Documentation technique CSUNVB

Documentation pour les éventuels nouveaux membres de l'équipe de développement.

A quoi sert le site du CSU ? Qui l'utilise et pourquoi ?

Le site internet actuellement en développement sera utilisé par les ambulanciers du CSU Nord Vaudois et Broye. Il sera utile aux ambulanciers afin de faciliter leurs tâches administratives quotidiennes qui jusqu'à aujourd'hui s'effectuent sur le papier.

Ce site fonctionnera en interne c'est à dire qu'uniquement les membres agréés auront la possibilité d'utiliser le site internet CSUNB. Par membre agréés,s'entend les secouristes.

Il y aura deux grands types d'utilisations du site. D'une part il y aura les utilisateurs simples qui ne peuvent que remplir les rapports et les signer et d'autres part il y aura les administrateurs qui sont les responsables qui créent, préparent et gèrent les rapports et la plateforme en général.

Un utilisateur peut se connecter à toutes les bases mais certaines actions ne sont possibles que si l'utilisateur est connecté à la base auquel il veut modifier des données.

Il y aura également une API permettant a une application modile de consulter et modifier certaines données.

Dans quel contexte (technique) fonctionne ce site ?

Le site sera hébérgé chez swisscenter. Une connexion internet sera donc nécessaire pour accèder au site.

Celui-ci sera accessible et utilisable avec un pc ou une tablette, car le site, a terme, devra être entierement responsive.

Il aura également une API pour communiquer avec une application mobile.

Qu'est-ce que je dois faire pour pouvoir essayer ce site ?

Pour l'instant, une version en développement, régulièrement mise à jour, est disponible à l'adresse [csunvb.mycpnv.ch]. Cependant, il est nécessaire de posséder un identifiant pour s'y connecter. Pour le récupérer, il faut s'adresser au chef de projet, M. Carrel.

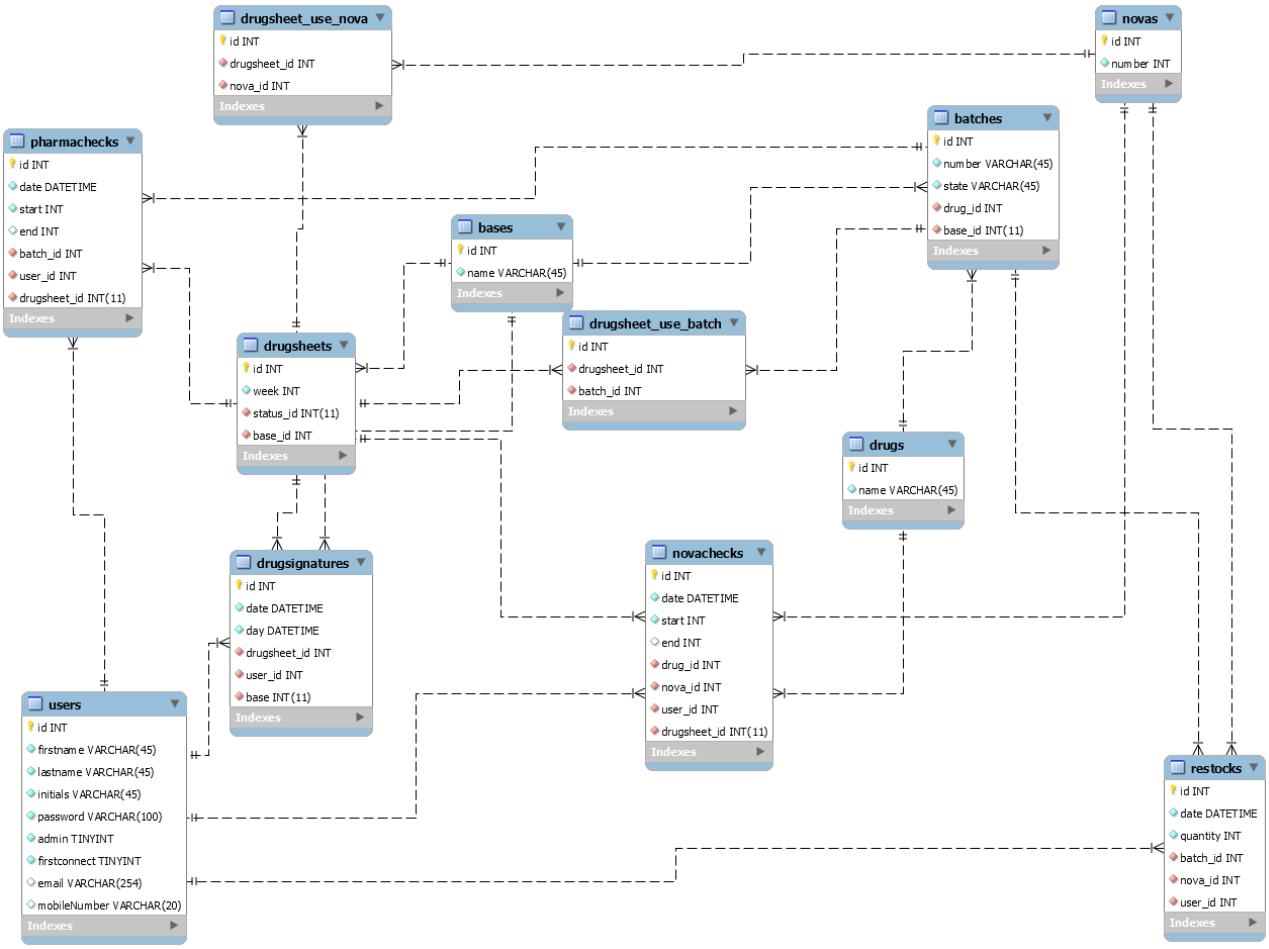
Quelles sont les données / informations que ce site manipule ?

Ce site internet est lié à une base de données qui contient toutes les données nécessaires pour la gestion administrative.

Le site est composé de 4 grandes sections :

- Les taches hebdomadaires : -- Permet la gestion des tâches à effectuer au cours de la semaine
- Les remises de garde : -- Permet la gestion des gardes à bord des ambulances (matériel, équipage, tâches, remarques pour la garde ...)
- Les stups -- Permet la gestion des stocks de médicaments dans les ambulances et à la base

MLD:



- L'administration -- Permet l'administration du site (utilisateurs, stocks, ...)

Données

Les différentes tables et données peuvent être retrouvées dans les MCD des différentes parties (DOC\SQL)

De quels composants le site est-il fait ?

Le site est basé sur la méthode MVC (Model, vue, controleur), pour chaque partie du site (garde, tâches hebdomadaire, ect) un fichier de chaque type est créé dans le dossier associé.

- Il y as un dossier Doc contenant la documentation
- Il y as un dossier public qui contient l'index, le js, le CSS et les assets, ce dossier sera utilisé comme root du serveur web afin que le reste du site ne soit pas accessible en tappant l'url
- A la racine on trouve les fichiers globalhelpers.php (les fonctions communes a toutes les parties), path.php (les chemins), policies.php (les politiques d'accès), pageList (les différentes paramètre $_GET['action'] possible pour chaque page)
- Dans le dossier Vue on retrouve le gabarit et le fichier helpers.php (fonction général d'affichage)
- Le dossier PHPMailer packet utilisé pour envoyer de mail

Quelles technologies est-ce que je dois connaître pour pouvoir développer ce site ?

Les languages PHP et javascripts (utilisant parfois JQuery) sont indispensables pour travailler sur ce projet. Ajax est aussi utilisé afin de réaliser des requêtes asyncrones au serveur.

Il faut aussi être à l'aise avec le html et le css pour tout ce qui est de la mise en forme

Il est aussi nécessaire de connaître mysql car il y aura plusieures requête MYSQL pour interroger la base de donnée.

Le choix de ces languages paraissent évidents pour le développement d'un site internet.

Résumé: -PHP, javascript -MySQL -html, css , bootstrap

Qu'est-ce que je dois installer sur mon poste de travail pour pouvoir commencer à bosser sur ce site ?

Les logiciels suivant sont ceux que nous avaons utiliser pour travailler. Des alternatives sont possible mais attention à la compatibilité.

Il faut posséder :

- Un environnement de développement: PhpStorm 2019.3.x https://www.jetbrains.com/fr-fr/phpstorm/
- PHP version 7.4.x
- Serveur de base de données: MySQL Community Server 8.0.23 https://dev.mysql.com/downloads/mysql/
- Client de base de données: MySQL Workbench (distribué avec MySQL serveur), Heidi SQL v11.2 https://www.heidisql.com/
- une adresse gmail servant pour l'envoi de mail
- drawio pour consulter ou éditer certains fichiers comme les MCD de base de donnée
- balsamiq pour créer ou éditer les maquettes

créer le fichier .const.php dans le dossier model, et y ajouter les bons paramètres en se référrant à .const.example.php dans le même dossier

créer le fichier .mailconf.php dans le dossier PHPMAILER, et y ajouter les bons paramètres en se référrant à .mailconfexemple.php dans le même dossier

désactiver la double authentification sur le compte gmail utilisé pour l'envois de mail et activer l'accès mois sécursié des applications (dans sécurité)

Exécuter les script mysql (du dossier DOC\SQL) dans l'odre suivant structure > data init puis finalement data test pour le développement ou data_prod pour la version sur swisscenter.

Utiliser le dossier public comme dossier root du serveur

Démarrer le serveur

Est-ce qu'on a des conventions de codage ?

La majorité de ce qui est de nature technique est rédigé en anglais: code, commentaires, noms de fonction, de fichiers, de variables, de base de données, de champs, ...

Le formatage du code php suit ce PhP Style Guide

Les fonctions sont précédées d'un bloc de commentaire qui a la forme suivante:

```
/**
* <nomFonction> : à quoi ça sert
* <paramètre1> : qu'est-ce qu'est + type
* <paramètre2> : qu'est-ce qu'est + type
*…
* return : ce que ça renvoie
**/
```

Les nom de fonction appelée depuis le javascript en ajax se termine par _ajax() pour les différencier

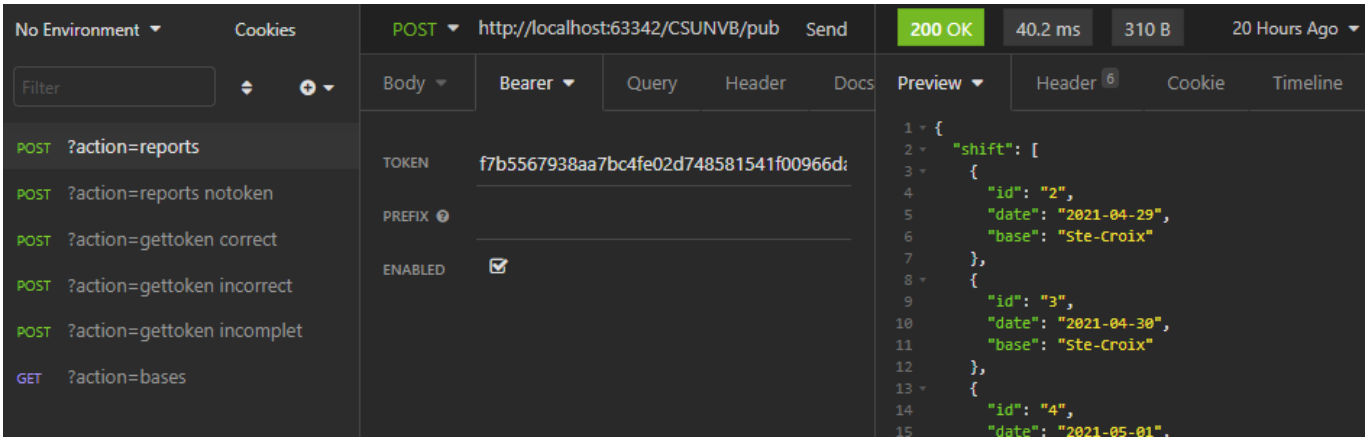
Stratégie de test

Il n'y a pas de tests unitaires implémentés dans le projet.

Pour effecturer les tests nous pouvons demander a plusieurs personnes de tester toute l'application dans plusieurs environnements. Il faut la tester sur plusieurs systèmes comme Windows, linux, mac pour les différentes particularités propres aux différents systèmes.

Nous savons que les utilisateurs vont utiliser tant un ordinateur qu'une tablette. Il faudra donc effectuer les tests avec les deux.

Pour l'API, nous utilisons le logiciel insomnia qui permet de faire des scénarios et tester les diverses fonctionnalités et vérifier les données qu'elle nous retourne.



Points techniques détaillés

Utilisation de l'API

Toutes les actions avec l'API sont décrites ci-dessous.

Les url disponibles pour effectuer des requètes à l'API sont les suviants: `https://[adresse du site]/api/index.php?action=[action]'

Action	Description

Action	Description
gettoken	L'API a un système d'authentification pour obtenir un token afin de pouvoir faire toutes les requêtes en nécéssitant un. Voir chapitre "gettoken"
bases	L'API permet de fournir la liste des bases sans fournir de token. Voir chapitre "bases"
reports	L'api permet d'avoir la liste des rapports de garde et de stupéfiants ou un utilisateur est référencé ou a effectué une action. Voir chapitre 'reports'.

gettoken

L'API a un système d'authentification pour obtenir un token afin de pouvoir faire toutes les requêtes en nécéssitant un.

Un formulaire en POST est requis pour s'authentifier. Les champs doives être 'initials' et 'password' et comme l'indique le nom du champ, le champ 'initials' doit être rempli avec les initials de l'utilisateur.

Si les identifiants sont corrects, l'API retourne un bearer token qui pourra être utilisé ensuite pour les requêtes demandant une authentification.

Les erreurs sont retournées par des erreurs HTTP:

Code d'erreur	Raison
400 - Bad Request	Le nom d'utilisateur ou/et le mot de passe n'ont pas été transmis correctement dans la requete POST
401 - Unauthorized	Utilisateur ou/et mot de passe incorrect
500 - Internal Server Error	Un problème a été rencontré au niveau du serveur web

Les tokens n'ont pas de durée de validité mais sont écrasés et recréés après chaque authentification de l'utilisateur à l'API.

bases

L'API permet de fournir la liste des bases sans fournir de token.

L'API retournera la liste des bases sous forme de JSON formaté de cette façon:

reports

L'api permet d'avoir la liste des rapports de garde et de stupéfiants ou un utilisateur est référencé ou a effectué une action.

L'utilisateur doit s'authentifier à l'aide d'un token bearer. Voir chapitre 'gettoken'.

L'API peut retourner des erreurs HTTP:

Code d'erreur	Raison
400 - Bad Request	Aucun token n'a été transmis ou dans un format invalide
401 - Unauthorized	Token non-valide

L'API retournera la liste des rapports en JSON formatés de cette façon:

```
{
  "shift": [
    {
      "id": "2",
      "date": "2021-04-29",
      "base": "Ste-Croix"
    },
    {
      "id": "3",
      "date": "2021-04-30",
      "base": "Ste-Croix"
    },
    {
      "id": "4",
      "date": "2021-05-01",
      "base": "Ste-Croix"
    },
    {
      "id": "5",
      "date": "2021-05-03",
      "base": "Ste-Croix"
    }
  ],
  "drug": [
    {
      "id": "23",
      "week": "2044",
      "base": "Ste-Croix"
    },
    {
      "id": "42",
      "week": "2111",
      "base": "Ste-Croix"
    }
  ]
}
```


Comment marche le système de routage entre les différentes page du site ?

Le site utilise un modèle MVC et le fichier index.php et toujours appelé et il sert pour le routage, (et aussi à définir les différents paramètres PHP et à appeler les autres fichiers utiles). Le paramètre servant au routage est un GET "action" est utilisé dans toutes les url du site, il définit l'action que l'on cherche à effectuer, par exemple : http://localhost:8080/index.php?action=home si l'utilisateur n'est pas connecté il aura un choix d'action possible restreint par un switch un fonction de cette action, sinon si il est connecté et qu'une fonction d'un des contrôleur possède le même nom que le paramètre GET "action", celle-ci sera appelée, sinon la fonction home() sera appelée par défault

Comment marche le système de policies ?

Avec le système de routage ci-dessus, si un utilisateur connait le nom de l'action, il pourra entrer l'url et effetuer l'action même si son rôle ne devrait pas le lui permettre, C'est pourquoi, il faut vérifier que l'utilisateur a bien les droits d'effectuer cette action :

Le fichier policies.php liste les actions possible et leur niveau de droit nécéssaire

Les actions possible par tout le monde (niveau 0) n'ont pas besoin d'être renseignées

Lorsqu'on tente d'accèder un une page, la fonction ican("nom de l'action") est appelée (dans index.php) afin de savoir si l'utilisateur possède un niveau de privillège suffisant, si ce n'est pas le cas il est redirigé vers la

page d'accueil.

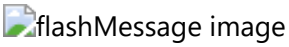
Qu'est-ce que c'est que ce champ 'slug' dans la table 'status' ?

Un slug est un identifiant sous contrôle du code de l'application. Il se situe entre l'id de base de donnée dont on ne peut jamais présumer de la valeur dans le code et la valeur affichée. Exemple: status 'Ouvert', qui a un slug 'open' et un id 2. Si je veux sélectionner les rapports ouverts, je fait un select « where slug='open' » . Si l'id de l'état 'open' est différent dans une autre db => ça marche, si un jour je veux changer le terme visible de « Ouvert » en « Actif » par exemple, je peux le faire en ne changeant que des données.

Voir cette référence (parmi tant d'autres)

Qu'est-ce que c'est que le 'flashmessage' ?

Le flash message est un message en haut de la page utilisé pour indiquer à l'utlisateur si une action s'est effectuée correctement ou si il y a une erreur.

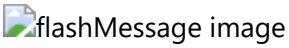


Celui-ci peut être affiché de deux manières différentes :

- En php, avec la fonction : setFlashMessage("mon message") lors du chargemement d'une nouvelle page
- En javascript, avec la fonction : flashMessage("mon message") si la page n'est pas rechargée

Comment marchent les commentaires épinglés ?

Le système d'épinglage permet un commentaire d'un rapport de garde, d'être affiché sur les prochains rapport. Par exemple : "Les piles de la radio sont plates" pour l'indiquer au prochains groupes jusqu'à ce que le problème soit réglé.



Au survol d'un commentaire, un bouton pour épinglé gris est visible, lors que le commentaire est épinglé il sera visible sur les prochains rapports et aura champs champ carryOn à 1 dans la base de donnée. Pour désactiver l'épinglage cliquer à nouveau sur l'icone "épinglé" cela aura pour conséquence de remplir le champs "endOfCarryOn" avec la date du rapport sur lequel la personne se trouve, à partir de ce moment, le message ne sera plus affiché sur les prochains rapports.

La selection des commentaire de la base de donnée à affiché sur le rapport se fait deux manière :

- Si le commentaire est lié au rapport, il est affiché ainsi que l'heure et la personne qui l'a écris
- Si le commentaire n'est pas lié au rapport mais qu'il est sur la même base et que :

  (date du commentaire < date du rapport sélectionné) et que (date du rapport sélectionné < endOfCarryOn du commentaire ou endOfCarryOn est null) Alors celui-ci est affiché en tant que commentaire épinglé et la date de celui-ci est aussi affichée

Comment marchent les pop-up de confirmation ?

En bas du gabari est présent un squelette de pop-up de confirmation, ensuite il peut être affiché est adapté comme on le souhaite dans modal.js grâce aux functions : showModal, setTitleModal, setBodyModal,

addBodyModal, setSubmitModal

Comment est clôturé un rapport ?

Actuellement le principe n'est pas encore appliqué rapports de stupéfiants

Pour les parties tâches hebdomadaire et garde :

Lors du clic sur le boutons fermer, une requête AJAX est envoyée au serveur qui retourne le nombre de tâches manquante

- Confirmation simple si le rapport est complet
- Confirmation avec avertissement si le rapport n'est pas complet

Pour les rapports clôturés

- Les tâches non-validées sont affichée en rouge
- Une icône /!\ indique l'erreur dans la liste des rapport, au survol de l'icone est affiché le nombre de tâches non effectuées

De plus les initiales de la personnes ayant clôturé un rapport seront affichées en haut de celui-ci

Navbar

La Navbar est présente sur toutes les pages tant que l'utilisateur est connecté

Pour faire correspondre l'action (paramètre GET) à l'onglet de la Navbar, le fichier pageList.php est utilisé Il contient une liste des actions possible pour chaque onglet de la navbar comme par exemple : $shiftPages = array("shiftList","shiftShow","shiftLog");

Pour chaque onglet de la navbar si le paramètre $_Get['action'] est présent dans la liste, celui sera mis en évidence

Edition Rapport Todo

Pour éditer un rapport un icone crayon est présent, et pour sortir de ce mode il faut fermer le formulaire d'édition

Les listes déroulentes jour et créneau possèdent un champ vide avec comme paramètre "selected disabled hidden" afin d'avoir une option par défaut non-sélectionnable

Si les champs jour et créneau sont renseignés, deux champs permettent d'ajouter des action existante ou d'en crée une nouvelle.

Envois de formulaire en javascript

La fonction javascript post() permet de créer et d'envoyer direcement un formulaire en post sans avoir à le créer dans le code html

Il faut lui fournir les paramètres path et un tableau des paramètres comme ceci :

```
const param = { param1: 1, param2: 2 };
post("?action=exemple", param )
```

Comment fonctionne le formulaire de modification des données d'un rapport de stupéfiant ?

Les données sont envoyées par méthode POST comme nous le ferions en général avec des formulaires HTML.

La subtilité de ce formulaire est que tous les champs qui ont des données à envoyer ont un attribut "name" qui est un index de tableau. Cela permet d'avoir du côté du Controller PHP une variable qui est un array que l'on peut parser efficacement avec un foreach.

Comment fonctionnent les boutons qui sont déjà à l'intérieur d'un autre formulaire dans la partie des rapports de stupéfiants ?

La problématique est qu'il ne faut pas faire de formulaire dans un autre formulaire. Donc la solution suivante a été mise en place : Les boutons appellent une fonction javascript en lui donnant les données en paramètres et la fonction créent un formulaire caché qui est ajouté à la fin de la page et envoyés.