

Technologies web

- **Produire une application web :**

Notre application web est accessible via n'importe quel navigateur, déployée sur Lakartxela et accès au code sur GitHub.

Interfaces utilisateur

- **Interface professionnelle et responsive :**

Pour notre application nous avons choisi d'utiliser la librairie Tailwind CSS, qui reste similaire à Bootstrap mais offre plus de liberté dans les classes CSS.

- **Installation de Tailwind CSS :**

Bien que Tailwind a des similitudes avec Bootstrap, le processus d'installation est différent. A l'aide du gestionnaire de paquet npm, on viens télécharger la bibliothèque Tailwind dans notre projet.

```
npm install -D tailwindcss
npx tailwindcss init
```

Nous avons maintenant un nouveau fichier créer dans notre projet : `tailwind.config.js`
Dans ce fichier, il faut y préciser le chemin de nos templates pour indiquer à Tailwind où chercher des classes à appliquer.

Pour cela il faut modifier la ligne `content` du fichier.

```
/** @type {import('tailwindcss').Config} */ module.exports = {
  content: ["./src/**/*.html", ".js"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Ensuite, il faut créer un fichier `input.css` et un fichier `output.css` (ou un autre nom) dans lequel on viens ajouter les directives pour chaque couches que Tailwind doit couvrir dans le fichier css principal.

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Enfin, on rentre la commande suivante dans un terminal

```
npx tailwindcss -i ./src/input.css -o ./src/output.css --watch
```

Cette commande vas indiquer à Tailwind de prendre connaissance des directives dans le fichier input.css, ensuite va lire tous les fichiers templates indiquer dans les config et va appliquer du code css correspondant dans le fichier output.css.

Le guide d'installation de Tailwind est disponible avec plus de détails sur le [site](#).

- **Responsivité avec Tailwind :**

Comme Bootstrap, Tailwind permet de gérer la responsivité des pages grâce à des classes.

Il suffit de définir des styles de tailles différents en fonction de la taille de l'écran. Comme pour Bootstrap cela respecte tailles "universelles" d'écran avec des clauses pour chacune de ces tailles.

Par exemple `sm` pour petit écran, `md` pour écran "moyen", `lg` pour grand écran.

- **Intégration de composant JavaScript :**

A part un petit code JavaScript pour un carrousel, notre application ne dispose pas de bibliothèque JavaScript.

Si devons installer une bibliothèque sur notre projet nous aurions probablement utiliser un CDN qui consiste à importer directement dans le fichier html, un fichier js en utilisant simplement une balise.

```
<script src="https://cdn.jsdelivr.net/npm/lodash@4.17.21/lodash.min.js">
</script>
```

Pour des cas plus spécifiques, on pourrait également installer les bibliothèque directement sur le projet avec npm.

Intégration d'une base de données

Notre application dispose d'une base de données avec laquelle elle interagit. Il est possible de consulter la base données sur [lakartxela](#) . C'est dans cette base que nous stockons nos événements / actualités. les informations des utilisateurs et autres.

Gestion des données

A l'aide de notre base de données, on est capable d'ajouter du contenus dans la base, de supprimer, de consulter, de lister ou encore d'en modifier le contenu.

Ex à montrer:

- Lister : liste des événements / actualités sur la page d'accueil.
- Consulter : Consulter les informations relatives à son compte dans la page compte (nécessite d'être connecté)
- Ajouter : Créer un nouvel utilisateur en s'inscrivant sur la page d'inscription.
- Modifier : Modifier les informations de son compte dans la page compte (nécessite d'être connecté).
- Supprimer : Supprimer son compte de la base de données dans la page compte (nécessite d'être connecté).

Formulaires interactifs

Notre application dispose de nombreux formulaires interactifs permettant à l'utilisateur diverses choses, comme proposer la création d'un événement (page proposer un événement).

Mettre à jour les informations de son compte (page compte).

S'authentifier grâce à son adresse e-mail (page connexion).

Chacun de ces formulaires ont une première validation côté client en ajoutant des restrictions dans les balises `<input>`

Puis une seconde validation, plus poussée, côté serveur en php avec usage de règles de validations etc.

Enfin, nous sommes capable de gérer l'affichage de messages d'erreurs en fonctions de l'erreur, pour pouvoir informer l'utilisateur sur son erreur et où elle se trouve.

Monter la page de proposition d'événement car les autres formulaires n'ont l'affichage des erreurs.

Gestion des comptes utilisateurs

- **Système d'authentification :**

Notre application dispose d'un système d'authentification sécurisé permettant à l'utilisateur de se connecter à son compte s'il en possède un. Il peut se connecter à l'aide de son adresse e-mail et d'un mot de passe qui est d'abord hasher avant d'être comparé à celui de la base de données.

- **Fonctionnalités supplémentaires : système d'inscription :**

Notre application dispose d'un système d'inscription si l'utilisateur souhaite se créer un nouveau compte pour utiliser pleinement l'application. L'utilisateur peut entrer de manière sécurisée ces données et son mot de passe est hasher avant d'être enregistré en base de données.

Nous avons également prévu une page en cas d'oubli de passe, cependant elle n'est pas fonctionnelle car nous avons préféré nous pencher sur les fonctionnalités principales de l'application.

Sécurité des application

- **Validation des données côté serveur :**

Chaque formulaire de notre application possède une vérification des données côté serveur en particulierité les formulaire qui manipulent des données sensibles comme des mots de passe.

- **Protection contre les failles XSS :**

Nous nous sommes prémunis des failles XSS dans notre application afin d'éviter toute fuite de données ou injections de script malveillants.

- **Protection contre les injections SQL :**

Nous évitons également toutes injections SQL dans nos formulaire à l'aide de requêtes préparées pour éviter qu'un utilisateur malveillant compromette la base de données.

Fichiers de configuration

Pour notre application, nous avons regrouper toute les données de configuration dans un seul fichier YAML, ce dernier contient les informations de connexion à la base de données. Mais également d'autres paramètres globaux nécessaires au fonctionnement de l'application, comme la variable `PREFIX_TABLE` .