

RAPPORT DE STAGE – BTS SIO SLAM 2eme année – Janvier/Février 2022



A. Présentation de l'administration

Les DANE (Délégation Académique au Numérique Educatif) sont des services de l'Education Nationale réparties dans toutes les Académies en France afin d'assurer l'utilisation du numérique en France. Travaillant avec les recteurs, les secrétaires généraux des Académies et tous les acteurs de l'éducation, la DANE maintient des relations et des partenariats entre les domaines de recherche et le corps enseignant par le biais de formations et de projets afin de concevoir des outils utiles dans l'e-éducation.

La DANE de l'académie de Nancy-Metz a été créée en 2013 et s'est par la suite implantée à Nancy, 10 rue de Santifontaine. Christine Françoise en est à la tête, accompagnée de son adjoint : Jean-Albert Foffano. Leurs équipes, d'une trentaine de personnes, cherchent à concevoir des stratégies d'apprentissage grâce au numérique pour les classes allant de l'école primaire au lycée. Ils mobilisent, forment, animent, et organisent toutes sortes de personnels et d'évènements pour valoriser l'utilisation du numérique dans l'enseignement.

B. Contexte de mon projet

Le projet LUNE est une plateforme web pour répondre aux besoins des professeurs sur la recherche de ressources pédagogiques. Il facilite le partage des informations disponibles sur les différentes plateformes de l'Education Nationale grâce à un moteur de recherche et une intelligence artificielle de recommandation.

Un enseignant qui cherche une ressource pédagogique, pour n'importe quelle matière, se perd vite devant les milliers de liens proposés par les moteurs de recherche globaux. Les ressources sont souvent éloignées, pauvres en informations et le professeur finit par abandonner ses recherches. Son cours ne sera donc pas enrichi grâce à ces ressources, ce qui représente une perte pour l'enseignant comme pour les élèves. C'est pourquoi la DANE souhaite réaliser un portail qui proposera de retrouver les nombreuses ressources disponibles, en libre accès et gratuites dans l'ensemble de l'Académie Nancy-Metz.

Le projet LUNE n'en est qu'à ses débuts en termes de fonctionnalités. C'est pourquoi le service m'a engagé en tant que stagiaire développeur afin d'implémenter de nouvelles fonctionnalités pour ce projet.

C. Contraintes

En termes de langage de développement côté Back-end, la plateforme adopte une architecture MVC. C'est le framework Symfony qui a été choisi pour développer cette plateforme afin d'avoir une meilleure structure du code. Ainsi que le langage SQL pour permettre de communiquer avec la base de données pour effectuer des insertions, des modifications ou des suppressions...

En termes de langage de développement côté Front-end, j'ai utilisé la bibliothèque JQuery pour faciliter l'utilisation de fonctionnalités et de syntaxe de JavaScript pour permettre de gérer les événements dû aux actions des utilisateurs.

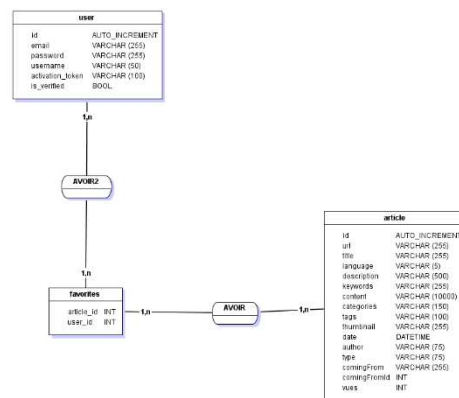
HTML et CSS pour ajouter du contenu et styliser les pages web.

Également, l'utilisation de FileZilla, un client FTP, pour pouvoir gérer la modification des pages pour la correction de bugs sur le site de la DANE.

Du fait de la pandémie de la COVID-19, l'organisation en termes de présence dans le service a été modifiée. 3 jours de télétravail et 2 jours en présentiel.

D. Réalisation / Organisation

J'ai réalisé le MCD du projet LUNE qui n'en est seulement qu'à son commencement :



Également, le planning de réalisation des tâches avec des timelines pour chaque tâche avec son intitulé, sa description et le résultat attendu.

Période	Mission		
	Intitulé	Nature	Attendus
Le 03/01/2022	Accueil Prise en main du projet	Accueil dans le service Prise de connaissance des éléments existants en rapport avec la mission.	
Du 04/01/2022 au 07/01/2022	Fonctionnalité de favoris	Implémentation d'un système de favoris dans le portail pour les utilisateurs authentifiés	La possibilité pour les utilisateurs de garder des articles en favoris, et de les retrouver facilement
Du 10/01/2022 au 14/01/2022	Filtre : les plus vus / moins vus	Implémentation d'un système de filtres permettant de classer les résultats de recherche selon le nombre de vues	La possibilité pour les utilisateurs de trier les résultats par vues
Du 17/01/2022 au 21/01/2022	Tri par matière Tri par date	Implémentation d'un système de filtres permettant de classer les résultats de recherche selon la date de publication ou par la matière	La possibilité pour les utilisateurs de trier les résultats par date ou par matière
Du 24/01/2022 au 11/02/2022	Correction de bugs	Remise en état de fonctionnement d'une base de données recensant les ressources numériques pour l'inclusion scolaire	Page Fonctionnelle
		Remise en état de fonctionnement de l'API permettant de récupérer les données destinées au moteur de recommandation suite à la mise à jour de PHP.	API Fonctionnel

Mission 1 : Fonctionnalité de favoris d'articles

Avant de commencer à coder, il fallait réfléchir à une relation entre les tables concernant les articles et les utilisateurs. J'ai donc opté pour une relation 'ManyToMany'.

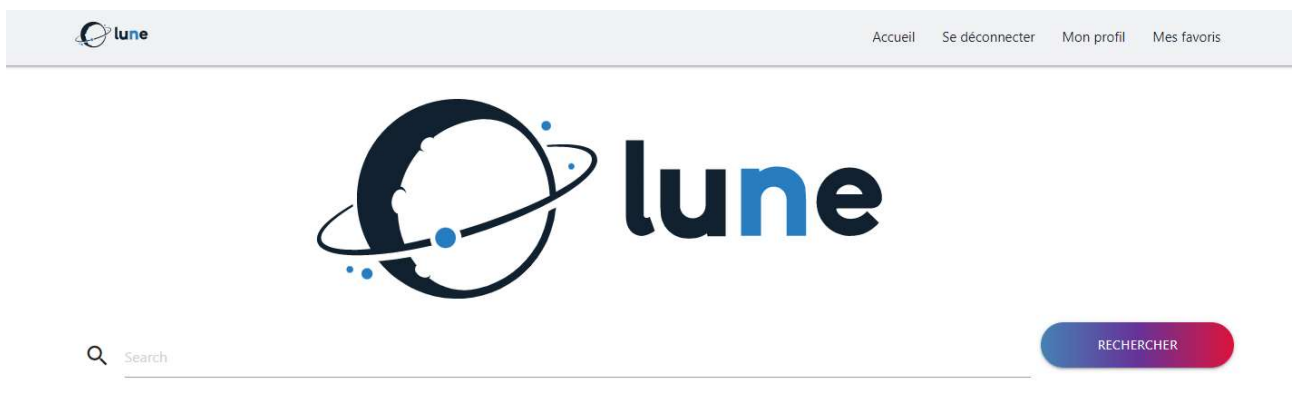
C'est à dire que chaque utilisateur peut avoir plusieurs articles en favoris et que chaque article en favoris peut avoir plusieurs utilisateurs.

J'ai mis en place une section 'mes Favoris' à disposition de l'utilisateur accessible depuis la barre de navigation du site pour qu'il puisse accéder à ses articles favoris.

```
/**
 * @Route("/favorite", name="favorite")
 */
public function index(): Response
{
    $myFavorites = $this->getUser()->getFavorites();

    return $this->render('favorite/index.html.twig', [
        'controller_name' => 'FavoriteController',
        'myFavorites' => $myFavorites
    ]);
}
```

Résultat :



Favoris



Eduthèque Maths
cycle 2 :
Dénombrement de 0
à 9

20-06-2018



Edumédia SVT cycle 3
: utilisation de vidéos
interactives

18-05-2020



Genially, SVT, 5ème :
réaliser un escape
game à distance

05-12-2021



Recherche en e-
éducation - Ressource
Lilemø

13-12-2021

Ensuite, j'ai commencé la partie développement de cette fonctionnalité, j'ai dû trouver un moyen de vérifier pour chaque article qui s'affiche en fonction de la recherche effectuée, si l'utilisateur connecté en possède en favoris.

```
var verif = [];

elem["favorites"].forEach((articleFavorisUser) => {

  if({{ app.user.id }} == articleFavorisUser[o2+"id"])
  {
    verif.push(1);
  }
  else
  {
    verif.push(0);
  }
})

if(verif.includes(1))
{
  string += `<a class="red-text" data-x="1-{{ elem[o+'id'] }}"><i class="material-icons">favorite</i></a>`; // Pour retirer
}
else
{
  string += `<a class="red-text" data-x="0-{{ elem[o+'id'] }}"><i class="material-icons">favorite_border</i></a>`; // Pour ajouter
}
```

Résultat :

Sélectionner un filtre



Lycée 4.0, LibMol, 1ère SVT : Comparer des molécules avec LibMol

28-11-2021

👁 4



Lycée 4.0, Anagène, 1ère SVT : Anagène en ligne

20-11-2021

👁 29



Genially, SVT, 5ème : réaliser un escape game à distance

05-12-2021

👁 18



Classe inversée en SVT avec l'ENT

30-05-2020

👁 0



Puis, je me suis occupé de l'ajout et de la suppression d'articles en favoris en récupérant l'évènement de click de l'utilisateur pour vérifier s'il fallait ajouter ou supprimer l'article de ses favoris.

```

$('.red-text').each(function() {
    $(this).on('click', function () {

        var element = $(this);
        var splitting = element.data('x').split('-');
        var leI = element.find("i").eq(0);

        var numberAddOrRemove = splitting[0];
        var idArticle = splitting[1];

        if(numberAddOrRemove == 0)
        {
            $.ajax({
                url:         '/addFavorite',
                type:         'GET',
                dataType:     'text',
                async:        true,
                data: {idArt: idArticle},

                success: function(data, status) {

                    const x = 1;
                    var p = x+"-"+idArticle;

                    element.data('x', p);
                    leI.html("favorite");

                },
                error : function(xhr, textStatus, errorThrown) {
                    showErrorMessage();
                }
            });
        }
    });
});

```



```

    }
    else if(numberAddOrRemove == 1)
    {
        $.ajax({
            url:        '/removeFavorite',
            type:        'GET',
            dataType:    'text',
            async:        true,
            data: {idArt : idArticle},

            success: function(data, status) {

                const x = 0;
                var p = x+"-"+idArticle;

                element.data('x', p);
                leI.html("favorite_border");

            },
            error : function(xhr, textStatus, errorThrown) {
                showErrorMessage();
            }
        });
    }
});
});
})

```

Pour finir, j'ai créé dans le controller qui permet la gestion du système de favoris, les deux méthodes qui vont permettre de supprimer ou d'ajouter un article à ses favoris.

```

public function addFavorite(Request $req): Response
{
    if($req->isXmlHttpRequest())
    {
        $idArticle = $req->query->get('idArt');

        $entityManager = $this->getDoctrine()->getManager();
        $article = $entityManager->getRepository(Article::class)->find($idArticle)->addFavorite($this->getUser());
        $entityManager->persist($article);
        $entityManager->flush();
        return new Response("Ajouter");
    }
}

```

```

public function removeFavorite(Request $req): Response
{
    if($req->isXmlHttpRequest())
    {
        $idArticle = $req->query->get('idArt');

        $entityManager = $this->getDoctrine()->getManager();
        $article = $entityManager->getRepository(Article::class)->find($idArticle)->removeFavorite($this->getUser());
        $entityManager->persist($article);
        $entityManager->flush();
        return new Response("Supprimer");
    }
}

```

Mission 2 : Système de filtre d'articles 'plus vus/moins vus'

J'ai commencé par créer une liste déroulante avec comme option « Plus vus » et « Moins vus ».

```

<select id="sorts">
    <option value="0">Sélectionner un filtre</option>
    <option value="1">Les moins vus</option>
    <option value="2">Les plus vus</option>
    <option value="3">Les plus récents</option>
    <option value="4">Les moins récents</option>
</select>

```

Puis, j'ai récupéré la valeur de tri de la liste déroulante grâce à l'événement 'onchange' qui va permettre de définir un tri à réaliser sur la recherche d'articles en appelant la fonction `loadNewArticles()`. C'est une requête AJAX qui permet la communication avec le serveur à l'aide de code JavaScript en arrière-plan pendant que la page est affichée à l'écran.

```

$('#sorts').on('change', function()
{
    $('#articlesContainer').empty();
    var sort2 = $(this).val();
    var offset2 = 0;
    loadNewArticles(sort2,offset2);
});

```

```

function loadNewArticles(sort,offset){

    $('#articlesContainer').append(`
    <div class="center-align" id="loading" style="margin-bottom: 1.5em">
        <div class="preloader-wrapper active">
            <div class="spinner-layer spinner-blue-only">
                <div class="circle-clipper left">
                    <div class="circle"></div>
                </div><div class="gap-patch">
                    <div class="circle"></div>
                </div><div class="circle-clipper right">
                    <div class="circle"></div>
                </div>
            </div>
        </div>
    </div>
    `)

    $.ajax({
        url:         '/search/ajaxSearch',
        type:         'POST',
        dataType:     'json',
        async:         true,
        data:          {q:'{{ search }}', off:offset, sorting:sort},

        success: function(data, status) {

            showArticles(data,sort,offset);

        },
        error : function(xhr, textStatus, errorThrown) {
            showErrorMessage();
        }
    });
}

```

Cette requête AJAX va appeler la méthode 'ajaxSearch' dans le controller qui gère le processus de recherche des articles en fonction de la recherche de l'utilisateur et de réaliser le bon tri grâce à la valeur du tri passé en paramètre dans cette requête AJAX.

```

/**
 * @Route("/search/ajaxSearch", name="searchAjax")
 */
public function ajaxSearchArticles(Request $request) {

    if($request->isXmlHttpRequest())
    {

        $articles = $this->searchArticles($request->get('q'));

        $off = $request->get('off');

        $sort = $request->get('sorting');

        if($off == null) $off = 0;

        $finalArticles = [];

        switch($sort) {
            case 1:
                usort($articles['articles'], [$this, 'compLessViews']);
                break;
            case 2:
                usort($articles['articles'], [$this, 'compMoreViews']);
                break;
            case 3:
                usort($articles['articles'], [$this, 'compMoreRecent']);
                break;
            case 4:
                usort($articles['articles'], [$this, 'compLessRecent']);
                break;
            case 5:
                $articles = $this->sortSubject($articles);
                usort($articles['articles'], [$this, 'compSubjects']);
                break;
        }
    }
}

```

```

public function compLessViews($a, $b){

    return $a["article"]->getVues() - $b["article"]->getVues();

}

public function compMoreViews($a, $b){

    return $b["article"]->getVues() - $a["article"]->getVues();

}

```

```

for($i = $off; $i < count($articles['articles']) && $i < $off + 20; $i++)
{
    $finalArticles['articles'][] = $articles['articles'][$i]['article']->toArray();

    if(isset($articles['articles'][$i]['matiere']))
    {
        $finalArticles['articles'][count($finalArticles['articles']) - 1]['matiere'] = $articles['articles'][$i]['matiere'];
    }
}

if(count($articles['articles']) == 0) $finalArticles['articles'] = [];









$finalArticles['info'] = ['articlesLeft' => (count($articles['articles']) > $off + 20)];

return new JsonResponse($finalArticles);

```

Enfin, en cas de succès de ma requête AJAX, je récupère les articles triés et je les passe en paramètre dans la fonction 'showArticles()' pour pouvoir les afficher.

Les plus vus

 <p>Formation BRNE sciences cycle 4 04-12-2019 👁 140</p>	 <p>Maskott Sciences 01-11-2021 👁 140</p>	 <p>Ressources numériques éducatives 05-05-2020 👁 92</p>	 <p>Je suis professeur 30-04-2020 👁 61</p>
 <p>Programmation au lycée, 180 ressources 02-05-2020 👁 51</p>	 <p>BRNE Sciences cycle 3 : Comprendre l'évolution d'un écosystème 20-11-2018 👁 51</p>	 <p>Lycée 4.0, Anagène, 1ère SVT : Anagène en ligne 20-11-2021 👁 29</p>	 <p>120 tutoriels vidéos BRNE 12-03-2020 👁 24</p>

Pour terminer cette mission et donner plus de poids à ce système de vues d'articles, j'ai dû traiter un fichier Json de données des traces des utilisateurs qui recense l'action faite par l'utilisateur ainsi que son IP, le temps passé sur la page, le système d'exploitation, l'URL de la page consultée...

Le fichier que j'ai utilisé provient d'un plugin réalisé par un ancien stagiaire pour le site de la DANE.

Grâce à ce fichier, dans le cas de ma mission, je vais pouvoir comptabiliser le nombre de consultation des articles qui ont été visionnés par les utilisateurs en récupérant les URL.

```
{
  "id": "186546",
  "userID": "afd2b77dd46fb84fafb55a365d060f56",
  "ON_url": "https:\\\\dane.ac-nancy-metz.fr\\brne-geographie-5e-les-risques-technologiques\\",
  "user_ip": "121.200.245.50, 192.168.12.186",
  "action": "timeSpent",
  "id_session": "23a3fabcd43182ad2b8abfd10d492115",
  "data": {
    "comingFrom": "https:\\\\www.google.com\\",
    "timeSpent": "5.183",
    "os": "Windows 10",
    "language": "en-US",
    "browser": "Mozilla\\5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\\537.36 (KHTML, like Gecko) Chrome\\96.0.4664.45 Safari\\537.36"
  },
  "date": "2021-12-03 01:54:24"
}
```

```
{
  "id": "186915",
  "userID": "f7ed75c46579facf6d155d15a0ee392f",
  "ON_url": "https:\\\\dane.ac-nancy-metz.fr\\recherche-e-education-projet-metal\\",
  "user_ip": "185.24.186.4, 192.168.12.186",
  "action": "copy",
  "id_session": "9021ede1a331cc04ef7977007d60bc97",
  "data": {
    "text": "subventionn\\u00e9 par l\\u2019appel \\u00e0 projets e-FRAN",
    "os": "Windows 10",
    "language": "fr",
    "browser": "Mozilla\\5.0 (Windows NT 10.0; Win64; x64) AppleWebKit\\537.36 (KHTML, like Gecko) Chrome\\94.0.4606.81 Safari\\537.36"
  },
  "date": "2021-12-03 11:04:59"
}
```

J'ai créé un script PHP qui va commencer par récupérer les URL des articles de la base de données du portail Lune.

```
try
{
    $db = new PDO ('mysql:host=localhost;dbname=portailune', 'root','');
    echo "connexion reussie";
}
catch(PDOException $e)
{
    print "Erreur : " . $e->getMessage() . "<br>";
}

// Recuperation des URL des articles de la BDD

$req = $db->prepare("SELECT `url` FROM article");
$req->execute();
$urlsBDD = $req->fetchAll(PDO::FETCH_OBJ);
$urlsBddSimplify = [];

for($i = 0; $i < count($urlsBDD); $i++)
{
    array_push($urlsBddSimplify, $urlsBDD[$i]->url);
}
```

Ensuite, j'ai procédé à une extraction et à la récupération des URL des articles visionnés par les utilisateurs.

```
// Extraction et recuperation des URL visionnees par les utilisateurs  
// dans le fichier JSON
```

```
$jsonData = file_get_contents("tracesBackupComplete.json");  
$jsonDecodeData = json_decode($jsonData, true);
```

```
$test = [  
    'url' => [  
        ]  
];
```

```
foreach($jsonDecodeData as $url)  
{  
    array_push($test['url'], $url['ON_url_']);  
}
```

Puis, j'ai réalisé un tri des URL qui ne se trouvent pas dans la base de données pour gagner du temps sur l'exécution du script. Ce qui va permettre de ne pas réaliser de test sur des URL qui ne seraient pas dans la base de données.

```
// Suppression des URL du tableau des URL du fichier JSON qui ne sont pas dans la BDD
```

```
foreach($test['url'] as $key => $url2)  
{  
    if(!in_array($url2, $urlsBddSimplify))  
    {  
        unset($test['url'][$key]);  
    }  
}
```


Pour finir l'étape de traitement, il a fallu compter les URL identiques et les rassembler.

```
$vuesArticles = [];  
foreach($test['url'] as $key => $value)  
{  
    if(!isset($vuesArticles[$value]))  
    {  
        $vuesArticles[$value] = 0;  
    }  
    $vuesArticles[$value]++;  
}  
var_dump($vuesArticles);
```

```
array (size=483)  
'https://dane.ac-nancy-metz.fr/brne-geographie-5e-les-risques-technologiques/' => int 2  
'https://dane.ac-nancy-metz.fr/brne-sciences-cycle-3-comprendre-lorganisation-du-systeme-solaire/' => int 12  
'https://dane.ac-nancy-metz.fr/ent-quizniere-espagnol-lycee-handicap-et-inclusion/' => int 132  
'https://dane.ac-nancy-metz.fr/je-suis-referent-numerique-2/' => int 40  
'https://dane.ac-nancy-metz.fr/ressources-numeriques-educatives/' => int 92  
'https://dane.ac-nancy-metz.fr/metropolidane-2/' => int 5  
'https://dane.ac-nancy-metz.fr/ent-allemand-cycle-4-production-ecrite-vetements/' => int 17  
'https://dane.ac-nancy-metz.fr/brne-anglais-cycle-4-noel-royaume-uni/' => int 469  
'https://dane.ac-nancy-metz.fr/brne-svt-cycle-4-une-dissection-pas-a-pas-du-nerf-sciatique/' => int 21  
'https://dane.ac-nancy-metz.fr/cycle-4-thymio-decouvrir-les-comportements-du-robot/' => int 4  
'https://dane.ac-nancy-metz.fr/brne-espagnol-4e-artisanat-mexicain-los-alebrijes/' => int 17  
'https://dane.ac-nancy-metz.fr/brne-maths-cycle-4-utiliser-exerciseurs/' => int 8  
'https://dane.ac-nancy-metz.fr/tutoriels-video-brne/' => int 24  
'https://dane.ac-nancy-metz.fr/brne-svt-cycle-4-antibiogrammes/' => int 21  
'https://dane.ac-nancy-metz.fr/je-suis-professeur-en-espagnol/' => int 123  
'https://dane.ac-nancy-metz.fr/ent-deploiement-dans-le-premier-degre/' => int 6  
'https://dane.ac-nancy-metz.fr/gar-gestionnaire-acces-ressources/' => int 12  
'https://dane.ac-nancy-metz.fr/brne-decouvrir-un-reseau-informatique/' => int 9  
'https://dane.ac-nancy-metz.fr/edutheque-histoire-cycle-3-francois-ier-protecteur-des-arts-lettres/' => int 12  
'https://dane.ac-nancy-metz.fr/kits-escape-games-interlangues/' => int 31  
'https://dane.ac-nancy-metz.fr/ent-moodle-differentiation-restriction-acces/' => int 6  
'https://dane.ac-nancy-metz.fr/tutoriels/' => int 13  
'https://dane.ac-nancy-metz.fr/college-lab-amenagement-des-espaces/' => int 8  
'https://dane.ac-nancy-metz.fr/brne-anglais-cycle-4-evaluer-comprehension-orale/' => int 7  
'https://dane.ac-nancy-metz.fr/ent-histoire-de-ruby-bridges/' => int 15
```

Enfin, j'ai parcouru les articles de ma base de données et j'ai ajouté les vues correspondant aux articles.

```
foreach($vuesArticles as $key => $vuesURL)  
{  
    $req2 = $db->prepare("UPDATE article SET `vues` = $vuesURL WHERE `url` = :url");  
    $req2->bindParam(':url',$key,PDO::PARAM_STR);  
    $req2->execute();  
}
```


Mission 3 : Système de filtre d'articles 'plus récents/moins récents'

Ce système de filtre d'articles est équivalent au système de tri pour les vues.


Mais au lieu de traiter les vues, on traite les dates de publications des articles.

Q test RECHERCHER


Les moins récents ▼




TRAvais Académiques Mutualisés 2018-19
09-07-2018
0 0
♥




ENT : Mon Bureau Numérique
10-07-2018
86
♥




Recherche avec l'université
11-07-2018
0
♥




Préparer sa rentrée avec les BRNE et Eduthèque
23-07-2018
0
♥




Scratch en cycle 4 pour raconter une histoire
08-08-2018
6
♥



BRNE SVT Cycle 4 : apprendre à commenter un graphique
01-09-2018
3
♥



BRNE SVT cycle 4 : reproduction sexuée des plantes à fleurs
01-09-2018
4
♥



BRNE SVT cycle 4 : fiche sur l'utilisation d'un microscope
01-09-2018
14
♥

Mission 4 : Système de filtre d'articles par matière

Pour réaliser ce système de tri, j'ai commencé par remplir un tableau avec le maximum de matières. Ainsi que deux tableaux qui permettront de faciliter le traitement des valeurs des champs. Ces différents tableaux m'aideront à attribuer une matière aux articles recherchés.

/// A REVOIR CE PARAGRAPHE ///

```
$matieres = ['mathematiques','anglais','francais','histoire','geographie','histoire-geographie','sciences de la vie et de la terre','technologie','lettres','education musicale','education physique et sportive','musique','design','education civique','histoire des arts','albanais','allemand','amharique','ressources','arabe','armenien','arts plastiques','peinture','sculpture','dessin','architecture','photographie','bambara','basque','berbere','breton','bulgare','cambodgien','programmation','catalan','chimie','chinois','cinema','coreen','corse','creole','danois','danse','espagnol','finnois','grec','hindi','hongrois','indonesien','informatique','islandais','italien','japonais','laotien','latin','macedonien','malaisien','malgache','neerlandais','norvegien','persan','peul','philosophie','sciences physiques','polonais','portugais','roumain','russe','serbe','ses','slovaque','suedois','tahitien','theatre','turc','agroalimentaire','arts appliques','genie civil','biotechnologies','coiffure','commerce','comptabilite','conception','electronique','electrotechnique','hotellerie','restauration','tourisme','industries graphiques','imprimerie','mecanique','plasturgie','metallurgie','secretariat','sport','chaudronnerie','ferrotherie','soudure','transport','tpe','ppcp','idd','tpe','audiovisuel','litterature','poesie','archives','communication','sciences de l'information','economie','gestion','psychologie','sociologie','urbanisme','paramedical','social','production mecanique','economie-gestion','recherche','actus'];
```

```
$matieresAcronyme = ['svt','eps','maths','hda','ses','sms','pac','tpe','objets programmables'];
$matieresAcronymeReplace = ['sciences de la vie et de la terre','education physique et sportive','mathematiques','histoire des arts',
'sciences economiques et sociales','sciences medico-sociales','projet artistique et culturel','travaux personnels encadres','programmation'];
```

```
$search = array('À', 'Á', 'Â', 'Ã', 'Ä', 'Å', 'Ç', 'È', 'É', 'Ê', 'Ë', 'Ì', 'Í', 'Î',
'Ï', 'Ò', 'Ó', 'Ô', 'Õ', 'Ö', 'Ù', 'Ú', 'Û', 'Ü', 'Ý', 'à', 'á', 'â', 'ã', 'ä', 'å', 'ç', 'è',
'é', 'ê', 'ë', 'ì', 'í', 'î', 'ï', 'ð', 'ó', 'ô', 'õ', 'ö', 'ù', 'ú', 'û', 'ü', 'ý', 'ÿ', ',', '!', '...', '.');
$replace = array('A', 'A', 'A', 'A', 'A', 'A', 'C', 'E', 'E', 'E', 'E', 'I', 'I', 'I',
'I', 'O', 'O', 'O', 'O', 'O', 'U', 'U', 'U', 'U', 'Y', 'a', 'a', 'a', 'a', 'a', 'a', 'c', 'e',
'e', 'e', 'e', 'i', 'i', 'i', 'i', 'o', 'o', 'o', 'o', 'o', 'o', 'u', 'u', 'u', 'u', 'y', 'y', ',', '!', '...', '.');
```

Pour rechercher une matière qui correspondrait le plus possible à chaque article, j'ai effectué un traitement sur trois champs de la base de données qui seraient les plus aptes à trouver une matière : catégories, mots-clés et son titre.

Chaque article ne passe pas forcément sur les trois traitements, par exemple si un article à trouver une matière dès le premier traitement grâce au champ catégories. L'algorithme passe à l'article suivant.

```
// Premier traitement sur le champ : catégories

foreach($articles['articles'] as $key => $article){

    $categoryArticle = str_replace($search,$replace,$article['article']->getCategories());
    $categoryArticleLower = mb_strtolower($categoryArticle, 'UTF-8');
    $categoryArticleExplode = explode("|", $categoryArticleLower);

    $articles['articles'][$key]['matiere'] = null;
    for($i = 0; $i < count($categoryArticleExplode); $i++)
    {
        if(in_array($categoryArticleExplode[$i],$matieresAcronyme))
        {
            $categoryArticleModified = str_replace($matieresAcronyme,$matieresAcronymeReplace,$categoryArticleExplode[$i]);
            $key2 = array_search($categoryArticleModified, $matieres);
        }
        else
        {
            $key2 = array_search($categoryArticleExplode[$i], $matieres);
        }

        if($key2 !== false)
        {
            $matiere = $matieres[$key2];
            $matiere = ucfirst($matiere);
            $articles['articles'][$key]['matiere'] = $matiere;
            break;
        }
    }
}
}
```



```
// Deuxième traitement sur le champs : mots-clés
```

```
foreach($articles['articles'] as $key => $article)
{
    if($articles['articles'][$key]['matiere'] === null)
    {
        $keywordsArticle = str_replace($search,$replace,$article['article']->getKeywords());
        $keywordsArticleLower = mb_strtolower($keywordsArticle, 'UTF-8');
        $keywordsArticleExplode = explode(" ", $keywordsArticleLower);

        for($j = 0; $j < count($keywordsArticleExplode); $j++)
        {
            if(in_array($keywordsArticleExplode[$j],$matieresAcronyme))
            {
                $keywordsArticleModified = str_replace($matieresAcronyme,$matieresAcronymeReplace,$keywordsArticleExplode[$j]);
                $key2 = array_search($keywordsArticleModified, $matieres);
            }
            else
            {
                $key2 = array_search($keywordsArticleExplode[$j], $matieres);
            }

            if($key2 !== false)
            {
                $matiere = $matieres[$key2];
                $matiere = ucfirst($matiere);
                $articles['articles'][$key]['matiere'] = $matiere;
                break;
            }
        }
    }
}
```

```
// Troisième traitement sur le champs : titre
```

```
foreach($articles['articles'] as $key => $article)
{
    if($articles['articles'][$key]['matiere'] === null)
    {
        $titleArticle = str_replace($search,$replace,$article['article']->getTitle());
        $titleArticleLower = mb_strtolower($titleArticle, 'UTF-8');
        $titleArticleExplode = explode(" ", $titleArticleLower);

        for($k = 0; $k < count($titleArticleExplode); $k++)
        {
            if(in_array($titleArticleExplode[$k],$matieresAcronyme))
            {
                $titleArticleModified = str_replace($matieresAcronyme,$matieresAcronymeReplace,$titleArticleExplode[$k]);
                $key2 = array_search($titleArticleModified, $matieres);
            }
            else
            {
                $key2 = array_search($titleArticleExplode[$k], $matieres);
            }

            if($key2 !== false)
            {
                $matiere = $matieres[$key2];
                $matiere = ucfirst($matiere);
                $articles['articles'][$key]['matiere'] = $matiere;
                break;
            }
        }
    }
}
```

Dans le cas où aucune matière n'a pu être attribuée à des articles après ces trois traitements, la matière `Divers` lui est attribué.

```
foreach($articles['articles'] as $key => $article)
{
    if($articles['articles'][$key]['matiere'] === null)
    {
        $articles['articles'][$key]['matiere'] = "Divers";
    }
}
```

Les étapes suivantes sont similaires aux précédents tris que j'ai réalisés.

Allemand



Les musiciens de Brême,
un Escape Game en
Allemand en cycles 2 & 3

07-10-2020

👁 6



Allemand



Powerpoint, Bac Pro :
Enregistrer un
commentaire audio en
classe inversée

19-05-2020

👁 1



Allemand



ENT : évaluer un exposé
oral avec le module
"travail à faire" du CdT

18-05-2020

👁 2



Anglais



Je suis professeur en
anglais

01-01-2020

👁 13



Anglais



Reading Challenge -
Anglais, cycle 4

07-10-2020

👁 0



Anglais



BRNE Anglais cycle 4 :
proposer une remédiation
en grammaire

01-09-2018

👁 11



Arts plastiques



Lycées 4.0 Galerie virtuelle
arts plastiques

04-10-2019

👁 2



Divers



Aménagement de salles
flexibles au lycée Kastler
de Stenay

15-11-2019

👁 5



Mission 5 : Correction de bugs

À la suite d'une montée de version de PHP sur le site de la DANE passant de la version 7.4 à 8.0.15, certains bugs sont apparus.

Comme le fonctionnement d'une page qui a été compromis. Il s'agit d'une page permettant de choisir des critères de recherches afin d'adapter au mieux les logiciels et applications à présenter aux élèves à besoins éducatifs particuliers.

Le problème était minime mais gravement endommageant puisque l'accès à cette page était impossible. Il manquait en paramètre de la fonction 'mysqli_close()', la variable de connexion qui certes avec la version 7.4 de PHP fonctionnait sans précision de paramètre mais indispensable avec la version 8+.

```
$link = mysqli_connect("$dbhost","$dbuser","$dbpass","$dbbase");

if (!$link)
{
    echo "&Eacute;chec de la connexion &agrave; la base de donn&eacute;es..." ;
    exit();
}

$requete="SHOW TABLE STATUS";

$resultat = mysqli_query($link,$requete);

while($ligne=mysqli_fetch_row($resultat))
{
    if ($ligne[0]=="applihandi")
    {
        $compte=$ligne[4];
    }
}

mysqli_close($link);
```

Actuellement 56 enregistrements dans la base

Il n'est pas obligatoire de saisir des critères à rechercher. Dans ce cas vous obtiendrez la totalité des logiciels repérés.

Si vous cochez des cases, le moteur cherchera les critères en mode "ET" ; plus vous sélectionnez de critères plus vous restreignez le champ des possibles...

Niveau :

☐ Cycle 1 - ☐ Cycle 2 - ☐ Cycle 3 - ☐ Cycle 4 - ☐ Lycée

Matériel :

☒ Ordinateur - ☐ Tablette

Système :

☐ Android - ☐ iOS - ☐ macOS - ☐ Linux - ☐ Windows

Nécessite une connexion internet active pour être utilisée:

☒ Sans avis - ☐ Non - ☐ Oui

Aide pour :

☐ Dire - ☐ Lire - ☐ Écrire - ☐ Mémoriser - ☐ Compter - ☐ Calculer - ☐ Reasonner - ☐ Tracer - ☐ Créer - ☐ Adapter

Réinitialiser les champs de la recherche

Valider la demande de consultation

Un nouveau bug est apparu, le plugin qui permet de récupérer les traces des utilisateurs lors de leur passage sur le site de la DANE n'était plus capable de récupérer les fichiers contenant les traces à l'aide d'une clé API.

Pour corriger ce bug, j'ai dû le traiter en deux problèmes.

Le premier problème était la visibilité des méthodes : 'apiGetTracesInfo()' qui permet de d'afficher le nombre de fichiers de traces ainsi que le nom des fichiers.

Ainsi que la méthode 'apiGetTraces()' qui sert à afficher le contenu d'un fichier grâce à une clé API et en précisant dans l'URL son numéro.

Aucune visibilité n'était précisée pour ces deux méthodes. J'ai donc rajouté une visibilité 'public static' afin d'avoir un accès total à ces méthodes.

```

/**
 * API returns stored traces info from url https://dane.ac-nancy-metz.fr/wp-json/luneAPI/getTraces/<key>
 * @param WP_REST_Request $request
 * @return array|WP_Error
 */
public static function apiGetTracesInfo(WP_REST_Request $request){
    if(!APIController::keyExist($request['key'])) return new WP_Error( 'wrong_key', 'key invalid', array( 'status' => 404 ));

    $docs = DataExporter::getFileName();
    return ['nbTraces' => count($docs), 'docs' => $docs];
}

/**
 * API returns stored traces from url https://dane.ac-nancy-metz.fr/wp-json/luneAPI/getTraces/<key>/<file number>
 * @param WP_REST_Request $request
 * @return false|string|WP_Error
 */
public static function apiGetTraces(WP_REST_Request $request){
    if(!APIController::keyExist($request['key'])) return new WP_Error( 'wrong_key', 'key invalid', array( 'status' => 404 ));

    // $dir = CD_PLUGIN_PATH . '/traces_backup';
    $dir = TRACE_KEY_PATH;

    if(!is_numeric($request['doc'])) return new WP_Error( 'wrong_doc', 'doc invalid', array( 'status' => 404 ));
    $i = (int)$request['doc'];
    $docs = DataExporter::getFileName();

    if($i < 0 or $i >= count($docs)) return new WP_Error( 'wrong_doc', 'doc invalid', array( 'status' => 404 ));

    $file = $dir. '/' . $docs[$request['doc']];

    return file_get_contents($file);
}

```

Le deuxième problème était le chemin d'accès vers le dossier contenant les fichiers de traces.

De ce fait, j'ai créé une variable globale dans le fichier index du site ayant pour valeur le chemin vers le dossier.

```
define( 'TRACE_KEY_PATH', "/web/dane/wp-content/uploads/lune");
```

Puis, j'ai simplement remplacé les valeurs des variables contenant le chemin vers le dossier contenant les traces dans les méthodes effectuant des actions pour les récupérer et les afficher.

```

public static function showSavedTracesData(){
    // $dir = CD_PLUGIN_PATH . '/traces_backup';
    $dir = TRACE_KEY_PATH;

```



```
/**
 * Return the name of all traces file name
 * @return array
 */
public static function getFilesName(){
    // $dir = CD_PLUGIN_PATH . '/traces_backup';
    $dir = TRACE_KEY_PATH;
```

E. Conclusion

Le projet Lune n'est pas encore terminé, de nombreuses fonctionnalités sont encore à implémenter ainsi que le développement du moteur de recommandations.

Pour conclure, ce stage m'a apporté une expérience professionnelle supplémentaire par rapport à mon futur métier de développeur. J'ai pu appliquer lors de ce stage les compétences que j'ai apprises lors de ma formation de BTS et notamment approfondis mes compétences en termes de langage Front-end tel que JavaScript.

Je remercie le Recteur de l'académie, Monsieur HUART Jean-Marc et toute l'équipe de la DANE et tout particulièrement mon tuteur, Gilles SCHAECK, qui m'ont accompagné tout au long de ce stage.