Chapitre 1 : Présentation du cadre de projet

Introduction

Dans ce chapitre nous commençons par présenter l’organisme d’accueil « Bee coders » au sein duquel j’ai effectué ce projet de fin d’études par la suite je vais introduire les solutions existantes et leurs limites et finir par présenter la solution proposée.

1. **Présentation de l’organisme d’accueil**

Beecoders, fondée en 2020, est une entreprise de services numériques ayant achevée plus que 500 projets et plus que 1400 heures de support dans le secteur de consulting.

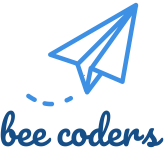
 Beecoders est spécialisée dans le développement web et mobile des solutions digitales répondant aux besoins du client (Android et IOS), le design web, le consulting IT pour booster les performances des employés dans le secteur d’IT et les formations à distance à travers la plateforme « Bridge ».

Figure 1 : Logo Bee coders

1. **Etude de l’existant**

Dans le sillage de la pandémie mondiale, le recrutement à distance est une pratique   
inévitable. L'objectif des entreprises est de pouvoir maintenir ses activités face aux bouleversements du monde du travail. Pour accompagner cette évolution du recrutement et optimiser sa recherche d'emploi, les demandeurs d'emploi ont également l'obligation de savoir manipuler les canaux digitaux. Pour y voir plus clair, nous allons exposer l'évolution du e-recrutement en Tunisie, ainsi que les sites de recrutement les plus connus.

De nombreuses entreprises prennent l'habitude d'embaucher des employeurs en ligne pendant la vague de coronavirus. Les entreprises placent des offres d'emploi sur de nombreuses plateformes pour cibler les meilleurs talents selon leurs propres conditions. De plus, la dématérialisation du recrutement facilite certaines activités qui peuvent être réalisées en freelance ou à domicile. La crise sanitaire a également facilité le passage au e-recrutement ou recrutement en ligne.

Ceux qui envisagent de recruter publient leurs offres d'emploi sur plusieurs portails de recrutement en ligne et même dans des journaux locaux pour découvrir des candidats dignes de ce nom. Le processus de recrutement peut se dérouler en plusieurs étapes : tri des lettres de motivation et CV, tests, entretiens en visioconférence pour présélectionner les candidats et exprimer les attentes des recruteurs, formation aux nouveaux outils à distance.

En conséquence, le recrutement en ligne peut mieux cibler les candidats avec des offres d`emploi largement diffusées, réduisant ainsi la charge de travail des recruteurs. Les offres d'emploi sont également accessibles 24 heures sur 24, 7 jours sur 7 et peuvent être mises à jour.

En Tunisie par exemple, on retrouve un certain nombre de plateformes qui exploitent le potentiel d’internet dans la recherche du candidat idéal. Citons entre autres emploi.nat, keejob, Tanitjob, Tunisie travail, Linkedin. Ces plateformes de recrutement en ligne offre des centaines de nouvelles offres chaque jour.

Citons par exemple :

* **La plateforme emploi.nat.tn :**C’est le site d’offres d’emploi de l’Etat ou du réseau de l’ANETI. Il propose des offres et demandes d’emploi, des avis de concours, des annonces de presse, des appels d’offres et résultats, des programmes d’encouragement à l’emploi, de présentation de l’ANETI, projet BAD, coopération internationale etc. On peut également y trouver un espace candidat, un espace promoteur et un espace entreprise ainsi qu’un espace pour les investisseurs et le travail indépendant. Les membres disposent d’un login et mot de passe pour accéder au site.
* [**Keejob**](https://www.tanitjobs.com/blog/355/les-sitesd-emploi-en-tunisie/)**:** Ce site d’offres d’emploi propose également un espace candidat et espace entreprise avec inscription. Les candidats peuvent mettre leur CV sur le site et paramétrer leur compte. Les entreprises peuvent aussi gérer leurs annonces et recevoir des candidatures
* [**Tunisie-travail.net**](https://www.tanitjobs.com/blog/355/les-sitesd-emploi-en-tunisie/)**:** Cette portail d’offres d’emploi diffuse des concours de la fonction publique, des offres d’emploi actualisées en Tunisie, en freelance et à l’étranger. Il permet aux recruteurs de publier gratuitement leurs offres d’emploi.
* [**Tanitjobs**](https://www.tanitjobs.com/)**:** Un site d’offres d’emploi spécialisé. Les candidats peuvent déposer leur CV sur le site. Les recruteurs peuvent aussi s’inscrire sur le site pour dénicher rapidement le profil en adéquation à leurs besoins. Le site propose une expérience utilisateur satisfaisante en permettant à ce dernier de trier parmi beaucoup de critères l’offre adaptée à ses besoins. Les offres se déclinent aussi en diverses catégories. Le site est mis à jour d'une façon régulière.
* [**Linkedin**](https://www.tanitjobs.com/)**:** Un site de réseautage professionnel. c'est tout à la fois un réseau social et un site d’emploi. Vous l'utilisez pour rester en contact avec vos relations professionnelles et rechercher de nouvelles opportunités de carrière. Mais pour les recruteurs, LinkedIn est un moteur de recherche plus qu'autre chose.

1. **Problématique**

La grande majorité des sociétés qui recrutent décident de contacter les anciens employeurs indiqués dans le CV, surtout s’il s’agit d’un poste à responsabilité. Cela leur permet de vérifier l’authenticité des renseignements sur les profils et le comportement du candidat. Il y a également des postes qui imposent le test psychotechnique ou le test graphologique. L’autre problématique de ce processus,

C’est que les chercheurs d’emplois se démotivent à faire de recherche sur le net dès qu’ils ont envoyé quelques demandes sans suite. Ils se souviennent de l’entreprise qui ne répond pas aux demandes.

Ce type de recrutement a aussi le désavantage d’être moins efficace sur la fidélisation des candidats embauchés notamment s’ils sont des jeunes de la génération Y et Z. Du fait de leur maîtrise des nouvelles technologies et de leur créativité, ils ne restent pas longtemps dans une entreprise.

En outre, l’inexistence de statistiques sur ces sites d’embauche se traduit par le manque d’informations nécessaires créant ainsi un besoin à étudier. En effet les données publiées sont stockées et non analysées, à travers des statistiques primordiales les professionnels seront capables de se différencier et se positionner dans le marché ce qui n’est pas possible sans analyse.

1. **Critique de l’existant**

La dispersion des données sur plusieurs sites et le manque d’analyse au sein de ses sites nous a permis de dégager les défaillances suivantes :

* Perte de temps : Le fait que les offres d’emploi sont éparpillées, il est difficile de les accéder toutes à la fois. Par conséquence on a une nécessité de faire le tour des offres disponibles sur le marché ce qui entraine une énorme perte de temps pour les étudiants et les professionnels.
* Manque de visibilité globale sur le marché : Vu que certaines entreprises choisissent de publier leurs offres sur un site bien précis, les chercheurs risquent par conséquent un accès limité d’offres.

* Inexistence d’analyses :Les utilisateurs ne sont pas capables de se positionner sur le marché à cause de l’absence d’analyse et statistiques des offres sur ses sites

1. **Solution proposée**

Après une étude de l’existant, j’ai proposé une collecte de données nécessaires qui sont dispersées sur différents sites. Les publications de chaque site ont été récoltées en appliquant sur ces derniers la méthode WEB SCRAPING.

Par suite un traitement a été effectué sur ces données dans le but d'offrir une information qui est de meilleure qualité pour une utilisation ultérieure.

Ces données ont abouti à une aptitude de repérer en temps réel les KPI (indicateurs de performance) des propositions d’embauche afin de suivre les éléments clefs.

Cette distinction facilite la recherche optimale pour le profil du demandeur.

En effet ces KPI construisent un Tableau de bord qui est un outil de pilotage aidant à la prise de décision, dans le but de mettre des analyses complexes à la portée des professionnels et des étudiants car ceux-ci n'ont pas toujours la capacité d'interpréter la data brute qu’est les offres d’emploi.

Une représentation simple et générale est de la sorte idéale pour ces professionnels. Tel qu’énoncé dans cette partie je propose cette solution qui assure la pertinence des données, l’efficience et la rapidité de l’affaire de recherche d’emploi.

1. **Méthodologie de travail**

La méthodologie Agile est une méthode de gestion de projet basée sur un cycle de développement centré sur le client. Le client est impliqué dans la mise en œuvre du début à la fin du projet. Grâce aux méthodes agiles, les demandeurs bénéficient d'une meilleure visibilité sur la gestion des tâches qu'avec les méthodes traditionnelles. L'influence du client dans le processus permet à l'équipe d'obtenir des commentaires réguliers afin que les changements nécessaires puissent être appliqués directement.

Le principe de base est de fournir une version minimale du logiciel puis par un processus itératif d'intégrer des fonctionnalités supplémentaires dans cette base. Le processus itératif regroupe une série d'instructions, se répétant autant de fois que nécessaire selon le besoin.

* 1. **Pourquoi SCRUM**

Scrum est une méthodologie de développement agile utilisée dans le développement de logiciels basée sur un processus itératif et incrémental.

L'objectif principal de Scrum est de satisfaire le besoin du client à travers un environnement de transparence dans la communication, la responsabilité collective et le progrès continu. Le développement part d'une idée générale de ce qu'il faut construire, en élaborant une liste de caractéristiques ordonnées par priorité (product backlog) que le propriétaire du produit souhaite obtenir.

* + 1. **Les acteurs du SCRUM**

**Product Owner**

Le *product owner* (PO) est le représentant des clients et utilisateurs dans le cadre du projet. Il est en charge de la tenue du backlog produit, notamment de l'ordre dans lequel vont être traitées les éléments de ce backlog (user stories, fonctionnalités à développer). En d'autres termes, il est « responsable » du succès du produit.

**Scrum Master**

Le scrum master est l’animateur ou le coach de l’équipe technique, chargé principalement de définir les rôles et les objectifs en assurant que le principe scrum se déroule convenablement.

**L’équipe de développement**

L'équipe de développement est responsable de la conversion des exigences définies par le propriétaire du produit en fonctions utilisables. Il est multidisciplinaire et possède toutes les compétences nécessaires pour exécuter le projet sans avoir besoin de faire appel à des services externes.

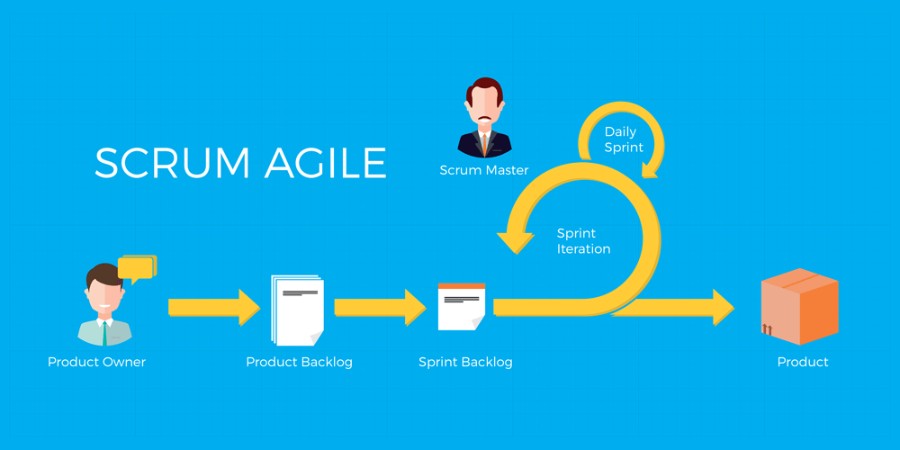


Figure 2: Cycle de vie projet Scrum

* 1. **Méthodologie de conception**

L’UML est une abréviation de Unified Modeling Language, est un langage de modélisation standardisé composé d'un ensemble intégré de diagrammes, développé pour aider les développeurs de systèmes et de logiciels à spécifier, visualiser, construire et documenter la conception des systèmes logiciels, ainsi que pour la modélisation commerciale et d'autres systèmes non logiciels.

L'UML représente un ensemble de meilleures pratiques d'ingénierie qui ont fait leurs preuves dans la modélisation de systèmes vastes et complexes. L'UML est une partie très importante du processus de développement de logiciels. Il utilise principalement des notations graphiques pour exprimer la conception de projets logiciels.

* 1. **Conclusion**

Dans ce chapitre, j’ai présenté l’organisme d’accueil, critiqué l’existant et ses limites et présenté la solution que j’ai envisagée.

Pour conclure je vise la conception de cette solution qui pourra remédier aux éléments existants et aux limites étudiés activant ainsi l’accès à un modèle d’analyse important pour les professionnels.

Dans le chapitre suivant, je vais étudier les besoins et la conception de mon système.

Chapitre 2 : Etat de l’art

**Introduction**

Dans ce chapitre on va présenter le web scraping en citant son histoire, ces outils et conclure par identifier la solution proposée dans ce projet

1. **l’informatique décisionnelle**

L’informatique décisionnelle est une branche de l’informatique de plus en plus adoptée par les entreprises. Connue également sous l’appellation Business Intelligence (et souvent BI) en anglais, elle désigne l’ensemble des méthodes et outils informatiques utilisés généralement pour piloter une organisation. Le meilleur exemple de solution d’informatique décisionnelle est le tableau de bord.

Ainsi, L’informatique décisionnelle rassemble une variété d'outils, d'applications et de méthodologies pour collecter des données à partir de source internes et externes, les préparer pour l'analyse, développer des données et lancer des requêtes sur ces ensembles de données. Ensuite on utilise ces outils pour créer des rapports, des tableaux de bord et des visualisations de données afin de mettre les analyses à la disposition des décideurs.

La capacité d'analyse des données est de plus en plus importante pour les entreprises. Le rythme du changement exige que les entreprises soient capables de réagir rapidement aux demandes changeantes des clients et des conditions environnementales.

1. **Web scraping**

A l'ère du big data, la collecte de données (textes, images, fichiers hébergés) sur Internet devrait être au centre de toutes les réflexions pour les années à venir. Le web scraping ou le screen scraping est l'un des meilleurs outils pour collecter et analyser les données d'entreprise. C’est le processus d’extraction des données des sites web. Ce sont des scripts, des programmes informatiques, qui sont chargés d'extraire des informations. D’abord cette technique permet de réutiliser le contenue déjà utilisé dans un site web dans le but de l'afficher sur un autre en multipliant par conséquence sans effort le nombre de page disposant d’un même contenu. En addition, l’augmentation exponentielle du marché de e-commerce entraine la nécessité d’être bien informé sur les concurrents et dans ce sens le web scraping peut être utilisé comme un outil de surveillance des concurrents (veille concurrentielle).

* 1. **L’histoire du Web scraping**

Le Web scraping ou Web harvesting est une technique d'indexation. La plupart l'associent à l'extraction de données Web, le moyen le plus efficace et le plus simple de copier de gros volumes d'informations en ligne. Cependant, le web scraping est né dans un but complètement différent et qu'il a fallu près de deux décennies pour qu'il se transforme en web scraping que nous connaissons maintenant.

* + 1. **La création de Wanderer**

En 1993, le premier concept de Crawler était né .World Wide Web Wanderer développé par Matthew Gray a l’institut de technologies de Massachussets était le premier robot d'exploration Web basé sur Perl dont le seul but était de mesurer la taille du Web. De plus Wanderer a été utilisé pour générer un index appelé Wandex qui lui a donné le potentiel de devenir le premier moteur de recherche universel du World Wide Web.

* + 1. **La création de JumpStation**

Dans la même année 1993, JumpStation était né et est devenu le premier moteur de recherche Web basé sur un robot. Grâce à lui, des millions de pages Web ont été indexées de sorte qu’Internet s'est transformé en une plate-forme open source de données sous diverses formes.

* + 1. **La création de BeautifulSoup**

En 2004, l’HTML parser BeautifulSoup a été créer, c’est une bibliothèque d'algorithmes couramment utilisés écrits en langage de programmation Python. BeautifulSoup a aidé à saisir le sens de la structure du site et à analyser le contenu dans les conteneurs HTML ce qui facilite au programmeur d’économiser des heures de travail. Internet est devenu cette immense source d'informations à laquelle toute personne disposant d'un ordinateur et d'une connexion Internet avait accès, les internautes avaient commencé à en tirer parti en extrayant les informations dont ils disposaient.

En effet, la quantité de données téléchargées de nos jours ne peut pas être manuellement copiée ou collée ce qui exige que d'autres moyens d'obtention d'information devaient être développés**.**

* + 1. **L’apparition des Web Scrapers visuels**

Estimé en 2006, le web scraping tel que nous le connaissons était né. Le logiciel de Web scraping visuel Web Integration Platform version 6.0 qui a été lancé par Stefan Andresen, a permis aux utilisateurs de mettre en évidence les informations nécessaires d'une page Web et de structurer ces données dans un fichier utilisable ou une base de données permettant aux non-programmeurs de rejoindre et extraire facilement des données du Web.

* 1. **Les outils du Web scraping**

Il existe plusieurs types d'outils qui peuvent être utilisés pour effectuer du web scraping: Des outils open source qui réduisent le coût du web scraping et facilitent son exécution. Les outils les plus couramment utilisés sont Scrapy, Selenium, BeautifulSoup et Puppeteer

Les bibliothèques les plus couramment utilisées par les développeurs de Web scraping avec le langage Python sont Beautiful Soup, Scrapy et Selenium, mais chaque bibliothèque a ses propres avantages et inconvénients, et la perfection n'existe pas dans ce monde. Afin d'expliquer les différents aspects de chaque bibliothèque et ses différences on va les définir :

* Beautiful Soup est une bibliothèque Python de grattage Web pour extraire des données de fichiers HTML et XML, qui prend également en charge Python 3 et Python 2.4.
* Scrapy est un framework Python pour le scraping Web à grande échelle. Il vous donne tous les outils dont vous avez besoin pour extraire efficacement les données de votre site Web, les traiter comme vous le souhaitez et les stocker dans la structure et le format de votre choix.
* Selenium est un outil open source pour automatiser les navigateurs Web. Il fournit une interface unique qui vous permet d'écrire des scripts de test dans des langages de programmation tels que Ruby, Java, NodeJS, PHP, Perl, Python et C#.

Dans le tableau (citant la ref), nous avons représenté une étude comparative selon 3 axes : l’extension, la performance de chaque outil et l’écosystème

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Extension** | **Performance** | **Ecosystème** |
| **Scrapy** | Utiliser pour les projets larges et complexes | Les bibliothèques existantes ne peuvent pas battre les performances de Scrapy grâce à sa fonctionnalité intégrée, à savoir l'utilisation d'appels système asynchrones. | Ayant un bon écosystème, nous pouvons utiliser des proxys et des VPN pour automatiser la tâche.  C'est l'une des raisons du choix de la bibliothèque pour les projets complexes. |
| **Beautiful Soup** | Utiliser pour les petits projets et à basse complexité | Beautiful Soup est assez lent pour effectuer une certaine tâche, mais nous pouvons surmonter ce problème à l'aide du concept de multithreading | Cette bibliothèque a beaucoup de dépendances dans l'écosystème.  C'est l'un des inconvénients de cette bibliothèque pour un projet complexe. |
| **Selenium** | Utiliser pour une taille de donnés limitée | Il peut gérer jusqu'à une certaine plage mais n'est pas équivalent à Scrapy. | Caractérisé par un bon écosystème pour le développement, mais les proxys ne peuvent pas être utilisés très facilement. |

Tableau 1: Etude comparative des bibliothèques

1. **Visualisation des données**

Face à des quantités de données importante, il est important d’utiliser les outils pour visualiser les données afin de les rendre compréhensibles. , La Data Virtualisation est une étape majeure du processus BI. En effet le rôle de cette étape est d’[éclairer les informations récoltées](https://ia-data-analytics.fr/solution-bi/datavisualisation/datavisualisation-role-clef-data-scientist/) en les traduisant par des représentations visuelles claires et accessibles à tous. Contrairement à des données brutes ou des tableurs complexes remplis de données, la Data Visualisation va permettre de créer des visuels qui permettent de comprendre les informations essentielles rapidement.

* 1. **Création de tableau de bord**

Les tableaux de bord sont des vues d'informations et de données qui vous permettent de surveiller les événements, de prendre des décisions, d'informer les autres et de visualiser les tendances. Les tableaux de bord sont conçus pour afficher plusieurs visualisations fonctionnant ensemble sur un seul écran. Ils fournissent une vue complète de vos données et fournissent des informations pertinentes pour aider à la prise de décision.

Les tableaux de bord de gestion sont conçus pour présenter des données précises et contextuelles pour guider les actions ou les décisions. Sa mise en place peut être longue et source de bugs si elle n'est pas bien pensée. Pour éviter ces erreurs, une approche adaptée est nécessaire. D’abord il faut identifier l’audience et les besoins pour définir la finalité du tableau de bord. Ensuite, on définit les indicateurs clés de performance (la traduction française des KPI) en déterminant les questions auxquelles on veut rependre. Par la suite, on passe à l’analyse et traitement des données pour les préparer à la dernière étape qui est l’élaboration du tableau de bord qui consiste à choisir la structure et les graphiques nécessaires pour la visualisation.

* + 1. **Tableau comparatif des logiciels de création des tableaux de bords**

Tableau 2: Etude comparative des logiciels

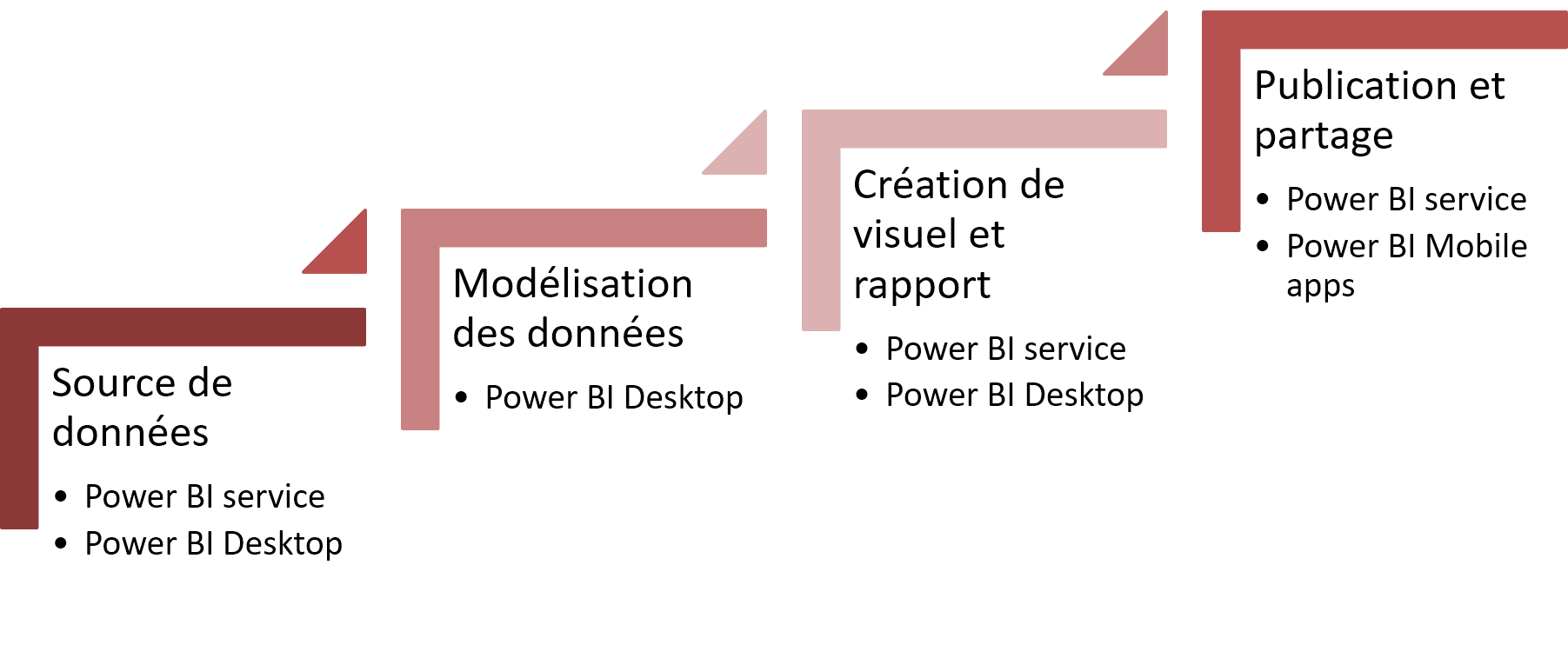
En ce qui suit une comparaison entre les trois outils les plus utilisés pour la création des tableaux de bord :

|  |  |  |  |
| --- | --- | --- | --- |
| **Fonctionnalités** |  |  |  |
| capacités de visualisation | plate-forme facile à utiliser | capacités graphiques et visualisation parfaites | outil d'analyse inclue |
| Fonctions analytiques avancées | Supporte les langages R  et python et leurs visualisations associées | Support complet des langages R et Python et des outils natifs pour le clustering | Pas de support de Python ou R  Le clustering Peut être accessible via des API |
| Cloud | Outil développé pour le cloud disponible sur AZUR | Peut être déployé sur des plateformes tierces comme AWS et AZUR | Entièrement managée en Saas |
| Prix | 9.99$ par utilisateur et par mois | Entre 35$ et 70$ par utilisateur par mois | 20$ par utilisateur par mois |
| Information supplémentaires | Limite à 10 Go de stockage dans le cloud | Limite à 100 Go de stockage dans le cloud | Limite à 500 Go de stockage dans le cloud |

* 1. **Power-BI**

Microsoft définit Power BI comme une suite d'outils d'analyse marketing. C’est une solution d’analyse de données permettant de visualisations de données personnalisées et interactives avec une interface suffisamment simple pour que les utilisateurs finaux créent leurs propres rapports et tableaux de bord.

Habituellement le flux de travail dans Power BI commence par la connexion à une source de données et la création d’un rapport avec Power BI Desktop. Ce rapport est ensuite publié sur le service en ligne Power BI, pour le partager et permettre aux utilisateurs de le consulter depuis le service en ligne ou sur leurs appareils mobiles.



Flux de travail dans Power BI

Il existe plusieurs solutions logicielles similaires au Power-BI tel que QlikView, Tableau et Spago-BI. Pour ce projet nous utiliseront Power-BI pour les raisons précitées :

* Les diverses fonctionnalités offertes

Power-BI se connecte pratiquement à n’importe quelle source de données. Il offre de nombreux visuels et model pour présenter les KPI

* Le cout est relativement faible

Microsoft offre son application Power-BI desktop gratuitement, mais l’achat des différentes licences de Power-BI aide à tirer avantage de toutes les fonctionnalités de sécurité, partage, etc. La plus commune des licences est la licence pro vendue à 10$ par utilisateur par mois.

* Performance et Sécurité

Microsoft fournit des fonctionnalités de sécurité complètes pour le stockage de données dans le cloud. Power-BI vient avec une interface de gestion des accès, de la sécurité et de la gouvernance.

**Conclusion**

Chapitre 3 Analyse des besoins

1. **Définition des rôles**

Le tableau suivant présente les membres et leurs rôles :

|  |  |
| --- | --- |
| **Rôle** | **Nom** |
| **« Product Owner »** | **Ahmed Neffati** |
| **«  Scrum Master »** | **Ahmed Neffati** |
| **« Développeur »** | **Zaibi Ahmed Khalil** |

Tableau 3:Répartition des rôles entre les acteurs

1. **Spécification des besoins**

Des besoins fonctionnels et non fonctionnels ont été fixés suite aux feedbacks des étudiants qui ont utilisé la plate-forme de E-Learning Bridge affiliée à Beecoders.

Le besoin exposé par nos utilisateurs est exprimé à l’aide d’un diagramme de cas d’utilisation et une description textuelle.

**2.1 Besoins fonctionnels :**

Cette partie décrit les besoins fonctionnels de notre projet :

* Consulter les indicateurs liés aux types de contrats, exemple :
* Consulter le nombre de contrats CDI
* Consulter le nombre de contrats CDD
* Consulter le nombre de contrats CIVP
* Consulter les indicateurs liés aux Rémunérations proposées, exemple :
* Consulter la Rémunération proposée maximale
* Consulter la Rémunération proposée minimale
* Consulter les indicateurs liés aux Niveaux d’étude demandés, exemple :
* Consulter le nombre des offres demandant Bac+5
* Consulter le nombre des offres demandant Bac+3
* Consulter le nombre des offres demandant une Certification
* Consulter les indicateurs liés aux Expériences demandées, exemple :
* Consulter le nombre des offres demandant 0 à 2 ans d’expérience
* Consulter le nombre des offres demandant 2 à 4 ans d’expérience
* Consulter le nombre des offres demandant 8 à 10 ans d’expérience
* Consulter la localisation des offres, exemple :
* Consulter le nombre des offres localisé dans le Grand Tunis
* Consulter le nombre des offres localisé dans à Sousse
* Filtrer les données :
* Filtrer les données par catégorie
  1. Besoins non fonctionnels :

Cette partie décrit les besoins non fonctionnels de notre projet :

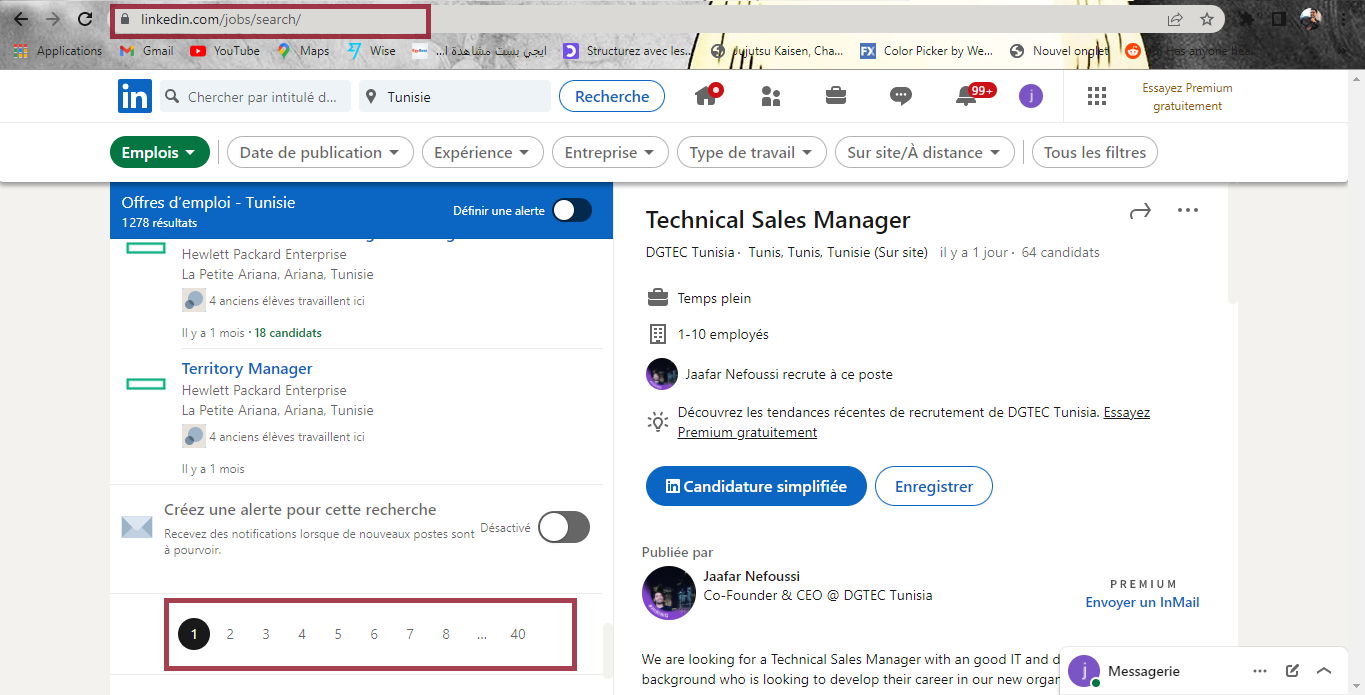
* Ergonomie du tableau de bord :
* Le choix de visualisation qui représente les indicateurs
* Choisir des couleurs qui respectent la charte de Beecoders
* Conception :
* Respecter les normes de modélisation « UML »

1. **Tableau du Backlog**

**3.1 Definition**

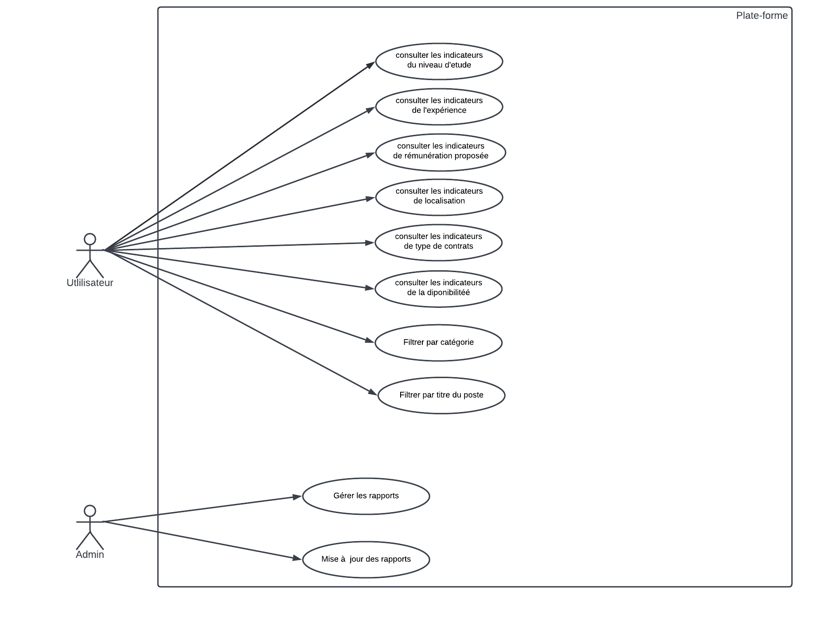
Pour faire simple, l’on peut définir le **backlog scrum** comme étant un **outil**qui sert à recueillir tous les **besoins du client**, dont l’agence spécialisée doit se servir pour travailler sur le projet. De ce fait, le **backlog produit** contient une liste de fonctionnalités utiles à la conception d’un produit.

* 1. **Backlog du produit**



|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Histoire Utilisateur (User Story)** | **Priorité** | **Risque** |
| **1** | **en tant qu'utilisateur, je veux consulter le niveau d'étude demandé par nombre d'offre.** |  |  |
| **2** | **en tant qu'utilisateur, je veux consulter l'expérience demandée par nombre d’offres.** |  |  |
| **3** | **en tant qu'utilisateur, je veux consulter les types de contrats demandés par nombre d’offres.** |  |  |
| **4** | **en tant qu'utilisateur, je veux consulter le nombre d’offres demandées par région.** |  |  |
| **5** | **en tant qu'utilisateur, je veux consulter la rémunération proposée maximale** |  |  |
| **6** | **en tant qu'utilisateur, je veux filtrer le niveau d’étude demandé par nombre d’offres par catégorie.** |  |  |
| **7** | **en tant qu'utilisateur, je veux filtrer l’expérience demandée par nombre d’offres par catégorie.** |  |  |
| **8** | **en tant qu'utilisateur, je veux filtrer le type de contrat demandé par nombre d’offres par catégorie.** |  |  |
| **9** | **en tant qu'utilisateur, je veux filtrer la rémunération maximale proposée par catégorie.** |  |  |
| **10** | **en tant qu'utilisateur, je veux filtrer le nombre d’offres par région par catégorie.** |  |  |

1. **Diagramme de cas d’utilisation global**

****

**Conclusion**

1. **Sprint (0)**

Le but du sprint (0) est la collecte de données, tel que présenté dans le chapitre 2 la méthode choisi pour atteindre ce but est le scrape des sites cibles (les sites)

* 1. **Préparation de l’environnement** 
     1. **Installation de la librairie Selenium**

Afin de préparer convenablement notre environnement de travail, on débute par l’installation de la librairie Selenium qui se fait en deux étapes :

* Utilisation de cette commande pour exécuter l’installation :



* Par suite ,il est indispensable d’installer le WebDriver qui se fait à travers le lien ci-dessous :

<https://chromedriver.chromium.org/downloads>

Le WebDriver est un framework Web qui permet d'exécuter des tests multi-navigateurs. Selenium WebDriver habilite de choisir un langage de programmation pour créer des scripts de test. Après avoir chargé le web driver on le met dans une variable :

browser=webdriver.Chrome("C:\selinium\chromedriver.exe")

* + 1. **Importation des librairies nécessaires**

1. **Keejob**

Pourscraper les offres à partir du site Keejob.tn on procède à l’import des librairies ci-dessous :

from urllib.parse import unquote

from selenium import webdriver

from selenium.webdriver.common import keys

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

import time

import csv

1. **Jobi**

Pourscraper les offres à partir du site Keejob.tn on procède à l’import des librairies ci-dessous :

from urllib.parse import unquote

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

import time

import csv

1. **LinkedIn**

Pourscraper les offres à partir du site Keejob.tn on procède à l’import des librairies ci-dessous :

from urllib.parse import unquote

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

import time

import csv

* 1. **Accès aux sites**

1. **Keejob**

Afin d’acquérir la capacité d’accéder au site Keejob.tn on utilise le WebDriver pour ouvrir le navigateur et procéder à la navigation

browser.get("https://www.keejob.com/offres-emploi")

1. **Jobi**

Afin d’acquérir la capacité d’accéder au site Jobi.tn on utilise le WebDriver pour ouvrir le navigateur et procéder à la navigation

browser.get("https://www.jobi.tn/#!/?q=page=")

1. **LinkedIn**

Afin d’acquérir la capacité d’accéder à LinkedIn on utilise le WebDriver pour ouvrir le navigateur et procéder à la navigation de plus l’authentification automatique

browser.get("https://www.linkedin.com/")

username=browser.find\_element\_by\_id("session\_key")

username.send\_keys("zaibi-ahmed@hotmail.fr")

password=browser.find\_element\_by\_id("session\_password")

password.send\_keys("\*\*\*\*\*\*\*\*\*\*\*\*")

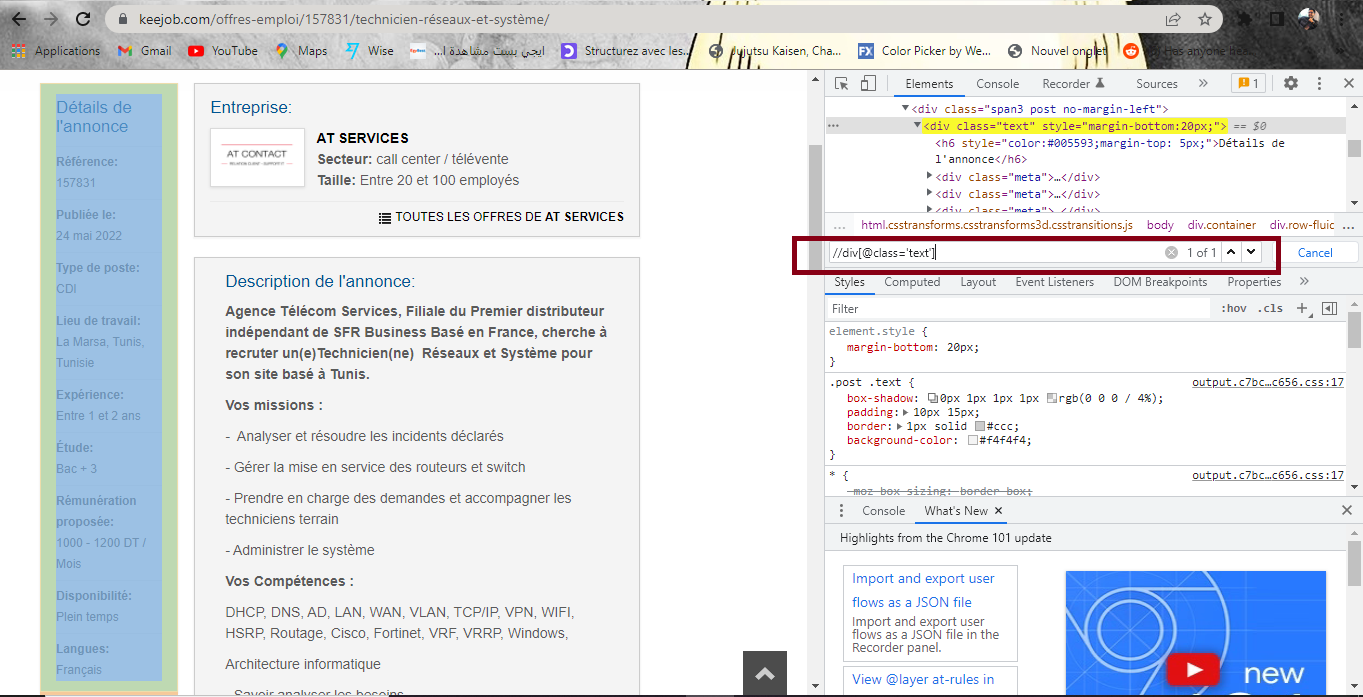
login\_button=browser.find\_element\_by\_class\_name("sign-in-form\_\_submit-button")

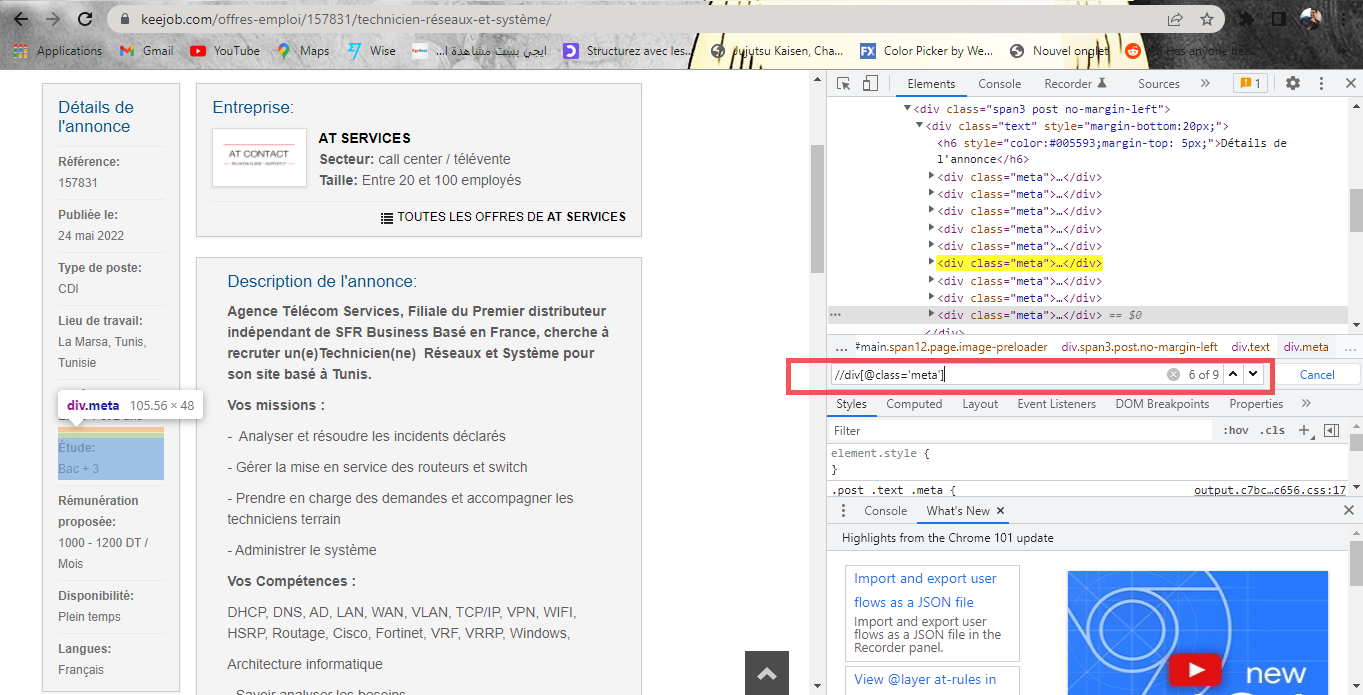
login\_button.click()

* 1. **Localisation des informations**

Afin de localiser les attributs des informations qu’on veut extraire il faut analyser la structure des pages à scraper dans chaque site, pour cela on navigue sur la page cible, on clique sur le bouton droit et on choisit le menu « inspecter élément ».

1. **Keejob**

En inspectant la page qui contient les informations qu’on veut extraire du site Keejob.tn on remarque qu’il y a un élément parent commun donc on peut créer une liste pour les extraire ensemble avec le même nom de classe qui est ‘text’ et le nom de chaque enfant de cet élément qui ont le nom de classe ‘meta’.

****

L’exécution se fait à travers le code suivant :

elements=browser.find\_elements(By.XPATH,"//div[@class='text']")

elements=elements[0].find\_elements\_by\_class\_name('meta')

Puisqu’on a utilisé un dictionnaire on ajoute ces éléments au dictionnaire avec le code suivant :

for j in elements:

            try:

                key=j.find\_element\_by\_tag\_name('b').text[:-1]

                if(not key in keys):

                    keys.append(key)

                    print(keys)

                value=j.text.split('\n')[1].replace(',',';')

                dict[key]=value

            except:

                pass

Pour les éléments qui n’ont pas un élément parent, on les extrait individuellement dans le dictionnaire avec le code suivant :

dict={}

dict['title']=browser.find\_elements(By.XPATH,'//h2[@class="job-title"]')[0].text

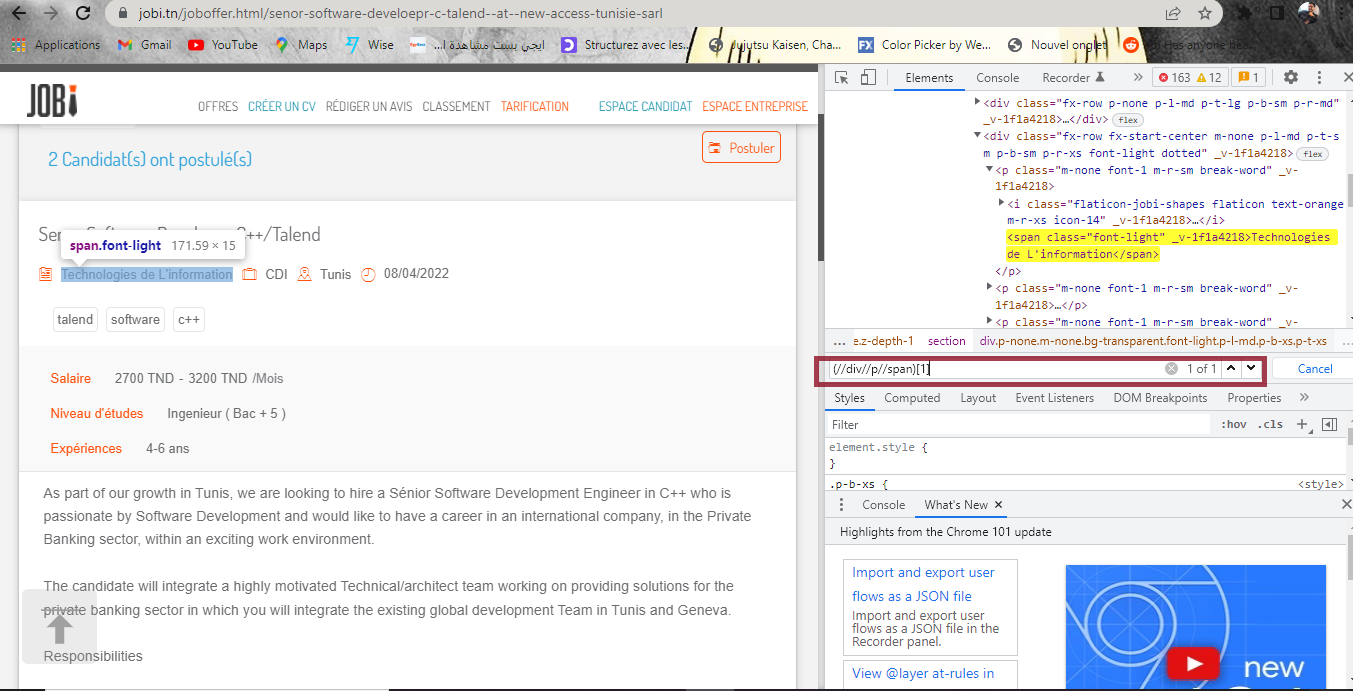
dict['company']=browser.find\_elements(By.XPATH,"//div[@class='span9 content']//b//a")[0].text

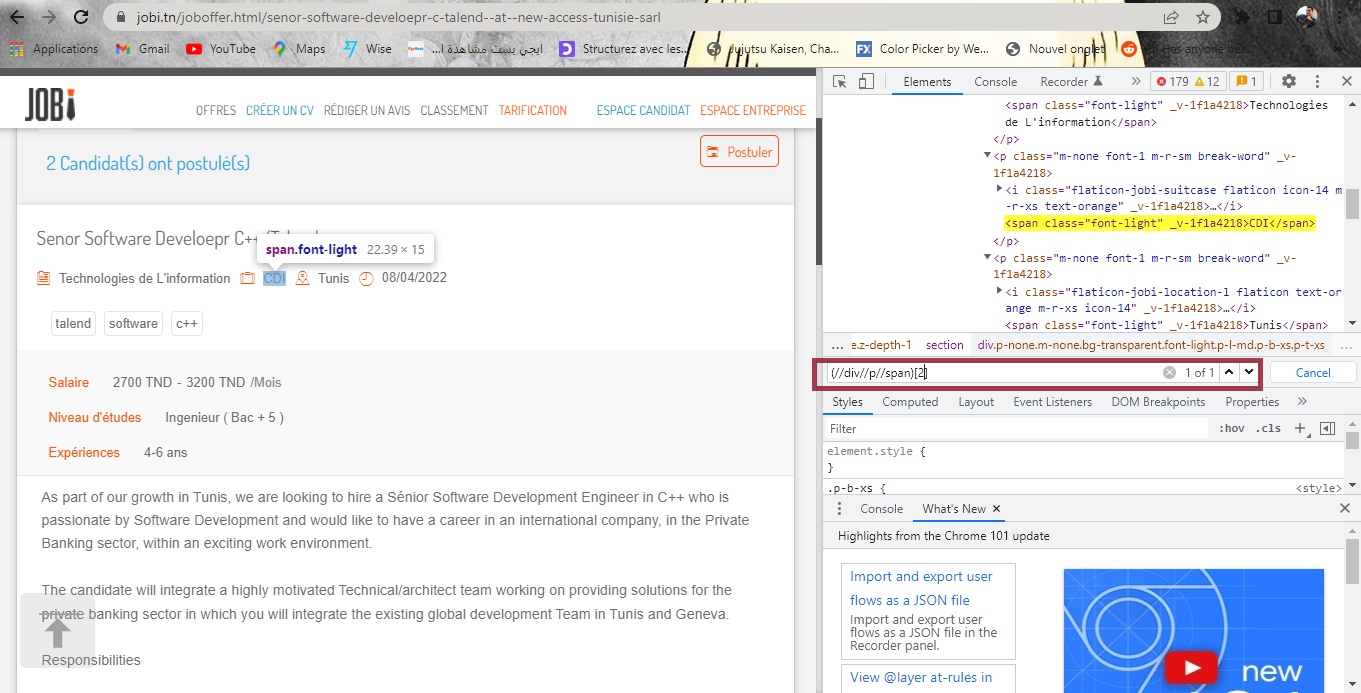
dict['categorie']=browser.find\_elements(By.XPATH,"//div[@class='span9 content']")[0].text.split('\n')[1][9:]

dict['lien']=unquote(jobs[i])

1. **Jobi**

En inspectant la page qui contient les informations qu’on veut extraire du site Jobi.tn on remarque que tous les éléments ont des XPATH différents, donc on va extraire chaque élément tout seul. Comme l’indique les deux figures ci-dessous, une différence émerge entre les XPATH de la catégorie et le type de contrat de cette offre.

****

****

On doit en conséquent identifier l’XPATH de chaque élément individuellement et le mettre dans un dictionnaire avec le code suivant :

            try :

                dict['title']=browser.find\_elements(By.XPATH,'//div//h4')[0].text

            except:

                pass

            try :

                dict['lien']=unquote(jobs[i])

            except:

                pass

            try :

                dict['company']=browser.find\_elements(By.XPATH,"//div//a//span[@class='font-light font-1-5 text-white font-uppercase']")[0].text

            except Exception as err:

                print(err)

            try :

                dict['categorie']=browser.find\_elements(By.XPATH,"(//div//p //span)[1]")[0].text

            except Exception as err:

                print(err)

            try :

                dict['Type de poste']=browser.find\_elements(By.XPATH,"(//div//p//span)[2]")[0].text

            except:

                pass

            try :

                dict['Lieu de travail']=browser.find\_elements(By.XPATH,"//div[@class='col m11']")[0].text

            except:

                pass

            try :

                dict['Expérience']=browser.find\_elements(By.XPATH,"(//div//span[@class='col m8'])[2]")[0].text

            except:

                pass

            try :

                dict['Étude']=browser.find\_elements(By.XPATH,"(//div//span[@class='col m8'])[1]")[0].text

            except:

                pass

            try :

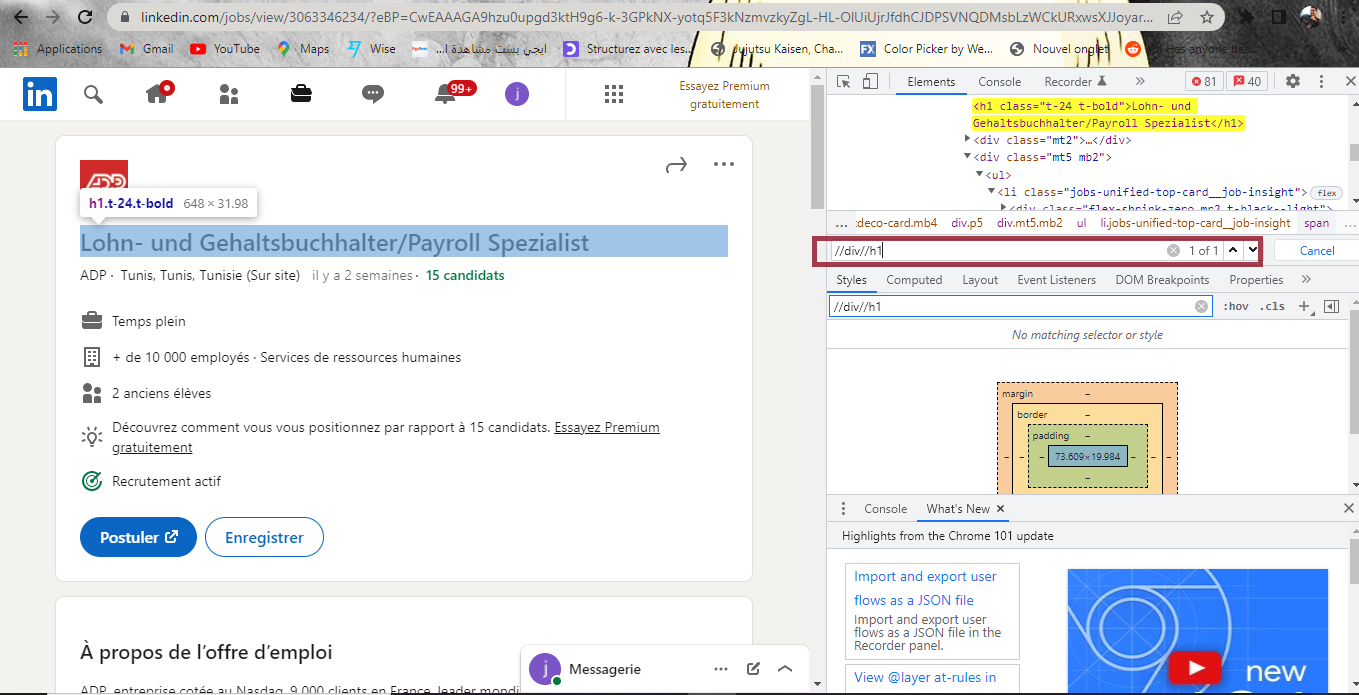
                dict['Rémunération proposée']=browser.find\_elements(By.XPATH,"(//div//span[@class='col m8 fx-row'])")[0].text.replace('\n','')

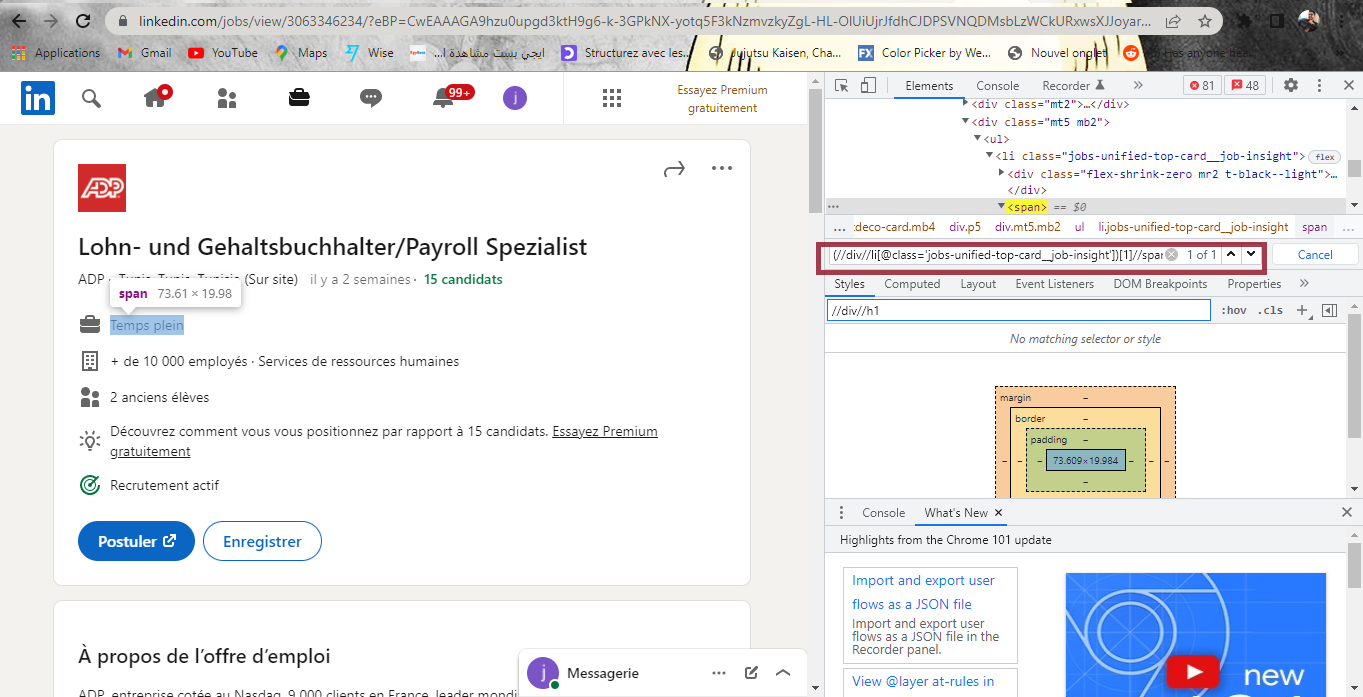
            except:

                pass

1. **LinkedIn**

En inspectant la page qui contient les informations qu’on veut extraire du site LinkedIn.com on remarque que tous les éléments ont des xpath différents, donc on va extraire chaque élément tout seul. Comme l’indique les deux figures ci-dessous, une différence émerge entre les XPATH du titre de poste et la disponibilité de cette offre.



****

On doit en conséquent identifier l’XPATH de chaque élément individuellement et le mettre dans un dictionnaire avec le code suivant :

            dict={}

            try :

                dict['title']=browser.find\_elements(By.XPATH,'//div//h1')[0].text

            except Exception as err:

                print(err)

                print('title')

            try :

                dict['lien']=unquote(jobs[i])

            except Exception as err:

                print(err)

                print('Lien')

            try :

                dict['company']=browser.find\_elements(By.XPATH,'//div//span//span[@class="jobs-unified-top-card\_\_company-name"]')[0].text.replace(',','.')

            except Exception as err:

                print(err)

                print('company')

            try :

                dict['disponibilité']=browser.find\_elements(By.XPATH,"(//div//li[@class='jobs-unified-top-card\_\_job-insight'])[1]//span")[0].text.split(".")[0]

            except Exception as err:

                print(err)

                print('disponibilité')

            try :

                dict['Lieu de travail']=browser.find\_elements(By.XPATH,"(//div//span//span[@class='jobs-unified-top-card\_\_bullet'])[1]")[0].text.replace(',','.')

            except Exception as err:

                print(err)

                print('Lieu')

            try :

                dict['Rémunération proposée']=browser.find\_elements(By.XPATH,"//div[@id='SALARY']//div//p")[0].text.replace(',','.')

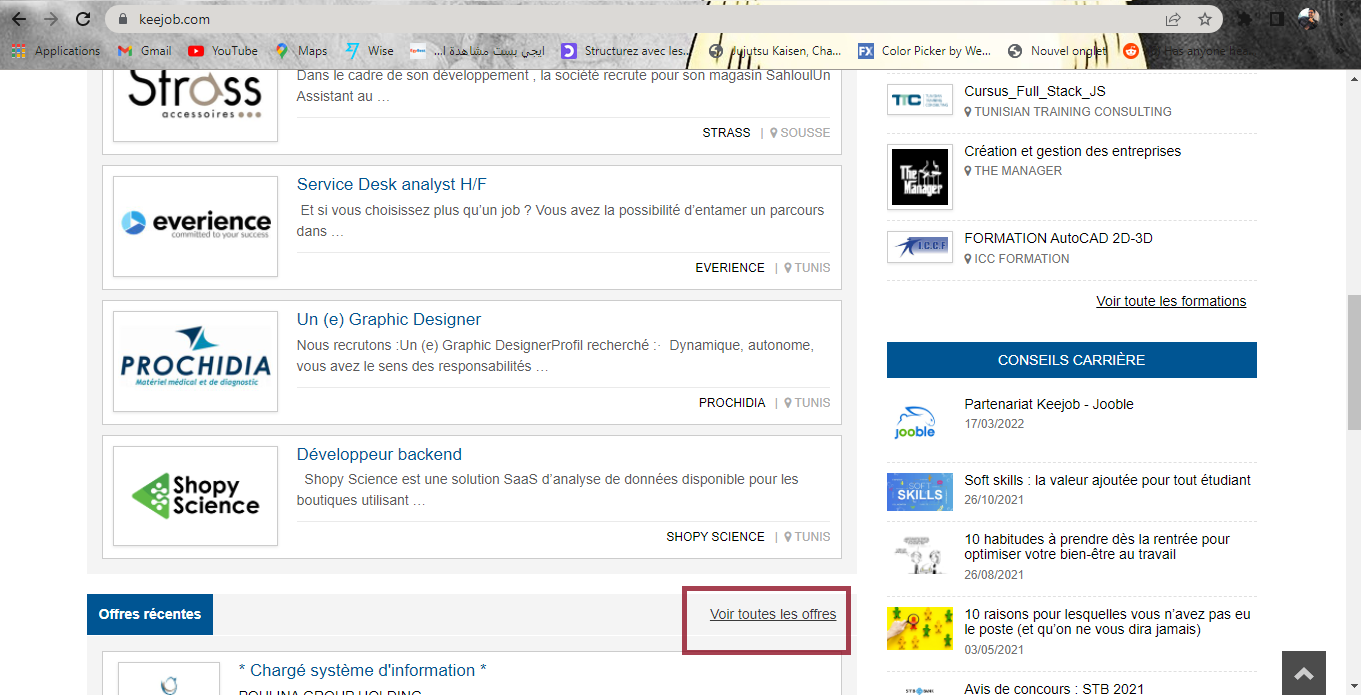
            except Exception as err:

                print(err)

                print('Rémunération')

* 1. **Parcourir chaque offre dans chaque page**

1. **Keejob**

Pour parcourir toutes les offres dans toutes les pages il faut comprendre la structure du site. Apres avoir navigué sur keejob.com, on clique sur « Voir toutes les offres ».

****

Par conséquent, on se trouve sur la première page contenant les offres dont on doit connaitre le lien pour pouvoir créer le scripte de navigation.

Puisque le nombre de pages ne peut pas excéder cent on a écrit le code suivant pour parcourir toutes les pages :

for page in range (1,100):

    browser.get("https://www.keejob.com/offres-emploi/?page="+str(page))

Par suite, on crée une liste de tous les éléments contenants le nom des offres dans la page à partir de son XPATH et on extrait leurs href dans le but de les mettre dans une autre liste qui détaille les liens de chaque offre à accéder pour extraire les informations. Ceci est exécuté à travers ce code :

jobs1=browser.find\_elements(By.XPATH,'//div//h6//a')

    jobs=[]

    for job in jobs1:

        jobs.append(job.get\_attribute('href'))

    print(len(jobs))

    for i in range(len(jobs)):

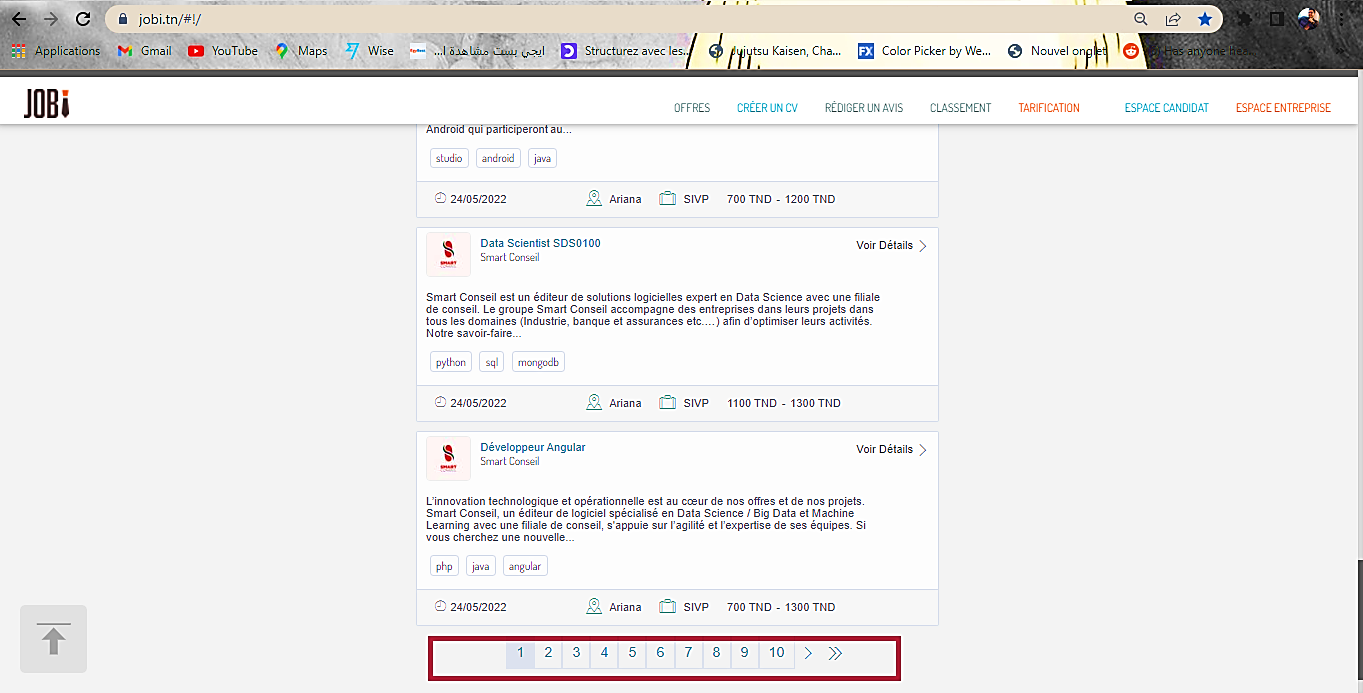
        print(i+1)

        browser.get(jobs[i])

1. **Jobi**

Pour parcourir toutes les offres dans toutes les pages il faut comprendre la structure du site.

Après avoir navigué sur Jobi.tn on se trouve automatiquement sur la première page contenant les offres dont on doit connaitre le lien pour pouvoir créer le scripte de navigation.

****

Puisque le nombre de pages ne peut pas excéder quarante on a écrit le code suivant pour parcourir toutes les pages :

for page in range (1,40):

        browser.get("https://www.jobi.tn/#!/?q=page="+str(page)+"&sortBy=updated\_at&sortOrder=desc")

Par suite, on crée une liste de tous les éléments contenants ‘Voir Detail’ des offres dans la page à partir de son XPATH et on extrait leurs href dans le but de les mettre dans une autre liste qui détaille les liens de chaque offre à accéder pour extraire les informations. Ceci est exécuté à travers ce code :

 jobs1=browser.find\_elements(By.XPATH,'//div[@class="font-1-2"]//a')

        jobs=[]

        time.sleep(5)

        for job in jobs1:

            jobs.append(job.get\_attribute('href'))

        print(len(jobs))

        for i in range(len(jobs)):

            print(i+1)

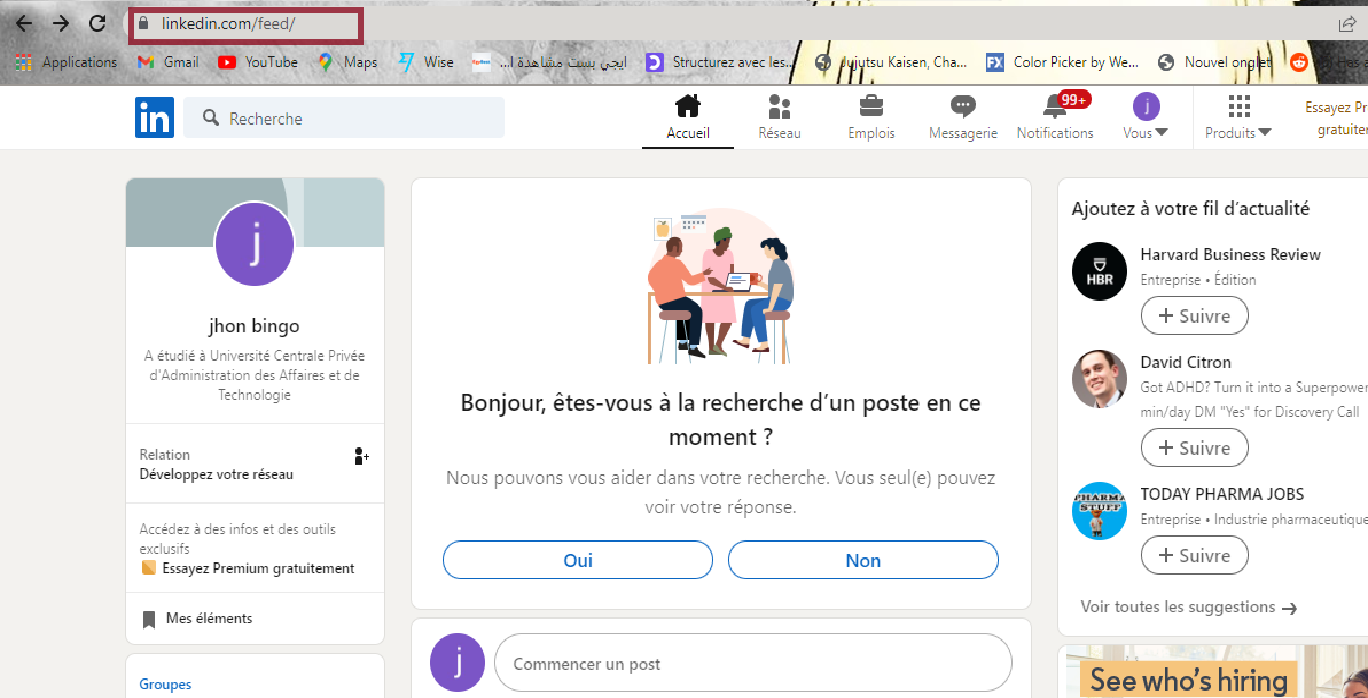
            browser.get(jobs[i])

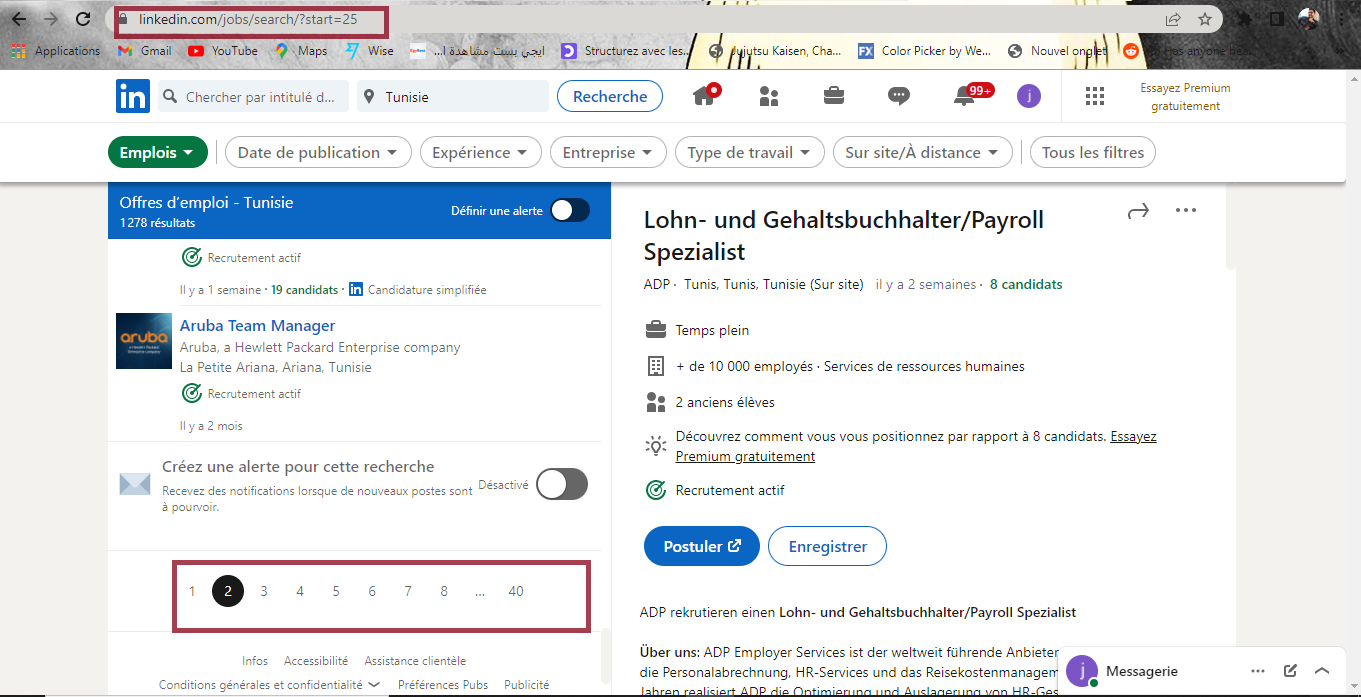
1. **LinkedIn**

Pour parcourir toutes les offres dans toutes les pages il faut comprendre la structure du site.

Après l’authentification, on navigue sur LinkedIn.com on cherche la page contenant les offres dont on doit connaitre le lien pour pouvoir créer le scripte de navigation.

Ceci est observé dans les figures ci-dessous :





On remarque que le lien change dès qu’on change de page, Ainsi on crée un scripte pour agir sur cette différence afin de parcourir les pages d’une façon optimale et Puisque le nombre d’offre comme prévue dans les figure ne peut pas excéder 975 on a écrit le code suivant pour parcourir toutes les pages :

    for page in range (0,975):

        if page==0 :

            browser.get("https://www.linkedin.com/jobs/search/?start="+str(page))

        else:

            browser.get("https://www.linkedin.com/jobs/search/?start="+str(page\*7))

Par suite, on crée une liste de tous les éléments contenants le nom des offres dans la page à partir de son XPATH et on extrait leurs href dans le but de les mettre dans une autre liste qui détaille les liens de chaque offre à accéder pour extraire les informations. Ceci est exécuté à travers ce code :

        jobs1=browser.find\_elements(By.XPATH,"//div//div//a[@class='disabled ember-view job-card-container\_\_link job-card-list\_\_title']")

        jobs=[]

        time.sleep(5)

        for job in jobs1:

            jobs.append(job.get\_attribute('href'))

        print(len(jobs))

        for i in range(len(jobs)):

            print(i+1)

            browser.get(jobs[i])

* 1. **Enregistrer les données scrapées dans un fichier csv**

En premier lieu on a utilisé un dictionnaire quiest une collection dans le but d’associer une clé à une valeur pour manipuler les cases libres dans les éléments ciblé. En outre, on a utilisé le module csv pour insérer le dictionnaire dans le fichier csv.

1. Keejob

Pour créer le fichier Csv pour les données collectées à partir du site Keejob.tn et l'en-tête de la table on utilise le code suivant :

with open('C:\\Users\\TOSHIBA\\Desktop\\pfe\\Keejob.csv','w',encoding="utf-16") as f\_object:

    csv\_columns = ['Publiée le', 'Type de poste','Expérience', 'Étude', 'Rémunération proposée', 'Disponibilité', 'Langues', 'Mobilité']

    writer\_object = csv.DictWriter(f\_object, delimiter=',', fieldnames=csv\_columns)

    writer\_object.writeheader()

**Ensuite, pour enregistrer le dictionnaire dans le fichier csv on utilise le code suivant :**

writer\_object.writerow(dict)

1. Jobi

Pour créer le fichier Csv pour les données collectées à partir du site Jobi.tn et l'en-tête de la table on utilise le code suivant :

with open('C:\\Users\\TOSHIBA\\Desktop\\pfe\\jobi.csv','w',encoding="utf-16") as f\_object:

    csv\_columns =['title','lien','company','categorie','Type de poste','Lieu de travail','Expérience','Étude','Rémunération proposée']

    writer\_object = csv.DictWriter(f\_object, delimiter=',', fieldnames=csv\_columns)

    writer\_object.writeheader()

**Ensuite, pour enregistrer le dictionnaire dans le fichier csv on utilise le code suivant :**

writer\_object.writerow(dict)

1. LinkedIn

Pour créer le fichier Csv pour les données collectées à partir du site LinkedIn.com et l'en-tête de la table on utilise le code suivant :

with open('C:\\Users\\TOSHIBA\\Desktop\\pfe\\linkedin.csv','w',encoding="utf-16") as f\_object:

    csv\_columns =['title','lien','company','disponibilité','Lieu de travail','Rémunération proposée']

    writer\_object = csv.DictWriter(f\_object, delimiter=',', fieldnames=csv\_columns)

    writer\_object.writeheader()

**Ensuite, pour enregistrer le dictionnaire dans le fichier csv on utilise le code suivant :**

writer\_object.writerow(dict)