

RAPPORT DU PROJET

Gestion de stock de la Croix Rouge



BTS Systèmes Numériques Informatique et Réseaux | Session 2022

Thomas VALLET

Matenin TRAORE

Nethusan SURESHKUMAR

Table des matières

I. Remerciements.....	4
II. Partie commune	
Présentation de la société.....	5
Présentation du projet.....	6
Objectifs du projet.....	7
Spécifications du projet.....	8
Cahier des charges.....	10
Diagramme de Gantt.....	11
Diagramme de cas d'utilisation.....	12
Diagramme de déploiement.....	13
Diagramme de séquence.....	14
Répartition des tâches.....	15
Partie physique : QR Code.....	16
Partie théorique.....	16
Partie physique.....	18
III. Partie étudiant 1 (Matenin TRAORE)	
Introduction.....	21
Matériel utilisée.....	23
Tâches personnelles.....	24
Base de données.....	24
Serveur WAMP.....	25
API REST.....	26
Raspberry PI 4.....	30
Journal Lumineux.....	32

Fin de test.....	34
Conclusion.....	42
IV. Partie étudiant 2 (Thomas VALLET)	
Introduction.....	43
Tâches personnelles.....	44
Logiciel de codage.....	45
Langages de programmation.....	46
HTML.....	47
CSS.....	50
JavaScript.....	51
PHP.....	53
Logiciel de connexion du mini PC.....	55
Logiciel phpMyAdmin.....	57
Résultats du site.....	59
Conclusion.....	62
V. Partie étudiant 3 (Nethusan SURESHKUMAR)	
Introduction.....	63
Tâches personnelles.....	64
Mise en place du mini PC.....	65
Base de données.....	67
Codage.....	69
Conclusion.....	79
VI. Conclusion.....	80



I. Remerciements

Nous souhaitons adresser nos remerciements à toute l'équipe enseignante pour nous avoir soutenus, guider et émettre leurs avis tout au long du projet.

Dans un premier temps, nous remercions Mr Iervese, Mr Georges et Mr Hagot qui, pendant treize semaines au lycée Louis Armand de Nogent sur Marne, en tant qu'enseignants d'informatique et réseaux nous ont guidés pour élaborer tout le projet avec les programmes que nos pratiquons.

Dans un second temps, nous remercions également Mr Schlee, en tant qu'enseignant de physique-chimie qui nous a aidés à avancer dans la partie physique du projet. Une fois leurs critiques et leurs conseils annoncés, cela nous a permis d'améliorer le projet, puis de le finaliser.



II. Partie commune

Présentation de la société

La Croix-Rouge française (CRF) est une association d'aide humanitaire française fondée en 1864. Elle a pour objectif de venir en aide aux personnes en difficultés en France et à l'étranger. Ses missions fondamentales sont l'urgence, le secourisme, l'action sociale, la formation, la santé et l'action internationale.

Reconnue d'utilité publique depuis 1945, la Croix-Rouge française est l'une des 191 sociétés nationales du Mouvement international de la Croix-Rouge et du Croissant-Rouge. Elle compte plus de 62 000 bénévoles et 17 000 salariés. Son réseau est constitué de 1 062 implantations locales, 108 délégations départementales et territoriales et 634 établissements sociaux, médico-sociaux et sanitaires, répartis sur l'ensemble du territoire français, y compris dans les départements et les territoires d'outre-mer. La Croix-Rouge française est par ailleurs présente dans 27 pays pour y développer des programmes spécifiques afin d'améliorer notamment l'accès à l'eau, la précarité alimentaire et la santé des personnes vulnérables.



Présentation du projet

Le projet de la Croix-Rouge est constitué de trois étudiants :

1 – Matenin TRAORE

2 – Thomas VALLET

3 – Nethusan SURESHKUMAR

Il s'agit de fournir un outil simple de gestion de stock pour avoir toujours une quantité suffisante des produits à forte demande ainsi que pour gérer les produits possédant des dates de péremption.

La croix rouge possède en permanence un stock de produits et de denrées et non périssables. Chaque produit est rangé et répertorié dans des armoires et sur des étagères dans un ordre précis. Pour être disponible en permanence, ces produits doivent être en quantité suffisante et donc posséder une limite de stock minimale.

Actuellement, la gestion de ces produits n'étant pas informatisée, il est nécessaire de surveiller en permanence l'état des stocks de chacun des produits. Mais il arrive trop fréquemment que le stock de certains produits arrive en dessous de la limite minimale.

Pour améliorer la gestion de stock, la croix rouge souhaite automatiser la gestion de stock de ces produits.

Domaine d'activité du système support d'étude :	<input type="checkbox"/> télécommunications, téléphonie et réseaux téléphoniques ; <input checked="" type="checkbox"/> informatique, réseaux et infrastructures ; <input type="checkbox"/> multimédia, son et image, radio et télédiffusion ; <input checked="" type="checkbox"/> mobilité et systèmes embarqués ; <input type="checkbox"/> électronique et informatique médicale ; <input type="checkbox"/> mesure, instrumentation et micro-systèmes ; <input type="checkbox"/> automatique et robotique.
---	---



Objectifs du projet

Améliorer la gestion des denrées périssables (avec dates limites)

Améliorer la disponibilité de certains produits avec une quantité de stock minimale à ne pas dépasser



Spécifications du projet

Dans ce projet, il y existe plusieurs contraintes de réalisation.

Contraintes financières (budget alloué) :

Respecter les contraintes matérielles imposées

Contraintes de développement (matériel et/ou logiciel imposé / technologies utilisées) :

Contraintes matérielles et logicielles :

Le serveur Web et la base de données seront hébergés sur une Raspberry PI 3.

La base de données de stockage

Voici la liste minimales des données qui doivent apparaître dans la base de données :

- Il existe 4 catégories de produits : Formation, Secours, Hygiène, Alimentaire.
- Pour chaque type de produit (ou de sous-produit) :
 - Le nom du produit
 - La catégorie
 - L'emplacement (n° de l'armoire, n° de l'étagère)
 - La quantité minimum (si nécessaire)
 - Le QrCode
 - S'il s'agit d'un produit avec date de péremption (vrai faux)
 - Le nom du produit père s'il s'agit d'un sous-produit
- Pour chaque produit (ou sous-produit) entrant/sortant
 - le type de produit (ou sous-produit)
 - la quantité
 - la date de péremption

La notion de sous catégorie ou de sous produit est optionnelle.

L'application des utilisateurs du stock

L'application qu'utiliseront les utilisateurs du stock de la Croix Rouge devra leur permettre

- A) de saisir l'approvisionnement d'un produit dans le stock,
- B) de saisir la sortie d'une quantité d'un produit du stock,
- C) de consulter la liste des produits dont la quantité minimale est dépassée,
- D) de retirer des produits dont la date de péremption est dépassée.

Reprenons ces points un à un en les détaillant :

- A) Saisir l'approvisionnement d'un produit dans le stock
 - 1) Identification du produit en scannant le QrCode.
 - 2) Saisie de la quantité.
 - 3) Saisie de la date de péremption, elle pourra être indiquée
- B) Saisir la sortie d'une quantité d'un produit du stock
 - 1) Identification du produit en scannant son QrCode
 - 2) Choix parmi une liste de produits à date de péremption la plus courte (s'il s'agit d'un produit à date de péremption)
 - 3) Saisie de la quantité sortie (avec vérification que la quantité indiquée soit inférieure à la quantité disponible en stock)
- C) Consulter la liste des produits dont la quantité minimale est dépassée
 - 1) Afficher les informations précises sur l'emplacement du produit périssable sur la zone de stockage (n° armoire, n° étagère) ainsi que la quantité des produits périssables
- D) Retirer des produits dont la date de péremption est dépassée.
 - 1) Identification du produit en scannant son QrCode
 - 2) Choix parmi une liste de produits à date de péremption atteinte
 - 3) Saisie de la quantité sortie (avec vérification que la quantité indiquée soit inférieure à la quantité disponible en stock)

L'application de l'administrateur

Seul l'administrateur peut créer de nouvelles catégories, de nouveaux produits ou de nouveaux sous-produits.

Les outils de développement

- Les applications en C++ seront développées avec les EDI suivants (au choix) : Qt Creator et C++ Builder
- Le développement Web se fera en HTML5, CSS3, PHP5 et JS.

Tous les programmes doivent être simples robustes et léger, ils doivent être indépendants de la machine utilisée

Les configurations doivent être facilement reproductibles : toutes les configurations seront stockées dans des fichiers textes

Contraintes qualité (conformité, délais, ...)

Applications WEB compatibles avec tous les navigateurs.

Contraintes de fiabilité, sécurité :

Tous les programmes doivent être simples robustes et léger, ils doivent être indépendants de la machine utilisée

Les configurations doivent être facilement reproductibles : toutes les configurations seront stockées dans des fichiers textes.



Cahier des charges

La gestion d'un stock de matériel et de denrée de plusieurs centaines de kilos nécessite une automatisation de la gestion, du rangement, des quantités et de la surveillance des dates de péremption autour d'une base de données.

Localiser un produit dans la zone de stock

Chaque catégorie de produit sera identifiée par un QR Code. Ce QR Code sera affiché sur l'étagère et/ou sur l'armoire où se trouve le produit.

L'utilisateur utilisera soit un lecteur QR Code relié à un PC ou une application dédiée sur un Smartphone pour ajouter ou retirer un produit de la zone de stock.

Ajouter un produit dans la base de données

L'utilisateur scannera le QR Code du produit à ajouté. Il ajoutera dans la base de données les informations suivantes sur le produit à ajouter :

1. La quantité du produit s'il s'agit d'une donnée critique
2. La date de péremption du produit s'il s'agit d'une donnée critique

De plus, chaque produit doit être répertorié avec son lieu de stockage (n° d'armoire, et n° d'étagère dans l'armoire)

Une alerte doit être émise, soit sur un afficheur, soit par SMS ou par mail pour les produits dont la date de péremption ou la quantité minimale est dépassée.

Retirer un produit de la base de données

L'utilisateur qui souhaite prendre un produit dans le stock doit scanner le QR Code du produit puis saisir les informations suivantes :

1. La quantité du produit s'il s'agit d'une donnée critique
2. La date de péremption s'il s'agit d'une donnée critique. S'il existe des dates de péremption plus courte sur le produit choisi, une alerte sera alors émise.

Administre la base de données

L'administrateur de la base de données pourra ajouter de nouvelles catégories de produit. Il devra alors indiquer l'armoire et l'étagère de cette nouvelle catégorie, générer un QR Code et l'imprimer pour l'apposer sur l'étagère.

Diagramme de Gantt

Répartition hebdomadaire des tâches du projet (13 semaines de 13 heures)

	Tâches	Nombre d'heures	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
0.1	Cas d'utilisation														
0.2	Déploiement														
0.3	Séquence														
1.1	Écriture de l'API REST sur le serveur web : produits à date de péremption dépassée et à quantité minimale d'article atteinte														
1.2	Écriture client REST - récupération des produits à date de péremption dépassée et/ou à quantité minimale atteinte														
1.3	Extraction des données REST reçues (au format XML ou JSON)														
1.4	Affichage de plusieurs messages consécutifs sur le journal lumineux														
1.5	Intégration des données reçues dans les trames à envoyer au journal lumineux														
2.1	Dessiner une esquisse de site ergonomique, simple d'utilisation et complet														
2.2	Mise en page du site web														
2.3	Gestion de l'approvisionnement d'un produit dans le stock à partir de son QrCode														
2.4	Gestion de la sortie d'un produit du stock à partir de son QrCode														
2.5	Gestion de l'affichage des produits périssables														
2.6	Gestion de l'affichage des produits dont la quantité minimale est atteinte														
3.1	Mise en place du MCD														
3.2	Installation de la Raspberry, du serveur Web, de la BDD et des outils nécessaires														
3.3	Hébergement de la base de données														
3.4	Définition du contenu et du codage du QrCode														
3.4	Gestion de l'ajout d'une nouvelle catégorie														
3.4	Gestion de l'ajout d'un nouveau produit dans le stock et génération de son QrCode pour impression.														
3.4	Affichage de l'évolution du stockage des produits avec des graphiques.														
3.8	Chef de projet														
	Rapport								R 1						R 3

Diagramme de cas d'utilisation

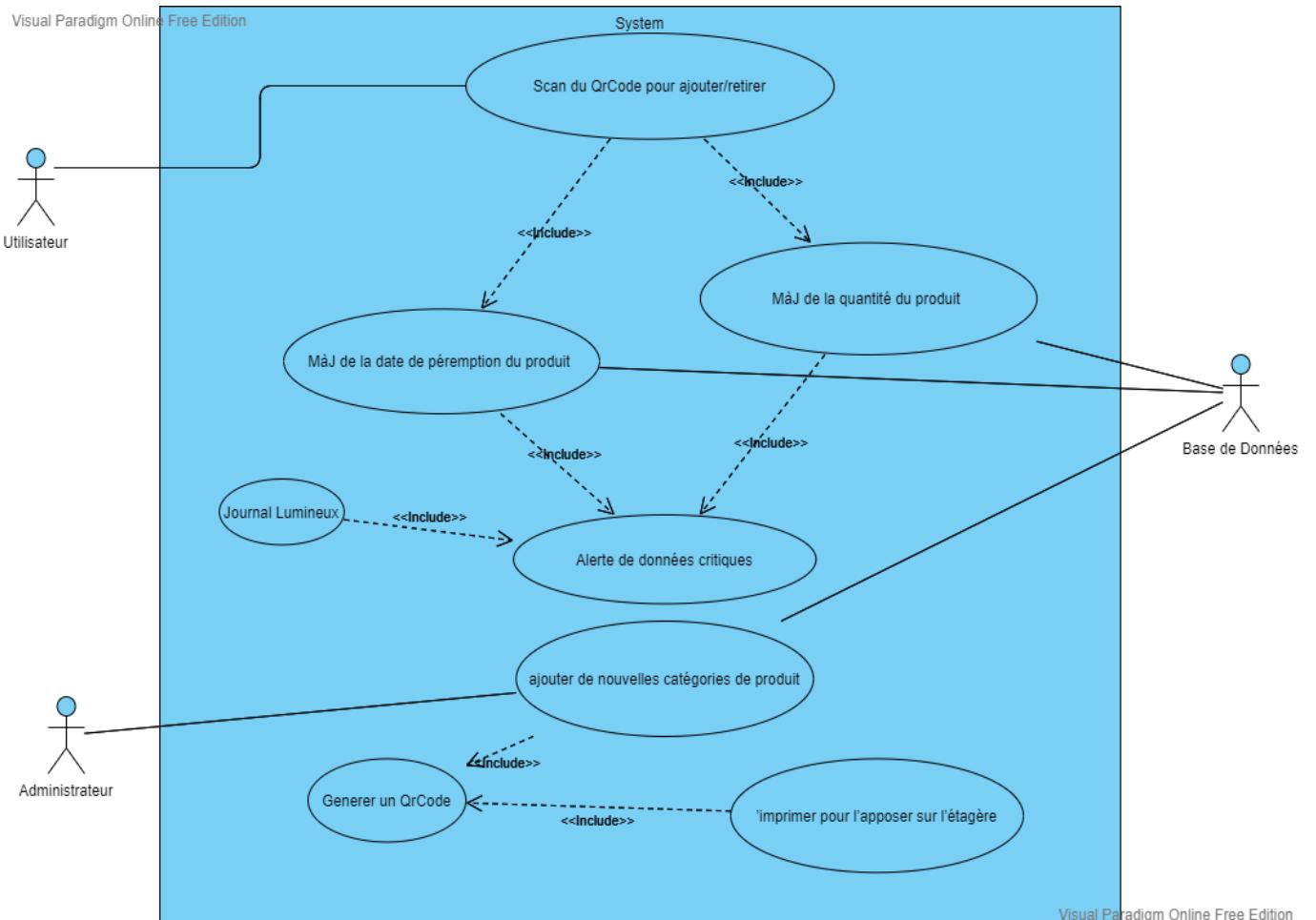


Diagramme de déploiement

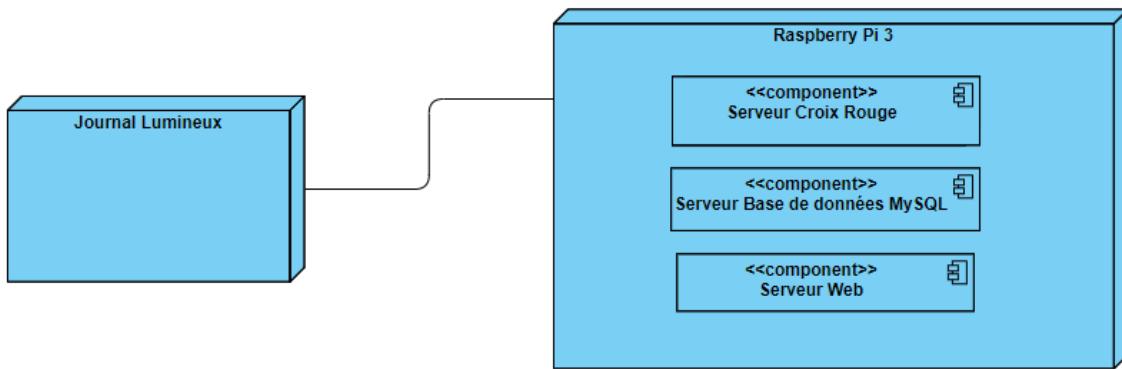
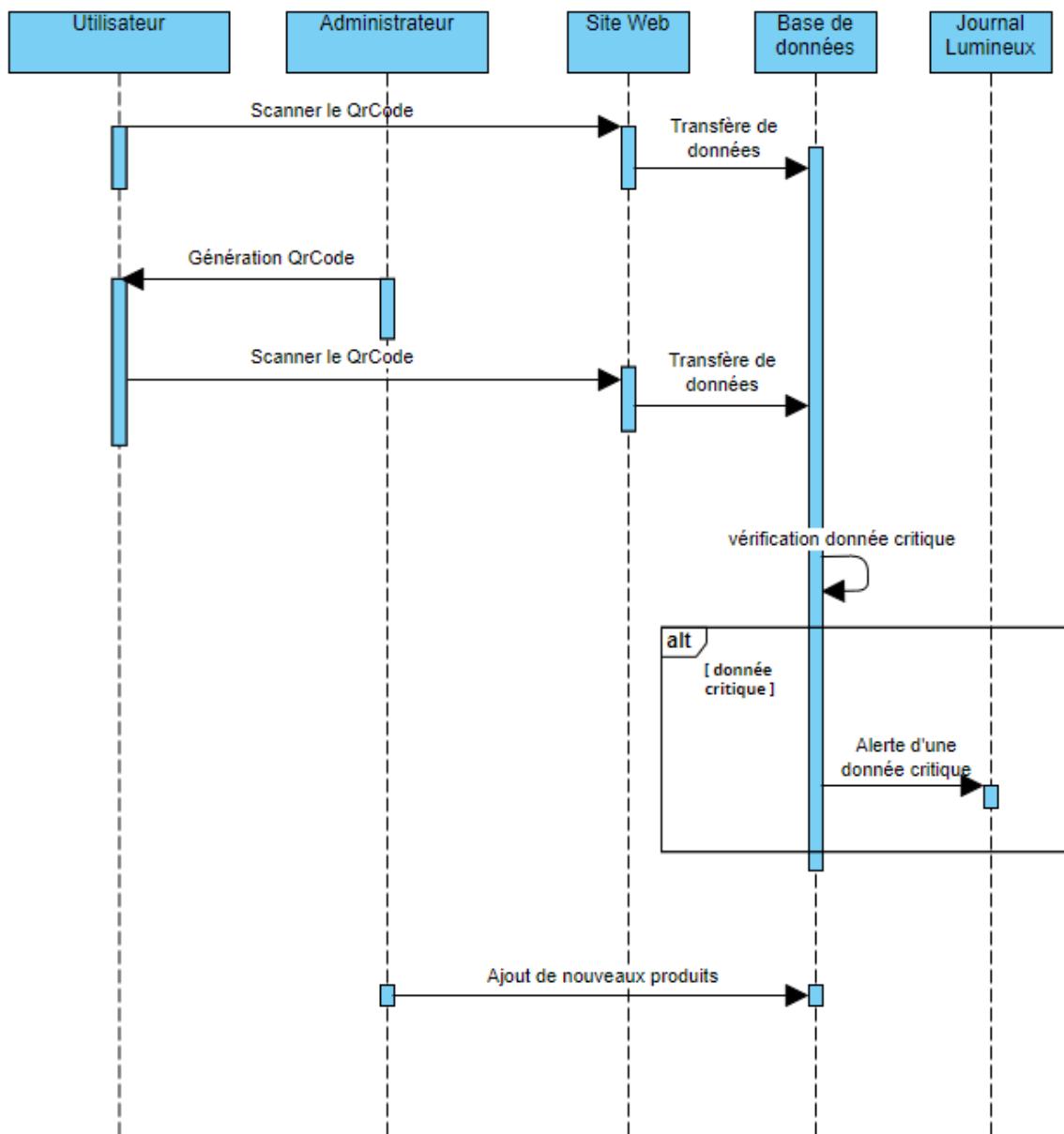


Diagramme de séquence



Répartition des tâches

	Fonctions à développer et tâches à effectuer	
Étudiant 1 EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	BDD + API REST + Journal lumineux	<ul style="list-style-type: none"> ◦ 1.1 Écriture de l'API REST sur le serveur web : produits à date de péremption dépassée et produits à quantité minimale d'article atteinte ◦ 1.2 Écriture d'un client REST sur la Raspberry du journal lumineux - récupération des produits dont la date de péremption est dépassée et dont la quantité minimale est atteinte ◦ 1.3 Extraction des données REST reçues (au format XML ou JSON) ◦ 1.4 Affichage de plusieurs messages consécutifs sur le journal lumineux ◦ 1.5 Intégration des données reçues dans les trames à envoyer au journal lumineux
Étudiant 2 EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	Développement du site web – partie utilisateur	<ul style="list-style-type: none"> ◦ 2.1 Dessiner une esquisse de site ergonomique, simple d'utilisation et complet ◦ 2.2 Mise en page du site web ◦ 2.3 Gestion de l'approvisionnement d'un produit dans le stock à partir de son QrCode ◦ 2.4 Gestion de la sortie d'un produit du stock à partir de son QrCode ◦ 2.5 Gestion de l'affichage des produits périmés ◦ 2.6 Gestion de l'affichage des produits dont la quantité minimale est atteinte
Étudiant 3 EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	BDD + Développement du site web – partie administrateur	<ul style="list-style-type: none"> ◦ 3.1 Mise en place du MCD ◦ 3.2 Installation de la Raspberry, du serveur Web, de la BDD et des outils nécessaires ◦ 3.3 Hébergement de la base de données ◦ 3.4 Définition du contenu et du codage du QrCode ◦ 3.5 Gestion de l'ajout d'une nouvelle catégorie ◦ 3.6 Gestion de l'ajout d'un nouveau produit dans le stock et génération de son QrCode pour impression. ◦ 3.7 Affichage de l'évolution du stockage des produits avec des graphiques. ◦ 3.8 Chef de projet

Partie physique : QR Code

→ Partie théorique

Définition

Un code QR, en anglais QR Code, en forme longue quick response code, « code à réponse rapide », est un type de code-barres à deux dimensions.

Les différents types de QR Code

Il existe deux types de QR Code. Nous avons des QR Codes dit statiques et des QR Codes dit dynamiques.



STATIC
QR Codes are permanent



DYNAMIC
QR Codes are editable

QR Code Statique : Les codes QR statiques ont une URL intégrée avec une destination fixe. L'URL fait partie du QR Code, ce qui signifie que nous ne pouvons pas modifier des QR Codes dit statiques.

QR Code Dynamique : Les codes QR dynamiques ont une URL courte intégrée dans le code, qui peut rediriger l'utilisateur vers l'URL du site Web de destination. L'URL de destination peut être modifiée après la génération du code QR, tandis que l'URL courte incorporée dans le code reste la même.

Les tailles

Un QR Code doit être soumis à des normes au niveau de la taille pour pouvoir être lu pour être certains que notre QR Code soit lisible il ne faut plus qu'il soit plus petit que 3cm par 3cm.

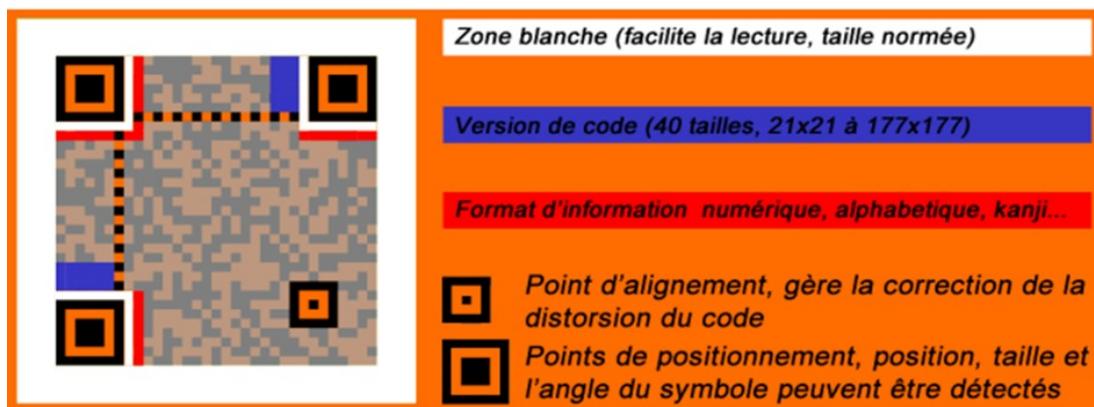
Les types de redondance

En effet, la redondance est employée dans le QR Code de manière à prévenir toute altération du motif et ainsi fournir au lecteur l'information codée originale sans problèmes.

Il existe deux types de redondance, nous avons une redondance dite active qui envoie des informations simultanément (redondance informationnelle) et une redondance dite passive qui envoie des informations à la demande (redondance physique ou matérielle).

Nous sommes dans le cas du QR Code dans un type de redondance active.

Codage des données



- **Les points noirs correspondent aux données et aux touches de corrections d'erreur**

Les versions

Les versions de symboles du code QR vont de la version 1 à la version 40. Chaque version a une configuration de module ou un nombre de modules différents. (Le module fait référence aux points noirs et blancs qui composent le QR Code).

→ Partie physique

Corrections d'erreur

Le QR Code utilise le système de correction d'erreur Reed-Solomon, les QR Codes peuvent incorporer des images, telles que logos ou dessins, sans perdre les informations utiles à la lecture du code.

Le code contient jusqu'à 30 % de redondance.

Capacité à corriger les erreurs :

- Niveau L : environ 7 % de redondance
- Niveau M : ~ 15 %
- Niveau Q : ~ 25 %
- Niveau : ~ 30 %

Distance

La taille d'impression du QR Code dépend de la distance à laquelle le lecteur va le scanner.

Informations

Dans un QR Code la couleur Blanche correspond au 0 et le Noir au 1, les trois motifs dans les coins des QR Codes permettent la détection de l'orientation des QR Codes.



Format information

Les modèles de format contiennent des informations sur la tolérance d'erreur et le modèle de masque de données et facilitent la lecture du code.



Modèle de synchronisation

À l'aide de ces lignes, le scanner détermine la taille de la matrice de données.

Masquage

Si une module dans le code QR est « masqué », cela signifie simplement que s'il s'agit d'un module clair, il doit être remplacé par un module sombre, et s'il s'agit d'un module sombre.

Il doit être remplacé par un module clair. En d'autres termes, le masquage signifie simplement basculer la couleur du module.

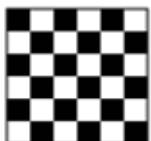
Il existe 8 masques différents et il va falloir trouver celui qui donne le résultat le plus équilibré : 000 ; 001 ; 010 ; 011 ; 100 ; 101 ; 110 ; 111.

Quoi masquer ?

Les modèles de masque doivent UNIQUEMENT être appliqués aux modules de données et aux modules de correction d'erreur. Autrement dit :

- ◆ Ne masquez pas les modèles de fonction (modèles de recherche, modèles de synchronisation, séparateurs, modèles d'alignement)
- ◆ Ne masquez pas les zones réservées (zone d'informations sur le format, zone d'informations sur la version)

Differentes masques QR Code



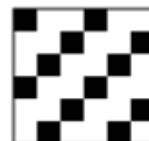
Mask 000
 $(i + j) \% 2 = 0$



Mask 001
 $i \% 2 = 0$



Mask 010
 $j \% 3 = 0$



Mask 011
 $(i + j) \% 3 = 0$



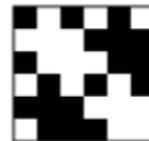
Mask 100
 $(i/2 + j/3) \% 2 = 0$



Mask 101
 $(i*j) \% 2 + (i*j) \% 3 = 0$

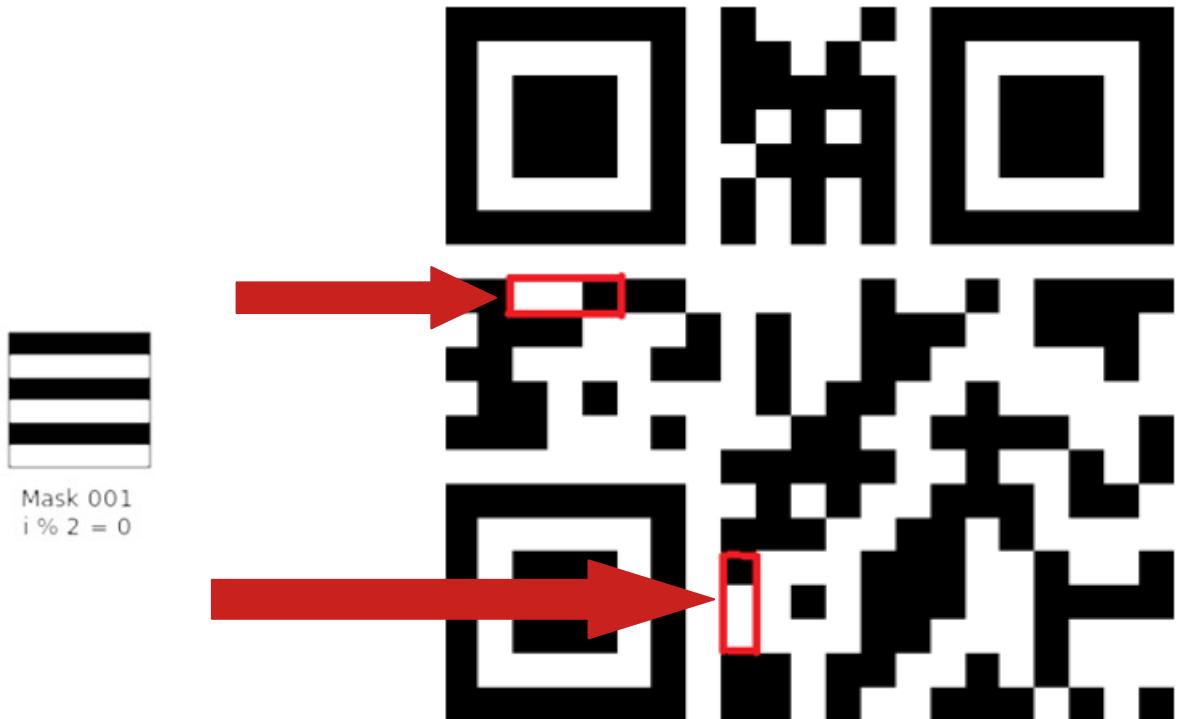


Mask 110
 $((i*j) \% 3 + i*j) \% 2 = 0$



Mask 111
 $((i*j) \% 3 + i + j) \% 2 = 0$

Décodage Masque



Noir & Noir = Blanc

Blanc & Blanc = Blanc

Noir & Blanc (différence) = Noir

Le décodage partiel du QR Code sera effectué lors de la revue 3 de la partie Physique.

III. Partie étudiant 1 (Matenin TRAORE)

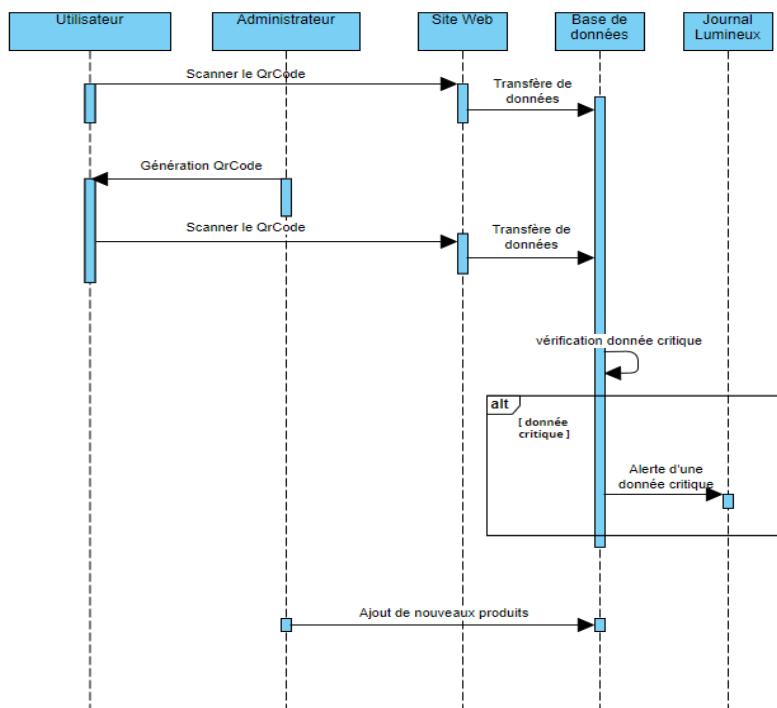
Introduction

La Croix Rouge dispose de différent produit qui correspondent à différente catégorie, il y a 4 catégories qui sont : Formation, Secours, Hygiène et Alimentation.

Dans chaque produit on y retrouve son nom, son emplacement, sa quantité & quantité minimum, son QR Code (Permettant le scan d'un produit sur l'interface web) et enfin sa catégorie.

Mon projet consiste à mettre en place un système de gestion de produit qui permettrait à la Croix Rouge d'avoir un contrôle sur les produits qu'elle possède qui arrive à une date de péremption.

Voici la représentation du système à travers le temps :



L'utilisateur fera un **scan** de QR code sur le site web un **transfert de données** se fera par la suite sur la base de données, il y aura également la possibilité que l'administrateur **génère un QR code** qu'il renvoi à l'utilisateur qui amène à un **scan de QR code** puis à un **transfert de données** suite à sa réception par la base de données j'entamerai **une vérification de donnée critique** s'il y a donnée critique alors j'envoie une **alerte d'une donnée critique** l'administrateur à également la possibilité **l'ajout à de nouveaux produits.**



Visual Studio Code

J'ai choisi ce IDE car il possède de nombreux avantage par le cahier des charges de mon projets.

- Un terminal directement intégrer
- Possibilité de déboguer des applications directement sur l'éditeur
- Fonctionnalité **IntelliSense** assistant de complétion de code qui liste chaque ligne de code que je produis, qui propose des suggestions lors de l'écriture ce qui permet d'écrire mon code plus rapidement et plus proprement.
- Débogueur permettant de vérifier les erreurs dans le code et d'identifier la ou les causes avec des suggestions.
- Éditeur Gratuit

Cet éditeur ma permis de me sentir à l'aise car je l'utilise régulièrement et également car il ma apporter un gain de temps et d'énergie.



Pour ce projet je me suis également servi de **WampServer**
Qui est une plateforme de développement Web de type WAMP,
permettant de faire fonctionner localement (sans avoir à se connecter
à un serveur externe) des scripts PHP.

MATÉRIEL UTILISÉE

Le matériel mis à ma disposition pour ce projet est :

- **Un Ordinateur**

L'ordinateur me permettra de mettre en œuvre et de tester mes différents codes, le système d'exploitation utilisé sera Windows 10 car je possède une meilleure affinité avec le codage sous Windows.

- **Une Raspberry PI 4**



La Raspberry PI 4 va me permettre d'héberger et d'exécuter mon code en C++, elle sera liée avec le journal lumineux.

- **Un Journal Lumineux**



Le journal lumineux qui est liée avec la Raspberry PI 4 va me permettre d'y afficher des messages qui correspondent à la date de péremption des différents produits chez La Croix Rouge.

Tâches personnelles

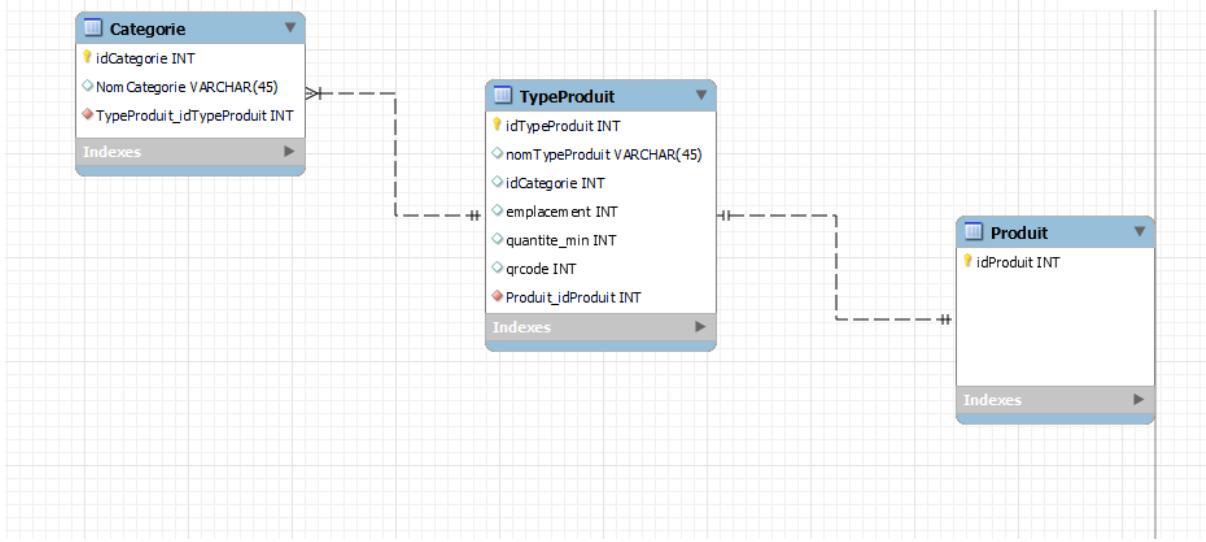
→ Base de données

J'ai dans ce projet en premier lieu réfléchi à la structure d'une base de données en fonction des besoins de La Croix Rouge, la base de données va nous permettre de rassembler les différentes informations sur les produits que la croix rouge possède.

Elle nous permettra également de pouvoir avoir le contrôle sur les produits on va pouvoir :

- Mettre à jour les informations d'un produit (Nom, Quantité, Date de Péremption)
- Ajouter des nouveaux produits
- Retirer des produits
- Extraire les informations d'un produit

Pour mettre en place l'architecture je me suis servi du logiciel MySQL WorkBench



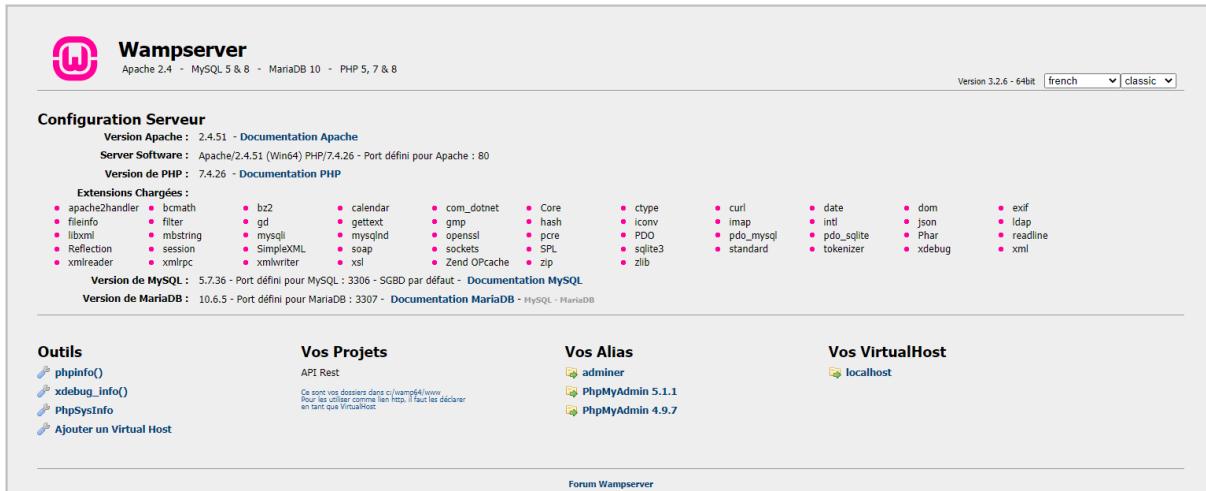
Architecture base de données

Notre base de données contient **3 tables** :

- Catégorie : Va définir la catégorie d'un produit qui peut être un produit de type **Formation, Secours, Hygiène ou Alimentation**.
- TypeProduit : Va contenir les informations sur un produit spécifique (nom du type de produit, catégorie, emplacement, quantité minimal, QR code).
- Produit : Va contenir les produits de La Croix Rouge.

→ Serveur WAMP

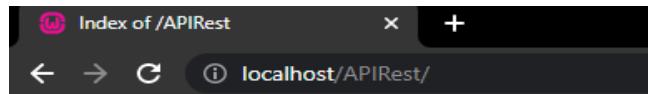
J'ai dans ce projet utilisé un serveur WAMP, ce serveur va dans un premier temps être utile pour la phase test de mon projet, je vais pouvoir tester le code que je vais créer et pouvoir par la suite basculer sur le serveur commun via une Raspberry que mon camarade Nethusan SURESHKUMAR (PROJET 3) aura mis en place.



The screenshot shows the Wampserver control panel interface. It includes sections for Configuration Server, Extensions Chargées, Outils, Vos Projets, Vos Alias, and Vos VirtualHost. The Configuration Server section displays Apache 2.4.51, MySQL 5 & 8, MariaDB 10, and PHP 5, 7 & 8. The Extensions Chargées section lists various PHP extensions. The Outils section contains links for phpinfo(), xdebug_info(), PhpsysInfo, and Ajouter un Virtual Host. The Vos Projets section shows API Rest. The Vos Alias section lists adminer, PhpMyAdmin 5.1.1, and PhpMyAdmin 4.9.7. The Vos VirtualHost section lists localhost.

Installation du serveur WAMP

Le serveur WAMP va contenir mon répertoire **API Rest (Visible dans Vos Projets)** il contiendra les fichiers rest.php (le code de l'API Rest) et model.php (l'affichage du code).



Index of /APIRest

Name	Last modified	Size	Description
Parent Directory		-	
model.php	2022-05-11 10:20	9	
rest.php	2022-05-23 08:52	772	

Visualisation de l'accès à mon répertoire via l'url : localhost/APIRest/

→ API REST

L'API Rest va me permettre de communiquer avec la base de données et d'extraire les informations ici la date de péremption des produits dont l'échéance approche et également les produits dont la quantité atteint un seuil critique.

Le langage de programmation utilisé pour cette partie sera le PHP.

```
<?php

// Requière
require ('./model.php');

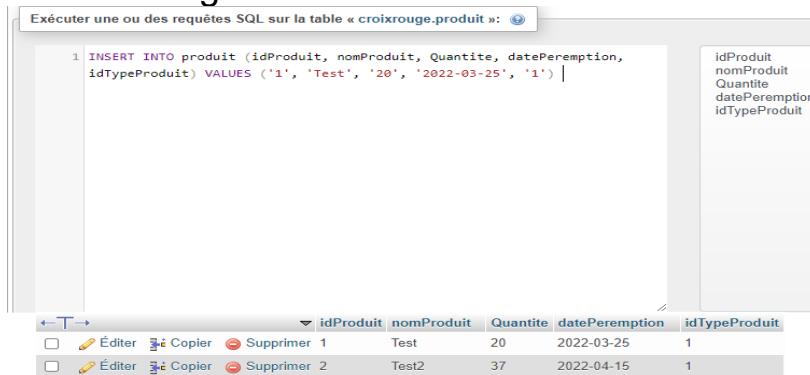
// Connexion Base de données
$mysqli = new mysqli("localhost","root","","croixrouge");

*Connexion à la base de données locale*
```

Pour pouvoir communiquer avec la base de données je dois m'y connecté, je me connecte à la base de données en utilisant **new mysqli** qui attend pour argument

[1] l'hôte, [2] l'identifiant, [3] le mot de passe et [4] la base de données

Je fais également une inclusion du fichier model.php en utilisant **require** en php qui permettra l'affichage.



The screenshot shows a MySQL database interface. On the left, there is a text area where an SQL query is being typed:

```
Exécuter une ou des requêtes SQL sur la table « croixrouge.produit »: 
1 INSERT INTO produit (idProduit, nomProduit, Quantite, datePeremption, idTypeProduit) VALUES ('1', 'Test', '20', '2022-03-25', '1')
```

To the right of the query, there is a table definition:

	idProduit	nomProduit	Quantite	datePeremption	idTypeProduit
--	-----------	------------	----------	----------------	---------------

Below the table definition, there is a data grid showing four rows of product data:

	idProduit	nomProduit	Quantite	datePeremption	idTypeProduit
<input type="checkbox"/>	1	Test	20	2022-03-25	1
<input type="checkbox"/>	2	Test2	37	2022-04-15	1
<input type="checkbox"/>	3	Test3	50	2022-05-18	1
<input type="checkbox"/>	4	Test4	67	2022-06-13	1

A message at the bottom of the interface reads: *Insertion de produit de test dans la base de données*

Je vais par la suite ajouter dans la base de données des produits de test qui me permettront l'analyse de mon code, pour cela je vais exécuter une requête SQL dans la base de données.

La requête **INSERT INTO** me permettre l'ajout de produit elle attend pour argument

[1] La table (ici produit), [2] Le nom des colonnes (ici idProduit, nomProduit, Quantite, datePeremption,idTypeProduit), [3] Les Valeurs

Après avoir ajouté les produits de test dans la base de données je vais pouvoir les récupérer en PHP.

```
// Requête quantité minimale d'article atteinte
$quantite = "SELECT NomTypeProduit,Quantitemin FROM TypeProduit";
$res= $mysqli->query($quantite);
$rep=$res->fetch_assoc();
// Affichage
echo nl2br(" | Quantité Minimal | \n");
print_r($rep);
```

rest.php - Code Récupération de la Quantité Minimal

Je vais dans un premier temps récupérer la quantité minimale d'un produit, pour cela je vais créer une variable **\$quantite** ou je vais stocker la requête SQL me permettant de récupérer l'information sur la quantité minimal, je vais ensuite créer une variable **\$res** qui va contenir la réponse de la requête, je vais ensuite me connecté dans la base de données avec la variable **\$mysqli** puis faire un **query** de la requête pour la récupérer, La variable **\$rep** va me permettre de créer un tableau associatif de la réponse **\$res** en utilisant **fetch_assoc()**, puis je vais faire un **echo nl2br** pour identifié la réponse et un **print_r** de la réponse **\$rep** pour afficher la requête final.

| Quantité Minimal |
 Array ([NomTypeProduit] => Test [Quantitemin] => 10)

Affichage de la récupération de la Quantité Minimal

Après avoir récupérer la quantité minimale je vais récupérer les produits qui arrivent à date de péremption je vais rentamer le même processus, il y aura des changements au niveau des variables **[\$quantite → \$peremption]** **[\$res → \$resp]** **[\$rep → \$repp]** et je vais ajouter une boucle **while** avec pour condition **\$repp** pour pouvoir récupérer toutes les dates qui arrivent à péremption.

```
// Requête produit à date de péremption dépassée
$peremption = "SELECT nomProduit,datePeremption FROM produit ORDER BY datePeremption";
$resp= $mysqli->query($peremption);
while($repp=$resp->fetch_assoc()) {
// Affichage
echo nl2br("\n");
echo nl2br(" | Date de péremption | \n");
print_r($repp);
}

| Date de péremption |
Array ( [nomProduit] => Test [datePeremption] => 2022-03-25 )
| Date de péremption |
Array ( [nomProduit] => Test2 [datePeremption] => 2022-04-15 )
| Date de péremption |
Array ( [nomProduit] => Test3 [datePeremption] => 2022-05-18 )
| Date de péremption |
Array ( [nomProduit] => Test4 [datePeremption] => 2022-06-13 )
```

rest.php - Code Récupération de la Date de Péremption & Affichage

```

<?php

// Requière
require ('./model.php');

// Connexion Base de données
$mysqli = new mysqli("localhost", "root", "", "croixrouge");

$req_type=$_SERVER['REQUEST_METHOD'];

if($req_type==='GET') {

    // Quantité Minimal
    $quantite = "SELECT NomTypeProduit,Quantitemin FROM TypeProduit";
    $res= $mysqli->query($quantite);
    $Quantite=$res->fetch_assoc();
    echo nl2br(" Réponse : ");
    echo json_encode($Quantite);
    echo "<br />";
    echo "<br />";

    // Date de péremption
    $peremption = "SELECT nomProduit,datePeremption FROM produit ORDER BY datePeremption";
    $resp= $mysqli->query($peremption);

    while($repp=$resp->fetch_assoc()) {
        $peremption = $repp;
        echo nl2br(" \n Réponse : ");
        echo json_encode($peremption);
    }

}

?>

```

rest.php - Code Extraction des données REST en JSON

Après avoir pu communiquer avec la base de données et pu extraire des données sur des produits, je vais maintenant extraire les données en format JSON cela me permettra plus tard d'envoyer des messages au journal lumineux, pour obtenir les données en format JSON je fais un **echo json_encode** sur les variables **\$Peremption** et **\$Quantite**.

Réponse : {"NomTypeProduit": "Test", "Quantitemin": "10"}

Réponse : {"nomProduit": "Test", "datePeremption": "2022-03-25"}
 Réponse : {"nomProduit": "Test2", "datePeremption": "2022-04-15"}
 Réponse : {"nomProduit": "Test3", "datePeremption": "2022-05-18"}
 Réponse : {"nomProduit": "Test4", "datePeremption": "2022-06-13"}

Affichage de l'extraction des données sous format JSON

J'en profite également pour créer une variable **\$req_type** qui va me permettre d'accéder à **REQUEST_METHOD** avec la variable **\$_SERVER**, je vais pouvoir sous condition de **if** rendre les données accessibles uniquement par la méthode **GET**.

Request

Method: GET Request URL: http://localhost/API%20Rest/rest.php

Parameters ^

Headers	Variables
<input type="checkbox"/> <input type="button" value="Toggle source mode"/> <input type="button" value="Insert headers set"/>	
Header name	Header value
ADD HEADER	
A Headers are valid	
Headers size: 0 bytes	
200 OK 6.60 ms	DETAILS
<input type="checkbox"/> <input type="button"/> <input type="button"/> <input type="button"/>	
Réponse : {"NomTypeProduit": "Test", "Quantitemin": "10"} Réponse : {"nomProduit": "Test", "datePeremption": "2022-03-25"} Réponse : {"nomProduit": "Test2", "datePeremption": "2022-04-15"} Réponse : {"nomProduit": "Test3", "datePeremption": "2022-05-18"} Réponse : {"nomProduit": "Test4", "datePeremption": "2022-06-13"}	

*Visu
alisation de la requête GET avec un REST Client*

En utilisant un REST Client (Extension Advanced REST Client) sous chrome, je peux vérifier si ma requête GET est prise en compte et si les données sont reçues en choisissant la méthode GET et entrant l'URL qui donne accès aux données.

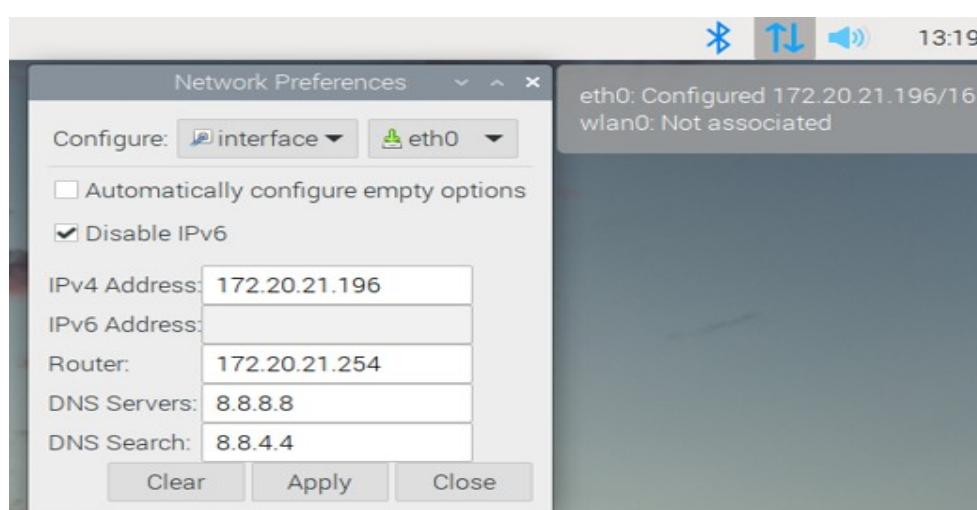
Après observation je reçois bien mes données confirmé par la réponse '**200 OK**' qui en HTTP indique la réussite d'une requête.

→ RASPBERRY PI 4

La Raspberry Pi 4 est un ordinateur adapté aux besoins de base en matière de programmation, dans mon cas je vais utiliser la Raspberry Pi 4 pour pouvoir communiquer avec le journal lumineux, mon code en C++ sera hébergé dans celle-ci.



Dans un premier temps j'ai installé un système d'exploitation à partir de la carte SD de la Raspberry, j'ai choisi comme système d'exploitation **Raspberry Pi OS (32-BIT)** qui est le système d'exploitation de base.

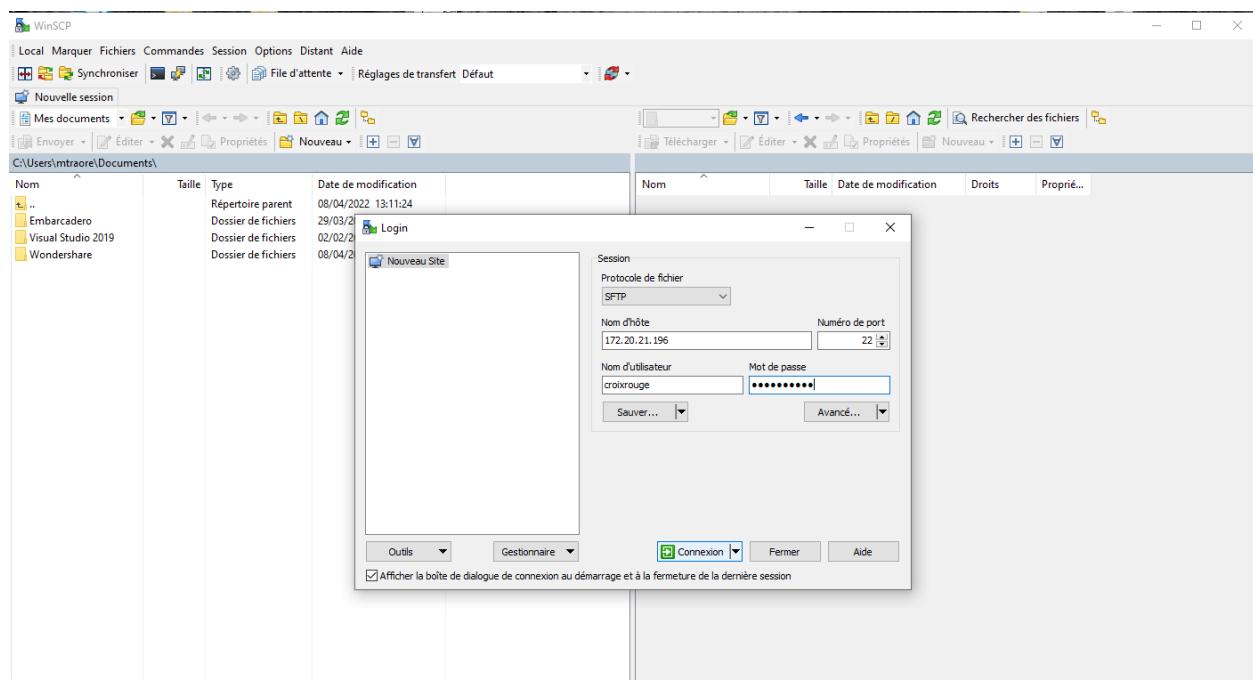


Configuration du réseau de la Raspberry PI 4

Après avoir installé le système d'exploitation dans la Raspberry et d'avoir accédez à l'interface de celle-ci, j'ai configuré ces paramètres réseaux j'y ai modifier :

- **L'adresse IPV4** : le masque étant de 255.255.255.0 j'ai configuré l'adresse IP de sorte à ce qu'elle soit dans le même réseau que celui du lycée.
- **Le routeur** : Paramétrer par le fait d'être la dernière adresse du réseau, et qui permet l'accès à internet.
- **DNS Servers** : Paramétrer par 8.8.8.8 utilisé pour empêcher le blocage des sites Web et des noms de domaine liés au DNS.
- **DNS Search** : Paramétrer par 8.8.4.4 géré par Google dans le but de rendre Internet et le système DNS plus rapides, plus sûrs et plus sécurisés.

J'ai donc maintenant la Raspberry PI 4 qui est installée et configurée en étant sous le même réseau que ma machine et également le journal lumineux.



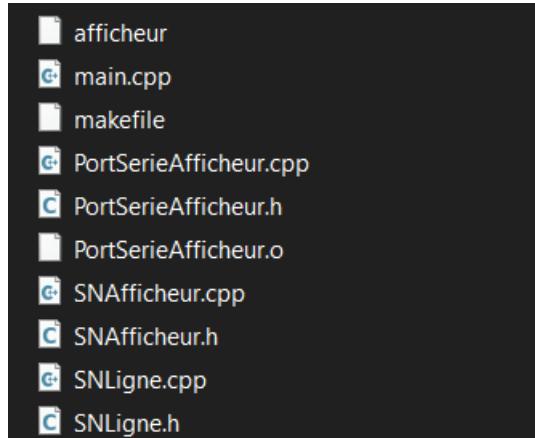
Connexion à distance sur la Raspberry PI 4

Pour pouvoir accéder dans la Raspberry PI 4 j'utilise le logiciel WinSCP qui est un client SFTP (SSH File Transfer Protocol).

L'hôte qui correspond à l'adresse IP de la Raspberry, Le numéro de port, Le nom d'utilisateur et le mot de passe sont requis pour permettre la connexion.

→ JOURNAL LUMINEUX

Après avoir accédez dans la Raspberry PI 4, je vais y créer un répertoire « afficheur » dans le répertoire « Public » ce qui donnera pour chemin **Public/afficheur**.



Fichier contenue dans le répertoire de l'afficheur

Après y avoir intégrer les fichiers nécessaires à l'envoi d'un message sur le journal lumineux je vais utiliser les classes **SNLigne & SNAfficheur** (Fourni), dans le **main.cpp** pour pouvoir envoyer un message (le langage de programmation utilisé pour cette partie sera le C++).

```
#include <iostream>
#include "SNAfficheur.h"
#include "SNLigne.h"
using namespace std;

int main(int argc, char * argv[])
{
    SNAfficheur aff;// Cr ation d'un objet 'aff' de la classe SNAfficheur
    aff.OuvrirPortSerie("/dev/ttyUSB0");// Appel de la m ethode OuvrirPortSerie() avec aff pour ouvrir le port s rie
    SNLigne ligne;
    ligne.ModifierMessage("Test2"); // Envoi d'un message sur le journal lumineux
    aff.EnvoyerLigne(ligne);// Appel de la m ethode EnvoyerTrame() avec aff pour envoyer une trame
    aff.FermerPortSerie();// Fermer le port s rie avec la m ethode FermerPortSerie()
    return 0;
}
```

**main.cpp - Code d'envoi de message sur le journal lumineux **

- Dans le **main.cpp** je vais en premier lieu faire les inclusions des classes nécessaires pour cela je vais faire un **#include** de **SNLigne** et **SNAfficheur**.
- Ensuite je vais créer dans le **int main ()** un objet **aff** de la classe **SNAfficheur** (l'objet va me permettre de faire un appel de méthode contenue dans cette classe).
- Je vais en suite avec l'objet **aff** faire un appel de méthode ici **OuvrirPortSerie()** elle va me permettre d'ouvrir le port série du journal lumineux elle attend pour argument : **nomPortSerie**

```
bool OuvrirPortSerie(string nomPortSerie);
```

- Je vais en suite ensuite créer un objet **ligne** de la classe **SNLigne**.
- Ensuite avec l'objet **ligne** je vais faire un appel de méthode ici **ModifierMessage()** elle va me permettre de définir le message que je souhaite envoyé au journal lumineux elle attend pour argument : **message**

```
inline void ModifierMessage( string message)
```

- J'appelle la méthode **EnvoyerLigne()** avec l'objet **aff** pour envoyer le message elle attend pour argument : **ligne**

```
bool EnvoyerLigne(SNLigne ligne);
```

- Et enfin j'appelle la méthode **FermerPortSerie()** avec l'objet **aff** pour pouvoir fermer le port série du journal lumineux elle n'attend aucun argument.

```
PS C:\Users\mtraore> ssh croixrouge@172.20.21.196
croixrouge@172.20.21.196's password:
Linux raspberrypi 5.15.32-v7l+ #1538 SMP Thu Mar 31 19:39:41 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon May 23 08:23:58 2022 from 172.20.21.48
croixrouge@raspberrypi:~ $ cd Public/afficheur
croixrouge@raspberrypi:~/Public/afficheur $ make
g++ -Wall -c main.cpp
g++ -Wall SNAfficheur.o SNLigne.o PortSerieAfficheur.o main.o SNServeurTcpMonoClient.o -o afficheur
croixrouge@raspberrypi:~/Public/afficheur $ ./afficheur
Test
Trame envoyée : <ID00><L1><PA><FE><MA><WC><FE>Test42<E>
croixrouge@raspberrypi:~/Public/afficheur $
```

Visualisation Réception du message par le journal lumineux

Pour pouvoir tester mon code je vais avec l'éditeur de code **Visual Studio Code** accédez au terminal, puis je vais faire une connexion **SSH** sur la Raspberry PI 4 une fois connecté j'accède à mon répertoire « Public/afficheur » avec **cd Public/afficheur** j'effectue un **make** pour pouvoir compiler le code et un « **. /afficheur** » pour envoyer le message. Le journal Lumineux reçoit bien la trame qui contient le message ici « **Test** ».

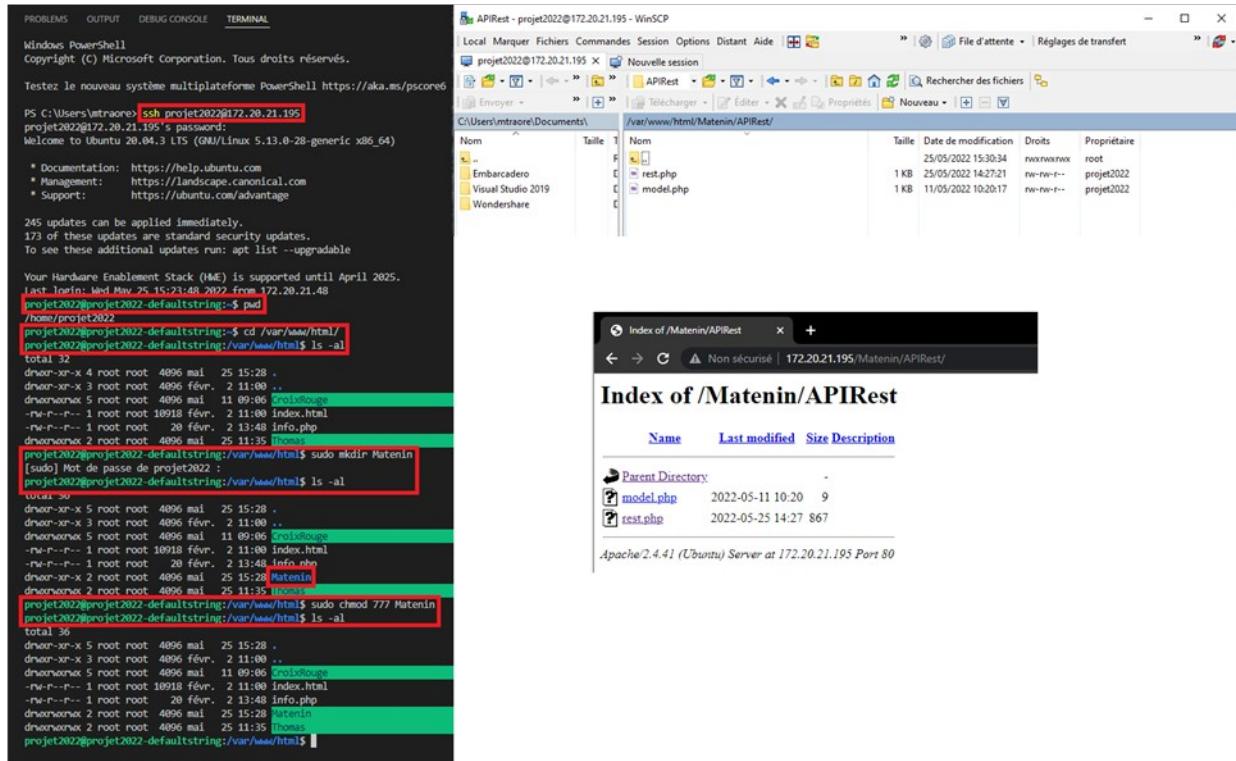


*Réception du message « *Test* » par le journal lumineux*

→ **Fin de test**

À ce stade de mon projet je peux faire un point sur le fonctionnement des différentes parties, mon API Rest est fonctionnelle et je peux récupérer les données critiques, la Raspberry PI 4 est mise en place et communique avec le journal lumineux et enfin je peux envoyer des messages sur le journal lumineux.

Je peux maintenant importer mes codes d'API Rest vers le serveur commun que mon camarade Nethusan SURESHKUMAR (PROJET 3) a mis en place.



The screenshot shows a Windows PowerShell window on the left and a WinSCP interface on the right. In the PowerShell window, the user connects via SSH to a Linux server at 172.20.21.195. They change to the /var/www/html/Matenin/APIRest directory and list files. They then create a new directory named 'Matenin' and change into it. Inside 'Matenin', they list files again, showing 'rest.php' and 'model.php'. Finally, they run 'ls -al' to show the full directory listing, including the hidden '.' and '..' entries. The WinSCP interface shows the local directory 'C:/Users/traore' and the remote directory '/var/www/html/Matenin/APIRest/'. The browser window on the right shows the index of the 'Matenin' directory.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/powershell

PS C:\Users\traore> ssh projet2022@172.20.21.195
projet2022@172.20.21.195's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-28-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

245 updates can be applied immediately.
173 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Wed May 25 15:23:48 2022 from 172.20.21.48
projet2022@projet2022:~$ pwd
/home/projet2022
projet2022@projet2022:~$ cd /var/www/html/
projet2022@projet2022:~$ ls -al
total 32
drwxr-xr-x 4 root root 4096 mai 25 15:28 .
drwxr-xr-x 3 root root 4096 février 2 11:00 ..
drwxrwxr-x 5 root root 4096 mai 11 09:06 CroixRouge
-rw-r--r-- 1 root root 10918 février 2 11:00 index.html
-rw-r--r-- 1 root root 20 février 2 13:48 info.php
drwxrwxr-x 2 root root 4096 mai 25 11:35 Matenin
drwxrwxr-x 2 root root 4096 mai 25 11:35 Thomas
projet2022@projet2022:~$ sudo mkdir Matenin
[sudo] Mot de passe pour projet2022 :
projet2022@projet2022:~$ ls -al
total 36
drwxr-xr-x 5 root root 4096 mai 25 15:28 .
drwxr-xr-x 3 root root 4096 février 2 11:00 ..
drwxrwxr-x 5 root root 4096 mai 11 09:06 CroixRouge
-rw-r--r-- 1 root root 10918 février 2 11:00 index.html
-rw-r--r-- 1 root root 20 février 2 13:48 info.php
drwxrwxr-x 2 root root 4096 mai 25 15:28 Matenin
drwxrwxr-x 2 root root 4096 mai 25 11:35 Thomas
projet2022@projet2022:~$ sudo chmod 777 Matenin
projet2022@projet2022:~$ ls -al
total 36
drwxr-xr-x 5 root root 4096 mai 25 15:28 .
drwxr-xr-x 3 root root 4096 février 2 11:00 ..
drwxrwxr-x 5 root root 4096 mai 11 09:06 CroixRouge
-rw-r--r-- 1 root root 10918 février 2 11:00 index.html
-rw-r--r-- 1 root root 20 février 2 13:48 info.php
drwxrwxr-x 2 root root 4096 mai 25 15:28 Matenin
drwxrwxr-x 2 root root 4096 mai 25 11:35 Thomas
projet2022@projet2022:~$ ls

```

Importation de l'API Rest vers le serveur commun

Toujours dans le terminal je vais maintenant me connecter en **SSH** sur le serveur commun, je vais vérifier que je suis dans le bon répertoire avec la commande **pwd**, puis je vais accéder à l'**index** dans **/var/www/html** je vérifie le contenu avec **ls -al**, puis je vais me créer un répertoire que je nomme « **Matenin** » avec la commande **sudo (accès au droit)** puis **mkdir (pour la création du répertoire)** suivi du nom ici Matenin je vérifie une nouvelle fois avec **ls -al** et je vois mon prénom mais je n'ai pas les droits je vais donc faire un **sudo** une nouvelle fois suivi de **chmod (pour l'attribution de droit) 777 (pour l'autorisation maximum)** et enfin le nom de mon répertoire, le répertoire Matenin à maintenant la possibilité de créer des dossiers sous son répertoire.

Après m'être créé un répertoire sous le serveur commun je vais importer mes fichiers **rest.php** et **model.php** en me connectant avec WinSCP, comme vue ci-dessus j'ai ensuite accès à mon index avec mes fichiers.

```

rest.php
C:\> Users > intrao > AppData > Local > Temp > scp18419 > var > www > html > Matenin > APIRest > rest.php
1 <?php
2
3 // Affichage des erreurs
4 error_reporting(E_ALL);
5 ini_set('display_errors', 1);
6
7 // Requière
8 require ('./model.php');
9
10 // Connexion Base de données
11 $mysqli = new mysqli("localhost","projet","projet","Croix Rouge");
12
13 $req_type=$_SERVER['REQUEST_METHOD'];
14
15 if($req_type ==='GET') {
16
17     // Quantité Minimal
18     $Quantite = "SELECT NomTypeProduit,Quantitemin FROM TypeProduit";
19     $res= $mysqli->query($Quantite);
20     $Quantite=$res->fetch_assoc();
21     echo nl2br(" Réponse : ");
22     echo json_encode($Quantite);
23     echo "<br />";
24
25     // Date de Péremption
26     $peremption = "SELECT nomProduit,datePeremption FROM Produit ORDER BY datePeremption ASC";
27     $resp= $mysqli->query($peremption);
28
29     while($repp=$resp->fetch_assoc()) {
30         $peremption = $repp;
31         echo nl2br(" \n Réponse : ");
32         echo json_encode($peremption);
33     }
34 }
35
36
37 ?>

```

rest.php – Code Récupération des nouvelles données dans le serveur commun

Pour pouvoir récupérer les nouvelles données je vais modifier les arguments **[2]** et **[3]** de la connexion à la base de données avec **new mysqli** en projet pour identifiant et projet pour mot de passe, le dernier argument **[4]** va devenir '**Croix Rouge**'.

Après changement je récupère toutes les données de la base de données classé sous ordre du plus critique au plus ancien.

Method: GET Request URL: http://172.20.21.195/Matenin/APIRest/rest.php
 SEND

Parameters		Headers	Variables
<input type="checkbox"/> <input type="checkbox"/> Toggle source mode <input type="button" value="Insert headers set"/>			
Header name	Header value		
<input type="button" value="ADD HEADER"/> Headers are valid		Headers size: 0 bytes	
200 OK 13.30 ms		<input type="button" value="DETAILS"/>	
<div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Réponse : {"NomTypeProduit": "Riz long", "Quantitemin": "10"}

 Réponse : {"nomProduit": "Lait", "datePeremption": "2022-05-01"}
 Réponse : {"nomProduit": "Riz", "datePeremption": "2022-05-01"}
 Réponse : {"nomProduit": "Oeuf", "datePeremption": "2022-05-06"}
 Réponse : {"nomProduit": "Oeuf", "datePeremption": "2022-05-06"}
 Réponse : {"nomProduit": "Ble", "datePeremption": "2022-05-23"}
 Réponse : {"nomProduit": "Tomate", "datePeremption": "2022-07-28"}
 Réponse : {"nomProduit": "Pomme de terre", "datePeremption": "2022-07-31"}
 Réponse : {"nomProduit": "Pomme de terre", "datePeremption": "2022-07-31"}
 Réponse : {"nomProduit": "Pomme de terre", "datePeremption": "2022-07-31"}
 Réponse : {"nomProduit": "Riz", "datePeremption": "2022-09-01"}
 Réponse : {"nomProduit": "Tomate", "datePeremption": "2022-09-01"}
 Réponse : {"nomProduit": "Pomme", "datePeremption": "2022-11-22"}
 Réponse : {"nomProduit": "Lait", "datePeremption": "2022-12-25"}
 Réponse : {"nomProduit": "Pates", "datePeremption": "2109-09-05"}
 Réponse : {"nomProduit": "Pates", "datePeremption": "2109-09-05"}
 Réponse : {"nomProduit": "Farine", "datePeremption": "2222-03-22"}
 Réponse : {"nomProduit": "Farine", "datePeremption": "2222-03-22"}
 Réponse : {"nomProduit": "Eau", "datePeremption": "2222-03-22"}
 </div>			

Visualisation de la requête GET avec un REST Client

En utilisant le même REST Client (Extension Advanced REST Client) sous chrome, je peux vérifier si ma requête GET est prise en compte et si les

données sont reçues en choisissant la méthode GET et entrant l'URL qui donne accès aux données.

En indiquant l'hôte du serveur (172.20.21.195) suivi du chemin d'accès aux données

Après observation je reçois bien mes données confirmées par la réponse '**200 OK**' qui en HTTP indique la réussite d'une requête.

```

main.cpp x rest.php
C: > Users > mtraore > AppData > Local > Temp > scp17649 > home > croixrouge > Public > afficheur > main.cpp
22
23     SNClientTCP monClient;
24
25     char reponseHTTP[1500];
26
27 // while(true) {
28 //     monClient.SeConnecterAUnServeur("172.20.21.195",80);
29 //     char requeteHTTP[] = "GET /Matenin/APIRest/rest.php HTTP/1.1\r\nHost: 172.20.21.195\r\nConnection: keep-alive\r\n\r\n";
30 //     monClient.Envoyer(requeteHTTP, strlen(requeteHTTP));
31 //     monClient.Recevoir(reponseHTTP,1500);
32
33     cout<<reponseHTTP<<endl;
34
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL
Last login: Mon May 30 09:27:00 2022 from 172.20.21.48
croixrouge@raspberrypi:~ $ cd Publi
-bash: cd: Publi: Aucun fichier ou dossier de ce type
croixrouge@raspberrypi:~ $ cd Public
croixrouge@raspberrypi:~/Public $ cd afficheur
croixrouge@raspberrypi:~/Public/afficheur $ make
g++ -Wall -c main.cpp
g++ -Wall SNAfficheur.o SNSigne.o PortSerieAfficheur.o main.o SNServeurTcpMonoClient.o SNClientTCP.o -o afficheur
croixrouge@raspberrypi:~/Public/afficheur $ ./afficheur
HTTP/1.1 200 OK
Date: Mon, 30 May 2022 07:38:40 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 352
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
Réponse : [{"NomTypeProduit":"Riz long","Quantitemin":"10"}]<br /> <br />
Réponse : [{"nomProduit":"Lait","datePeremption":"2022-06-30"}]<br />
Réponse : [{"nomProduit":"Tomate","datePeremption":"2022-08-18"}]<br />
Réponse : [{"nomProduit":"Riz","datePeremption":"2036-10-23"}]<br />
Réponse : [{"nomProduit":"Pates","datePeremption":"2036-10-30"}]
croixrouge@raspberrypi:~/Public/afficheur $ 

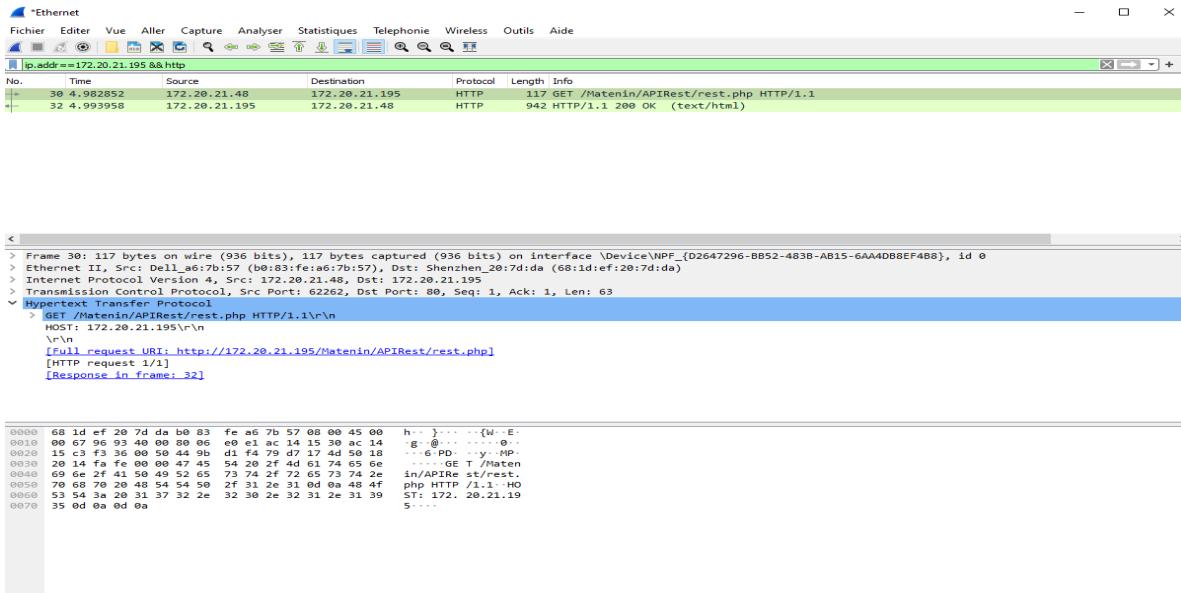
```

main.cpp – Récupération des données en C++

Pour pouvoir envoyer les données critiques au journal lumineux il faut que je puisse dans un premier temps extraire les données JSON en C++ pour pouvoir par la suite envoyer au journal lumineux les variables.

- Dans le **main.cpp** je vais créer un objet **monClient** de la classe **SNClientTCP** (Fourni)
- Je vais ensuite créer un tableau de 1500 caractère nommé **reponseHTTP**
- Puis je vais avec l'objet **monClient** faire un appel de méthode de la classe **SNClientTCP** ici **SeConnecterAUnServeur()** cela va me permettre de me connecter au serveur commun.
- Je vais ensuite créer une variable nommée « **requeteHTTP** » qui va contenir la requête GET qui permet d'accéder aux données.
- En utilisant l'objet **monClient** je vais faire un appel de la méthode **Envoyer()** et je vais mettre en premier argument la **requeteHTTP** et en deuxième arguments le renvoi de la longueur de la chaîne.
- Puis je vais recevoir la **réponseHTTP** par l'appel de méthode **Recevoir()** de la classe **SNClientTCP**, elle va me permettre de recevoir la réponse au maximum de 1500 caractères.

Pour vérifier mon code je vais dans le terminal accéder à l'afficheur par **CD Public/afficheur** puis je vais faire un **make** suivi d'un **./afficheur**, et je peut voir que dans la console je récupère les données du **rest.php** (L'administrateur a réduit le nombre de données entre temps).



Visualisation de la réception avec WireShark

En entrant l'adresse IP du serveur commun sur WireShark et en faisant une recherche sur des données HTTP, je retrouve ma requête GET.

```
// Création d'un objet de type string
string reponse(reponseHTTP);
cout<<reponse<<endl;

// Commencement au premier caractère de la réponse HTTP
unsigned int positionDepart = 0;

// Extraire NomProduit

string produitATrouver = "nomProduit\":\"";
unsigned int positionTrouvee = reponse.find(produitATrouver, positionDepart);
unsigned int positionDebutTexte = positionTrouvee + produitATrouver.length();
cout << "POS DEBUT : " << positionDebutTexte << endl;
cout << "CHAR : " << reponse[positionDebutTexte] << endl;

unsigned int positionFinTexte = reponse.find("\",positionDebutTexte);
cout << "POS FIN : " << positionFinTexte << endl;

string texteExtrait = reponse.substr(positionDebutTexte, positionFinTexte-positionDebutTexte);
cout << "NOM PRODUIT: " << texteExtrait << endl;

// Extraire Date de Péremption

string peremptionATrouver = "datePeremption\":\"";
unsigned int positionTrouvee2 = reponse.find(peremptionATrouver, positionDepart);
unsigned int positionDebutTexte2 = positionTrouvee2 + peremptionATrouver.length();
cout << "POS DEBUT : " << positionDebutTexte2 << endl;
cout << "CHAR : " << reponse[positionDebutTexte2] << endl;

unsigned int positionFinTexte2 = reponse.find("\",positionDebutTexte2);
cout << "POS FIN : " << positionFinTexte2 << endl;

string texteExtrait2 = reponse.substr(positionDebutTexte2, positionFinTexte2-positionDebutTexte2);
cout << "DATE DE PEREMPTION : " << texteExtrait2 << endl;
```

main.cpp – Code Extraction des données en C++

Maintenant qu'il est possible de récupérer la réponse HTTP en C++ il est temps d'extraire les données critiques qui arrive à date de péremption et les envoyer sur le journal lumineux.

- Dans un premier temps je créer une variable **reponse** ou je stock la **reponseHTTP** le type sera un **string** pour pouvoir utiliser les fonctions de **string**.
- Je déclare une variable dite **positionDepart** qui désigne la première lettre de la réponse HTTP (le H de HTTP).
- Je déclare la variable **produitATrouver** qui va contenir le mot clé **nomProduit** délimiter par un ‘ : ’ à rechercher.
- Je déclare la variable **positionTrouvee** qui va pointer la **reponse** et utilisé la fonction **find** de string pour trouver **produitATrouver** à partir de la **positionDepart**.
- Je déclare la variable **positionDebutTexte** qui va contenir la variable **positionTrouvee** additionné à **produitATrouver** qui va renvoyer la longueur de la chaîne avec **length()**.
- Je fais ensuite deux couts une pour la position du premier caractère avec la variable **positionDebutTexte** et l'autre pour le caractère correspondant avec la variable **reponse** et de **positionDebutTexte**.
- Après avoir trouver la position du début il nous faut celle de la fin pour cela on créer la variable **positionFinTexte** et on pointe vers la **reponse** pour trouver avec **find** la délimitation et **positionDebutTexte** puis un cout de la **positionFinTexte**
- Et pour finir on créer la variable **texteExtrait** qui va contenir la **reponse** cette fois ci avec l'utilisation de **substr** qui va renvoyer une chaîne de caractère entre la **positionDebutTexte** et la **positionFinTexte-positionDebutTexte**, et enfin un cout de **textExtrait** qui correspond au nom d'un produit
- Le même principe pour la date de péremption (Je rajoute un '2' à chaque variable identique)

```

croixrouge@raspberrypi:~/Public/afficheur $ make
g++ -Wall -c main.cpp
g++ -Wall SNAfficheur.o SNLigne.o PortSerieAfficheur.o main.o SNServeurTcpMonoClient.o SNClientTCP.o -o afficheur
croixrouge@raspberrypi:~/Public/afficheur $ ./afficheur
HTTP/1.1 200 OK
Date: Mon, 30 May 2022 07:47:20 GMT
Server: Apache/2.4.41 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 492
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

Réponse : {"NomTypeProduit":"Riz long","Quantitemin":"10"}<br /> <br />
Réponse : {"nomProduit":"Tomate","datePeremption":"2022-08-18"} <br />
Réponse : {"nomProduit":"Lait","datePeremption":"2023-01-01"} <br />
Réponse : {"nomProduit":"Ble","datePeremption":"2031-06-30"} <br />
Réponse : {"nomProduit":"Eau","datePeremption":"2035-12-11"} <br />
Réponse : {"nomProduit":"Riz","datePeremption":"2036-10-23"} <br />
Réponse : {"nomProduit":"Pates","datePeremption":"2036-10-30"} <br />

POS DEBUT : 329
CHAR : T
POS FIN : 335
NOM PRODUIT: Tomate
POS DEBUT : 355
CHAR : 2
POS FIN : 365
DATE DE PEREMPTION : 2022-08-18
croixrouge@raspberrypi:~/Public/afficheur $ 
```

Visualisation de l'extraction des données

En compilant une nouvelle fois le programme avec **make** et en lançant l'afficheur avec **./afficheur** je peux extraire le nom d'un produit et sa date de péremption en fonction de sa position de fin et de sa position du début.

```

// Envoi des données au journal lumineux

SNAfficheur aff;// Création d'un objet 'aff' de la classe SNAfficheur

aff.OuvrirPortSerie("/dev/ttyUSB0");// Appel de la méthode OuvrirPortSerie() avec aff pour ouvrir le port série

SNLigne ligne;// Création d'un objet 'ligne' de la classe SNLigne

ligne.ModifierMessage(texteExtrait +" Peremption le : "+ texteExtrait2); // Définition du message (Concaténation)

aff.EnvoyerLigne(ligne);// Envoi du message
aff.FermerPortSerie();// Fermeture du port série

```

```

POS DEBUT : 329
CHAR : T
POS FIN : 335
NOM PRODUIT: Tomate
POS DEBUT : 355
CHAR : 2
POS FIN : 365
DATE DE PEREMPTION : 2022-08-18
Trame envoyée : <ID00><L1><PA><FE><MA><WC><FE>Tomate Peremption le : 2022-08-1841<E>
croixrouge@raspberrypi:~/Public/afficheur $ 
```

main.cpp – Code envoi des données critiques sur le journal lumineux

En modifiant les arguments de la méthode **ModifierMessage()** en **texteExtrait** qui correspond à le nom d'un produit critique et **texteExtrait2** qui correspond à la date d'un produit critique je peut envoyé une alerte sur le journal lumineux, comme vue dans la console le journal lumineux a reçu « Tomate Peremption le : 2022-08-18 ».



Réception d'un produit à date de péremption



CONCLUSION

Pour conclure ce projet ma permis d'apprendre énormément de chose sur la programmation mais également sur les protocoles à suivre pour mener à bien un projet, j'ai pu vivre une expérience semblable à la vie d'un développeur dans le monde de l'entreprise, le fait de travailler en équipe, d'avoir des dates d'échéance et un client ma permis de développer un esprit « entreprise », le fait d'avoir également pu suivre l'avancement de notre projet tout au long des 5 mois est très gratifiant, j'ai pu renforcé mes connaissance en programmation dans les langages C++ et PHP mais également dans la conception d'un projet (diagramme de séquence, diagramme UML, diagramme de déploiement, architecture d'une base de données).

Ayant pour ambition de devenir développeur informatique ce projet ma conforté en cette envie.



IV. Partie étudiant 2 (Thomas VALLET)

Introduction

L'étudiant n°2 (Thomas VALLET) s'occupe du développement du site web – partie utilisateur, qui est une fonction à développer.

Le but de cette partie est de créer un site web pour la croix-rouge, où on doit s'occuper des produits appelés par des QR Codes dans un stock.

Les QR Codes sont intégrés dans cette partie du projet. Ils sont générés dans la base de données, lorsqu'on en scanne un, le nom du produit apparaît. Des informations du produit à remplir sont présentes sur le site comme la quantité, la date de péremption et la catégorie de produit. Dès que ces informations sont remplies, on peut ajouter les produits dans un stock (la base de données) avec des quantités différentes. Puis il y a la possibilité de les retirer, en diminuant le nombre de quantité si on le souhaite.

Toute cette partie se fait avec du code en langage web. Mais pour que tout fonctionne, l'utilisateur doit être en lien avec l'administrateur. C'est-à-dire se connecter à la même adresse IP (mini PC), dans le même répertoire d'échange.

Tâches personnelles

Voici les tâches personnelles de l'étudiant n°2 (Développement du site web – partie utilisateur) pour réaliser une grande partie du projet :

Thomas VALLET

Développement du site web – partie utilisateur

1. Dessiner une esquisse de site ergonomique, simple d'utilisation et complet
2. Mise en page du site web
3. Gestion de l'approvisionnement d'un produit dans le stock à partir de son QR Code
4. Gestion de la sortie d'un produit du stock à partir de son QR Code
5. Gestion de l'affichage des produits périmés
6. Gestion de l'affichage des produits dont la quantité minimale est atteinte

Logiciel de codage

Pour réaliser le site web pour la croix-rouge, il faut faire du codage avec plusieurs langages web dans un logiciel qui le pratique.

Le logiciel de programmation web le plus simple et le plus adapté pour développer un site web s'appelle Notepad++. Il permet de rendre le site plus simple à manier.

Notepad++



Notepad++ est un éditeur de texte libre générique, fonctionnant sous Windows, codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, HTML, PHP, JavaScript, makefile, art ASCII, doxygen, .bat, MS fichier ini, ASP, Visual Basic/VBScript, SQL, Objective-C, CSS, Pascal, Perl, Python, R, MATLAB, Lua, TCL, Assembleur, Ruby, Lisp, Scheme, Properties, Diff, Smalltalk, PostScript et VHDL ainsi que pour tout autre langage informatique, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque.

Langages de programmation

Une fois dans le logiciel de codage Notepad++, il faut utiliser les langages qui sont faits pour créer un site web pour la croix-rouge.

Les langages à utiliser pour le projet sont : HTML, CSS, JavaScript et PHP.

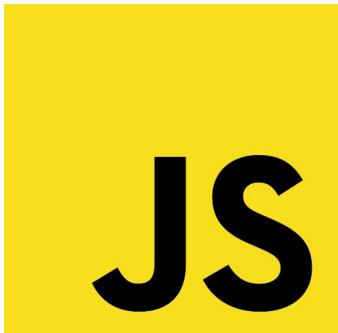
HTML



CSS



JavaScript



PHP



→ HTML

Introduction

Le HyperTextMarkupLanguage, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web.

Pourquoi ce langage ?

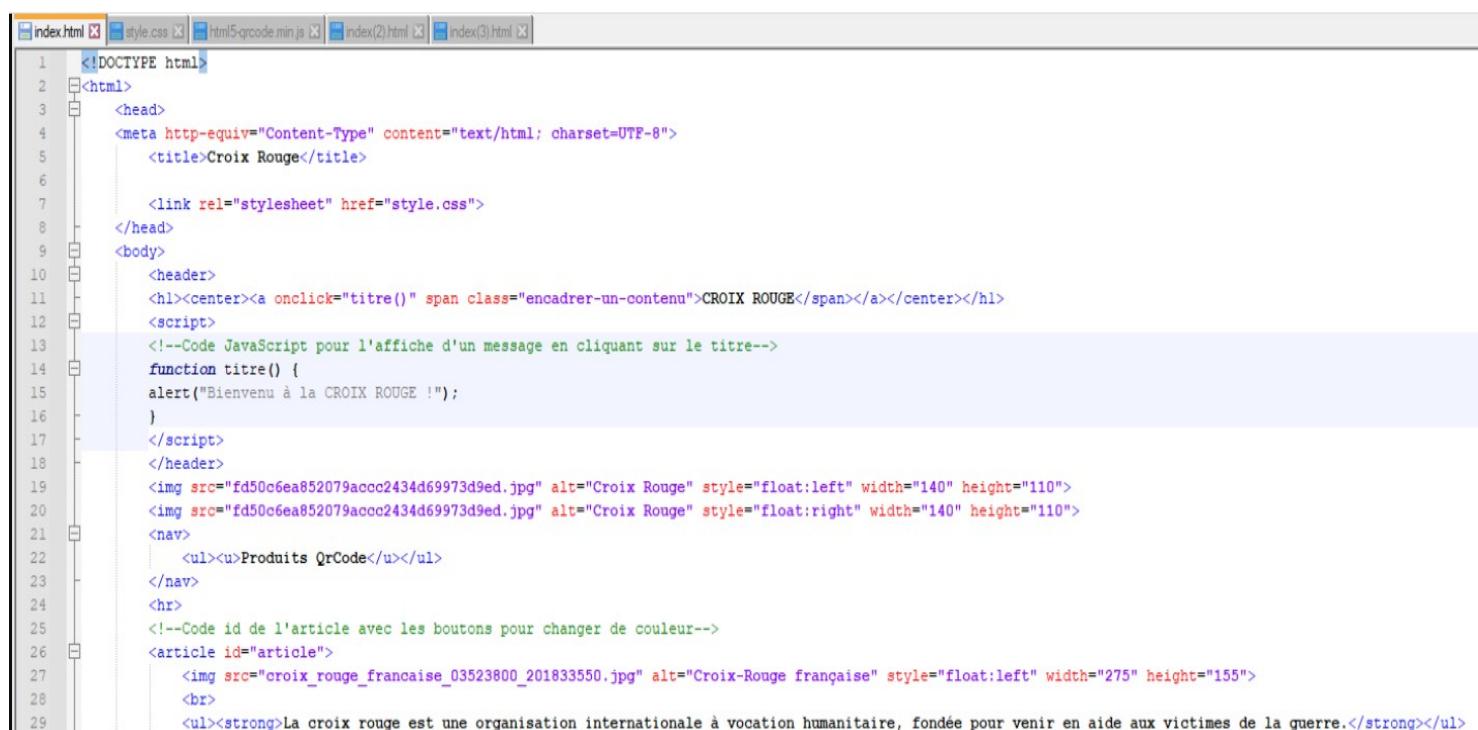
Ce langage a été choisi, car c'est celui qui permet de élaborer la partie visuelle du site une fois qu'un utilisateur entre dedans.

Il permet aussi dans le logiciel Notepad++ d'insérer des images, des textes écrits, des boutons... et surtout d'autres langages web (CSS, JavaScript, PHP).

Aperçus des codes HTML

Voici plusieurs aperçus des codes HTML dans le logiciel Notepad++ :

Partie du code HTML (site Croix-Rouge)



```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5       <title>Croix Rouge</title>
6
7       <link rel="stylesheet" href="style.css">
8   </head>
9   <body>
10    <header>
11      <h1><center><a onclick="titre()" span class="encadrer-un-contenu">CROIX ROUGE</span></a></center></h1>
12      <script>
13        <!--Code JavaScript pour l'affiche d'un message en cliquant sur le titre-->
14        function titre() {
15          alert("Bienvenu à la CROIX ROUGE !");
16        }
17      </script>
18    </header>
19    
20    
21    <nav>
22      <ul><u>Produits QRCode</u></ul>
23    </nav>
24    <hr>
25    <!--Code id de l'article avec les boutons pour changer de couleur-->
26    <article id="article">
27      
28      <br>
29      <ul><strong>La croix rouge est une organisation internationale à vocation humanitaire, fondée pour venir en aide aux victimes de la guerre.</strong></ul>

```

Partie du code HTML (Ajouter un produit)

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5              <title>Croix Rouge</title>
6
7              <link rel="stylesheet" href="style.css">
8      </head>
9      <body>
10         <script src="html5-qrcode.min.js"></script>
11         <style>
12             .result{
13                 background-color: green;
14                 color:#fff;
15                 padding:20px;
16             }
17             .row{
18                 display:flex;
19             }
20         </style>
21         <form method="POST" action="ajouter-produit.php">
22             <div class="row">
23                 <div class="col">
24                     <div style="width:500px;" id="reader"></div>
25                 </div>
26                 <div class="col" style="padding:30px;">
27                     <h4>SCAN RESULT</h4>
28                     <div id="result">Result Here</div>
29                     <br><br>
30                     <button>Ajouter</button>
31                     <br><br>
32                     <div id="result">
33                         <label for="option_nom">Nom du produit : </label>
34                         <input type="text" id="option_nom" name="NomProduit" required><br />
35                         <label for="option_quantite">Quantité</label>
36                         <input type="number" id="option_quantite" name="Quantite" min="0" max="1000" required><br />
37                         Date de Peremption : <input type="date" name="datePeremption"><br />
38                         <label for="option_quantite">Catégorie</label>
39                         <select name="produit" id="produit_option" required>
40                             <option value="">-----</option>
41                             <option value="Formation">Formation</option>
42                             <option value="Secours">Secours</option>
43                             <option value="Hygiène">Hygiène</option>
44                             <option value="Alimentation">Alimentation</option>
45                         </div>
46                     </div>
47                 </div>

```

Partie du code HTML (Retirer un produit)

```

6         <link rel="stylesheet" href="style.css">
7     </head>
8     <body>
9     <header>
10    <center><h1><u>PRODUITS AJOUTES A RETIRER</u></h1></center>
11    </header>
12    <!--Boutons pour retirer le nom d'un produit-->
13    <form action="ajouter-produit.php"><table style="width:100%">
14    <tr>
15        <td><h3 id='heading'>Produit : Eau</h3>
16        <br>
17        <label for="option_quantite">Quantité disponible</label><br>
18        <input type="number" id="option_quantite" name="Quantite" min="0" max="100" value="100" required><br />
19        <br><button onclick='remove()'>Retirer</button></td>
20    <td><h3 id='heading'>Produit : Oeuf</h3>
21    <br>
22    <label for="option_quantite">Quantité disponible</label><br>
23    <input type="number" id="option_quantite" name="Quantite" min="0" max="70" value="70" required><br />
24    <br><button onclick='remove()'>Retirer</button></td>
25    <td><h3 id='heading'>Produit : Pomme</h3>
26    <br>
27    <label for="option_quantite">Quantité disponible</label><br>
28    <input type="number" id="option_quantite" name="Quantite" min="0" max="45" value="45" required><br />
29    <br><button onclick='remove()'>Retirer</button></td>
30    <td><h3 id='heading'>Produit : Lait</h3>
31    <br>
32    <label for="option_quantite">Quantité disponible</label><br>
33    <input type="number" id="option_quantite" name="Quantite" min="0" max="30" value="30" required><br />
34    <br><button onclick='remove()'>Retirer</button></td>
35    <td><h3 id='heading'>Produit : Tomate</h3>
36    <br>
37    <label for="option_quantite">Quantité disponible</label><br>
38    <input type="number" id="option_quantite" name="Quantite" min="0" max="17" value="17" required><br />
39    <br><button onclick='remove()'>Retirer</button></td>
40    <td><h3 id='heading'>Produit : Pates</h3>
41    <br>
42    <label for="option_quantite">Quantité disponible</label><br>
43    <input type="number" id="option_quantite" name="Quantite" min="0" max="50" value="50" required><br />
44    <br><button onclick='remove()'>Retirer</button></td>
45    <td><h3 id='heading'>Produit : Riz</h3>
46    <br>
47    <label for="option_quantite">Quantité disponible</label><br>
48    <input type="number" id="option_quantite" name="Quantite" min="0" max="23" value="23" required><br />
49    <br><button onclick='remove()'>Retirer</button></td>
50    <td><h3 id='heading'>Produit : Pomme de terre</h3>
51    <br>
52    <label for="option_quantite">Quantité disponible</label><br>
53    <input type="number" id="option_quantite" name="Quantite" min="36" max="36" value="36" required><br />
54    <br><button onclick='remove()'>Retirer</button></td>
55    </tr>
56    </table></form>
57    <script>
58        <!--Code JavaScript pour retirer le nom d'un produit-->
59        function remove()
60        {
61            var element = document.getElementById("heading");
62            element.remove();
63        }
64    </script>
65    </body>
66

```

→ CSS

Introduction

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.

Pourquoi ce langage ?

Ce langage a été choisi, car c'est le seul qui permet de bien mettre en forme, d'appliquer un design au site de la croix-rouge. Il est en lien avec le langage HTML, et il peut être impliquer dans ce code.

Aperçus des codes CSS

Voici plusieurs aperçus des codes CSS dans le logiciel Notepad++ :

Partie du code CSS (site Croix-Rouge) Code CSS (Ajouter un produit)

```

1 header {
2     background-color: red;
3 }
4
5 h1 {
6     margin: 0;
7     padding: 13px;
8     font-size: 80pt;
9     color: white;
10    font-family: Lucida Console;
11 }
12
13 nav {
14     font-size: 29pt;
15     text-align: center;
16     font-family: Lucida Console;
17     color: red;
18 }
19
20 hr {
21     border: none;
22     border-top: 3px double #333;
23     color: #333;
24     overflow: visible;
25     text-align: center
26 }
27
28 article {
29     font-size: 13pt;
30     text-align: center;
31     color: black;
32     font-family: Lucida Console;
33 }
34
35 footer {
36     display: flex;
37     justify-content: center;
38     background-color: red;
39     font-size: 17.5pt;
40     padding: 0;
41     font-family: Lucida Console;
42 }
43
44 button {
45     cursor: pointer;
46     margin: auto;
47     border-radius: 12px;
48 }
49
50 p {
51     font-size: 22pt;
52     color: white;
53 }
54
55 li {
56     font-size: 15pt;
57 }

```

```

2 <style>
3 .result{
4     background-color: green;
5     color:#fff;
6     padding:20px;
7 }
8 .row{
9     display:flex;
10 }
11 </style>

```

→ JavaScript

Introduction

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML, et CSS, JavaScript est au cœur des langages utilisés par les développeurs web.

Pourquoi ce langage ?

Ce langage a été choisi, car il permet de rendre des détails du site plus poussés. Dans le site web, on a la possibilité d'afficher des messages en cliquant sur des textes, de changer de couleur une petite partie de la page en cliquant sur un bouton (ex : Bleu fond)...

Aperçus des codes JavaScript

Voici plusieurs aperçus des codes JavaScript dans le logiciel Notepad++ :

Partie du code JavaScript (site Croix-Rouge)

```

54      <!--Code JavaScript pour changer la couleur de la partie "article" du site-->
55      function article(coul){document.getElementById("article").style.backgroundColor=coul;}
56      </script>
57
58      </ul>
59      <ul style="list-style-type:none;">
60          <p><strong>Auteurs</strong></p>
61          <li>Thomas VALLET</li>
62          <li>Matenin TRAORE</li>
63          <li>Nethusan SURESHKUMAR</li>
64      </ul>
65      &nbsp;&nbsp;&nbsp;
66      <ul style="list-style-type:none;">
67          <p><strong>Catégories</strong></p>
68          <li><a onclick="myFunction()">Formation</li>
69          <li><a onclick="myFunction1()">Secours</li>
70          <li><a onclick="myFunction2()">Hygiène</li>
71          <li><a onclick="myFunction3()">Alimentation</li>
72      </ul>
73      <script>
74          <!--Code JavaScript pour l'affiche d'un message en cliquant sur les catégories-->
75          function myFunction() {
76              alert("Formation");
77          }
78          </script>
79          function myFunction1() {
80              alert("Secours");
81          }
82          </script>
83          function myFunction2() {
84              alert("Hygiène");
85          }
86          </script>
87          function myFunction3() {
88              alert("Alimentation");
89          }
90      </script>
91
92

```

Code JavaScript (*Ajouter un produit*)

```

39     function onScanSuccess(qrCodeMessage) {
40         document.getElementById('result').innerHTML = '<span class="result">' + qrCodeMessage + '</span>';
41         document.getElementById("option_nom").value=qrCodeMessage;
42
43         var xhttp = new XMLHttpRequest();
44         xhttp.onreadystatechange = function() {
45             if (this.readyState == 4 && this.status == 200) {
46                 // Typical action to be performed when the document is ready:
47                 //document.getElementById("option_quantite").innerHTML = xhttp.responseText;
48
49                 const obj = JSON.parse(xhttp.responseText);
50                 document.getElementById("option_quantite").value=obj[0].Quantite;
51
52             }
53         };
54         var url="http://172.20.21.195/Thomas/rest.php/produit/"+qrCodeMessage;
55         xhttp.open("GET", url , true);
56         xhttp.send();
57     }
58     function onScanError(errorMessage) {
59         //handle scan error
60     }
61     var html5QrcodeScanner = new Html5QrcodeScanner(
62         "reader", { fps: 10, qrbox: 250 });
63     html5QrcodeScanner.render(onScanSuccess, onScanError);
64 </script>
65 </form>
```

Partie du code JavaScript - QR Code (*Ajouter un produit*)

```

1 !function(t,e){"object"==typeof exports&&"undefined"!=typeof module?e(exports):"function"==typeof define&&define.amd?defi:
2
3 /** Html5QrcodeScanner ***/
4 "use strict";function _classCallCheck(a,b){if(!(a instanceof b))throw new TypeError("Cannot call a class as a function")};
```

→ PHP

Introduction

HypertextPreprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

Pourquoi ce langage ?

Ce langage a été choisi, car il permet de réaliser des requêtes depuis la base de données, où se trouve l'administrateur.

Grâce aux tables et aux champs enregistrés dans la base de données, le site peut prendre en compte les informations remplies. Ces informations doivent être affichées une fois que l'action est faite.

Aperçus du code PHP

Voici plusieurs aperçus du code PHP dans le logiciel Notepad++ :

Code PHP (Ajouter un produit)

```

ajouter-produit.php model.php
1 <?php
2 include_once('./model.php');
3 print_r($_POST);
4 $id=ConnectDB();
5 $requete = "INSERT INTO Produit(nomProduit, Quantite, datePeremption, idTypeProduit) VALUES('Eau', '94', '2222-03-22', '1')";
6 executerRequete($id, $requete, array());
7 /*  $recupQuantité = $_POST["option_quantite"];
8 $recupProduit = $_POST["produit"];
9 $mysqli = new mysqli("localhost","projet","projet","Croix Rouge");
10 $req="Select * from joueurs where qrcode=$reader and quantite=$option_quantite" ;
11 $reponseAPI=executerRequete($id,$req,array());
12 foreach($_GET as $cle =>$valeur){
13     $$cle=$valeur
14 }
15 */
16 ?>

```

Code app.ini

app - Bloc-notes

Fichier Edition Format Affichage Aide (Fichier app.ini pour connexion à la base de données)

```
host=localhost;
database=Croix Rouge;
user=projet;
password=projet;
```

Partie du code PHP – rest.php

```

1  <?php
2
3      include_once('./model.php');
4      $id=ConnectDB();
5
6      $req_type=$_SERVER['REQUEST_METHOD'];
7      $req_path=$_SERVER['PATH_INFO'];
8
9      $req_data=explode('/',$req_path);
10
11     $header=apache_request_headers();
12
13
14 //////////////////////////////////////////////////////////////////// Requête HTTP : GET ///////////////////////////////
15
16 if($req_type==='GET'){
17     $reponseAPI=array();
18     if($req_data['1']=='produit'){
19         // Je cherche plus d'information sur les produits
20         if(isset($req_data['2'])){
21             $req="select Quantite from Produit where nomProduit=?";
22             $reponseAPI=executerRequete($id,$req,array($req_data['2']));
23         }
24     }
25     echo(json_encode($reponseAPI));
26 }
27
28 //////////////////////////////////////////////////////////////////// Requête HTTP : POST ///////////////////////////////
29 if($req_type==='POST'){
30     if($req_data['1']=='vol'){
31         // Insérer un vol
32         $donneesVolJSON=json_decode(file_get_contents("php://input"),true);
33         $donneesVolAssoc=$donneesVolJSON['donneesVol'];
34
35         // J'extrait les données du json ==> $nom,$time,$etats et $commandes
36         foreach($donneesVolAssoc as $c=>$v)
37             $$c=$v;
38
39         // Je cherche ou j'insère Id Produit
40         $reqProduit="select idProduit from Produit where nomProduit=?";
41         $res=executerRequete($idProduit,$reqProduit,array($idProduit));
42
43         if(empty($res)){
44             $reqProduit="INSERT into Produit (id) values(?)";
45             $rep= preparerRequete($idProduit,$reqProduit);
46             executerRequetePrepare($rep,array($idProduit));
47             $idProduit=re recupererLeDernierIdInserer($id);
48
49         }else
50             $idProduit=$res[0]['idProduit'];
}

```

Logiciel de connexion du mini PC

Pour réussir à fonctionner le site web de la croix-rouge, il faut être en lien avec l'administrateur. L'administrateur pratique ses tâches personnelles sur un mini PC. Son mini PC contient une adresse IP (172.20.21.195).

Le logiciel qu'il utilise s'appelle WinSCP. Pour être en lien avec l'administrateur, il est nécessaire de faire des commandes dans l'invité de commandes.

Connexion projet2022@projet2022-defaultstring: / var/www/html

```
c:\> projet2022@projet2022-defaultstring: /var/www/html
Microsoft Windows [version 10.0.19044.1645]
(c) Microsoft Corporation. Tous droits réservés.

P:\>ssh projet2022@172.20.21.195
The authenticity of host '172.20.21.195 (172.20.21.195)' can't be established.
ECDSA key fingerprint is SHA256:1Z4U0WnNcCrY08Ws8ALIf88pZS/L4RNtzbM4zzC9JBo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.20.21.195' (ECDSA) to the list of known hosts.
projet2022@172.20.21.195's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.13.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

245 updates can be applied immediately.
173 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Mar 14 10:33:52 2022 from 172.20.21.55
projet2022@projet2022-defaultstring:~$ pwd
/home/projet2022
projet2022@projet2022-defaultstring:~$ cd /var/www/html/
projet2022@projet2022-defaultstring:/var/www/html$ ls -al
total 28
drwxr-xr-x 3 root root 4096 mars 11 13:47 .
drwxr-xr-x 3 root root 4096 févr. 2 11:00 ..
drwxrwxrwx 5 root root 4096 mai 11 09:06 CroixRouge
-rw-r--r-- 1 root root 10918 févr. 2 11:00 index.html
-rw-r--r-- 1 root root 20 févr. 2 13:48 info.php
projet2022@projet2022-defaultstring:/var/www/html$ mkdir Thomas
mkdir: impossible de créer le répertoire «Thomas»: Permission non accordée
projet2022@projet2022-defaultstring:/var/www/html$ sudo mkdir Thomas
[sudo] Mot de passe pour projet2022 :
projet2022@projet2022-defaultstring:/var/www/html$ ls -al
total 32
drwxr-xr-x 4 root root 4096 mai 23 09:32 .
drwxr-xr-x 3 root root 4096 févr. 2 11:00 ..
drwxrwxrwx 5 root root 4096 mai 11 09:06 CroixRouge
-rw-r--r-- 1 root root 10918 févr. 2 11:00 index.html
-rw-r--r-- 1 root root 20 févr. 2 13:48 info.php
drwxr-xr-x 2 root root 4096 mai 23 09:32 Thomas
projet2022@projet2022-defaultstring:/var/www/html$ sudo chmod 777 Thomas
projet2022@projet2022-defaultstring:/var/www/html$ ls -al
total 32
drwxr-xr-x 4 root root 4096 mai 23 09:32 .
drwxr-xr-x 3 root root 4096 févr. 2 11:00 ..
drwxrwxrwx 5 root root 4096 mai 11 09:06 CroixRouge
-rw-r--r-- 1 root root 10918 févr. 2 11:00 index.html
-rw-r--r-- 1 root root 20 févr. 2 13:48 info.php
drwxrwxrwx 2 root root 4096 mai 23 09:32 Thomas
projet2022@projet2022-defaultstring:/var/www/html$
```



Répertoire d'échange html – projet2022@172.20.21.195 – WinSCP

/var/www/html/					
Nom	Taille	Date de modification	Droits	Proprié...	
..		02/02/2022 11:00:05	rwxr-xr-x	root	
CroixRouge		11/05/2022 09:06:46	rwxrwxrwx	root	
Thomas		23/05/2022 09:32:14	rwxrwxrwx	root	
index.html	11 KB	02/02/2022 11:00:09	rw-r--r--	root	
info.php	1 KB	02/02/2022 13:48:42	rw-r--r--	root	

Il y a certaines conditions pour créer un dossier puis mettre nos fichiers dont on a énormément besoin. Ces conditions doivent être les mêmes que celles de l'administrateur.

Dossier + fichiers dans Thomas – projet2022@172.20.21.195 - WinSCP

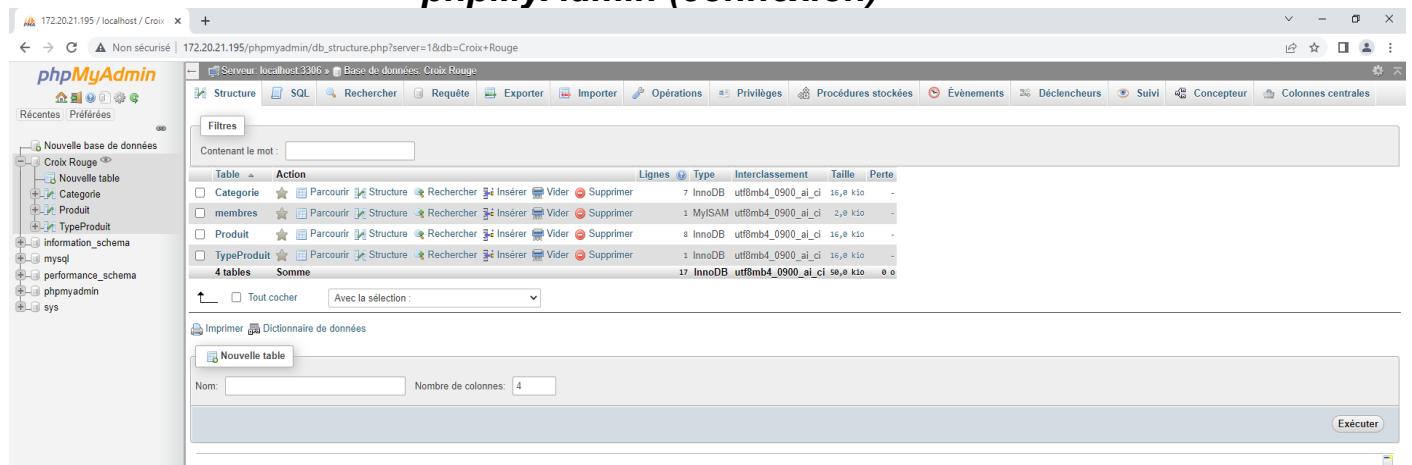
Nom	Taille	Type	Date de modification
..		Répertoire parent	14/03/2022 10:10:28
Bureau - Raccourci.lnk	1 KB	Raccourci	14/03/2022 10:10:28

Nom	Taille	Date de modification	Droits	Proprié...
..		25/05/2022 15:28:59	rwxr-xr-x	root
style.css	1 KB	05/04/2022 14:19:45	rw-rw-r--	projet2...
rest.php	7 KB	25/05/2022 15:01:25	rw-rw-r--	projet2...
QR Code (Tomate).png	2 KB	15/04/2022 13:28:42	rw-rw-r--	projet2...
QR Code (Riz).png	2 KB	15/04/2022 13:36:11	rw-rw-r--	projet2...
QR Code (Pomme).png	2 KB	15/04/2022 13:27:39	rw-rw-r--	projet2...
QR Code (Pomme de terre).png	2 KB	25/05/2022 10:56:09	rw-rw-r--	projet2...
QR Code (Pates).png	2 KB	15/04/2022 13:35:36	rw-rw-r--	projet2...
QR Code (Oeuf).png	2 KB	15/04/2022 13:27:01	rw-rw-r--	projet2...
QR Code (Lat).png	2 KB	15/04/2022 13:28:14	rw-rw-r--	projet2...
QR Code (Farine).png	2 KB	15/04/2022 13:37:09	rw-rw-r--	projet2...
QR Code (Eau).png	2 KB	15/04/2022 13:26:23	rw-rw-r--	projet2...
Projet+associatif+de+la+Croix-Rouge+française_202010....pdf	1 249 KB	08/03/2022 15:22:24	rw-rw-r--	projet2...
projet_associatif_CRF.pdf	90 KB	08/03/2022 16:31:31	rw-rw-r--	projet2...
Notepad+ .lnk	1 KB	01/02/2022 17:01:50	rw-rw-r--	projet2...
model.php	2 KB	23/05/2022 10:37:29	rw-rw-r--	projet2...
Informations sur les produits.txt	1 KB	22/04/2022 15:15:05	rw-rw-r--	projet2...
index.html.bak	5 KB	20/05/2022 14:11:58	rw-rw-r--	projet2...
index.html	5 KB	20/05/2022 14:16:10	rw-rw-r--	projet2...
index(3).html.bak	2 KB	10/05/2022 14:14:28	rw-rw-r--	projet2...
index(3).html	4 KB	30/05/2022 11:32:32	rw-rw-r--	projet2...
index(2).html.bak	2 KB	20/05/2022 14:18:10	rw-rw-r--	projet2...
index(2).html	3 KB	30/05/2022 09:47:22	rw-rw-r--	projet2...
html5-qrcode.min.js	312 KB	01/04/2022 15:17:09	rw-rw-r--	projet2...
fond-bleu.png	5 KB	04/02/2022 14:28:03	rw-rw-r--	projet2...
fd50c6ea052079acc2434d69973d9ed.jpg	7 KB	01/02/2022 15:28:16	rw-rw-r--	projet2...
fd50c6ea052079acc2434d69973d9ed - Copie.jpg	7 KB	01/02/2022 15:28:16	rw-rw-r--	projet2...
exemple QR Code.PNG	9 KB	05/04/2022 15:38:02	rw-rw-r--	projet2...
download_icone_128877.png	5 KB	11/03/2022 13:54:57	rw-rw-r--	projet2...
croix_rouge_française_03523800_20183350.jpg	35 KB	07/02/2022 10:26:34	rw-rw-r--	projet2...
app.ini	1 KB	23/05/2022 08:44:03	rw-rw-r--	projet2...
ajouter-produit.php.bak	1 KB	23/05/2022 08:44:37	rw-rw-r--	projet2...
ajouter-produit.php	1 KB	30/05/2022 11:29:58	rw-rw-r--	projet2...

Logiciel phpMyAdmin

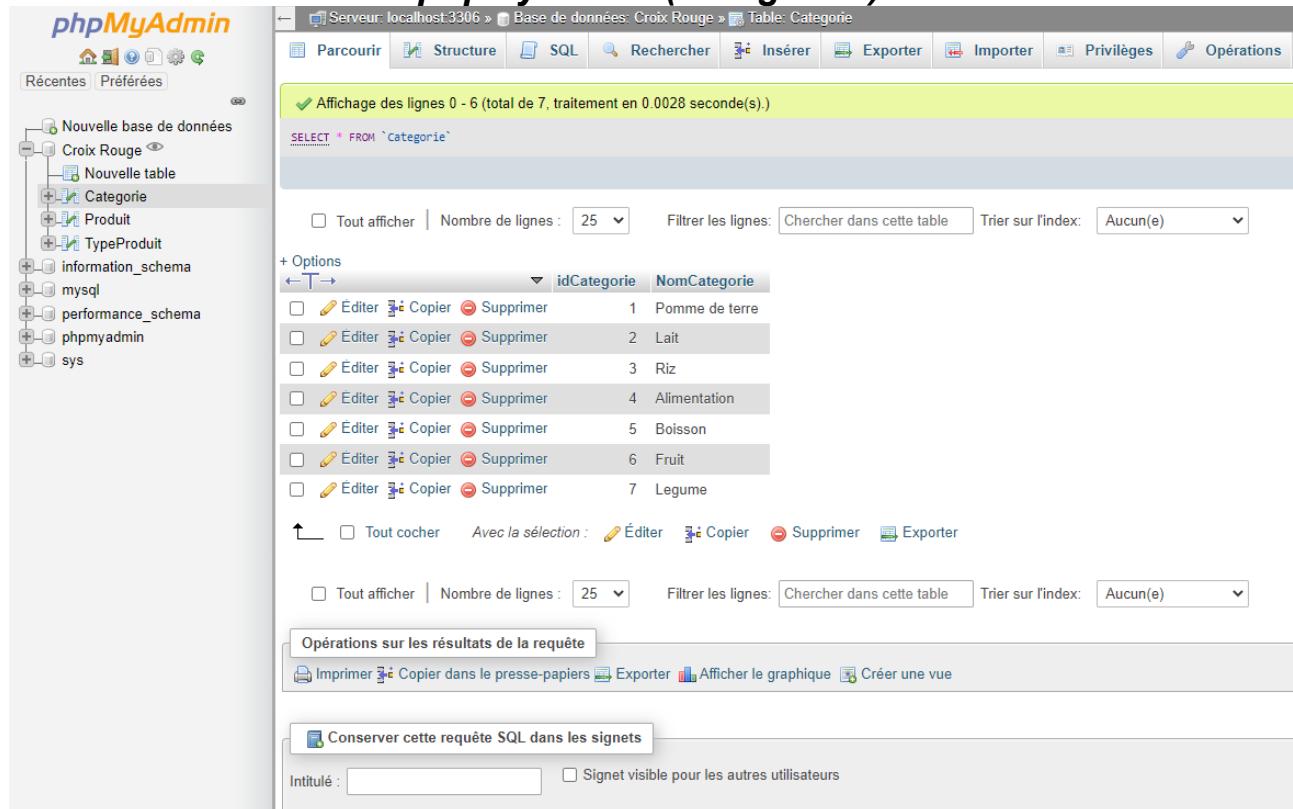
L'utilisateur a besoin de savoir ce qu'a produit l'administrateur dans le logiciel phpMyAdmin. Pour réaliser le site web avec les codes de programmation web, il est nécessaire de visionner les tables et les champs dans la base de données créée.

phpMyAdmin (connexion)



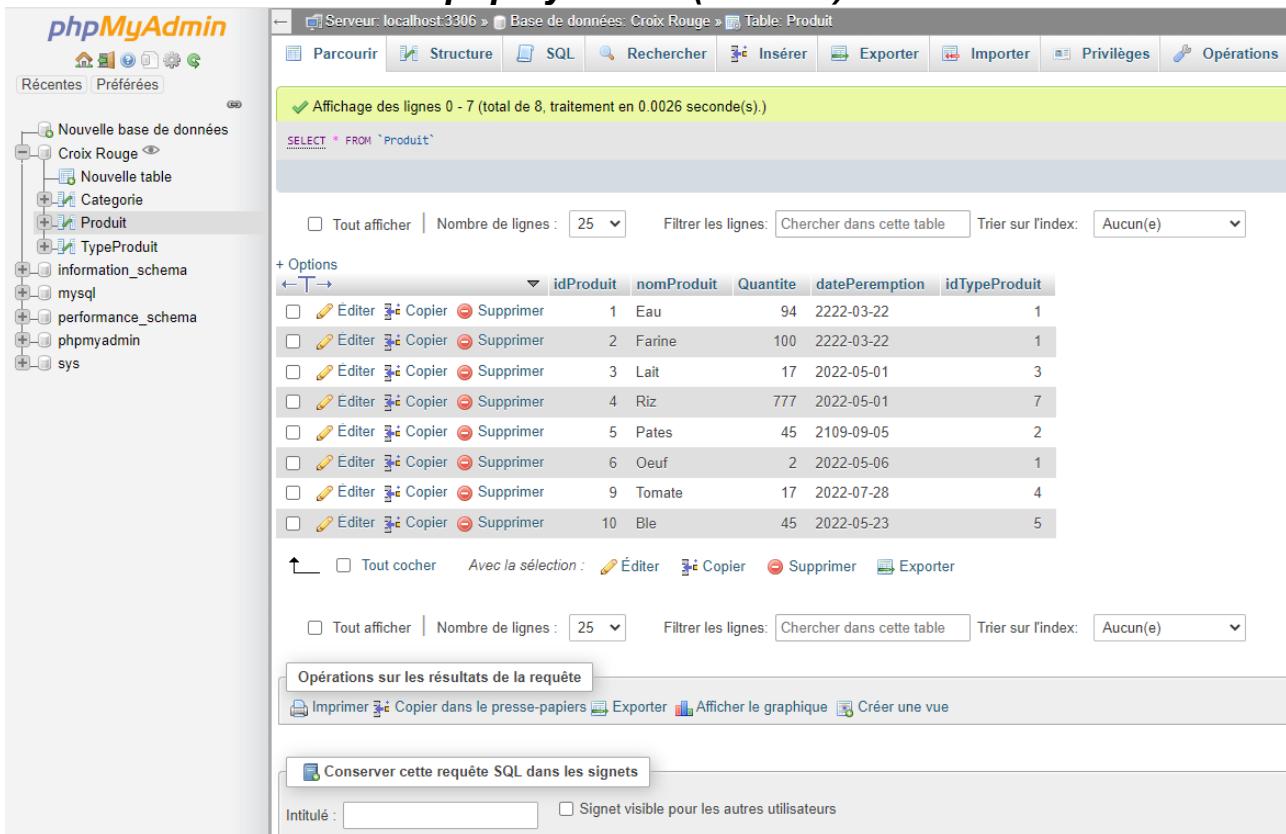
The screenshot shows the phpMyAdmin interface connected to a MySQL server at localhost:3306. The current database is 'Croix Rouge'. The left sidebar shows the database structure with tables: Categorie, membres, Produit, and TypeProduit. The main area displays the structure of the 'Produit' table, which has 8 rows and 4 columns. Below the table, there is a section for creating a new table with 4 columns.

phpMyAdmin (Catégories)



The screenshot shows the phpMyAdmin interface connected to a MySQL server at localhost:3306. The current database is 'Croix Rouge'. The left sidebar shows the database structure with tables: Categorie, membres, Produit, and TypeProduit. The main area displays the data from the 'Categorie' table, which contains 7 rows of food items. The table has two columns: 'idCategorie' and 'NomCategorie'. At the bottom, there are options to print, copy to clipboard, or export the data.

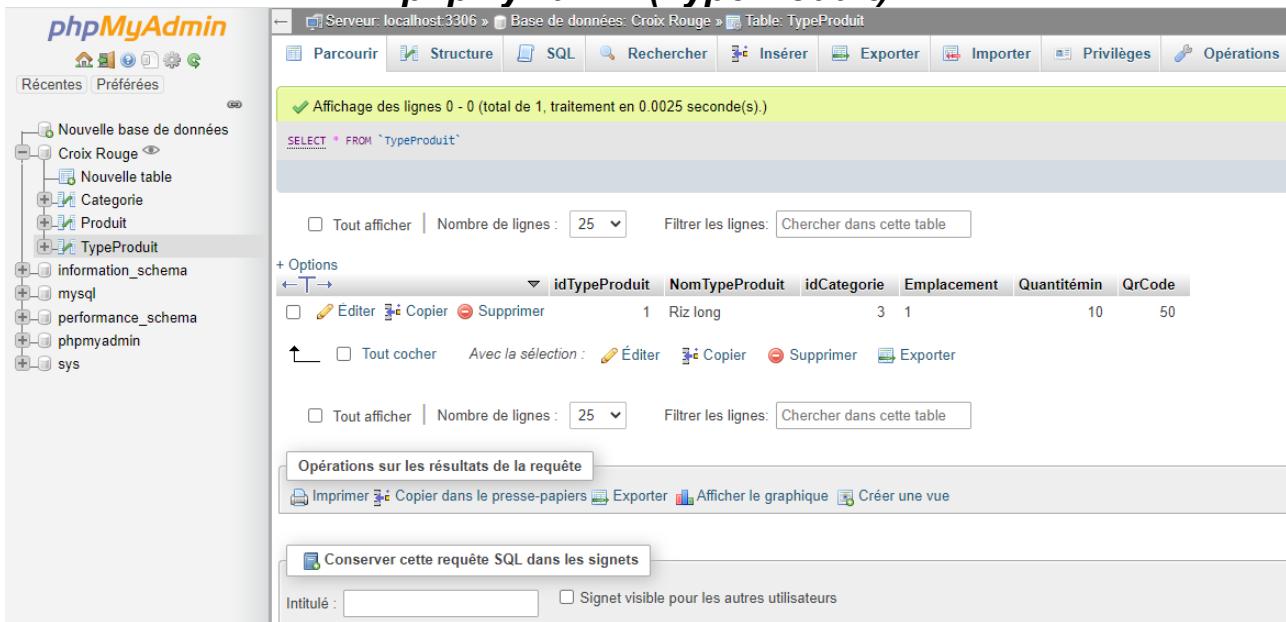
phpMyAdmin (Produit)



The screenshot shows the phpMyAdmin interface for the 'Produit' table in the 'Croix Rouge' database. The table has columns: idProduit, nomProduit, Quantite, datePeremption, and idTypeProduit. The data is as follows:

	idProduit	nomProduit	Quantite	datePeremption	idTypeProduit
1	1	Eau	94	2222-03-22	1
2	2	Farine	100	2222-03-22	1
3	3	Lait	17	2022-05-01	3
4	4	Riz	777	2022-05-01	7
5	5	Pates	45	2109-09-05	2
6	6	Oeuf	2	2022-05-06	1
7	7	Tomate	17	2022-07-28	4
8	8	Ble	45	2022-05-23	5
9	9				
10	10				

phpMyAdmin (TypeProduit)



The screenshot shows the phpMyAdmin interface for the 'TypeProduit' table in the 'Croix Rouge' database. The table has columns: idTypeProduit, NomTypeProduit, idCategorie, Emplacement, Quantitémin, and QrCode. The data is as follows:

	idTypeProduit	NomTypeProduit	idCategorie	Emplacement	Quantitémin	QrCode
1	1	Riz long	3	1	10	50

Résultats du site

Site web CROIX ROUGE



CROIX ROUGE

[Produits qrCode](#)

La croix rouge est une organisation internationale à vocation humanitaire, fondée pour venir en aide aux victimes de la guerre. La Croix-Rouge française (CRF) est une association d'aide humanitaire française fondée en 1864. Elle a pour objectif de venir en aide aux personnes en difficulté en France et à l'étranger. Ses missions fondamentales sont l'urgence, le secourisme, l'action sociale, la formation, la santé et l'action internationale.

Dans cette organisation, il existe des produits pour notre santé. Pour savoir les informations sur les produits, aller dans «Ajouter un produit» ou «Retirer un produit» :

[Ajouter un produit](#)
[Retirer un produit](#)

Auteurs	Catégories	Référence
<input type="button" value="Bleu (fond)"/> <input type="button" value="Blanc (fond)"/> <input type="button" value="Blanc (table)"/> <input type="button" value="Noir (table)"/>	Thomas VALLET Matenin TRAORE Nethusan SURESHKUMAR Formation Secours Hygiène Alimentation	Croix-Rouge-française Télécharger

Site web CROIX ROUGE (changement de couleur)



La croix rouge est une organisation internationale à vocation humanitaire, fondée pour venir en aide aux victimes de la guerre. La Croix-Rouge française (CRF) est une association d'aide humanitaire française fondée en 1864. Elle a pour objectif de venir en aide aux personnes en difficulté en France et à l'étranger. Ses missions fondamentales sont l'urgence, le secourisme, l'action sociale, la formation, la santé et l'action internationale.

Dans cette organisation, il existe des produits pour notre santé. Pour savoir les informations sur les produits, aller dans «Ajouter un produit» ou «Retirer un produit» :

[Ajouter un produit](#)
[Retirer un produit](#)

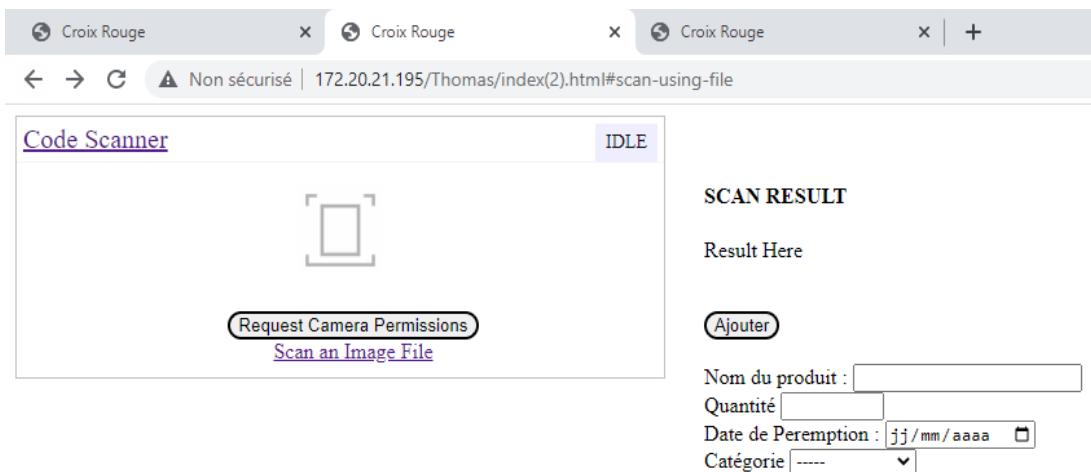
Site web CROIX ROUGE (message d'alerte)



172.20.21.195 indique
 Bienvenu à la CROIX ROUGE !

OK

Page Ajouter un produit (QR Code - scanner)



Code Scanner

IDLE

Request Camera Permissions

Scan an Image File

SCAN RESULT

Result Here

Ajouter

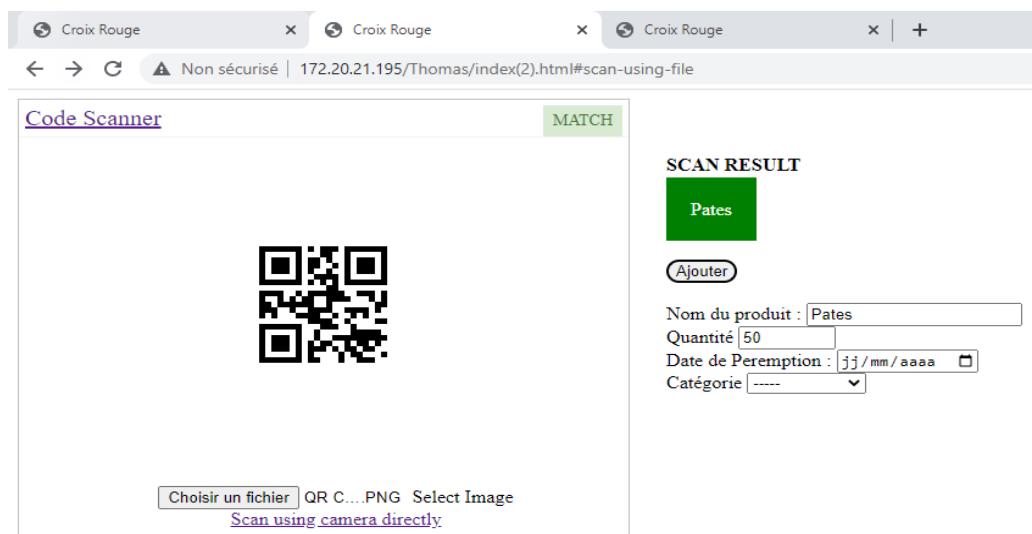
Nom du produit :

Quantité

Date de Peremption : jj / mm / aaaa

Catégorie

Page Ajouter un produit (QR Code - réussi)



Code Scanner

MATCH

Pates

Ajouter

Nom du produit : Pates

Quantité 50

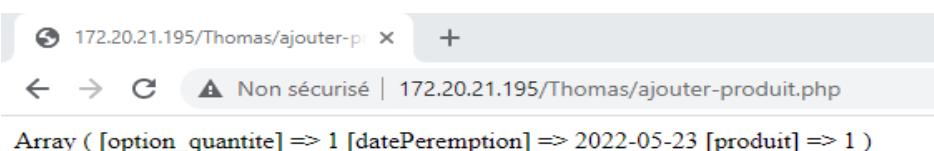
Date de Peremption : jj / mm / aaaa

Catégorie -----

Choisir un fichier QR C...PNG Select Image
Scan using camera directly

→ en choisissant un fichier après avoir pris en photos un QR Code

Page Ajouter un produit (produit ajouté dans la base de données)



172.20.21.195/Thomas/ajouter-p

Non sécurisé | 172.20.21.195/Thomas/ajouter-produit.php

Array ([option_quantite] => 1 [datePeremption] => 2022-05-23 [produit] => 1)

phpMyAdmin (Produit – produits ajoutés)

172.20.21.195 / localhost / Croix Rouge

Non sécurisé | 172.20.21.195/phpmyadmin/sql.php?server=1&db=Croix+Rouge&table=Produit&pos=0

phpMyAdmin

Récentes Préférées

Nouvelle base de données Croix Rouge Nouvelle table Categorie Produit TypeProduit information_schema mysql performance_schema phpmyadmin sys

Serveur: localhost:3306 > Base de données: Croix Rouge > Table: Produit

Parcourir Structure SQL Rechercher Insérer Exporter Importer Priviléges Opérations

Affichage des lignes 0 - 16 (total de 17, traitement en 0.0022 seconde(s)).

SELECT * FROM `Produit`

Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier sur l'index: Aucun(e)

	idProduit	nomProduit	Quantité	datePeremption	idTypeProduit
<input type="checkbox"/>	1	Eau	94	2222-03-22	1
<input type="checkbox"/>	2	Farine	100	2222-03-22	1
<input type="checkbox"/>	3	Lait	17	2022-05-01	3
<input type="checkbox"/>	4	Riz	777	2022-05-01	7
<input type="checkbox"/>	5	Pates	45	2109-09-05	2
<input type="checkbox"/>	6	Oeuf	2	2022-05-06	1
<input type="checkbox"/>	9	Tomate	17	2022-07-28	4
<input type="checkbox"/>	10	Ble	45	2022-05-23	5
<input type="checkbox"/>	11	Pomme de terre	50	2022-07-31	1
<input type="checkbox"/>	12	Pomme de terre	50	2022-07-31	1
<input type="checkbox"/>	13	Pomme de terre	50	2022-07-31	1
<input type="checkbox"/>	14	Pates	150	2109-09-05	1
<input type="checkbox"/>	16	Lait	40	2022-12-25	8
<input type="checkbox"/>	17	Oeuf	40	2022-05-06	2
<input type="checkbox"/>	18	Tomate	33	2022-09-01	4
<input type="checkbox"/>	19	Riz	900	2022-09-01	3
<input type="checkbox"/>	20	Eau	100	2022-08-27	1

Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier sur l'index: Aucun(e)

Page Retirer un produit

Croix Rouge

Non sécurisé | 172.20.21.195/Thomas/index(3).html

PRODUITS AJOUTES A RETIRER

Produit : Eau	Produit : Oeuf	Produit : Pomme	Produit : Lait	Produit : Tomate	Produit : Pates	Produit : Riz	Produit : Pomme de terre
							
Quantité disponible 100	Quantité disponible 70	Quantité disponible 45	Quantité disponible 30	Quantité disponible 17	Quantité disponible 50	Quantité disponible 23	Quantité disponible 36
<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>	<input type="button" value="Retirer"/>



Conclusion

Pour conclure, le bilan de cette partie du projet est quand même satisfaisant. Le programme fonctionne grâce au mini PC de l'administrateur. Le site fonctionne si tous les fichiers sont inclus dans le même répertoire d'échange avec les mêmes droits sur l'adresse IP (172.20.21.195).

Il manquait plusieurs tâches à réaliser :

1- Gestion de l'affichage des produits périmés

2- Gestion de l'affichage des produits dont la quantité minimale est atteinte

Il y avait beaucoup de code PHP à réaliser les dernières tâches, c'est-à-dire : faire des requêtes SQL pour la base de données et modifier des codes déjà construits dans des anciens projets. Le début de la partie été facile à réaliser, pour les langages HTML, CSS et JavaScript, puis par la suite, cela n'a pas été aussi évident.

Il n'y avait pas que du code à effectuer, mais aussi des manipulations pour être en lien avec l'administrateur. Les codes effectués par les profs servent pour beaucoup de projets dans le futur.

Le développement web été évidemment présent dans ce type de projet. Les tâches de l'approvisionnement et du retrait du produit sont faites vers la base de données. Les profs d'informatique devaient être présents pour avancer sur le projet. Il y avait moyen d'arriver plus loin si la connaissance du projet été plus présente et plus simple.

V. Partie étudiant 3 (Nethusan SURESHKUMAR)

Introduction

Comme expliqué précédemment dans l'ébauche du rapport de projet, la gestion des stocks nécessite un admin qui puisse gérer la gestion d'ajout de catégorie, de produit et sa génération de son QR Code. L'admin doit posséder un visuel explicite et clair de la base de données pour cela on va afficher aussi l'évolution du stock par l'aide d'un graphique.

Les outils à ma disposition sont des logiciels Visual Studio Code qui permettent de créer le codage dans les langages nécessaires que sont le PHP, le JSON, le JavaScript et le HTML.

SmarTTY permettra la communication à distance entre mon PC et le serveur Web (Mini PC).

Les matériels requis sont l' ordinateur et aussi le Mini PC qui possède Apache et phpMyAdmin pour la partie Serveur Web et le pc détiendra la base de données.

L'objectif est d'avoir une interface ergonomique simple d'utilisation et ayant toutes les fonctionnalités nécessaires qu'un administrateur afin qu'il puisse manier la base de données à sa guise avec l'aide de mes travaux effectués.



Tâches Personnelles

Ma partie se résume à la gestion de la base de données, développement du site web qui servira comme outils à l'administrateur.

Pour cela il faut effectuer :

- La mise en place du MCD.
- L'installation du Mini-PC, serveur Web et de la base de données et autres outils nécessaires.
- L'hébergement de la base de données.
- La définition du contenu et du codage du QR Code.

Je dois permettre à l'administrateur de :

- Gestion de l'ajout d'une nouvelle catégorie.
- Gestion de l'ajout d'un nouveau produit dans le stock et génération de son QR Code pour impression.
- Affichage de l'évolution du stockage des produits avec des graphiques.

Et comme tâche supplémentaire on essaiera de pouvoir essayer de posséder un outil de modification d'un produit afin de corriger la date ou la quantité.

Pour la fin on regroupe tous les travaux et de rendre un peu plus ergonomique pour l'administrateur.

Mise En Place du Mini PC

Le mini pc servira à pouvoir stocker tout le contenu et le codage.

Celui-ci permet à la croix rouge de pouvoir gérer en local la base de donnée possédant toutes les informations des stocks des produits denrées et leur date de péremption. Et on y intégrera le serveur Web.

A.Linux

Premièrement on a installé Linux sur le mini PC, un os robuste et fiable, un système réactif et rapide. Par dessus il est peu gourmand et sa licence est gratuite.

B.Paramètre Réseau

Lorsqu'on possède Linux et qu'on a créer un compte utilisateur et qu'on peut accéder à l'interface de Linux pour pouvoir paramétrier une ip fixe.



Avoir une ip fixe permettra à notre serveur Web d'avoir toujours la même adresse donc d'être discernable au notre et permet de se connecter aisément.

On utilisera l'IP IPV4 qui est 172.20.21.195 et son masque 255.255.255.0, donc la méthode utilisée est manuelle.

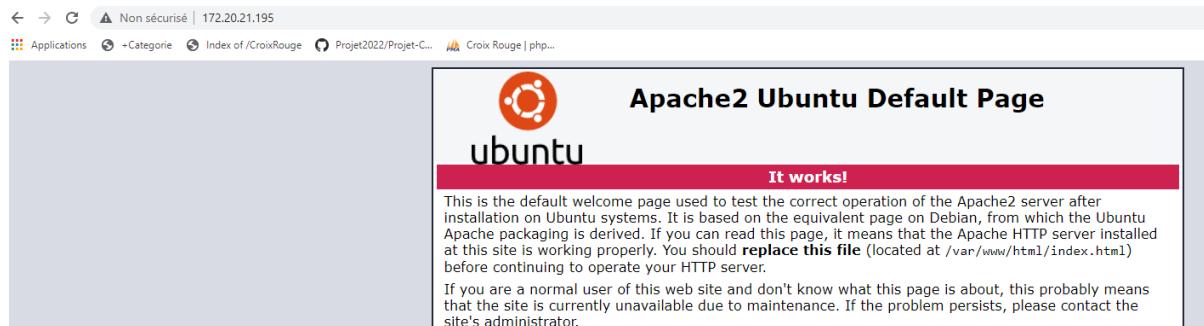
C.Apache

Ensuite il faudra installer Apache pour la création d'un serveur Web, Un serveur web Apache peut être un excellent choix pour exécuter votre site web sur une plateforme stable et polyvalente.

Donc on écrit dans le terminal (cmd) cette commande suivant on peut l'effectuer à travers SMARTTY pour le faire depuis son pc avec une connexion à distance:

```
$ sudo apt install apache2
```

Après l'installation terminée nous obtiendrons cette page par défaut sur sur l'adresse IP.

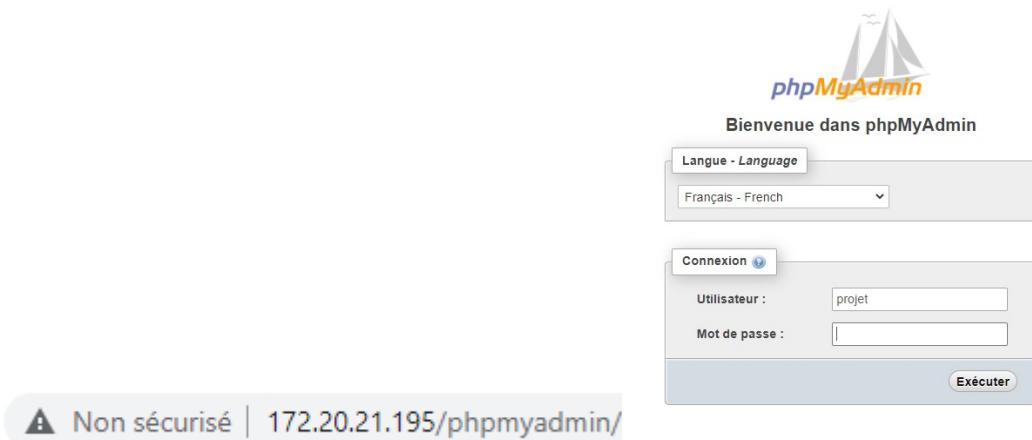


D.phpMyAdmin

Il faut bien posséder une base de données pour la gestion de stock et on va aussi l'héberger sur le Mini PC et pour cela on va installer phpMyAdmin.

```
-$ sudo apt install phpmyadmin
```

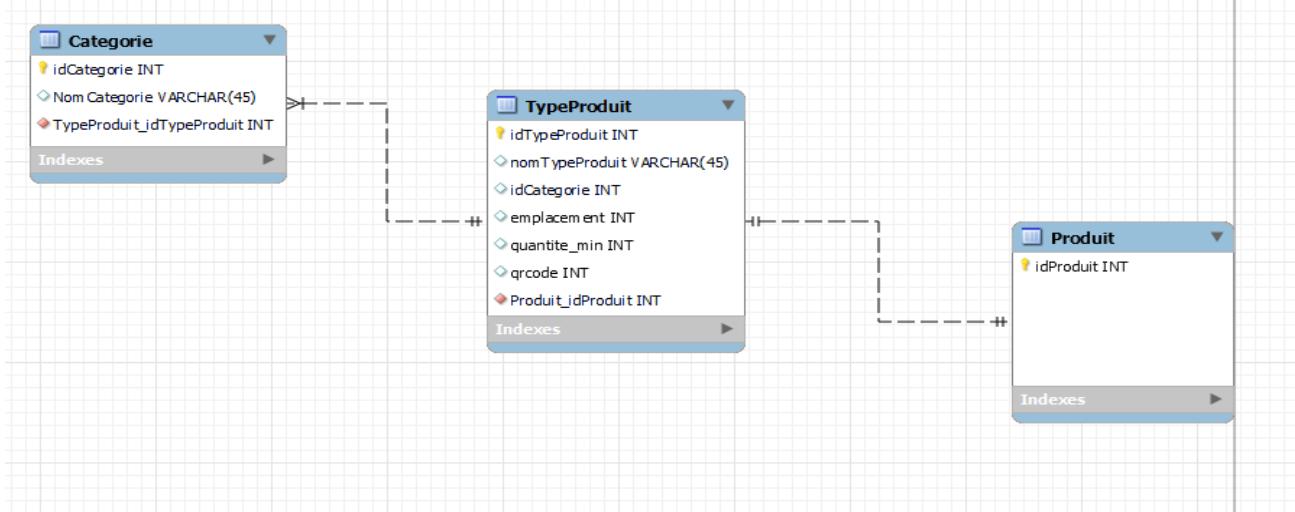
Après toutes étapes d'installations nous pouvons atteindre la page de connexion en tapant /phpmyadmin.



Il faut se connecter avec un identifiant et d'un mot de passe créé précédemment à l'aide du mini pc et acquérir les permissions de la base de données.

Base de données

L'élaboration des tables peut avoir lieu ensuite j'ai décidé de le faire à la main en utilisant MySQL Workbench pour juste avoir une représentation visuel de la base.



Pour ensuite la créer dans phpMyAdmin la base de donnée se nommant "Croix Rouge" et ayant les tables suivantes: "Categorie", "Produit", "TypeProduit".

Table	Action	Lignes	Type	Interclassement	Taille	Perte
Categorie	Parcourir Structure Rechercher Insérer Vider Supprimer	7	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
Produit	Parcourir Structure Rechercher Insérer Vider Supprimer	21	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-
TypeProduit	Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_0900_ai_ci	16,0 kio	-

et les champs de chaque catégorie ci-dessous :

idCategorie **NomCategorie** : "IDCategorie"; "Nomcategorie"

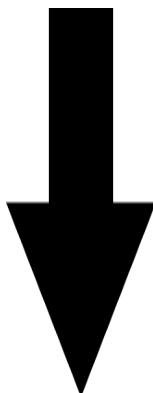
+ Options

 idProduit nomProduit Quantité datePeremption idTypeProduit :
 "IdProduit"; "nomProduit"; "Quantite"; "datePeremption"; "IdTypeProduit"

idTypeProduit **NomTypeProduit** **idCategorie** **Emplacement** **Quantitémin** **QrCode** :
 "idTypeProduit"; "NomTypeProduit"; "idCategorie"; "Emplacement";
 "Quantitemin"; "QrCode"

Hébergement

Pour l'héberger on passe par MySQL Workbench qui permet de mettre en forme la pour ensuite l'exporter en format SQL pour ensuite l'importer sur phpMyAdmin.



Screenshot of the phpMyAdmin Import interface. The top navigation bar includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, and More. Below the tabs, a section titled "File to Import:" contains instructions: "File may be compressed (gzip, bzip2, zip) or uncompressed. A compressed file's name must end in .[format].[compression]. Example: .sql.zip". There is a "Browse your computer:" input field with a "Choose File" button, which is highlighted with a red border. Next to it is the file path "one_example_com.sql" and the note "(Max: 200MIB)". Below this is a "Character set of the file:" dropdown menu set to "utf-8".

Codage

Pour la création de tout le contenu nécessaire pour satisfaire l'attente et des requêtes de l'administrateur. Nous créerons des applications à travers des pages HTML. Pour afin d'avoir les outils de gestion ou autres pour notre administrateur.

1) Le QR Code

Pour pouvoir générer le QR Code d'un produit nous passons par une librairie de google chrome : [Google Hosted Libraries](#)

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
<script type="text/javascript">
$(function() {
    var $img = $('#result img');
    var $parent = $img.parent();
    $img.load(function() {
        $img.css({'top':$parent.height() / 2 - $img.height() / 2, 'left':$parent.width() / 2 - $img.width() / 2}).show();
    });
    $('#submit').click(function() {
        $img.attr('src', 'https://chart.googleapis.com/chart?cht=qr&' + $('#generator form').serialize());
    });
});
</script>
```

(portion du code de la page)

Ce qui permet à l'aboutissement de ce formulaire assez simple.
 Juste à écrire le nom du produit dont lequel on souhaite créer un QR Code.

QR Code

Size:

150x150
 200x200
 250x250
 300x300

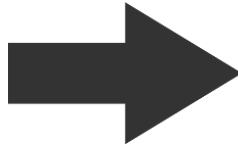
Produit a ajouter:

QR Code

Size:

150x150
 200x200
 250x250
 300x300

Produit a ajouter:





2) Les Requêtes SQL

Les requêtes vont permettre de gérer la base de données afin de pouvoir ajouter/enlever ou encore modifier les valeurs.

Avant tout requête envoyé il faut d'abord pouvoir se connecter à la base de donnée et dont ce but il faut passer par la méthode de mysqli_connect:

```
/* page: modification.php */
//connexion à la base de données:
$BDD = array();
$BDD['host'] = "localhost";
$BDD['user'] = "projet";
$BDD['pass'] = "projet";
$BDD['db'] = "Croix Rouge";
$mysqli = mysqli_connect($BDD['host'], $BDD['user'], $BDD['pass'], $BDD['db']);
if(!$mysqli) {
    echo "Connexion non établie.";
}
```

Localhost car le fichier sera exécuté par le mini PC, suivant l'identifiant et le mot de passe défini durant la création de la base.

Croix Rouge le nom de la base de données afin d'accéder à ses tables.

A) Les Catégories

Notre admin souhaite pouvoir ajouter de nouvelles catégories si il veut par exemple rajouter une catégorie Secours, Hygiène ou Alimentaire.

Pour cela nous devons insérer dans la base de donnée notre nouvelle ligne de catégorie, donc ici la requête SQL sera “INSERT”.

Donc par exemple :

```
!mysqli_query($mysqli,"INSERT INTO Categorie SET NomCategorie='".$_POST['NomCategorie']."'")
```

Pour la mise en page du formulaire d'insertion sera simple vu que l'admin n'aura besoin de taper le nom de la nouvelle catégorie.

```
<form method="post" action="addcategory.php">
    Nom du Categorie (Lettre) : <input type="text" name="NomCategorie">

    <input type="submit" value="Ajoutez">
</form>
```

Ajoutez des catégories

Ajoutez des catégories dans la base.

Nom du Categorie (Lettre) : Ajoutez

B) Produit

L'insertion de nouveau produit doit être primordialement gérer l'administrateur pour l'ajouter c'est exactement la même procédure que celle de la catégorie mais sauf qu'il y a plus de champs qu'un simple nom à ajouter.

```
<form method="post" action="addproduct.php">
    Nom du TypeProduit (Lettre) : <input type="text" name="NomTypeProduit">
    <br />
    Quantité minimum : <input type="text" name="Quantitémin">
    <br />
    Date de Péremption : <input type="date" name="datePeremption">
    <br />
    IDTypeProduit (0-9) : <input type="text" name="idTypeProduit">
    <br />
    <input type="submit" value="Ajoutez">
</form>
<?php
```

Ajoutez des produits

Insertion de nouveaux produit dans la base.

Nom du Produit (Lettre) :	<input type="text"/>
Quantité :	<input type="text"/>
Date de Péremption :	<input type="date"/> jj/mm/aaaa
IDTypeProduit (0-9) :	<input type="text"/>
<input type="button" value="Ajoutez"/>	

Comme le montre le formulaire la quantité dans l'unité kg ou si c'est un liquide en L ou encore si c'est par pièce/unité.

La date de péremption si elle en ne possède pas faut mettre une date lointaine. L'idTypeProduit permet de différencier des produits qui peuvent paraître similaire exemple: lait de soja =/ lait .

Mais si la quantité augmente ou diminue, la date péremption doit être modifiée. La modification d'une quelconque information d'un produit doit être possible donc ici la requête nécessaire dans ce cas précis est celui de l' "UPDATE".

Par exemple :

```
if(!mysqli_query($mysqli,"UPDATE Produit SET Quantite='".$_POST['Quantite']."' , datePeremption='".$_POST['datePeremption']."' , idTypeProduit='".$_POST['idTypeProduit']."' where nomProduit='".$_POST['nomProduit']."' )){
```

“Where” permet de retrouver le nom du produit dont on veut apporter des modifications.

Voici la mise en page :

Modifiez des produits

Modification des informations apartir du nom du produit identique à la base.

Nom du Produit (Lettre) :	<input type="text"/>
Quantité :	<input type="text"/>
Date de Peremption :	<input type="text"/> jj/mm/aaaa <input type="checkbox"/>
IDTypeProduit (0-9) :	<input type="text"/>
<input type="button" value="Modifiez"/>	

Elle est exactement pareille que celle de l'ajout de produit.

Évidemment je précise que le nom du produit doit être identique à celui inséré dans la base de données. Les trois autres valeurs sont celles que je veux modifier.

C'est une partie bonus qui me semblait nécessaire, donc je l'ai produite.

3) Le graphique

Affichage de l'évolution du stockage des produits avec des graphiques permets un suivi simple pour l'administrateur, il permet à lui de savoir les actions qu'il doit effectuer et de prévenir avant les incidents.

Pour la génération d'un graphique nous allons utiliser les outils proposer par AMcharts.

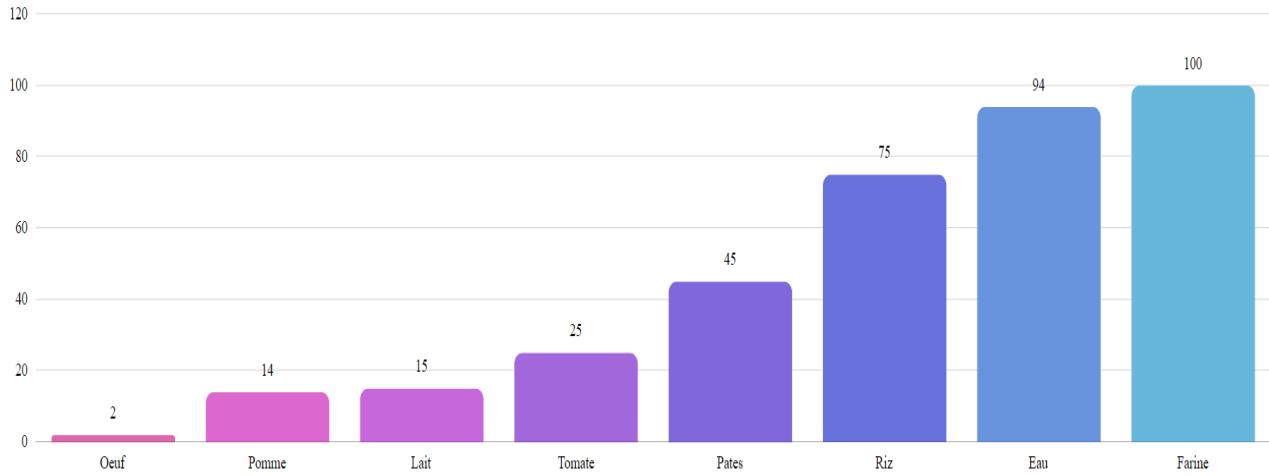
AMCharts est un outil permettant de créer des graphiques en JavaScript ou en HTML5 par la librairie qu'elle propose.



Nous allons passer par un diagramme en bâton afin de voir la quantité qui sera triée par ordre du plus petit au plus grand.

```
<!-- Resources -->
<script src="https://cdn.amcharts.com/lib/4/core.js"></script>
<script src="https://cdn.amcharts.com/lib/4/charts.js"></script>
<script src="https://cdn.amcharts.com/lib/4/themes/animated.js"></script>
```

Le “/lib” qui permet de savoir que c'est une librairie qu'on utilise puis “/4” qui indique la version de AMCharts. Le core charts qui contiennent les information afin de créer la forme du diagramme et ensuite “animated” permet de produire des animations du graphique afin de le rendre plus esthétique.



Ce graphique sera actualisé automatiquement toutes les 30 secondes afin de ne pas avoir de valeurs obsolètes.

```
<!-- HTML -->
<meta http-equiv="refresh" content="30">
<div id="chartdiv"></div>
```

Ce graphique aura besoin de valeur de la base de données pour cela il faut un fichier qui permet de regrouper les valeurs présentes dans la base.

```
$sql2 = "SELECT * FROM Produit";
$result = $mysqli->query($sql2);

// Fetching data from the database
// and storing in array of objects
while($row = $result->fetch_array()) {
    $studentdata[] = array(
        "Produit" => $row["nomProduit"],
        "Quantite" => $row["Quantite"],
        // "Total_Score" => $row["score"],
        // "Articles" => $row["articles"],
    );
}
```

Ici on stocke les valeurs dans un tableau.

Après avoir stocké les valeurs il faut la création d'un fichier dynamique JSON qui contiendra les données.

```
// Creating a dynamic JSON file
$file = "YXdonnee.json";
```

Sauf que ces données faut les convertir en JSON pour les mettre et de pouvoir manipuler celles ci:

```
if(file_put_contents($file,
    json_encode($studentdata)))
    echo(json_encode($studentdata));
```

Ce qui produit ce fichier de données possédant les données brutes entre des guillemets et des crochets.

```
{"Produit":"Lait","Quantite":"15"}, {"Produit":"Riz","Quantite":"75"}  

[{"Produit":"Eau","Quantite":"94"}, {"Produit":"Farine","Quantite":"100"},  

 {"Produit":"Tomate","Quantite":"25"}, {"Produit":"Pomme","Quantite":"14"}]  

, {"Produit":"Pates","Quantite":"45"}, {"Produit":"Oeuf","Quantite":"2"},
```

Pour enfin on revient dans le fichier du graph dans l'objectif de relier les fichiers entre eux et permettre de donner des valeurs à notre graphique.

```
chart.dataSource.url='./Donnee.php';
```

Il faut bien dire que l'abscisse sont les noms et en ordonné c'est la quantité.

```
var series = chart.series.push(new am4charts.ColumnSeries());
series.dataFields.categoryX = "Produit";
series.dataFields.valueY = "Quantite";
series.tooltipText = "{valueY.value}"
```

Ce sont exactement les mêmes valeurs que celles présentes dans la base de données.

	+ Options	← T →		idProduit	nomProduit	Quantite	datePeremption	idTypeProduit
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	Lait	20	2023-01-07	1
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	Riz	23	2036-10-23	3
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	Pates	50	2036-10-30	3
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	Tomate	17	2022-08-18	5
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	Eau	100	2035-12-11	7
<input type="checkbox"/>	Éditer	Copier	Supprimer	6	Ble	39	2031-06-30	6
<input type="checkbox"/>	Éditer	Copier	Supprimer	7	Eau	80	2035-12-31	7

4) Finitions

Pour rendre la gestion et le maniement de la base de données beaucoup plus simple nous avons créé ces pages et il faut regrouper afin de le rendre beaucoup plus ergonomique .

Ce que nous avons comme rendu final.

Menu Admin



La page avec ses différents onglets et sa barre de navigations.

```

</head>
<h1>Menu Admin</h1>
<body>

    <nav>
        <ul>
            <li><a href="graph.html">Stock</a></li>
            <li><a href="addcategory.php">+Categorie</a></li>
            <li><a href="addproduct.php">+Produit</a></li>
            <li><a href="modification.php">Modifier</a></li>
            <li><a href="qrcode.html">QrCode</a></li>
        </ul>
    </nav>
    <h2>Graphique du stockage des produits denrees</h2>

```

Comme il faut rajouter ceci à toutes les pages, on passe par trois fichiers pour contenir le code HTML car nos fichiers sont en PHP.

```

<?php
    include("header.php");
    include("menu.php");
?>

```

Le header contient l'en-tête de la page donc le nom de la page plus le titre qui apparaît.

```

<!DOCTYPE html>
<html>
    <head>
        <title>Menu</title>
        <link rel="stylesheet" type="text/css" href="Menu.css">
    </head>
    <h1>Menu Admin</h1>
    <body>

```

Le menu est donc la barre de navigation dont on l'ajoute à toutes les pages pour avoir permettre de revenir sur page précédente ou à une nouvelle.

```
<nav>
  <ul>
    <li><a href="graph.html">Stock</a></li>
    <li><a href="addcategory.php">+Categorie</a></li>
    <li><a href="addproduct.php">+Produit</a></li>
    <li><a href="modification.php">Modifier</a></li>
    <li><a href="qrcode.html">QrCode</a></li>
  </ul>
</nav>
```

Cette barre on y attribuera un fichier CSS afin de rendre la barre plus intuitive.

```
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  background-color: #ca0000;
  overflow: hidden;
}

li {
  float: left;
}

li a {
  color: white;
  padding: 20px;
  text-decoration: none;
  display: block;
}

li a:hover {
  background-color: blueviolet;
```

Permet d'avoir une bande rouge comme barre de navigation.



Le nom de chaque section en blanc.

Cette ligne permet lorsque notre souris est au dessus d'un section de le faire changer de couleur.

Comme ici où la section "stock" est en violet et les autres en rouge.



Et donc le footer est le dernier fichier à inclure qui permet d' encapsuler le code en HTML.

```
<?php
  include("footer.php");
?>
```

```
1  </body>
2  </html>
```

Tout ceci permet de se balader avec un chemin bien précis et simple au lieu d'avoir un index incompréhensible comme celui ci-dessous.

Index of /CroixRouge

	Name	Last modified	Size	Description
	Parent Directory		-	
	BDD.html	2022-04-06 08:19	297	
	Donnee.php	2022-04-13 10:36	1.1K	
	JSPCQ/	2022-04-13 10:45	-	
	Menu.css	2022-05-11 08:57	809	
	Menu.html	2022-05-10 15:33	469	
	YXdonnee.json	2022-05-30 09:45	212	
	addcategory.php	2022-05-11 09:00	3.4K	
	addproduct.php	2022-05-23 08:33	4.0K	
	ext/	2022-05-11 09:06	-	
	footer.php	2022-05-11 08:27	16	
	graph.html	2022-05-11 09:03	2.7K	
	header.php	2022-05-10 16:33	159	
	images/	2022-05-11 08:34	-	
	menu.php	2022-05-11 08:27	294	
	modification.php	2022-05-30 08:31	3.7K	
	qrcode.html	2022-05-23 08:38	2.6K	

Apache/2.4.41 (Ubuntu) Server at 172.20.21.195 Port 80



Conclusion

Ayant comme ambition d'être développeur plus tard, mon projet a été pour moi une réelle occasion de m'initier aux mondes du développement. J'ai pu apprendre à être plus autonome au fil de recherches et d'études pour aboutir à mon projet. Mais aussi à travailler en équipe comme nous avions un projet en commun, par exemple avec la conception de la base de données en équipe ou encore les diagrammes UML comme nos codes sont directement liés et se complètent.

Le fait de voir l'avancement d'un projet après de longue durée est très satisfaisant. Mon projet a été très enrichissant et j'ai pu en apprendre énormément du début à la fin. J'ai énormément progressé en langage en PHP et en SQL et développé mes connaissances sur les librairies et aussi le JavaScript et le JSON. De la conception d'un projet (diagramme, étude du projet, base de données, contrainte matériel) à la fin (codage, test, fonctionnement).

Une fois les programmes codés et finies pour chaque étudiant, notre but était de rassembler nos programmes afin que le projet principal puisse marcher.

Il nous reste un point à régler pour le bon déroulement du projet.

En effet, nous continuons à terminer les quelques dernières modifications afin qu'on puisse échanger des données avec le groupe de Supervision.

Mon point de vue sur le bilan définitif du projet est globalement satisfaisant malgré les détails qui pouvaient être peaufinés.

Malgré tout à l'heure actuelle le projet n'est pas encore fini nous essayons de le terminer.

J'aimerai aussi remercier les professeurs qui ont pu apporter leur aide parfois par exemple pour amchart, malgré tout j'ai pu développer l'autonomie durant le projet.



IV. CONCLUSION

Pour conclure, le projet sur la Croix-Rouge a été une réussite pour le groupe qui pendant 13 semaines ont pratiqué pleins d'activités différentes qui ont un rapport avec l'informatique et réseaux.

Il a fallut pendant toute cette période analyser et comprendre le but du projet. La partie physique été présente, et avait un rapport avec le projet. La partie du QR Code fut une compréhension de ce qui se passe à l'intérieur et de son intérêt.

Tous codes de programmation fut présents pour réaliser ce qui a été demandé. Les utilisateurs ont réalisé toutes leurs tâches pendant ces 13 semaines, même chose pour l'administrateur.

L'administrateur commençait par écrire les informations sur les produits, l'utilisateur du site web CROIX ROUGE les utilise pour effectuer des requêtes (les ajouter dans la base de données) et pour les retirer en prenant en compte les quantités. Puis l'utilisateur de la Raspberry pi 4 devait prendre en compte les dates limites des produits, pour alerter les dates de péremption.

Ce projet pourrait être présenté dans le futur, lorsqu'on entre dans le domaine du travail. Si on doit faire un projet professionnel plus tard, il y a la possibilité de s'inspirer du projet Croix-Rouge.