

Technologies web

Fiche d'unité

21 décembre 2023

Connaissances préalables recommandées

La personne qui étudie doit connaître et savoir utiliser les notions suivantes :

- programmation orientée objet ;
- programmation réseau (sockets, utilisation de TCP et UDP, utilisation de HTTP).

Les unités POO et DAI permettent d'acquérir ces connaissances.

Objectifs

Depuis sa création au début des années 90, le Web ne cesse d'évoluer. Riche en liberté et en innovation, il est devenu une plateforme de développement incontournable. Les paradigmes qui le composent ont chacun eu leur moment de gloire. De pages statiques en passant par des applications dynamiques, le Web a su s'adapter aux besoins du marché. Avec l'explosion des services mobiles, des réseaux sociaux et plus récemment de l'Internet des Objets, les vagues d'innovation dans le domaine du Web se succèdent les unes aux autres. L'objectif principal de cette unité est d'étudier les grandes tendances passées et actuelles, d'appréhender les principes, les standards, les technologies et les outils qui sont en train de s'imposer sur le marché. Tout en étudiant les bonnes pratiques du domaine et en déployant des solutions adaptées aux besoins.

À l'issue de cette unité d'enseignement, la personne qui étudie sera capable de :

Statique

- illustrer les différentes parties qui composent l'internet (CDN, DNS, Conceptual model) ;
- créer des pages Web statiques (HTML, CSS, Javascript) ;
- proposer des manières de communiquer avec de la clientèle (Wireframe, Penpot) ;
- réaliser une interface utilisateur à l'aide de librairie de composants (Bootstrap) ;
- appliquer les bonnes pratiques du travail en équipe (Git, EditorConfig, Extension IDE) ;
- déployer un site statique ;
- expliquer les avantages et les inconvénients de générer un site statique (Hugo, Jekyll, etc.) ;

Dynamique

- expliquer la différence entre statique et dynamique ;
- comprendre l'écosystème Node.js (asynchronicité, Event loop, modules) ;
- appliquer les bonnes pratiques (TypeScript, Formatter, Linter, Git, etc.) ;
- développer une application CRUD avec un framework (NestJS) ;
- créer des pages Web dynamique à l'aide du patron MVC et d'un langage de templating (Nunjucks) ;
- créer des formulaires et les valider ;
- évaluer à quel point une application web est optimisée pour les moteurs de recherche (SEO) ;
- déployer une application NestJS ;

Stack Node.js

- expliquer la cission entre le frontend et le backend ;
- concevoir des interfaces utilisateur avec un framework réactif (React) ;
- comprendre l'architecture REST (client-server, statelessness, cacheable, uniform interface, layered system, code on demand) ;
- choisir quelle paradigme utiliser pour communiquer sur le réseau (Fetch, Server-Sent Events, Websockets, WebRTC, etc.) ;
- programmer des applications utilisant le réseau (Fetch, Websockets) ;
- écrire des tests automatisés pour des applications Node.js ;
- sécuriser une API à l'aide d'une API Key ;
- déployer une application Node.js ;

Cette unité est organisée avec une orientation très pratique. Les concepts présentés sont appliqués pour réaliser des applications Web de plus en plus conséquentes.

Contenu et formes d'enseignement

Cours - Statique	18
illustrer les différentes parties qui composent l'internet	1
créer des pages Web statiques	5
proposer des manières de communiquer avec de la clientèle	3
réaliser une interface utilisateur à l'aide de librairie de composants	2
appliquer les bonnes pratiques du travail en équipe	1
déployer un site statique	2
expliquer les avantages et les inconvénients de générer un site statique	2
Test 1 : Statique	2

Cours - Dynamique	18
expliquer la différence entre statique et dynamique	1
comprendre l'écosystème Node.js	4
appliquer les bonnes pratiques	2
développer une application CRUD avec un framework	3
créer des pages Web dynamique à l'aide du patron MVC et d'un langage de templating	2
créer des formulaires et les valider	2
évaluer à quel point une application web est optimisée pour les moteurs de recherche	1
déployer une application NestJS	1
Test 2 : Dynamique	2

Cours - Stack Node.js	21
expliquer la cission entre le frontend et le backend	1
concevoir des interfaces utilisateur avec un framework réactif	5
comprendre l'architecture REST	2
choisir quel paradigme utiliser pour communiquer sur le réseau	2
programmer des applications utilisant le réseau	3
écrire des tests automatisés pour des applications Node.js	2
sécuriser une API à l'aide d'une API Key	1
déployer une application Node.js	3
Test 3 : Node.js	2

Laboratoire	39
Partie 1 : Statique	12
Partie 2 : Dynamique	12
Partie 3 : Stack Node.js	15

Répartition pour chaque cours (nombre de périodes)

Statique

- Introduction (1), HTML + CSS (4), JavaScript (1)
- Wireframe + Penpot (3), Practical work 1 (3)
- Bootstrap (2), Git + EditorConfig + Extension IDE (1), Practical work 1 (3)
- GitHub Pages (2), Hugo (2), Practical work 1 (2)
- Evaluation (2), Practical work 1 (4)

Dynamique

- Introduction (1), Node.js Ecosystem + Advanced JavaScript + TypeScript (5)
- Formatter + Linter (1) + NestJS (3) + Practical work 2 (2)
- MVC + Nunjucks (2), Forms + validation (2), Practical work 2 (2)
- SEO (1), Deployment (1), Practical work 2 (4)
- Evaluation (2), Practical work 2 (4)

Stack Node.js

- Introduction (1), React (5)
- REST (2), Fetch + Server-Sent Events + Websockets + WebRTC (2), Practical work 3 (2)
- WebSockets + WebRTC (3), Practical work 3 (3)
- Jest (2), API key (1), Practical work 3 (3)
- Deployment (3), Practical work 3 (3)
- Evaluation (2), Practical work 3 (4)

Contrôle de connaissances

Cours

L'acquisition des matières de cet enseignement sera contrôlé au fur et à mesure par des tests tout au long de son déroulement. Avant chaque cours, les personnes rendent une série de questions sur le cours précédant qui fera une note complète à la fin de la partie. Il y aura au moins 2 tests d'une durée totale d'au moins 3 périodes.

Le matériel autorisé pour chaque est :

- de quoi écrire ;
- une feuille de résumé recto-verso.

Les feuilles recto-verso seront réutilisables pour l'examen.

Laboratoire

Ils seront évalués sur la base de réalisation d'applications web, ainsi qu'une présentation orale, à 3 reprises au minimum.

Examen

L'ensemble des objectifs de formation sera vérifié lors d'un contrôle final commun écrit d'une durée de 90 minutes.

Le matériel autorisé est :

- de quoi écrire ;
- Une feuille de résumé recto-verso par test effectué durant le semestre.

Note finale

Cours	30%
Laboratoire	20%
Examen	50%