 IAMI

*Confidential*

TODO FEATURES

## Table of contents

Table of Contents

[Executive Summary 2](#_Toc220247721)

[Objectives 2](#_Toc220247722)

[Background & Context 2](#_Toc220247723)

[Approach & Methodology 2](#_Toc220247724)

[Timeline 2](#_Toc220247725)

[Results & Deliverables 2](#_Toc220247726)

[Next Steps 2](#_Toc220247727)

# SECURITE

**Priorité CRITIQUE (à faire immédiatement)**

1. Implémenter le rate limiting sur les routes de login
2. Implémenter la protection brute force
3. Corriger la gestion de SECRET\_KEY (erreur si absente)

**Priorité HAUTE (sous 1 semaine)**

1. Appliquer les headers de sécurité dans Flask
2. Compléter la validation des mots de passe
3. Ajouter un verrouillage de compte après échecs répétés

**Priorité MOYENNE (sous 1 mois)**

1. Améliorer la CSP (nonces au lieu de unsafe-inline)
2. Ajouter un monitoring des tentatives de connexion suspectes
3. Implémenter un système d'alertes sécurité

**Priorité BASSE (améliorations continues)**

1. Rotation automatique des tokens
2. Audit de sécurité régulier
3. Tests de pénétration

# API VSA

* memberMetadata
  + Recuperer la Business Unit ?
    - Trouver une solution pour le staff interne
  + Recuperer un max d’informations supple2mentaires pour memberMetadata
  + Beaucoup d’email [elitys@elitys.com](mailto:elitys@elitys.com) d’où ça vient ?
    - **http://localhost:5001/api/CV/memberMetadata?filename=cand-22416\_Nomena\_RAMAROMANANA\_cv.pdf**
  + Beacoup de numéro de téléphone 0000000000 d’où ca vient ?
    - **http://localhost:5001/api/CV/memberMetadata?filename=cand-88827\_Zeshan\_BHATTI\_cv.pdf**
* Recupérer les missions passées et les missions en cours
* Opportunities
  + Explication : a quoi correspond ces dates ?
    - closingDate
    - startingDate
    - endingDate
* Opportunities
  + Statut d’un besoin -> sales\_stage\_code ( cree un mapping pour garder les memes choses que sur le site
* **RT\_Client-> RT**
* LOST -> Perdu
* WON-> Gagné
* QUALIFICATION-> Besoin reçu
* ABANDON-> Abandonné
* Stand\_by\_client -> Stand-by
* Envoi\_cv\_client->Dossier Transmis
  + Pour les opportunités il faut pouvoir récupérer le champs et le mettre dans manager exemple : "salesFullname":"LANDAIS Louis" => si le manager existe pas dans IAMI, on ecrit pas
  + Mapper avec es customers/prospects :
    - accountName (le nom )
    - accountType ( customer ou prospect )
  + entity = business unit
  + mapper pour recuperer le bon customer/prospects avec la clef accountCode qui correspond a tiersCode dans l’api pour les customers
  + Des CVs ne vont pas avec le candidat
    - <http://localhost:5001/api/CV/downloadCV?filename=int-1828_Sana_BAKARITHIELEN_cv.pdf>
  + Le salaire est annuel pour les candidats annuel mais il est mensuel pour les internes?
    - **http://localhost:5001/api/CV/memberMetadata?filename=cand-97006\_Sana\_ALLANI\_cv.pdf**
* customers et contacts
  + customers il faut recuperer le tiersCode qui permet de faire la jointure avce le customer pour le ctact (tiersCode (customers) meme id que Code (contacts))
  + contacts : Job =
  + "activity": "ENERGIE" => Secteur
  + Customer adresse
    - "mainAddr1": "33 Rue de l'Industrie,",
    - "mainAddr2": "",
    - "mainAddr3": "",
    - "mainZipcode": "74240",
    - "mainCity": "GAILLARD",
    - "mainCountry": "France",

# FEATURES

* créer prospect\_detail
* run\_matching\_competences\_from\_cv\_to\_besoin, la barre de recherche doit respecter le style central avec menu deroulant et filtre, pas de modale d’attente, le matching ne fonctionne pas
  + **Information**Erreur lors du matching : name 'Pool' is not defined
* Le bouton de suppression des besoins ne fonctionnent pas
* Dans @searchCv sur la carte quand je clique sur un bouton sur la carte le profil s’ouvre, ça fonctionne bien mais quand je clique pour le fermer, il y a 3 points de la modale d’attente et ils ne disparaissent jamais
* Les modifications recentes ont cassé les menus deroulants de @preferences.html
* reunionProspection
  + revoir le style
  + recuperation des CVs ne fonctionnent pas
* je veux supprimer dans la page @private\_platform/src/frontend/templates/cv/generer/genererCvFormaliseClassique.html la possibilité d'importer un pdf directement, je veux que ce soit seulement des candidats du site avec leur CV a eux autrement dit il faut toujours que le cv pdf viene d'un cv (profileCv)
* prospects
  + modale Filtre avancé ne fonctionne pas, je clique mais elle ne s’ouvre pas
  + Personnaliser les colonnes mn’a pas le bon style, regarde les autres style des modales pour recuperer le style centralisé
* Dans les modales Filtre avancé, j’aimerais qu’il y ait menu deroulant avec recherche en tapant comme le style centrale pour les users, les noms, les prenoms…
* Dans @besoinDetail j’ai une adresse comme ca 19 RUE JULES CARTERET, 69007, LYON, je clique sur Geocoder, et j’ai ce message, ❌ Erreur de géocodage: Aucun résultat trouvé pour cette adresse
  + Enlever le champs Budget
* AU moment du login, je veux le logo du client qui s’adapte au lien du client /login/<<lient>>, par defaut quand on se login depuis internet pas de logo
  + Meme comportement pour les mails de set up de mot de passe ou de reinitialisation
* Changer le logo du chatbot, reparer l’indexation des document dans @prefenrences
* Vue carte dans liste\_cv ame2liorer les filtres en haut en mettant des barres de recherches quand lesquels on peut taper pour filtrer
  + Pas de probleme de surcharge s’il y a bcp de CV geolocalisés ?
* Doublon CV trouver un moyen pour ne pas créer autant de surcharge
* Quand je clique dans pretention salariale dans profileCv sans rien remplir Erreur: Erreur lors de la sauvegarde: (pymysql.err.DataError) (1366, "Incorrect decimal value: '' for column 'pretention\_salariale' at row 1")

[SQL: UPDATE cvs SET pretention\_salariale=%(pretention\_salariale)s WHERE cvs.cv\_id = %(cvs\_cv\_id)s]

[parameters: {'pretention\_salariale': '', 'cvs\_cv\_id': 41723}]

(Background on this error at: <https://sqlalche.me/e/20/9h9h>)

* Dans @listeCV mettre un filtre superieure, inferieur sur le salaire
* Sur la page de comparaison je veux créer le scring de matching a ce moment la entre besoin et CV
  + aussi je veux que l’IA donne une recommendation sur quelles informations on peut donner au customer ou au besoin quand on lui presente le besoin ou le CV
* Revoir le template comparaison, ajouter l’appel a l’IA dedans
* centraliser la fonction find\_besoins\_by\_identifiers dans un file utils et l'utiliser dans tous les templates/route ou on en a besoin, supprimer toutes les fonctions qui font ca et qui peuvent etre centralisable
* le menu deroulant Business\_unit est vide dans filtre avance dans la page @listeCV
* regarder le statut candidat quand on synchronise depuis VSA
* dans editer\_CV menu deroulant des beosins, il faut voir que le titre du besoin et le statut c’est tout
* dans edit\_survey, il faut enlever tous les boutons ‘voir les reponses’ et corriger les styles de certains containers
* dans matching\_from\_CV\_to\_besoin.html mettre la barre de recherche et filtre en tapant en un seul object
* refonte de la generation de d’attestation d’emploi
* recuperation des PE avec VSA
* Finir la restructuratin des parties accessibles et leur acces
* Revoir la génération de rupture de PE avec le delais de prevoyance quand il aut 48h (poru faire du ouvré)
* Rajouter la date d’entrée du CV dans VSA
* Push des candidats concernant les besoins
* Voir la visibilité/missions,
* Liste inter contrat (ajouter le nombre de besoin), ajouter alerte et ajout todo list appeler ce besoin car mon intercontrat est apparu dans un besoin, hiearchie des todo est plus importante de laquel, todo alerte ‘
* Acces au compte, on a acces a tous les
* Continuer la modification de CV\_missions pour que la creation des missions soient coherente avec le nouveau format

## MATCHING COMPETENCES

Optimisations appliquées à group\_columns

Option 1 : Cache du mapping Excel (déjà implémenté)

* Problème : Le fichier Excel est relu à chaque appel
* Solution : Cache avec clé basée sur le chemin et la date de modification
* Gain estimé : 50-90% si appelé plusieurs fois

Option 2 : Vérifications O(1) avec set (déjà implémenté)

* Problème : c in df\_docs.columns est O(n) à chaque vérification
* Solution : Conversion en set pour des vérifications O(1)
* Gain estimé : 10-30% selon le nombre de colonnes

Option 3 : Opérations groupées (déjà implémenté)

* Problème : Opérations in-place dans une boucle fragmentent le DataFrame
* Solution : Préparer toutes les opérations, puis les appliquer en une fois avec pd.concat()
* Gain estimé : 50-80% selon la taille du DataFrame

Option 4 : Version vectorisée alternative (disponible)

* Fonction group\_columns\_vectorized() disponible
* Utilise un mapping colonne → nouvelle colonne pour éviter les recalculs
* Gain estimé : 20-40% supplémentaire pour très grands DataFrames

Option 5 : Précalculer le mapping une fois (recommandé)

* Si le mapping ne change pas souvent, le précalculer et le passer en paramètre
* Évite complètement la lecture Excel

Souhaitez-vous que je :

1. Teste la version optimisée actuelle ?
2. Crée une version qui accepte un mapping précalculé en paramètre ?
3. Ajoute des logs de timing pour mesurer l'amélioration ?

La version actuelle devrait être 3-5x plus rapide. Testez et dites-moi si vous voulez d'autres optimisations.

## LOGOS

AUDIT des logos, bien les ranger une fois pour toute

## Processus recrutement

* Informations liées au CV
* Paramétrisation
* Génération de CV
* Activité du CV
* Sourcing
  + Filtre par catégorie
  + Filtre sur CV complet
  + Filtration expérience
  + Filtration géographique
* Ajout de CV
  + Vérification de CV déjà existant
  + Matching au moment de l’upload
  + Recupération des Donner
  + Si CV déjà existant
    - Annuler
    - Accepter quand meme
    - Remplacer le CV dans un des profil

## Processus Business

* Ajout de besoin
* Matching