

Projet

- Technologies Émergentes -

La Boutique Intelligente

Deronsart Nicolas
Mayeux Liam
Tancrez Victor

Concept du Projet

-

La Boutique Intelligente

I/ Composition du groupe

Pour la réalisation du projet nous sommes un groupe de 3 personnes composé de :

- Deronsart Nicolas
- Mayeux Liam
- Tancrez Victor

II/ Concept du projet : Utilité et Utilisateurs

L'objectif que nous nous sommes fixés pour la réalisation de ce projet est de créer une "Boutique Intelligente".

L'objectif est de créer une interface de gestion de la boutique pour les employés et le propriétaire :

- Notre projet permettra tout d'abord d'enregistrer le nombre de passages dans la boutique, grâce à un capteur comptant les personnes y entrant.
- De plus, pour chaque personne qui entrera dans la boutique une image prise automatiquement par une caméra sera analysée par une intelligence artificielle qui permettra d'identifier le genre du visiteur, ce qui permettra au gérant de pouvoir obtenir des statistiques quant aux visiteurs de sa boutique.
- Un système de gestion automatique de la température du magasin sera intégré.
- La gestion du stock pourra être possible depuis l'application.
- L'ambiance de la boutique sera gérable depuis un onglet de l'application permettant de choisir la musique diffusée dans le haut-parleur du magasin.
- Il sera également possible d'ajouter un capteur de flamme pour activer automatiquement l'alarme incendie en cas de problème.

Cette interface sera donc utilisée par les employés de la boutique pour gérer au mieux leur commerce, et offrir aux visiteurs une expérience de consommateur optimale.

III/ Besoins répondus

Grâce à notre projet les responsables pourront gérer et automatiser leurs boutiques efficacement.

Le compteur de personne, existant déjà actuellement dans de nombreux commerces, est ici amélioré permettant également d'obtenir des données supplémentaires sur le sexe et la tranche d'âge des visiteurs.

Ces informations leur permettront de mieux connaître leur clientèle, et par ce fait mieux savoir quel public viser lors de leurs démarches publicitaires.

La température régulée automatiquement permettra aux potentiels acheteurs de profiter d'une température agréable selon la météo actuelle de la ville dans laquelle ils se situent.

La gestion du stock automatique facilitera le travail des vendeurs qui pourront savoir rapidement si un produit est encore disponible, mais aussi préviendra lorsqu'un produit se rapproche de la fin du stock.

La gestion d'ambiance permettra d'améliorer l'attractivité de la boutique en la rendant plus agréable et attirante pour les consommateurs.

IV/ Présence sur le marché

Des projets de magasins connectés sont actuellement en développement mais sur le marché on ne trouve pour le moment que de petites idées selon les boutiques. Comme par exemple le compte personne à l'entrée ou la classique gestion de température comme on peut déjà trouver dans des foyers.

Le but de notre projet sera de combiner plusieurs idées pour faciliter le plus possible la gestion de la boutique.

De plus, certains projets de boutique connectée tel que celui d'Amazon, consistant à suivre l'utilisateur dans la boutique et d'enregistrer numériquement tous ses achats pour le prélever automatiquement à la sortie de la boutique, retirent donc tout contact entre vendeurs et consommateurs.

Ici ce n'est absolument pas notre objectif. Nous recherchons uniquement à faciliter la gestion pour le personnel, leur permettant de se concentrer sur les clients pour les aider aux mieux dans leurs achats.

V/ Canaux d'entrée de data

Les capteurs utilisés seront :

- Un compteur de personnes, connectivité objet-à-objet.
- Une caméra, connectivité objet-à-passerelle.
- Un thermomètre, connectivité objet-à-cloud.
- Un capteur de flamme, connectivité objet-à-objet.

VI/ Ambition de réalisation

L'application web, utilisable depuis un ordinateur, est entièrement fonctionnelle, mais la réalisation matérielle n'est quant à elle pas totalement effectuée dû au coût des composants.

Pour récupérer les images à analyser, la webcam de l'ordinateur sera utilisée en démonstration.

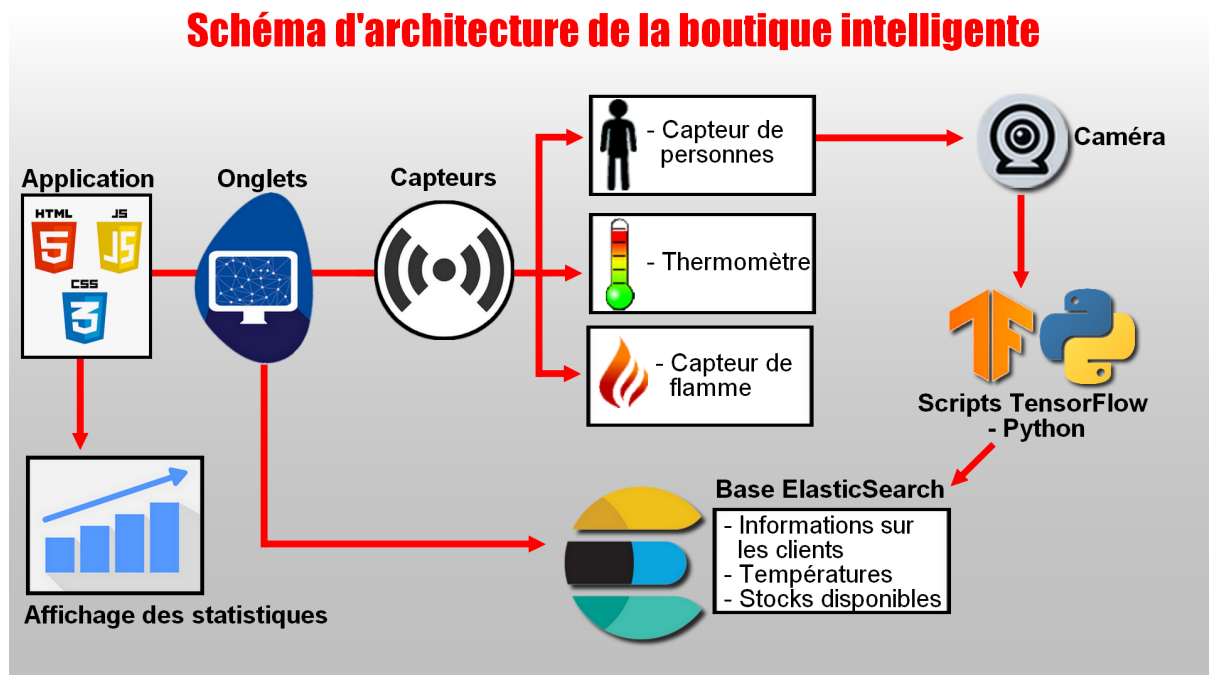
Pour les autres données, pour éviter la dépense d'argent des scripts seront créés pour simuler les données, comme si l'application était actuellement en cours d'utilisation.

Document Architecture Technique

-

La Boutique Intelligente

I/ Schéma visuel d'architecture



II/ Choix techniques pour la réalisation

L'application utilisateur permettant d'interagir avec la boutique intelligente est une application web réalisée en PHP, JavaScript, HTML5 et CSS.

Le traitement des données images est effectué dans un script Python 3.8.5 utilisant le Deep Learning avec un CNN (Convolutional Neural Network), et plus précisément les librairies :

- Keras 2.4.3 pour le traitement des images avant de les introduire dans le réseau.
- TensorFlow 2.5.0 pour le réseau de neurones à convolution permettant d'analyser le contenu des images.

Pour la gestion de la base de données Elasticsearch en version 7.16.3 est utilisé.

Le PHP version 7.3 est utilisé pour interagir avec les données de Elasticsearch depuis le terminal grâce à `shell_exec()`.

III/ Solutions alternatives mises en place

Dû au manque de moyens pour la réalisation, les composants matériels seront pour la plupart remplacés par des moyens informatiques simples à mettre en œuvre.

- Le compteur de personnes a été remplacé par le fait de compter le nombre de personnes faisant un achat dans la boutique.
- Pour la reconnaissance d'image on peut envoyer une indication à la caméra pour prendre une photo depuis notre application.
- La caméra utilisée pour la démonstration sera celle de la webcam de l'ordinateur.
- Le thermomètre intérieur sera remplacé par un script générant de manière aléatoire des valeurs de température plausibles toutes les 2 minutes.
- Le thermomètre extérieur correspond aux données météo récupérées sur internet en temps réel.

IV/ Description des sources de données

Pour les sources de données de base dans l'application, mais également ajoutées au fur et à mesure de l'utilisation concrète de l'application, différents formats seront utilisés :

- Format d'images provenant de la caméra de sécurité reliée à l'ordinateur.
- Fichier audio .mp3 pour les musiques à diffuser dans la boutique.
- Format JSON pour le reste des données, il sera détaillé dans la partie suivante ci-dessous.

V/ Modélisation de la base ElasticSearch

Notre base ElasticSearch est composée de différents indexs contenant des fichiers JSON contenant toutes les informations nécessaires au bon fonctionnement de l'application.

Ainsi pour chaque document JSON sera écrit ci-dessous son index ainsi qu'un exemple de contenu possible pour présenter la structure de chaque document.

Index : shop

Contient la température à l'intérieur et à l'extérieur, et l'état On/Off des différents appareils connectés.

```
{
  "temperatureInside" : 20,
  "temperatureOutside": 15,
  "personSensor" : "On",
  "flameSensor" : "On",
  "heating": "on"
}
```

Index : customer

Contient l'ensemble des clients qui sont représentés par leur civilité (Mr ou Mme), prénom, nom, adresse mail, numéro de téléphone et nombre de points de fidélités.

```
{
  "civility" : "Mr",
  "firstName" : "Jason",
  "lastName" : "Dupont",
  "mail" : "jason.dupont@gmail.com",
  "phone" : "06 06 06 06 06 ",
  "loyaltyPoints" : 0
}
```

Index : shopdailydata

Contient toutes les données de la date pour la boutique pour chaque heure. On y retrouve la température, le nombre de clients et le chiffre d'affaires.

```
{
  "date" : "03-07-22",
  "7":{
    "temperature": 10,
    "numberCustomers": 10,
    "numberMen": 6,
    "numberWomen": 4,
    "turnover" : 500
  },
  "8":{
    "temperature": 12,
    "numberCustomers": 14,
    "numberMen": 6,
    "numberWomen": 8,
    "turnover" : 600},
  "9":{
    "temperature": 14,
    "numberCustomers": 15,
    "numberMen": 7,
    "numberWomen": 8,

```

```
    "turnover" : 700},
  "10":{
    "temperature": 14,
    "numberCustomers": 20,
    "numberMen": 12,
    "numberWomen": 8,
    "turnover" : 800},
  "11":{
    "temperature": 16,
    "numberCustomers": 25,
    "numberMen": 5,
    "numberWomen": 20,
    "turnover" : 900},
  "12":{
    "temperature": 18,
    "numberCustomers": 30,
    "numberMen": 12,
    "numberWomen": 18,
    "turnover" : 1000},
  "13":{
    "temperature": 20,
    "numberCustomers": 40,
    "numberMen": 25,
    "numberWomen": 15,
    "turnover" : 2000},
  "14":{
    "temperature": 19,
    "numberCustomers": 60,
    "numberMen": 30,
    "numberWomen": 30,
    "turnover" : 3000},
  "15":{
    "temperature": 18,
    "numberCustomers": 50,
    "numberMen": 20,
    "numberWomen": 30,
    "turnover" : 1500},
  "16":{
    "temperature": 19,
    "numberCustomers": 30,
    "numberMen": 16,
    "numberWomen": 14,
    "turnover" : 2000},
  "17":{
    "temperature": 20,
    "numberCustomers": 20,
    "numberMen": 15,
    "numberWomen": 5,
    "turnover" : 3000},
  "18":{
```

```
    "temperature": 18,  
    "numberCustomers": 20,  
    "numberMen": 5,  
    "numberWomen": 15,  
    "turnover" : 4000},  
  "19":{  
    "temperature": 17,  
    "numberCustomers": 15,  
    "numberMen": 7,  
    "numberWomen": 8,  
    "turnover" : 2000}  
}
```

Index : product

Contient tous les articles en stock dans le magasin, chaque article a un nom, une quantité et un prix.

```
{  
  "item" : "tondeuse",  
  "quantity" : 2,  
  "price" : 99.99  
}  
{  
  "item" : "pot de peinture (500ml)",  
  "quantity" : 2,  
  "price" : 30.50  
}
```

Rapport Technique

-

La Boutique Intelligente

I/ Utilisation de notre projet

Tout d'abord pour faire fonctionner le projet une connexion internet est nécessaire, de plus il faut mettre le dossier **projetTec** dans le répertoire **/var/www/html**.

À la racine vous retrouverez le dossier src contenant l'intégralité du code source de notre application mais aussi un script bash nommé **lancement.sh**.

Vous pourrez d'ici lancer l'application depuis le terminal à l'aide de la commande suivante : **bash lancement.sh**.

L'application sera ainsi lancée en ouvrant firefox mais il lancera aussi le script bash **genereData.sh**, qui sera expliqué plus tard.

II/ Détail du contenu du code source

Le code source est divisé en 3 sous-dossiers : **data**, **image_recognition** et **pageweb**. Dans cette partie nous reviendrons sur les contenus de ces différents dossiers.

1. Le dossier **image_recognition**

Ce premier dossier est celui contenant l'Intelligence Artificielle pour la reconnaissance d'image pour le genre des clients.

Le fichier **gender_recognition.py** prends en entrée une image d'un humain et détermine si c'est un homme ou une femme.

Les poids de l'IA entraînée sont sauvegardés dans le sous-dossier **checkpoints**.

Un script bash viendra ensuite récupérer l'information et l'ajouter aux données du jour dans shopdailydata.

Technologies utilisées :

- Python
- TensorFlow

Type de données :

- .json
- .jpg

2. Le dossier **data**

Le but de ce dossier est de stocker les données de la boutique sous le format JSON.

Le fichier [genereData.sh](#) permet de simuler les données de nos capteurs et notamment la température intérieure en la modifiant de cette façon, si le chauffage est allumé alors on augmente la température intérieure de 1 degré toutes les 2 minutes jusqu'à une limite de 40, sinon si le chauffage est éteint alors on baisse la température de 1 degré toutes les 2 minutes, et ce jusqu'à un minimum qui est la température extérieure.

Technologies utilisées :

- ElasticSearch
- Langage shell

Type de données :

- .json

3. Le dossier **pageweb**

Ce dernier dossier contient tout le code source de notre l'application web organisé sous forme de sous-dossiers pour les images et pour les différentes pages. Le premier, [pageweb/img](#), contient toutes les images servant dans l'application. Les autres dossiers servent à créer chacune des pages de notre site.

3.1. **pageweb/Page1**

Elle correspond à la page d'accueil.



[Accueil.php](#) est la page principale, elle permet de naviguer dans les autres pages, et toutes les autres pages ont une redirection vers celle-ci.

Son fonctionnement est très simple, tout d'abord elle affiche la devanture de notre magasin en utilisant notamment le fichier css se trouvant dans le dossier parent.

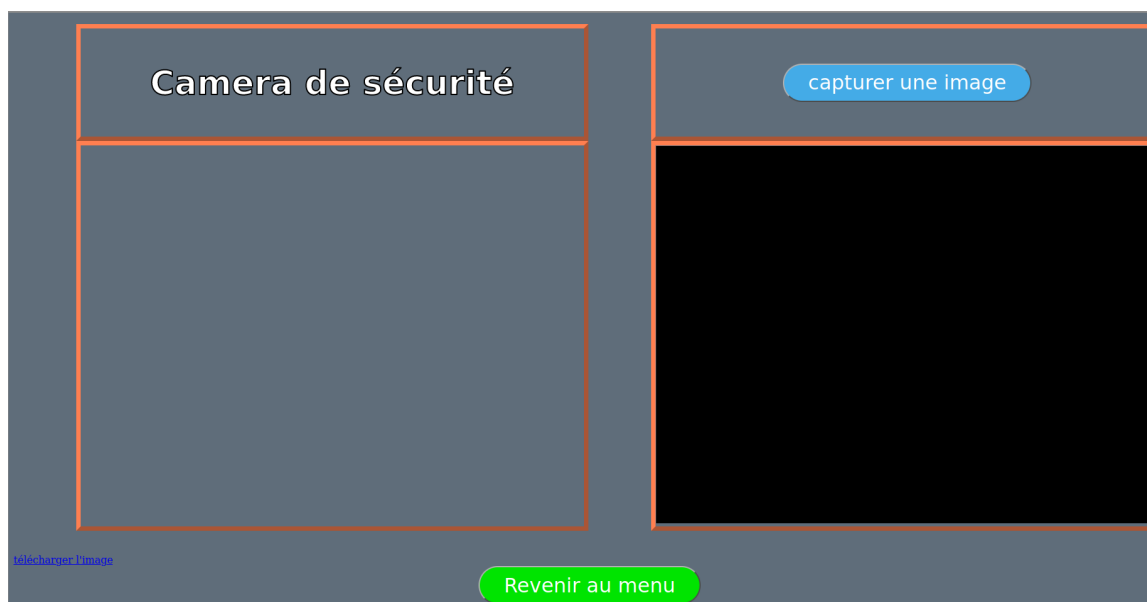
Nous disposons ici de 6 boutons, chacun menant vers une des autres pages.

Technologies utilisées :

- HTML5 / CSS
- JavaScript

3.2. [pageweb/Page2](#)

Elle correspond à la gestion de la caméra de sécurité.



Cette page affiche le flux de la caméra en direct à gauche de l'écran, et à sa droite on peut capturer une image, si on veut ensuite enregistrer cette image on clique sur le lien en bas à gauche.

L'image enregistrée est directement ajoutée au répertoire d'image, situé dans le dossier [src/image_recognition](#) afin d'être analysé.

Technologies utilisées:

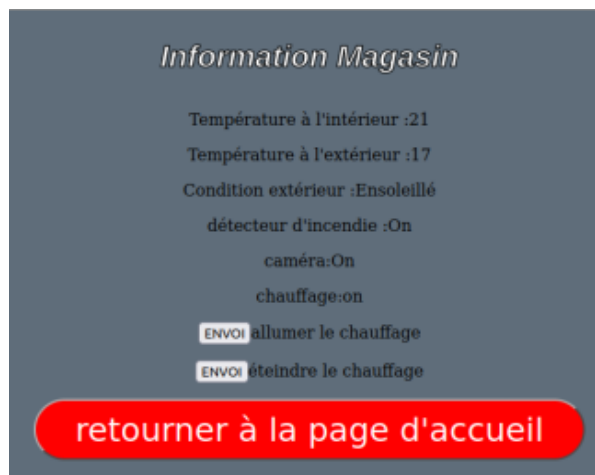
- HTML5 / CSS
- JavaScript

Type de données :

- Flux vidéo
- Images

3.3. pageweb/Page3

Elle permet d'obtenir les informations du magasin en temps réel.



Ces informations sont récupérées depuis Elasticsearch à l'indice shop, elle accède aux données grâce à l'outil `shell_exec()` disponible en PHP qui permet d'effectuer une commande dans le terminal de l'ordinateur.

On peut alors lancer des requêtes du type `shell_exec("curl-XPOST 'http://localhost:9200/shop/_update_by_query\' -H 'Content-Type: application/json\' -d'@queryON'");}` qui utilise le fichier [Page3/queryON](#). Ce fichier permet alors de modifier le champ heating à ON si il était à OFF auparavant, c'est-à-dire allumer le chauffage. De la même manière on utilisera [Page3/queryOFF](#) pour l'éteindre.

Pour changer la température, on l'augmente de 1 degré toutes les 2 minutes si le chauffage est allumé, sinon on la diminue de 1 grâce au script [data/genereData.sh](#).

Cette page récupère également en direct les données météo de Paris (ville de notre boutique fictive) à l'adresse <https://www.prevision-meteo.ch/services/json/Paris> pour afficher la température extérieure.

Technologies utilisées :

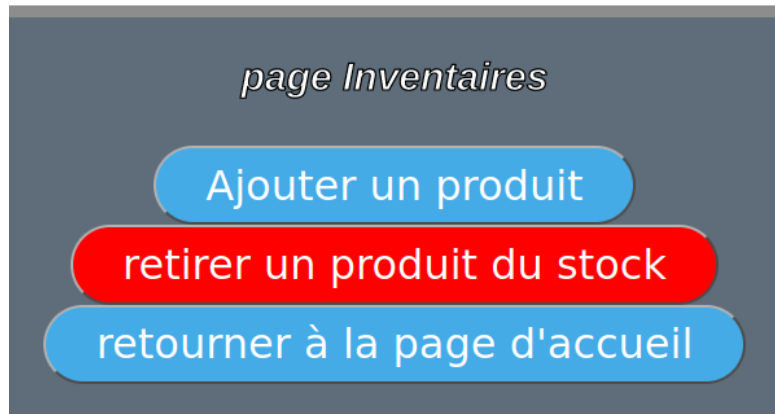
- HTML5 / CSS
- PHP
- Elasticsearch
- Shell (grâce à `shell_exec()`)

Type de données :

- .json

3.4. pageweb/Page4

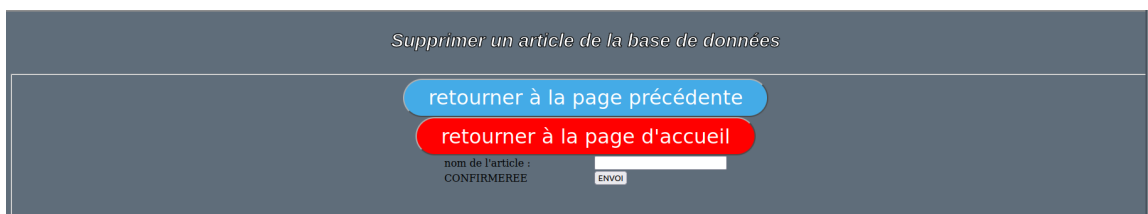
[pageInventory.html](#) est une page intermédiaire contenant 3 boutons permettant d'accéder au stock de la boutique.



page 4.1 est la page permettant d'ajouter un article au stock du magasin.



page 4.2 est la page permettant de supprimer un article au stock du magasin.



Pour ces 2 pages, de la même manière que pour la page 3, on utilise `shell_exec()` pour accéder au terminal depuis PHP et ainsi ajouter ou supprimer un indice à inventory.

Technologies utilisées :

- HTML5 / CSS
- PHP
- Elasticsearch
- Shell

3.5. pageweb/Page5

C'est la page permettant de choisir la musique à diffuser dans le magasin, au choix parmi les fichiers contenus dans **Page5/Songs**.



Il est ici également possible d'ajouter une musique à ce dossier à partir d'un bouton.

Technologies utilisées :

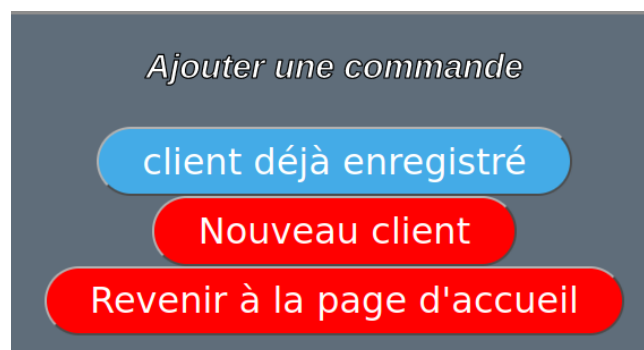
- HTML5 / CSS
- JavaScript

Type de données :

- .mp3

3.6. pageweb/Page6

ajouteCommande.html est une page intermédiaire contenant 2 boutons pour la création de nouvelles commandes.



page 6.1 est la page permettant d'ajouter une commande pour un client existant déjà dans la base de données.

The screenshot shows a web form titled "Ajouter une commande". At the top, there are two buttons: "retourner à la page précédente" (blue) and "retourner à la page d'accueil" (red). Below these buttons, the form fields are labeled: "Civilité :", "Nom :", "Prénom :", "article :", "quantité :", and "CONFIRMER". To the right of these labels are input fields. A small "ENVOI" button is located at the bottom right of the form.

page 6.2 est la page permettant d'ajouter une commande pour un client n'existant pas encore dans la base de données.

The screenshot shows a web form titled "Ajouter une commande". On the left side, there is a label "Saisissez vos coordonnées". At the top, there are two buttons: "retourner à la page précédente" (blue) and "retourner à la page d'accueil" (red). Below these buttons, the form fields are labeled: "Civilité :", "Nom :", "Prénom :", "Adresse mail:", "Num :", "article :", "quantité :", and "CONFIRMER". To the right of these labels are input fields. At the bottom right, there are two buttons: "ENVOI" and "VIDER".

La première page ajoute une commande pour le client indiqué, en retirant du stock la quantité demandée pour l'article.

- On vérifie d'abord que le client existe bien dans l'indice customer avec cette une requête ElasticSearch.
Si ce client n'existe pas alors on fait une redirection vers la page 6.2 avec les informations déjà pré remplis grâce aux variables de session PHP.
- Sinon on vérifie si la quantité demandée est disponible, si tel est le cas on l'indique à l'écran.
- On met ensuite à jour les points de fidélités pour le client.
- Pour finir on augmente le turnover (situé dans shopdailydata) de l'heure actuelle. Pour ce faire, on a besoin de récupérer l'heure et de faire une requête ElasticSearch.

La deuxième page fonctionne de la même manière mais permet également d'enregistrer le client s'il n'est pas encore dans la base de données.

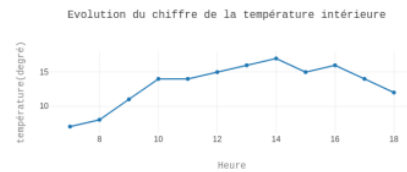
Technologies utilisées :

- HTML5 / CSS
- PHP
- ElasticSearch
- Shell

3.7. pageweb/Page7

Page permettant de visualiser les données de la base de données ElasticSearch.

indiqué la date : mm-dd\YY:



[retourner à la page d'accueil](#)



L'utilisateur donne une date dans le format indiqué et des graphiques sont créés pour représenter les données récupérées dans shopdailydata pour le jour choisi.

Technologies utilisées :

- HTML5 / CSS
- JavaScript (avec la librairie matplotlib.js)
- ElasticSearch
- Shell