender le cycle de vie d'un projet web	<ul style="list-style-type: none"> • Cycle de vie d'un projet (Définition, étapes...) • Analyse des besoins • Conception • Développement • Tests et déploiement • Maintenance et évolutivité 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Modélisation du projet en 	45%
	A.2- Modéliser les besoins client par un diagramme de cas d'utilisation	<ul style="list-style-type: none"> • Introduction au langage de modélisation UML • Définition du diagramme des cas d'utilisation • Acteurs • Cas d'utilisation • Relation entre acteurs et cas d'utilisation : <ul style="list-style-type: none"> - Relation d'association 		

		<ul style="list-style-type: none"> - Multiplicité • Relations entre cas d'utilisation : <ul style="list-style-type: none"> - Relation d'inclusion - Relation d'extension - Point d'extension • Relation de généralisation ou de spécialisation • Description textuelle des cas d'utilisation : <ul style="list-style-type: none"> - Identification - Description des scénarios - Exigence non fonctionnelle 	<p>utilisant les diagrammes de cas d'utilisation, de classes et de cas d'utilisation</p> <ul style="list-style-type: none"> - Elaboration des diagrammes UML à l'aide d'un outil de modélisation - Proposition de la démarche de la solution 	
	A.3- Modéliser les données du projet par un diagramme de classes	<ul style="list-style-type: none"> • Définition du diagramme de classes • Visibilité, encapsulation des méthodes et attributs • Attributs et méthodes d'instance • Attributs et méthodes de classe • Contraintes • Attributs calculés (dérivé) • Multiplicité (cardinalité) • Constructeur et destructeurs • Modèles de classe • Relations entre classes : 		

		<ul style="list-style-type: none"> - Relation de dépendance - Associations reliant plusieurs classes <ul style="list-style-type: none"> o Association n-aire o Classe association - Associations particulières <ul style="list-style-type: none"> o Composition o Agrégation - Relation d'héritage • Classes concrètes et abstraites • Relation de réalisation et notion d'interfaces 		
	A.4- Modéliser les interactions d'objets d'un système à l'aide d'un diagramme de séquence	<ul style="list-style-type: none"> • Rôle du diagramme de séquences • Distinguer le diagramme de séquence boîte noire du diagramme de séquence boîte blanche • Diagramme de séquence comme boîte blanche • Délimitation du diagramme de séquence : <ul style="list-style-type: none"> - L'objet - La ligne de vie - Les messages • Types de messages : <ul style="list-style-type: none"> - Message synchrone 		

		<ul style="list-style-type: none"> - Message asynchrone • Fragments d'interactions combinés : <ul style="list-style-type: none"> - Fragment d'interaction avec l'opérateur d'option « opt » - Fragment d'interaction avec l'opérateur d'alternative « alt » - Fragment d'interaction avec l'opérateur de boucle « loop » - Fragment d'interaction avec l'opérateur Parallèle « par » - Fragment d'interaction avec l'opérateur d'interruption « break » - Fragment d'interaction avec l'opérateur d'omission « ignore » - Fragment d'interaction avec l'opérateur d'affirmation « assert » 		
	A.5- Elaborer des diagrammes UML à l'aide d'un outil de modélisation	<ul style="list-style-type: none"> • Environnement de travail et fonctionnalités • Création et édition d'un diagramme de classe • Sauvegarde et impression • Génération du code source 		

		<ul style="list-style-type: none"> Création et édition d'un diagramme des cas d'utilisation Création et édition d'un diagramme de séquences 		
B. Représenter la vue dynamique d'un système	B.1- Décrire les changements d'états d'objets à l'aide d'un diagramme d'état transition	<ul style="list-style-type: none"> Rôle du diagramme états-transitions Etat Evénement Transition externe Transition interne Action et activité Point de choix État composite État historique 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap 	20%
	B.2- Décrire le comportement d'un système à l'aide d'un diagramme d'activités	<ul style="list-style-type: none"> Rôle du diagramme d'activités Action Activité Nœuds d'activités Transitions Nœuds d'actions (action node) Nœuds d'objet (object node) et les flots d'objet Pins d'entrée/sortie 	<ul style="list-style-type: none"> Projet de synthèse <ul style="list-style-type: none"> Modélisation du projet en utilisant les diagrammes d'état et d'activités Elaboration des diagrammes UML à l'aide d'un outil de modélisation 	

		<ul style="list-style-type: none"> Nœuds de stockage de données (data store node) Partitions ou couloirs d'activités (travées – swinlanes) 	<ul style="list-style-type: none"> Proposition de la démarche de la solution 	
C. Créer une maquette pour le développement web	C.1- Comprendre les principes de la conception de l'expérience et l'interface utilisateur	<ul style="list-style-type: none"> Principes de l'UX Design : <ul style="list-style-type: none"> Eléments de l'expérience utilisateur (persona, usabilité, affordance...) Conception visuelle (choix des polices et couleurs) Pratiques de prototypage Principes de l'UI Design : <ul style="list-style-type: none"> Critères ergonomiques du web Chartes d'interface Principes de navigation et d'interaction Wireframes Mockups Prototypes 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap Projet de synthèse <ul style="list-style-type: none"> Structuration des wireframes du projet sur papier Construction des wireframes avec Figma Maquettage de l'application 	25%
	C.2- Structurer un wireframe sur papier	<ul style="list-style-type: none"> Importance du maquettage dans le développement web Structure d'un wireframe sur papier 		

	C.3- Construire un wireframe avec Figma	<ul style="list-style-type: none"> • Prise en main de Figma • Adaptation de wireframe aux différents supports 	avec Figma - Proposition de la démarche de la solution	
	C.4- Créer une première maquette avec Figma	<ul style="list-style-type: none"> • Création d'un univers visuel • Création d'une maquette avec Figma 		
D. Préparer l'environnement de développement web	D.1- Appréhender l'environnement de développement web	<ul style="list-style-type: none"> • Comparaison entre les différentes technologies web • Environnement de développement intégré (IDE) • Langages • Bibliothèques • Serveurs de bases de données • Serveurs web 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Préparation de l'environnement de développement 	10%
	D.2- Choisir les Frameworks de développement web	<ul style="list-style-type: none"> • Frameworks Frontend • Frameworks Backend 		



			<ul style="list-style-type: none">- Préparation des Frameworks de développement Frontend et Backend- Proposition de la démarche de la solution	
--	--	--	---	--

Compétence 10 : « Adopter l'approche agile »	
Code de la compétence : DIA_DEVOWFS_TS-10	Durée : 100 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et/ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De base documentaire • À l'aide : <ul style="list-style-type: none"> - Méthode agile : scrum - Logiciel de gestion de projet : Jira - Gestion de versions : Git - Outil CI/CD : GitLab CI - Qualité du code : SonarQube 	<ul style="list-style-type: none"> • Bonne analyse d'un cahier des charges ; • Bonne utilisation de la méthode agile dans la gestion de projet ; • Bonne gestion des versions ; • Mesure calculée de la qualité du code ; • Mise en œuvre appropriée de la chaîne Devops.
Éléments de la compétence	Critères particuliers de performance
A. Connaître les fondamentaux de la gestion de projet	<ul style="list-style-type: none"> • Bonne compréhension des concepts de gestion de projet ; • Différenciation claire des différentes méthodes de gestion de projet ; • Utilisation optimale d'un cahier des charges ; • Préparation claire des différents paramètres d'un projet.
B. Planifier un projet	<ul style="list-style-type: none"> • Analyse appropriée d'un cahier des charges ; • Préparation détaillée d'un projet ; • Utilisation correcte du diagramme de Gantt.

C. Adopter l'approche Agile dans la gestion de projet	<ul style="list-style-type: none"> • Bonne compréhension de la méthode Agile Scrum ; • Bonne maîtrise du processus de la méthode Scrum ; • Identification détaillée des rôles et responsabilités d'un projet ; • Bonne maîtrise des sprints et artéfacts scrum.
D. Mettre en œuvre des outils de gestion de versions et de mesure de la qualité du code	<ul style="list-style-type: none"> • Identification claire de l'intérêt de la gestion de version ; • Idée claire sur les outils existants de gestion de versions ; • Utilisation aisée de l'outil Git ; • Utilisation appropriée de l'outil Gitlab ; • Bonne maîtrise des métriques de la qualité du code ; • Manipulation appropriée de l'outil SonarQube.
E. Mettre en œuvre les outils de la chaîne du DevOps	<ul style="list-style-type: none"> • Identification claire du lien entre l'agilité et DevOps ; • Maîtrise de la méthodologie CALMS ; • Bonne maîtrise de l'architecture de base d'un pipeline CI/CD ; • Utilisation judicieuse de Gitlab CI.

Suggestions Pédagogiques

Compétence 10 :	Adopter l'approche agile	Code : DIA_DEVOWFS_TS-10
DURÉE : « 100 h »	Compétences Préalables : Compétences 1/2/3/4/5/7	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 9/11/12	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Connaître les fondamentaux de la gestion de projet	A.1- Découvrir les Concepts de gestion de projet	<ul style="list-style-type: none"> Concepts de gestion de projet : <ul style="list-style-type: none"> Définition d'un projet, Gestion de projets, Ressources de projet, Livrables de projet, Charte de projet. Parties prenantes de projet : <ul style="list-style-type: none"> Acteurs internes et externes du projet. Principaux rôles dans un projet informatique : <ul style="list-style-type: none"> Chef de projet, Matrice d'assignation des responsabilités. Caractéristiques de base d'un projet : 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap Projet de synthèse 	10%

		<ul style="list-style-type: none"> - Objectifs, - Résultat ou produit attendu, - Durée, - Activités, - Ressources. • Contraintes dans la gestion d'un projet : <ul style="list-style-type: none"> - Délai, - Coût, - Qualité. 	<ul style="list-style-type: none"> - Identification des ressources, des livrables, de la charte, des parties prenantes, des rôles, des caractéristiques et contraintes liées au projet - Choix de l'agilité comme méthode de gestion pour les prochaines étapes - Proposition de la démarche de la solution 	
	A.2- Découvrir les différentes méthodes de gestion de projet	<ul style="list-style-type: none"> • Méthodes prévisibles (cascades, V, Y) : <ul style="list-style-type: none"> - Définition et méthode, - Avantages de cette méthodologie, - Inconvénients. • Méthodes imprévisibles (Agile) : <ul style="list-style-type: none"> - Signification d'Agile en gestion de projet, - Manifeste Agile, Les principes fondateurs, - Méthodes Agiles. • Cycle en V vs. Méthodes agiles 		
B. Planifier un projet	B.1- Analyser le cahier des charges	<ul style="list-style-type: none"> • Compréhension des besoins client, • Contexte du projet, 	<ul style="list-style-type: none"> • Seul ou en groupe 	10%

		<ul style="list-style-type: none"> • Périmètre du projet, • Détection des risques liés à la nature du projet, • Proposition des solutions possibles. 	<ul style="list-style-type: none"> • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) 	
	B.2- Préparer le projet	<ul style="list-style-type: none"> • Répartition de l'ensemble des fonctionnalités en tâches, • Estimation de la durée de réalisation de chaque tâche, • Ordonnancement des tâches, • Chemin critique, • Echancier et la chronologie des tâches, • Affectation des ressources aux tâches, • Maîtrise des coûts, • Détermination des points de validation, • Elaboration du digramme de Gantt. 	<ul style="list-style-type: none"> • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Analyse détaillée du cahier des charges - Préparation du projet (fonctionnalités, tâches, chemin critique, affectation des ressources, coûts, points de validation, diagramme de Gantt) - Proposition de la 	

			démarche de la solution	
C. Adopter l'approche Agile dans la gestion de projet	C.1- Appréhender la méthodologie Agile Scrum	<ul style="list-style-type: none"> • Définition de la méthode Agile Scrum, • Manifest Agile (valeurs et principes), • Processus de la méthode Scrum : pre-game, game, post-game, • Rôles et responsabilités : <ul style="list-style-type: none"> - Equipe de développement, - Product owner, - Scrum master, - Sakeholders, • Evénements Scrum : <ul style="list-style-type: none"> - Sprint, - Planification des sprints, - Mêlée quotidienne (daily Scrum), - Revue de Sprint (Sprint review), - Rétrospective Sprint (Sprint retrospective). • Artéfacts Scrum : <ul style="list-style-type: none"> - Définition, 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse : <ul style="list-style-type: none"> - Utilisation de l'outil Scrum/Jira dans la gestion du projet - Proposition de la démarche de la solution 	40%

		<ul style="list-style-type: none"> - Carnet de produit (backlog product), - Carnet de Sprint (Sprint backlog), - Incrément (Increment). 		
	C.2 – Manipuler l'outil de gestion de projet Agile (Scrum/Jira)	<ul style="list-style-type: none"> • Présentation de Jira, • Installation et configuration d'un compte Jira, • Création de projet, • Choix du type de projet (projet géré par l'équipe, projet géré par l'entreprise), • Création d'un backlog product, • Ajout des différents types de tickets : <ul style="list-style-type: none"> - Epic, - Story, - Bug, - Tâche, - Sous tâche. • Planification d'un sprint : <ul style="list-style-type: none"> - Identification des tickets du backlog sprint - Création de tâche : <ul style="list-style-type: none"> ▪ Affectation de ticket du backlog, ▪ Affectation du membre de l'équipe à 		

		<p>la tâche,</p> <ul style="list-style-type: none"> ▪ Complexité de la tâche, ordre de priorité de la tâche, Détermination des dépendances entre tâches, ▪ Estimation de la durée de réalisation de la tâche en jour/homme, ▪ Détermination des dates début et fin, ▪ Etat de la tâche (à faire, en cours, terminé...). <ul style="list-style-type: none"> • Manipulation du tableau de bord de sprint <ul style="list-style-type: none"> - Consultation des tâches, - Modification, - Suppression. • Utilisation de la feuille de route (roadmap) : <ul style="list-style-type: none"> - Définition, - Utilité pour avoir une vue d'ensemble, - Consultation de la feuille de route, • Génération des rapports Agile : <ul style="list-style-type: none"> - Définition, 		
--	--	---	--	--

		<ul style="list-style-type: none"> - Utilité, - Types de rapports (sprint, graphique burndown, burndown de version, graphique de vélocité), - Génération de rapports. 		
D. Mettre en œuvre des outils de gestion de versions et de mesure de la qualité du code	D.1- Manipuler les outils de gestion de versions (Git/Gitlab)	<ul style="list-style-type: none"> • Intérêt de la gestion de version, • Présentation des outils existants de gestion de versions, • Présentation de Git, • Présentation de Gitlab, • Comparaison Git vs Gitlab, • Création d'un compte Gitlab, • Présentation des fonctionnalités de Gitlab : <ul style="list-style-type: none"> - Gestion de profil, - Tableau de bord, - Repository, - Pull request, - Etc. • Manipulation des dépôts avec Gitlab, • Installation de Git, • Manipulation de Git user interface (GUI), 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Préparation des outils de gestion de versions Git/Gitlab - Gestion des versions du 	20%

		<ul style="list-style-type: none"> Manipulation des commandes de base de Git (Git bash), Notion de branches avec Git, Création des branches avec Git, Gestion des conflits de fusion avec Git. 	projet avec Git, en utilisant les différentes fonctionnalités vues dans le cours	
	D.2- Manipuler l'outil de mesure de la qualité du code (SonarQube)	<ul style="list-style-type: none"> Notions des métriques de la qualité du code, Présentation des outils existants de mesure de la qualité du code, Présentation de SonarQube, Installation de SonarQube, Intégration de SonarQube avec les outils ALM (Application Life Cycle Management : IDE, langages, bases de données, outils de gestion de versions...), Configuration de SonarQube, Analyse de code source à travers la génération de rapports sur : <ul style="list-style-type: none"> Identification des duplications de code, Mesure du niveau de documentation, Respect des règles de programmation, Détection des bugs potentiels, 	<ul style="list-style-type: none"> Préparation de l'outil SonarQube Mesure de qualité du projet, à l'aide de l'outil Proposition de la démarche de la solution 	

		<ul style="list-style-type: none"> - Evaluation de la couverture de code par les tests unitaires, - Analyse du design et de l'architecture d'une application. 		
E. Mettre en œuvre les outils de la chaîne du DevOps	E.1- Introduire la chaîne DevOps	<ul style="list-style-type: none"> • Introduction aux concepts DevOps (définition, avantages, outils), • Lien entre l'agilité et DevOps, • Caractéristiques de la méthodologie DevOps (CALMS) : <ul style="list-style-type: none"> - Culture, - Automatisation, - Lean, - Measure, - Share, • Définition des notions : <ul style="list-style-type: none"> - Intégration continue (Continuous Integration CI), - Livraison continue/Déploiement continue (Continuous Delivery/Continuous Deployment CD). 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse : <ul style="list-style-type: none"> - Mise en place de CI/CD avec Gitlab - Proposition de la démarche de la solution 	20%

	E.2- Mettre en place la CI/CD avec Gitlab	<ul style="list-style-type: none"> • Définition de GitLab CI, • Configuration de Git pour GitLab CI, • Intérêt d'un pipeline DevOps (automatisation des étapes de livraison logicielle), • Architecture de base d'un pipeline CI/CD, • Workflow du pipeline : <ul style="list-style-type: none"> - Phase de création/construction de l'application, - Phase de test, - Phase de distribution, - Phase de déploiement. • Configuration du pipeline : <ul style="list-style-type: none"> - Jobs (défini ce qu'il faut faire), - Stages (Défini le moment d'exécution des jobs), • Manipulation d'un pipeline (démarrage, arrêt, suppression...) • Configuration du CI/CD avec le fichier gitlab-ci.yml 		
--	---	--	--	--

Compétence 11 : « Gérer les données »	
Code de la compétence : DIA_DEVOWFS_TS-11	Durée : 100 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et / ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De spécifications techniques - De cahier des charges • À l'aide : <ul style="list-style-type: none"> - D'un éditeur de texte - De serveurs de base de données (MySQL, MongoDB) - L'invite interactive (Shell) 	<ul style="list-style-type: none"> • Maîtrise du langage de programmation procédurale de MySQL ; • Bonne connaissance des outils de protection d'une base de données MySQL ; • Définition claire d'une base de données NoSQL ; • Manipulation correcte des bases de données NoSQL.
Éléments de la compétence	Critères particuliers de performance
A- Exploiter les fonctionnalités avancées d'un SGBD relationnel	<ul style="list-style-type: none"> • Maîtrise du langage de programmation procédurale ; • Bonne maîtrise de l'utilisation des instructions et des structures de contrôle ; • Bonne gestion des transactions ; • Gestion adaptée des exceptions ; • Utilisation appropriée des procédures stockées, fonctions et déclencheurs ; • Optimisation d'une base de données MySQL ; • Implémentation correcte des règles de sécurité et des droits d'accès dans un serveur MySQL ;

	<ul style="list-style-type: none"> • Bonne gestion des comptes utilisateurs d'une base de données MySQL.
B- Exploiter les fonctionnalités des bases de données NoSQL MongoDB	<ul style="list-style-type: none"> • Définition claire des bases de données NoSQL ; • Mise en place appropriée d'une base de données MongoDB • Utilisation correcte des documents dans une base de données NoSQL <ul style="list-style-type: none"> - JSON • Manipulation correcte des requêtes CRUD et des opérateurs ; • Utilisation optimale des méthodes de recherche de l'information dans une base de données MongoDB ; • Exécution convenable des requêtes depuis des programmes python ; • Sécurisation judicieuse d'une base de données MongoDB.

Suggestions Pédagogiques

Compétence 11 :	Gérer les données	Code : DIA_DEVOWFS_TS-11
DURÉE : « 100 h »	Compétences Préalables : Compétences 1/2/6/8	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 9/10/12	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A- Exploiter les fonctionnalités avancées d'un SGBD relationnel	A.1 –Maîtriser le langage de programmation procédurale sous MySQL	<ul style="list-style-type: none"> Rappel sur le langage de manipulation des données (SQL), Présentation du langage de programmation procédural de MySQL, Distinction des types des programmes MySQL : <ul style="list-style-type: none"> Procédures stockées, Fonctions, Déclencheurs. Maîtrise des instructions de bases : <ul style="list-style-type: none"> Structure d'un bloc d'instruction, Portée des objets, 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap Projet de synthèse <ul style="list-style-type: none"> Modélisation de la base de données 	45%

		<ul style="list-style-type: none"> - Casse et lisibilité, - Identificateurs, - Commentaire, - Variable (variable scalaire, variable de session), - Affectation, - Opérateurs. • Structure de contrôle : <ul style="list-style-type: none"> - Structures conditionnelles (trois formes IF, CASE), - Structures répétitives (WHILE, REPEAT .. UNTIL, LOOP). • Manipulation des données (Insertion, Modification, Suppression), • Gestion des transactions, • Gestion des exceptions, • Manipulation des curseurs, • Création de procédures stockées et des fonctions : 	<ul style="list-style-type: none"> - Préparation de la base de données avec MySQL - Préparation des procédures stockées, fonctions et déclencheurs utiles pour la bonne gestion de la base de données dans les prochaines étapes du projet - Proposition de la démarche de la solution 	
--	--	---	---	--

		<ul style="list-style-type: none"> - Syntaxe de PS et fonction, - Appel de PS et fonction. • Mise en place des déclencheurs. 		
	A.2 – Optimiser une base de données MySQL	<ul style="list-style-type: none"> • Optimisation de la structure de la base de données (Création et utilisation d'Index), • Optimisation des requêtes SQL, • Optimisation de la configuration de serveur MySQL (Paramètres de configuration). 		
	A.3 – Protéger la base de données MySQL	<ul style="list-style-type: none"> • Définition de la sécurité générale du serveur, • Implémenter les règles de sécurité et droits d'accès au serveur MySQL : <ul style="list-style-type: none"> - Rôle, - Droits d'accès, - Contrôle d'accès, - Causes des erreurs, - Hashage de mots de passe. • Gestion des comptes utilisateurs. 		

B- Exploiter les fonctionnalités des bases de données NoSQL MongoDB	B.1- Découvrir les bases de données NoSQL	<ul style="list-style-type: none"> • Définition des bases de données NoSQL, • Comparaison entre les bases de données traditionnelles et NoSQL, • Caractéristiques des bases de données NoSQL, • Différents types de bases de données NoSQL (document, clé / valeur, colonne, graphe), • Comparaison entre les BDD NoSQL selon les critères suivants : <ul style="list-style-type: none"> - Coût, - Cohérence des données, - Disponibilité du système, - Langage d'interrogation, - Fonctionnalité. • Positionnement de MongoDB par rapport aux différentes solutions. 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Manipulation de la base de données avec MongoDB - Mise en place des requêtes depuis des programmes Python - Sécurisation de la base de données MongoDB - Proposition de la démarche de la solution 	55%
	B.2- Mettre en place une base de données MongoDB	<ul style="list-style-type: none"> • Installation sur Windows et Linux : <ul style="list-style-type: none"> - Installation du serveur, - Création d'un répertoire pour stocker les données, 		

		<ul style="list-style-type: none"> - Lancement du serveur. • Administration du serveur MongoDB, • Création de bases de données (show dbs, use db, show collections, drop database), • Création d'une collection de documents. 		
	B.3-Modéliser les documents	<ul style="list-style-type: none"> • Structure d'un document JSON, • Différences entre modéliser pour MongoDB versus une base de données relationnelles, • Modélisation des liens : <ul style="list-style-type: none"> - Enchâssement (embedding), - Liaisons (Linking). • Utilisation des espaces de noms, des collections et des documents, • Les types de données (chaîne, numérique, booléen, object, array, objectId, date). 		
	B.4- Manipuler les données avec MongoDB	<ul style="list-style-type: none"> • Manipulation des requêtes de base CRUD (insert, update, delete) ; • Recherche de l'information find(), 		

		<p>findOne(),Distinct())</p> <ul style="list-style-type: none"> Utilisation des opérateurs, Tri des documents. 		
	B.5- Effectuer des requêtes depuis des programmes Python	<ul style="list-style-type: none"> Présentation et installation de pymongo, Connexion des bases de données avec le serveur MongoDB, Création des requêtes : <ul style="list-style-type: none"> Requêtes simples, Création des indexs, Requêtes d'agrégation, Requêtes de modifications. 		
	B.6 – Sécuriser une base de données MongoDB	<ul style="list-style-type: none"> Import / export des données (mongoexport, mongoimport) Sécurité des accès (authentification). 		

Compétence 12 : « Développer en front-end »	
Code de la compétence : DIA_DEVOWFS_TS-12	Durée : 100 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et / ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De spécifications techniques - De cahier des charges • À l'aide : <ul style="list-style-type: none"> - Éditeur de code source comme Visual Studio Code - Git 	<ul style="list-style-type: none"> • Bonne connaissance de toutes les notions du développement front end ; • Différenciation claire entre front end et back end ; • Bonne maîtrise des différentes fonctionnalités du Framework React JS ; • Exploitation détaillée des différents concepts de Redux.
Éléments de la compétence	Critères particuliers de performance
A. Appréhender les fondamentaux du web front end	<ul style="list-style-type: none"> • Bonne connaissance des fondamentaux de base du développement front end ; • Distinction apparente entre front et back end ; • Idée claire sur les concepts de base d'une SPA ; • Différenciation claire entre Spa et MPA.
B. Exploiter le Framework React JS	<ul style="list-style-type: none"> • Bonne manipulation de npm dans l'installation de l'environnement de développement React ; • Idée claire sur l'architecture d'une application React ; • Bonne exploitation de toutes les fonctionnalités de React ; • Déploiement correct d'une application React.

C.Exploiter Redux avec React JS	<ul style="list-style-type: none">• Acquisition précise des concepts de Redux (Store, Reducers, States et Actions) ;• Manipulation correcte des différents concepts et fonctionnalités Redux ;• Bonne manipulation des APIs Redux ;• Bonne maîtrise des moyens de débogage des changements en temps réel ;• Utilisation optimale des middlewares.
---------------------------------	---

Suggestions Pédagogiques

Compétence 12 :	Développer en front-end	Code : DIA_DEVOWFS_TS-12
DURÉE : « 100 h »	Compétences Préalables : Compétences 1/2/3/4/5	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 9/10/11/13	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Appréhender les fondamentaux du web front end	A.1 – Introduire les Concepts de base du web front end	<ul style="list-style-type: none"> Rappels des fondamentaux des sites web statiques Différence entre front end et back end Présentation des Frameworks front end 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap 	10%
	A.2 – Appréhender les concepts de base d'une application web monopage (SPA)	<ul style="list-style-type: none"> Notion d'application web monopage SPA Single Page Application (SPA) vs Multi-Page Application (MPA) Fonctionnement d'une application web de type SPA Avantages et inconvénients d'une application web de type SPA 		

B. Exploiter le Framework React JS	B.1- S'initier aux principes de React	<ul style="list-style-type: none"> • Présentation de React • Installation de l'environnement de développement React à l'aide de npm (installation de NodeJS, npm,...) • Intérêt du développement des composants avec React (Components) • Architecture d'une application React 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) 	45%
	B.2- Maîtriser les fonctionnalités de base de React	<ul style="list-style-type: none"> • Utilisation de CRA (Create React App) : <ul style="list-style-type: none"> - Notion de CRA - Installation de CRA - Lancement de CRA - Prise en main de CRA • Création d'élément DOM à l'aide de JSX : <ul style="list-style-type: none"> - Définition de JSX - DOM virtuel vs DOM réel - Éléments en JSX - Attributs en JSX - Expressions en JSX - Imbrication d'éléments JSX - Rendu JSX (ReactDOM.render()) 	<ul style="list-style-type: none"> • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Préparation de l'environnement de développement React - Utilisation de React dans le développement des différentes fonctionnalités du front end de l'application (composants, propriétés, événements, états, feuilles de styles, assets, animations, ...) 	

		<ul style="list-style-type: none"> • Présentation des fonctionnalités des composants React (propriété, événement, état) • Création d'un composant React : <ul style="list-style-type: none"> - Composant simple - Composant de type classe - Composant de type fonction • Manipulation des propriétés (Props) • Communication inter-composants (envoi, réception de données) • Gestion d'événements (Events) • Gestion d'état (States) • Création des styles à l'aide de : <ul style="list-style-type: none"> - Feuille de style pour un composant - Style en ligne - Modules CSS • Utilisation des assets • Gestion d'animations • Utilisation des listes (map(), key,...) 	<ul style="list-style-type: none"> - Utilisation des fonctionnalités avancées de React dans le projet (routage, hooks, ...) - Utilisation des tests dans les étapes de réalisation de la partie front end - Déploiement - Proposition de la démarche de la solution 	
--	--	---	---	--

		<ul style="list-style-type: none"> • Affichage conditionnel (if, opérateurs logiques, opérateurs ternaires...) • Création de formulaire • Consommation des API (Fetch API) 		
	B.3- Maîtriser les fonctionnalités avancées de React	<ul style="list-style-type: none"> • Réutilisation du code à l'aide de la composition • Routage : <ul style="list-style-type: none"> - Concept de routage - Installation de React Router - Utilisation des éléments (Switch, Route, withRouter, Link,...) - Routage imbriqué • Utilisation de Hooks • Utilisation des tests sur les composants (présentation, environnement de tests...) • Déploiement d'application 		
C.Exploiter Redux avec React JS	C.1-Appréhender les concepts de bases de Redux	<ul style="list-style-type: none"> • Présentation de la bibliothèque Redux • Installation de l'API Redux • Concepts de Redux : 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites 	45%

		<ul style="list-style-type: none"> - Store - Reducers - States - Actions • Fonctionnement d'une application Redux • State React vs State Manager Redux 	<p>du formateur</p> <ul style="list-style-type: none"> • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse 	
	C.2- Manipuler les fonctionnalités importantes de Redux	<ul style="list-style-type: none"> • Création d'une application Redux de base : <ul style="list-style-type: none"> - Création du State - Création du Reducer - Création des actions - Création du Store • Envoi des actions à partir du composant (useDispatch) • Utilisation du state dans un composant (useSelector) • Liaison du store Redux au Provider • Manipulation des states avec la librairie Immer: <ul style="list-style-type: none"> - Introduction à Immer 	<ul style="list-style-type: none"> - Préparation de l'environnement de développement Redux - Utilisation de Redux dans le développement des différentes fonctionnalités - Proposition de la démarche de la solution 	

		<ul style="list-style-type: none"> - Cas d'utilisation de Immer - Intérêt de Immer • Manipulation des selectors : <ul style="list-style-type: none"> - Utilisation des props dans les selectors - Réalisation des calculs dans les selectors - Déclaration globale des selectors (en dehors des composants) • Événement synchrone vs événement asynchrone • Manipulation des événements asynchrones dans un composant (useStore, getState, dispatch) : <ul style="list-style-type: none"> - Fonctionnement d'un événement asynchrone - Appel asynchrone 		
	C.3- Redux avancé	<ul style="list-style-type: none"> • Combinaison des reducers (combineReducers) : <ul style="list-style-type: none"> - Combinaison des reducers 		

		<p>manuellement</p> <ul style="list-style-type: none"> - Intérêt de combineReducers - Utilisation de combineReducers • Débogage des changements d'état d'une application en temps réel (devTools) : <ul style="list-style-type: none"> - Intérêt de devTools - Installation de devTools (extension navigateur vs librairie à installer) - Utilisation de devTools dans Redux • Interaction avec les API Redux (status, pending, resolved, rejected...) • Optimisation de l'utilisation de Redux avec Redux-Toolkit : <ul style="list-style-type: none"> - Utilité de Redux-Toolkit - Installation de Redux-Toolkit - Utilisation des fonctions Toolkit (configureStore, createAction, createReducer) 		
--	--	---	--	--

		<ul style="list-style-type: none"> - Utilisation des slices • Extension des fonctionnalités Redux à l'aide des middlewares : <ul style="list-style-type: none"> - Utilité des middlewares - Présentation des middlewares importants - Utilisation d'un middleware de login (redux-logger) - Utilisation d'un middleware de manipulation d'actions asynchrones (redux-thunk) 		
--	--	--	--	--

Compétence 13 : « Développer en back-end »	
Code de la compétence : DIA_DEVOWFS_TS-13	Durée : 120 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et / ou en groupe À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales De consignes De spécifications fonctionnelles De spécifications techniques De cahier des charges -des projets réels À l'aide : <ul style="list-style-type: none"> IDE (phpstorm ...) D'un éditeur de texte (VScode..) De logiciels à installer : <ul style="list-style-type: none"> Composer MongoDB Mysql postman Framework Laravel CMS Wordpress 	<ul style="list-style-type: none"> Bonne connaissance des frameworks APIs et CMS PHP ; Bonne compréhension des fonctionnalités du Framework PHP Laravel ; Maîtrise parfaite des concepts Model View Controller (MVC) ; Bonne maîtrise de la programmation Laravel en utilisant toutes les fonctionnalités associées ; Gestion appropriée de la sécurité ; Bonne manipulation de l'ORM Eloquent ; Manipulation correcte des tests ; Utilisation correcte d'un CMS dans l'administration d'un site.
Éléments de la compétence	Critères particuliers de performance
A. Découvrir le Framework PHP Laravel	<ul style="list-style-type: none"> Bonne connaissance des frameworks APIs et CMS PHP ; Bonne maîtrise des concepts Model View Controller (MVC) ; Bonne préparation de l'environnement de développement Laravel ;

	<ul style="list-style-type: none"> • Création réussie d'un premier projet Laravel.
B. Programmer avec Laravel	<ul style="list-style-type: none"> • Gestion appropriée du routage ; • Utilisation adéquate des Middlewares ; • Adoption aisée de la protection CSRF ; • Manipulation appropriée des différents fondements du modèle MVC ; • Bonne gestion des erreurs ; • Utilisation adéquate de la journalisation ; • Bonne maîtrise du principe du broadcasting ; • Bonne utilisation des collections, assets, contracts et Helpers ; • Gestion appropriée des événements, fichiers et packages ; • Gestion adéquate des tâches.
C. Approfondir la programmation Laravel	<ul style="list-style-type: none"> • Bonne gestion de la sécurité ; • Interaction appropriée avec une base de données ; • Bonne manipulation de l'ORM Eloquent • Prise en main correcte des tests.
D. Administrer un site à l'aide d'un CMS	<ul style="list-style-type: none"> • Manipulation adaptée des éléments essentiels d'un CMS ; • Bonne utilisation d'un CMS dans la personnalisation graphique d'un site ; • Installation correcte des extensions et des plugins ; • Utilisation convenable des composants CMS.

Suggestions Pédagogiques

Compétence 13 :	Développer en back-end	Code : DIA_DEVOWFS_TS-13
DURÉE : « 120 h »	Compétences Préalables : Compétences 1/2/3/4/5/6/7/8/10/11	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 12/15	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Découvrir le Framework PHP Laravel	A.1 – Découvrir les notions fondamentales des Frameworks PHP	<ul style="list-style-type: none"> Présentation des Frameworks PHP (Laravel, Symfony, Zend...) Intérêt du Framework back end Laravel Concepts Model View Controller (MVC) : <ul style="list-style-type: none"> Architecture 3-tiers MVC Motifs de conception (Design Patterns) Présentation des API PHP Présentation des CMS PHP (Wordpress, Prestashop...) 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap 	10%

	A.2 – Préparer l’environnement de Laravel	<ul style="list-style-type: none"> • Architecture du Framework Laravel • Installation complète de Laravel (Composer, commandes PHP Artisan) • Configuration de l’environnement Laravel • Création d’un premier projet • Lancement du serveur Laravel 	<ul style="list-style-type: none"> • Projet de synthèse <ul style="list-style-type: none"> - Préparation de l’environnement de développement Laravel - Proposition de la démarche de la solution 	
B. Programmer avec Laravel	B.1- Connaître les fondements du modèle MVC Laravel	<ul style="list-style-type: none"> • Gestion du routage : <ul style="list-style-type: none"> - Routage de base (redirection, affichage, liste de routes) - Paramètres de routage - Routes nommées - Groupe de routage - Liaisons de modèles de routes - Routes de repli - Usurpation (spoofing) de la méthode du formulaire - Accès à la route courante - Cross Origin Resource Sharing (CORS) - Mise en cache des routes 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l’aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Préparation de l’environnement de développement Redux 	30%

		<ul style="list-style-type: none"> • Utilisation des Middleware (définition, enregistrement, paramétrage, terminate) • Protection CSRF • Manipulation des contrôleurs : <ul style="list-style-type: none"> - Intérêt des contrôleurs - Implémentation de contrôleur - Contrôleur Middleware - Contrôleur de ressources - Injection de dépendances • Manipulation des requêtes http : <ul style="list-style-type: none"> - Interaction avec les requêtes - Input - Fichiers • Manipulation des réponses http : <ul style="list-style-type: none"> - Création - Redirection - Types de réponses - Réponse Marco • Manipulation des vues : <ul style="list-style-type: none"> - Création - Transmission des données aux vues 	<ul style="list-style-type: none"> - Utilisation de Laravel dans le développement des différentes fonctionnalités de la partie back-end de l'application - Proposition de la démarche de la solution 	
--	--	---	--	--

		<ul style="list-style-type: none"> - Affichage des compositeurs - Optimisation des vues • Création des template Blade • Génération d'URL • Manipulation des sessions HTTP • Validation des données d'entrée • Gestion des erreurs • Journalisation (logging) 		
	B.2- Maîtriser le Framework Laravel	<ul style="list-style-type: none"> • Manipulation des commandes Artisan • Broadcasting : <ul style="list-style-type: none"> - Présentation du concept - Installation côté client - Installation côté serveur - Manipulation des événements de diffusion • Manipulation du cache • Manipulation des collections • Compilation des assets (Mix) • Utilisation des contracts 		

		<ul style="list-style-type: none"> • Gestion des événements • Gestion de fichiers (configuration, instances, récupération, stockage, suppression) • Manipulation de Helpers • Prise en compte de multi-langages (localization) • Manipulation de l'API Symfony Mailer • Manipulation des notifications (génération, envoi, types) • Développement de packages • File d'attente (Queue) : <ul style="list-style-type: none"> - Création d'un job - Job Middleware - Dispatching jobs - Job Batching • Planification des tâches (task scheduling) : <ul style="list-style-type: none"> - Définition des schedules - Exécution d'un schedule 		
--	--	--	--	--

C. Approfondir la programmation Laravel	C.1 Gérer la sécurité	<ul style="list-style-type: none"> • Bibliothèques d'authentification (JWT, PASSPORT, Sanctum) • Authentification • Chiffrement • Autorisation • Hachage • Réinitialisation de mot de passe • Vérification d'e-mail 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap 	40%
	C.2 Interagir avec la base de données	<ul style="list-style-type: none"> • Mise en route : <ul style="list-style-type: none"> - Configuration de la base de données - Connexions - Exécution de requêtes SQL (requêtes CRUD) - Gestion des transactions - Connexion la Commun Line Interface (CLI) • Générateur de requêtes (Query Builder) : <ul style="list-style-type: none"> - Select, join, union, where, when, ordering, grouping, limit, offset, insert, update, delete - Déverrouillage 	<ul style="list-style-type: none"> • Projet de synthèse <ul style="list-style-type: none"> - Sécurisation de l'application - Interaction avec la base de données, en utilisant toutes les fonctionnalités vues dans le cours - Manipulation de l'ORM Eloquent - Utilisation des tests dans les étapes de réalisation de la partie back-end 	

		<ul style="list-style-type: none"> - Débogage • Pagination de la base de données • Gestion de migration (génération, structure, exécution, manipulation des tables, colonnes, indexes et events) • Création de Seeders (utilisation des modèles factories, appels de seeders additionnels, désactivation d'événements de modèles) • Insertion des données d'un formulaire dans une base de données • Utilisation de Redis 	<ul style="list-style-type: none"> - Proposition de la démarche de la solution 	
	C.3 Manipuler l'ORM Eloquent	<ul style="list-style-type: none"> • Intérêt de l'Object Relational Mapper ORM Eloquent • Manipulation des modèles (Génération, récupération, insertion, suppression...) • Relations Eloquent : <ul style="list-style-type: none"> - Définition - Types de relations Eloquent - Relations polymorphes 		

		<ul style="list-style-type: none"> - Relations dynamiques - Interrogation de relations - Agrégation, insertion et mise à jour des modèles associés • Manipulation des collections • Transformation des valeurs d'attributs Eloquent (accessor, mutator, casting) • Sérialisation 		
	C.4 Prendre en charge les tests	<ul style="list-style-type: none"> • Prise en main de l'environnement de test PHPUnit • Configuration de l'environnement de test • Création des tests • Exécution des tests • Tests HTTP • Tests Console • Tests Navigateur • Tests de bases de données • Mocking 		

D. Administrer un site à l'aide d'un CMS	D.1- Manipuler les éléments essentiels d'un CMS	<ul style="list-style-type: none"> Définition d'un CMS Présentation des CMS existants Installation d'un CMS Présentation de la structure interne d'un CMS Création de bases de données et éléments d'administration Gestion des utilisateurs Manipulation des fonctionnalités (section, catégorie, article, publication, workflow) Déploiement 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap Projet de synthèse <ul style="list-style-type: none"> Réalisation d'un site basique en utilisant le CMS Proposition de la démarche de la solution 	20%
	D.2- Personnaliser graphiquement un site à l'aide d'un CMS	<ul style="list-style-type: none"> Différents éléments d'une page web et leurs dispositions Manipulation des éléments (menu, formulaire, disposition) Utilisation de thèmes graphiques Assemblage des éléments dans un site 		
	D.3- Manipuler les outils avancés d'un CMS	<ul style="list-style-type: none"> Installation d'extensions supplémentaires Installation de plugins CMS Utilisation de composants CMS 		

Compétence 14 : « Créer une application Cloud native »	
Code de la compétence : DIA_DEVOWFS_TS-14	Durée : 90 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> • Individuellement et / ou en groupe • À partir : <ul style="list-style-type: none"> - De mises en situations écrites et orales - De consignes - De spécifications fonctionnelles - De spécifications techniques - De cahier des charges • À l'aide : <ul style="list-style-type: none"> - D'un éditeur de texte (Visual Studio Code) - Du framework Laravel /Express JS - D'un outil Devops (GillLab CI) - De logiciels : <ul style="list-style-type: none"> ▪ Docker ▪ RabbitMQ ▪ Azure cloud ▪ Azure Kubernetes Service (AKS) 	<ul style="list-style-type: none"> • Bonne connaissance de toutes les notions théoriques du cloud native ; • Manipulation correcte des conteneurs ; • Bonne appréhension des microservices ; • Création d'une application fonctionnelle en microservices ; • Maîtrise du déploiement avec AKS ; • Bonne maîtrise des outils d'automatisation des processus de création et déploiement d'une application cloud.
Éléments de la compétence	Critères particuliers de performance
A. Introduire le cloud native	<ul style="list-style-type: none"> • Compréhension parfaite du concept du cloud ; • Différenciation claire entre cloud privé, public et hybride ; • Bonne connaissance des services du cloud (IAAS, PAAS, SAAS) ; • Bonne compréhension de l'approche cloud native.
B. Manipuler les conteneurs	<ul style="list-style-type: none"> • Bonne maîtrise de la notion du conteneur ;

	<ul style="list-style-type: none"> • Prise en main aisée de Docker ; • Manipulation correcte des applications à conteneur unique ; • Manipulation correcte des applications Multi-conteneurs.
C. Créer des microservices	<ul style="list-style-type: none"> • Bonne compréhension de l'architecture des microservices ; • Différenciation claire entre les architectures monolithique et des microservices ; • Idée claire sur les protocoles de communication ; • Bonne connaissance des moyens du conteneur Docker pour la création d'une application en microservices. • Manipulation aisée de RabittMQ.
D. Déployer via Azure Kubernetes Service	<ul style="list-style-type: none"> • Bonne maîtrise des notions fondamentales de Azure Cloud ; • Bonne connaissance des composants principales d'Azure ; • Manipulation correcte de Azure Kubernetes Servic (AKS).
E. Automatiser les processus	<ul style="list-style-type: none"> • Création d'une application en cloud native fonctionnelle ; • Déploiement adéquat d'une application cloud native.

Suggestions Pédagogiques

Compétence 14 :	Créer une application Cloud native	Code : DIA_DEVOWFS_TS-14
DURÉE : « 90 h »	Compétences Préalables : Compétences 1/2/3/4/5/6/7/8/10/13	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 15	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Introduire le cloud native	A.1 – Définir le cloud	<ul style="list-style-type: none"> • Concept du cloud et ses avantages ; • Exemple des fournisseurs cloud ; • Différence entre cloud privé, public et hybride ; • Services du cloud (IAAS, PAAS, SAAS). 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap 	10%
	A.2 – Définir l'approche cloud native	<ul style="list-style-type: none"> • Définition ; • Avantages ; • Vue générale sur les caractéristiques du cloud native : <ul style="list-style-type: none"> - Automatisation des processus du développement et de déploiement, - microservices et conteneurs. 		

B. Manipuler les conteneurs	B.1- Appréhender la notion du conteneur	<ul style="list-style-type: none"> Définition ; Différence entre machine virtuelle et conteneur ; Avantages ; 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur 	30%
	B.2- Prendre en main Docker	<ul style="list-style-type: none"> Installation de Docker ; Terminologies Docker : images, Containers, Docker Daemon, Docker Client, Docker Hub ; 	<ul style="list-style-type: none"> À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) 	
	B.3- Réaliser une application à conteneur unique	<ul style="list-style-type: none"> Configuration d'une image d'une simple application à l'aide de Dockerfile ; Commandes de base de docker ; 	<ul style="list-style-type: none"> Quiz MindMap Projet de synthèse 	
	B.4- Réaliser une application multi-conteneurs	<ul style="list-style-type: none"> Commandes de base de docker-compose ; Configuration du fichier docker-compose.yml ; Démarrage et gestion des conteneurs ; 	<ul style="list-style-type: none"> - Préparation de Docker - Utilisation de docker - Configuration du fichier docker-compose.yml - Manipulation des conteneurs - Proposition de la démarche de la solution 	

C. Créer des microservices	C.1- Définir l'architecture des microservices	<ul style="list-style-type: none"> • Différence entre l'architecture monolithique et l'architecture des microservices ; • Concepts de base des microservices ; • Protocoles de communication : AMQP (Advanced Message Queuing Protocol), API Gateway ... ; • Exemple d'architectures microservices (application web décomposée en plusieurs services) ; 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse 	20%
	C.2- Créer une application en microservices avec le conteneur Docker	<ul style="list-style-type: none"> • Création des services avec leur base de données ; • Développement des API REST de chaque service ; • Lancement d'un conteneur pour chaque service ; • Installation et configuration de RabbitMQ ; • Notions de Messages, Queues et Exchanges de RabbitMQ ; • Communication entre microservices avec 	<ul style="list-style-type: none"> - Création des microservices avec Docker - Communication entre les microservices - Proposition de la démarche de la solution 	

		RabbitMQ (Créer des producteurs et consommateurs, envoyer, recevoir et mettre des messages en files d'attente ...) ;		
D. Déployer via Azure Kubernetes Service	D.1- Introduire Azure	<ul style="list-style-type: none"> Définition de Azure Cloud ; Notions de base en Azure Cloud ; Composantes principales d'Azure : Régions, Ressources Azure, Zones de disponibilité ; Groupes de ressources, Azure Resource Manager, Abonnements ; Exemples des services offerts par Azure ; Création d'un compte azure ; 	<ul style="list-style-type: none"> Seul ou en groupe Selon les instructions verbales ou écrites du formateur À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) Quiz MindMap Projet de synthèse <ul style="list-style-type: none"> Déploiement de l'application via Azure Kubernetes Service Proposition de la démarche de la solution 	20%
	D.2- Utiliser Azure Kubernetes Service	<ul style="list-style-type: none"> Définition de Kubernetes ; Définition d'Azure Kubernetes Service : utilité et avantages ; Différence entre K8S (Kubertnetes) et AKS (Azure Kubernetes Service) Vue globale sur l'architecture AKS ; Exploration de la terminologie AKS : Pools, Nodes, Pods, Container, Deployment, Manifest; 		

		<ul style="list-style-type: none"> • Création d'un cluster Kubernetes ; • Commandes de base de Kubectl ; • Déploiement d'un conteneur dans un cluster Kubernetes ; 		
E. Automatiser les processus	E.1- Manipuler une application avec l'architecture des microservices en appliquant Devops	<ul style="list-style-type: none"> • Rappel du concept DevOps ; • Configuration d'un pipeline CI sous GitLab CI; • Création d'une application en cloud native ; • Déploiement d'une application cloud native. 	<ul style="list-style-type: none"> • Seul ou en groupe • Selon les instructions verbales ou écrites du formateur • À l'aide des supports fournis par le formateur (polycop, documents, cours et un exemple de cahier des charges) • Quiz • MindMap • Projet de synthèse <ul style="list-style-type: none"> - Manipulation et déploiement en cloud native - Proposition de la démarche de la solution 	20%

Compétence 15 : « Réaliser un projet de synthèse »	
Code de la compétence : DIA_DEVOWFS_TS-15	Durée : 60 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> En groupe restreint de préférence binôme À partir : <ul style="list-style-type: none"> De mises en situations écrites et orales D'un cahier des charges De consignes De spécifications fonctionnelles De base documentaire À l'aide : <ul style="list-style-type: none"> IDE, langages, éditeurs, logiciels de gestion de projet Agile, Frameworks et l'ensemble des technologies étudiées et manipulées durant la formation 	<ul style="list-style-type: none"> Bonne compréhension des spécifications techniques Modélisation correcte de l'application Complétude fonctionnelle de l'application Ergonomie de l'application Respect des délais Sécurisation de l'application Documentation de l'application Soutenance du projet devant un groupe de stagiaires et formateurs
Éléments de la compétence	Critères particuliers de performance
A. Appréhender un projet de conception d'application web	<ul style="list-style-type: none"> Compréhension des objectifs de l'application Compréhension des besoins précis de l'utilisateur Compréhension des fonctionnalités de l'application Planification objective des activités Bonne estimation des coûts et délais Maîtrise de l'outil de gestion de projet Agile
B. Concevoir l'application	<ul style="list-style-type: none"> Qualité de la modélisation UML Respect de la charte graphique Conformité de la maquette avec la charte graphique

	<ul style="list-style-type: none"> • Enchaînement d'interfaces utilisateurs • Respects des règles ergonomiques issues de l'expérience utilisateur
C. Réaliser l'application	<ul style="list-style-type: none"> • Respect des bonnes pratiques de codage • Complétude des fonctionnalités de l'application en phase avec les besoins spécifiques de l'utilisateur • Complétude des tests • Qualité de la documentation fournie • Niveau de sécurisation de l'application
D. Réaliser la veille technologique	<ul style="list-style-type: none"> • Maîtrise des techniques de recherche • Maîtrise des techniques de capitalisation de l'information aux fins d'une veille technologique

Suggestions Pédagogiques

Compétence 15 :	Réaliser un projet de synthèse	Code : DIA_DEVOWFS_TS-15
DURÉE : « 60 h »	Compétences Préalables : Compétences 1/2/3/4/5/6/7/8/9	
Type de compétences : Spécifique	Compétences en parallèles : Compétences 10/11/12/13/14	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Appréhender un projet de conception d'application web	A.1- Analyser la demande des utilisateurs de l'application	<ul style="list-style-type: none"> Spécifications du cahier des charges Caractéristiques techniques de l'application Fonctionnalités attendues Utilisateurs cibles Rôles des utilisateurs 	<ul style="list-style-type: none"> De préférence en groupe (binôme de stagiaires) Exercices et travaux pratiques : <ul style="list-style-type: none"> Description par le stagiaire dans ses propres mots des différentes composantes d'un projet Établissement d'un plan d'exécution de projet informatique suite à des mises en situation à partir d'un mandat ou à partir d'un cahier 	15%
	A.2- Planifier les activités à réaliser	<ul style="list-style-type: none"> Planification du travail : <ul style="list-style-type: none"> Tâches à effectuer Estimation de la durée Estimation des coûts Détermination des outils nécessaires pour développer l'application Détermination des outils nécessaires pour 		

		exploiter l'application	de charges et de dossiers de spécifications techniques - Utilisation d'un logiciel de gestion de projets Agile	
B. Concevoir l'application	B.1- Modéliser l'application à l'aide d'UML	<ul style="list-style-type: none"> • Conception des diagrammes UML • Conception d'interfaces utilisateurs • Conception de la base de données • Conception des composants 	<ul style="list-style-type: none"> • De préférence en groupe (binôme de stagiaires) • Etudes de cas proposées par le formateur, (le cas échéant, études de cas proposées par le stagiaire et validées par le formateur) permettant de reproduire toutes les étapes d'un projet de conception d'application : <ul style="list-style-type: none"> - Planification des activités de travail - Gestion des activités de travail - Produire une interface utilisateur - Conception de la base de données - Conception des composants 	30%
	B.2- Réaliser un prototype de l'application	<ul style="list-style-type: none"> • Maquettage de l'application (wireframe, maquette) • Prototypage de l'application (contenus, interactions entre éléments) • Effets graphiques (éléments visuels, animations...) 		

C. Réaliser l'application	C.1- Développer la partie front end	<ul style="list-style-type: none"> • Conception des interfaces graphiques • Exploitation des Frameworks étudiés (Bootstrap, React-Redux....) • Tests et validation • Documentation des instructions 	<ul style="list-style-type: none"> • De préférence en groupe (binôme de stagiaires) • Etudes de cas proposées par le formateur, (le cas échéant, études de cas proposées par le stagiaire et validées par le formateur) permettant de reproduire toutes les étapes d'un projet de développement d'application : <ul style="list-style-type: none"> - Codage de l'application à l'aide d'un langage de programmation - Préparation de l'application pour l'exploitation - Tests de l'application - Déploiement de la solution - Rédaction de la documentation 	40%
	C.2- Développer la partie back end	<ul style="list-style-type: none"> • Exploiter les Frameworks étudiés (Laravel...) • Création des composants • Codage des programmes • Optimisation du code • Tests et validation • Documentation des instructions 		
	C.3- Sécuriser l'application	<ul style="list-style-type: none"> • Accès sécurisé et hiérarchisé des utilisateurs • Accès sécurisé des données 		
	C.4- Restituer l'application	<ul style="list-style-type: none"> • Déploiement de l'application • Guide d'installation • Aide en ligne • Soutenance du travail devant un groupe de stagiaires et formateurs 		

D. Réaliser la veille technologique	D.1- Appréhender l'utilité de la recherche de l'information	<ul style="list-style-type: none"> • Démonstration de l'utilité de recherche d'information par des mises en situation • Formulation des objectifs de recherche • Détermination des différents aspects de recherche • Utilisation des agents intelligents pour la recherche sur Internet. • Recherche de sources d'informations (site web, documents techniques, manuels de références, associations professionnelles...) • Critères de recherche thématique 	<ul style="list-style-type: none"> • De préférence en groupe (binôme de stagiaires) • Travaux pratiques : <ul style="list-style-type: none"> - Objectif de la recherche ainsi que les différents aspects à considérer - Sélection des ressources disponibles - Recherche de l'information utile - Extraction de l'information utile en utilisant des techniques de prise de notes - Comparaison des informations recueillies - Validation des informations recueillies - Détermination des utilitaires appropriés - Constitution d'une banque 	15%
	D.2- Exploiter les ressources techniques disponibles	<ul style="list-style-type: none"> • Choix des sources de référence • Appropriation de l'outil • Consignation des informations utiles sur un sujet donné • Méthodes de prise de notes • Vérification de la pertinence de l'information extraite • Recherche d'outils sur des sites spécifiques • Constitution d'une banque d'outils 		

	D.3- Consigner les résultats de la recherche	<ul style="list-style-type: none"> • Constitution d'un dossier technique sur un sujet donné : <ul style="list-style-type: none"> - Consignes de constitution de dossier - Modèle d'un dossier • Lecture aisée de ce dossier par les autres stagiaires 	d'outils <ul style="list-style-type: none"> - Rédaction des résultats des recherches - Consignation des résultats des recherches 	
--	--	--	--	--

Compétence 16 : « S'intégrer en milieu professionnel »	
Code de la compétence : DIA_DEVOWFS_TS-16	Durée : 160 heures
Contexte de réalisation	Critères généraux de performance
<ul style="list-style-type: none"> Individuellement et sous la supervision du formateur encadrant et du tuteur en entreprise À partir : <ul style="list-style-type: none"> Besoin spécifique en entreprise De mises en situations écrites et orales De consignes De spécifications fonctionnelles De base documentaire À l'aide : <ul style="list-style-type: none"> Divers outils disponibles selon le lieu du stage 	<ul style="list-style-type: none"> Plan d'action de recherche active de stage Adoption d'une attitude professionnelle Respect des échéances Résolution de la problématique posée Qualité des productions Cohérence entre le rapport fourni et les tâches effectuées Soutenance des activités réalisées durant la période de stage
Éléments de la compétence	Critères particuliers de performance
A. Préparer son stage en milieu de travail	<ul style="list-style-type: none"> Cohérence du plan de recherche d'emploi/stage Respect des règles de communication écrite/orale Pertinence de la lettre de motivation Description pertinente des techniques de recherche de stage
B. Réaliser des activités en milieu de travail	<ul style="list-style-type: none"> Attitudes professionnelles positives Manipulation judicieuse des données numériques Résolution correcte des problèmes Respect d'une démarche méthodique Respect des règles de santé/sécurité

	<ul style="list-style-type: none"> • Respect de la hiérarchie • Bonne exécution des tâches à effectuer • Capacités de planification des activités de travail • Pertinence du journal de bord
C. Rédiger un rapport faisant état des activités exercées	<ul style="list-style-type: none"> • Sens de l'autonomie • Rédaction du rapport selon les normes • Pertinence des informations du rapport • Cohérence entre le rapport et le vécu en entreprise • Bonne présentation des activités du stage • Transition entre la perception du métier (cursus) et la réalité du milieu de travail

Suggestions Pédagogiques

Compétence 16 :	S'intégrer en milieu professionnel	Code : DIA_DEVOWFS_TS-16
DURÉE : « 160 h »	Compétences Préalables : Compétences 1/2/3/4/5/6/7/8/9/10/11/12/13/14/15	
Type de compétences : Spécifique	Compétences en parallèles : _	

ÉLÉMENTS DE LA COMPÉTENCE	APPRENTISSAGES DE BASE	ÉLÉMENTS DE CONTENU	ACTIVITÉS D'APPRENTISSAGE	DURÉE SUGGÉRÉE
A. Préparer son stage en milieu de travail	A.1- Décrire les attitudes nécessaires à la recherche dynamique d'un lieu de stage	<ul style="list-style-type: none"> Esprit d'initiative Sens des responsabilités Attitude positive Esprit méthodique 	<ul style="list-style-type: none"> Activités et jeux de rôles permettant aux stagiaires d'acquérir : <ul style="list-style-type: none"> Prise de connaissance des informations et modalités relatives aux stages Critères de sélection des entreprises Choix des entreprises susceptibles de recevoir des stagiaires 	
	A.2- Exploiter les moyens de recherche de stage	<ul style="list-style-type: none"> Buts du stage : <ul style="list-style-type: none"> Observation de diverses facettes du métier Réalisation d'activités professionnelles Renforcement des habiletés cognitives et perceptuelles 		

		<ul style="list-style-type: none"> - Changement de perception qu'entraîne un séjour en entreprise - Familiarisation avec le milieu • Documents officiels : <ul style="list-style-type: none"> - Lois - Règlements - Conventions diverses (de stage...) - Assurances - Politiques de l'entreprise • Types d'entreprises : <ul style="list-style-type: none"> - Grande - Moyenne - PME, Start up • Catégories d'entreprises : <ul style="list-style-type: none"> - Services - Production • Répertoire des entreprises • Consultation d'une banque de données • Lettre de demande de stage • Curriculum vitae 	<ul style="list-style-type: none"> - Démarches pour décrocher un stage - Attitudes et comportements en entreprise 	
--	--	--	---	--

		<ul style="list-style-type: none"> • Contacts téléphoniques • Messagerie électronique • Plateformes de réseautage (LinkedIn...) • Demandes d'entrevues 		
	A.3- Décrire le comportement à adopter en milieu de travail	<ul style="list-style-type: none"> • Attitude d'écoute • Sens de l'observation • Respect des règles de santé / sécurité • Tact et discrétion • Attitude positive • Communication de qualité • Intérêt marqué pour toute nouvelle expérience de travail • Souci de l'excellence 		
B. Réaliser des activités en milieu de travail	B.1- Connaître l'environnement du milieu de travail	<ul style="list-style-type: none"> • Milieu socio-économique : <ul style="list-style-type: none"> - Produits - Marché • Associations professionnelles • Structures • Équipement • Évolution technologique 	<ul style="list-style-type: none"> • Activités sur : <ul style="list-style-type: none"> - Aspects du métier qui diffèrent de la formation reçue - Ecosystème, environnement socio-économique - Compréhension des termes du 	

		<ul style="list-style-type: none"> • Relations interpersonnelles • Santé et sécurité • Éléments à consigner : <ul style="list-style-type: none"> - Possibilité du marché du travail (nouveaux emplois, emplois à la hausse, création d'emplois) - Conditions de travail (horaire, salaire, santé et sécurité au travail) - Contraintes du marché du travail (chômage, compétition, mobilité, formation, spécialité, développement technologique, instabilité économique) 	<p>cahier des charges sous la supervision de l'encadrant en entreprise</p> <ul style="list-style-type: none"> - Observation du contexte de travail et diverses facettes du métier - Réalisation des tâches professionnelles telles qu'elles sont spécifiées dans le CC - Participation à divers échanges interprofessionnels 	
	B.2- Réaliser des activités professionnelles	<ul style="list-style-type: none"> • Tâches professionnelles en rapport : <ul style="list-style-type: none"> - Au service à la clientèle - À la planification du travail - À la demande de l'utilisateur final - Au diagnostic - À l'installation d'ordinateurs et de périphériques 		

		<ul style="list-style-type: none"> - À l'installation d'applications - Au développement d'applications - Aux divers paramétrages et configurations - À l'entretien d'un parc informatique - Etc... 		
	B.3- Adopter une méthodologie d'organisation du travail	<ul style="list-style-type: none"> • Journal de bord comprenant : <ul style="list-style-type: none"> - Evénements de la journée - Activités réalisées - Fiches de travail - Objectifs d'apprentissages - Objectifs personnels... • Importance du journal de bord : <ul style="list-style-type: none"> - Suivi des activités - Evaluation - Validation (tuteur en entreprise/formateur) 		
C. Rédiger un rapport faisant état des	C.1- Rédiger un compte rendu d'activités selon les	<ul style="list-style-type: none"> • Observations : <ul style="list-style-type: none"> - Sur le contexte de travail - Sur les tâches observées 	<ul style="list-style-type: none"> • Activités sur : <ul style="list-style-type: none"> - Rapport faisant état des 	

activités exercées	normes en vigueur	<ul style="list-style-type: none"> • Tâches effectuées • Éléments d'un rapport : <ul style="list-style-type: none"> - Présentation - Introduction - Développement - Conclusion • Validation • Soutenance (présentation orale des activités réalisées en stage devant un groupe de stagiaires et formateurs) 	<p>observations et tâches effectuées par le stagiaire au cours du stage</p> <ul style="list-style-type: none"> - Comparaison de la perception du métier avec les réalités du milieu de travail - Présentation orale des activités réalisées en stage devant un groupe de stagiaires et formateurs - Synthèse/bilan personnel vis-à-vis du vécu en milieu de travail 	
	C.2- Enumérer ses aptitudes associées au métier	<ul style="list-style-type: none"> • Aptitudes : <ul style="list-style-type: none"> - Au plan professionnel - Au plan social • Goûts • Champs d'intérêt : <ul style="list-style-type: none"> - Personnels - Professionnels 		
	C.3- Comparer les perceptions du métier avec les réalités du	<ul style="list-style-type: none"> • Métier et formation : <ul style="list-style-type: none"> - Éléments du processus - Outillage 		

	milieu de travail	<ul style="list-style-type: none"> - Équipement - Technologie - Rythme de production - Tâches et opérations - Importance et indices de difficultés relatives aux tâches et aux étapes du processus - Autorité - Ponctualité - Assiduité 		
--	-------------------	---	--	--

