



Référentiel de formation

Titre Professionnel Développeur Web Full Stack

Nomenclature du niveau de qualification : Niveau 6

Code(s) NSF :

- **326 : Informatique, traitement de l'information, réseaux de transmission**

Formacode(s) :

- **71654 : logiciel système gestion bases données**
- **31090 : programmation web**
- **31054 : informatique**
- **31098 : programmation orientée objet**

N° de fiche

RNCP36400
26X32628

Année 2022-2023



Activités visées :

Le développeur Full Stack travaille à la fois sur le front-end et le back-end d'un site web ou d'une application (la partie client et la partie serveur). Maîtrisant les principales technologies et les principaux langages de programmation actuels, il se distingue par sa grande polyvalence. Au-delà du développement informatique qui constitue le cœur de son métier, il est amené à exercer une grande diversité d'activités et peut même prendre en charge, seul, l'ensemble des composantes de la chaîne de valeur de son produit.

Alors que les métiers de l'informatique en général et du développement en particulier se spécialisent, le métier de développeur Full Stack s'inscrit à contre-courant de cette tendance, répondant à une demande forte et sans cesse croissante.

À l'origine principalement destiné à répondre aux besoins des start-ups et des TPE/PME qui ne disposaient pas de moyens leur permettant d'engager plusieurs développeurs spécialisés, le métier de développeur Full Stack répond désormais également aux besoins des entreprises de services numériques (ESN), des entreprises de taille intermédiaire (ETI) et grands groupes sur des postes tels que « développeur front-end », « développeur back-end » ou encore « lead developer » qui nécessitent un champ de compétences techniques large et une vision métier transversale.

La certification « Développeur Full-stack » a ainsi été conçue avec comme objectifs de répondre aux besoins d'un marché en tension sur le plan du recrutement :

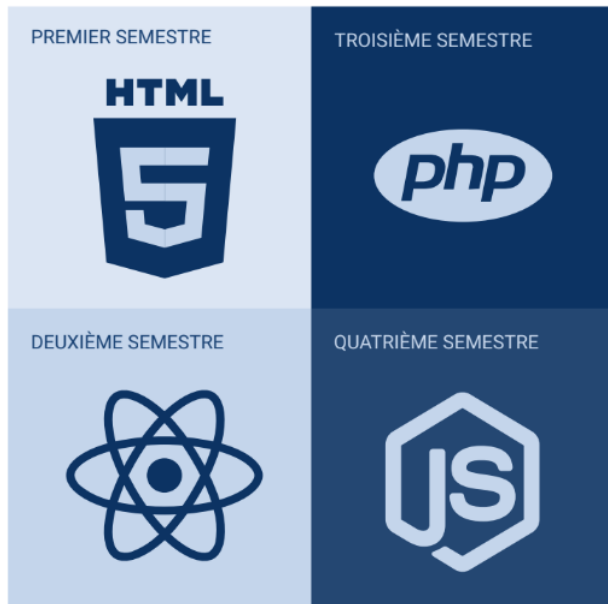
- En attestant des compétences de spécialistes possédant un haut niveau de technicité mais aussi une grande polyvalence
- En permettant la sécurisation des parcours professionnels assis sur des compétences qui permettent une évolution vers de nombreuses spécialisations métiers



Le déroulement de la formation

première année
FRONT-END

deuxième année
BACK-END



L'alternance, formation en centre de formation et entreprise, vise à rendre les étudiants opérationnels pour qu'ils puissent s'insérer professionnellement. Pour atteindre cet objectif, notre équipe pédagogique met tout en œuvre pour que les étudiants puissent monter en compétences au travers d'un projet innovant et stimulant.

Ce projet réalisé au fil de l'année est découpé en 4 grandes étapes :

Année 1

- 1 Premier semestre : réaliser un site statique en HTML / CSS avec Javascript.
- 2 Second semestre : réaliser une application web avec React.

Année 2

- 3 Troisième semestre : réaliser un site dynamique avec PHP, Symfony et MySQL.
- 4 Quatrième semestre : réaliser des architectures en micro-service avec Node.js et MongoDB.

La formation est réalisée en distanciel, seule la soutenance finale a lieu en présentiel sur le site de Paris.

Le rythme de l'alternance est : 1 semaine de formation, 3 semaines en entreprise

Contrats possibles : Apprentissage ou professionnalisation

75% de pratique + 25 % de théorie dans chaque module*



Thèmes à inclure au maximum dans les modules qui s'y prêtent : Ecologie, RSE

Module	Nombre d'heures	Coef	Année
Pré-rentree - Visioconférence d'accueil	3		
Santé et sécurité au travail	4		
Semestre 1 Développement front-end : Les fondamentaux	210		1
S1 Les bases du développement web	14		1
S1 Introduction aux langages HTML et CSS	21		1
S2 Prototypage avec Figma	14		1
S2 Responsive design	21		1
S3 Gestion de projet agile	21		1
S3 Introduction au versionnage avec git	14		1
S4 Introduction à Javascript	35		1
S5 Javascript, approfondissement	35		1
S6 Introduction au référencement naturel	21		1
S6 Introduction à l'UI et à l'UX Design	14		1
Semestre 2 Développement front-end : Le framework React	210		1
S7 Introduction au framework React	35		1
S8 L'écosystème React	35		1
S9 Les API REST	14		1
S9 Les mécanismes de stockage du navigateur	21		1
S10 L'automatisation des tâches avec Webpack	14		1
S10 Les procédures de test	21		1
S11 React Native	35		1
S12 Expression orale & création de supports	21		1
S12 Marketing de soi	14		1
S13 Préparation au projet, méthodologie mémoire, oral blanc (visio)	35		
Nombre d'heures de formation en 1ère année	455		
Semestre 3 Développement back-end : Apache, PHP & MySQL + symfony selon promo	210		2



S14 Programmation procédurale en PHP	35		2
S15 Programmation orientée objets en PHP	35		2
S16 Introduction aux bases de données MySQL	35		2
S17 Sécurité des bases de données	14		2
S17 Le stockage en PHP	21		2
S18 Introduction à la cryptographie	14		2
S18 Configurer la stack LAMP	21		2
S19 Écosystème PHP	35		2
Semestre 4 Architectures micro-service : Node.js & MongoDB	210		2
S20 Introduction à Node.js	35		2
S21 Introduction à MongoDB	35		2
S22 Introduction à la conteneurisation avec Docker	35		2
S23 Les architectures micro-services	21		2
S23 Server-side rendering & prerendering	14		2
S24 Introduction à l'authentification	35		2
S25 Droit de l'informatique	35		2
S26 Projet - S26 Entraînement soutenances - Mémoire écrit (rapport d'alternance et dossier de conception) - Soutenance du projet - Projet pratique (site) - Note du Maître d'apprentissage	35	/ 3 3 3 2	2
Total d'heures en 2ème année	455		
TOTAL Général	910		

soit 26 semaines de formation sur deux ans

Pour la soutenance, les formateurs indiqueront si l'étudiant a une mention :

- **défavorable**
- **passable**
- **favorable**



Le programme

Visioconférence d'accueil

Objectifs :

- Présenter la formation en détails,
- Présenter le projet - écrit + soutenance
- Accompagner dans la découverte de la plateforme Yparéo et de ses fonctionnalités,
- Communiquer toutes les informations relatives au déroulement de la formation,
- Présenter les coefficients et des critères de validation,
- Transmettre les noms et coordonnées des principaux interlocuteurs,
- Présenter le Mastère
- Lire le règlement intérieur
- Répondre à l'ensemble des questions.

Santé et sécurité au travail

Que ce soit en tant qu'étudiant dans le cadre de sa période de formation en milieu professionnel, qu'apprenti lors de l'alternance en entreprise ou que nouvel embauché, le jeune est confronté à une nouvelle réalité qu'il n'a jamais appréhendée.

La compréhension des situations de travail dans toutes leurs dimensions, y compris celles de la santé et sécurité au travail, conduira le jeune à faire des propositions pour une meilleure maîtrise de l'économie globale de l'activité de l'entreprise.

SEMESTRE 1 Développement front-end :Les fondamentaux

Objectifs du semestre :

- Découvrir les rudiments du développement web,
- réaliser les premières pages web,
- réaliser un site complet utilisant HTML, CSS et Javascript.

Les sujets techniques abordés sont simples afin d'être accessibles au plus grand nombre et de laisser la place à d'autres aspects importants du processus de développement :

- bonnes pratiques de programmation
- prototypage



- outils de travail collaboratif
- gestion de projet agile
- et plus encore

Les technologies utilisées lors du semestre 1 :

1 Le protocole HTTP : HTTP est le protocole du web. Il permet à un client de récupérer des ressources auprès d'un serveur.

2 Les langages HTML et CSS

Les langages HTML et CSS permettent de structurer et de mettre en forme le contenu des pages.

3 La bibliothèque Bootstrap : Bootstrap est une bibliothèque logicielle permettant de simplifier le développement d'un site responsive.

4 Le langage Javascript : Javascript est le langage de programmation du navigateur. Il permet d'ajouter de la logique et de l'interactivité à vos pages.

5 Figma : Figma est un outil en ligne permettant de prototyper vos applications. Le prototypage est une étape indispensable de la réalisation d'un projet front-end.

6 La méthode Scrum : Scrum est une méthode de gestion de projet issue du giron des méthodes agile que vous retrouverez fréquemment au sein des entreprises du secteur.

7 Le gestionnaire de version Git : Git est un gestionnaire de version. Il permet de conserver un historique des modifications de votre projet et de travailler à plusieurs en limitant les risques.

Compétences :

- **Participer à une réunion en identifiant la typologie du demandeur (client) pour comprendre sa demande et participer à l'élaboration d'une réponse adaptée à ses besoins**
- **Conseiller le Maître d'Ouvrage en mettant à contribution ses connaissances et compétences techniques pour la réalisation du cahier des charges**
- **Participer à l'élaboration d'une planification réaliste en tenant compte de ses propres contraintes et compétences afin de garantir l'atteinte des objectifs fixés pour la réussite du projet**
- **Concevoir une ou plusieurs maquettes « wireframe » en utilisant un outil spécialisé afin de fournir une ébauche au demandeur**
- **Mettre en œuvre un environnement de développement collaboratif adapté au projet d'application afin d'optimiser le temps de développement, le transfert de compétences auprès de ses pairs et la qualité logicielle**
- Mettre en œuvre et maîtriser l'utilisation d'un système de gestion de code source distribué (SCM) permettant de conserver l'historique des développements, d'organiser la collaboration des développeurs et d'appliquer un workflow standard pour la revue de code et le suivi des bugs
- Mettre en œuvre un environnement de développement local ou distribué reproductible reposant sur l'utilisation de la virtualisation système ou de la virtualisation applicative



- Utiliser un environnement de développement intégré (IDE) adapté aux technologies sur lesquelles repose l'application et tirer partie de ses fonctionnalités
- Sélectionner et mettre en oeuvre les outils et méthodes Agile de développement afin de collaborer efficacement avec les acteurs du projet d'application
- **Comprendre le cycle de développement et mettre en oeuvre les principales méthodes de gestion de projet de développement afin de les appliquer au sein d'une équipe (XP, SCRUM, DSDM, ASD).**
- **Rédiger des compte-rendu d'activité destinés aux membres d'une équipe de projet afin de permettre le suivi de l'avancement du projet et la traçabilité des réalisations techniques**

Les bases du développement web SEMAINE 1

Au programme HTTP • HTML • CSS

Module 1 : Les bases du développement web

2 jours : 14h

Durant ce premier module de l'année, vous découvrirez le fonctionnement général du web et les outils dont vous vous servirez pour travailler sur vos projets.

Objectifs du module :

le fonctionnement général du web

- l'architecture client / serveur
- le rôle du navigateur et du serveur
- le protocole HTTP
- les langages clients et serveurs
- les bases de données

Mise en place de votre environnement de travail :

- installer un dualboot Ubuntu
- découvrir l'environnement de bureau Gnome
- installer la stack LAMP
- installer et configurer VS Code

Module 2 : Introduction aux langages HTML et CSS

3 jours : 21h

Vous apprendrez les langages HTML et CSS, indispensables pour réaliser un site. Vous en comprendrez les règles et la terminologie. À l'issue de cette première semaine vous serez capable d'écrire vos premières pages web.

Objectifs du module :

le langage HTML

- les langages de présentation
- code HTML minimal



- structure d'une balise HTML
- les balises sémantiques
- les balises de structure

le langage CSS

- sélecteurs et propriétés
- système de boîte
- priorité des sélecteurs
- couleur et typographie
- bien structurer son contenu
- flexbox

les outils du navigateur

- voir le code source
- inspecter l'élément
- analyser le trafic réseau

Le responsive design SEMAINE 2

Au programme : Figma • HTML • CSS • Bootstrap

Module 3 : Prototypage avec Figma

2 jours : 14h

Avant d'agir, il est bon de réfléchir. Dans ce module, vous apprendrez à vous servir de Figma pour réaliser un prototype de vos applications. Vous apprendrez à cette occasion de bonnes pratiques d'ergonomie que vous pourrez employer pour concevoir des interfaces utilisateurs agréables et claires.

Objectifs du module :

le prototypage avec Figma

- pourquoi réaliser un prototype ?
- les étapes de création d'un prototype
- l'interface de Figma
- créer des groupes
- créer des vues

les principes d'ergonomie et de design

- conventions et habitudes
- sign & feedback
- espacement & alignement
- hiérarchie du contenu

Module 4 : Responsive design

3 jours : 21h



Une fois familiarisé avec le prototypage, vous intégrerez votre travail au sein d'une page. Ce faisant, vous professionnaliserez votre approche en découvrant des méthodes pour organiser votre CSS employées dans l'industrie.

Objectifs du module :

les méthodologies CSS

- CSS sémantique
- OOCSS
- BEM

le responsive design en CSS

- le responsive design avec flexbox
- les règles conditionnelles « media »
- tester la responsivité
- l'approche "mobile first"
- créer une navigation

les bases de Bootstrap

- système de grille
- les formulaires

!! Début du projet • les composants

Travailler en équipe SEMAINE 3

Au programme Scrum • HTML • CSS • Git

Module 5 : Gestion de projet agile

3 jours : 21h

Le développement est une étape importante de la réalisation d'un projet, elle n'est cependant pas la seule. Dans ce module vous apprendrez toutes les étapes qui permettent de passer d'une idée à une concrétisation de qualité.

Objectifs du module :

les différents cycles de vie d'un projet

- Cycle de cascade
- Cycle en V
- Cycle en spiral
- Qu'est-ce que l'agilité ?

les bases de la méthode Scrum

- L'utilité des méthodes agiles
- Le Scrum Guide
- La terminologie Scrum
- Les rôles de Scrum



- Les grandes étapes d'un projet Scrum
- Les étapes d'un sprint
- Outils de planification projet
- Sensibilisation au "pair programming"

Module 6 : Introduction au versionnage avec git

2 jours : 14h

Programmer en groupe n'est pas une tâche simple. Des systèmes de versionnage tels que git permettent néanmoins de la simplifier et aident à résoudre les problèmes qui peuvent survenir.

Objectifs du module :

le versionnage avec git

- Qu'est-ce qu'un système de versionnage
- Le système de branche
- La ligne de commande
- Dépôt distant avec github
- Les bonnes pratiques

Javascript niveau I Semaine 4

Au programme : Javascript, Html, CSS

Module 7 : Introduction à Javascript

5 jours : 35h

Javascript permet d'ajouter une couche d'interactivité à vos pages. Lors de ce module, vous découvrirez ce langage qui sera votre porte d'entrée dans le monde de la programmation.

Objectifs du module :

La programmation avec Javascript ES5

- Qu'est ce que Javascript
- Qu'est ce que EcmaScript
- les variables
- réaliser des opérations mathématiques
- manipuler des chaines de caractères
- les conditions
- les boucles
- les fonctions
- manipuler les tableaux
- manipuler les objectifs

La manipulation du contenu d'une page

- qu'est ce que le DOM
- Manipuler le DOM
- Gérer les événements
- Découverte des API du navigateur



Javascript niveau II - Semaine 5

Au programme : Javascript - HTML - CSS

Module 8 : Javascript, approfondissement

5 jours : 35h

Javascript est un langage complexe dont il convient de comprendre les subtilités. Ce module vise à présenter les plus importantes et à approfondir ce qui a été vu lors de la semaine précédente.

Vous apprendrez :

les spécificités de Javascript

les ajouts des normes EcmaScript 6, 7 et 8

- La portée des variables
- Les fonctions auto-appelantes
- Les types primitifs et leur surcouche objet
- Les passages par référence
- Les fonctions anonymes
- Les closures
- Les prototypes

• let, const

• fonctions fléchées

• opérateur de décomposition

• opérateur de destructuration

la programmation asynchrone avec

JavaScript

• Qu'est-ce que la programmation asynchrone

• Délais et intervalles

• Gérer des promesses

• Contrôler le flux d'exécution à l'aide d'async & await

Le référencement naturel - SEMAINE 6

Au programme : HTML

Module 9 : Introduction au référencement naturel

3 jours : 21h

Pour un propriétaire de site web, sa visibilité sur les moteurs de recherche est la question la plus déterminante. Le référencement naturel est un ensemble de bonnes pratiques de développement qui permet d'augmenter ses chances d'être bien placé dans les résultats.

Objectifs du module :



Les outils pour le référencement

- Google insight
- Google analytics
- Ahref

Les bonnes pratiques de référencement naturel

- Les métadonnées
- La ligne de flottaison
- L'accessibilité
- Les liens et le "google juice"

Les techniques d'optimisation

- Minification
- Optimisation des images
- Compression
- Lazy loading

Module 9.2 : Initiation à l'UI et à l'UX Design

2 jours : 14h

Objectifs du module

- Définitions
- comprendre les enjeux
- les notions clés

Semestre 2 Développement front-end : Le framework React

Objectifs du semestre :

Nombreuses sont les évolutions qui ont ébranlé le monde du développement front-end ces dernières décennies. Parmi elles, les frameworks Javascript tel que React ont causé des transformations architecturales notables. C'est l'objet du second semestre. Dominant son marché, React est devenu un outil indispensable pour le bagage technique de tout développeur front-end.

- Approfondir la maîtrise de Javascript
- Découvrir les outils et les pratiques utilisées dans l'industrie
- Optimiser les processus de travail
- S'initier au développement mobile avec React Native

Les technologies utilisées lors du semestre 2

1 Le framework React : React est un framework Javascript permettant de diviser son code en composants réactifs.

2 Redux : Redux est un système autonome souvent employé avec React permettant de gérer un système d'état partagé entre des composants.



3 Immer : Immer est une bibliothèque Javascript permettant d'assurer l'immuabilité des objets Javascript. Ce faisant il vient résoudre certains problèmes qui peuvent se présenter lorsque l'on utilise Redux.

4 La bibliothèque Material UI : Material UI est une bibliothèque de composants React conçu par Google permettant de concevoir rapidement des applications.

5 Axios : Axios est une petite bibliothèque Javascript permettant d'effectuer des requêtes HTTP.

6 Le bundler Webpack : Webpack est un outil qui permet d'automatiser certaines tâches techniques. La plupart des projets front-end modernes l'utilisent dès la création.

7 Le linter ESLint : Un linter est un outil permettant d'assurer l'homogénéité du code source.

8 Le moteur de test Mocha : Mocha permet de réaliser des tests de son code Javascript. Il est souvent employé avec la bibliothèque d'assertion Chai.

9 Le framework React Native : React Native est une implémentation de React qui permet de produire des applications mobiles natives.

Compétences :

- **Rédiger les spécifications techniques de besoin (STB) d'un projet d'application ou de site web à partir d'un cahier des charges afin de décrire de manière exhaustive les exigences à satisfaire en termes d'utilisation**
- **Modéliser l'application logicielle et ses données en utilisant une méthode standard afin d'abstraire la réalité et mieux comprendre le système à réaliser**
- Utiliser le formalisme du langage de modélisation unifié UML et de UP afin de traduire les besoins de l'application
- Constituer le dossier de modélisation comprenant les diagrammes, les cas d'utilisation, les classes d'analyse et décrivant l'architecture logicielle n-tiers de l'application
- Utiliser des patrons de conceptions afin d'appliquer des solutions logicielles standards
- **Concevoir l'architecture des bases de données à l'aide d'un outil de modélisation afin de représenter la structure de la couche de persistance nécessaire au fonctionnement de l'application ou du site web**
- Recenser les informations / données du domaine
- Établir le schéma entité-association des données et définir le schéma de la base de données afin de permettre l'élaboration d'une base de données normalisée
- Dans le cas d'une base de données existante, déduire le schéma entité-association des données de la base à faire évoluer
- **À partir du dossier de spécifications fonctionnelles et techniques, déterminer l'architecture logicielle de l'application ou du site web**
- **Développer la partie front-end d'une application web, hybride, mobile ou desktop en utilisant plusieurs langages de programmation et en appliquant les bonnes pratiques d'UX, de sécurité informatique et d'écoconception**
- À partir d'une maquette et/ou d'une charte graphique, structurer et intégrer une interface utilisateur responsive pour le web
- À partir d'une maquette, structurer et intégrer une interface utilisateur mobile first ou desktop en utilisant une technologie hybride (web app / progressive web app)



- Sélectionner et mettre en oeuvre un ou plusieurs langages de programmation front-end adaptés aux exigences et contraintes de l'application ou du site web
- Sélectionner et mettre en oeuvre un framework de développement frontend adapté aux exigences et contraintes de l'application ou du site web
- Développer des composants front-end graphiques et / ou fonctionnels en choisissant des structures de données adaptées et des algorithmes pertinents afin d'assurer la qualité logicielle
- Appliquer les bonnes pratiques UX (expérience utilisateur)
- Appliquer les recommandations de sécurité pour le développement frontend afin de s'inscrire dans une démarche de « sécurité en profondeur »
- Appliquer les bonnes pratiques d'éco-conception afin de minimiser l'impact écologique de la partie front-end de l'application
- **Tester la partie front-end de l'application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés**
- Élaborer un plan de test logiciel exhaustif
- Réaliser des tests de conformité du langage de balisage structurant le contenu des interfaces web, mobile hybride et desktop
- Réaliser des tests unitaires pour les composants fonctionnels
- Réaliser des tests fonctionnels, utilisateurs et de compatibilité avec les différentes plateformes
- **Industrialiser le développement de la partie front-end de l'application et automatiser les processus d'assurance qualité**
- Automatiser l'exécution d'un vérificateur / correcteur de syntaxe pour chaque langage de programmation constituant le code source de la partie front-end, si nécessaire
- Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers)
- Utiliser un groupeur de modules afin de configurer et d'exécuter la chaîne de build et ainsi optimiser les performances de l'interface
- **Consommer une API de manière sécurisée en s'appuyant sur des mécanismes d'identification et d'authentification robustes afin d'intégrer des sources de données et des services tiers**
- **Développer une API permettant à des services tiers de consommer des données produites par l'application au travers de formats ouverts**
- Sélectionner un format d'API adapté aux caractéristiques de l'application et à son environnement (consommateurs d'API)
- Sélectionner et mettre en oeuvre un format d'échange de données adapté aux caractéristiques de l'application et à son environnement (consommateurs d'API)
- Sécuriser l'accès à l'API en implémentant des mécanismes d'identification et d'authentification robustes
- Documenter et publier une API
- **Tester la partie back-end de l'application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés**



- Élaborer un plan de tests logiciels exhaustif pour le back-end
- Réaliser des tests unitaires sur les composants du back-end
- Réaliser des tests fonctionnels sur les composants du back-end
- **Industrialiser le développement de la partie back-end de l'application ou du site web et automatiser les processus d'assurance qualité**
- Mettre en oeuvre un processus d'intégration continue de manière à minimiser les erreurs humaines et de rationaliser le cycle de développement de la partie back-end de l'application
- Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers) de la partie back-end de l'application
- Automatiser la construction de l'application en configurant la chaînes de build, l'exécution des tests, l'exécution des outils d'assurance qualité et l'analyse statique de sécurité
- Faire évoluer ses connaissances et compétences en développement backend
- Formaliser et mettre en oeuvre une démarche de veille technologique
- Utiliser un outil de knowledge management afin de capitaliser la connaissance et faciliter sa transmission

Découverte du framework React - SEMAINE 7

Au programme : React • Webpack

Module 10 : Introduction au framework React

5 jours : 35h

Durant ce module, vous découvrirez l'utilité d'un framework Javascript et apprendrez à réaliser des applications web en utilisant le framework React.

Objectifs du module :

L'utilité d'un framework Javascript

- Qu'est-ce que React.js ?
- Le principe de composants
- Mise en place d'un projet React avec create-react-app
- Cas d'utilisation
- Les alternatives à React

la syntaxe fonctionnelle de React

- Créer un composant fonctionnel
- Les hooks prédéfinis
- Les hooks personnalisés
- Le routage

l'utilité du gestionnaire de dépendances NPM

- Qu'est-ce que node.js ?
- Quelle utilité pour les projets front-end ?
- Installer des dépendances avec NPM



L'écosystème React - SEMAINE 8

Au programme React • Redux • Immer • Material UI

Module 11 : L'écosystème React

5 jours : 35h

React est rarement utilisé seul. Il vient avec un large écosystème logiciel qui permet d'étendre ses possibilités et de simplifier certains aspects du développement. Ce module vise à présenter les principaux membres de cet écosystème.

Objectifs du module :

les concepts de Redux

- Store
- Reducer
- Dispatcher
- Action
- Middleware
- Implémenter Redux dans son application React

l'immuabilité avec Immer

- Pourquoi avoir des objets immutables ?
- Le fonctionnement de Immer

la bibliothèque Material UI

- Installer Material UI
- Survol de quelques composants

La persistance - SEMAINE 9

Au programme : React • Redux • Axios

Module 12 : Les API REST

2 jours : 14h

Les API REST permettent à une application React d'interagir avec un serveur. Vous apprendrez comment ses API fonctionnent et comment les utiliser dans vos projets React.

Objectifs du module :

le concept d'API REST

- Qu'est-ce qu'une API
- Les différents types d'API
- Comment consommer une API REST en Javascript



l'utilisation des API REST avec React

- Découverte de quelques API REST existantes
- L'utilisation de useEffect
- Gérer les erreurs

Module 13 : Les mécanismes de stockage du navigateur

3 jours : 21h

Il arrive que l'on ait besoin de stocker des informations d'une navigation à l'autre. Après avoir vu comment récupérer des données depuis un serveur, nous verrons dans ce module comment les faire persister dans votre navigateur.

Objectifs du module :

les différents mécanismes de stockage
du navigateur

- Les cookies
- Les contraintes légales vis à vis des cookies
- Le localStorage
- IndexedDB

Le pipeline de production - SEMAINE 10

React • Webpack • Babel • ESLint • Mocha

Module 14 : L'automatisation des tâches avec Webpack

2 jours : 14h

La plupart des projets front-end modernes intègrent des outils d'automatisation des tâches pour optimiser le code en production et élargir le champ des possibles lors du processus de développement.

Objectifs du module :

les bases de webpack

- Qu'est-ce qu'un bundler ?
- La configuration de webpack
- Les plugins

la configuration de Webpack

- séparer l'environnement de développement et de production
- séparer les fichiers de configuration avec webpack-merge

la configuration de Babel

- le concept de transpiler



- la configuration de babel
- les plugins et les presets
- configurer .browserlistrc
- gérer les modules UMD

Module 15 : Les procédures de test

3 jours

Tester son application est une étape obligatoire pour qui veut garantir un minimum de sécurité et de stabilité. Vous apprendrez à concevoir une procédure de test pertinente et saurez comment l'implémenter avec Mocha et Chai.

Objectifs du module :

les différents types de tests

- Les tests unitaires
- Les tests d'intégration
- Les tests bout-en-bout

l'implémentation de tests avec Mocha

- découverte de la bibliothèque Mocha
- la bibliothèque d'assertion Chai
- bonnes pratiques de rédaction de tests unitaires
- mockage des appels API

React Native - SEMAINE 11

Au programme React Native

Module 16 : React Native

5 jours : 35h

React Native utilise React et JSX pour développer des applications mobiles natives.

Objectifs du module :

les bases de React Native

- L'utilité de React Native
- Les différences avec React dans le navigateur
- Les composants natifs
- Exporter et installer un APK la mise en place d'un projet avec React Native
- Installer Android Studio
- Installer un émulateur Android
- Créer un projet avec Expo

Présentation de



projet

SEMAINE 12 - thématique :

Au programme

React Native

Module 17 : Expression orale et création de supports

3 jours : 21h

Programmer n'est pas suffisant, encore faut-il savoir convaincre. Ce module vous permettra de vous préparer pour la présentation de votre projet.

Objectifs du module :

créer des supports et les présenter à l'oral

- Création d'un support (powerpoint)
- Règles de design graphique
- Règles éditoriales (numérotation, sommaire, fil d'ariane, etc.)
- Structurer son discours
- Capter son auditoire

Compétences attestées :

Savoir créer un support de présentation

Savoir convaincre

Module 17.2 Marketing de soi

2j : 14h

Le Marketing de soi a pour objectif de mieux se connaître pour ensuite mieux se vendre. C'est un peu le personal branding 2.0. La méthodologie est principalement axée sur la connaissance de soi et l'amélioration de son image sur le Web.

- mieux connaître sa personnalité : points fort, points faibles
- e-reputation
- mes outils : CV, LM, portfolio, vidéo, book, LinkedIn
- recherche de job

Semestre 3 Développement back-end : Apache, PHP & MySQL

Objectifs du semestre :

PHP a été conçu en 1994, d'après une étude*, il représenterait en 2022 78% des langages serveur employés sur Internet, tant par des petits sites que par des services internationaux à très haut trafic.

- connaître le langage PHP
- connaître le système de gestion de base de données MySQL



- connaître l'administration système avec GNU/Linux et le serveur web Apache2
- Connaître les bonnes pratiques de sécurité.

Les technologies utilisées lors du semestre

1 Le langage PHP : PHP est un langage de programmation côté serveur qui vous permet de générer des pages HTML, d'implémenter votre logique métier, d'interagir avec votre base de données et de faire tout ce dont vous avez besoin.

2 Le SGBD MySQL : MySQL est un SGBD (Système de Gestion de Base de Données). Il gèrera pour vous le stockage de vos données et permet d'y accéder via le langage SQL.

3 La modélisation de BDD : La modélisation de la base de données est une étape indispensable de la conception d'un back-end.

4 Le serveur HTTP Apache2 : Apache2 est un serveur HTTP. Il permet de rendre accessible vos pages web en ligne ou sur votre réseau local.

5 L'utilitaire OpenSSL : OpenSSL est une bibliothèque logicielle et un utilitaire dédié à la cryptographie. Il vous sera notamment utile pour créer des certificats.

6 Le framework PHPUnit : PHPUnit est un framework de test en PHP. Il vous permettra de garantir la stabilité de vos applications PHP.

Compétences :

- **Développer la couche de persistance des données en appliquant les bonnes pratiques de sécurité informatique et de performance**
- Mettre en oeuvre et administrer des bases de données relationnelles de manière sécurisée
- Exploiter et interroger des bases de données relationnelles depuis le backend de l'application
- Mettre en oeuvre et administrer des bases de données NoSQL de manière sécurisée
- Exploiter et interroger des bases de données NoSQL
- Faire évoluer les bases de données
- **Développer la partie back-end d'une application en utilisant plusieurs langages de programmation et en appliquant les bonnes pratiques de sécurité informatique et d'écoconception**
- Sélectionner et mettre en oeuvre un ou plusieurs langages de programmation back-end adaptés aux exigences et contraintes de l'application ou du site web
- Sélectionner et mettre en oeuvre un framework de développement backend adaptés aux exigences et contraintes de l'application ou du site web
- À partir du référentiel de modélisation, développer des composants back-end fonctionnels en choisissant des structures de données adaptées et des algorithmes pertinents afin d'assurer la qualité logicielle
- Appliquer les recommandations de sécurité pour le développement backend afin de s'inscrire dans une démarche de « sécurité en profondeur »
- Appliquer les bonnes pratiques d'écoconception afin de minimiser l'impact écologique de la partie back-end de l'application



- **Consommer une API de manière sécurisée en s'appuyant sur des mécanismes d'identification et d'authentification robustes afin d'intégrer des sources de données et des services tiers**
- **Développer une API permettant à des services tiers de consommer des données produites par l'application au travers de formats ouverts***
- Sélectionner un format d'API adapté aux caractéristiques de l'application et à son environnement (consommateurs d'API)
- Sélectionner et mettre en oeuvre un format d'échange de données adapté aux caractéristiques de l'application et à son environnement (consommateurs d'API)
- Sécuriser l'accès à l'API en implémentant des mécanismes d'identification et d'authentification robustes
- Documenter et publier une API
- **Tester la partie back-end de l'application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés**
- Élaborer un plan de tests logiciels exhaustif pour le back-end
- Réaliser des tests unitaires sur les composants du back-end
- Réaliser des tests fonctionnels sur les composants du back-end
- **Industrialiser le développement de la partie back-end de l'application ou du site web et automatiser les processus d'assurance qualité**
- Mettre en oeuvre un processus d'intégration continue de manière à minimiser les erreurs humaines et de rationaliser le cycle de développement de la partie back-end de l'application
- Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers) de la partie back-end de l'application
- Automatiser la construction de l'application en configurant la chaînes de build, l'exécution des tests, l'exécution des outils d'assurance qualité et l'analyse statique de sécurité.
- **Faire évoluer ses connaissances et compétences en développement backend**
- Formaliser et mettre en oeuvre une démarche de veille technologique
- Utiliser un outil de knowledge management afin de capitaliser la connaissance et faciliter sa transmission
- **Préparer, mettre en oeuvre et automatiser la mise en production d'une application dans différents types d'environnements techniques d'exploitation**
- Identifier les différents types de plateformes d'hébergement et sélectionner une plateforme d'hébergement pertinente et adaptée aux exigences techniques de l'application (dimensionnement) en prenant en compte l'optimisation des coûts
- Définir l'architecture, mettre en oeuvre et administrer des services d'hébergement dans un environnement dédié, cloud ou conteneurisé en appliquant les bonnes pratiques de sécurité
- Mettre en oeuvre et exploiter un système d'orchestration de conteneurs
- Mettre en oeuvre un environnement de qualification externe
- Élaborer un diagramme de déploiement et mettre en oeuvre un système de déploiement automatisé d'une application pour différents types d'hébergement, avec ou sans processus d'intégration continue, en respectant les bonnes pratiques DevOps



- Automatiser la mise en oeuvre des architectures à l'aide d'un gestionnaire de configuration ou d'un système d' « Infrastructure as Code »
- Gérer des noms de domaines, administrer les serveurs de noms de domaine, et mettre en oeuvre des certificats de sécurité
- **Sécuriser et superviser l'application et l'environnement technique d'exploitation afin d'en assurer le maintien en production**
- Sécuriser les services et les réseaux d'une architecture d'hébergement
- Mettre en oeuvre les liaisons sécurisées entre les différents services et les composants de l'application
- Mettre en oeuvre des systèmes de sauvegarde et de redondance de l'infrastructure d'hébergement
- Identifier les métriques et indicateurs de performance de l'application et des services d'hébergement correspondants; définir un « budget » performance
- Réaliser un audit de performance d'une application et des services d'hébergement correspondants
- Rédiger un rapport d'audit de performance de l'application et des services d'hébergement correspondants; émettre des préconisations
- Préparer et déployer des mesures correctives pour l'amélioration des performances de l'application et de son environnement
- Mettre en oeuvre un système de journalisation et implémenter une stratégie d'audit des journaux de l'application
- Mettre en oeuvre un système de supervision d'une application et de services d'hébergement, définir des sondes et des alertes (état des services et sécurité)
- Détecter, diagnostiquer et analyser l'origine des bugs et problèmes techniques rencontrés
- Préparer et déployer des mesures correctives pour les erreurs ou les failles de sécurité identifiées dans des délais adaptés
- **Produire la documentation technique de l'application et alimenter une base de connaissances afin de capitaliser l'information et d'effectuer un transfert de compétences auprès de ses pairs**
- Commenter son code source en respectant les standards et bonnes pratiques en la matière afin de permettre la génération automatique de documentation
- Mettre en oeuvre la génération automatique de documentation du code source afin de l'intégrer dans un processus d'intégration continue
- Rédiger la documentation technique d'un projet afin d'effectuer un transfert de compétences auprès de ses pairs
- Rédiger la documentation d'une API afin de faciliter sa consommation par des tiers

Le PHP procédural - SEMAINE 13

Au programme : PHP

Module 18 : Programmation procédurale en PHP

5 jours : 35h

La programmation procédurale est le paradigme de programmation le plus naturel pour un ordinateur. Dans ce module vous apprendrez les bases du langage PHP et produirez vos premiers sites dynamiques.



Objectifs du module :

- La syntaxe du PHP
 - La portée des variables et des fonctions en PHP
 - Les tableaux et tableaux associatifs
 - Les variables superglobales
 - Gérer les sessions et les cookies
 - Quelques API standards de PHP
- les bases du langage PHP

Le PHP orienté objets - SEMAINE 14

Au programme :PHP

Module 19 : Programmation orientée objets en PHP

5 jours : 35h

Si la programmation procédurale est conforme à la manière de fonctionner d'un ordinateur, la programmation orientée objets fait sens pour la manière de penser d'un être humain. Vous apprendrez les concepts de ce paradigme de programmation et saurez les implémenter en PHP.

Objectifs du module :

les concepts de la programmation orientée objet avec PHP

- Classe, objet & instanciation
- Les diagramme de classe (UML)
- Les méthodes magiques
- L'héritage
- La surcharge de méthodes
- Les portées
- Getters & setters
- Les classes abstraites

le patron de conception architectural MVC

- Concevoir un modèle
- Concevoir une vue
- Concevoir un contrôleur
- La réécriture d'URL
- Concevoir un routeur
- L'autoloading

Compétences :

- Utiliser le formalisme du langage de modélisation unifié UML et de UP afin de traduire les besoins de l'application



- Constituer le dossier de modélisation comprenant les diagrammes, les cas d'utilisation, les classes d'analyse et décrivant l'architecture logicielle n-tiers de l'application
- Utiliser des patrons de conceptions afin d'appliquer des solutions logicielles standards

Le SGBD MySQL - SEMAINE 15

Au programme : MySQL

Module 20 : Introduction aux bases de données MySQL

5 jours : 35h

À présent que vous savez réaliser un programme serveur avec PHP, il est utile de savoir comment stocker les données de manière persistante grâce aux bases de données. Dans ce module vous découvrirez le SGBD MySQL.

Objectifs du module :

les principaux concepts des BDD relationnelles

- Les concepts généraux du modèle relationnel
- Créer une base de données avec PhpMyAdmin
- Les types
- La syntaxe du langage SQL
- Concevoir des scripts SQL

comment modéliser une base de données

- Réaliser un MCD
- Dériver un MCD en MLD
- Implémenter un MPD

MySQL avec PHP - SEMAINE 16

Au programme : PHP • MySQL

Module 21 : Sécurité des bases de données

2 jours : 14h

Après avoir vu comment créer et manipuler une base de données, vous verrez dans ce module quels sont les problèmes de sécurité qui peuvent survenir et comment s'en prémunir.

Objectifs du module :

le concept d'injection SQL

- Comment fonctionne une injection SQL
- Quelle en est la cause ?
- Exploiter une SQLi simple



- Exploiter une SQLi en aveugle
- Y remédier avec un requêtes préparée

les mécanismes de sécurisation d'une base de données

- Les modules d'authentification
- Authentification et autorisation

Module 22 : Le stockage en PHP

3 jours : 21h

Les bases de données sont un moyen de faire persister des données. Il en existe d'autres, notamment les fichiers. Vous verrez dans ce module comment les manipuler et ce que l'on peut en faire.

Objectifs du module :

Les différents mécanismes de stockage server

- Connexion à une BDD avec PDO
- Réaliser des requêtes complexes
- SQLite
- Lire et écrire sur un fichier
- Les différents formats de stockage
- Créer et interpréter un fichier de configuration

L'administration système - SEMAINE 17

Au programme : Linux • Apache2 • OpenSSL

Module 23 : Introduction à la cryptographie

2 jours : 14h

La cryptographie est un sujet majeur des sciences de l'information et des pratiques de sécurité. Ce module est une introduction pratique à la cryptographie. Vous comprendrez les concepts derrière les certificats et saurez comment les produire.

Objectifs du module :

la cryptographie avec openssl

- Chiffrement symétrique
- Chiffrement asymétrique
- L'échange de clés avec diffie-hellman
- Hashage, somme de contrôle et signatures
- Les certificats X.509
- le protocole TLS 1.3
- Le handshake TLS 1.3
- Les suites cryptographiques



Module 24 : Configurer la stack LAMP

3 jours : 21h

PHP ne marche pratiquement jamais seul. La pratique courante est d'y faire appel par l'intermédiaire d'un serveur HTTP comme apache2. Vous apprendrez dans ce module à configurer apache2, PHP et MySQL pour la production.

Objectifs du module :

la configuration d'Apache2

- Installer et configurer Apache2
- Installer et configurer PHP
- Installer et configurer PHP-FPM
- Tester les performances avec l'utilitaire "ab"
- Sécuriser la configuration de PHP-FPM
- Installer le module php-mysql

L'écosystème PHP - SEMAINE 18

Au programme : PHP

Module 25 : Écosystème PHP

3 jours : 21h

Fort de ses longues années d'existence, PHP a vu se développer autour de lui un écosystème logiciel vaste et mature. Le gestionnaire de dépendances "composer" permet de simplifier l'accès à cet écosystème.

Objectifs du module :

le gestionnaire de dépendances composer

- Les archives PHP
- Créer une archive phar
- Le principe de gestionnaire de dépendances
- Les principales commandes

le gestionnaire de dépendances composer

Réaliser des tests avec PHPUnit

- Écrire ses tests avec PHPUnit
- Les principales commandes
- Installer un le framework PHPUnit

Semestre 4 Architectures micro-service : Node.js & MongoDB

Les objectifs du semestre :



Les architectures en micro-service que vous étudierez supportent les applications des grandes multinationales que vous utilisez au quotidien. À l'issue de ce dernier semestre vous connaîtrez les pratiques de pointe dans l'industrie. Cette architecture peut être mise en œuvre avec tous les langages. Vous l'appréhenderez cependant par le prisme de Node.js et MongoDB. Vous découvrirez une autre manière d'écrire du Javascript avec Node.js et le concept de base de données NoSQL avec MongoDB.

Les technologies utilisées lors du semestre :

- 1 Node.js : Node.js est un environnement d'exécution du Javascript en dehors du navigateur. Son API et son vaste écosystème le rendent propice au développement côté serveur.
- 2 Le SGBD MongoDB : MongoDB est un système de gestion de base de données de type NoSQL.
- 3 Docker : Docker est un service permettant de gérer des conteneurs. Ces derniers permettent un déploiement rapide des applications et un environnement de travail maîtrisé.
- 4 Le protocole OAuth2 : OAuth2 est un protocole dédié à la délégation. C'est un protocole majeur dans la construction d'une authentification dans le cadre d'une architecture en micro-service.
- 5 OpenID Connect : OpenID Connect est une surcouche au protocole OAuth2 dédié à l'identification des utilisateurs.

Compétences :

- **Mettre en oeuvre les outils et techniques permettant de respecter les aspects réglementaires au regard des données informatiques traitées et stockées**
- Identifier les données et traitements informatiques concernés par les lois, normes et règlements (CNIL, RGPD,...)
- Implémenter les moyens techniques et technologiques permettant d'assurer le respect des lois, normes et règlements (CNIL, RGPD,...) applicables aux données traitées par une application et stockées dans une base de données.
- Connaître et respecter les lois et règlements en matière de propriété intellectuelle au regard des composants logiciels et codes sources informatiques
- **Mettre en oeuvre les outils et techniques permettant de respecter les aspects réglementaires au regard de l'accessibilité numérique**
- Identifier, dans une application, les éléments de présentation (UX), traitements et contenus informationnels concernés par les obligations d'accessibilité numérique.
- Implémenter les moyens techniques et technologiques permettant d'assurer le respect des obligations d'accessibilité numérique (RGAA - Référentiel général d'amélioration de l'accessibilité).
- **Mettre en oeuvre les outils et techniques permettant de mesurer et de suivre les performances marketing d'une application**
- Identifier les outils et techniques adaptés et pertinents au regard du contexte d'une application afin de mesurer et de suivre ses performances marketing
- Implémenter les outils et techniques permettant de mesurer et de suivre les performances marketing d'une application (analytics)



- **Mettre en oeuvre les outils et techniques permettant d'améliorer les performances SEO (Search Engine Optimization / référencement naturel) d'une application logicielle**
- Identifier les critères techniques d'une application ayant un impact sur le référencement naturel : les facteurs bloquants et les facteurs favorisant
- Identifier et savoir utiliser les outils d'assistance au référencement naturel afin d'analyser les performances SEO d'une application
- Implémenter les outils et techniques favorisant la SEO
- Implémenter les outils et techniques permettant la SMO (Social Media Optimization)
- Prévoir un plan de migration SEO dans le cadre de l'évolution d'une application
- **Mettre en oeuvre les outils et techniques permettant de monétiser une application**
- Identifier les fournisseurs et solutions de paiement en ligne adaptés au contexte et à l'environnement d'une application
- Implémenter une solution de paiement en ligne adaptée au contexte et à l'environnement d'une application, en appliquant les principes standards de sécurité informatique
- Identifier les fournisseurs et solutions de monétisation (advertising) adaptés au contexte et à l'environnement d'une application
- Implémenter une solution de monétisation (advertising et marketing d'affiliation) adaptée au contexte et à l'environnement d'une application

Node.js - SEMAINE 19

Au programme : Node.js

Module 26 : Introduction à Node.js

5 jours : 35h

Javascript est un langage de programmation dépendant de son contexte d'exécution. Le navigateur en est un, Node.js en est un autre. Il vous permet d'utiliser Javascript pour manipuler les ressources de votre système plutôt que celles de votre navigateur.

Objectifs du module :

Connaître les bases de node.js

- Différence entre le JavaScript sur navigateur et sur Node.js
- L'API Node.js
- Les fonctions synchrones et asynchrones
- Réaliser des scripts simples avec Node.js

Connaître le concept d'API REST

- Créer une API REST avec Express.js
- Envoyer des données JSON
- Découverte des middlewares
- Configurer le CORS

MongoDB - SEMAINE 20



Au programme : MongoDB

Module 27 : Introduction à MongoDB

5 jours : 35h

MongoDB est un système de gestion de bases de données comme l'est MySQL. À la différence de ce dernier, MongoDB est un système dit "No SQL". La manière de concevoir le stockage dans ce contexte est très différent du modèle relationnel.

Objectifs du module :

la modélisation et l'implémentation des bases de données MongoDB

. Introduction aux BDD NoSQL

- Les différences avec les BDD relationnelles
- Installer MongoDB
- La syntaxe de MongoDB

le théorème CAP

- Les différents SGBD
- Consistency, availability & partition tolerance
- Faire un choix

Docker - SEMAINE 21

Au programme : Docker • Node.js

Module 28 : Introduction à la conteneurisation avec Docker

5 jours : 35h

Docker est un service permettant de gérer des conteneurs. Il permet d'exécuter des services dans un environnement isolé que contrôle le développeur. Il offre donc la possibilité d'exécuter simultanément différentes instances de PHP ou de Node.js par exemple.

Objectifs du module :

Connaître les bases de docker

- Introduction à la conteneurisation
- La terminologie de Docker
- Créer un Dockerfile
- Faire interagir différents conteneurs
- Faire persister des données
- Automatiser la création de conteneurs avec docker-compose

L'architecture micro-service - SEMAINE 22

Au programme - Docker • Node.js

Module 29 : Les architectures micro-services



3 jours : 21h

L'architecture en micro-service est un patron de conception architectural basé sur la séparation des préoccupations. Elle est très utilisée dans l'industrie du cloud et sert de back-end aux applications telles que Netflix ou Uber par exemple.

Objectifs du module :

le principe d'architecture en micro-services

- Séparation des préoccupations
- Les API Gateway
- Services stateless & stateful

Module 30 : Server-side rendering & prerendering

2 jours : 14h

Les applications web développées avec un framework souffrent de quelques inconvénients. Le fonctionnement même de ces framework rendent le premier rendu assez long et posent des problèmes de référencement. Les techniques de server-side rendering et de prerendering permettent de le résoudre.

Objectifs du module :

Le rendu côté serveur

- Différence entre SSR et prerendering
- Générer du HTML côté serveur depuis son code JSX
- Hydratation du client
- Implémenter le SSR avec Express.js
- Implémenter le prerendering

L'authentification - SEMAINE 23

Au programme : Docker • Node.js • OAuth2 • OpenID Connect

Module 31 : Introduction à l'authentification

5 jours : 35h

L'authentification, la délégation et l'autorisation constituent la base de toute bonne politique de sécurité. Dans ce module vous verrez ce qui différencie ces différents concepts et comment les mettre en oeuvre dans le contexte d'une architecture en micro-services.

Objectifs du module :

les bases du contrôle d'accès

- Différence entre délégation, authentification et autorisation
- Le contrôle d'accès dans les architectures micro-service
- Les stratégies de contrôle d'accès (MAC, DAC, RBAC, ABAC)

implémenter une politique de contrôle d'accès



- Qu'est-ce qu'un JWT ?
- la délégation avec OAuth2
- l'authentification avec OpenID Connect
- implémenter un système RBAC

Droit de l'informatique - SEMAINE 24

Au programme : Licenses • RGPD

Module 32 : Droit de l'informatique

3 jours : 21h

Les programmes sont considérés comme des œuvres de l'esprit. À ce titre ils sont protégés par le droit d'auteur et sont soumis aux réglementations tant nationales qu'européennes.

Objectifs du module :

Les différents régimes de protection des oeuvres

- Droit commun et droit civil
- Différence entre copyright et droit d'auteur
- Loi française et européenne en matière de droit d'auteur

Les licences

- Licence libre et licence propriétaire
- Licence copyleft et permissives
- Les différentes licences libres

La RGPD

- La notion de données personnelles
- Les droits des utilisateurs
- Les devoirs des hébergeurs de contenu

Le projet de création de site web

Le projet de création de site web est composé de plusieurs parties :

- 1 **projet opérationnel de création de site web**
- 1 écrit en lien avec l'alternance et le projet de création de site web (cf guide méthodologique mémoire)
- 1 soutenance (+ powerpoint de présentation + soutenances blanches)

Le projet de création de site web est réalisé sur les deux années de la formation. L'étudiant choisit le thème de son projet qui reste le même pour la validation du titre.

La formation/les modules et le projet avancent de concert. Le projet est divisé en étapes et chaque étape correspond à une semaine de formation. Les travaux pratiques réalisés en



formation servent autant à assimiler les notions qu'à avancer sur le projet avec l'aide du formateur.

Le projet opérationnel de création de site web

Partie I) 1ère année : Réalisation d'un site vitrine responsive avec un agenda en react

L'étudiant choisit son sujet / thème à la fin du premier trimestre. Le sujet doit être validé par Cloud Campus, de ce fait, les projets débiteront à partir de ce moment-là.

La première partie du projet, lors de la 1ère année, se réalise en deux temps :

Premier temps : Création du site vitrine

- Créer une maquette Figma + intégration responsive (1 mois)
- Organiser Scrum et Git pour avancer sur le développement (1 mois)
- Développer les fonctionnalités Javascript (2 mois)

Choisir une entreprise réelle ou fictive dans un secteur d'activité donné et réaliser un site vitrine qui présente l'activité de l'entreprise en prenant en compte les contraintes métier (menu pour un restaurant, dates et programme pour un évènement, tarifs et options pour une société de service, etc.).

Lors de la soutenance, l'étudiant présentera :

- L'historique de ses sprints
- Le dépôt git
- La maquette Figma
- Le code source

L'étudiant développera les fonctionnalités suivantes avec Javascript sans utiliser de bibliothèque externe :

- Un carrousel animé
- Un menu "dropdown"
- Une F.A.Q sous forme d'accordéon
- Un widget affichant les heures et les minutes

L'étudiant devra également développer un formulaire, il peut s'agir d'un formulaire de contact, de réservation, de demande de devis ou quoi que ce soit de pertinent par rapport au sujet choisi. L'étudiant vérifiera les entrées utilisateur en Javascript et fournira un feedback visuel lorsque la saisie est correcte ou incorrecte.

Il faut également pouvoir ajouter des champs dynamiques au formulaire. Par exemple : ajouter des champs de contact ou bien des champs de date de réservation, etc.

Deuxième temps : Intégration d'un agenda en react

- Concevoir les composants, les vues et implémenter un routeur (1 mois)



- Implémenter Redux pour gérer l'état partagé entre vos composants (1 mois)
- Réceptionner les informations depuis l'API REST et irriguer l'application avec les données (1 mois)
- Configurer manuellement webpack et rédiger vos tests unitaires et d'intégration (1 mois)

Concevoir un agenda permettant de visualiser des événements.

Les événements seront réceptionnés depuis une API REST conçue par le formateur.

Votre agenda devra respecter le cahier des charges suivant :

- Afficher une vue des années, des mois et des semaines
- Afficher une vue pour un événement
- Afficher une liste des événements
- Sur la vue des années: afficher le nombre d'événements
- Sur la vue des mois : afficher les jours où il y a des événements
- Sur la vue des semaines : Afficher les heures auxquelles il y a un événement La navigation entre les vues devra se faire en cliquant sur les cases du calendrier.

Par exemple :

Vous voyez les années en arrivant sur le site, vous accédez aux mois d'une année en cliquant sur la case de l'année correspondante, puis vous accédez au jour du mois en cliquant sur la case du mois. Enfin vous accédez à la vue d'un événement en cliquant sur celui-ci dans la vue des jours.

Présentation devant le jury :

L'étudiant passe sa soutenance blanche devant un jury 2 mois après la fin de la première année.

La présentation se fera à distance en visioconférence **(20 min de présentation + 10 min de questions / réponses)**.

Cet examen est blanc (il ne compte pas dans la validation du titre) → toutefois, il est important de prendre son projet au sérieux car il s'agit d'un entraînement en conditions réelles pour le jury final.

Partie II) Réalisation d'un blog simple avec un back-office

Énoncé :



Concevoir un blog avec PHP et MySQL. Pour ce faire, vous utiliserez le patron architectural MVC.

Le front-office de votre blog doit comporter les pages suivantes :

- Accueil
- Historique des articles
- Article
- Inscription et authentification

Votre page d'accueil devra contenir :

- Un paragraphe de présentation du blog
- Une présentation de tous les auteurs
- Les 6 derniers articles
- Les 6 articles les plus aimés

La page d'historique des articles devra par défaut afficher tous les articles triés par date de parution et devra proposer des filtres pour rechercher précisément un article selon certains critères. Les filtres suivants doivent être implémentés et peuvent être combinés :

- Catégorie
- Chaîne de caractères (à chercher dans le nom)
- Le nombre minimum de "j'aime"
- Une date de publication minimum et maximum

La page d'article devra afficher :

- Le titre
- La date de parution
- L'auteur
- Les catégories auxquelles il appartient
- Le nombre de "j'aime"
- Tous les commentaires
- Les derniers articles de la même catégorie

Les utilisateurs non authentifiés doivent pouvoir ajouter un like et les utilisateurs authentifiés doivent pouvoir publier un commentaire.

Le back-office de votre blog ne doit être accessible qu'aux utilisateurs ayant le privilège "éditeur".

Il doit permettre :

- D'éditer la présentation du blog
- D'écrire un brouillon d'article
- De publier un brouillon
- De supprimer des commentaires sur un article
- De créer des catégories
- De passer un utilisateur en tant qu'éditeur
- De modifier sa propre présentation d'auteur



Il devra être sécurisé avec HTTPS, activer la réécriture d'URL et être configurable via un fichier de configuration.

Lors de la soutenance, vous présenterez un MCD et un MLD de votre base de données ainsi qu'un script pour la déployer rapidement avec des valeurs par défaut.

4 étapes :

- Mettre en place l'architecture MVC (1 mois)
- Modéliser et implémenter la base de données (1 mois)
- Sécuriser la BDD, créer le fichier de configuration et continuer le développement (1 mois)
- Configurer la réécriture d'URL et le HTTPS avec un certificat auto signé (1 mois)

Présentation devant le jury :

Vous passerez devant un jury 1 mois après la fin du semestre de deuxième année.

La présentation se fera à distance en visioconférence **(20 min de présentation + 10 min de questions / réponses)**.

Cet examen est blanc (il ne compte pas dans la validation du titre) → toutefois, il est important de prendre son projet au sérieux car il s'agit d'un entraînement en condition réelle pour le jury final.

Partie III) 2ème année : Réaliser une application de paris sportifs

Enoncé :

Vous réaliserez un front-end simple avec React pour réaliser des paris sportifs.

Le front-end devra consommer les données issues d'une API que vous créerez en node.js avec MongoDB. L'API devra aller chercher les données depuis une API existante donnant des résultats sportifs.

L'API devra implémenter les fonctionnalités suivantes :

- Inscription
- Connexion
- Voir les matches à venir
- Voir les résultats des matches passés
- Voir la composition des équipes qui jouent dans un match
- Parier sur un match
- Lister les paris passés et leurs résultats
- Gérer un profil utilisateur



Chaque fonctionnalité devra faire l'objet d'une route :

- Vous devrez réaliser des tests d'intégration sur vos routes
- L'authentification devra être réalisée avec OAuth et OpenID Connect
- Les journaux & les sauvegardes devront être envoyés sur un serveur externe
- Les sauvegardes devront être réalisées périodiquement et automatiquement

5 étapes :

- Créer les routes de l'API avec Node.js et commencer le développement de front (1 mois)
- Implémenter la base de données (1 mois)
- Conteneuriser les services (1 mois)
- Séparer l'application en micro-services et créer un serveur pour le serveur-side rendering (1 mois)
- Implémenter le système d'authentification (1 mois)

Critères de notation : voir guide méthodologique

Présentation devant le jury :

Vous passerez devant un jury 1 ou 2 mois après la fin de la deuxième année.

La présentation se fera en présentiel (20 min de présentation + 10 min de question / réponse = 30 minutes). + délibération/harmonisation par les membres du jury

L'étudiant devra se munir d'une présentation Powerpoint et d'un dossier écrit présentant le projet dans son ensemble sous forme de mémoire (cf guide méthodologique).

Composition du jury :

- représentants entreprise / maître d'apprentissage / professionnel
- représentants école

le temps de présence par jury (comprenant la préparation, le face à face avec le candidat, les corrections :

45 minutes par candidat : 5 minutes de relecture du mémoire professionnel papier, 20 minutes de présentation, 10 minutes de questions/réponses, 10 minutes de délibération et de notation.

Les conditions d'obtention du titre

Le passage de la soutenance, le projet, la remise du mémoire professionnel écrit et la note du maître d'apprentissage déterminent la validation du titre.



Si le mémoire professionnel écrit est rendu en retard, la pénalité suivante s'applique : -0.5 par jour de retard.

L'étudiant devra obtenir une note supérieure ou égale à 10/20 à l'ensemble que compose le mémoire professionnel écrit, le projet, la soutenance et la note du maître d'apprentissage pour valider le titre.

En cas d'échec, une session de rattrapage sera organisée.

Secteurs d'activité et types d'emplois

Secteurs d'activités :

Entreprise de services numériques (ESN ou ex-SSII) réalisant des prestations de développement d'applications, en régie ou au forfait.

Structure utilisatrice, de type entreprise du secteur privé ou public, possédant un service dédié aux études et aux développements informatiques.

Activité d'informaticien d'études indépendant.

Type d'emplois accessibles :

Concepteur développeur, concepteur d'applications informatiques.

Développeur d'applications, développeur informatique.

Développeur web, développeur back-end.

Développeur d'applications mobiles, développeur web mobile.

Ingénieur d'études et développement.

