



**HONORIS UNITED UNIVERSITIES**

**2022 / 2023**

**SPECIALITY : Data Science**

**End of the year project report**

**Realized by :**

**Eya Belguith  
Jihen Souissi  
Mohamed Dhibi  
Ahmed Toukebri  
Mohamed Magri  
Aya Chlibi**

**Supervised by:**

**M. Abderrahmen Benarous   Professor at Esprit  
Ms. Ghada Setti**

# Table of Contents

1. Abstract.....	1
2. Introduction.....	2
3. Methodology.....	3
4. Business understanding.....	6
5. Problematic.....	8
6. Fonctional requirements.....	10
7. Non fonctional requirements.....	12
8. SMART.....	13
9. Key performance indicator.....	14
10. Data collection.....	15
11. Data storage.....	17
12. Data understanding.....	18
13. Data preparation.....	20
14. Data Modeling & Evaluation.....	22
15. Deployment.....	28
16. Conclusion and perspectives.....	32
17. Refrences.....	33

# List of figures

- figure 1 . IBM master plan in data science based projects.....5
- figure 2 . Esprit Connect website .....8
- figure 3 . our career center platform JOBIT logo .....9
- figure 4 . BigQuery google warehouse .....17
- figure 5 . figure 5 .Screenshot of Welcome page.....28
- figure 6 . Screenshot of JOBIT Home page.....28
- figure 7 . Screen shot of "About JOBIT" page.....28
- figure 8 .Screenshot of Sign In Page.....29
- figure 9 .Screenshot of "Complaints received by the PE" Page.....29
- figure 10 .Screenshot of Uploading offers Page.....29
- figure 11 .Screenshot of the dashboard Page.....30
- figure 12 .Screenshot of the dashboard Page.....30
- figure 13 .Screenshot of the Sign Up Page for students.....31
- figure 14 .Screenshot of "filling the infos" Page.....31
- figure 15 .Screenshot of the Complaints Page.....31

# List of Acronyms

- EP employability pole
- IBM International Business Machines
- NLP Natural language processing
- BI Business intelligence
- KPI Key Performance Indicator
- AI artificial intelligence
- ELT Extract Load Transform
- GUI Graphical user interface
- SMART specific measurable achievable relevant timely
- OCR Optical Character Recognition

# **1. ABSTRACT**

In a fast-paced world, traditional ways for both job and candidate searching which take up too much time and effort, are no longer relevant, thus an AI-generated career center platform seemed to be an ideal solution. hence, we went ahead and did a fair amount of research to get hold more of the difficulties faced by the employability pole of the Esprit Group whilst dealing with a mass amount of received emails containing job and internship offers in many forms (images, Word, PDFs..) then we proceeded to collect and prepare the needed data for the main purpose of training a model that will not only simplify the EP's tasks but also expedite his assignments. consequently, we gained a deeper knowledge regarding Natural Language Processing, NoSQL databases, ELT, and data warehousing. The corollary is that This project turned out to be very interesting and enriching for our professional experience.

## **2. INTRODUCTION**

In the last few years, there was a noticeable expansion in the Tunisian Job labour, increasing the number of job seekers leading graduates often to struggle finding suitable employment opportunities due to the high competitivety of the market . As a matter of fact, According to the International Labour Organization "the unemployment rate in Tunisia was 16.2% in 2020, which was aggravated by the COVID-19 pandemic". In response to this challenge, the Esprit Group, a leading educational institution in Tunisia, recognized the importance of supporting the employability of its graduates and assigned us, a group of computer science engeneering students specializing in data science, the developement of a career center platform, utilizing artificial intelligence technologies in order to match students with suitable job openings and help improve their employability.

This technical report provides a follows-up to the produced model we came up with , for other institutions seeking to improve the employability of their graduates and demonstrates the potential impact of technology in addressing complex societal challenges.

### **3. Methodology**

**IBM Master Plan** is a comprehensive system for managing complex technological projects (design, development, testing, and deployment) that was created in the 1960s by IBM to ensure that projects were completed on time, within budget, and to the required quality standards.

The IBM Master Plan is divided into four main phases: definition, planning, execution, and delivery.

- definition phase : identifying the project's requirements and objectives, as well as defining the scope of the project.
- planning phase : developing a detailed plan for the project ; timelines, budgets, and resource requirements.
- execution phase : the development of the project ; design, coding, testing, and integration.
- delivery phase : the final delivery of the project to the customer or end-user.

The latter was a significant success helping IBM manage its projects more effectively. The system was eventually adopted by other companies and became a widely recognized framework for managing complex projects.

The same as any other IT related projects, The IBM Master Plan Methodology for Data Science based Projects is a structured approach to successfully help their execution.

This methodology's key steps in a Data science context:

- Defining the Business Problem: identify the business problem that will be addressed by the project ; understanding the business objectives, identifying the key concerned people, and defining the scope of the project.
- Developing a Project Plan: one that outlines the tasks, timelines, and resources required to complete the project. it should also include a risk management plan and a communication plan.
- Collecting and Preparing Data: This involves identifying relevant data sources, cleaning and transforming the data, and ensuring the quality of the data.
- Analyzing Data: to gain insights and identify patterns ; exploratory data analysis, data visualization, and statistical modeling.
- Developing Models: that can be used to predict future outcomes or classify data into different categories (machine learning algorithms, deep learning models, or other predictive models.)
- Evaluating Models: to evaluate the performance of the models and determine which models are most effective in addressing the business problem.
- Deploying Solution: and integrating it into the business processes. (developing a user interface, training employees on how to use the solution, and monitoring the solution's performance..)

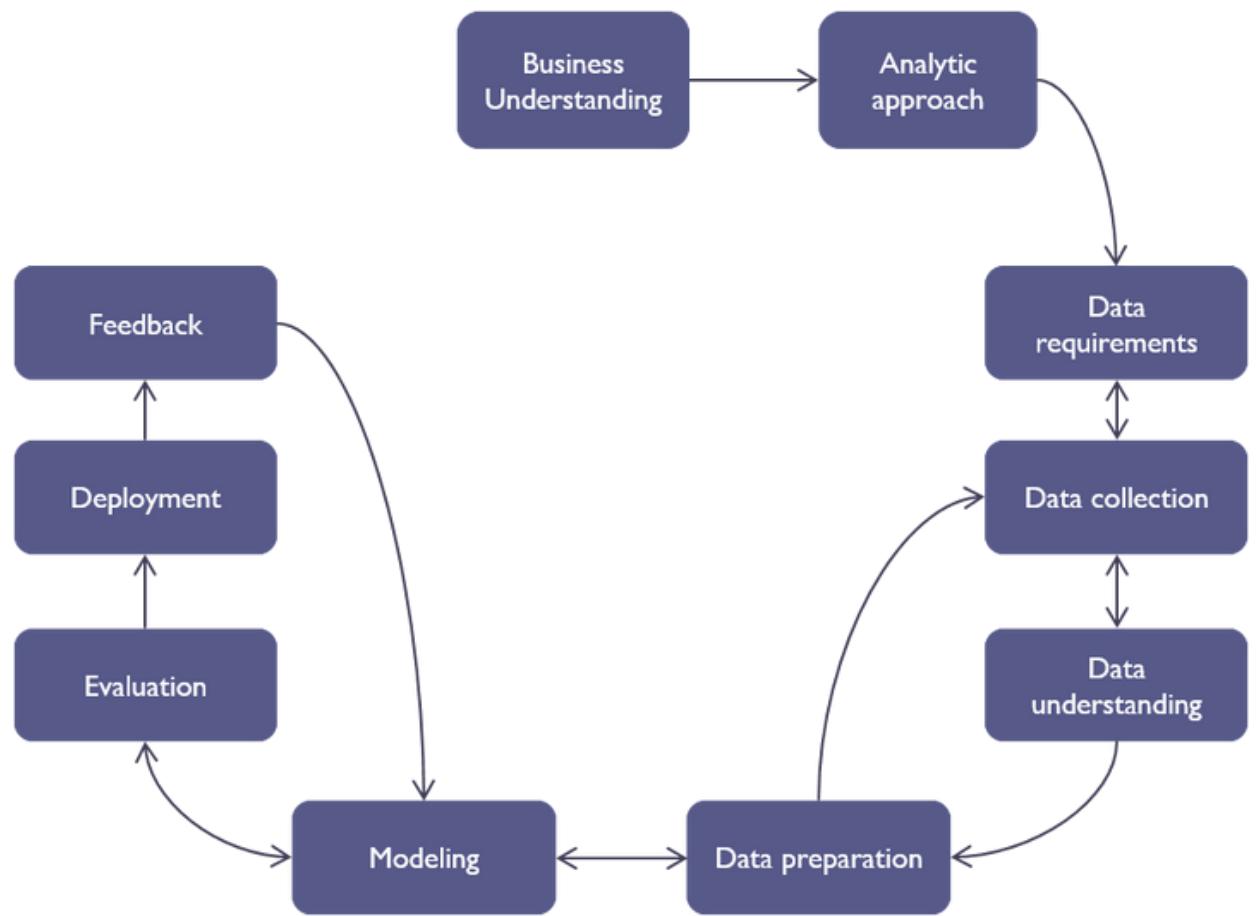


figure 1 . IBM master plan in data science based projects

## **4. Business understanding**

Employability as a whole is a set of achievements, skills, and personal attributes that makes graduates more likely to gain job offerings and attain the desired success in chosen occupations, which benefits them, the workforce, the community, and eventually the economy.

In a more demanding employment market, the need for mentorship and guidance for students and alumni to choose their careers wisely is crucial. that's when career centers are proven to be of use since they're specially designed to :

- Help students sharpen their job search skills
- Support students in defining and reaching career goals
- Connect students with suitable careers
- Offer employer connections and networking opportunities

Principally in Esprit college, we have Esprit Group's Career Center that offers Technical and non-technical workshops, Hackathons, Business visits.. and Esprit Connect (the platform) for the main purpose of diving deeper into its usefulness, we created a survey to gather information from Esprit students, in which we asked first about their grade and specialty, where mainly 4th grader TIC students were Respondents. then whether they used the platform esprit connect, discovering that 40% didn't and that the 60% who did, 0% were totally satisfied and 64.7% were unsatisfied. continuing with the platform users, we asked whether they were able to find jobs and 30% answered yes. finally, by asking about the problems they encountered using the platform, some answers we received were "it's more tic oriented", "it's chaotic" "wasting time just searching for my specialty" and "it lacks internships, as a 3rd-year student I'm not yet interested in job offers"

# 5. PROBLEMATIC

The (EP) employability pole of the Esprit group, receives way too many job and internship offers in various formats (image, Word, PDF, e-mail, ...) which led to the difficulty of manually managing this enormous amount of data since they encounter the Lack of data analytics capabilities; meaning that they have no visibility on the skills required by the job market, the geographical distribution of the offers, thus, eventually resorting to broadcasting everything which leads to Ineffectiveness of properly matching candidates with offers.

## 5.1. Study Of The Existing

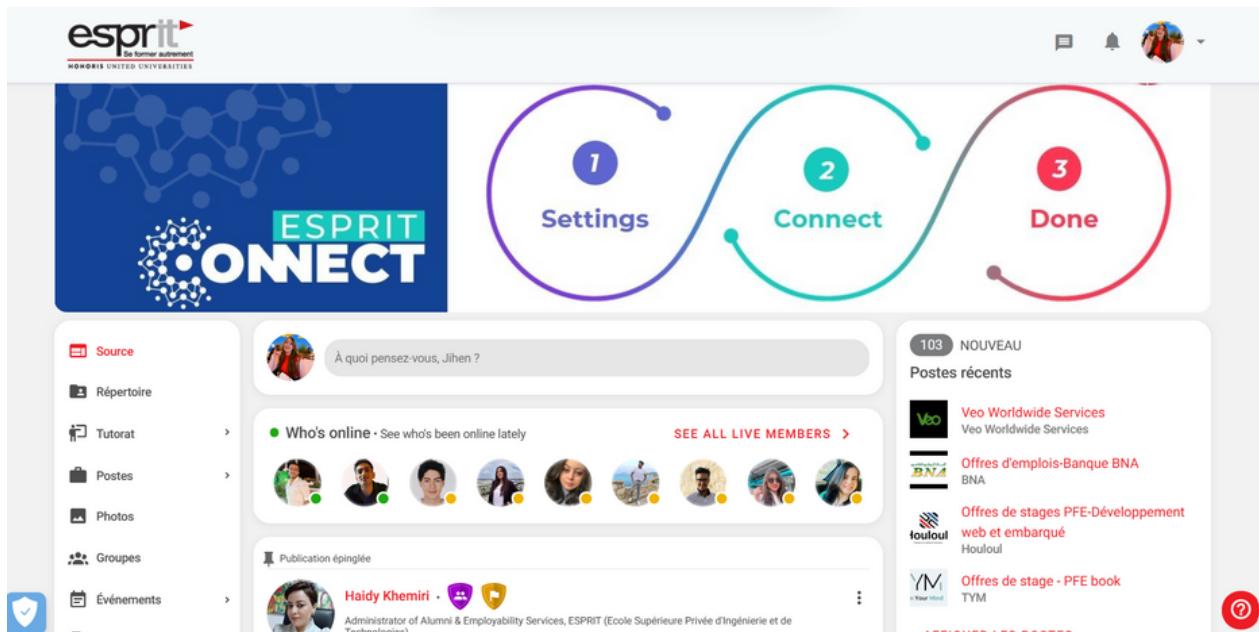


figure 2 . Esprit Connect website

## **5.2. Solution**



figure 3 . our career center platform JOBIT logo

In order to overcome the limitations of the existing Website, we are going to create a dynamic web application entitled “JobIt”, that guarantee the best matching between th students and the offers

we will provide an interface (dashboard) for the PE having analytical studies about the offers containing :

- Number of offers per period ,the company with highest number of offers
- Geographical distribution (in Tunisia and in the world)...
- Technologies/trade/fields

# **6. FUNCTIONAL REQUIREMENT**

## **6.1. BUSINESS OBJECTIVES**

We came up with two business objectives that will be the base of our work:

- Reducing the amount of time required to find suitable job opportunities/profiles, done by both profiling. (A solution that can efficiently and effectively manage the large volume of job and internship offers received in different formats.) and Matching and recommendation candidates (A system that can match candidates with suitable job offers, with a certain percentage based on their skills and qualifications.)
- Giving the EP more visibility into the job market by data extraction (The ability to extract insights and trends from the data, such as the skills required by the job market and the geographical distribution of the offers.) and Reporting and predicting ( A solution that provides real-time data analytics and reporting capabilities based on KPI's :
  - Number of offers per period (month, quarter, year)
  - number of offers by Technologies, trades, or fields
  - the geographical distribution of the job offers nationally and internationally )

# **6. FUNCTIONAL REQUIREMENT**

## **6.2. DATA SCIENCE OBJECTIVES**

we will be focusing on both :

- Descriptive analytics ;

-Web scraping to collect job offers and esprit students from Linkedin

-NLP to extarct and summarize informations from PDFs and images taken from EP emails

-clustering can be used to categorize and describe the data

-scatter plots can be used to understand the geographical distribution of the job offers

- Predictive analytics;

-regression to predict the number of offers to obtain popular fields in different area

-optimization algorithms to find the best candidates for a job offer

-NLP to match job offers with students

# **7.NON FUNCTIONAL REQUIREMENTS**

## **7.1. PERFORMANCE**

- Security: Connection establishment requirements.
- Response Time: Processing Time and Initial Loading and Subsequent Loading Times
- compatibility: Defines how the system can co-existe with another system in the same enviroment .
- Availability : Describes how likely the system is accessible for a user at a given point of time
- Reliability : specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions

## **7.2. GUI**

- Usability : How easy is it for a customer to use the system
- GUI ergonomics :Includes the determination of user needs, interfaces design, user support and usability testing.

## **8. SMART**

SMART goals are specific, measurable, achievable, relevant, and time-bound objectives for a project. They define what the project is trying to accomplish and provide a clear target for the project team to work towards.

For our project, we have chosen the following:

- Specific: Develop a career center application with an automatic recommendation system that matches the offers with the suitable candidates, a students profiling system according to their skills and education and a reporting system.
- Measurable: Increase the match rate between the company offers and the students compared to the current system.
- Achievable: Utilize existing data from the existing system, emails and external sources such as LinkedIn
- Relevant: Make the search for offers easier for students and give the companies with the offers a smaller group of students with the right requirements for the proposed jobs.
- Time-bound: Complete the development of the application by the end of May and deploying it in the next year.

## **9.KPI**

A key performance indicator (KPI) is a measurable value that demonstrates how effectively a company is achieving key business objectives.

It's a tool to evaluate their progress and success at reaching targets.

A KPI is a goal that you work towards achieving.

Chosen KPIs:

The KPI's that we are taking are:

-OFFER AND STUDENT RETENTION: it is the percentage of students and job/internship offers that remain active in the matching process over time.

-MATCH ACCURACY: the percentage of correctly matched job/intership offers to students

-TIME TO MATCH :the time it takes to match a student with a job/intership offer.

# **10.DATA COLLECTION**

## **10.1. JOB OFFERS**

we used **imaplib** library to retrive emails received by <pole-employabilite-esprit@esprit.tn> extracting the wanted elements ; subject, from, body, where we noticed that the body is in different formats ; text, images and pdfs

- use of **PyPDF2** library to open the pdf files, create a pdf reader object and get the number of pages in each one
- use of **pdf2image** which is a python module that we used to convert PDF pages into high fidelity images it's a wrapper around the library poppler
- use of **pillow** which is a library containing all the basic image processing functionality->using it for image resizing, rotation and transformation.
- use of **Pytesseract** (a wrapper for Google's Tesseract-OCR Engine) to converte images (collected from emails / converted from PDFs, in order to extracts text from images without a text layer and outputs it into a new searchable text file)
- use of **googletrans** to translate all the texts into english

-using Gensim library for document indexing and similarity retrieval with large corpora and later on removing stop words that we added manually

-use of NLTK(Natural Language Toolkit)a suite of libraries and programs, with word\_tokenize() method it helps splitting texts into tokens.

- we also applied web scrapping on Linkedin for data augmentation of Job offers

## **10.2. STUDENTS**

we applied web scrapping on Linkedin to retrive Esprit students

-use of **Selenium** for enabling rapid, repeatable web-app testing

-use of **Beautiful Soup** a library making it easier to scrape information from LinkedIn ->pulling data out of HTML

# 11. DATA STORGE

we loaded our csv files into google cloud **bigquery** a fully managed enterprise data warehouse that could help us manage and analyze our data with built-in features like machine learning, and business intelligence

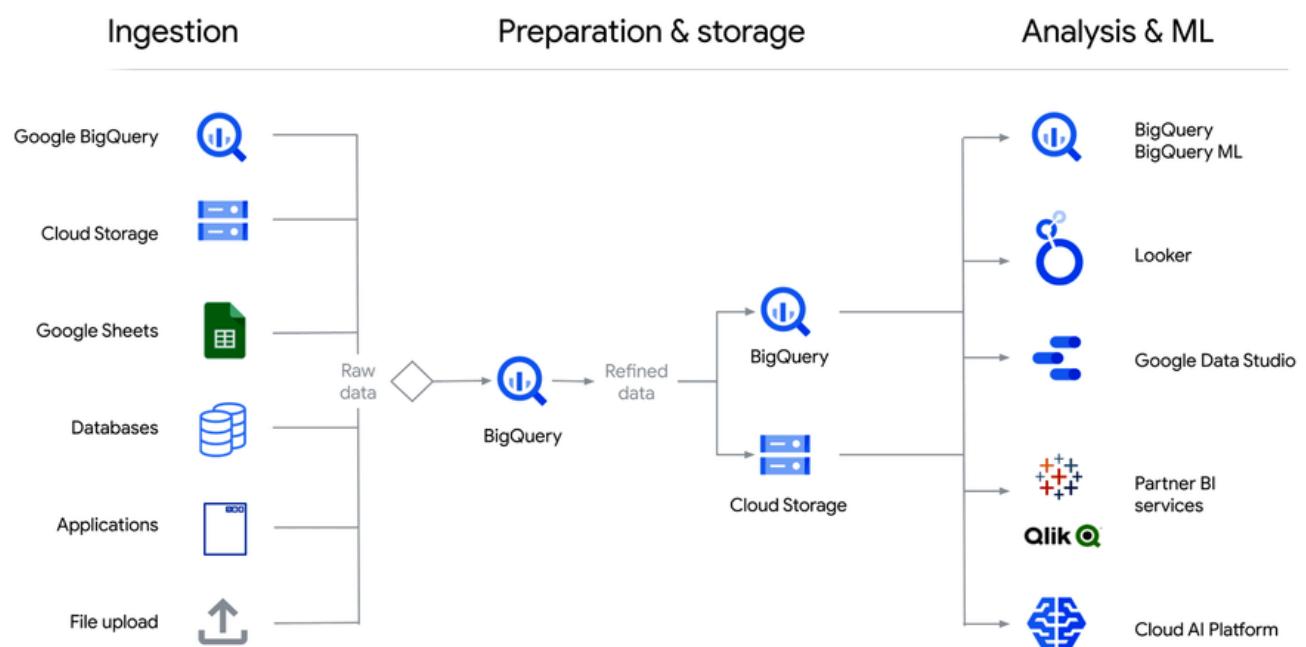


figure 4 . BigQuery google warehouse

# **12.DATA UNDERSTANDING**

## **(data quality )**

Data quality is a measure of the condition of data based on factors such as: accuracy, completeness, consistency, reliability and whether it's up to date.

### **12.1 students**

Since most Esprit students have LinkedIn accounts, data scrapping from there was ideal since each profile has important information. Thus the dataset we realized is accurate, complete (all the public information on one's profile), reliable since LinkedIn is a trusted platform with the audience and tools, and up to date since we specifically searched for current Esprit students.

All the public informations found on their profiles :

- PERSONAL STATEMENT description / headline
- connectionCount / connectionDegree / connectionUrl / mutualConnectionUrl / mutualConnectionText / connectedOn / subscribers
- LINKS linkedinProfile / imgUrl / linkedinSlesNavigatorUrl
- EDUCATIONION school / schoolUrl / schoolDegree / schoolDateRange / school2 / schoolUrl2 / schoolDegree2 / schoolDateRange2
- SKILLS allskills / skill1 / endorsement1,...., skill6 / endorsement6
- CONTACT INFO userId / profileId / website / facebookUrl / email

- IDENTITY SET firstName / LastName / fullName / birthday / location
- EXPERIENCES company / companyUrl / companyWebsite / jobTitle / jobDescription / jobLocation / jobDateRange
- civilityFromDrobContact / websiteFromDropContact / mailFropDropContact / qualificationFromDropContact / phoneNumberFromDropContact

## 12.2 Job Offers

The main source of job and internship offers is the different emails broadcasted by the EP, plus the drive provided for us containing PDFs of last-to-date offers.

And for data augmentation, we used Linkedin scrapping.

Overall, the data mainly contains :

- type : of the offer {job/summer Internship/PFE}
- from: the email sender
- Sujet: the purpose of the mail
- Skills: the required expertise
- link: included links to submit an application in

# **13. DATA PREPARATION**

## **13.1 Profiling dataset for Students**

- Manual and randomized feature selection (based on our knowledge of the problem and the data)
- example : dropped "skill1" and "skill2" which are just an extraction of each list in the column "allskills "
- -> new data: fullname , headline, location, schooldegree, jobtitle, linkedinprofil, email ,all skills
- created new function that detects nun ( since some columns are objects)
- replacing Nan in "email" by "first name" + "last name" @esprit.tn
- droping rows with "fullName" being Nan
- changing objects into str to remove remaining nan
- merging 2 columns
- example : "degree1" and "degree2" into "degree" devised by !

## **13.2 Matching dataset for Students**

- creating a list containing all of the students skills (in uppercase/ removing space / translating in English using GoogleTrans library / removing suffixes ; same skill written differently)

- creating a new list from the previous containing only unique skills ->reduced the size into its 1/4
- defining a new dictionary based on the regrouping of related skills
- creating 60 new binary columns based on the skills created list and each student skills

### **13.3. Dashboard dataset for Offers**

**Provided job offers+EP emails:**

- removing punctuation and such using nlp and stopwords by nltk
- using gensin library to transform full text into words
- using Chatgpt to get personalized dictionary

### **13.4. Matching dataset for Offers**

for the skills ,same process as the students data

### **13.5. Profiling dataset for Offers**

linkedin job offers:

- extracting the skills from the "description"

# **14. DATA Modeling & evaluation**

## **14.1 Matching and recommendation candidates**

### **Binary/Jaccard Similarity**

Calculating the Jaccard similarity between the competences of each offer and the skills of each student. It then stores these match values in a list called "match\_values".

The Jaccard similarity is a measure of similarity between two sets. It is calculated as the size of the intersection of the sets divided by the size of the union of the sets.

The outer for loop iterates over each offer in the "offerprofiling" dataframe. The inner for loop iterates over each student in the "studentprofiling" dataframe. For each offer-student pair, the Jaccard similarity between their competences and skills is calculated and stored in the "jaccard\_similarities" list.

Once all Jaccard similarities have been calculated for an offer, the list of match values for that offer is added to the "match\_values" list.

After all match values have been calculated, the "match\_values" list is converted to a dataframe called "match\_values\_df". Finally, the "offers accepted by binary" column of the "studentprofiling" dataframe is updated with the match values for each student.

The apply method is used to apply a lambda function to each row of the transposed "match\_values\_df" dataframe. The lambda function returns a string that lists the index values of the match values that are greater than 0.7 (i.e., the match values that indicate a strong match). The "|" character is used to separate the index values in the string.

The transpose method is then called on the "match\_values\_df" dataframe to match the shape of the "studentprofiling" dataframe. However, the last line of the code, which simply calls "match\_values\_df.transpose()", seems to be unnecessary and does not do anything.

### **Cosine similarity**

implementing a matching algorithm between job offers and students based on their skills. It creates an empty matrix to store the match values between each offer and each student, and then iterates through each offer and student to calculate their match value using cosine similarity between the skills listed in their respective profiles. The match values are stored in a list and converted to a dataframe. Finally, the code updates the "offers accepted by binary" column of studentprofiling based on the match values. Specifically, it sets this column to a string of the indices of the offers that the student accepted, separated by a pipe character.

- Creating an empty matrix for storing matches The script creates an empty matrix called matches with rows for each offer and columns for each student. This matrix will be filled with the match values computed using cosine similarity.
- Defining a function to pad lists The pad\_lists function is defined to pad two lists with a specified value to make them the same length. This function is used later to ensure that the lists of skills for each student and job offer are the same length when computing cosine similarity.
- Iterating through each offer and student The script then iterates through each row in offerprofiling and studentprofiling using a for loop. For each offer and student pair, the script calculates the cosine similarity between the skills of the offer and the skills of the student using scikit-learn's cosine\_similarity function.
- Storing match values The cosine similarity value for each offer-student pair is stored in a list called cosine\_similarities. This list is then added to another list called match\_values, which stores the cosine similarity values for all offer-student pairs.
- Converting match values to a DataFrame Once all of the cosine similarity values have been computed, the match\_values list is converted to a Pandas DataFrame called match\_values\_df. This DataFrame has rows for each offer and columns for each student, and each cell contains the cosine similarity value between the corresponding offer and student.
-

- Updating studentprofiling DataFrame Finally, the script updates the studentprofiling DataFrame by adding a new column called "offers accepted by Spacy". This column contains a string that lists the offers that each student has a positive match with, separated by a "|" symbol. The values in this column are computed by transposing match\_values\_df, iterating through each row, and concatenating the index of each column where the cosine similarity value is greater than 0.

### **BERT-base-uncased**

\_a deep learning algorithm used for natural language processing tasks like text classification, question answering, and language modeling. Specifically, the algorithm is used to encode the skills of both students and job offers into numerical vectors that can be compared using cosine similarity to find the best matches.

- Loading the Pre-trained BERT Model and Tokenizer: The pre-trained BERT (Bidirectional Encoder Representations from Transformers) model and tokenizer are loaded from the 'bert-base-uncased' model. This is a popular pre-trained model for NLP tasks that was trained on a large corpus of text.
- Defining the Encoding Function: The encode\_text function takes text as input and returns the encoded text using the BERT model. The function first tokenizes the text using the BERT tokenizer, then passes it through the BERT model. The function concatenates the last four hidden layers, averages the token-level embeddings across all tokens, and returns the resulting vector.

- Encoding Students: The allSkills column of the studentprofiling dataframe is concatenated into a single string for each student, and then each of these strings is passed through the encode\_text function to generate a vector representation of each student's skills. These vectors are stored in the student\_vectors list and added as a new column to the studentprofiling dataframe.
- Encoding Offers: Each offer in the offerprofiling dataframe is passed through the encode\_text function to generate a vector representation of each offer's required skills. These vectors are stored in the offer\_vectors dictionary.
- Finding the Best Matching Offer for Each Student: For each student, the cosine similarity is calculated between the student's skill vector and each offer's skill vector. The top three offers with the highest similarity scores are selected as the best matches for that student. The best matches are stored in the matches list and added as a new column to the studentprofiling dataframe.
- Filtering the Best Matches: For each student, the offerprofiling dataframe is filtered to include only the best matching offers. The index of the first matching offer is stored in the m list, which is then added as a new column to the studentprofiling dataframe.

The **tokenizer** used in this code is the BERT tokenizer, which is used to split the text into individual words or subwords and convert them into integer indices that can be used as input to the BERT encoder.

The **vectorizer** used in this code is the CountVectorizer from scikit-learn, which is used to convert the text data into a bag-of-words representation. The CountVectorizer counts the frequency of each word in the text and represents the text as a vector of word counts.

The **cosine similarity** metric is used to measure the similarity between the offer and student vectors. The cosine similarity measures the cosine of the angle between two vectors and ranges from -1 to 1, where 1 indicates perfect similarity and -1 indicates perfect dissimilarity. In this code, the cosine similarity is used to find the best matching offer for each student by comparing the student vector to each offer vector.

# 15. Deployment

## 15.1 Home Page

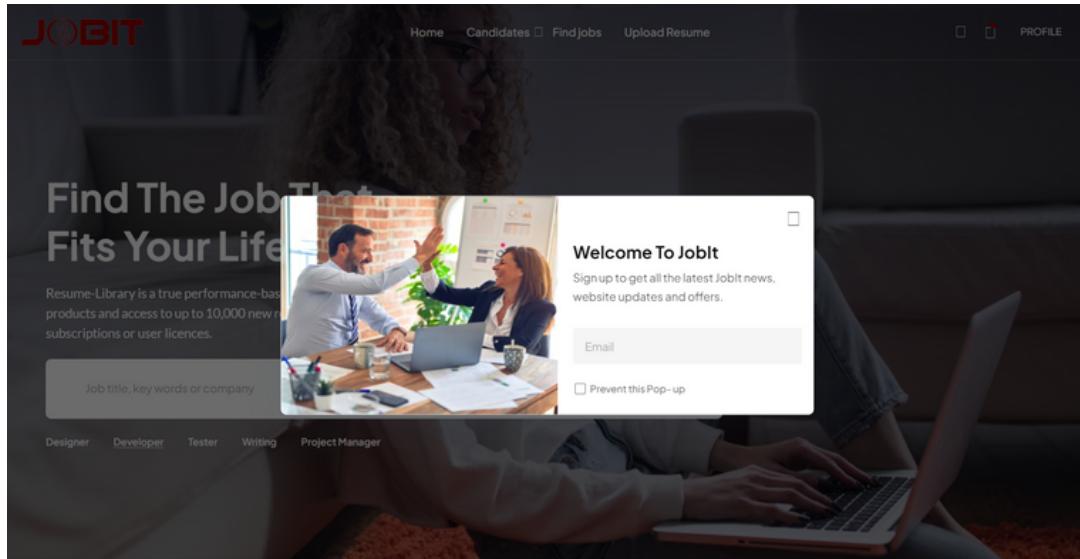


figure 5 . Screenshot of Welcome page

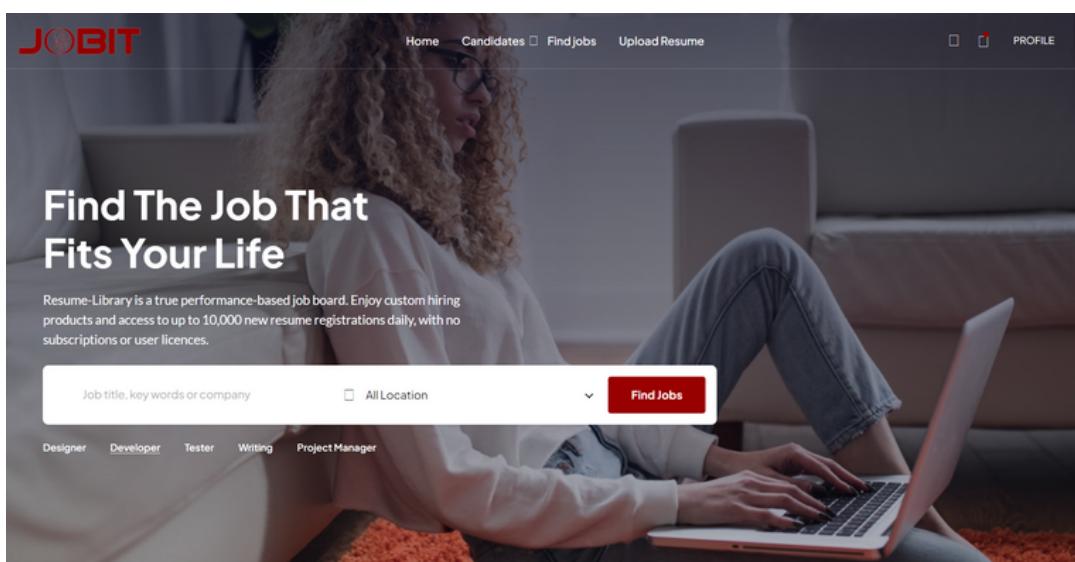


figure 6 . Screenshot of JOBIT Home page

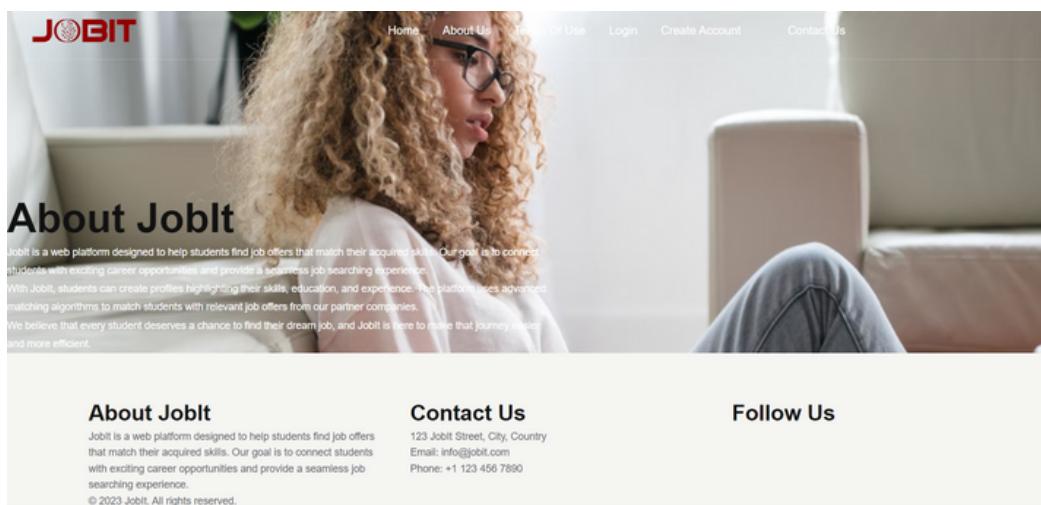


figure 7 . Screen shot of "About JOBIT" page

## 15.2 BackOffice

The screenshot shows a login form titled "Log In". At the top, there is a navigation bar with links: Home, About Us, Terms Of Use, Login, Create Account, and Contact Us. Below the navigation bar, the URL "HomeLogin" is visible. The main form has fields for "Username or email address\*" containing "jhen@jhen.com" and "Password\*" containing "\*\*\*\*\*". There are checkboxes for "Remember me" and "Forgot password?". Below the form, there is a section titled "or sign up with" featuring "Continue with Facebook" and "Continue with Google" buttons.

figure 8 .Screenshot of Sign In Page

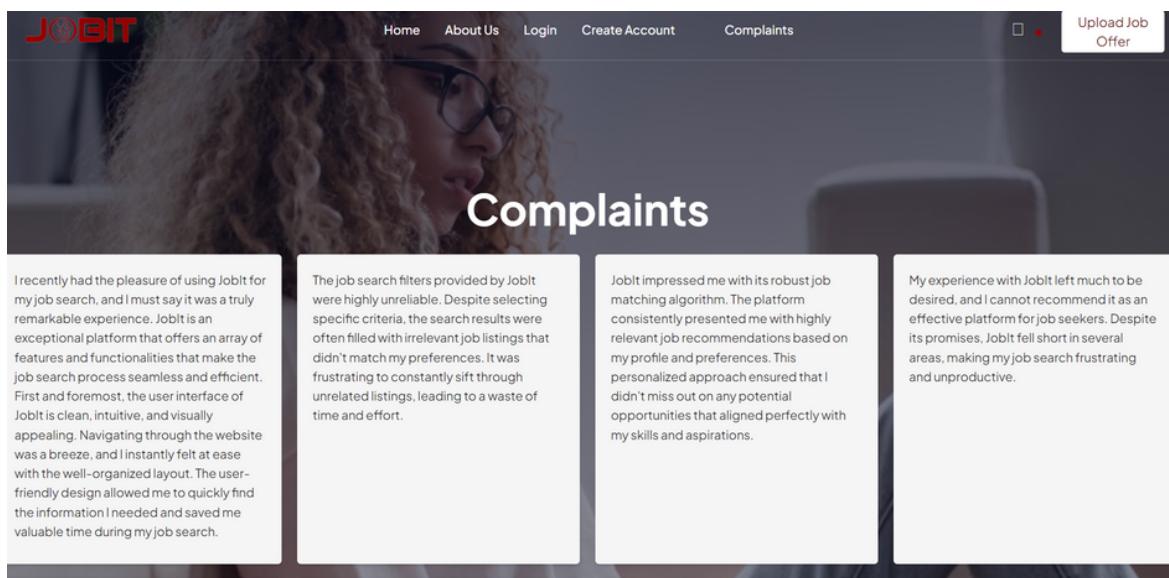


figure 9 .Screenshot of "Complaints received by the PE" Page

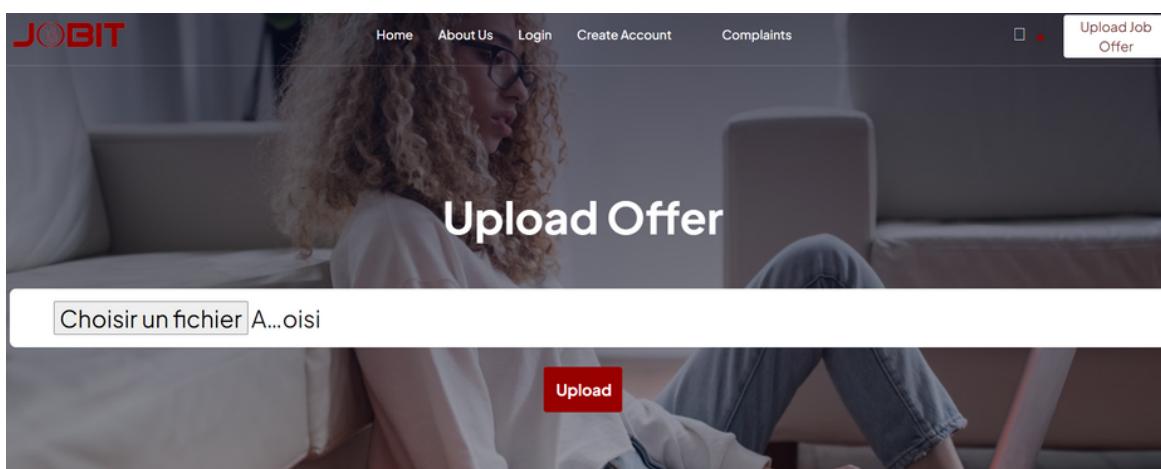


figure 10 .Screenshot of Uploading offers Page

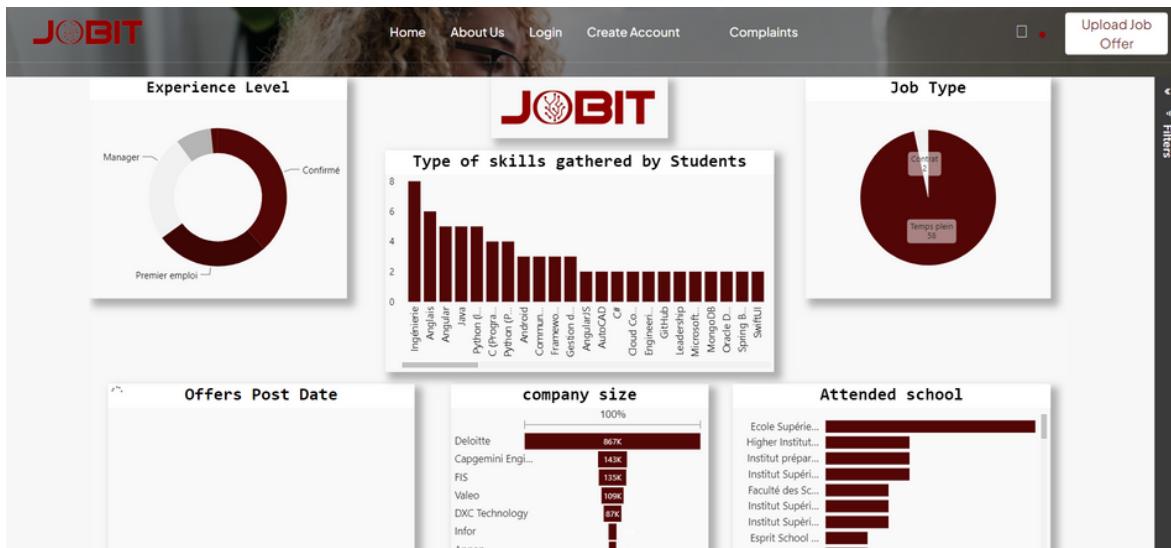


figure 11 .Screenshot of the dashboard Page

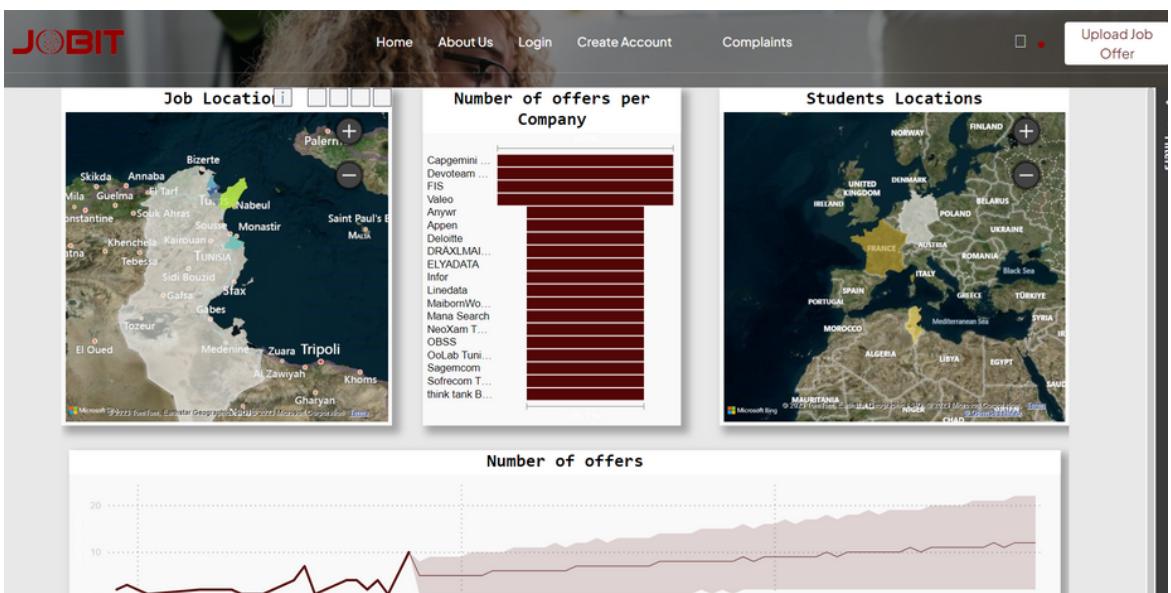


figure 12 .Screenshot of the dashboard Page

## 15.3 FrontOffice

The screenshot shows a registration form titled "Register Here". It includes fields for Username\*, Firstname, Lastname, Select your Role: (a dropdown menu), and Email. There is also a placeholder text "Create userNName here" in the first input field.

Home · Create Account

## Register Here

Username \*

Firstname

Lastname

Select your Role:

Email

figure 13 .Screenshot of the Sign Up Page for students

The screenshot shows a resume management interface. On the left is a sidebar with navigation links: Dashboard, Profile (Overview, Profile Setting), Resumes (selected), My Applied, Saved Jobs (highlighted), Alerts Jobs, Messages, Following Employers, Meeting, Change Passwords, Delete Profile, and Log Out. The main area has sections for "Resumes", "CV File" (with a "Upload" button and placeholder "Choisir un fichier Aucun fichier choisi"), "About Me" (with a rich text editor toolbar), and "Your LinkedIn Account" (with a URL input field containing "https://www.linkedin.com/in/jihen-souissi-b15a7a207/" and a "Browse" button). A status bar at the bottom shows the URL "127.0.0.1:8000/etudiant/candidates-save-jobs.html".

JOBIT

Home Candidates Findjobs Upload Resume PROFILE

Dashboard

Profile

Overview

Profile Setting

Resumes

My Applied

Saved Jobs

Alerts Jobs

Messages

Following Employers

Meeting

Change Passwords

Delete Profile

Log Out

127.0.0.1:8000/etudiant/candidates-save-jobs.html

## Resumes

CV File

Upload

Choisir un fichier Aucun fichier choisi

About Me

Your LinkedIn Account

https://www.linkedin.com/in/jihen-souissi-b15a7a207/

Browse

figure 14 .Screenshot of "filling the infos(CV, about me, LinkedIn account)" Page

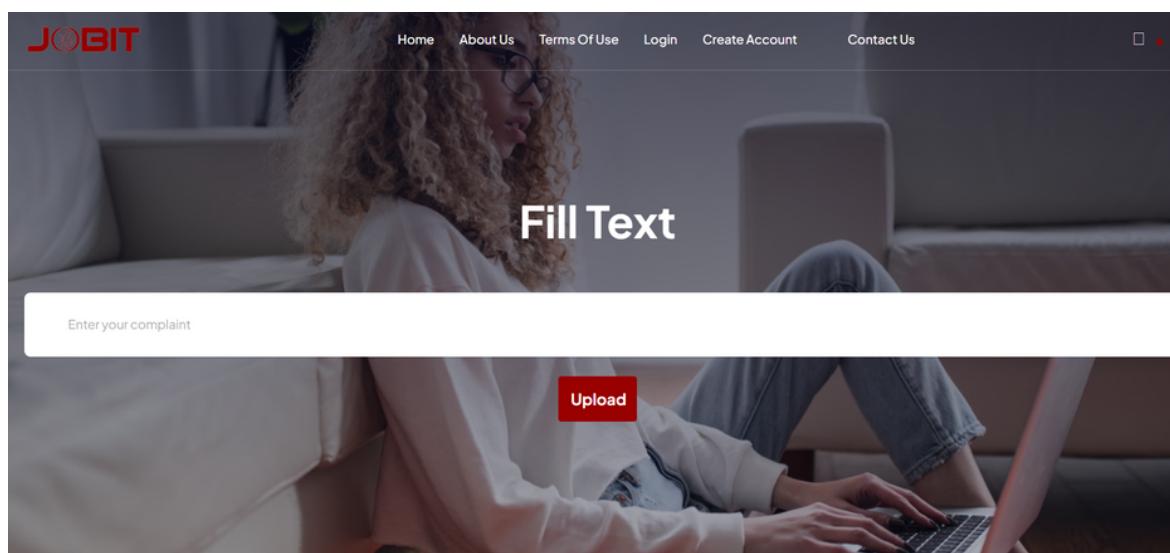


figure 15 .Screenshot of the Complaints Page

## **16. Conclusion and perspectives**

In conclusion, "JOBIT" the career center platform, driven by data science and artificial intelligence, has revolutionized job searching and recruitment.

Its advanced used algorithms match candidates with relevant and app job opportunities, saving time and increasing efficiency.

JOBIT's user-friendly interface and personalized job recommendations benefit both students, and the employability pole.

With further advancements, JOBIT has the potential to reshape career development and job placements.

# 17. REFERENCES

[1]IBM

<https://www.techtarget.com/searchitchannel/definition/IBM-International-Business-Machines>

[2]NLP <https://www.ibm.com/topics/natural-language-processing>

[3]KPI <https://www.qlik.com/us/kpi>

[4]OCR

[https://en.wikipedia.org/wiki/Optical\\_character\\_recognition](https://en.wikipedia.org/wiki/Optical_character_recognition)

[5]SMART

<https://corporatefinanceinstitute.com/resources/management/smart-goal/>

[6] imaplib <https://www.pythontutorial.net/python-basics/python-imaplib/>

[7]pytesseract <https://www.projectpro.io/recipes/what-is-pytesseract-python-library-and-do-you-install-it>

[8] ELT <https://www.ibm.com/topics/elt>



## ESPRIT SCHOOL OF ENGINEERING

**www.esprit.tn - E-mail : contact@esprit.tn**

**Siège Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889**

**Annexe : 1-2 rue André Ampère - 2083 - Pôle Technologique - El Ghazala - Tél +216 70 250 000 - Fax +216 70 685454**