Optimisation, fonctionnalités,…

Fonctionnalités

Recherche sur le site = recherche dans l'information en base de données

- **Formulaire de recherche en HTML**

```
<form method="GET" action="index.php">
    <input type="text" name="search" placeholder="Rechercher un article...">
    <button type="submit">Rechercher</button>
</form>
```

- **Traitement**

```
<?php
    …
    if (isset($_GET['search'])) {
        $search = htmlspecialchars($_GET['search']);
        $query = $pdo->prepare("SELECT * FROM articles WHERE titre LIKE :search OR contenu LIKE :search ORDER BY
        date_publication DESC");
        $query->execute([‘search' => "%$search%"]);
    …
?>
```

→ Vérifie si une recherche est lancée

→ Protection contre les injections SQL

→ % caractère joker qui remplace un texte utilisé avec LIKE

Amélioration des performances

Compression, optimisation,…

- Amélioration de l'**expérience utilisateur** (UX)
- Meilleur **référencement** (Google pénalise les sites lents)
- Optimisation des **coûts serveur** (Moins de requêtes et moins de charge)
- **KPI**
  - FCP (First Contentful Paint): Temps avant l'affichage du premier élément
  - TTI (Time to Interactive): Temps avant que la page devienne interactive
  - LCP (Largest Contentful Paint): Temps d'affichage du plus grand élément visible
  - …
- Ex: Google Lighthouse (Chrome > Outils de développement > Lighthouse)

  Fournit un **rapport complet** sur les **performances** de la page en cours

Amélioration des performances

Frontend

- **Minification** de fichier css et js (enlever les espaces, commentaires,…)
- **Compresser** les images PNG/JPG remplacé par WebP, AVIF
- Lazy loading (ne **charger que les images visibles**)

  <img src="face.webp" alt="Figure de l'utilisateur optimisé" **loading="lazy"**>
- CDN (Content Distribution Network) Semblable à un **proxy**
- Optimisation des polices (**Eviter le chargement inutile des variantes**)

  Préférer les formats optimisés (woff2,…), hébergement local, polices systèmes,…
- Rem: font-display: swap; **Affiche le texte immédiatement** avec une police de secours, puis la remplace par la police demandée dés son chargement

Amélioration des performances

Backend

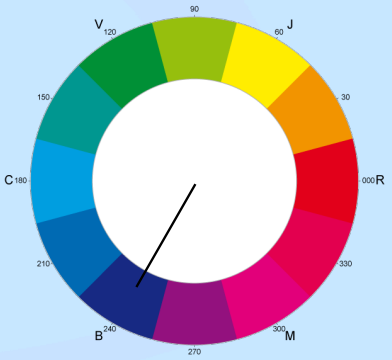
- **Cache navigateur:** Cache-Control: public, max-age=31536000

  Le navigateur garde en cache les fichiers pendant un an
- Utiliser **HTTP/2** ou **HTTP/3**: Charge plusieurs fichier en une seule connexion

  HTTP/3 utilise QUIC, un protocole plus rapide que TCP
- Activer **Keep-Alive**: Evite d'ouvrir une nouvelle connexion à chaque requête
- Optimisation DB: Indexation, pagination, requête préparées,…
- …

Les couleurs

Expérience utilisateur



Cercle chromatique

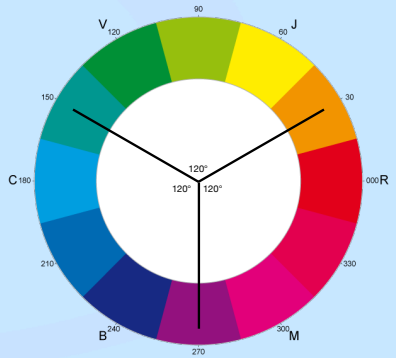
Ex: Teinte Bleu pur

- **Primaires: Rouge vert et bleu** ne peuvent pas être obtenu en mélangeant d'autres couleurs
- **Secondaires: Cyan, magenta et jaune** obtenu en mélangeant 2 couleurs primaires
- **Tertiaires:** Issue du mélange d'une couleur primaire et d'une couleur secondaire
- **Modèles chromatique:**
  - **RGB:** Utilisé pour les écrans, chaque couleur = combinaison de rouge, vert et bleu (**HEX**: Représentation hexadécimale du RGB)
  - **HSL:** Teinte (Hue), Saturation et Luminosité

Bleu pur	R=0, G=0, B=255 HEX: #0000FF H=240°, S=100%(=pur), L=50%

Les couleurs

Expérience utilisateur



Cercle chromatique

- **Fort contraste:** 2 couleurs opposées sur le cercle chromatique
- **Douceur:** 3 couleurs analogues, proches sur le cercle chromatique
- **Triadique:** Equilibre entre contraste et harmonie; designs dynamiques et équilibrés. Couleurs équidistantes sur le cercle chromatique (120°)

1 couleur principale = dominante: fond ou éléments principaux

= identité de la marque

2 couleurs = couleurs d'accentuation: boutons, éléments interactifs, accents,…

Ex: Orange, vert, violet

Outil de choix: Adobe Color, Coloors,….

Contraste et accessibilité

Expérience utilisateur, WCAG/ARIA

- **Ratio de contraste des couleurs:** recommandé est 4.5:1 pour le texte normal, 3:1 pour les textes en gras de grande taille
- **Outil de test et ratio:** webAIM Contrast Checker
- **Attributs ARIA (Accessible Rich Internet Applications).**
  - ajout des rôles pour identifier les éléments interactifs
    - <button **role="button"**>Cliquez-moi</button>
  - ajout d'états et propriété dynamiques
    - <button **aria-expanded="false"**>Menu</button> La liseuses dira « Bouton menu non développé »
  - ajout description accessible
    - <input type="text" **aria-label="Nom"**> La liseuses dira « Champ de saisie Nom »
  - Masquer aux lecteurs
    - <div **aria-hidden="true"**>Texte décoratif La liseuses ne dira rien

Préparation à la production

Testing

Test unitaire, …

- **En php pour tester**
  - **var_dump():** Affiche la valeur et le type d'une variable
  - **print_r():** Affiche une variable (plus lisible pour les tableaux)
  - **echo:** Affiche simplement une valeur
  - **die() ou exit():** Coupe l'exécution pour voir jusqu'où le script va
  - **isset() / empty():** Vérifie si une variable existe ou est vide
  - **Instructions conditionnelles:** Pour vérifier si le code passe bien dans le bon bloc (if, else, etc.)

```
$tab = ['nom' => 'Adam', 'age' => 30];
```

print_r($tab);

```
Array
(
    [nom] => Adam.
    [age] => 30
)
```

var_dump($tab);

```
array(2) {
    ["nom"]=>
    string(4) "Adam"
    [« age"]=>
    int(30)
}
```

- **XDebug outil permettant de débogguer le code php**
  - Points d'arrêts, exécution pas à pas, valeurs des variables,…
- **PHPUnit outil pour le testing des développement en php**
  - 1 test=1 comportement à vérifier
  - Tester les cas limites: Erreurs, Valeurs nulles,…
  - Utiliser les assertions: assertEquals, asserTrue,…

```
use PHPUnit\Framework\TestCase;

class MathTest extends TestCase {
    public function testAddition() {
        $this->assertEquals(4, 2 + 2);
    }
}
```

Hébergement et mise en ligne

Serveur distant

- **Exemple: https://www.infinityfree.com** GRATUIT
  - **Panneau de gestion - Gestionnaire de fichier/Accès FTP - Website Builder - Script Installer - Sous-Domaine - Base de données - Bande passante - Stockage - SSL - …**