

Rapport

Conception et Programmation des Interfaces

Application web de gestion d'agent de police : permettant à un agent de police d'avoir des informations sur des personnes arrêtées pour un Contrôle.

Encadré par : **Mme Hajer Ouerghi**

Etudiant : **Soumaya Dwihech**

Année Universitaire 2021-2022

Chapitre 1 :

Critère ergonomique

On a 8 critères principaux, commençant par :

- **Guidage :**

C'est un moyen mis en œuvre pour conseiller, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur (messages, labels, formats de saisie, etc.).

Il contient 4 sous critères :

- Incitation
- Groupement/distinction entre items
- Retour utilisateur
- Lisibilité

- **Charge de travail :**

Ce critère caractérisé par une réduction de la charge perceptive des utilisateurs et augmentation de l'efficacité du dialogue.

Il contient deux sous critères :

- Brièveté :
- Densité informationnelle

- **Contrôle explicite :**

Ce critère regroupe 2 aspects différents qui sont :

- La prise en compte par le système des actions des utilisateurs
- Le contrôle des utilisateurs sur le traitement de leurs actions

D'où, il se compose par deux sous critères qui sont :

- Actions explicites (Par exemple, ne pas faire des modifications dans un document sans accord de l'utilisateur.)
- Contrôle utilisateur (Par exemple, laisser la décision à l'utilisateur.)

- **Adaptabilité :**

C'est la capacité du système à réagir selon le contexte, et selon les besoins et préférences des utilisateurs.

Ici, on a deux sous critères qui sont :

- Flexibilité
- Prise en compte de l'expérience de l'utilisateur

- **Gestion des erreurs :**

On a 3 sous-critères participant à la gestion des erreurs :

- Protection contre les erreurs
- Qualité des messages d'erreurs (c'est-à-dire messages claires, compréhensibles montrant la cause et la solution.)
- Correction des erreurs

- **Homogénéité et cohérence :**

Ici, il faut que les choix de conception de l'interface soient conservés pour des contextes identiques et différents pour des contextes différents.

- **Significations des codes :**

C'est-à-dire qu'il ne faut pas donner à l'utilisateur un message d'erreur informatique par exemple « erreur du console », mais plutôt un message d'erreur clair et facile à comprendre tels que « username incorrect » ou « verify your login or password ».

- **Compatibilité :**

Ici, il faut prendre en compte les caractéristiques du public cible tels que l'âge, handicaps, niveau «technologique»...

Chapitre 2 :

Réalisation et Résultats

Introduction :

La phase de réalisation est une phase qui inclut la phase de programmation. Dans ce chapitre, je présenterai des recherches techniques, l'environnement de travail et les différents outils de développement (matériaux ou logiciels) réservés à la réalisation de ce mini projet, ainsi que quelques captures qui fournissent l'interface.

Environnement logiciel :

Les logiciels consacrés pour la réalisation de ce mini projet sont :

- **Visual Studio code :**

Microsoft Visual Studio est l'environnement de développement intégré (IDE) de Microsoft. J'ai choisi VS Code car il est utilisé pour développer des programmes informatiques ainsi que des sites web, des applications web, des services web et des applications mobiles ...



Figure 1 : Logo Visual Studio (VS)

- **Robot 3t :**

Robo 3T est un logiciel gratuit et convivial qui utilise une petite quantité de ressources disponibles sur une seule machine. J'ai choisi ce logiciel car il simplifie l'utilisation de la base de données MongoDB et fournit une interface graphique simple pour MongoDB. Le but d'utiliser le Robot3t est d'exécuter les requêtes de MongoDB.



Figure 2 : Logo Robot 3t

Cette figure ci dessous montre la base de données construite nommée «db_SOA» qui contient 2 classes (admin, users) :

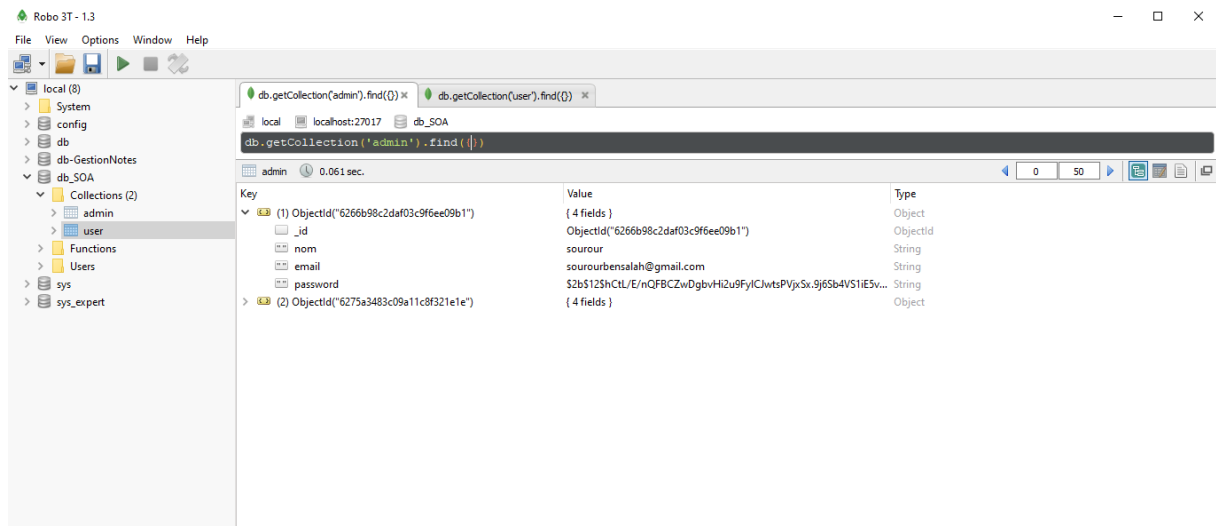


Figure 3 : BD «db_SOA»

- **MongoDB :**

MongoDB est un système de gestion de base de données orienté document, ce qui signifie qu'il stocke les données au format de document JSON. Je pense que c'est le moyen le plus naturel de visualiser les données et qu'il est plus efficace que les modèles traditionnels basés sur des lignes et des colonnes.



Figure 4 : Logo mongodb

Avant tous, il est nécessaire d'activer le serveur de mongodb par la commande « **mongod** » comme représente la figure suivante :



Figure 5 : Activation de serveur

- **Postman :**

Postman est un logiciel qui va nous permettre d'appeler et tester une API. Je choisis de l'utiliser pour tester les requêtes et voir les données retournées qui vont être utilisées dans le front end.



Figure 6 : Logo Postman

Environnement de développement web :

- **ReactJs :**

J'ai utilisé la flask et react pour l'application à développer. Tout d'abord, je veux présenter l'environnement de développement web « ReactJs » : ReactJs est une bibliothèque JavaScript pour créer des interfaces utilisateur. J'ai choisi cette bibliothèque car elle favorise la création d'applications Web basées sur le principe du DOM virtuel progressif en créant des composants dépendants de l'état et en générant des pages HTML à chaque changement d'état.

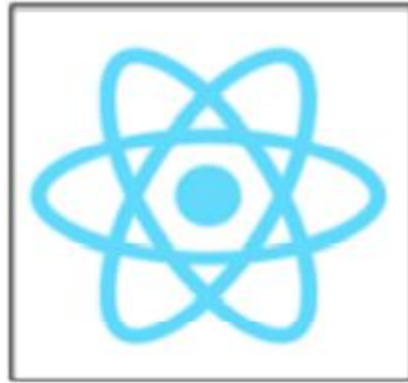


Figure 7 : Logo reactJs

- **Flask :**

Maintenant, je présente le backend qui est réalisé par le framework « Flask » : je veux créer mon application Web Python avec Flask. Flask est simple et facile à démarrer en tant que débutant. Flask est un framework de développement web Python open source. Son objectif principal est de maintenir la légèreté associée au système de modèles pour maintenir la flexibilité de la programmation Python.



Figure 8 : Logo Flask

Choix de langage utilisé :

- **Python V3.7.1 :**

J'ai choisi Python comme langage de programmation car c'est un langage de programmation d'objets, multi-paradigme et multiplateforme et il favorise la programmation impérative, structurée, fonctionnelle et orientée objet.



Figure 9 : Logo du langage de programmation python

Interface de l'application :

Dans cette région, je présente le résultat de la phase de réalisation à travers des captures d'écran illustrant les fonctionnalités de l'application finale.

- **Interface page d'accueil :**

La page d'accueil comme indiquée dans la figure ci-dessous est la première interface qui apparaît :



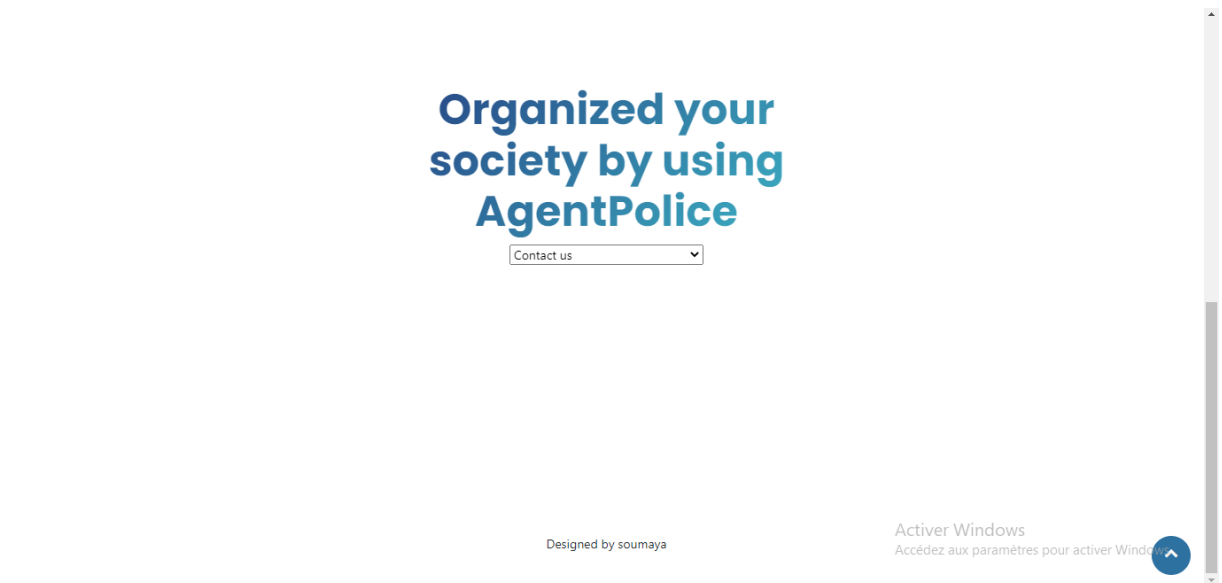


Figure 10 : Interface page d'accueil

Les critères ergonomiques respectés dans cette interface sont :

- * **Guidage par incitation** : la liste déroulante du menu est un exemple illustratif de guidage par incitation : nous fournissons la liste d'entrées attendue à l'utilisateur.
 - * **Guidage par lisibilité** : la perception des informations textuelles et iconographiques est aisée et claire ainsi que leurs propriétés et mises en page.
 - * **Charge de travail** :
 - **La brièveté et la densité des informations** : les informations mises à jour sur cette interface sont concises et limitées au maximum afin de réduire la charge de lecture de l'utilisateur.
 - * **Adaptabilité** :
 - **expérience utilisateur** : cette interface d'accueil respecte les différents niveaux d'expérience utilisateur.
 - * **Homogénéité et cohérence** : cette interface d'accueil respecte une logique cohérente de présentation (Espacements, tailles, Alignements, ponctuation, formulations...)
 - * **Compatibilité** : le jargon informatique est évité afin de mettre en valeur le logique métier
- **Interface «authentification» :**
La figure ci-dessous montre la partie d'authentification d'administrateur :

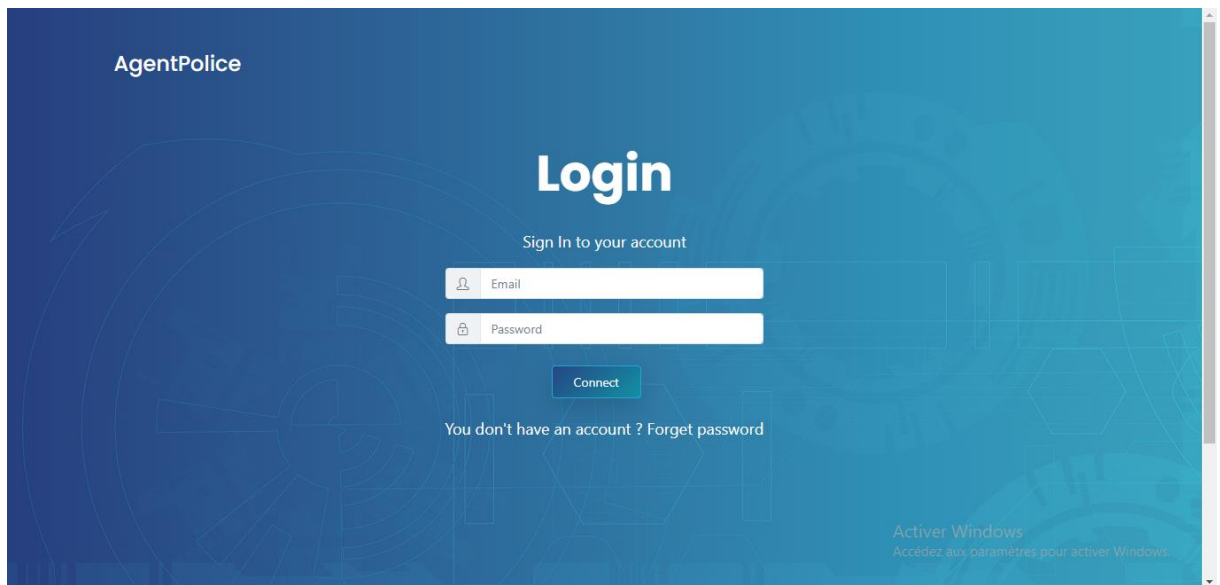


Figure 11 : Interface login

S'il est déjà un compte, il doit saisir son email et son mot de passe pour accéder facilement à son espace de l'application :

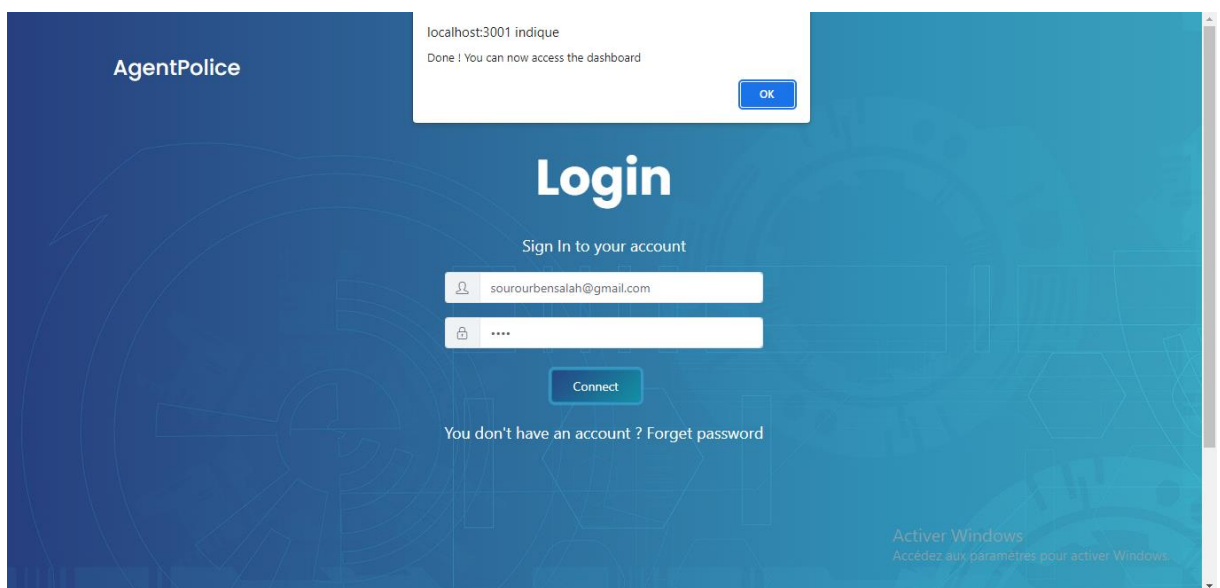


Figure 12 : Interface login correcte

Cette interface répond aux critères suivants :

*** Guidage par incitation :**

- indication des champs
- Réduction du risque d'erreur par le contrôle de saisie (messages d'aide)

*** Guidage par lisibilité**

- * **Guidage par retour utilisateur** : nous montrons à l'utilisateur ce que fait le système lors de l'enregistrement ou connexion (message d'alerte « done ! you can now access the dashboard »)
- * **Charge de travail** : brièveté et densité informationnelle
- * **Contrôle explicite** :
 - **Actions explicites** : une erreur dans les données inscrites dans les champs sera signalée à l'utilisateur mais jamais modifiée sans sa confirmation (figure 14)
- * **Adaptabilité** :
 - **expérience utilisateur**
- * **Homogénéité et cohérence**
- * **Compatibilité**

Si ses coordonnées sont erronées ou mal écrites, un message d'alerte s'affichera pour l'en informer et le guider pour corriger son erreur comme les indique les deux figures suivantes :

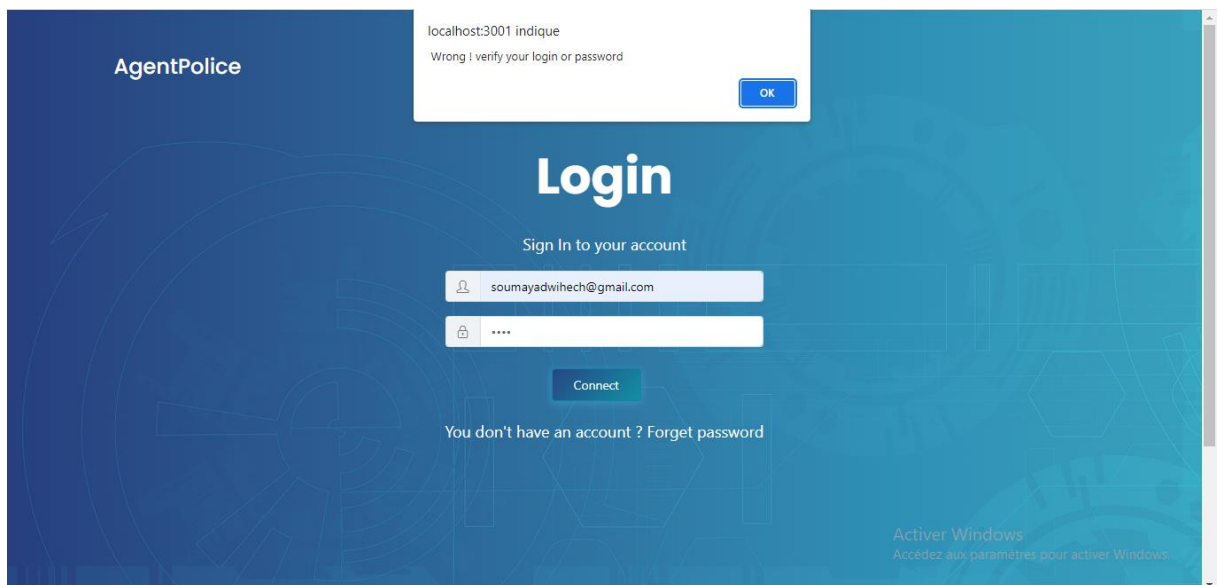


Figure 13 : Interface login incorrecte

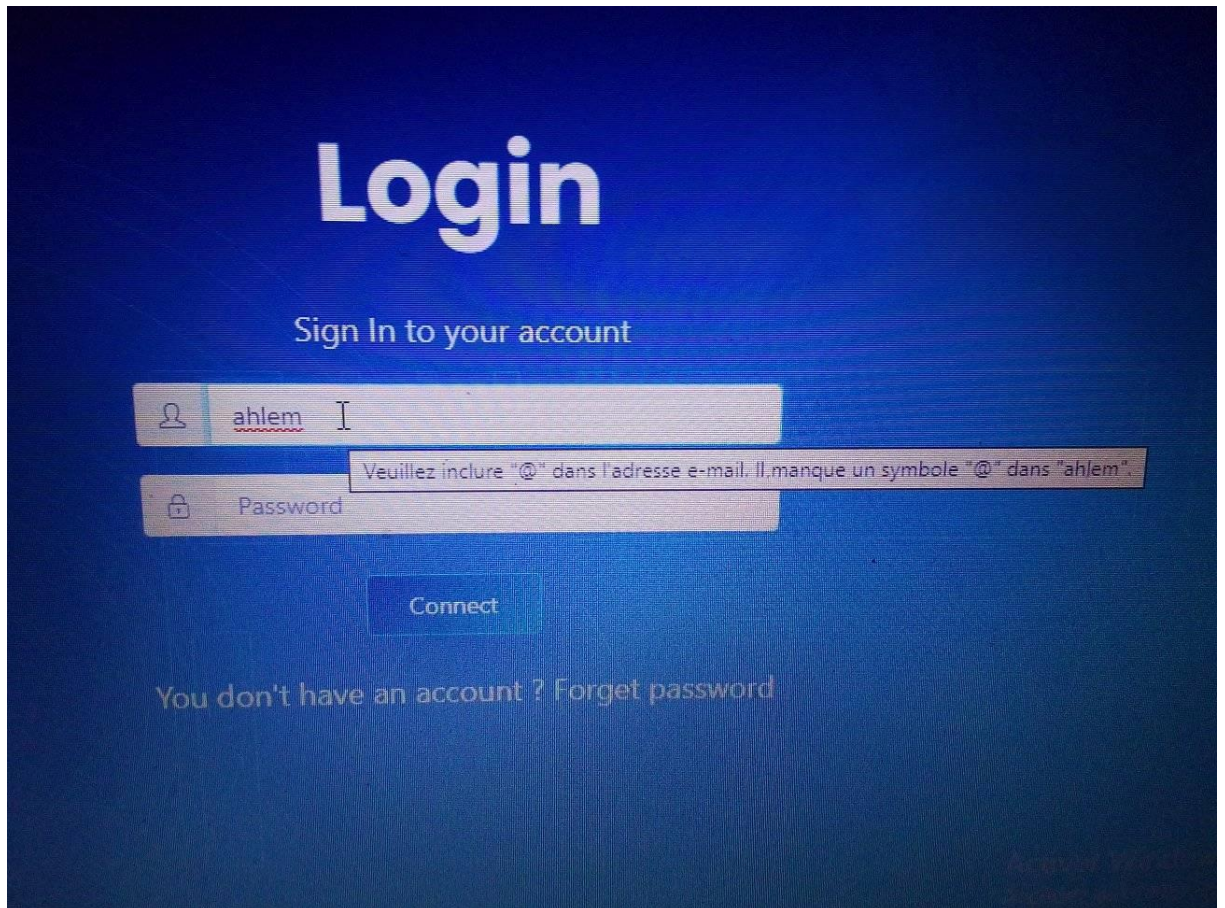


Figure 14 : Interface login mal écrite

De même, ces deux interfaces (figures 13 et 14) reprennent les mêmes critères que la précédente mais avec deux nouveaux critères ergonomiques qui sont :

*** Significations des codes (figure 13) :**

Ici, le message d'erreur est clair et facile à comprendre (« wrong ! verify your login or password ») d'où la correction d'erreur faite devient plus facile pour un utilisateur.

*** Gestion des erreurs :**

- **Qualité des messages d'erreur (figure 14) :** si une erreur est commise dans les données d'entrée, un contrôle de saisie est mis à la disposition de l'utilisateur afin de clarifier l'erreur tels que (« veuillez inclure @ »).

Dans le cas contraire, si l'utilisateur oublie son mot de passe, il suffit de cliquer sur « **forget mot de passe** » et il récupérera son mot de passe par e-mail. De même, s'il n'a pas de compte, il s'enregistrera en cliquant sur le lien « **You don't have an account ?** » :

The image shows a 'Sign up' form with the following elements:

- Title:** 'Sign up' in a large, bold, black font.
- Subtitle:** 'Create your account' in a smaller, gray font.
- Form Fields:**
 - Name:** A text input field with a person icon on the left and the text 'salma' inside.
 - Email:** A text input field with an '@' icon on the left and the text 'salmadwihech@gmail.com' inside.
 - Password:** A text input field with a lock icon on the left and four dots '....' inside.
 - Confirm Password:** A text input field with a lock icon on the left and four dots '....' inside. This field is highlighted with a blue border.
- Buttons:**
 - Register:** A blue button with white text.
 - Reset:** An orange button with white text.

Figure 15 : Interface register

Cette interface est caractérisée par les critères suivants :

* **Guidage par incitation :**

- indication des champs
- Réduction du risque d'erreur par le contrôle de saisie (messages d'aide)

* **Guidage par lisibilité**

* **Guidage par distinction entre 2 boutons :** on a fait la distinction entre les 2 boutons "register" et "reset" par la couleur et la position

* **Charge de travail :**

- brièveté et densité informationnelle

* **Contrôle explicite :**

- **Actions explicites :** une erreur dans les données inscrites dans les champs sera signalée à l'utilisateur mais jamais modifiée sans sa confirmation

* **Adaptabilité :**

- expérience utilisateur

* **Gestion des erreurs :**

- **qualité des messages d'erreur :** si une erreur est commise dans les données d'entrée, un contrôle de saisie est mis à la disposition de l'utilisateur afin de clarifier l'erreur

* **Homogénéité et consistance**

* **Compatibilité**

The image shows a web interface for user registration. At the top, a white modal box with a blue border displays the message "localhost:3001 indique Registered successfully" and an "OK" button. Below this, the main form has the heading "sign up" and the subtext "Create your account". It contains four input fields: a username field with the value "salma", an email field with "salmadwihech@gmail.com", and two password fields, both masked with four dots. At the bottom right of the form are two buttons: a blue "Register" button and an orange "Reset" button.

Figure 16 : Interface register avec succès

De même, cette interface représente les mêmes critères que la précédente (figure 15) mais avec un nouveau critère qui est :

* **Guidage par retour utilisateur :** nous montrons à l'utilisateur ce que fait le système lors de l'enregistrement (« registred successfully »)

Dès qu'il clique sur le bouton « **Connecter** », il se rend sur le tableau de bord (dashboard) :

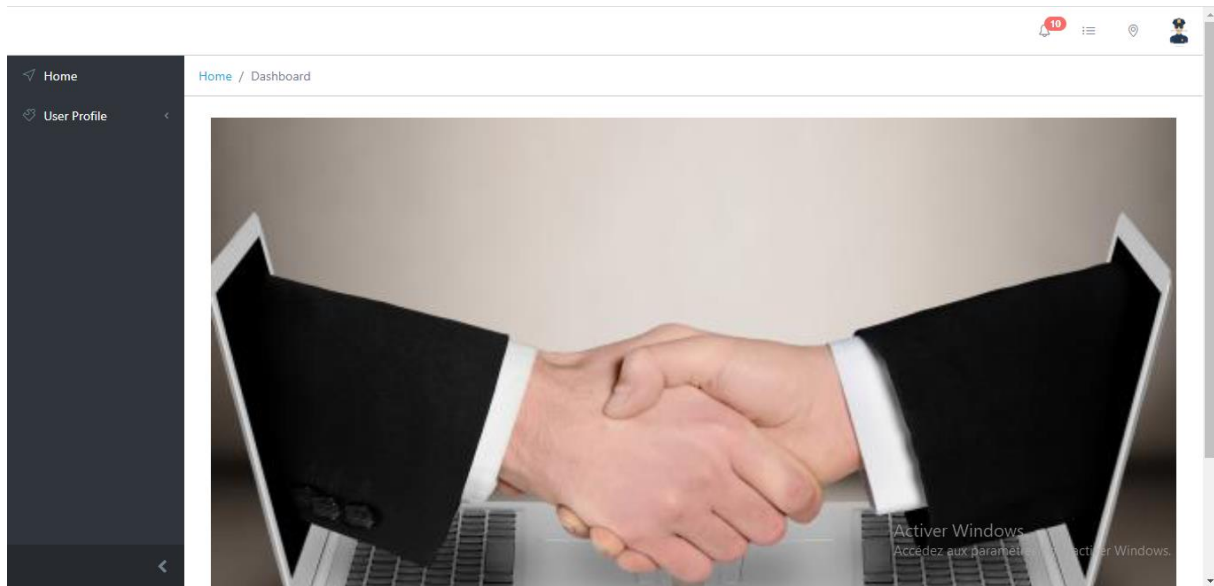


Figure 17 : Interface Dashboard

Cette interface représente les critères suivants :

- * **Charge de travail** : brièveté et densité informationnelle
- * **Homogénéité et consistance**
- * **Compatibilité**

Ici, le policier va mettre le numéro Cin d'un passager pour vérifier s'il est recherché ou non ?

Figure 18 : Interface recherche des informations avec message d'erreur

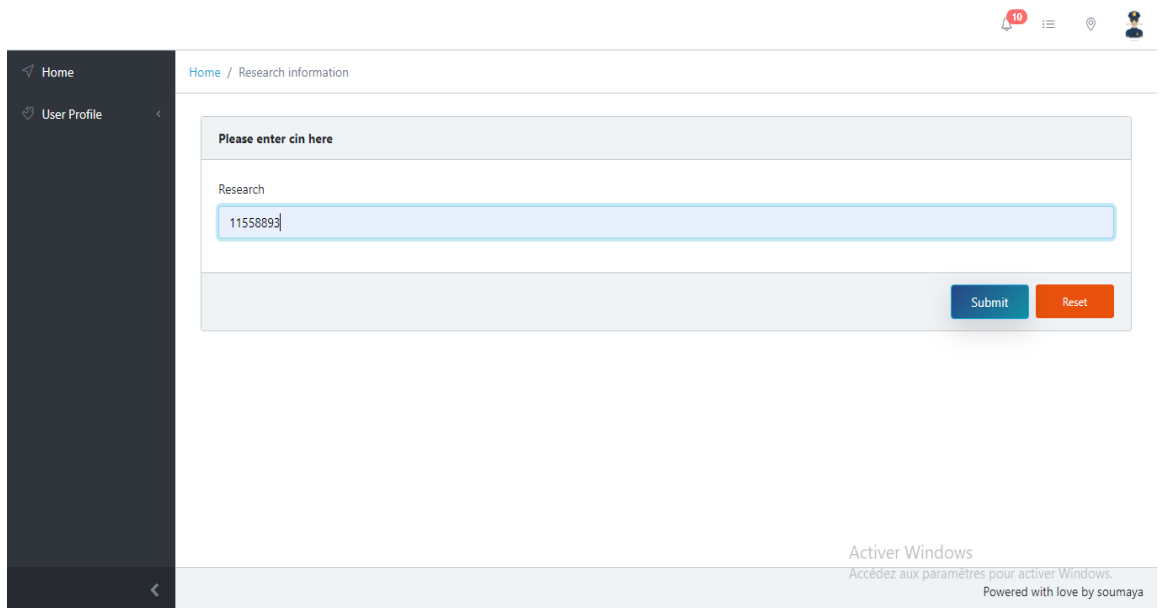


Figure 19 : Interface recherche des informations correctes

Ces deux interfaces représentent les critères suivants :

- * **Guidage par incitation :**

- indication des champs
- Réduction des risques d'erreur par le contrôle de saisie (les messages d'aides par exemple « **enter cin here** »)

- * **Guidage par lisibilité**

- * **Charge de travail :** Brièveté et Densité informationnelle

- * **Adaptabilité :** expérience de l'utilisateur

- * **Gestion des erreurs :**

- **qualité des messages d'erreurs :** si une erreur est faite au niveau des données de saisie, un contrôle de saisie pour le numéro de Cin est effectué «**Must not exceed 8 digits**» (Figure 18)

- * **Homogénéité et cohérence**

- * **Compatibilité**

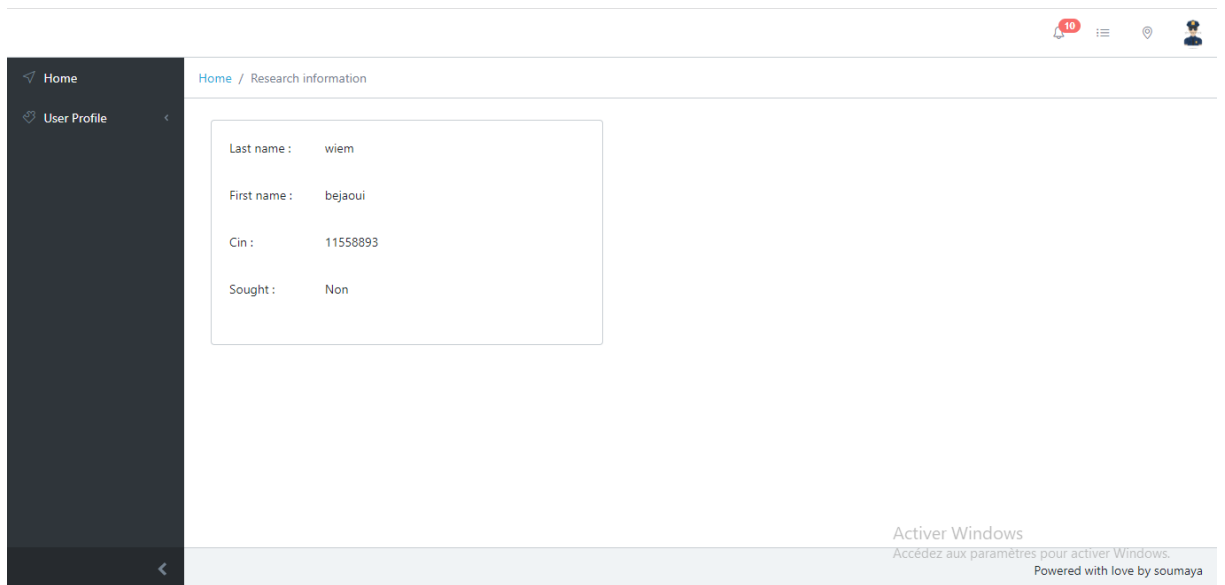


Figure 20 : Interface affichage des informations

Ici, l'interface est caractérisé par :

- * **Lisibilité**
- * **Charge de travail** : Brièveté et Densité informationnelle
- * **Homogénéité et cohérence**

Maintenant, si l'administrateur peut voir ses coordonnées ou va les modifier, il lui suffit d'accéder au «Admin profile» :

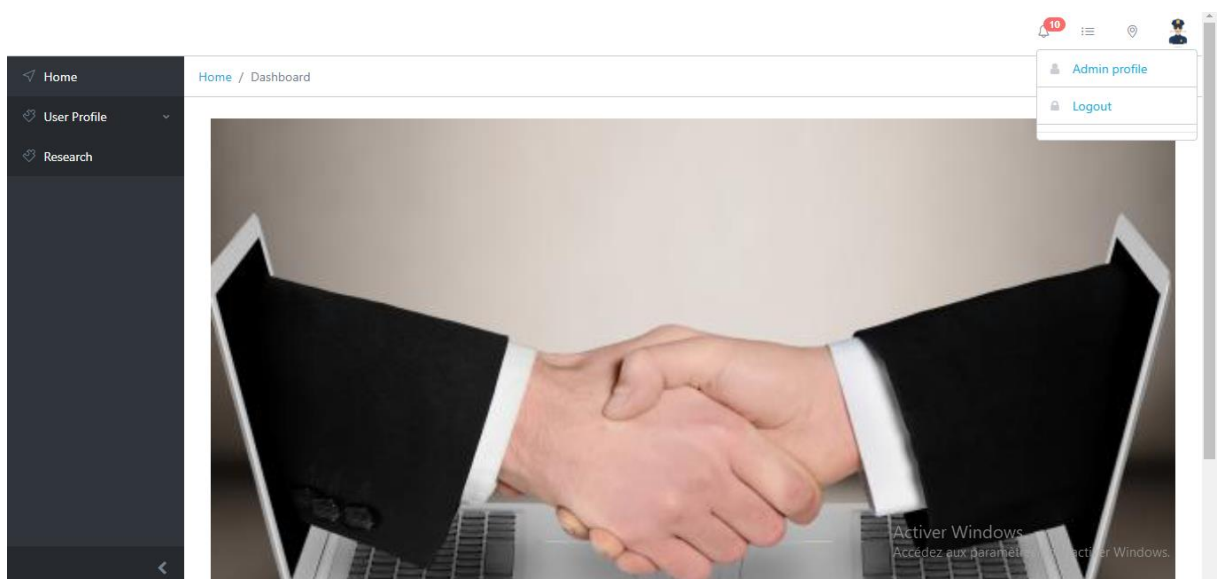


Figure 21 : Interface de profile d'admin

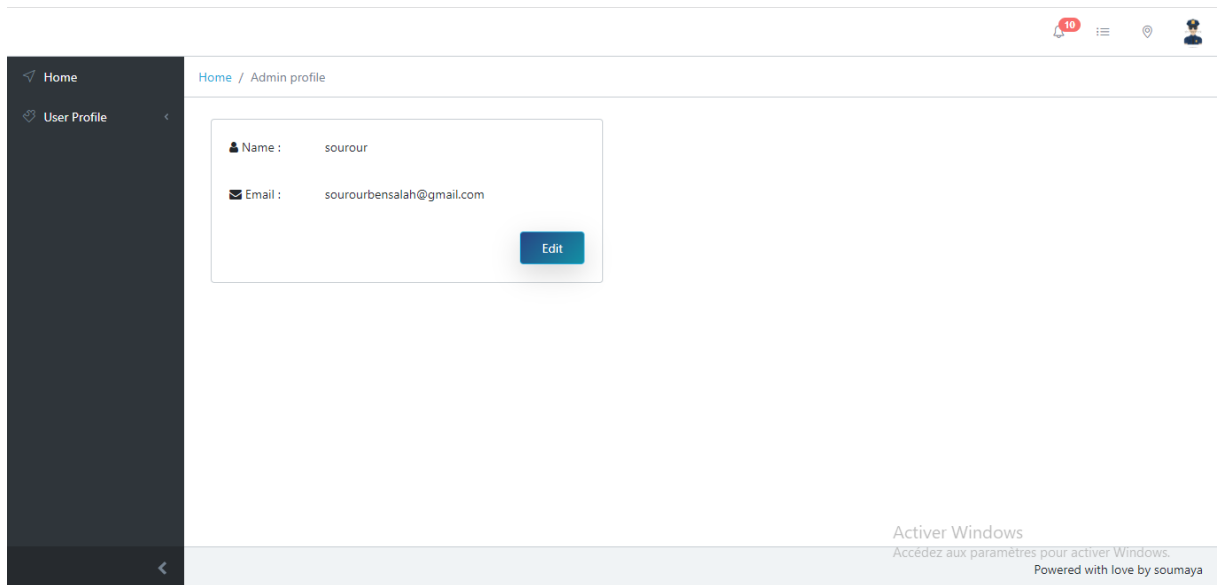


Figure 22 : Interface de profile d'admin

De même, cette interface se caractérise par :

- * **Lisibilité**
- * **Charge de travail** : Brièveté et Densité informationnelle
- * **Homogénéité et cohérence**

Et enfin pour déconnecter, il suffit de cliquer sur le bouton « logout » :

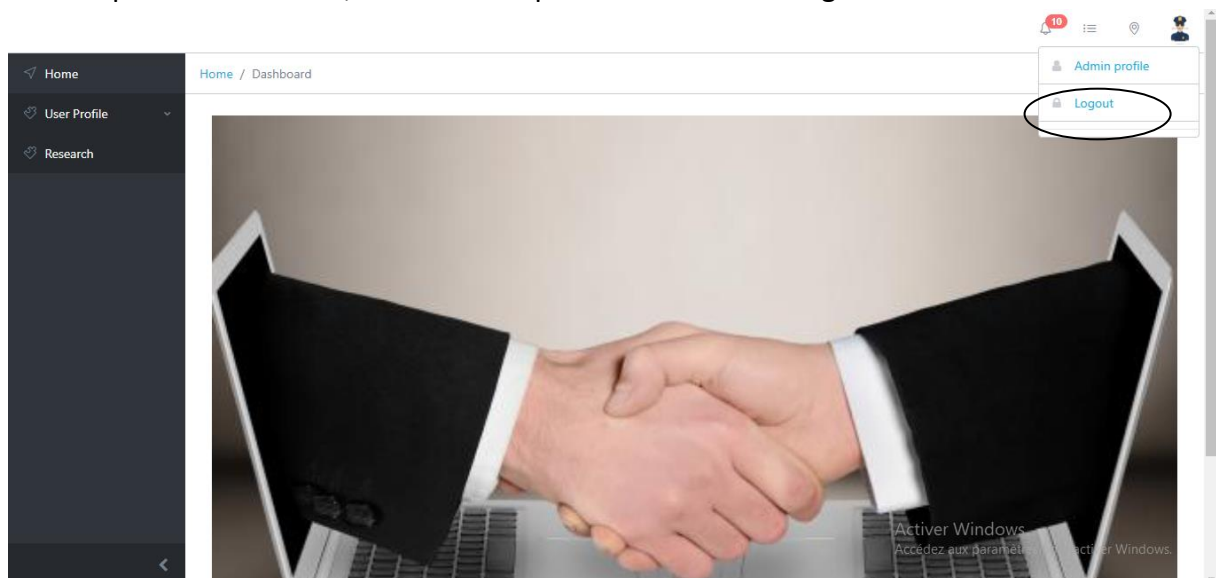


Figure 23 : Interface logout