

FORMATION DÉVELOPPEUR WEB ET WEB MOBILE

DOSSIER CURSUS

SESSION 2023

Maxime Lemée

Table des matières

1 Front-End.....	4	1.4. Développer une interface utilisateur web dynamique.....	28
1.1. Maquetter une application.....	5	1.4.1 Algorithmique appliquée.....	28
1.1.1 Analyse informatique / UML.....	5		
Définitions.....	5	Définition.....	28
Cas d'usage.....	5	Cas d'usage.....	28
Diagramme des Cas d'utilisation.....	5	Chaîne de caractère.....	28
Diagramme d'Activité.....	6	Opérations mathématiques.....	29
Diagramme de Classe.....	7	Variables.....	29
1.2. Gestion de projet.....	9	Booléens 0 / 1.....	29
1.2.1 Gestion de projet.....	9	Les fonctions : function(){ }.....	29
Définitions.....	9	Les Tableaux[].....	29
1.2.1.1 Étapes projet.....	10	Les conditions if() { }.....	30
1.2.1.2 Méthodologie (à visée informatique).....	11	Les boucles for(){} /while(){}.....	30
Cas d'usage.....	12	La programmation orientée objet.....	31
1.2.2 Maquetter une application.....	13	1.4.2 JavaScript.....	32
Définition.....	13		
Cas d'usage.....	13	Définition.....	32
1.3. Réaliser une interface utilisateur web statique et adaptable.....	15	Cas d'usage.....	32
1.3.1 HTML.....	15	Travaux pratique.....	33
Définition.....	15	1.4.3 VUE.....	35
1.3.2 CSS.....	15		
Définition.....	15	Définition.....	35
Bootstrap.....	16	Cas d'usage.....	35
Cas d'usage.....	16		
1.3.3 Outil de versionning / Git.....	19	1.5. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.....	37
Définition.....	19	1.5.1 Créer un site avec Wordpress.....	38
Cas d'usage.....	19		
TP.....	20	Définition.....	38
1.3.4 Référencement.....	23	Cas d'usage.....	38
Définitions SEO SEA.....	23		
Cas d'usage.....	24	2 Back-End.....	40
Black hat.....	25	2.1. Créer une base de données.....	41
L'Open Graph.....	25	2.1.1 Base de donnée.....	41
Les données structurées.....	26		
Les Backlinks.....	26	Définition.....	41
Le maillage interne.....	26	Méthode MERISE MCD / MLD.....	42
Les Webs Analytics.....	27	MCD : conception <i>conceptuel des données</i>	42
		MLD Modèle logique des données	42
		MYSQL avec MySQL Workbench.	43
		Cas d'usage.....	45
		2.2. Développer les composants d'accès aux données.....	46
		2.2.1 PHP.....	46

Définition.....	46	Définition.....	63
Cas d'usage.....	46	Cas d'usage.....	63
Opérations algorithmiques.....	46		
Cookies et Session utilisateur.....	47		
Classe et objet.....	47	2.4.2 Administrer et personnalisé	
Méthode GET et POST.....	48		
Base de données.....	49	WORDPRESS.....	67
2.2.2 Symfony.....	50		
Définition.....	50	Cas d'usage.....	67
Cas d'usage.....	50	Hook.....	68
Test.....	56		
2.3. Développer la partie back-end d'une application Web ou Web Mobile.....	57	3 Compétences transverses.....	69
2.3.1 Serveur Web (LAMP).....	57	3.1. Sécurité.....	69
Définition.....	57	3.1.1 Définition.....	69
Cas d'usage.....	57	3.1.2 Cas d'usage.....	69
Serveur apache.....	59	Injection (SQL, Shell,...).....	70
2.3.2 Docker.....	60	Cross-site scripting (XSS).....	70
Définition.....	60	3.1.3 TP.....	71
Cas d'usage.....	60		
Docker compose.....	62	3.2. RGPD.....	73
2.4. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.....	63	3.2.1 Le registre.....	74
2.4.1 Créer une e_boutique avec Prestashop.....	63	3.2.2 LA CNIL.....	75
		3.3. Veille technologique.....	76
		3.3.1 Définition.....	76
		3.3.2 Cas d'usage.....	76
		3.4. Anglais.....	77
		3.4.1 Cas d'usage.....	77

1 Front-End

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

La partie Front-End correspond à toute la partie visuelle et interactive du site et/ou de l'application web.

L'ébauche est conceptualisée dans l'étape du maquettage et est ensuite codée avec du HTML pour le balisage, le CSS pour le style et le JAVASCRIPT pour la partie interactive du site.

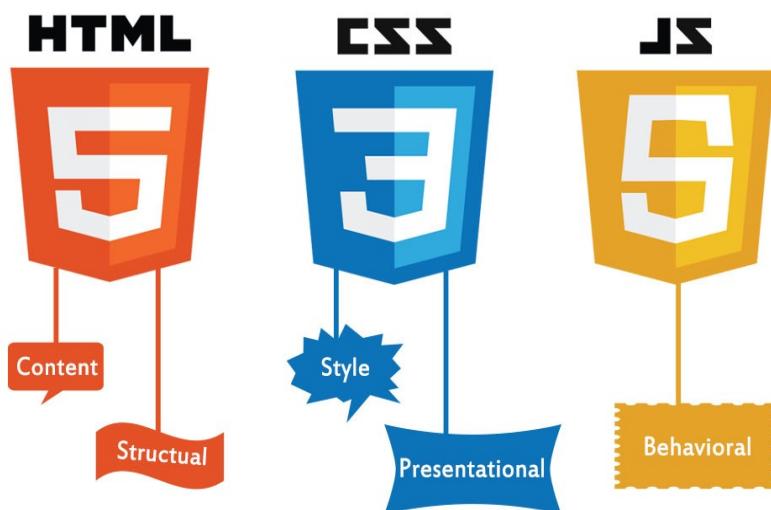


Figure 1: Le trio indissociable

1.1. Maquetter une application

1.1.1 Analyse informatique / UML

Définitions

UML (Unified Modelisation Language) est un langage de modélisation orienté objet qui permet d'établir un cadre méthodologique à base de diagramme centré sur l'architecture (logicielle) pilotée par les besoins des utilisateurs. C'est un langage utilisant une sémantique précise et reconnaissable par tous.

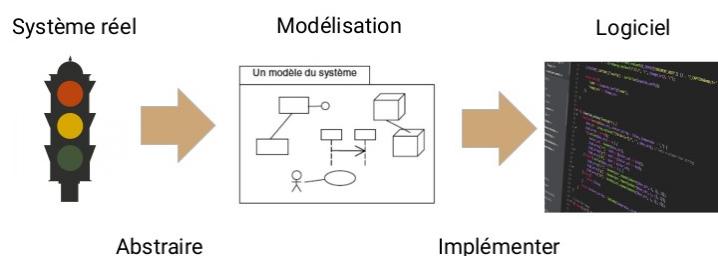


Figure 2: Modélisation UML

Cas d'utilisation

Diagramme des Cas d'utilisation

Nous avons commencé par voir le diagramme des cas d'utilisation avec un exercice pratique

C'est un diagramme permettant de cibler les besoins des utilisateurs en identifiant les acteurs et leurs cas d'utilisations.

Pour la Figure 3, Nous avions une demande sur un projet d'informatisation de la gestion RH d'une entreprise, avec la gestion des demandes de congé, des absences, des notes de frais et des bulletins de salaire.

On peut voir les différents cheminement de chaque utilisation selon l'acteur (Salarié, membre RH et comptable) avec chacun leur cas d'utilisation entouré ainsi que les « includes » : ce sont les obligations qu'une tâche incombe et les extends qui permettent de savoir ce qu'un cas peut permettre de faire.

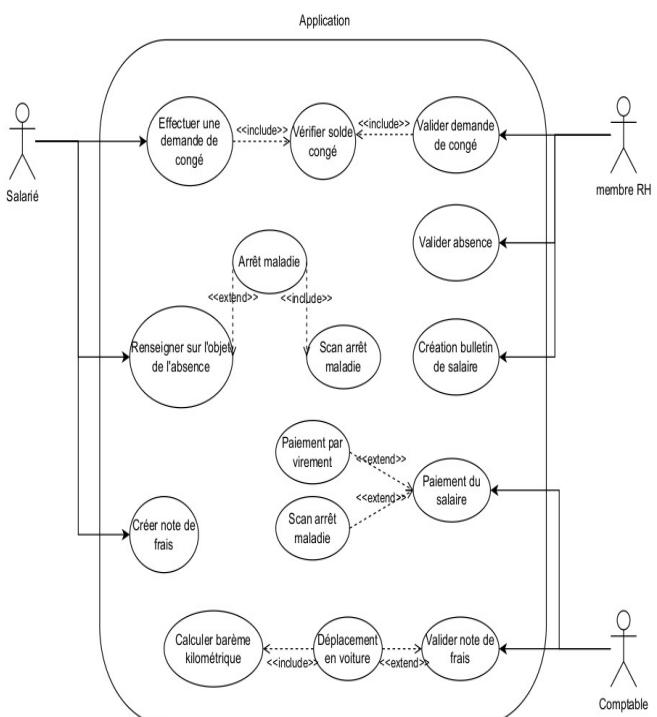


Figure 3: Diagramme des cas d'utilisation

Il y a également un système d'héritage entre acteur qui peut être appliqué . Dans l'exemple de la Figure 4, le chef d'équipe hérite du salarié, ce qui signifie qu'il peut effectuer en plus des siens tous les cas d'utilisation du salarié.

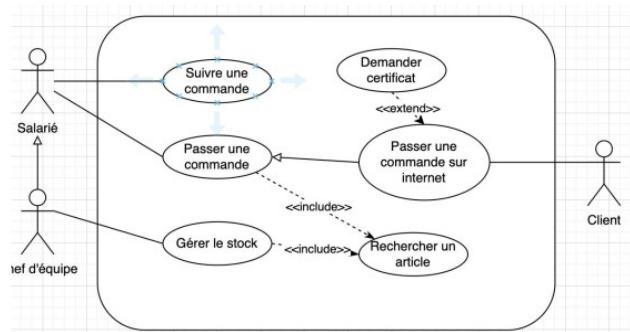


Figure 4: Exemple d'héritage cas d'utilisation

Diagramme d'Activité

Après, nous avons abordé le diagramme d'activité qui permet de représenter par une vue dynamique, pas à pas les flux d'activités centrés sur les actions afin d'obtenir une vision claire du cheminement de chaque tâche.



Figure 5: Diagramme de cas d'utilisation

Dans le même esprit que le diagramme des cas d'utilisation, nous avions un énoncé nous expliquant le contexte. Celui ci (Figure 5) nous demandait, dans le cadre d'une société, de gérer la détection d'incident ou d'intervention via des tickets avec une gestion de niveaux des différents techniciens.

Via les symboles correspondants (Figure 6) et pour plus de clarté, j'ai décidé de séparer le diagramme par acteur avec chacun leur colonne correspondant.



Figure 6: Symbole diagramme cas d'activité

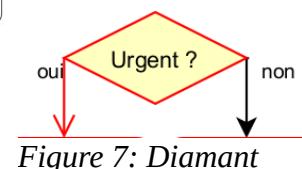


Figure 7: Diamant

J'ai également ajouté des diamants qui permettent d'effectuer des choix et des boucles quand cela est nécessaire .

Diagramme de Classe

Et pour finir, nous avons vu le diagramme de classe qui à pour but de représenter les différentes classes ainsi que leur relations.

Dans le cadre de nos exercices, nous devions automatiser la gestion d'une petite bibliothèque, c'est pourquoi nous devions étudier son fonctionnement et schématiser le diagramme de classe adéquat.

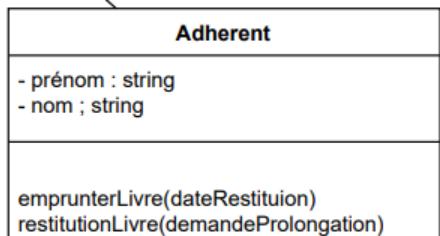


Figure 8: Exemple classe

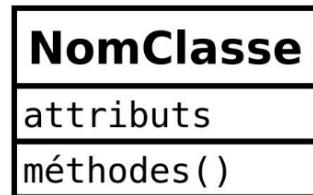


Figure 9: Modèle de classe

Il fallait prendre en compte toute les paramètres demandés comme par exemple le fait que les adhérents ont tous un prénom et un nom, l'ensemble des documents de la bibliothèque et leurs styles avec chacun leurs caractéristiques, l'inscription et la désinscription des adhérents et d'autres paramètres.

Ce qui donne le diagramme suivant (Figure 10)

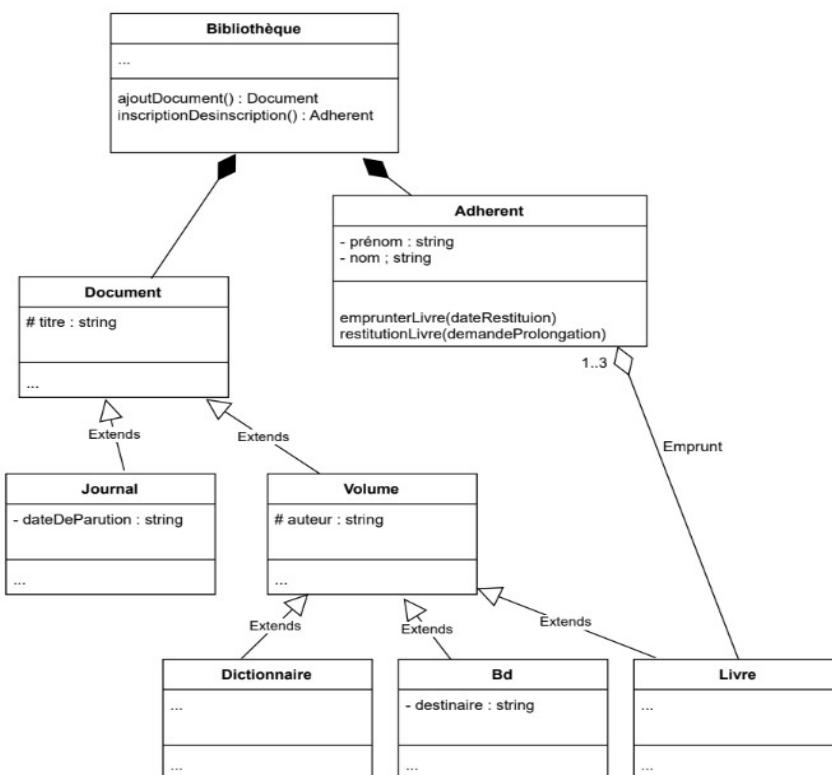


Figure 10: Exercice diagramme de classe

J'ai créé 8 classes et établi entre elles les relations nécessaires au bon fonctionnement de l'ensemble avec des relations d'agrégation (Figure 13 quand une classe peut contenir une ou plusieurs occurrences d'une autre classe) et de composition(Figure 12 idem que l'agrégation sauf que si l'une est détruite les classes liées le seront aussi)

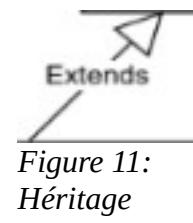
J'ai également mis les héritages entre les classes avec ce symbole (Figure 11), ce qui signifie que la classe enfant prend les attributs et les méthodes de la classe parent (celle qui est tombée flèche)

Certaines classes sont abstraites, c'est à dire qu'elle ne seront jamais instanciée : c'est le cas de la classe Document qui ne sert que pour ses enfants (Journal et Volume), de même pour Volume qui ne sert que pour ses enfants (Dictionnaire, Bd et Livre)

*Figure 13:
Agrégation*



*Figure 12:
Composition*



*Figure 11:
Héritage*

1.2. Gestion de projet

1.2.1 Gestion de projet

Tout projet, petit ou gros ,necessite un minimum d'organisation et d'informations afin d'aboutir le plus rapidement et le plus efficacement possible à terme.

C'est pourquoi il existe des outils et des méthodologies pour répondre à ce besoin comme des outils de veille, des méthodes de gestion de projet, des outils de versionning, et des diagrammes de conception pour toutes les étapes du projet qui viennent en amont de la programmation et qui permettent de bien penser au modèle dans sa globalité avant d'entrer la première ligne de code.

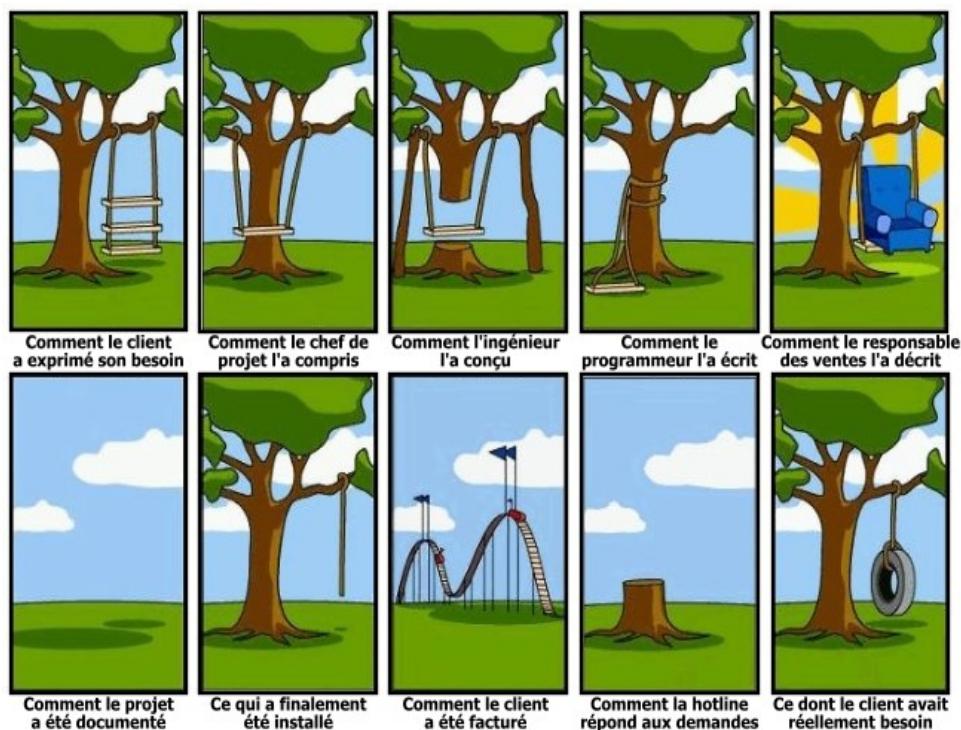


Figure 14: Exemple d'un mauvais suivi de projet !

Définitions

‘Gérer, c'est prévoir’

Gérer un projet sert à prévenir et à anticiper, durant la durée de vie, les demandes, les modifications, les risques grâce a des méthodes de management .

Étapes générale projet :

1. CDC ou briefing client avec un contexte, des problématiques et des objectifs
2. Étude commerciale ou de faisabilité en proposant au sein d'une synthèse, les solutions, un délai et un budget clair et compréhensible pour le donneur d'ordre.
3. OK ou NOK ?

4. Note de cadrage qui contiendra le contexte en 2 phrases, les objectifs du projet, les personnes concernées , le budget, les délais et les risques potentiels, le tout sur une page A4
5. CDC Détaillé et technique
 - Cahier des charges fonctionnel avec la description des fonctionnalités.
 - Valeur d'usage (valeur rajouté , se mettre à la place de l'utilisateur , facilité application)
 - Cahier des charges techniques (protocole de communication...etc, docker)

6. Planification GANTT et WBS

- WBS (Work Breakdown Structure), représentation graphique permettant d'estimer les plannings et les ressources nécessaires pour construire le projet : le projet est découpé en groupe de tâches ainsi qu'en sous groupes avec le temps alloué pour chaque tâche.(Figure 15)
- Planification de GANTT représentation dynamique des tâches créée en WBS

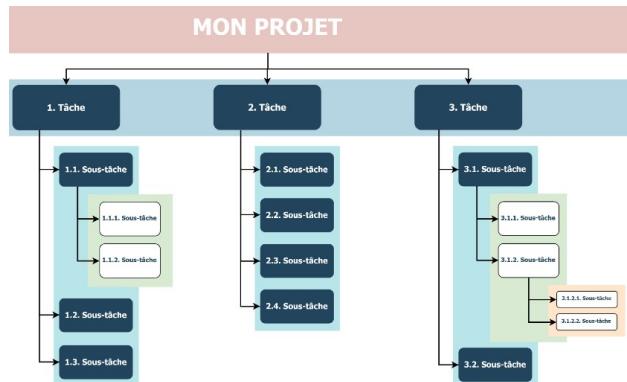


Figure 15: Représentation WBS

7. Documentation/tests/recette

- Recette : c'est un document à préparer en amont qui va lister les tests de l'application pour le client qui une fois effectués seront répertoriés sur un procès verbal de recette. C'est à dire que le client va donner son avis sur les tests effectués par rapport aux demandes du cahier des charges.

1.2.1.1 Étapes projet

- I. Ressources graphique de l'application , en interne ou externe si le besoin se présente.
- II. Organisation → livraison développement, base de données, sauvegarde, code, façon dont les mises à jours se feront ,choisir méthode de versionning
- III. Identifier les bons interlocuteurs, comment échanger? Mail, téléphone, réunion ?

Il faut surveiller les retards et être sincère et clair avec le donneur d'ordre ,se méfier des changements d'interlocuteurs, les projets qui traînent et qui font douter le donneur d'ordre.

1.2.1.2 Méthodologie (à visée informatique)

- Cycle en V

Rigide , à utiliser dans un contexte fonctionnel et technique stable

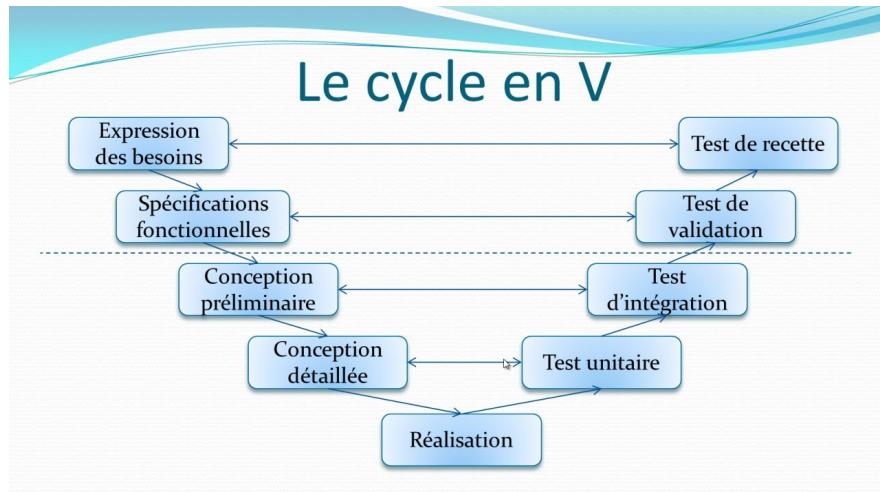


Figure 16: Cycle en V

- Agile – Modèle SCRUM

Méthode souple, à utiliser dans un contexte fonctionnel et technique instable

- Product owner qui est responsable des fonctionnalités (Backlog) et garant des délais
- Teams developer (autonome sur les choix technique) qui vont effectuer des sprints (1 ou plusieurs fonctionnalité du backlog)

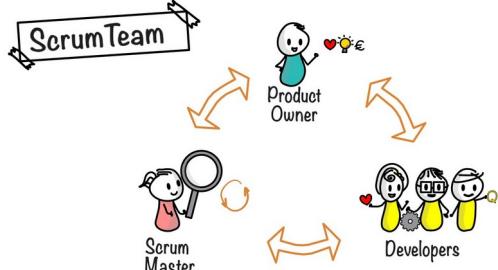


Figure 17: SCRUM Teams

On a ensuite abordé la gestion des risques dus aux retards de livraison, aux arrêts malades, l'obsolescence informatique ...etc. Ceux-ci peuvent être calculés via une matrice de criticité qui permettra de mesurer le degré de gravité du risque par rapport à la probabilité du risque et en fonction du résultat, mettre en place les mesures nécessaires les plus équilibrées possible, ni trop, ni pas assez.



Figure 18: Matrice des risques

Cas d'usage

Durant le cours, nous avons simulé la demande d'une application pour une société en interne .

La demande concernait la mise en place d'une application permettant de centraliser et de contrôler, via des niveaux d'accès, la création et l'accès aux fiches client et fournisseur ainsi que la visualisation des identifiants et mots de passe fournis au client et fournisseur pour leurs différents accès .

1. Dans un premier temps, au cours d'un entretien, nous avons pris des notes sur la demande. Nous avons réalisé dans la foulée le recueil de l'expression (Figure 19) qui est un document présentant une première ébauche de la demande en précisant la société, le contexte et les points à éclaircir.
2. Nous avons fait un brief oral et expliqué à notre « Équipe de développement » le but de l'application.
3. Nous avons par la suite réalisé la note de cadrage (Figure 20), c'est un document qui synthétise en quelques lignes la demande, le planning global, le budget et/ou les ressources humaines allouées.
4. L'équipe de développement a préparé les documents nécessaires à l'implantation, avec le maquettage de l'interface visuelle, les diagrammes UML, et le fichier SQL. Elle nous les a présentés une fois finis.
5. Nous avons, ensuite, réalisé le WBS() ainsi que le diagramme de GANTT(Figure 57) pour organiser sa conception au sein de l'équipe et estimer son temps de réalisation.
6. Et fini par écrire le cahier de recettes, c'est à dire un document qui sert à tester la conformité de l'application au moment de sa livraison. Son but est de confirmer ou non les points essentiels en les testant avec le client. Ce dernier décidera par la suite les corrections et remarques à apporter. Souvent , durant la conception d'un projet, il y a plusieurs cahiers de recettes permettant de valider des jalons important du projet .

Recueil de l'expression de besoin

Projet	Date
--------	------

L'entreprise

Le projet

Contexte et problématique

Les points à éclaircir

Figure 19: Exemple Recueil du besoin

Figure 20: Exemple Note de cadrage

Projet	Date
--------	------

Description

< Description de la raison d'être du projet et du bénéfice d'usage attendu >

Équipe

	Prénom, Nom	Rôle
Responsable		
Accountable		
Consulted		
Informed		

OU

Prénom Nom, fonction dans l'entreprise, rôle sur le projet
Prénom Nom, fonction dans l'entreprise, rôle sur le projet
Prénom Nom, fonction dans l'entreprise, rôle sur le projet
Prénom Nom, fonction dans l'entreprise, rôle sur le projet

Budget

Délai

Risque

1.2.2 Maquetter une application

Définition

Le maquettage consiste à créer une représentation visuelle du site selon la charte du client en simulant l'interactivité avec le futur utilisateur et en vérifiant le formfactor de l'ensemble suivant l'usage et le média sur lequel il sera diffusé.

Celui-ci peut se faire avec un logiciel dédié tel qu'adobe ou Figma  pour les plus connus qui permettent un échange avec le client qui peut valider ou proposer des modifications en simulant avec précision la navigation que l'on pourrait retrouver.

Une fois le design validé, le dossier est envoyé au développeur qui codera le résultat.



Cas d'usage

Durant le cours, nous avons conçu une maquette sur le sujet de notre choix, pour ma part, mon thème était un créateur de meuble « Living Colors »,

Figure 21: Page d'accueil

Avec le logiciel adobe XD, nous avons appris à s'organiser en commençant par la charte des couleurs, celle des différentes polices, la conception des composants comme le menu ou une fiche personne et un formulaire de contact.



Figure 22: Card

Dans un 2nd temps, nous avons créé le même site au format mobile avec les mêmes chartes mais conditionné au dimensions attendues.

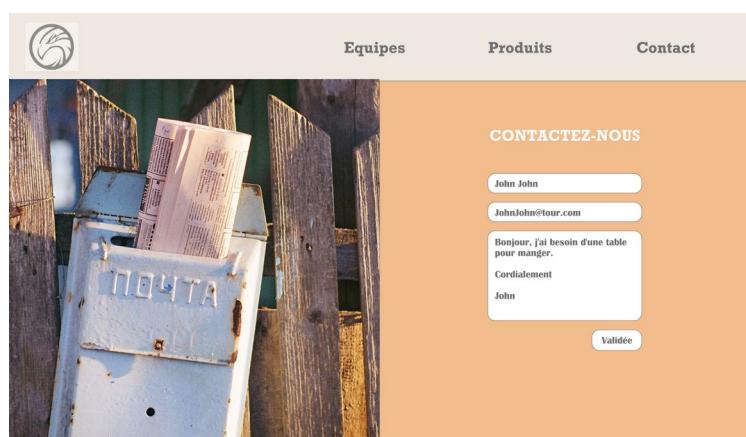


Figure 23: Page de contact

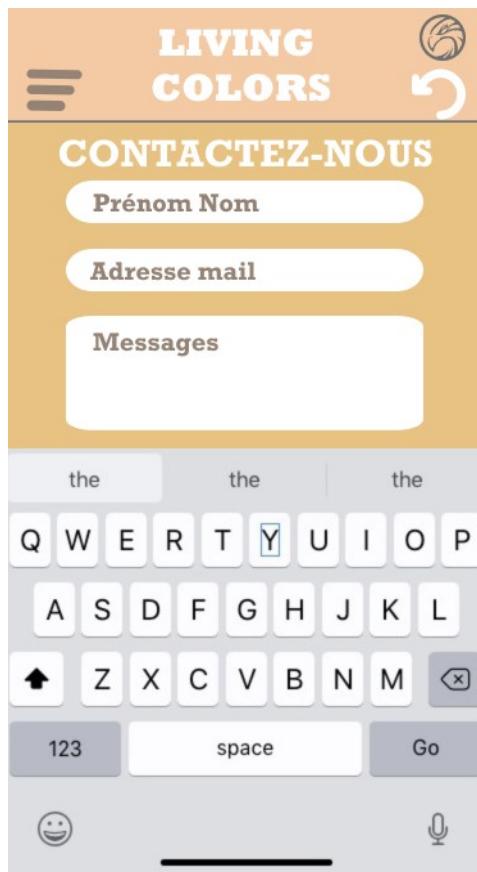


Figure 24: Contact mobile



Figure 25: Accueil mobile

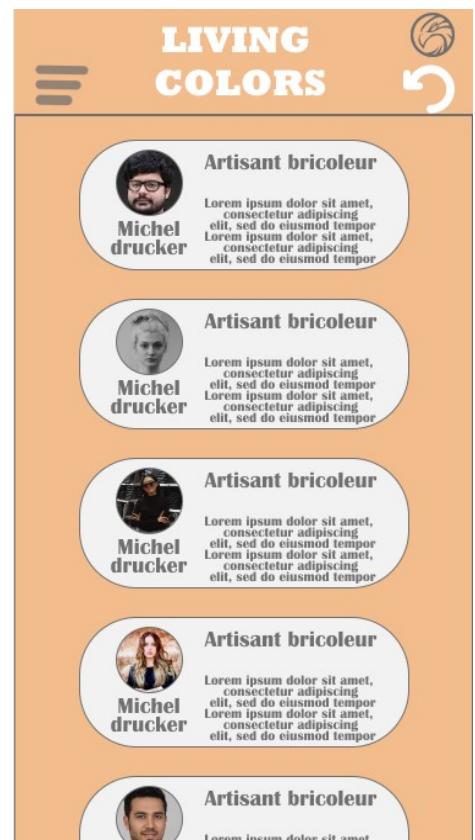


Figure 26: Card mobile

1.3. Réaliser une interface utilisateur web statique et adaptable

1.3.1 HTML

Définition

HTML qui signifie « HyperText Markup Language » est un langage qui permet à l'aide de balise de structurer sémantiquement une page internet et de mettre en forme son contenu, ce n'est pas un langage de programmation. Le navigateur affichera le contenu en prenant en compte les différentes balises présentes. Ce qui servira également au référencement du site par les moteurs de recherche comme google ou bing (Voir chapitre) Depuis 2014, HTML5 est la dernière version majeure.

Le HTML minimum dans un page internet est le suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre affiché dans la barre de titre du navigateur</title>
  </head>
  <body>
    <!-- C'est ici que vous mettrez votre contenu -->
  </body>
</html>
```



Figure 27:
Logo HTML

Comme exemple de balisage les plus courants, il y a les balises de Titre `<h1>` jusqu'à `<h5>` pour indiquer au navigateur l'importance du texte, les balises `<div>` qui enveloppe un bloc, la balise `<p>` pour intégrer un paragraphe, la balise `<form>` pour intégrer un formulaire, la balise `` pour afficher une image et encore bien d'autre.

1.3.2 CSS

Définition

CSS (feuilles de style en cascade ou Cascading Style Sheets) est un langage informatique permettant de mettre en forme le site internet grâce à des instructions que l'on va indiquer au navigateur web

- soit directement dans le fichier HTML en ciblant les éléments HTML que l'on veut styliser
- soit dans un fichier à part dans lequel on nommera avec des sélecteurs précisément le ou les éléments HTML ciblés



Figure
28: Logo
CSS

Si le style est directement intégrer au code html, il doit être entourer de balise <style> directement dans le sélecteur HTML

exemple: <html><p style="color: blue;"> Ceci est un paragraphe </p></html>

dans cette exemple, je change la couleur du texte du paragraphe en bleu ou alors soit dans un fichier à part ou l'on écrira plutôt ceci

```
exemple: p {  
    color : blue ;  
}
```

dans celui-ci, dans un fichier à part, j'ai sélectionné la balise HTML <p> et je lui ai appliqué la couleur bleu

Bootstrap

Boostrap est ce qu'on appelle un framework CSS qui facilite grandement l'implantation d'élément graphique au sein de notre application avec une gestion du responsive (c'est à dire rendre un site accessible et adaptable à tous les devices) grâce une bibliothèque d'élément prêt à l'emploi que l'on implante au sein du code HTML via des classes ,



Figure 29:
Logo
Bootstrap

Parmi celle là, on retrouve les buttons,

En implantant les classe 'btn btn-primary btn-secondary btn-success btn-alert ...etc ou les spinners



Figure 30: Boostrap Class BTN

avec les classes 'spinner-border text-primary text-secondary text-success ...etc



Figure 31: Bootstrap Class spinner

Cas d'usage

Pour le cours, en tant que fil rouge, nous avons réalisé un site de vente de biens immobiliers «quarter »

Pour le réaliser, nous avons utiliser un IDE,

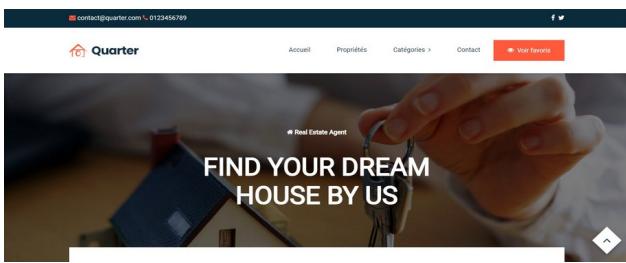
Un IDE est un logiciel qui aide à écrire du programme informatique grâce à une multitude d'options et d'aide comme l'auto complétion (le fait de proposer des mots ou partie de code à partir des premiers caractères tapés en fonction du code déjà présent) ou la coloration syntaxique très utile pour visualiser tous les détails du code qui peut parfois mélanger plusieurs langages, un débugueur intégré, du multi-fenêtres ou une grande quantité d'extensions pour améliorer son expérience au sein d'un projet. Pour les IDE les plus connus, on retrouve Visual code Studio, Sublim text ou NetBeans. Certains sont spécialistes dans leurs domaine comme PHPstorm(PHP) ou Code::block (C,C++)

Pour notre projet, nous avons utiliser Visual code studio .

Étape par étape, nous avons mis en place la disposition du site en commençant par le header de la page et la barre de navigation.

Pour ce faire, nous avons appris le principe de modèle de boîte FlexBox et de Grid qui permettent de placer les éléments sur la page avec une plus grande simplicité

exemple de code HTML qui a permis la création du site ,



on peut y voir la partie HEAD qui sert à informer le navigateur de toute les informations nécessaire sur le site avec la langue, le titre , les liens vers les différents feuille de style CSS et d'autre information.

Notre professeur nous a vivement conseillé pour tous nos futurs projet d'y intégrer une feuille de style spéciale qui s'appelle **normalize.css**, il s'agit d'une feuille de style créée pour corriger les défauts que l'on peut avoir sur les navigateur en corrigeant par exemple les marges ou les polices par défauts, elle est à insérer juste avant d'insérer notre propre feuille de style pour la simple raison que le navigateur lit de haut en bas donc commencera par implémenté le

normalize.css ET ensuite la notre .

Pour la gestion du CSS, nous avons installer un préprocesseur SCSS qui permet de fragmenter le code CSS dans plusieurs fichiers distincts(ce que le CSS nativement ne permet pas) et de pouvoir faire de la syntaxe indentée.

```
sass/
|- base/
|  |- _reset.scss      # Reset/normalize
|  |- _typography.scss # Typography rules
|  ...
|
|- components/
|  |- _buttons.scss    # Buttons
|  |- _carousel.scss   # Carousel
|  |- _cover.scss       # Cover
|  |- _dropdown.scss   # Dropdown
|  ...
|
|- layout/
|  |- _navigation.scss # Navigation
|  |- _grid.scss        # Grid system
|  |- _header.scss      # Header
|  |- _footer.scss      # Footer
|  |- _sidebar.scss     # Sidebar
|  |- _forms.scss        # Forms
|  ...
|
|- pages/
|  |- _home.scss        # Home specific styles
|  |- _contact.scss     # Contact specific styles
|  ...
|
|- sass-utils/
|  |- _variables.scss   # Sass Variables
|  |- _functions.scss    # Sass Functions
|  |- _mixins.scss       # Sass Mixins
|  |- _helpers.scss      # Class & placeholders helpers
|
|- vendors/
|  |- _bootstrap.scss    # Bootstrap
|  |- _jquery-ui.scss    # jQuery UI
|  ...
|
- style.scss          # Primary Sass file
```

Figure 33: SCSS Architecture Structure

```
/*_backgrounds.scss */
...
$use ../../utils/variables;
$use ../../utils/variables;

$bg-primary {
  background-color: variables.$color-primary;
  color: $white;
}

a {
  color: $white;
}

&:hover {
  color: variables.$color-secondary;
}

/*_typography.scss */
...
$use ../../utils/variables;
$use ../../utils/variables;

@import url("https://fonts.googleapis.com/css2?family=Roboto:wght@400;700;900&display=swap");
html {
  font-family: "Roboto", sans-serif;
  scroll-behavior: smooth;
}

body {
  height: 3000px;
}

h2 {
  font-size: 3rem;
  color: variables.$color-primary;
}

a {
  color: variables.$color-primary;
  text-decoration: none;
  transition: 0.5s;
}

&:hover {
  color: variables.$color-secondary;
}

/*_buttons.scss */
...
$use '/scss/utils/variables';

.btn {
  display: flex;
  gap: 0.5rem;
  padding: 1rem 2rem;
  border: 2px solid transparent;
  cursor: pointer;
  color: $red;
  text-decoration: none;
  transition: 0.5s;
}

.btn-primary {
  color: $red;
  background-color: variables.$color-primary;
}

&:hover {
  color: variables.$color-primary;
  background-color: $white;
  border-color: variables.$color-primary;
}

.btn-secondary {
  color: $white;
  background-color: variables.$color-secondary;
}

&:hover {
  color: variables.$color-secondary;
  background-color: $white;
  border-color: variables.$color-secondary;
}
```

37

Figure 32: Exemple de code SCSS fragmenté et indenté

Nous avons également vu la création de variable CSS qui permettent de changer rapidement la charte graphique d'un ensemble sans avoir à changer à tous les endroits du code où l'on s'en sert, on pourra désormais l'appeler et la comprendre grâce à un nom de variable compréhensible .

```
@use "../utils/variables";
```

```
.list-properties {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 2rem;
}
```

```
.property-card {
  display: flex;
  flex-direction: column;
  box-shadow: rgba(100, 100, 111, 0.2) 0px 7px 29px 0px;
```

```
&:hover {
  .property-card-header {
    img {
      transform: scale(1.1);
    }
  }
}
```

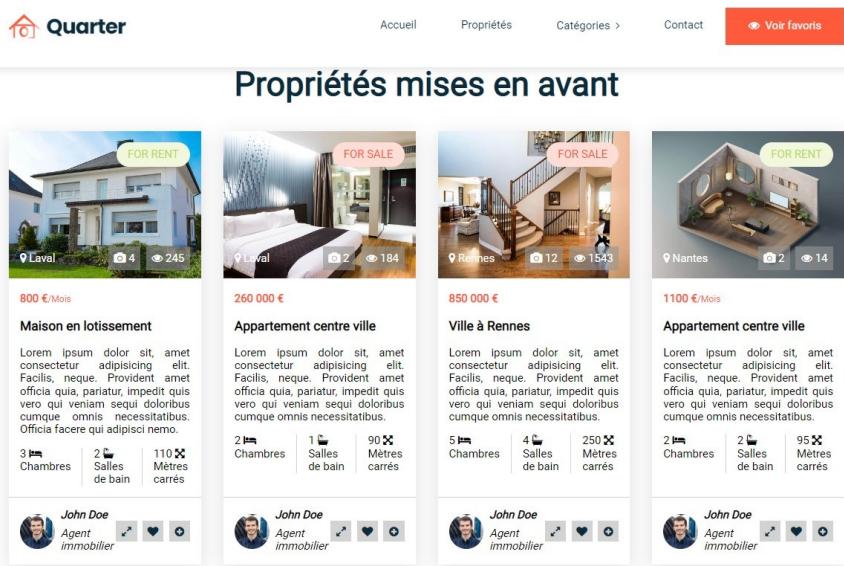
```
.property-card-header {
  position: relative;
  overflow: hidden;
```

```
img {
  width: 100%;
  transition: transform 0.5s;
```

```
.badge {
  position: absolute;
  top: 1rem;
  right: 1rem;
}
```

```
.header-info {
```

Dessin 1: Exemple de code Figure 34: Exemple de création 'property-card' CSS property



Pour les différentes installations nécessaires au bon fonctionnement et à l'affichage du site, nous avons utilisé le gestionnaire de paquet NPM qui permet l'installation de dépendances à la racine du projet à l'aide de lignes de commandes.

Pour Quarter, nous avons installé SASS via la commande `npm install -g sass` ainsi que webpack via `npm install --save-dev webpack webpack-cli`, ce dernier sert à compiler tout le projet en fichier exécutable par le navigateur car le langage SCSS ne sera pas reconnu par google chrome ou Firefox. Webpack possède également un serveur intégré qui nous permet de visualiser en temps réel sur navigateur le rendu de nos modifications.

1.3.3 Outil de versionning / Git

Définition

Git est un logiciel de gestion de version décentralisée créée en 2005 par Linus Torvalds (créateur également du noyau Linux), il est connu pour être facile à apprendre, très léger et performant grâce à ces gestions de branches, de numéro de version et de tag, il permet à tout moment de revenir à des points précis du développement, points que l'on appelle commit.

Au sein d'un groupe, Git facilite le travail entre les différents acteurs sur un même projet en limitant l'impact que peuvent avoir plusieurs développement de fonctionnalité ou de fixation de bug en parallèle pour ensuite les incorporer au fur et à mesure dans la branche principal.

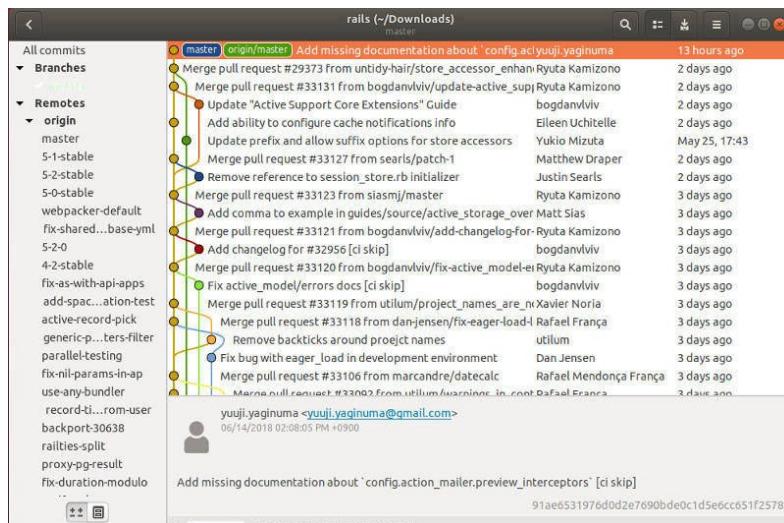


Figure 35: Exemple de projet utilisant Git

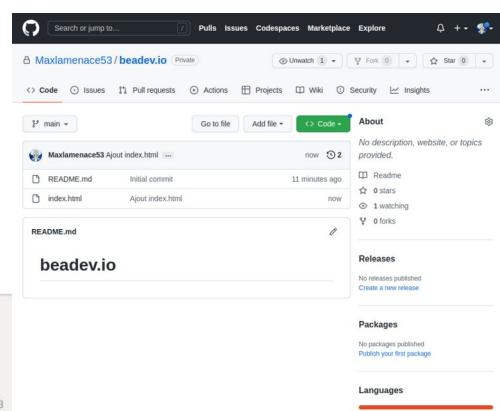


Figure 36: Github.com

Cas d'usage

Durant le cours, nous avons aborder Git en abordant, de prime abord, les nombreuses manipulations possibles d'un projet depuis sa création.

Nous avons créé le projet et initialisé le dossier à partir d'un terminal bash en se situant à la racine du projet et en lançant la commande `git init`. Cette commande a initialisé le dossier en créant des fichiers masqués propres à Git (*Figure 37*).

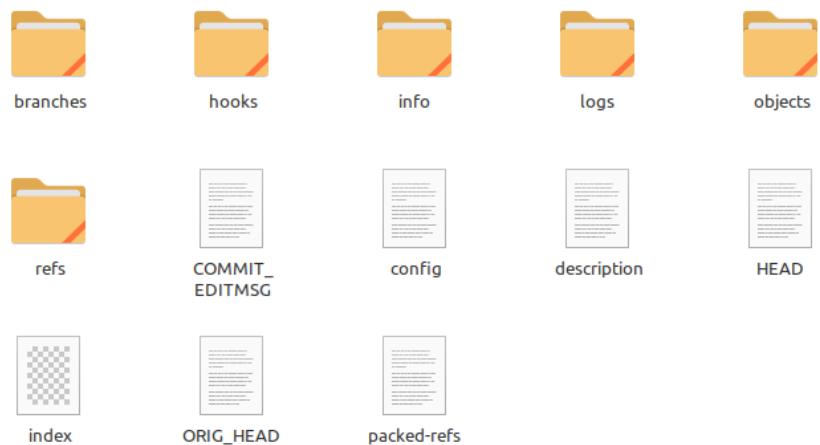


Figure 37: Dossier .Git (le . signifie caché)

Ensuite, nous avons créé une simple page HTML index.html et un about ainsi qu'un dossier image.

Nous avons fait notre premier commit en le préparant avec un git status qui nous montre les dossiers et fichiers non versionnés en rouge, on rajoute ceux voulus avec la commande git add, on refait un git status pour vérifier que les cibles sont bien passées en vert et enfin on peut lancer la commande git commit qui permet de prendre un instantané du projet en sauvegarde pour une restauration éventuelle.

Chaque commit est accompagné d'un message personnalisé, d'un id unique et d'une branche attitrée.

TP

Pour le TP, nous avons récupéré sur la plateforme Github.com un projet 'beadev.io.git' et lançant la commande git clone afin de le récupérer en local sur nos ordinateurs et de travailler dessus(Figure 38).

```
maxime@MaxVictus:~/Documents/DWWM$ git clone https://github.com/Maxlamenace53/beadev.io.git
Clonage dans 'beadev.io'...
Username for 'https://github.com': Maxlamenace53
Password for 'https://Maxlamenace53@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Réception d'objets: 100% (3/3), fait.
```

Figure 38: Git clone

Puis effectuer un git status pour vérifier s'il est à jour (Figure 40), nous y avons rajouté quelques fichiers, nous avons rajouté un tag(Figure 39, les tags sont généralement utilisés pour capturer un point de l'historique) .

On a ensuite vérifié l'historique des log avec un git log qui répertorie tous les commits effectués(Figure 41)

```
maxime@MaxVictus:~/Documents/DWWM/beadev.io$ git tag v0.10
```

Figure 39: git tag

```
maxime@MaxVictus:~/Documents/DWWM/beadev.io$ git status
Sur la branche main
Votre branche est à jour avec 'origin/main'.

rien à valider, la copie de travail est propre
```

Figure 40: git status

```
maxime@MaxVictus:~/Documents/DWWM/beadev.io$ git log
commit 0a2480de3193b0cd175e0037b2a284d2e1527d41 (HEAD -> main, tag: v0.10, origin/main, origin/HEAD)
Author: Maxlamenace53 <70057347+Maxlamenace53@users.noreply.github.com>
Date:   Fri Feb 3 16:10:13 2023 +0100

    Initial commit
```

Figure 41: git log

On peut voir sur la Figure 41 origin/main, cela signifie que l'on se situe sur la branche principal qui se nomme ‘main’.

```
maxime@MaxVictus:~/Documents/DWWM/beadev.io$ git push --set-upstream origin ajoutImageIndex
Enumération des objets: 7, fait.
Décompte des objets: 100% (7/7), fait.
Compression par delta en utilisant jusqu'à 12 fils d'exécution
Compression des objets: 100% (5/5), fait.
Écriture des objets: 100% (5/5), 70.53 Kio | 23.51 Mio/s, fait.
Total 5 (delta 0), réutilisés 0 (delta 0), réutilisés du pack 0
remote:
remote: Create a pull request for 'ajoutImageIndex' on GitHub by visiting:
remote:     https://github.com/Maxlamenace53/beadev.io/pull/new/ajoutImageIndex
remote:
To https://github.com/Maxlamenace53/beadev.io.git
 * [new branch]      ajoutImageIndex -> ajoutImageIndex
La branche 'ajoutImageIndex' est paramétrée pour suivre la branche distante 'ajoutImageIndex' depuis 'origin'.
```

Figure 42: git push

On a ensuite effectué toute une série de manipulation dont git add vu précédemment, git status, git commit et terminé par un git push qui permet d'envoyer le dernier commit sur notre ordinateur (dépôt local) sur notre compte github (dépôt distant) pour le sauvegarder en ligne, le consulter et surtout le mettre à disposition d'autre personne dans le cas de projet en équipe. Avec la commande git checkout ajoutImageIndex, nous avons créé une branche parallèle à la branche main et nous y avons déplacé le référentiel ‘origine’.

A la fin du TP, nous avions un projet avec 4 branches (Figure 43)et une multitude de commit avec différents tag (Figure 44)

```
maxime@MaxVictus:~/Documents/DWWM/beadev.io$ git branch --list
* ajoutImageIndex
  develop
  main
  nouvellePage
```

Figure 43: git branch --list

```
maxime@MaxVictus:~/Documents/DWWM/beadev.io$ git log
commit 9cdeca845a0e721a43eaa29fc8ecfef3a88d09ca (HEAD -> main, origin/main, origin/HEAD)
Merge: 6000df7 b6de9e7
Author: Maxime Lemée <maxime.lemee1@gmail.com>
Date:   Fri Feb 3 16:56:02 2023 +0100

    Merge branch 'nouvellePage'

commit b6d9e7055492aa0f4ad2a8e5b1d3d0113c9d69d (nouvellePage)
Author: Maxime Lemée <maxime.lemee1@gmail.com>
Date:   Fri Feb 3 16:54:36 2023 +0100

    nouvelle Page about

commit 6000df7bfe2bf30c45fdb262c8cbcce02726b30 (tag: v0.2.0, origin/ajoutImageIndex, ajoutImageIndex)
Author: Maxime Lemée <maxime.lemee1@gmail.com>
Date:   Fri Feb 3 16:37:58 2023 +0100

    Ajout image sur page index.html

commit a569c9bd31544980ce068a47e500377898635ca2f (tag: v0.10, develop)
Author: Maxime Lemée <maxime.lemee1@gmail.com>
Date:   Fri Feb 3 16:21:01 2023 +0100

    Ajout index.html

commit 0a2480de3193b0cd175e0037b2a284d2e1527d41
Author: Maxlamenace53 <70057347+Maxlamenace53@users.noreply.github.com>
Date:   Fri Feb 3 16:10:13 2023 +0100

    Initial commit
```

Figure 44: git log TP

La liste des commandes est encore très longue et non exhaustive, je ne peux tous les citer ici mais nous avons utiliser également :

- git show « numéro de commits » qui permet de voir le détail du commit voulue.
- git diff qui permet d'afficher les différences entre commit ou entre branches.
- git pull qui permet de mettre à jour son dépôt local par rapport au dépôt distant (github)
- git merge qui permet de fusionner des branches entre elles.

Même si l'usage du terminal est vivement conseillé, les IDE facilitent grandement le versionning de nos projets avec des outils visueles et rapides.

1.3.4 Référencement

Définitions SEO SEA

Le Référencement ou SEO (Search Engine Optimisation) est l'ensemble des pratiques mises en œuvre pour améliorer la position d'un site web sur la SERP (Search Engine Result Page) d'un navigateur de recherche comme google, bing..(Figure 46). Le SEO est primordial pour la bonne marche des sites car il améliore grandement leur visibilité lors d'une recherche d'un internaute.

Chaque moteur de recherche à ses propres règles et ses propres algorithmes qui demeurent secrets mais chacun à ses outils pour aider le développeur à améliorer les résultats de son site lors d'une recherche et ainsi avoir un impact sur le long terme et un suivi très précis de la fréquentation et ainsi connaître son ROI (Return On Investment).

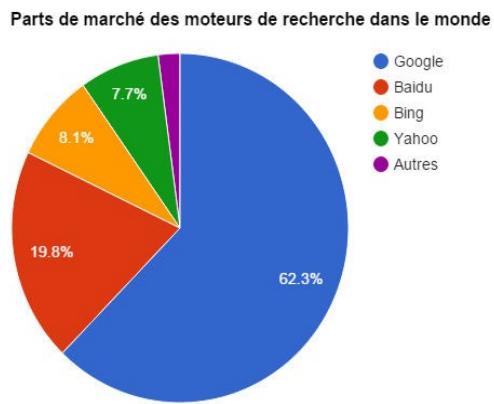


Figure 45: Parts des marché moteur de recherche



Figure 46: Principaux moteur de recherche

En parallèle du SEO (référencement naturel), il existe le SEA.

Le SEA (Search Engine Advertising) correspond à l'ensemble des actions qui relèvent de la création et de l'optimisation de publicités

LE SEA est payant, contrairement au SEO, mais l'impact est plus rapide, avec un contrôle sur les publicités (le contenu de notre site) affichées pour le moteur de recherche. Le modèle économique varie selon le moteur de recherche, par exemple chez google, c'est géré par le service google ads, qui facture au clic sur la publicité, idem pour le service de Microsoft bing ads.

Le référencement SEO de notre site par un moteur de recherche comme google s'effectue en grandes étapes :

1. **Crawl** : Examen du site web par un robot, il stocke les données visibles et il visite les liens du site .
2. **Indexation** : Étude des pages crawlées (HTML, Mot-clés, images...)
3. **Ranking** : Triage des données par rapport à l'algorithme du moteur de recherche

4. Affichage dans la SERP : Classement du site par rapport à la pertinence des étapes précédentes.

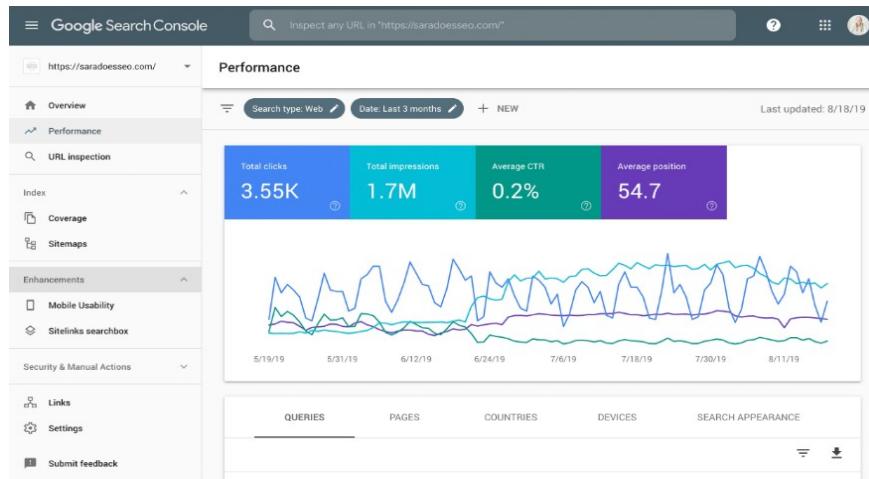


Figure 47: Search console Google

On a vu par la suite la SearchConsole(Figure 47) de google qui permet de voir toutes les informations concernant notre site en ligne.

Cas d'usage

On a réalisé pour notre projet personnel un fichier sitemap.xml(Figure 48)pour lister les pages importantes afin de les indiquer à google lors du crawl et de son indexation.

On a ensuite vu les optimisations SEO possibles avec la gestion du Titre de la page dans la balise head de l'HTML. Il est conseillé de limiter sa longueur si possible afin qu'il s'affiche correctement sans coupure lors d'une recherche, de même pour la méta description de notre site .

On a dans la continuité du sitemap, créé le fichier robots.txt qui a comme rôle de donner des instructions au moteur de recherche comme le fait de ne pas indexer telle page, d'interdire le crawl également.

```
This XML file does not appear to have any style information associated with it. The document tree is shown below.

<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9" xmlns:xhtml="http://www.w3.org/1999/xhtml"><![CDATA[&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/cayman-iles-basane&lt;/loc&gt;
    &lt;lastmod&gt;1999-06-09&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/vatican-etat-du-ecru&lt;/loc&gt;
    &lt;lastmod&gt;1997-07-07&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/beninbleu-outremer&lt;/loc&gt;
    &lt;lastmod&gt;1986-10-24&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/svalbard-et-jan-mayen-iles-violet&lt;/loc&gt;
    &lt;lastmod&gt;1997-09-11&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/liechtensteingris-perle&lt;/loc&gt;
    &lt;lastmod&gt;1972-04-24&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/burundibisque&lt;/loc&gt;
    &lt;lastmod&gt;1983-06-17&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/tongajaune-canari&lt;/loc&gt;
    &lt;lastmod&gt;1993-07-23&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/chine-rep-pop-cramoisi&lt;/loc&gt;
    &lt;lastmod&gt;2015-04-24&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/norfolk-iles-bleu-marine&lt;/loc&gt;
    &lt;lastmod&gt;1970-11-14&lt;/lastmod&gt;
&lt;/url&gt;
&lt;url&gt;
    &lt;loc&gt;http://127.0.0.1:33351/trick/marocbleu-canard&lt;/loc&gt;
    &lt;lastmod&gt;2007-10-23&lt;/lastmod&gt;
&lt;/url&gt;
&lt;/urlset&gt;</pre>

```

Figure 48: sitemap.xml

Nous avons ensuite réalisé un test de notre site avec l'aide de l'outil de google light house situé dans la console de son navigateur(Figure 49), celui-ci note le site sur une multitude de points et donne des solutions ou des voies vers lesquelles aller.

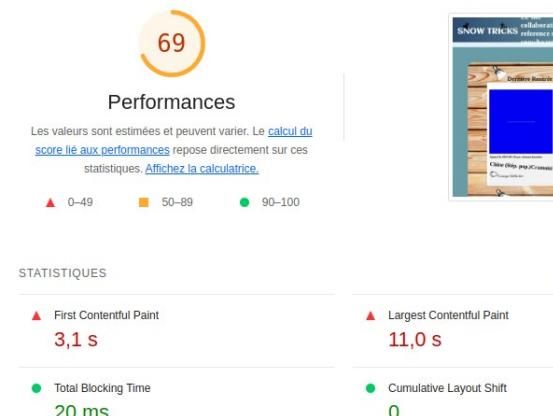


Figure 49: Test SEO avec l'outil google

On a vu également qu'il était conseillé d'avoir des urls explicites et compréhensibles par tous grâce à la mise en place de slugs.

Par la suite, nous avons vu certaines pratiques et certains outils quasiment indispensables pour l'obtention d'un bon référencement.

Black hat

Il existe des pratiques interdites pour booster le référencement de son site comme par exemple utiliser ou dupliquer le contenu comme les balises et les métadescription présentes sur une autre page pour l'utiliser sur notre site. On peut aussi être poussé à tromper le user-agent du navigateur. On peut être tenté de présenter un contenu différent de celui des robots en les cachant ou d'insérer un très grand nombre de mots clés.

L'Open Graph

C'est du contenu de notre application (image/text,,etc) que l'on veut partager via la balise <og>. C'est une norme mise en place par Facebook en 2010 et qui permet le partage de contenu sur les réseaux sociaux.

Par exemple (Figure 50), il me précise que mes balises img ne possèdent ni width ni height explicites.

On a vu également l'importance des balises H1 à H6 et des mots clés avec ici par exemple la présence d'un seul H1 pour chaque page

Il faut également bien alimenter le ALT des balises image afin de donner la description des photos au robot des navigateurs et pouvoir les afficher par la suite dans leur galerie d'images et également pour aider les malvoyants dans leur navigation.

Le contenu du site doit avoir au minimum 200 mots par page pour avoir un impact en incluant des mots-clés et leurs variantes avec du contenu mis à jour régulièrement .



Figure 50: Diagnostic test SEO

Les données structurées

Cela permet d'ajouter des données directement dans la SERP (comme les détails sur une recette de cuisine, ou les détails d'un article, son prix, en stock..) , On le fournit au navigateur grâce à un fichier JSON-LD(Figure 51) soit dans une balise <script type> au sein de notre code HTML, soit directement au sein des balises HTML avec la propriété « itemprop »

```
<script type="application/ld+json">
  {"@context": "http://schema.org", "@type": "Recipe", "name": "La meilleure recette de pâte à crêpes", "recipeCategory": "pâte à crêpes", "image": ["https://assets.afcdn.com/recipe/20211122/124598_w1024h576c1cx3176cy2107ctx1161cyt477cxb5347cyb3565.webp", "https://assets.afcdn.com/recipe/20211122/124598_w1024h768clc de farine", "3 c.à.s de sucre", "2 c.à.s d'huile", "50 g de beurre fondu", "5 cl de rhum", "3 oeufs entiers", "60 cl de lait"], "recipeInstructions": [{"@type": "HowToStep", "text": "de la retourner. Cuire ainsi toutes les crêpes à feu doux."}], "author": "isa", "description": "farine, sucre, huile, beurre, rhum, oeuf, lait", "keywords": "La meilleure recette de pâte à crêpes", "url": ["https://assets.afcdn.com/recipe/20211122/124598_w1024h576c1cx3176cy2107ctx1161cyt477cxb5347cyb3565.webp", "https://assets.afcdn.com/recipe/20211122/124598_w1024h768clc 09-26 13:45:25.000000"]} } == $0
</script>
```

Figure 51: JSON-LD données structurées

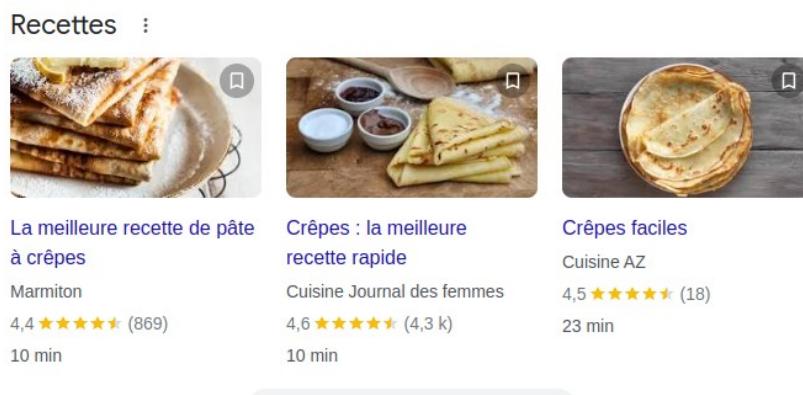


Figure 52: Exemple affichage grâce aux données structurées

Grâce au nommage normé du fichier JSON en rapport avec le site schema.org qui répertorie tous les termes à utiliser, google va connaître le sujet du script et les afficher en conséquence.

Les Backlinks

Penser à s'ajouter dans les annuaires les plus consultés comme ceux de google et trouver des partenariats avec d'autre sites afin d'être visible et de faire comprendre à google que notre site est intéressant car il est partagé.

Le maillage interne

Google va aussi vérifier le fonctionnement des liens internes afin de comprendre la structure de celui-ci et aider les internautes à naviguer.

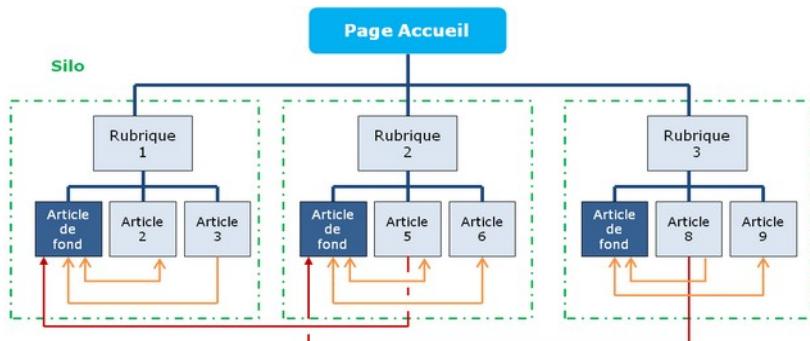


Figure 53: Maillage interne

Les Webs Analytics

Afin de connaître la fréquentation de telle ou telle page, de savoir par quels biais les internautes sont arrivés sur notre site, quels sont les mots clés utilisés, dans quels pays, on est le plus recherché, on utilise un outil qui permet d'analyser notre site afin de nous fournir un ensemble de renseignements .

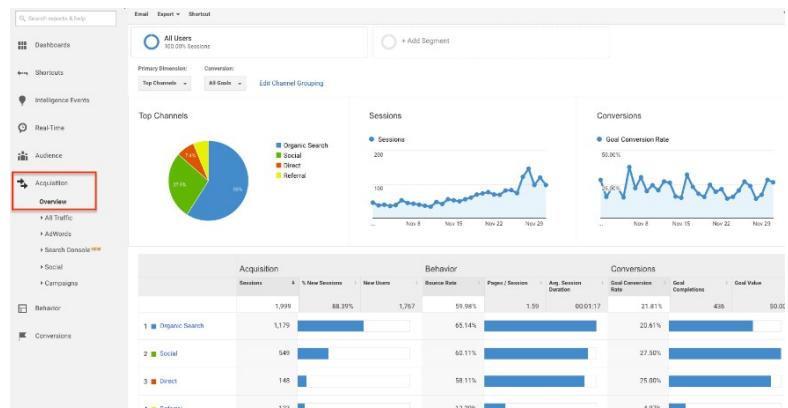


Figure 54: Google Analytics

Il existe une multitude d'outils pour cela, mais le plus connu et gratuit est l'outil google Analytics, qui est très bien fourni et qui possède l'avantage de répertorier tout l'écosystème google (Figure 54).

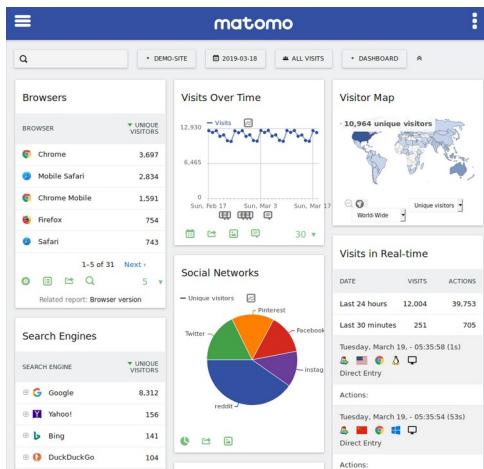


Figure 55: Matomo

Il existe également Matomo, qui est une solution sensée être plus respectueuse de notre vie privée (Figure 55).

Pour utiliser ces outils, il suffit d'insérer un bout de code fourni dans la balise HEAD de notre code HTML, c'est très simple à mettre en place.

1.4. Développer une interface utilisateur web dynamique

1.4.1 Algorithmique appliquée

Définition

L'algorithmique est l'étude et la production de règles et techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systématiques de résolution d'un problème permettant de décrire précisément des étapes pour résoudre un problème algorithmique. (source wikipedia)

Cas d'usage

Nous avons appris que les règles algorithmiques s'appliquent à tout domaine informatique et surtout à tout langage informatique. La façon de les écrire et d'y avoir accès varie mais la logique reste totalement identique. C'est un domaine très riche et varié.

Nous nous sommes exercés via le langage JavaScript avec le traditionnel `console.log("Hello World!");` qui permet d'afficher Hello World dans la console du navigateur.



```
JS 01_helloworld.js × ...
JS 01_helloworld.js
1 console.log("Hello World!");

DEBUG CONSOLE ... Filter (e.g. text, | exclude) Lancer le programme: i ↴
/home/maxime/.nvm/versions/node/v19.6.0/bin/node ./01_helloworld.js
Hello World!
helloworld.js:1 01
```

Figure 56: Premier `console.log`

Pour exécuter le code, nous utilisons le terminal dans l'IDE (Figure 56) Visual Studio Code ainsi que le debug fourni avec. La fonction `Console.log()` permet d'afficher ce que l'on souhaite dans la console, c'est un moyen très utile pour vérifier l'état d'une variable par exemple à un point précis dans le déroulement du programme.

Tout ce qu'on écrit dans un script , c'est à dire des lignes de codes, un morceau de programme ayant une fonction à exécuter, est lu par le navigateur web ou le serveur qui le lira naturellement du haut vers le bas et le relira à chaque recharge de page. Seuls les commentaires - que l'on écrit en JS(JavaScript) `// commentaire` pour une ligne et `/* commentaire */` quand le commentaire est sur plusieurs lignes - échappent à l'exécution.

Chaîne de caractère

(Figure 57) On a appris à afficher une chaîne de caractère en les entourant de simple ou de double quote, à les concaténer entre elles grâce au caractère `+` et à les afficher sur plusieurs lignes avec le caractère `\n`, on peut également échapper des caractères avec `\` devant la lettre voulue.

```
// Affichez la phrase: Afficher un \ c'est difficile
console.log("Afficher un \\ c'est difficile");

console.log("\nExercice 4");
//Affichez les phrases suivantes en utilisant une seule instruction console.log():
// Je suis la première ligne
// Je suis la deuxième ligne
console.log("Je suis la première ligne \nJe suis la deuxième ligne");
console.log("\nExercice 5");
// Le code suivant sépare intentionnellement le prénom de l'utilisateur des autres chaines pour
// Néanmoins le précédent développeur a fait des erreurs que vous devez corriger.
// Retirez les commentaires devant le code pour commencer
console.log("Bonjour " + 'Bob' + ", aujourd'hui nous faisons une promo 'la fidélité paye' !");
```

Figure 57: Chaîne de caractère

Opérations mathématiques

Les opérations mathématiques respectent les règles classiques de maths comme l'ordre des opérations, additions, divisions, multiplications et modulo qui nous donnent le reste d'une division.

Variables

(Figure 58)

On a appris ce qu'était une variable et comment les utiliser via des exercices de concaténation et de calcul.

On a vu les différents types, string, integer.

```
// On souhaite coder une horloge
// Veuillez corriger le code
let h = 13;
let m = 37;
let séparateur = "h";
let horaire = h + séparateur + m;
console.log("Bienvenue, il est " + horaire);
```

```
console.log("\nExercice 3");
// On souhaite afficher la distance pour arriver à une ville
// La distance pour arriver à Arles est de 187km
// Veuillez compléter le code
let ville = "Arles";
let distance = 187;
console.log("La distance pour " + ville + " est de " + distance + " km");
```

Figure 58: Variable

Booléens 0 / 1

(Figure 59)

Très utilisé car très économique en mémoire, il ne peut prendre que 2 états , vrai ou faux, true ou false, 1 ou 0

```
2  console.log("\nExercice 0")
3 // On veut créer test qui vérifie si une personne à droit d'accéder à notre application.
4 // Elle doit avoir moins de 65 ans
5 // Nous voulons faire le test pour une personne de 40 ans
6 // Pour faire cet exercice :
7 // 1. Retirez les commentaires devant le code
8 // 2. Remplacez uniquement les points d'interrogations "???" par des opérateurs
9 // de comparaison et/ou des opérateurs logiques
10 console.log(65 > 40);
11
12 console.log("\nExercice 1");
13 // Maintenant nous voulons vérifier aussi qu'elle a plus de 16 ans.
14 // Pour faire cet exercice :
15 // 1. Retirez les commentaires devant le code
16 // 2. Remplacez uniquement les points d'interrogations "???" par des opérateurs
17 // de comparaison et/ou des opérateurs logiques
18 console.log(65 > 40 && 16 < 40);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
/home/maxime/.nvn/versions/node/v19.6.0/bin/node ./03_nombres_exercices.js

Exercice 0
34

Exercice 1
13

Figure 59: Booléens

Les fonctions : function(){ }

(Figure 60)

Les fonctions permettent d'effectuer une ou des opérations précises sur un ou des éléments ciblés, c'est un des principes de base de la programmation.

On nomme la fonction, on lui rentre un ou des paramètres entre () ensuite entre {} on effectue des actions dessus et éventuellement avec un **return**, on fait ressortir un résultat mais pas obligatoirement.

```
40  console.log("\nExercice 2");
41 // Le stagiaire a créé une fonction qui retourne un message d'accueil
42 // Là encore, il est surpris que rien ne se passe, aidez le à corriger
43 // NE PAS MODIFIER
44
45 // }
46 // MODIFIEZ APRES
47
48
49 function ditBonjour(nom) {
50   return "Bonjour " + nom + " !";
51 }
52
53 console.log(ditBonjour("moi"));
54
55 console.log("\nExercice 3");
56 // Cette fois vous devez créer une fonction de zéro
57 // Créez une fonction qui RETOURNE le résultat de l'addition de deux chiffres
58 // Appelez cette fonction et affichez le résultat
59
60
61
62
63 function addition(a, b) {
64   return a + b;
65 }
66
67 console.log(addition(8,2));
68
```

PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL

Exercice 2
Bonjour moi !

Exercice 3
10

Figure 60: Fonction

Les Tableaux[]

(Figure 61)

Les tableaux sont des variables contenant une série de données trouvables grâce à un index (ou key), ces données peuvent être de n'importe quel type comme dans l'exemple de la figure 127 mais mélanger les types de variable au sein d'un même tableau est une mauvaise pratique.

```
// Déclarer un tableau
let monTableau = ["uneChaine", true, 34, "Ert", "bonjour", 0, false, true];
```

Figure 61: Tableau

Les conditions if() {}

(Figure 62)

Les conditions permettent de vérifier des états entre variables afin d'effectuer des actions ou pas avec. On utilise une condition If(condition vrai ou faux)

si vrai alors on exécute ce qu'il y a à l'intérieur des {} ;

Symbol	Description
==	Est égale à
>	Est supérieur à
>=	Est supérieur ou égale à
<	Est inférieur à
<=	Est inférieur ou égale à
!=	Est différent de

Figure 63: Symbole de comparaison

On utilise des symboles de comparaison(Figure 63)

```
console.log("\nExercice 2");
// // // Ecrivez sous la forme d'une fonction
// // // Si la personne a plus de 13 ans et fait plus de 150 cm
// // // Si elle a 13 ans ou moins mais fait plus de 180 cm
// // // Sinon afficher: "Vous ne pouvez pas rentrer"
function testAgeEtTaille(age, taille) {
    if (age > 13 && taille > 150) {
        console.log("Vous pouvez entrer");
    } else if (age < 13 && taille > 180) {
        console.log("Tu es grand pour ton age");
    } else {
        console.log("Vous ne pouvez pas rentrer");
    }
}
```

testAgeEtTaille(14, 140);

Figure 62: Conditions

Si la condition est fausse, alors le programme ignore le if et continue après celui-ci.

Les boucles for(){} /while(){}

Les boucles vérifient la condition entre () et tant que celle ci est bonne alors il exécute ce qu'il y a entre {} , il existe la boucle if (tant que vrai alors){On exécute ce code jusqu'à ce que la condition soit fausse}et la boucle while très ressemblante. Il faut toujours penser à rendre la condition fausse à un moment donné sinon on se retrouve dans une boucle infinie.

```
let compteur = 0;
while (compteur < 3) {
    console.log("Je suis le passage : " + compteur);
    compteur += 1;
}
```

Figure 64: Boucle while

```
for (let compteur = 0; compteur < 3; compteur = compteur + 1) {
    console.log("Instruction de la boucle for");
    console.log("compteur : " + compteur);
}
```

Figure 65: Boucle for

La programmation orientée objet

La POO ou programmation orientée objet est un paradigme de programmation, c'est-à-dire une façon de concevoir le code et de penser la programmation. La POO s'oriente autour de classe (un schéma ou un plan) que l'on va instancier ou appeler pour créer un objet ou une instance (à partir de ces plans de conceptions ou de son schéma de conception). On peut créer autant d'objet que l'on veut à partir d'une même classe, tous les objets créés seront indépendants les un des autres .

Durant les cours, nous avions à créer une classe véhicule et instancier des nouveaux véhicules en créant des objets avec différents paramètres, nous avions différents véhicules fabriqués à partir de la même classe. On a géré également l'héritage de classe avec extends et des méthodes que l'on peut appeler sur l'objet créé à partir de cette classe.

```
class Vehicule {
    nbreRoue;
    qtEssence;
    modele;

    constructor(nombreDeRoue, quantitéEssence, modeleDeVoiture) {
        this.nbreRoue = nombreDeRoue;
        this.qtEssence = quantitéEssence;
        this.modele = modeleDeVoiture;
    }

    demarrer() {
        return "VROUM," + this.modele + " démarre";
    }

    avance() {
        this.qtEssence = this.qtEssence - 2;
        return this.modele + " avance, il reste " + this.qtEssence + " litres.";
    }

    klaxonne() {
        return "TUTUT";
    }
}

class Voiture extends Vehicule {
    nbreRoue = 4;

    klaxonne() {
        return "POUIN";
    }

    roueArriere() {
        return this.modele + " ne peut pas faire de roue arrière !";
    }
}

class moto extends Vehicule {
    nbreRoue = 2;

    roueArriere() {
        return this.modele + " fais une roue arrière";
    }
}

let newVehicule = new Vehicule(8, 50, "Vévé");
let newMoto = new moto(85, 60, "Yamaha");
let newVoiture = new Voiture(9, 80, "peugeot");

console.log("-----Véhicule-----");
console.log(newVehicule);
console.log(newVehicule.demarrer());
console.log(newVehicule.avance());
console.log(newVehicule.avance());
console.log(newVehicule.klaxonne());
```

Figure 66: Classe et objet

Pour instancier un objet, on peut faire appel au constructor de la classe en indiquant tous les paramètres attendus par celui-ci

1.4.2 JavaScript

Définition

JavaScript est un langage de script exécuté par le navigateur principalement employé pour créer des pages web interactives, il est le plus souvent utilisé de concert avec le HTML et le CSS. Il peut également être utilisé comme langage back-end avec l'utilisation de Node.js coté serveur,



Figure 67: Logo JavaScript

Il peut être directement inclus dans le code HTML via des balises <script> ou dans un fichier '.js' à part et inclus en parallèle dans le fichier index.html dans la balise HEAD.

Cas d'usage

Durant les cours, nous avons réalisé des exercices divers pour nous apprendre la façon d'interagir avec les éléments HTML. Dans un premier temps, nous devions mettre dans une variable l'élément ou les éléments ciblés pour ensuite pouvoir effectuer des actions avec. Pour les sélectionner, il faut utiliser la fonction querySelector qui permet de sélectionner une classe, un attribut HTML ou un ID.

Exemple : `const formGame = document.querySelector("#form-game")`.

On sélectionne l'élément portant l'id #form-game dans le code HTML, et on l'utilise au sein de la fonction comme un addEventListener qui va écouter quand un formulaire est envoyé via un submit.

```
1 if (document.location.pathname.split("/").pop() === "justeprix.html") {
2     // Sélection des éléments de la page
3     const message = document.querySelector("#message");
4     const formGame = document.querySelector("#form-game");
5     const btnReplay = document.querySelector("#btn-replay");
6     const formInput = formGame.querySelector("input#user-input");
7     let counter = 0;
8
9     // Générer un nombre aléatoire
10    let max = 100;
11    let random = Math.round(Math.random() * max);
12    console.log(random);
13
14    // Écouter l'évènement de submit du formulaire
15    formGame.addEventListener("submit", function (event) {
16        event.preventDefault();
17        counter++;
18        console.log(counter);
19
20        // Récupérer la saisie de l'utilisateur
21        let userInput = formInput.value;
22        userInput = parseInt(userInput);
23
24        // Vérifier si la saisie est bonne
25        if (userInput === random) {
26            message.innerText = 'Vous avez gagné en ' + counter + ' tentatives';
27
28            counter.innerText = count;
29            // Afficher le bouton rejouer
30            btnReplay.classList.remove("d-none");
31        } else if (userInput > random) {
32            message.innerText = "C'est moins !";
33        } else {
34            message.innerText = "C'est plus !";
35        }
36    });
37
38    // Gestion du click sur bouton rejouer
39    btnReplay.addEventListener("click", function () {
40        btnReplay.classList.add("d-none");
41        message.innerText = "Veuillez saisir une proposition";
42        formGame.reset();
43        random = Math.round(Math.random() * max);
44        formInput.focus();
45        counter = 0;
46    });
47 }
```

On va pouvoir ensuite effectuer toute une série d'actions voulues avec des boucles et des conditions (voir chapitre Les conditions if() { }). Pour le cours, nous avons réalisé plusieurs exercices dont un qui consistait à créer un jeu pour deviner un chiffre entre 2 intervalles, un jeu où il fallait cliquer sur un bouton qui bougeait aléatoirement et un script de chiffrement. Nous changeons, par exemple, le texte dans des cas spécifiques, nous désactivons et réactivons des boutons ou des parties du code HTML. Nous modifions également dans le script le CSS des éléments avec la fonction `.style` ou `.classList`.

Tout cela nous construisait des pages dynamiques.

Nous avons également commencé à travailler et à construire nos premières fonctions avec leur paramètres, leurs conditions, les actions à effectuer si les conditions sont bonnes et éventuellement une valeur qu'elle retourne.

Assez vite, nous avons appris à se servir de la console de débogage du navigateur ou de l'IDE utilisée qui s'avère très utile grâce à ces points d'arrêt et la lecture ligne par ligne du code.

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. A file named 'script.js' is open, showing a snippet of JavaScript code. Several breakpoints are set in the code, indicated by small red dots. The right side of the screen displays the call stack, which is currently empty (labeled 'Couverte : N/A'). The status bar at the bottom indicates 'Ligne 12, colonne 1'.

```

let loupe = document.querySelector("#loader");
const searchForm = document.querySelector("#search-form");
const indexMusic = [];
const requestedLyric = document.querySelector("#requested-lyrics");
const btnSearch = document.querySelector("#btn-search");
loupe.style.display = "none";
loupe.innerHTML = "Chargement...";

searchForm?.addEventListener("submit", function (event) {
  event.preventDefault();
  const artist = this.querySelector("#artist")
    .value.toLowerCase()
    .replace(/\s/g, "-");
  const title = this.querySelector("#title")
    .value.toLowerCase()
    .replace(/\s/g, "-");
  console.log(artist);
  console.log(title);
  Search();
});

function Search() {
  if (!artist || !title || title === "") {
    alert("Veuillez renseigner un artiste ainsi qu'un titre");
  } else {
    loupe.style.display = "block";
    console.log(`Recherche pour ${artist} ${title}`);
  }
}

Search();

```

Figure 68: Console d'inspection du navigateur chrome

Travaux pratique

The screenshot shows the browser developer tools with the 'EXPLORATEUR' (File Explorer) and 'script.js' tabs selected. The 'script.js' tab contains the following code:

```

if (request.readyState === 4) {
  const lyrics = JSON.parse(response);
  lyrics.innerHTML = lyrics.lyrics;
  requestedLyric.innerHTML = "";
  lyrics.style.display = "block";
  requestedLyric.style.display = "none";
  btnSearch.disabled = true;
  fetch(`https://lyricsapi.pierre-jehan.com/music?title=${title}&artist=${artist}`);
}

```

Figure 69: Scripts pour la récupération de parole

En travaux pratique, nous avions un coder, un script de récupération de données sur une API via la méthode FETCH.

L'utilisateur rentre dans les champs la chanson dont il désire les paroles et avec le bouton envoyer, il soumet la demande, si la demande est possible alors on affiche les paroles. Durant cette exercice, nous avons appris à effectuer une requête avec un FETCH via la méthode GET qui permet de récupérer via une API des informations, au contraire de la méthode POST qui permet d'envoyer des données. Nous devions gérer les erreurs comme le fait de ne pas trouver les paroles en base de

données et formater le texte rentré par l'utilisateur dans les champs pour les rendre compatible avec l'API et les afficher en cas de requête réussie.

Nous devions dans ce cas, faire disparaître le texte '*search for a song to display the lyrics*' ainsi que la loupe et réinitialiser le formulaire pour qu'il soit prêt pour une prochaine requête.



Figure 71: Écran accueil

Artist Title
Search for a song to display the lyrics...

Oo-oo-oo, oo-oo-oo, oo-oo-oo. Oo-oo-oo, oo-oo-oo, oo-oo-oo. When :
oo-oo-oo, oo-oo-oo. Oo-oo-oo, oo-oo-oo, oo-oo-oo. When she was jus
her eyes. In the night, the stormy night, Away she flied. And dream of
la-la-la La-la-la-la. So lying underneath the stormy skies. She said oh-
para-para-paradise, Whoa-oh-oh oh-oooh oh-oh-oh. (Oh, oh. Oo-oo-oo
oo-oo-ooo...)

Figure 72: Requête réussi

1.4.3 VUE

Définition

VUE est un framework Front-End JavaScript flexible, rapide et censé être simple d'approche. Il permet de travailler sur la partie Front-End d'un projet en utilisant un DOM virtuel qui permet des changements d'états très réactifs.



Vue fonctionne en mode Single page Application, c'est à dire que le point d'entrée du navigateur pour la lecture de l'application web ne se fera qu'à partir d'un seul fichier « index.html ».

Vue fonctionne sur un principe de composant comprenant du JavaScript, du HTML et du CSS dans le même fichier que l'on implante à l'endroit que l'on veut et autant de fois que l'on veut. Il possède son propre routeur pour gérer les différentes pages et un store (pinia) qui permet de stocker des variables et fonctions. Vue possède également ses propres directives comme v-model qui lie les données du modèle à la vue et inversement, v-for qui est l'équivalent de la boucle for, v-if(v-else-if) pour la boucle if, v-on ou (@) qui permet d'associer un élément à un événement par exemple.

Cas d'usage

Dans le cadre de nos cours, nous avons réalisé divers exercices dont un todos qui allait chercher une liste déjà établie de tâches sur une API gratuite avec laquelle nous avons interagi.

```
Terminal
Fichier Édition Affichage Rechercher Terminal Aide
maxime@MaxVictus:~$ npm create vue
Vue.js - The Progressive JavaScript Framework

✓ Project name: todo
✓ Add TypeScript? ... No / Yes
✓ Add JSX Support? ... No / Yes
✓ Add Vue Router for Single Page Application development? ... No / Yes
✓ Add Pinia for state management? ... No / Yes
✓ Add Vitest for Unit Testing? ... No / Yes
✓ Add an End-to-End Testing Solution? ... No
✓ Add ESLint for code quality? ... No / Yes

Scaffolding project in /home/maxime/todo...

Done. Now run:

  cd todo
  npm install
  npm run dev

maxime@MaxVictus:~$
```

Figure 75: Crédit à la création du projet via le terminal bash son nom, l'utilisation ou non de Typescript,

Si l'installation du routeur nous intéresse, idem pour le store(pinia) .

Il nous propose également l'installation de VITE, qui est un compilateur et un serveur de test dans le même style que webpack mais réputé plus rapide (donc son nom).

Une fois toutes les commandes demandées et effectuées, on se retrouve avec une structure (Figure 74: hiérarchie Vue) de dossier déjà faite et organisée d'où l'intérêt de l'utilisation d'un framework.

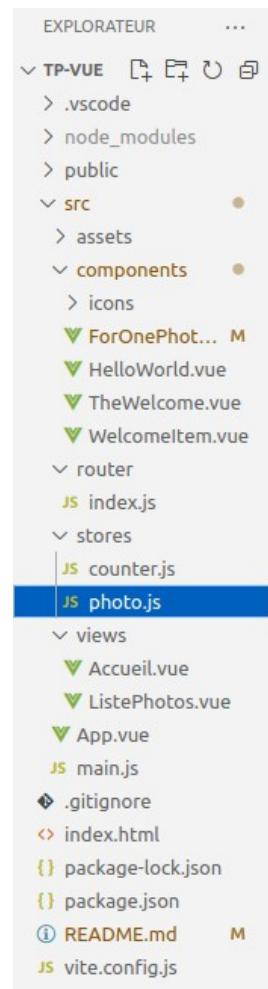


Figure 74: hiérarchie Vue

- Projet
 - Dossier src(sources)
 - assets (Images et autres éléments externe obligatoire)
 - composents (Instance utilisable en HTML)
 - router (Gestion des pages de l'application)
 - store (Bibliothèque de gestion d'état et de méthodes)
 - views (Pages de l'application)
 - Apps.vue (points entrée de l'application)
 - mains.js



Figure 76: Page accueil todos

Une fois tout cela créé et la page accueil (Figure 76: Page accueil todos) avec une simple balise `<h1>`, nous avons appris le principe de relation parents/Enfants avec les props qui permettent d'envoyer des données du parent à l'enfant et les emits qui permettent l'inverse.

Nous avons créé une page Todos en y incluant un composant (Figure 79: Composant Todo) et en bouclant dessus grâce à la directive v-for (Figure 78: Boucle v-for) et un fetch (Figure 77: Fetch)

```
async function load() {
  try {
    const resp = await fetch('https://jsonplaceholder.typicode.com/todos');
    const result = await resp.json();

    todos.value = result;
  }
  catch (err) {
    console.error(err);
  }
}
```

Figure 77: Fetch

```

1 <template>
2   <div class="todo" @click="toggleCompleted" :class="{ 'not-completed': !todo.completed }">
3     <span>{{ todo.title }}</span>
4   </div>
5 </template>
6
7 <script setup>
8 const props = defineProps({
9   todo: Object
0 });
1
2 function toggleCompleted() {
3   props.todo.completed = !props.todo.completed;
4 }
5
6 </script>
7
8 <style scoped>
9   .todo {
10     padding: 10px;
11     border: solid #333 1px;
12     border-radius: 2px;
13     margin-bottom: 2px;
14     transition: all 0.25s ease-in-out;
15   }
16
17   .not-completed {
18     background: #crimson;
19     color: white;
20   }
21 </style>

```

Figure 79: Composant Todo commerce

```

<template>
  <h1>Liste des todos</h1>

  <todo v-for="todo of todos"
        :key="todo.id"
        :todo="todo" />
</template>

```

Figure 78: Boucle v-for

1.5. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Liste des todos	
Accueil	
Todos	
delectus aut autem	quis ut nam facilis et officia qui
fugiat veniam minus	et porro tempora
laboriosam mollitia et enim quasi adipisci quia provident illum	qui ullam ratione quibusdam voluptatem quia omnis
illo expedita consequatur quia in	quo adipisci enim quam ut ab
molestiae perspiciatis ipsa	illo est ratione doloremque quia maiores aut
vero rerum temporibus dolor	ipsa repellendus fugit nisi
et doloremque nulla	repellendus sunt dolores architecto voluptatum
ab voluptatum amet voluptas	accusamus eos facilis sint et aut voluptatem
quo laboriosam deleniti aut qui	dolorum est consequatur ea mollitia in culpa

Figure 80: Affichage des Todos

1.5.1 Crée un site avec Wordpress

Définitions

Wordpress est un CMS (Content Management System) conçu en premier lieu, pour concevoir des blogs et des sites vitrines simples sans taper la moindre ligne de code ou le moins possible avec une gestion du contenu, un système d'administration, des médias, des droits d'accès et la possibilité d'installer des *plugins* (fonctionnalités) afin de concevoir l'application comme un site de *Wordpress commerce* avec le plugin Woocommerce, du multilingue, un système de newsletters .



Il existe une multitude de CMS comme Drupal, Joomla, Shopify mais Wordpress est le CMS le plus utilisé mondialement (Figure 82)

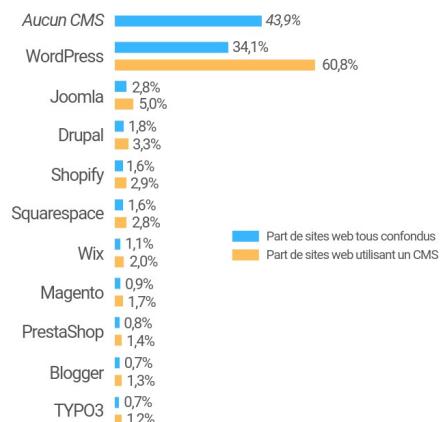


Figure 82: Source W3Tech juillet 2019

Cas d'usage

En premier lieu, nous avons téléchargé Wordpress et placé le fichier dans le dossier www afin de pouvoir l'ouvrir dans notre localhost.

En rentrant localhost/ ‘nomdudossierwordpress’ dans un navigateur, après une page de configuration à la base de donnée, on arrive sur la page d'administration principal (Figure 46)

C'est à partir de là que l'on administre tout notre site, Wordpress supporte également le multisite.

Nous avons installé un thème visuel twentytwenty choisi parmi un large panel de possibilités (Figure 84) avec 5555 thèmes disponibles le



Figure 83: Page d'accueil et d'administration Wordpress

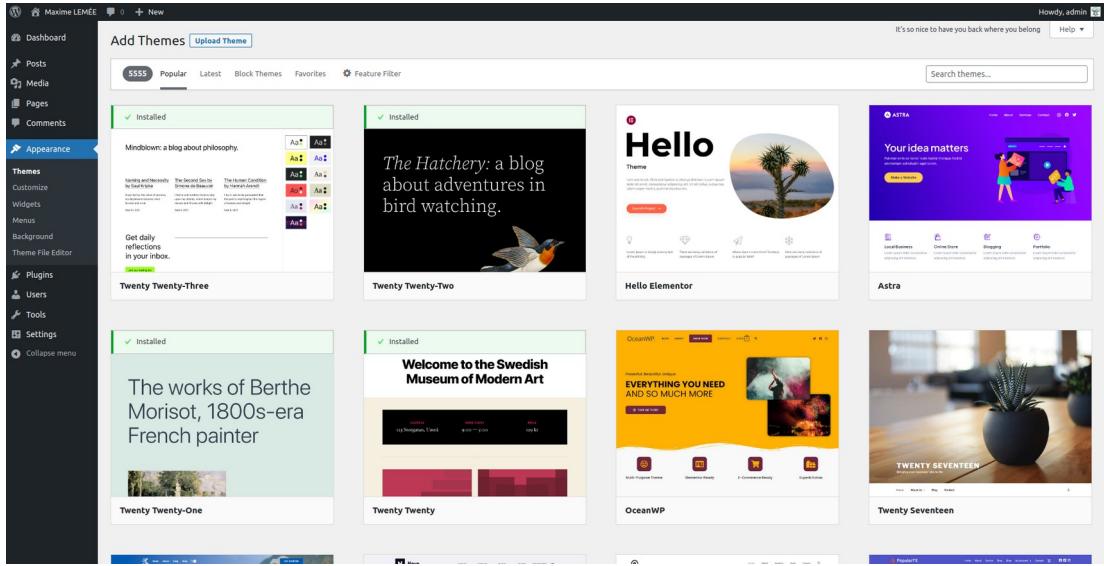


Figure 84: apparence/thème

28/04/2023 et naviguer au sein de celui-ci pour voir les possibilités de personnalisation avec le système Gutenberg de Wordpress.

La méthode Gutenberg permet de créer et/ou de personnaliser des pages à base de blocs très simplement (Figure 85).

Gutenberg permet de personnaliser les pages, de les lier entre elles, de créer des menus, sous menus sur plusieurs niveaux, d'implanter des widgets et vraiment beaucoup d'autres choses.

On a, durant le cours, fait le tour de toutes les fonctionnalités et les possibilités de base en création et administration, avec la gestion des rôles, des plugins.

The screenshot shows the WordPress Gutenberg editor interface. On the left, there's a sidebar with a search bar and tabs for 'Blocks', 'Patterns', and 'Media'. The main content area contains a large heading 'Nouvelle version de PHP' and a sub-section 'Version 8.2'. Below the heading is a bulleted list: '• item A' and '• item B'. At the bottom of the content area is a media block containing a thumbnail of a video titled 'Nocturnal Beasts' with the subtitle 'Wake Up Happy', a play button, and a timestamp '02:26'. The right side of the screen has a toolbar with buttons for 'Switch to draft', 'Preview', 'Update', and other options.

Figure 85: système Gutenberg de Wordpress

2 Back-End

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

La partie Back-End correspond à toute la partie non-visible d'une application ou d'un site web par l'utilisateur, elle englobe toutes les fonctionnalités nécessaires comme un serveur web, une base de données, un langage de script côté serveur (PHP, Java, python...etc) qui va gérer la sécurité des données, leurs accès et également gérer l'administration.

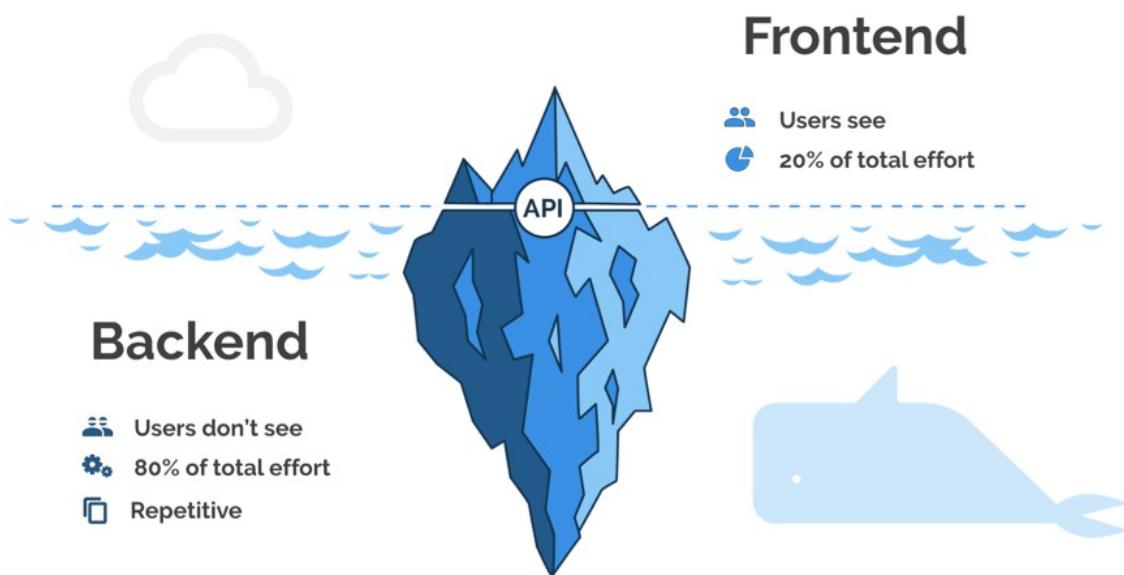


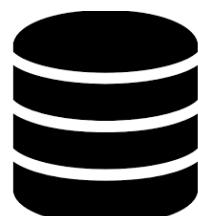
Figure 86: Ce que les utilisateurs ignorent souvent !

2.1. Créer une base de données

2.1.1 Base de donnée

Définition

Une base de données permet de stocker de manière pérenne des données contrairement aux variables qui sont temporaires car en mémoire vive.



Une base de données permet de stocker les données de manière organisée ce qui les rend facilement et rapidement accessibles.

Il en existe 2 grandes familles :

-Les bases de données relationnelles : strictes et sûres. Elles permettent d'intégrer des données à la source en respectant le principe A.C.I.D. (MySQL, Oracle, POSTGRESQL)

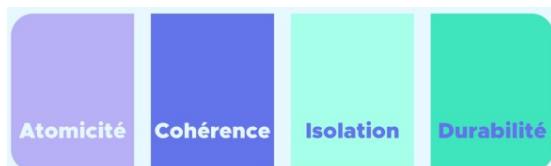


Figure 88: Principe A.C.I.D

-Les bases de données non relationnelles qui sont rapides et souples (MongoDB, Cassandra, Hbase).

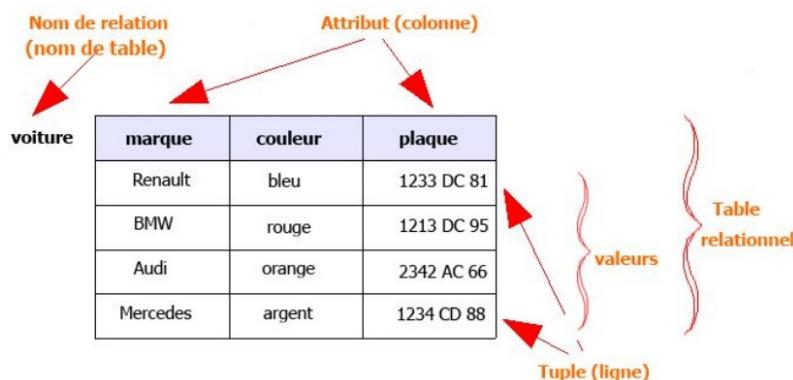


Figure 89: schéma table Bdd

Nous nous sommes concentré sur le style relationnelle qui ressemble visuellement à ceci.

Chaque ligne de la table devra être identifiable par un Id unique et jamais utilisée dans la même table (même en cas de suppression),

On peut attribuer à une table une clé étrangère pour faire référence à une autre table.

Les tables doivent respecter les lois de normalisation qui est une approche de conception permettant d'éviter les anomalies transactionnelles dues à une mauvaise modélisation pouvant entraîner des anomalies de lecture ou d'écriture, la redondance et les contres performances.

- Tous les attributs doivent être atomiques et non répétitifs.
- Une table doit contenir une clé primaire dont dépendent les autres attributs.
- Chaque attribut non identifiant ne doit dépendre que de la clé primaire, et de toute la clé primaire (dans le cas d'une clé composée).

Méthode MERISE MCD / MLD

La méthode MERISE est une méthode permettant la modélisation des données d'un système et la conception d'une base de données en 2 étapes.

MCD : conception conceptuel des données

Premièrement, on identifie les entités et leurs propriétés en n'oubliant pas l'Id propre à chaque entité.

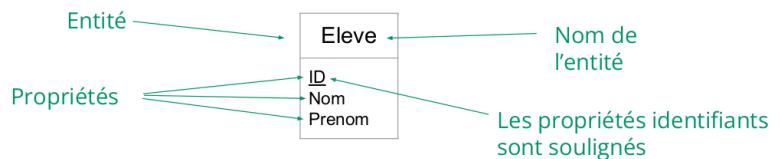


Figure 90: Entité MCD

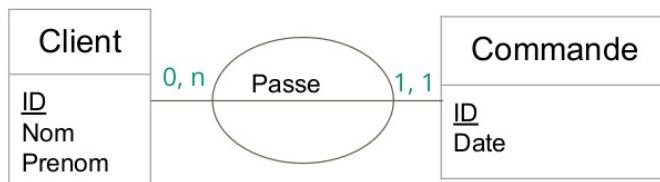


Figure 91: Relation MCD

Ensuite, on met en place les relations entre les entités en nommant celles-là.

Et enfin, on quantifie les relations soit un **0**, **1**(une seule) et **n**(plusieurs)

MLD Modèle logique des données

Le MLD transforme les entités en tables, les propriétés deviennent des attributs et les id deviennent des clés primaires, la représentation ne change pas par rapport au MCD

On regarde les relations entre les tables et en fonction de celle-ci, on effectue les intégrations des clés étrangères.

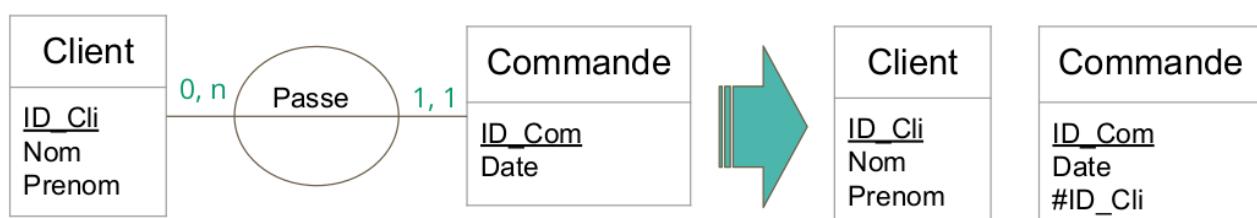


Figure 92: Relation 1-N (one to many)

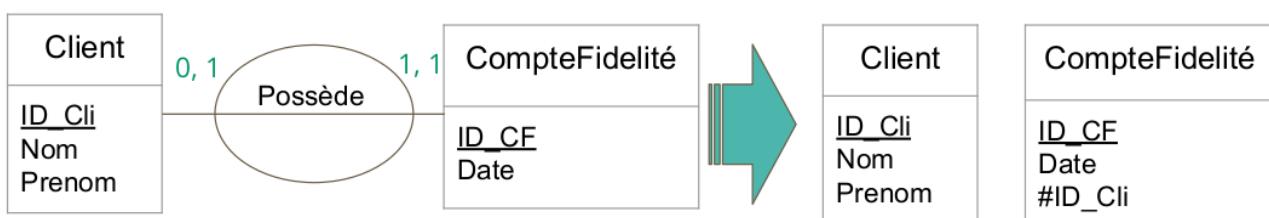


Figure 93: Relation (0,1)-(1,1)

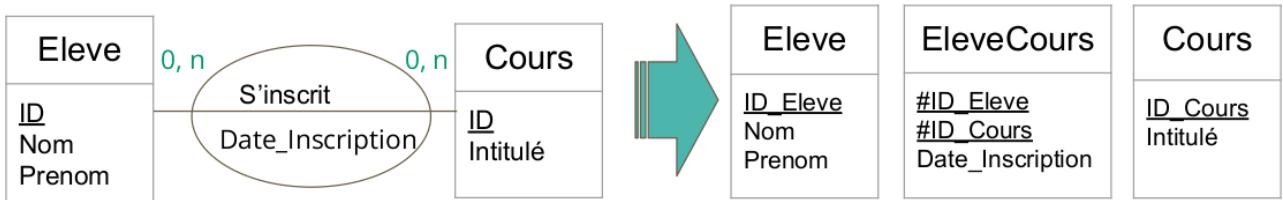


Figure 94: Relation N-N (many to many)

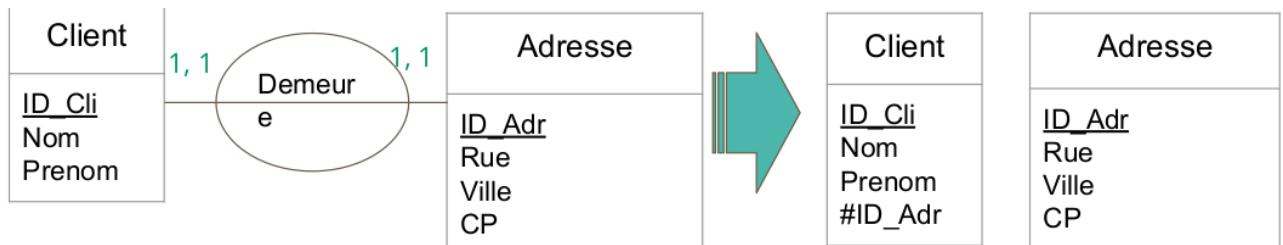


Figure 95: Relation (1,1)-(1,1)

MySQL avec MySQL Workbench

Une fois le schéma validé, on peut maintenant créer la base, cela se fait avec le langage de script SQL.

La liaison avec MySQL, que l'on a installé en amont peut se faire de plein de manière possible, pour ce qui est de notre cas, nous avons utilisé le logiciel MySQL Workbench.

Exemple de script (voir Figure 96: Script SQL)

SQL(Structured Query Language) est un langage standardisé.

Exemple de commande :

- **CREATE DATABASE** nombase => pour la création de la base de donnée ‘nombase’ .
- **USE** nombase => pour l’utilisation de la base de donnée ‘nombase’.
- **DROP DATABASE** nombase => Suppression de la base ‘nombase’.
- **CREATION TABLE** table1base => pour la création d’une table .

(

```
code INT, NOT NULL, AUTO INCREMENT, PRIMARY KEY
nom varchar(100),
prenom varchar(100));
```

```

1 •  create database IIA2;      -- on crée une Base de données
2 •  use IIA2;                -- On fait bien attention à utiliser la bonne base de donné en la sélectionnant
3
4
5 •  create table classe    -- Création de la table classe
6  (
7     ID int primary key NOT NULL auto_increment,
8     nom varchar(100)
9 );
10
11
12 •  create table Eleve   -- Création de la table Eleve
13  (
14     ID int primary key NOT NULL auto_increment,  --
15     nom varchar(100),
16     prénom varchar(100),
17     date_de_naissance date,
18     code_postal int,
19     alternant bool default false ,
20     mail varchar(100),
21     situation_familiale varchar(100),
22     ID_Classe INT NOT NULL,                      -- ID_Classe est la clé étrangère
23     foreign key(ID_Classe) references classe(ID) -- liaison de la clé étrangère avec la table classe
24
25 );
26
27 •  select * from classe ; -- on Affiche la table classe en entière
28 •  select * from Eleve ;  -- on Affiche la table Eleve en entière
29 •  INSERT INTO classe values  --
30     (null, 'BTS'),        -- ID 1
31     (null, 'Licence')    -- ID 2
32 ;
33
34
35

```

Added new script editor

Figure 96: Script SQL

Une des commandes les plus communes est la commande SELECT qui permet de récupérer des informations dans la base afin de les consulter, grâce à une multitude de commandes, on peut cibler précisément les informations voulues.

Exemple de commande :

- **SELECT * FROM nomtable ; =>** On sélectionne tous les éléments de la table nomtable)
- **SELECT nom FROM nomtable ; =>** On sélectionne la colonne nom de la table nomtable)
- **SELECT * FROM nomtable WHERE age > 50 ; =>** On sélectionne tous les éléments de la table nomtable qui ont un age supérieur à 50.

Comme conditions de recherche, on a **=, AND, OR, >, <, ≥, ≤, <>(différent que) , IS NOT NULL**

On peut joindre au SELECT d'autre table avec JOIN pour effectuer des jointures afin d'effectuer des recherches plus spécifiques.

```

Fichier Édition Affichage Rechercher Outils Documents Aide
1-TP BDD.sql (~/Documents/dwwm/BDD/TP)
CREATE TABLE langue_officielle
(
id int primary key NOT NULL auto_increment,
langue VARCHAR(100)
);

INSERT INTO langue_officielle (langue) VALUES
('français'),
('espagnol'),
('portugais'),
('allemand'),
('néerlandais'),
('anglais'),
('arabe'),
('amazighe'),
('amharique'),
('wolof'),
('thaï'),
('hindi'),
('mandarin');

CREATE TABLE pays
(
id int primary key NOT NULL auto_increment,
nom VARCHAR(100),
iso VARCHAR(2),
capital VARCHAR(100),
pib int,
idh FLOAT,
fete_national DATE,
id_cont int NOT NULL,
FOREIGN KEY(id_cont)REFERENCES continent(id)
);

INSERT INTO pays (iso, nom, capital, id_cont, pib, idh, fete_national) VALUES
('FR', 'france', 'paris', 1, 2775, 0.901, '1900-07-14'),
('ES', 'espagne', 'madrid', 1, 1122, 0.904, '1900-10-12'),
('PT', 'portugal', 'lisbonne', 1, 420, 0.864, '1900-06-10'),
('BE', 'belgique', 'bruxelle', 1, 492, 0.916, '1900-07-21'),
('DE', 'allemande', 'berlin', 1, 3781, 0.939, '1900-10-03'),
('NL', 'pays-bas', 'amsterdam', 1, 886, 0.944, '1900-04-27'),
('BR', 'bresil', 'sao paulo', 2, 1364, 0.765, '1900-09-07'),
('MX', 'mexique', 'mexico', 3, 1040, 0.779, '1900-09-16'),
('CA', 'canada', 'ottawa', 3, 1552, 0.926, '1900-07-01'),

```

SQL ▾ Espaces : 4 ▾ Lig 57, Col 14 INS

Figure 97: Exemple du script pour la création de la base

Cas d'usage

Comme travaux pratiques, nous avions à créer une table de données basée sur une liste de pays avec leurs caractéristiques listées sur un fichier Excel et d'effectuer des recherches spécifiques demandées avec des SELECT.

	A	B	C	D	E	F	G	H	I
1	Code ISO	Pays	Capital	Continent	Population (en milliers)	PIB (en milliards IDH)	Langues Officielles	Fête nationale	
2	FR	France	Paris	Europe	68014	2775	0,901 Français	14 juillet	
3	ES	Espagne	Madrid	Europe	46935	1122	0,904 Espagnol	12 octobre	
4	PT	Portugal	Lisbonne	Europe	10303	420	0,864 Portugais	10 juin	
5	BE	Belgique	Bрюссель	Europe	11507	492	0,916 Allemand	21 juillet	
6	DE	Allemagne	Berlin	Europe	83222	3781	0,939 Allemand	3 octobre	
7	NL	Pays-Bas	Amsterdam		17408	886	0,944 Néerlandais	27 avril	
8	BR	Brésil	Sao Paulo	Amérique du Sud	215000	1364	0,765 Portugais	7 septembre	
9	MX	Mexique	Mexico	Amérique du Nord	128650	1040	0,779 Espagnol	16 septembre	
10	CA	Canada	Ottawa	Amérique du Nord	38250	1552	0,926 Français	1 juillet	
11	US	Etats-Unis	Washington	Amérique du Nord	331449	20948	0,926 Anglais	4 juillet	
12	AR	Argentine	Buenos Aires	Amérique du Sud	45480	505	0,845 Espagnol	25 mai	
13	NG	Nigéria	Abuja	Afrique	219463	448	0,532 Anglais	1 octobre	
14	MA	Maroc	Rabat	Afrique	36472	118	0,686 Amazigh	30 juillet	
15	ET	Ethiopie	Addis-Abeba	Afrique	110871	81	0,448 Amharique	28 mai	
16	CM	Cameroon	Yaoundé	Afrique	27745	40	0,59 Anglais	20 mai	
17	SN	Sénégal	Dakar	Afrique	16209	16	0,514 Français	4 avril	
18	IN	Inde	New Delhi	Asie	1326093	2592	0,645 Anglais	15 août	
19	TH	Thaïlande	Bangkok	Asie	68977	509	0,777 Thaï	5 décembre	
20	CH	Chine	Pekin	Asie	1411780	14867	0,761 Mandarin	1 octobre	

Figure 98: Fichier de données au format Excel

The screenshot shows a SQL editor window with multiple sections of code and results.

Code Editor (Top):

```
-- exercice 4
SELECT continent.nom AS continent,
       SUM(population.population) AS population_totale
FROM continent
JOIN pays ON continent.id = pays.id_cont
JOIN pays_population ON pays.id = pays_population.id_pays
JOIN population ON population.id = pays_population.id_popu
GROUP BY continent.id

-- exercice 5
SELECT continent.nom AS Continent,
       MAX(pays.idh) AS IDH_Max,
       MAX(pays.nom) AS Pays
FROM pays
JOIN continent ON pays.id_cont = continent.id
GROUP BY continent.id

-- exercice 6
SELECT langue_officielle.langue AS Langue_Officiel,
       COUNT(pays.id) AS Nbre_de_Pays
FROM langue_officielle
JOIN pays_langue ON langue_officielle.id = pays_langue.id_lang
JOIN pays ON pays.id = pays_langue.id_pays
GROUP BY langue_officielle.id
ORDER BY Nbre_de_Pays ASC
```

Result Grid (Bottom Left):

#	Pays	langue
1	canada	anglais
2	etats...	anglais
3	nigeria	anglais
4	camer...	anglais
5	inde	anglais

SQL Bar (Bottom Right):

SQL ▾ Espaces : 4 ▾ Lig 1, Col 1 INS

Figure 100: Exemple de résultat SELECT

Figure 99: Exemple de SELECT

2.2. Développer les composants d'accès aux données

2.2.1 PHP

Définition

PHP (Hypertext Preprocessor) est un langage de scripts utilisé côté serveur soit seul (from scratch) soit avec un framework(Symfony, Laravel). Il est un des langages back-end les plus utilisés mondialement. Il est open source et sa dernière version PHP8 date de fin 2022.

Initialement faiblement typées, les dernières versions ont rattrapé cela grâce au déclaration de type ajouté depuis la version 7.

Cas d'usage

Nous avons, dans un premier temps, installé PHP sur nos ordinateurs, sur Linux, ce fut relativement simple avec la commande sudo apt-get install PHP.

Par la suite, nous avons commencé notre premier projet et écrit <?php phpinfo() ?>,

une commande qui une fois le fichier exécuté faisait apparaître la page conséquente de version de PHP .

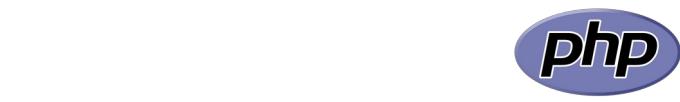


Figure 101:
Logo php



Figure 102: Top 4 languages Back-End 2023

The screenshot shows the configuration page for PHP Version 8.1.2-1ubuntu2.11. It includes sections for System, PHP API, and Configuration. The Zend Engine section at the bottom indicates it's version 4.1.2, Copyright (c) Zend Technologies.

Figure 103: Page version PHP

```
$sexe = "femme";
if ($sexe === "femme") {
    echo 'je suis une femme';
} else {
    echo 'je suis un homme';
}
```

Figure 104: Condition if avec un égalité

```
$heure = (int)13 ;
if($heure >= 8 && $heure <= 12){
    echo 'on est le matin';
}elseif ($heure >= 13 && $heure <= 20){
    echo 'on est l\'après midi';
}else{
    echo "on est la nuit";
}
?>
```

Figure 105: Condition if avec plusieurs instructions

programmation que l'on a déjà vu en algorithmique (Algorithmique appliquée) et en JavaScript comme les conditions IF.

```
<?php foreach ($todos as $todo) : ?>
<li class="todo-item <?= $todo['done'] === true ? 'low-opacity' : '' ?>">
    <span class="todo-name"><?= $todo['name'] ?></span>
    <a href="edit-todo.php?id=<?= $todo['id'] ?>">
        <button class="btn btn-primary btn-small"><?= $todo['done'] ? 'Annuler' : 'Valider' ?></button>
    </a>
    <a href="remove-todo.php?id=<?= $todo['id'] ?>">
        <button class="btn btn-danger btn-small">Supprimer</button>
    </a>
</li>
<?php endforeach; ?>
```

Figure 106: Boucle for sur des éléments HTML

Cookies et Session utilisateur

Pour garder les informations que l'utilisateur peut rentrer durant une session, on utilise des cookies qui permettent de stocker dans le navigateur ces informations.

C'est un ensemble de clés/valeurs.

Avec la fonction `setcookie`, on peut sauvegarder sur la machine client des informations que l'on récupérera avec `$_COOKIE['valeur']` ;

(Figure 107: fonction `$_COOKIE` et `setcookie`)

Une session est une instance de fonctionnement d'un programme qui a un début et une fin.

On peut y accéder via un tableau collaboratif `$_SESSION` créé par la fonction `session_start()` qui va créer un ID de session ou `PHPSESSID`, session que l'on peut détruire quand on le souhaite avec la fonction `session_destroy()` ou `$_SESSION = array()` ;.

```
$couleurFond = "#FFFFFF";
$couleurTexte = "black";

if(isset($_COOKIE['couleurs'])) {
    $couleurs = explode( separator: ',', $_COOKIE['couleurs']);
    if(isset($couleurs['couleurFond'])) {
        $couleurFond = $couleurs['couleurFond'];
    }

    if(isset($couleurs['couleurTexte'])) {
        $couleurTexte = $couleurs['couleurTexte'];
    }
}

if(isset($_POST['couleurFond']) && isset($_POST['couleurTexte'])) {
    $couleurs = [
        'couleurTexte' => $_POST['couleurTexte'],
        'couleurFond' => $_POST['couleurFond'],
    ];
    setcookie('couleurs', implode( separator: ',', $couleurs), time() + 3600*24*7);

    $couleurFond = $_POST['couleurFond'];
    $couleurTexte = $_POST['couleurTexte'];
}
?>
```

Figure 107: fonction `$_COOKIE` et `setcookie`

```
<?php
//Initialisation de la session
session_start();
$erreur = false;
if (isset($_POST['password']) && isset($_POST['nom'])) {
    $mdp = $_POST['password'];
    $nom = $_POST['nom'];

    if ($nom === 'admin' && $mdp === 'admin') {
        //Enregistrement des informations de connexion dans la session
        $_SESSION['administrateur'] = $nom;
        header( header: 'Location: info.php');die;
    }

    $erreur = "Identifiants incorrecte";
}
?>
```

```
<?php

session_start();
session_destroy();

header( header: 'Location: index.php');
```

Figure 109: Destruction de la session

Figure 108: Initialisation de la session

Classe et objet

Nous avons également vu les classes et les objets que l'on peut instancier à travers différents exercices.

Voir chapitre (La programmation orientée objet) pour plus de détails sur le fonctionnement des classes.

```
<?php
require_once('Vehicule.php');

class DeuxRoues extends Vehicule
{
    private int $cylindree;

    public function mettreEssence(float $nombreLitre): void
    {
    }

    /**
     * @return int
     */
    public function getCylindree(): int
    {
        return $this->cylindree;
    }
}
```

Figure 110: Classe DeuxRoues
extends Vehicule

```
<?php
class Vehicule
{
    /**
     * @param float $poids
     */
    public function setPoids(float $poids): void
    {
        $this->poids = $poids;
    }

    /**
     * @param string $couleur
     */
    public function setCouleur(string $couleur): void
    {
        $this->couleur = $couleur;
    }

    private string $couleur;
    private float $poids;

    /**
     * @param string $couleur
     * @param float $poids
     */
    public function __construct(string $couleur, float $poids)
    {
        $this->couleur = $couleur;
        $this->poids = $poids;
    }

    public function rouler(): string
    {
        return 'Le véhicule roule';
    }

    public function ajouterPersonne(float $poidsPersonne): void
    {
        $this->poids += $poidsPersonne;
    }

    public function getPoids(): float
    {
        return $this->poids;
    }

    public function mettreEssence(float $nombreLitre): void
    {
        $this->setPoids($this->poids + $nombreLitre);
    }
}
```

Figure 111: Classe
Vehicule avec méthode

```
<?php
require_once('Vehicule.php');

class DeuxRoues extends Vehicule
{
    private int $cylindree;

    public function mettreEssence(float $nombreLitre): void
    {
    }

    /**
     * @return int
     */
    public function getCylindree(): int
    {
        return $this->cylindree;
    }
}
```

Figure 112: Affichage des objets
instancié

Méthode GET et POST

GET et POST sont les méthodes qui permettent de partager des informations entre les pages de nos applications web.

GET au travers de l'URL et POST via le navigateur, ce sont des méthodes pas du tout sécurisées d'où l'intérêt de faire attention aux informations échangés (voir chapitre Sécurité).

Durant un exercice de Todo, nous devions supprimer un Todo en particulier dans une liste et pour cela, dans le code, si l'utilisateur clique sur le bouton supprimé alors celui-ci est redirigé vers une autre page (remove-todo.php) via une URL dans laquelle nous avons rajouté une clé correspondant à l'id de la Todo en question. On peut envoyer également des infos via une balise HTML <form> en précisant la méthode GET ou POST et en nommant l'élément que l'on veut envoyer avec la balise HTML <name> .

```
<a href="remove-todo.php?id=<?= $todo['id'] ?>">
    <button class="btn btn-danger btn-small">Supprimer</button>
</a>
```

Figure 113: Redirection avec clé 'id' possédant l'id de la Todo

La méthode POST est très semblable sauf que la valeur de la clé n'est pas visible par l'utilisateur 'lambda' mais visible simplement en ouvrant la console du navigateur dans l'onglet application.

On l'envoie dans un formulaire dans lequel on précise bien la méthode POST et la récupère avec `$_POST['valeur']` à l'endroit voulu, ciblé par la balise HTML `<action>`.

```

<?php

if (!isset($_GET['id']) || empty($_GET['id'])) {
    header('Location: index.php');
}

//Récupérer de la todo à modifier, indice : $_GET['id']
$todoId = $_GET['id'];
$filename = "data/todos.json";

//Charger la liste des todos dans le fichier data/todo.json, indice : file_get_contents
if (file_exists($filename)) {
    $data = file_get_contents($filename);
    $todos = json_decode($data, true);
    if(!is_array($todos)) {
        header('Location: index.php');
    }
}

//Chercher la todo à modifier en fonction de l'id, indice : foreach
foreach ($todos as $key => $todo) {
    if($todo['id'] === $todoId) {
        unset($todos[$key]);
    }
}

//On enregistre les modifications dans le fichier data/todo.json, indice : file_put_contents
file_put_contents($filename, json_encode($todos));

//On redirige vers index.php, indice : header('Location: index.php')
header('Location: index.php');

```

Figure 114: Page remove-todo.php avec la récupération de l'id avec `$_GET['id']`

Base de données

Afin de manipuler les données de notre application, nous avons appris comment se connecter à une base de données et de surcroît manipuler ces données comme désirer avec l'extension PDO (PHP Data Object) de PHP

Nous avons commencé par créer une fonction `connectBdd()` qui a le rôle de passerelle entre le script php et la base de données. Elle contient l'adresse de la base et les identifiants nécessaires à la connexion.

Ensuite, nous avons écrit une fonction pour aller chercher en base de données un 'objet' user grâce à une requête SQL avec son mail et son password.

```

function login($email, $password)
{
    $db = connectBdd();
    $sql = "SELECT * FROM user WHERE email = :email AND password = :password";
    $query = $db->prepare($sql);
    $query->execute([
        'email' => $email,
        'password' => $password,
    ]);

    return $query->fetchAll();
}

```

Figure 116: Exemple de SELECT avec PDO

On peut remarquer que l'on a extériorisé le mail et le password dans la requête pour éviter les injections SQL par des utilisateurs mal intentionnés.

```

function connectBdd()
{
    $dsn = 'mysql:host=localhost;dbname=tp_bdd';
    $user = 'max';
    $password = '';

    try {
        return new PDO($dsn, $user, $password);
    } catch (PDOException $e) {
        echo 'connexion échouée : ' . $e->getMessage();
    }
}

```

Figure 115: Function `connectBdd()`

2.2.2 Symfony



Définition

Symfony est un framework PHP back-end qui suit le principe MVC (Model View Controller) (Figure 118: Architecture Symfony) et PSR (PHP Standards Recommendations)

Figure 117:
Logo
Symfony

Il propose une architecture de travail stricte et des outils puissants afin de gagner en temps et en efficacité lors de la conception de l'application web, il nous assure un projet bien organisé, sécurisé et maintenable par plusieurs personnes grâce à une méthodologie claire et structurée.

C'est un framework personnalisable grâce à ses nombreux modules comme Doctrine qui gère la liaison avec la base de données et Twig pour générer des rendus HTML.

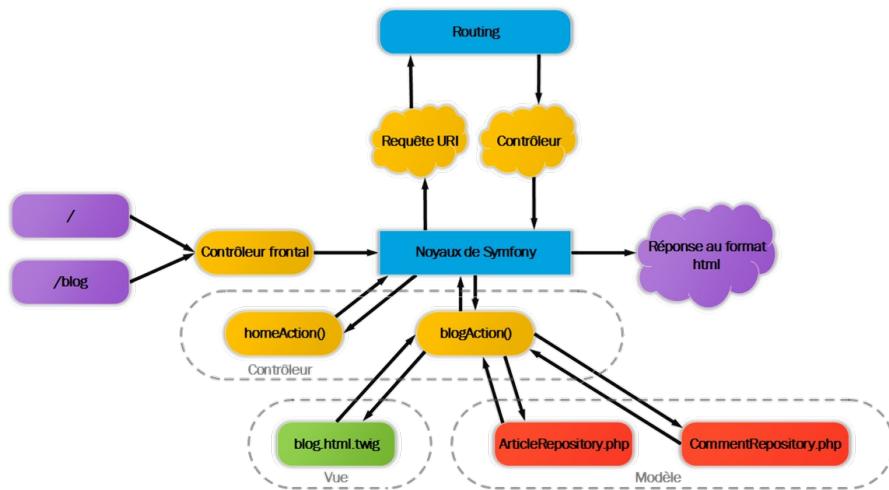


Figure 118: Architecture Symfony

Cas d'usage

Retenant notre projet Quater, dans un premier temps, nous avons créé la base de données après avoir créer le diagramme de classe (Figure 119: Diagramme de classe) et installer composer qui est le gestionnaire de paquet de PHP, l'équivalent de NPM pour JavaScript.

On a créé un nouveau projet avec la commande composer create-project symfony/skeleton: "6.2.*"

On a ainsi créé Quarter et installé npm avec toutes les dépendances nécessaires à l'utilisation de SCSS que l'on utilisaient durant nos cours de CSS/HTML.

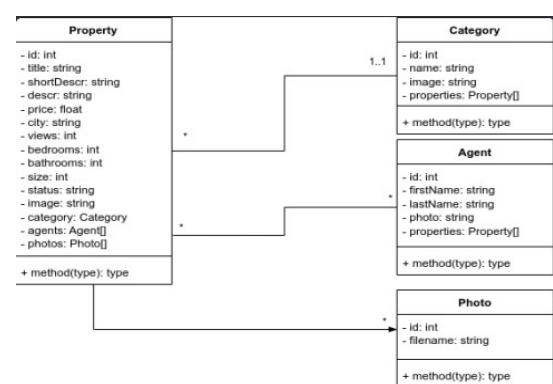


Figure 119: Diagramme de classe

```

Fichier Édition Affichage Rechercher Terminal Aide
maxime@MaxVictus:~/Documents/dwm/Dossier cours/Nouveau dossier$ composer create-project symfony/skeleton:"6.2.*" quarter
Creating a "symfony/skeleton:6.2.*" project at "./quarter"

https://repo.packagist.org could not be fully loaded (The "https://repo.packagist.org/packages.json" file could not be downloaded: SSL: Handshake timed out)
Failed to enable crypto
Failed to open stream: operation failed), package information was loaded from the local cache and may be out of date
Installing symfony/skeleton (v6.2.99)
- Installing symfony/skeleton (v6.2.99): Extracting archive
Created project in /home/maxime/Documents/dwm/Dossier cours/Nouveau dossier/quarter
Loading composer repositories with package information
Info from https://repo.packagist.org: #StandWithUkraine
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking symfony/flex (v2.2.5)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing symfony/flex (v2.2.5): Extracting archive
Generating autoload files
1 package you are using is looking for funding.
Use the `composer fund` command to find out more!

Run composer recipes at any time to see the status of your Symfony recipes.

Loading composer repositories with package information
Restricting packages listed in "symfony/symfony" to "6.2.*"
Updating dependencies
Lock file operations: 29 installs, 0 updates, 0 removals
- Locking psr/cache (3.0.0)
- Locking psr/container (2.0.2)
- Locking psr/event-dispatcher (1.0.0)
- Locking psr/log (3.0.0)
- Locking symfony/cache (v6.2.7)
- Locking symfony/cache-contracts (v3.2.1)
- Locking symfony/config (v6.2.7)
- Locking symfony/console (v6.2.7)
- Locking symfony/dependency-injection (v6.2.7)
- Locking symfony/deprecation-contracts (v3.2.1)
- Locking symfony/doctrine (v6.2.7)
- Locking symfony/error-handler (v6.2.7)
- Locking symfony/event-dispatcher (v6.2.7)
- Locking symfony/event-dispatcher-contracts (v3.2.1)
- Locking symfony/filesystem (v6.2.7)
- Locking symfony/finder (v6.2.7)
- Locking symfony/framework-bundle (v6.2.7)
- Locking symfony/http-foundation (v6.2.7)
- Locking symfony/http-kernel (v6.2.7)
- Locking symfony/polyfill-intl-grapheme (v1.27.0)
- Locking symfony/polyfill-intl-normalizer (v1.27.0)
- Locking symfony/polyfill-mbstring (v1.27.0)
- Locking symfony/routing (v6.2.7)

```

Figure 120: 1 Crédit à la création du projet symfony depuis le terminal bash

Ce qui nous a donné dans le répertoire l'arborescence suivante (Figure 122).

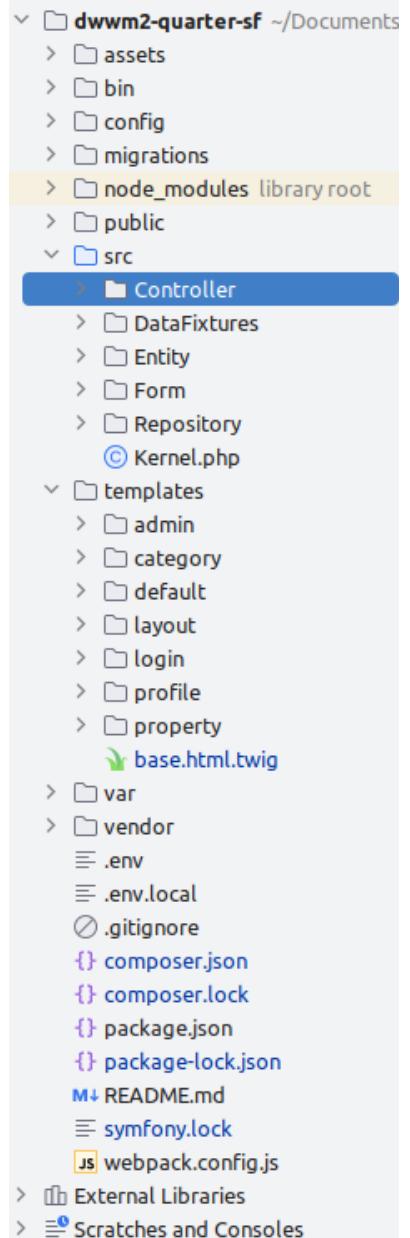
Cette arborescence respecte le principe du MVC :

le M (Model) se situe dans le dossier SRC/Repository

Le V (View) se situe dans le dossier templates

Le C (Controller) se situe dans le dossier SRC/Controller

Figure 121: Arborescence Symfony



Il y a également le dossier Config pour gérer toutes les options Packages.

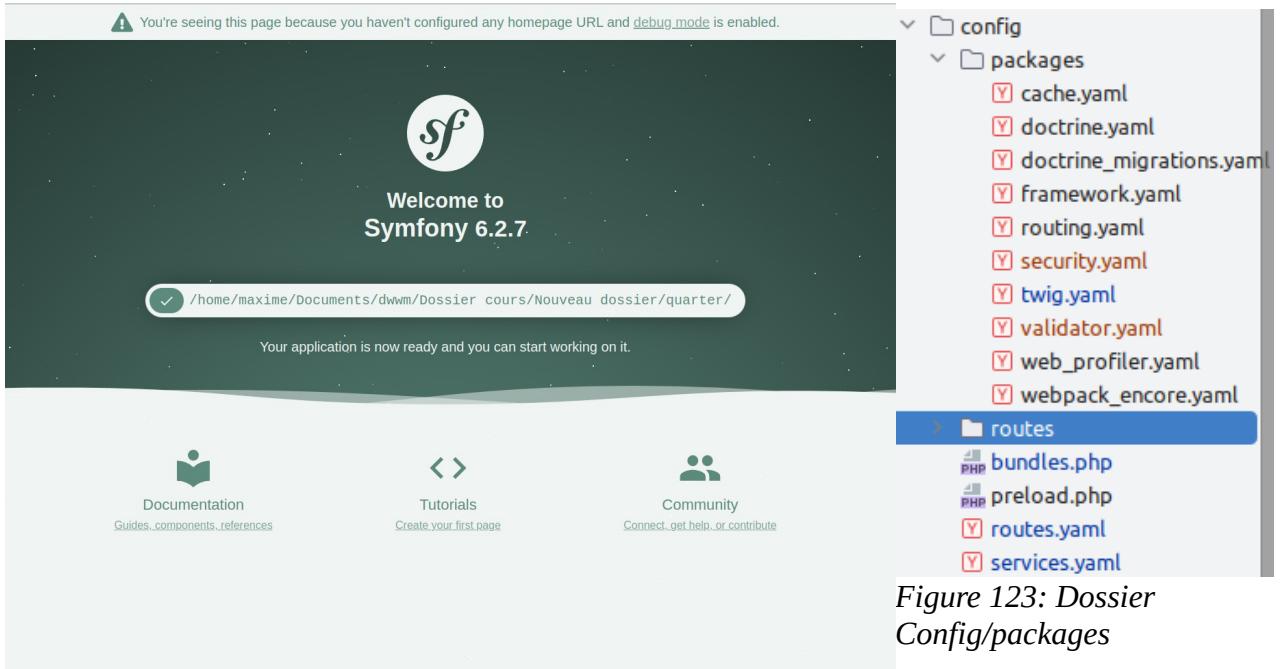


Figure 122: Page d'accueil de symfony

Durant nos cours , nous avons souvent été dans ce dossier pour de la configuration des bundles et des packages

Ensuite, on a installé

- le bundle maker via composer avec la commande `composer create-project symfony/skeleton:6.2.*` 'quarter qui permet de générer du code à partir de commandes simples dans le terminal, afin de créer des entités, des contrôleurs, des formulaires, etc, c'est un outil très puissant.
- Doctrine avec la commande `composer require orm` qui s'occupe de la relation avec la base de données avec ses propres requêtes et sa classe querybuilder.
- Twig avec la commande `composer require twig` qui nous sert à fabriquer les templates HTML avec le langage Twig

Une fois tout cela installé, avec Doctrine, nous avons généré la base de données.

On a créé les entités correspondant avec la commande `php bin/console make:entity` qui va générer les pages correspondantes et le code avec.

```
php bin/console doctrine:database:drop --force
php bin/console doctrine:database:create
php bin/console doctrine:migrations:migrate
```

Figure 124: Commande Doctrine

```
maxime@MaxVictus:~/Documents/dwmm/Dossier cours/Nouveau dossier/quarter$ php bin/console make:entity

Class name of the entity to create or update (e.g. GrumpyPuppy):
> user

created: src/Entity/User.php
created: src/Repository/UserRepository.php

Entity generated! Now let's add some fields!
You can always add more fields later manually or by re-running this command.

New property name (press <return> to stop adding fields):
>

Success!
```

Next: When you're ready, create a migration with `php bin/console make:migration`

```
maxime@MaxVictus:~/Documents/dwmm/Dossier cours/Nouveau dossier/quarter$
```

Figure 125: Exemple commande make:entity

Après avoir généré les entités avec leur repertory associé, on a généré les controllers

```
maxime@MaxVictus:~/Documents/dwmm/Dossier cours/Nouveau dossier/quarter$ php bin/console make:controller

Choose a name for your controller class (e.g. GentlePopsicleController):
> ProfileController

created: src/Controller/ProfileControllerController.php

Success!
```

Next: Open your new controller class and add some pages!

```
maxime@MaxVictus:~/Documents/dwmm/Dossier cours/Nouveau dossier/quarter$
```

Figure 126: Crédit de la controller ProfileController

On a ensuite installé Twig avec la comment composer req Twig et relancé la commande doctrine :migrate pour mettre automatiquement la base de données à jour. On a ensuite généré les fixtures avec `php bin/console make:fixture`, qui servira à créer des données fixtives, dans des fichiers Fixtures dans le dossier `src/DataFixtures`(Figure 121) pour tester le site en développement et lancer par la suite un `doctrine:fixture:load` pour les écrire en base de données.

```

maxime@MaxVictus:~/Documents/dwwm/Dossier cours/Nouveau dossier/quarter$ php bin/console make:fixture
The class name of the fixtures to create (e.g. AppFixtures):
> User

created: src/DataFixtures/User.php

Success!

Next: Open your new fixtures class and start customizing it.
Load your fixtures by running: php bin/console doctrine:fixtures:load
Docs: https://symfony.com/doc/current/bundles/DoctrineFixturesBundle/index.html
maxime@MaxVictus:~/Documents/dwwm/Dossier cours/Nouveau dossier/quarter$ █

```

Figure 127: Création de la fixture User

On a ensuite installé webpack-encore pour la compilation du scss avec la commande

- ✓ compose require symfony/webpack-encore-bundle suivis d'un
- ✓ npm **install** ainsi que
- ✓ npm **install** sass-loader@¹3,0,0 sass **--save-dev** et après quelque manipulations de fichiers,
- ✓ npm run watch pour lancé pour lancé la compilation active .

On a en parallèle de npm run watch, démarré le serveur interne fournis avec php en lançant dans un autre terminal la commande php -S localhost:8000 -t public. On a désormais une application web que l'on peut afficher sur notre localhost :8000 automatiquement mise à jour par rapport au modification live du code.

On a ensuite créé des utilisateurs avec la commande php bin/console make:user qui à Figure 128: Création de l'utilisateur 'User' générer tous les fichiers en fonction des options demandées.

Tout les requêtes SQL se font via Doctrine avec son langage le DQL et sa classe Querybuilder.

```

$user = Doctrine::getTable('user')->findOneById(2);
$user->phone = '+33 1 23 45 67 89';
$user->save();

```

Figure 130: UPDATE doctrine

```

function getCategoryDescription($id)
{
    $q = Doctrine_Query::CREATE()
        ->select('description')
        ->from('Category c')
        ->where('c.category_id = ?', $id)
    ;
    return $q->fetchOne();
}

```

Figure 129: SELECT doctrine

Le moteur de templates

Twig nous à servis à générer le visuel du site grâce à un ensemble de méthodes .

Une bonne pratique est de faire correspondre chaque controller avec un dossier dans le dossier templates(Figure 132) avec,dans chacun un fichiers index.html.twig correspondant.

On a également exporter la partie header et footer dans un dossier que l'on a inclus par la suite avec un { % include 'layout/_header' %} et un { % include 'layout/_footer' %} On a ensuite créé les pages en appelant des fonctions ,des boucles for (Figure 133) des propriétés exportées par le controller et sa route associée(Figure 134)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Quarter{% endblock %}</title>
    <link rel="icon" href="data:image/svg+xml,<svg xmlns=%22http://www.w3.org/2000/svg%22></svg>" />
    {% block stylesheets %}
      {{ encore_entry_link_tags('app') }}
    {% endblock %}

    {% block javascripts %}
      {{ encore_entry_script_tags('app') }}
    {% endblock %}
  </head>
  <body>

    {% include 'layout/_header' %}

    {% block body %}{% endblock %}

    {% include 'layout/_footer' %}

  </body>
</html>
```

Figure 131: base.html.twig / page principale

```
{% block body %}
  <h1>Détail de la catégorie : {{ category.name }}</h1>

  <ul>
    {% for property in category.properties %}
      <li>
        {{ property.title }}
      </li>
    {% endfor %}
  </ul>
{% endblock %}
```

Figure 133: Boucle For Twig avec importation de l'objet categorie du controller



Figure 132: Arborescence Dossier templates

```

#[Route('/category/{id}', name: 'app_category_show', requirements: ['id' => '\d+'])]
public function show(Category $category/*int $id, CategoryRepository $categoryRepository*/): Response
{
    /*
    $category = $categoryRepository->find($id);

    if ($category === null) {
        throw $this->createNotFoundException('Cette catégorie n\'existe pas !');
    }
    */

    return $this->render( view: 'category/show', [
        'category' => $category
    ]);
}

```

Figure 134: CategoryController

Test

Les tests servent à vérifier le bon fonctionnement de l'application sur différents niveaux et à bien valider son intégrité pour sa mise en production suite à sa conception.

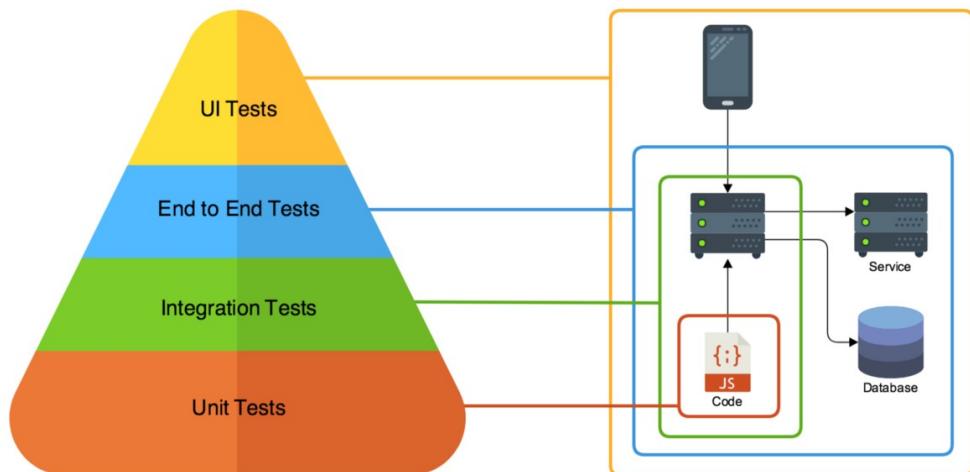


Figure 135: Hiérarchie des tests

2.3. Développer la partie back-end d'une application Web ou Web Mobile

2.3.1 Serveur Web (LAMP)

Définition

Un serveur web est un logiciel avec le matériel nécessaire mis à disposition d'un site ou d'une application web via internet au public ciblé. C'est un ensemble que l'on nomme différemment selon les systèmes mais qui correspond à la même entité, par exemple voici les plus utilisés :

- LAMP pour Linux (système d'exploitation), Apache (logiciel serveur HTTP), MySQL pour la base de données et PHP comme langage de script.
- WAMP Pour Windows (système d'exploitation), Apache (logiciel serveur HTTP), MySQL pour la base de données et PHP comme langage de script.

L'ensemble se compose généralement d'un système d'exploitation (Linux, Windows, mac , etc), d'un logiciel serveur web (apache, nginx), d'une Base de donnée (Oracle, MySQL, PostgreSQL..etc) et d'une langage de script (PHP, python, Perl, JavaScript ...etc)

Cas d'usage

Durant les cours, en premier lieu, nous avons abordé Linux qui est le système le plus utilisé pour gérer des serveurs à l'échelle mondiale, il a l'avantage d'être gratuit.

Open source est réputé fiable. Il possède une grande communauté et est régulièrement mis à jour.

Nous avons travaillé sur des machines distantes que l'école nous a fournies, accessibles via le protocole ssh.

Après ceci, nous avons appris les commandes primordiales pour naviguer dans le terminal Linux car la particularité est qu'il n'y a aucune interface graphique et que toute action quelle qu'elle soit s'effectue sur un terminal en ligne de commande (Figure 138: Terminal de commande Bash)(Figure 137: Exemple de commande de navigation)

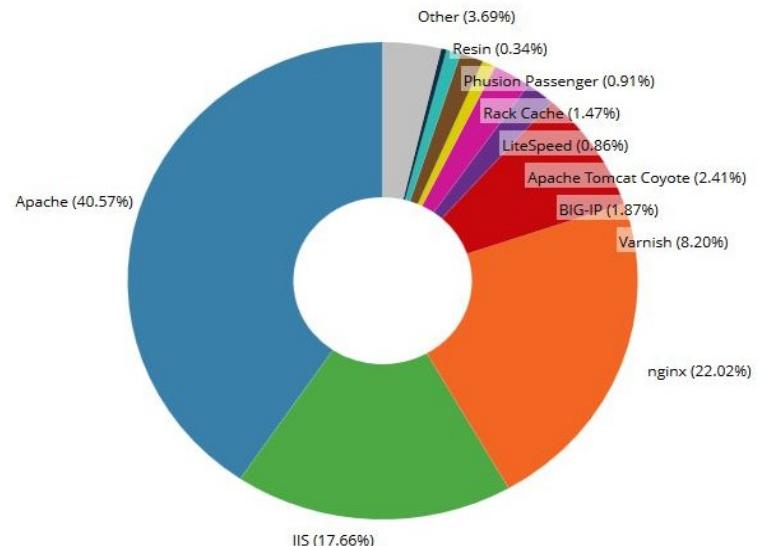


Figure 136: parts de marché des différents "web servers"

Se déplacer dans les dossiers	
/home/utilisateur	répertoire de travail de utilisateur
pwd	montre le nom du dossier de travail courant
cd	se déplacer vers le dossier /home/utilisateur
cd ~/Desktop	se déplacer vers le dossier /home/utilisateur/Desktop
cd ..	se déplacer vers le dossier parent
cd /usr/apt	se déplacer vers le dossier /usr/apt

Lister les fichiers et répertoires

ls -l dossier dir -l dossier	liste le contenu du répertoire dossier en mode détaillé
ls -a dir -a	liste tous les fichiers (y compris les fichiers cachés)
ls -d dir -d	liste les répertoires contenu dans le dossier
ls -t dir -t	trie les fichiers par date en débutant par les plus récents
ls -S dir -S	trie les fichiers par taille en débutant par les plus gros
ls -l more	liste par page grâce à more

Figure 137: Exemple de commande de navigation

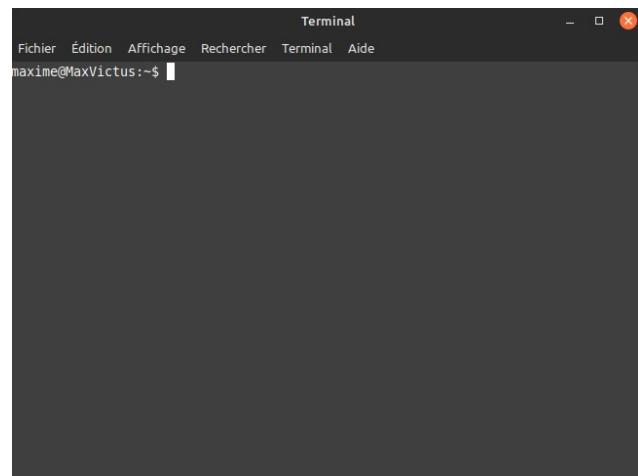


Figure 138: Terminal de commande Bash

/etc/apt/sources.list	fichier contenant la liste des dépôts
apt-get update	met à jour la liste des paquets disponibles à partir de sources.list
apt-get upgrade	met à jour les paquets déjà installés
apt-get dist-upgrade	met à jour votre distribution Ubuntu vers la version supérieure

Figure 140: Commande APT

Parmi les commandes les plus communes il y a les commandes APT(Figure 140: Commande APT) qui font référence à un système de paquet disponible à tous qui permet à partir de commande spécifique d'installé et de désinstallé des paquet(logiciel, pilote, et autres...) en toute simplicité

Durant les cours, nous avons appris à gérer les permissions que la commande ls- l permet de voir.

```
-rw-r--r-- 1 willy tech 62 mai 26 22:55 notes
          |   |
          +--- droits
          |
          +--- utilisateur
                  |
                  +--- propriétaire (u)
          |
          +--- groupe
                  |
                  +--- propriétaire (g)
```

Figure 142: Exemple de ls -l

La gestion des droits est un point très important.Ceux-là peuvent être administrés de 2 manières. Avec la commande CHMOD (Figure 141: Modifier les droits) soit en format en base, soit en symbolique.

cp fichier1 fichier2	copie fichier1 vers fichier2
cp fichier /dossier	copie fichier dans le répertoire dossier
cp -r dossier1 dossier2	copie du répertoire entiers dossier1 vers dossier2
rsync -a dossier1 dossier2	
mv fichier1 fichier2	renomme fichier1 en fichier2
mv dossier1 dossier2	renomme dossier1 en dossier2
mv fichier dossier	déplace fichier dans dossier
mv fichier1 dossier/fichier2	déplace fichier1 dans le répertoire dossier et le renomme en fichier2
mkdir dossier	crée le répertoire dossier
mkdir -p dossier1/dossier2	crée des répertoires imbriqués
rm fichier	supprime fichier du répertoire courant
rmdir dossier	supprime le répertoire dossier
rm -rf dossier	supprime le répertoire non vide dossier (force)
ln -s fichier lien	crée un lien symbolique lien vers fichier (raccourci)
find dossier -name fichier	cherche fichier dans dossier et ses sous répertoires
diff fichier1 fichier2	compare 2 fichiers ou répertoires

Figure 139: Gérer les fichiers et dossiers

- chmod <permissions> <fichiers>
2 formats pour les droits d'accès:
- Format en base 8 (abc):
 $a, b, c = r*4+w*2+x$ (r, w, x: booléens)
Exemple: chmod 644 <fichier>
(rw pour u, r pour g et o)
- Format symbolique. Facile à comprendre par des exemples:
chmod go+r: ajouter droit en lecture au groupe et aux autres
chmod u-w: supprimer droit d'écriture pour le propriétaire
chmod a-x: (a: all = tous). Enlever les droits d'exécution à tous les utilisateurs.

Figure 141: Modifier les droits

Serveur apache

Puis nous avons installé apache via la commande **sudo apt install apache2** et créé notre premier site qui affichait juste un titre et l'heure en PHP .

Les dossiers de configuration d'apache et de Linux en général se trouve dans le dossier `etc/apache2/` et les fichiers de(s) l'application(s) à afficher dans le fichier `var/www/`

Après avoir configuré correctement le fichier de configuration d'apache en créant un `virtualHost`, en précisant le chemin, en donnant un nom d'hôte et d'autres paramètres, on lance la commande `a2ensite 'nomduvirtualHost'` qui active le site (contrairement à la commande `2adisite 'nomduvirtualHost'` qui lui désactive l'accès pour de la maintenance par exemple) et on relance apache2 pour la prise en compte avec la commande `system/reload/apache2`.

Il existe aussi la commande `a2enmode` qui permet l'activation de mod (php par exemple) et `a2dismod` pour la désactivation.

Le site est après tout cela accessible via l'adresse du nom du `virtualHost` via les ports 80(HTTP) et 443(HTTPS) dans le réseau local.

Par sécurité, nous avons modifié le fichier situé `apache2/conf-enabled/security.conf` et modifié le token pour que le navigateur n'affiche pas la version de notre serveur et expose potentiellement ces failles..

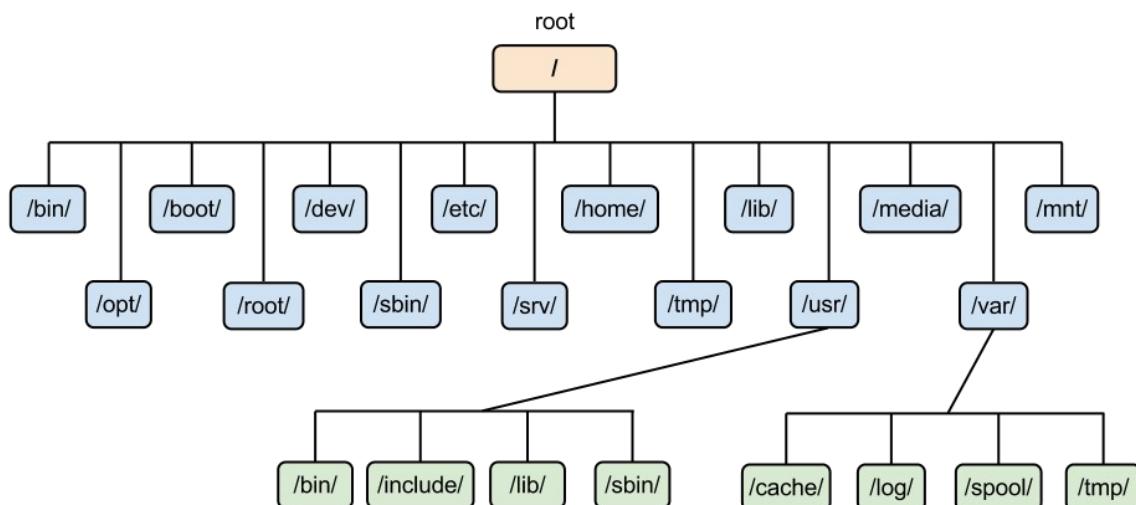
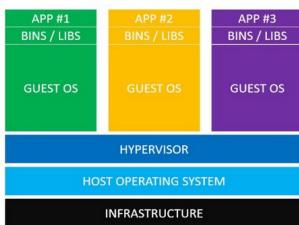


Figure 144: Hiérarchie fichiers Linux

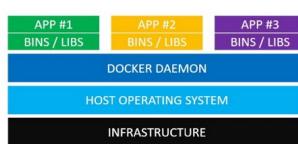
2.3.2 Docker

Définition

Un conteneur Docker est un processus ou un ensemble de processus isolés du reste du système qui reste léger. Docker utilise le principe d'encapsulage et n'est pas une machine virtuelle car à la différence d'un



Virtual Machines



Docker Containers

Figure 146: Difference Virtual Machines (VM) et Docker

VM, le processus ne contient pas toute la partie système d'exploitation (Figure 146) et est pensé dès le départ pour faire des conteneurs applicatifs en les isolants avec leurs dépendances propres dans des conteneurs ayant des cycles de vie courts.

C'est une solution de développement et de mise en production de plus en plus utilisée grâce à sa simplicité d'usage, sa documentation complète et sa

communauté active ainsi que la présence du site docker- hub qui répertorie un très grand nombre d'images prêtes à l'emploi et ce pour une multitude de technologie.

Cas d'usage



Dans un premier temps, nous avons installer Virtual box sur notre machine et installé une version légère d'Ubuntu se prénommant Lubuntu .

Après l'avoir démarré et mis à jour via son interface graphique, nous avons installé docker sur cette dernière en suivant la documentation officielle et effectué un tout premier test en tapant sudo docker run hello-world qui nous a affiché la confirmation de la bonne installation de l'ensemble. Nous pouvons donc travailler sur docker sur une machine virtuelle en toute sécurité car bien isolé.

```
maximed@maxime-VirtualBox:~$ sudo docker run hello-world
[sudo] Mot de passe de maxime :
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:fc6cf906cbfa013e80938cdf0bb199fdbdb86d6e3e013783e5a766f50f5dbce0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Figure 147: docker run hello-world



Figure 145: Logo Docker

Nous avons vu par la suite quelque commandes utiles :

- docker image **ls** pour lister les images présentes sur le système.
- docker ps pour listé les conteneurs en cours d'exécution ou avec l'option -a pour afficher tout les conteneurs actif ou non .
- docker system **prune -a** pour effacer les containers non utilisés
- docker log 'nom du conteneur' pour connaître les logs d'un conteneur en particulier.

Ensuite nous avons 'Dockerisé' une application à l'aide d'un fichier se nommant Dockerfile et situé à la racine du projet (Figure 148). C'est une suite d'instructions données à docker qu'il exécutera de manière procédural lorsqu'on lancera la commande docker run dans un terminal au même niveau de dossier. Pour l'exemple donné, docker va dans l'ordre :

1. Télécharger l'image node:10.24.1-alpine sur le docker-hub
(Cette image contiendra un linux très léger se nommant alpine ainsi que node en version 10.24.1)
2. Prendre le dossier courant et le copier dans /app/ dans le *Dockerfile* conteneur
3. Se déplacer dans le dossier /app/
4. Lancer dans un terminal la commande npm install
5. Exposé sur le port 2368 le contenu de ce dossier
6. Créer un volume
7. Lancer dans un terminal la commande npm run start

```
FROM node:10.24.1-alpine
ADD . /app/
WORKDIR /app
RUN npm install
```

```
EXPOSE 2368
VOLUME /app/logs
CMD npm run start
```

Ensuite, nous avons créé l'image liée au Dockerfile avec la commande

- *docker build -t imageappli* et continué avec la commande
- *docker run -d --name conteneur_appli imageappli* afin de créer le conteneur lié à l'image suivie par la commande
- *docker start conteneur_appli* pour démarrer le conteneur.

Nous avons aussi vu comment rentrer dans le conteneur avec la commande *docker exec -it conteneur_appli bash* ou (sh si bash ne serait pas installé dans la distribution Linux choisi) qui à pour effet de nous ouvrir un terminal directement dans le conteneur.

Docker compose

Nous avons vu docker compose qui est un outil capable d'exécuter des applications multi contenus, il se configure avec un fichier YAML et grâce à une seule commande, peut créer et démarrer le nombre de conteneurs que l'on souhaite avec chacun leurs contenus propres, leur configuration et les différents accès entre eux. Nous avons, par exemple, utilisé un site créé durant les cours sur le framework symfony pour désormais l'utiliser via docker. Pour ce faire, nous avons créé à la racine le fichier docker-compose.yaml(Figure 150), celui-ci est composé ainsi :

- 5 conteneurs distincts avec leurs configurations .
 - **Php**
 - Le build qui dirige vers un fichier Dockerfile spécifique(Figure 151)
 - Les volumes externes au conteneur à utiliser, très utile en développement, et leurs emplacements dans le conteneur
 - Le port à attribuer au conteneur
 - **node**
 - L'image avec sa version
 - Le volume dans lequel le mettre.
 - **bd**
 - L'image de mysql, la version et la configuration adéquate
 - Le volume associé pour éviter d'effacer les données à chaque génération du conteneur
 - **composer**
 - L'image latest (la plus récente)
 - Le volume utilisé
 - **maildev**
 - L'image
 - Une commande liée à sa configuration
 - Les ports à utiliser

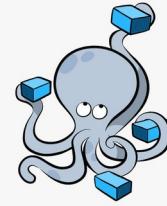


Figure 149: Logo docker compose

```
docker-compose.yml x
version: '3'
services:
  php:
    build: docker/php
    volumes:
      - ./var/www/html
      - /docker/apache/default-vhost:/etc/apache2/sites-enabled/000-default.conf
    ports:
      - 80:80
  node:
    image: node:14
    volumes:
      - ./var/www/html
  db:
    image: mysql:8
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: quarter
    volumes:
      - /docker/data/db:/var/lib/mysql
  composer:
    image: composer:latest
    volumes:
      - ./app
  maildev:
    image: maildev/maildev
    command: bin/maildev --hide-extensions STARTTLS
    ports:
      - "1080:1080"
      - "1025:1025"
```

Figure 150: docker-compose.yaml

```
FROM php:8.1-apache
RUN apt update && apt install -y libicu-dev
RUN docker-php-ext-configure intl
RUN docker-php-ext-install pdo pdo_mysql intl
RUN a2enmod rewrite
RUN usermod -u 1000 www-data
RUN groupmod -g 1000 www-data
```

Figure 151: Dockerfile

Nous avons finalisé avec un *docker compose up -d* pour la création de l'ensemble des conteneurs .

2.4. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

2.4.1 Crée une e_boutique avec Prestashop

Définition

Prestashop est un CMS (content management system) comme Wordpress mais spécialisé dans le commerce en ligne.

Il a été créé en 2007 par 5 étudiants et demeure encore à ce jour gratuit et OpenSource, il est reconnu pour être simple à utiliser et personnalisable grâce à une bibliothèque de modules sur la marketplace dédié

De base lors de son installation, il possède un grand nombre de modules qui couvrent beaucoup de besoins comme la gestion des produits, des catégories, les favoris, de nombreux tableaux de bord...etc.



Figure 152:
Logo

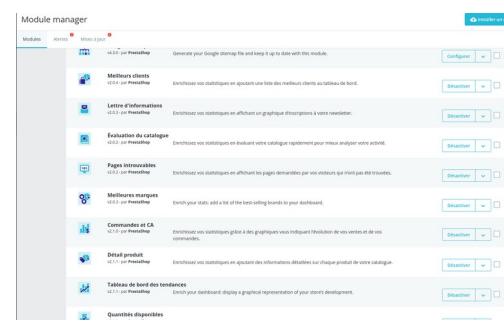


Figure 153: Exemple de modules pré-installé

Cas d'usage

On a, d'abord, téléchargé la dernière version de Prestashop sur le site officiel et placer le dossier dans le dossier var/www/html/, de là, on a pu y accéder grâce au serveur apache installé sur mon système sur mon localhost/nomduuprojet

Une fois la configuration du serveur php et de la base de donnée faite, on arrive sur la page principale, le tableau de bord(Figure 154), celui ci est composé à gauche :

- menu de navigation qui permet de gérer sa boutique (commande, catalogue, client, SAV, Statistiques),
- de la personnaliser avec les modules ou de modifier son apparence, on peut également gérer ses livraisons et les moyens de paiements par ce biais, ainsi que tout ce qui touche aux traductions et à l'international (taxes ou les zones géographiques visé par notre boutique)
- Et en dernier point, tout ce qui touche à la configuration et aux paramètres de la boutique comme les paramètres généraux de l'application, l'administration avec les différents droits suivant le statut de la personne, la gestion de ces personnes (mail, rôle, mot de passe, permissions), la gestion de la base de données et tout ce qui concerne la sécurité.

Au centre va se situer le tableau de bord représentant le déroulé de notre boutique avec beaucoup d'informations la concernant et bien entendu, tout cela entièrement personnalisable et filtrage à volonté.

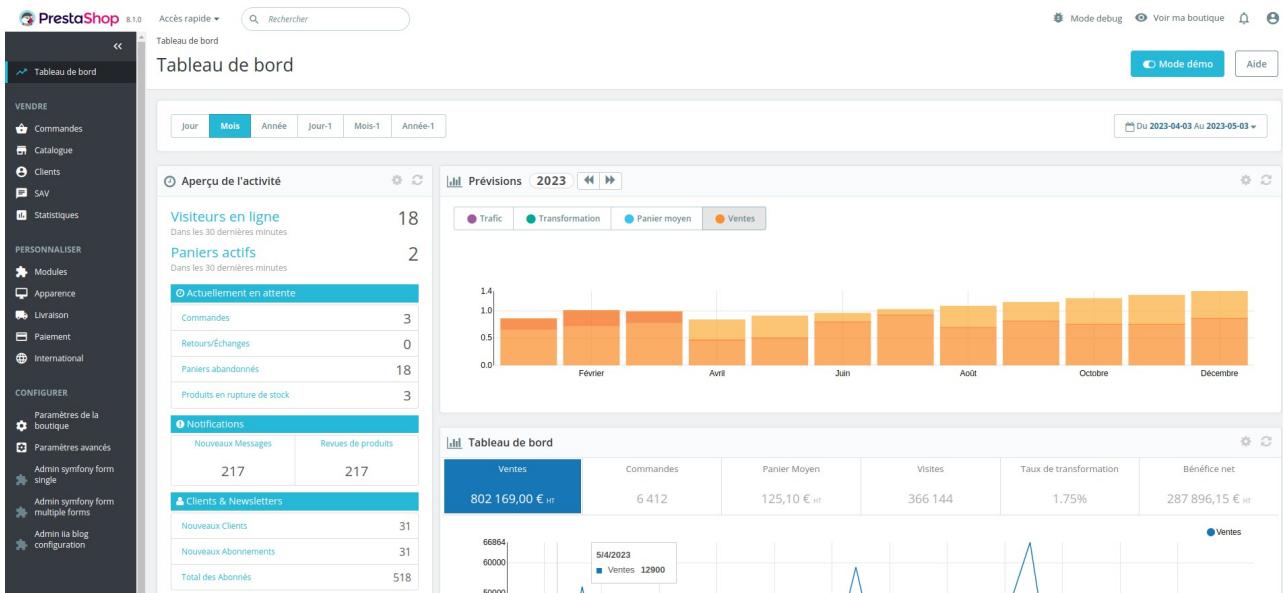


Figure 154: Tableau de bord Prestashop

Durant l'installation, Prestashop nous propose d'activer une boutique démo (Figure 156), à laquelle, nous avons fait quelques modifications comme celles d'ajouter une catégorie, un produit, une marque et un fournisseur, un nouveau transporteur ainsi qu'un nouveau mode de paiement, nous avons également modifié le Logo de notre boutique.

Toutes ces modifications se sont faites à partir du menu présent sur le tableau de bord sans modifier quoi que ce soit dans le code du projet.

Nous avons vu par la suite le fonctionnement interne de Prestashop qui tourne sur le framework symfony 4 et qui est en cours de migration vers la version 6.2 de symfony en utilisant le principe CQRS (Command Query Responsibility Segregation).

C'est un système qui fonctionne à base de commandes et de queries (SQ) avec 2 zones responsables bien distinctes pour traiter les requêtes.

Le CQRS va créer une couche applicative avec laquelle les modules vont pouvoir interagir sans se préoccuper de ce qui se trouve plus loin dans le système, c'est un système à long terme qui permettra des futurs changements sans être

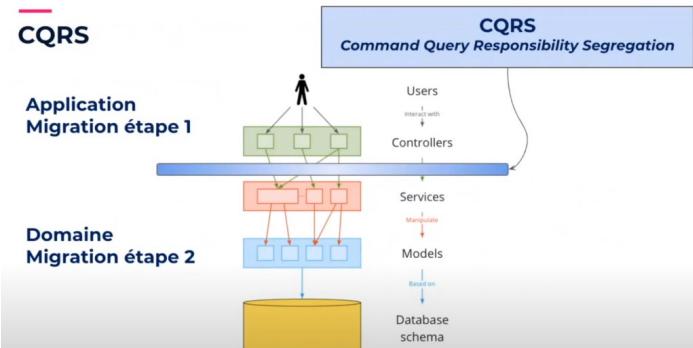


Figure 155: Migration symfony 4 à 6.2 grâce a CQRS

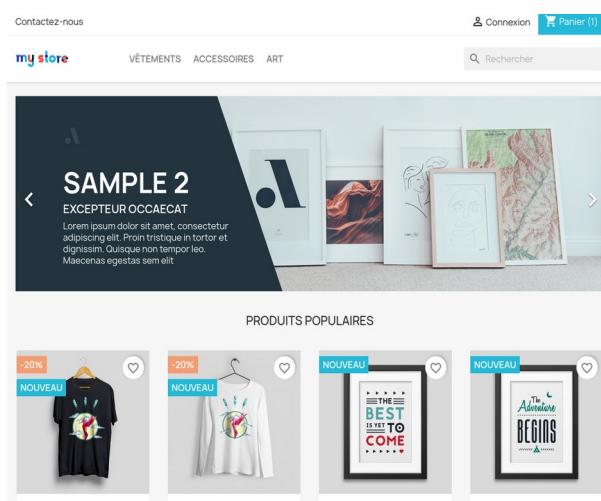


Figure 156: Boutique démo Prestashop

obligé de casser les modules déjà existants. Actuellement, les 2 systèmes, legacy (symfony4) et symfony 6 sont actifs et il est très largement conseillé de construire ses thèmes ou ses modules sur la dernière version.

Prestashop est fourni avec une documentation dédiée au développeur (Figure 157)

Figure 157: <https://devdocs.prestashop-project.org/>

logique d'un projet symfony avec les entities pour les objets, les controllers pour envoyer les données aux différentes routes, form pour les formulaires, et les repository pour les appels à la base de données.

Un dossier views pour tout ce qui est template Twig, d'un fichier `iablog.php` qui est le fichier principal de configuration du module et des fichiers composer pour ce qui est de la configuration des packages.

Prestashop marche avec le principe de hook, ce sont des points d'accroche dans le projet principal sur lesquels, soit on surcharge le code existant avec notre code, soit on y ajoute le nôtre. La liste des hooks est disponible sur la documentation officielle.

Nous avons par la suite construit notre propre module(Figure 158) dans le dossier modules .

Celui ci se compose d'un dossier SQL et d'un fichier pour générer la base de données, d'un dossier config pour les routes et les services, un dossier controller, d'un dossier src qui suit la

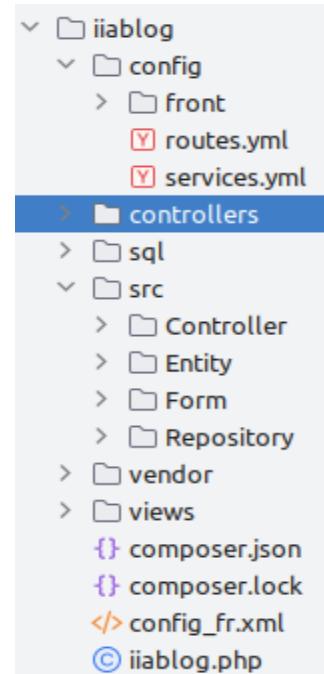


Figure 158: Module IABlog

```
ia_blog_configuration_form:
    path: /iablog/configuration
    methods: [GET, POST]
    defaults:
        _controller: 'IIA\Blog\Controller\ConfigurationController::index'
        # Needed to work with tab system
        _legacy_controller: IIABlogConfigurationForm
        _legacy_link: IIABlogConfigurationForm

ia_blog_category_create:
    path: /iablog/category/create
    methods: [GET, POST]
    defaults:
        _controller: 'IIA\Blog\Controller\BlogCategoryController::createBlogCategory'
```

Figure 159: Crédit des routes

```
public function install()
{
    $hook = [
        'displayLeftColumn',
        'monHook',
    ];

    return
        parent::install()
        // Enregistrement des points d'accroche
        // Traitement des tables pour la base de données
        && $this->registerHook($hook)
        && $this->installSQL()
        && Configuration::updateValue('IABLOG_NB_ARTICLES', 10);
}
```

Figure 160: On intègre les hooks voulus

```

{% set layoutHeaderToolbarBtn = {
    'add' : {
        'href': path('demo_configuration_multiple_forms'),
        'desc': 'Switch to Multiple Forms',
        'icon': 'add_to_photos'
    }
} %}

{% extends '@PrestaShop/Admin/layout.html.twig' %}

{% form_theme configurationType '@PrestaShop/Admin/TwigTemplateForm/prestashop_ui_kit.html.twig' %}

{% block content %}
{{ form_start(configurationType) }}
<div class="card">
    <h3 class="card-header">
        <i class="material-icons">settings</i> {{ 'Text form types'|trans({}, 'Modules.DemoSymfonyForm.Admin') }}
    </h3>
    <div class="card-body">
        <div class="form-wrapper">
            {{ form_widget(configurationType) }}
        </div>
    </div>
    <div class="card-footer">
        <div class="d-flex justify-content-end">
            <button class="btn btn-primary float-right" id="save-button">
                {{ 'Save'|trans({}, 'Admin.Actions') }}
            </button>
        </div>
    </div>
    {{ form_end(configurationType) }}
{% endblock %}

```

Figure 161: Template Twig

```

class iiablog extends Module implements WidgetInterface
{
    public function renderWidget($hookName, array $configuration)
    {
        return "IIA BLOG Widget";
    }

    public function getWidgetVariables($hookName, array $configuration)
    {
        return [
            'hookName' => $hookName,
        ];
    }
}

```

Figure 162: Import des hooks dans le fichier *iiablog.php*

```

public function getTabs()
{
    return [
        [
            'class_name' => 'IIABlogConfigurationForm',
            'visible' => true,
            'name' => 'Admin iia blog configuration',
            'parent_class_name' => 'CONFIGURE',
        ],
    ];
}

class BlogCategoryType extends CommonAbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options): void
    {
        $builder
            ->add( child: 'title', type: TextType::class, [
                'label' => 'Titre de la catégorie',
            ])
            ->add( child: 'description', type: TextareaType::class, [
                'label' => 'Description de la catégorie',
            ])
            ->add( child: 'active', type: CheckboxType::class, [
                'label' => 'Active',
                'required' => false,
            ]);
    }
}

```

Figure 164: Création du lien de configuration

Figure 163: Formulaire nouvelle catégorie

2.4.2 Administrer et personnaliser WORDPRESS

Cas d'usage

Après avoir vu comment réaliser notre site en no-code , on a ensuite abordé le principe des thèmes enfant qui permettent de modifier le thème parent choisi en le surchargeant, afin de modifier ou ajouter des comportements dynamiques ou visuels.



Figure 165: Hiérarchie template Wordpress

Un template Wordpress respecte scrupuleusement une hiérarchie précise(Figure 165) qu'on doit connaître afin de connaître la logique de navigation de l'application, comme la présence obligatoire d'un fichier index.php et d'un style.css .

On affiche le contenu que l'on veut avec la boucle de Wordpress(Figure 166: The loop Wordpress)qui permet d'itérer sur une liste d'article, un seul article, une liste de recette ou tout autres type de contenu

On peut modifier le thème parent, comme je le disais auparavant en surchargeant, grâce à des hooks (point d'accroche) présents au sein du thème parent, la liste de ces hooks se trouve dans la documentation dédiée au développeur (Figure 168) et de celle du template.

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
    <!-- Code HTML pour afficher un ou plusieurs éléments --&gt;
    &lt;?php endwhile; ?&gt;
&lt;?php else: ?&gt;
    <!-- Code HTML si aucun élément à afficher --&gt;
&lt;?php endif; ?&gt;</pre>

```

Figure 166: The loop Wordpress

On a ensuite crée un template de zéro en utilisant du scss (Figure 170) et parmi d'autres choses, on a écrit le code pour le header() en y implantant des fonctions propres à Wordpress pour y placer notre menu de navigation et notre custom logo .

On a ensuite créé un post personnalisé (un custom post type, CPT), par défaut, Wordpress est fourni avec 2 types de contenus, les articles et les pages, mais il est possible de créer grâce au CPT (Custom post type) notre propre type de contenu personnalisé.

Hook

- Hook Actions
- Hook Filtres

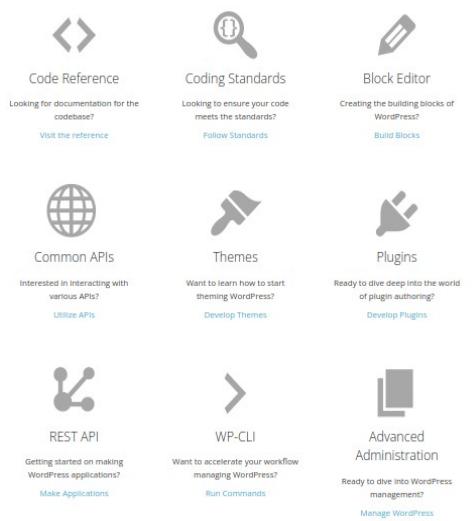


Figure 168: Aide au développeur documentation Wordpress

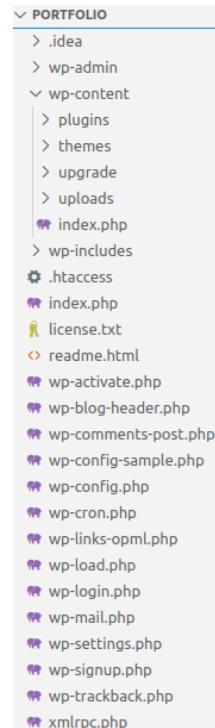


Figure 167: Dossier Wordpress

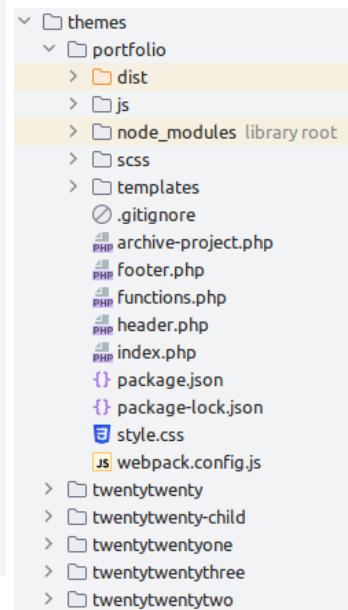


Figure 170: Thème portfolio

```
<!DOCTYPE html>
<html ><?php language_attributes(); ?><
<head>
    <meta charset="<?php bloginfo( show: 'charset'); ?>"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <?php wp_head(); ?>
</head>
<body <?php body_class(); ?>>
<?php wp_body_open(); ?>

<header class="site-header">

    <div class="container">
        <?php if (has_custom_logo()): ?>
            <?php the_custom_logo(); ?>
        <?php else: ?>
            <h1><?= get_bloginfo( show: 'name') ?></h1>
        <?php endif; ?>

        <?php if (has_nav_menu( location: 'main')): ?>
            <?php wp_nav_menu(['theme_location' => 'main', 'container' => 'nav', 'container_class' => 'main-menu']); ?>
        <?php endif; ?>
    </div>
</header>
```

Figure 169: Header,php wordpress

3 Compétences transverses

3.1. Sécurité

3.1.1 Définition

Les systèmes informatiques, quels que soient leurs usages et leurs formes sont de plus en plus présents dans nos vies quotidiennes et de plus en plus indispensables pour nos tâches les plus courantes comme consulter nos comptes, prendre un rendez vous chez un spécialiste, gérer ses données dans un cloud, gérer la gestion de sa société ou sa comptabilité. Les données qu'ils traitent sont en conséquence, de plus en plus sensibles .

Toutes ces raisons font que les opportunités pour les pirates de voler ces données sont de plus en plus fortes. Leurs raisons peuvent être variées : de l'escroquerie au vol de données contre rançons.

La sécurité est désormais omniprésente dans notre façon de voir le code, et celle-ci peut être accentuée grâce :

- A leur confidentialité, qui s'assure que les données ne sont visibles que par les personnes voulues.
- Leur intégrité qui s'assure que les données ne sont pas modifiables.
- La non-répudiation qui permet de garantir qu'une transaction ne peut être niée.
- A des authentifications fortes.

3.1.2 Cas d'usage

Durant le cours, nous avons, dans un premier temps, effectué une simple recherche sur le net pour trouver les dernières attaques informatiques répertoriées par la presse. (Error: Reference source not found). Les résultats parlent d'eux même en démontrant bien que les failles sont toujours présentes et sont régulièrement découvertes et mises à jour, même sur des systèmes présents depuis des années. Tous les secteurs peuvent être touchés comme les sites de commerce, les boîtes mails, les serveurs de jeux vidéos, les applications intranets même si non indexé par google.

Il existe un site officiel français [CERT.FR](#) (Centre gouvernemental de veille, d'alerte et de réponse aux attaques informatiques) qui va

The screenshot shows a search results page for "faille de sécurité" on a French news website. The results include:

- Google Chrome : une faille de sécurité découverte, une mise à jour est nécessaire**
actu | 30 days ago
Le navigateur Chrome de Google fait l'objet d'une **faille de sécurité** qu'il faut corriger par une mise à jour. (©New Africa / Adobe Stock) Google vient de publier vendredi 14 avril 2023 ...
- Alerte de Microsoft sur l'exploitation d'une faille de sécurité importante**
CNETFrance | 5 days ago
Le 23 octobre, Microsoft publiait un bulletin de sécurité critique qui révélait ... Et justement, Symantec a repéré un ver exploitant cette faille le 21 novembre. Le ver est identifié ...
- Une faille de sécurité découverte sur Google Chrome, une mise à jour déployée ...**
Maville | 29 days ago
Il s'agit de la première faille du genre détectée chez Google Chrome en 2023. Photo d'illustration. © REUTERS / Dado Ruvic / Illustration Une faille de ...
- TikTok : cette énorme faille expose vos informations personnelles, mettez vite l'a...**
Tom's guide | 9 days ago
Énormément de données très sensibles ont été dérobées via une faille de sécurité. Si vous êtes un utilisateur de TikTok, mettez à jour votre application ! À lire > WhatsApp, TikTok ...
- Une faille de sécurité d'Apple pourrait permettre aux pirates de contrôler totalement...**
Indigobuzz.fr | 13 days ago
Apple a publié des correctifs pour les principales failles de sécurité iOS et macOS. Les principales vulnérabilités de sécurité trouvées dans les appareils iOS et macOS pourraient permettre aux pirates potentiels de contrôler complètement l ...

Figure 171: Recherche 'faille de sécurité' 17/05/23

répertorié les failles et les attaques connus et leurs statuts ainsi que le site [ANSSI](#) qui possède toutes une série de documents sur les bonnes pratiques à avoir et les recommandations actuelles en matière de sécurité informatique. Il existe également le site OWASP qui est une communauté libre et ouverte à tous travaillant sur la sécurisation des applications web en publant des recommandations régulières.

En tant que développeur, il faut maintenir une veille active sur la sécurité à l'aide à ces différents sites parlant de news informatiques. Nous avons



Figure 173: OWASP

étudié les différentes failles de sécurité existantes et nous nous sommes concentrés sur les principales qui sont les attaques SSX , l'injection SQL, Exposition des données sensibles entre autres , nous les avons étudiez via le site [Root me](#) qui à base d'exercice nous fait comprendre le fonctionnement et comment cela se passe en condition réel

Injection (SQL, Shell,...)

Il s'agit de faire des requêtes SQL à travers les champ de texte disponible sur un site comme les champs de login (Figure 175)ou les champs de recherche afin de récupérer les données voulue dans la base de données comme les mots de passe ou d'autres données sensibles en y passant un script SQL

Cross-site scripting (XSS)

La faille XSS consiste à injecter du contenu, souvent un script dans Javascript par exemple dans un champ de formulaire du code HTML afin entre autres, de récupérer la session de l'utilisateur en cours ou aux cookies, c'est une attaque ciblant d'avantage l'utilisateur que l'application elle même.

Les changements du Top 10 pour 2021

Il y a trois nouvelles catégories, quatre catégories avec un changement de nom et de périmètre, ainsi que des consolidations dans ce Top 10 2021. Nous avons changé les noms si nécessaire pour se concentrer sur la cause plutôt que le symptôme.

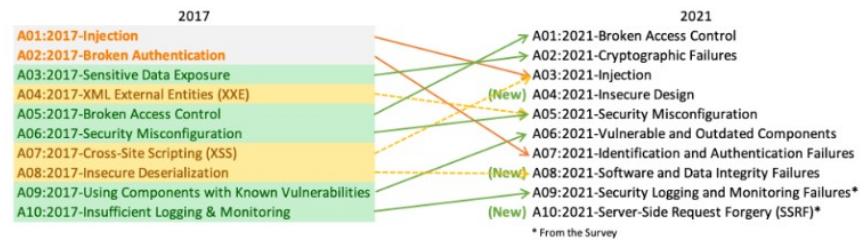


Figure 172: Top 10 des attaques 2021

MENACES ET INCIDENTS

Les rapports Menaces et Incidents détaillent l'état des connaissances et les investigations de l'ANSSI en analyse de la menace et traitements d'incidents		
10 février 2023	CERTFR-2023-CTI-002	Cyber Threat Overview 2022
24 janvier 2023	CERTFR-2023-CTI-001	Panorama de la cybermenace 2022
7 décembre 2022	CERTFR-2022-CTI-007	Intégration of Untrusted Software
23 novembre 2022	CERTFR-2022-CTI-006	Illustration des problématiques liées à l'intégration de logiciels non maîtrisés
25 mai 2022	CERTFR-2022-CTI-005	Menaces liées aux vols de cookies et contre-mesures

Figure 174: CERT.FR
comprendre le fonctionnement et comment cela se passe en condition réel

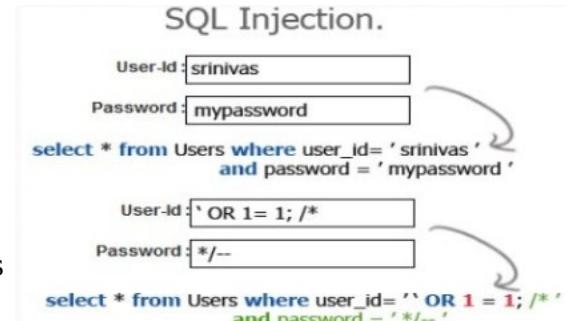


Figure 175: Injection SQL

Figure 176: Faille XSS

Cross Site Request Forgery (CSRF)

Un utilisateur se connecte sur un site et ouvre une session, il va ensuite sur un autre site créé par un hacker et exécute une action quelconque, celui là va alors envoyer une requête HTTP et usurpé l'identité de l'utilisateur, à partir de là, plusieurs scénario s'offre à lui comme l'intrusion d'un logiciel malveillant sur l'ordinateur ou reproduire la page et l'ensemble de la requête HTTP que l'utilisateur à fait pour accéder au premier site afin d'usurpé l'identité du site et tenté de lui soutiré des informations personnelles. L'attaque CSRF est une attaque dangereuse qui vise souvent à usurpé des personnes ou des sites.

3.1.3 TP

Pour travaux pratique, nous avions à examiner un site, de cibler les failles et de trouver les solutions adéquates.

Pour lancé des procédures de test, j'ai utilisé l'application Nikto (Figure 177) ainsi que owasp zap qui sont 2 logiciels qui ont testé le site en l'attaquant en ciblant toute les failles possibles.

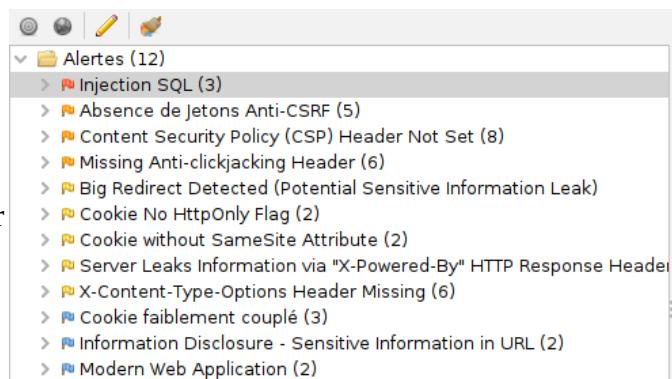
La base de donnée stockait les mots de passe en clair(Figure 179) ce qui est pas du tout conseillé et non sécurisé

J'ai utilisé la fonction php password_hash pour les hachés(Figure 180) et ainsi les rendre plus difficile d'accès en cas d'attaque.

Ensuite j'ai remarqué que les requêtes SQL n'était pas préparé et donc non sécurisé contre les injection SQL .(Figure 183), j'ai donc modifié le code pour sécuriser en la transformant en requête préparée(Figure 182).

```
Fichier Édition Affichage Rechercher Terminal Aide
maxime@MaxVictus:~$ nikto -h http://localhost:8080
- Nikto v2.1.5
-----
+ Target IP: 127.0.0.1
+ Target Hostname: localhost
+ Target Port: 8080
+ Start Time: 2023-03-28 14:52:02 (GMT2)
-----
+ Server: No banner retrieved
+ Retrieved x-powered-by header: PHP/8.1.2-ubuntu2.11
+ The anti-clickjacking X-Frame-Options header is not present.
+ Cookie PHPSESSID created without the httponly flag
+ No CGI Directories found (use '-c all' to force check all possible dirs)
+ OSVDB-44056: /sips/sipssys/users/a/admin/user: SIPS v0.2.2 allows user account info (including password) to be retrieved remotely.
+ OSVDB-27071: /phimageview.php?pic=javascript:alert(8754): PHP Image View 1.0 is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ OSVDB-3931: /myphphnuke/links.php?op=MostPopular&ratenum=[script]alert(document.cookie);[/script]&ratype=percent: myphphnuke is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ /modules.php?op=modload&nname=FAQ&file=index&myfaq=yes&id_cat=1&categories=%3Cimg%20src=javascript:alert(9456);%3E&parent_id=0: Post Nuke 0.7.2.3-Phoenix is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ /modules.php?letter=%22%3Cimg%20src=javascript.alert(document.cookie);%3E&op=modload&nname=Members_list&file=index: Post Nuke 0.7.2.3-Phoenix is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ OSVDB-4598: /members.asp?SF=%22;]alert(223344);function%20x(){v%20=%22: Web Wiz Forums ver. 7.01 and below is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ OSVDB-2946: /forum/members.asp?find=%22;]alert(9823);function%20x(){v%20=%22: Web Wiz Forums ver. 7.01 and below is vulnerable to Cross Site Scripting (XSS). http://www.cert.org/advisories/CA-2000-02.html.
+ OSVDB-3092: /css: This might be interesting...
+ OSVDB-18114: /reports/rwserverlet/server=repser+report=/tmp/hacker.rdf+destype=cache+desformat=PDF: Oracle Reports rwserver report Variable Arbitrary Report Executable Execution
+ 6544 items checked: 24 error(s) and 12 item(s) reported on remote host
+ End Time: 2023-03-28 14:52:06 (GMT2) (4 seconds)
-----
+ 1 host(s) tested
maxime@MaxVictus:~$
```

Figure 177: test nikto



WHERE			
	id	username	email
1	1	mflasquin	maxime.flasquin@gmail.com
2	2	maxime	maxime@mail.com

Figure 178: Test owasp zap

Figure 179: Mot de passe en clair

	id	username	email	password
1	1	mflasquin	maxime.flasquin@gmail.com	password
2	2	maxime	maxime@mail.com	password
3	3	claude	claudem@mail.com	\$2y\$10\$vgk8wKEpTrWs3Ll0fJQ00N5sXnGatF7wEic92FCZTb7g8t1ZguZu
4	4	claude	claudem@mail.com	\$2y\$10\$UQdy410rtad5x9ZT4dZnC.Cr2/zB.czgMXIcHuTFK140Vawrb4eNS

Figure 180: mot de passe haché

en effet, cela à pour effet d'extériorisé les paramètres et d'éviter les injections extérieur. .

```
$result = saveUser($_POST['username'], $_POST['email'],password_hash($_POST['password'], algo: PASSWORD_DEFAULT) );
if($result == true) {
    header( header: 'Location: index.php');
} else {
    echo "Une erreur est survenue " . $result;
}
```

Figure 181: fonction php password_hash

```
function logUser($email, $password)
{
    $connexion = connectDb();
    $sql = 'SELECT * FROM users WHERE email = "' . $email . '" AND password = "' . $password . '"';
    $stmt = $connexion->prepare($sql);
    $stmt->execute();

    return $stmt->fetchAll( mode: PDO::FETCH_OBJ);
```

Figure 183: Requête non préparé

```
function logUser($email, $password)
{
    $connexion = connectDb();
    $sql = 'SELECT * FROM users WHERE email = :email AND password = :password';
    $stmt = $connexion->prepare($sql);
    $stmt->execute([
        'email'=>$email,
        'password'=>$password
    ]);

    return $stmt->fetchAll( mode: PDO::FETCH_OBJ);
```

Figure 182: Requête préparé

J'ai également trouvé une faille XSS dans un champ de formulaire

J'y est appliqué la fonction php htmlentities qui évitera au navigateur d'exécuter ce qu'il pourrait prendre pour du code lors de sa lecture (Figure 184).

```
$users = logUser(htmlentities($_GET['email']), htmlentities($passwordInBdd));
if (!empty($users)) {
    $user = $users[0];
    $_SESSION['user'] = $user;
```

Figure 184: htmlentities pour contrer les attaques XSS

3.2. RGPD

Règlement générale de la protection des données personnelles

C'est un droit européen mis en application le 25 mai 2018 pour protéger les données à caractère personnel, ce règlement renforce le droit des personnes et responsabilise davantage les organismes publics et privés qui traitent les données (*Règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016, relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données*).

Définitions d'une donnée personnelle (art.4 RGPD) : Une donnée personnelle est toute information se rapportant à une personne identifiée ou identifiable.

Ce droit s'applique à toutes les organisations quelles qu'elles soient, qui traitent au sein de l'Union européen des données personnelles.

Ce droit est aussi imposé aux organisations hors UE qui traitent des données de résidents européens.

Un internaute, identifiable par son IP lors d'une visite sur un site laisse des informations sur sa position, son temps de connexion, son nom, son âge et d'autres données.

Il n'en a pas nécessairement conscience mais ces informations restent le temps de sa connexion ou même plus longtemps sur une session d'utilisateur. Et certaines sociétés comme Facebook ou Google sont très friandes de ces données car elles revêtent une importance stratégique fondamentale.

Ces données, pour les organisations les possédants, doivent être :

- Traitées de manière licite, loyale et transparente au regard de la personne concernée.
- Collectées pour des finalités déterminées, explicites et légitimes.
- Adéquates, pertinentes et limitées à ce qui est nécessaire au regard des finalités.
- Exactes et, si nécessaire, tenues à jour et si inexactes soient effacées ou rectifiées sans tarder.
- Conservées sous une forme permettant l'identification des personnes pendant une durée n'excédant pas celle nécessaire.
- Traitées de façon à garantir une sécurité appropriée.

Ce droit oblige à informer l'individu de manière claire et précise :

- La finalité de la collecte
- Du fondement juridique qui autorise le traitement des données.
- Les personnes ou services qui ont accès aux données.
- De la durée de conservation

- Des modalités d'exercice de ses droits d'accès, d'opposition, de rectification et de comment l'effectuer : par mail, sur son compte personnel, par voie postale, à un service identifié
- L'identité du responsable du fichier

On a vu qu'il était interdit d'utiliser l'adresse mail de la personne à des fins de prospection commerciale sans avoir préalablement obtenu son consentement et sans offrir au destinataire une faculté de désinscription.

Selon l'article 9 du RGPD, le traitement des données sensibles est interdit

- l'origine dite raciale ou éthique
- Les opinions politiques, philosophiques ou religieuses
- L'appartenance syndicale
- Un état de santé
- Une orientation sexuelle
- Les données génétiques ou biométriques
-

C'est le responsable du traitement qui est responsable (article 24 du RGPD) et cela même si l'entreprise fait sous traiter la gestion de ses données à un hébergeur par exemple.

3.2.1 Le registre

Toute organisation gérant des données personnelles doit maintenir un registre complet sous forme écrite et/ou forme électronique, mis à disposition des contrôleurs et qui identifiera les activités principales de l'entreprise.

C'est un document comportant

- le nom et les coordonnées du responsable du traitement
- Les finalités du traitement
- Une description des catégories de personnes concernées
- Les catégories de destinataires auxquelles les données personnelles ont été ou seront communiquées
- La cas échéant, les transferts de données à caractère personnel vers un pays tiers ou à une organisation internationale.
- Les délais prévus pour l'effacement des différentes catégories de données
- Une description générale des mesures de sécurité techniques et organisationnelles

La désignation d'un délégué est obligatoire pour les entreprises opérant des traitements à grande échelle et dans les autres cas elle est recommandée.

Le délégué désigné a pour missions

- D'informer et conseiller le responsable du traitement ou le sous traitant,
- De contrôler le respect du règlement,
- De dispenser des conseils en ce qui concerne l'analyse d'impact et de bien vérifier l'exécution de celle ci,
- Coopérer avec les autorités de contrôle
- De faire office de point de contact pour l'autorité de contrôle sur les questions relatives au traitement .

3.2.2 LA CNIL

La CNIL (Commission nationale de l'informatique et des libertés) est une commission créée en 1978 indépendante dans laquelle siège 18 membres élus pour 5 ans(Figure 185).

Elle a comme pouvoir :

- Fonction consultative sur les projets de loi et les décrets
- Analyse les conséquences des nouvelles technologies sur la vie privée des personnes
- A un pouvoir de contrôle et de sanction à l'égard de ceux qui ne respecteraient pas la RGPD, elle est même l'autorité de contrôle imposé par le RGPD
- Elle a également comme mission d'informer les particuliers comme les professionnels en menant des actions de communications grand public à travers tous les médias existants (presse, site internet, mail, réseaux sociaux)



Figure 185: Composition CNIL

C'est elle qui reçoit les plaintes des particuliers comme par exemple l'opposition au mail publicitaire, la vidéo-surveillance, géolocalisation des véhicules dans le milieu professionnel, les difficultés à exercer son droit d'accès.

Suite à ces plaintes, la CNIL peut soit :

- Prononcer un avertissement
- Mettre en demeure l'entreprise
- Limiter temporairement ou définitivement un traitement
- Ordonner de satisfaire aux demandes d'exercice des droits des personnes
- Ordonner la rectification, la limitation ou l'effacement des données.
- Prononcer une amende administrative pouvant aller jusqu'à 4 % du chiffre d'affaire

3.3. Veille technologique

3.3.1 Définition

Le monde de la technologie est un domaine en perpétuelle évolution.

La veille technologique consiste à rester informer des faits, innovations et manipulation sur les technologies que l'on maîtrise afin de proposer et de travailler avec toutes les dernières informations.



Figure 186: L'info est partout !

Il y a la veille passive qui est le fait de suivre globalement l'information des nouvelles technologies, les nouveautés ou les problèmes dans le monde dans la tech.

Et il y a la veille active qui se résume en 4 étapes :

1. Ciblage :

- Qu'est ce que c'est ? / Comment cela marche? / A quoi cela sert? / A qui? / Avantage et inconvénient ?

2. Récolte

- Moteur de recherche / Livre / Bouche à oreille / site de news



3. Analyse

- Comprendre le sujet / Avantages et inconvénients / Enjeux / Coût .

4. Présentation

- Contextualisation / Citer sources / Clair et net / Présentation neutre / données précises

3.3.2 Cas d'usage

Durant le cours on a eu une présentation sur le sujet tech de notre choix parmi une liste donnée, j'ai choisi le sujet 'Cloud computing' . ([Liens google Slides](#)) et conclut avec ses avantages et inconvénients.

3.4. Anglais

La langue anglaise est omniprésente dans le monde informatique, que ce soit au sein des langages informatiques les plus utilisés comme dans le PHP, le python, Javascript ou le java.

Il est aussi largement utilisé comme langage de base pour tout ce qui concerne les différentes documentations et également les sites d'aides entre développeurs comme stack overflow et github.

Même si aujourd'hui beaucoup de tutoriels peuvent être trouvés en français, les tutoriels anglais sont encore plus nombreux et peuvent être plus variés. L'anglais est la langue utilisée comme moyen de communication entre les différents développeurs quand ceux-ci viennent de plusieurs nations différentes.

3.4.1 Cas d'usage

Durant les cours, mon niveau n'étant pas très élevé, nous avons commencé à revoir les bases comme le vocabulaire lié à notre domaine d'étude

Example of part : a screen, the system Unit, a mouse, a keyword

Example and types of computers : Laptop, Netbook, Hybrid, tablet, Smartphone

we have describe the devices and compare them to one another

Nous avons revu les termes de comparaisons

example : A laptop is as small as a netbook, A desktop computer is bigger than a laptop

Nous nous sommes présentés en anglais et avons revu beaucoup de vocabulaire anglais ainsi que les tournures de phrases correctes.

Nous avons pratiqué en simulant des situations comme une discussion avec un collègue en arrivant dans une nouvelle société, ou un appel au téléphone afin de passer un message à un collègue absent.

Ces mises en situations nous ont fait travailler la prononciation des numéros de téléphone, des adresses postales et mails ainsi que le vocabulaire type pour faire répéter la personne ou lui demander de parler moins vite ou plus fort.

Durant la formation, nous avions à présenter chacun toutes les 3 semaines, un sujet de notre choix pour une durée de 5 min, pour nous entraîner à la prononciation face à des personnes .

Nous avons revu également beaucoup de vocabulaire, ainsi que les modaux, les verbes irréguliers, les nombres ordinaux, les questions, les différents pronoms, les formules de politesse, ...etc