Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

Dataset Creation from Job Portals using Scrapping and EDA Implementation to Enhance Employability

Priyanka Shah
S. K. Patel Institute of Management and Computer Studies, Kadi SarvaVishwavidyalaya University, Gandhinagar, India

Dr. Hardik B. Pandit
Faculty of Computer Science and Applications, Charotar University of Science & Technology, Changa, Gujarat, India

Dr. Paresh V. Virparia
Department of Computer Science, Sardar Patel University, VallabhVidyanagar, Gujarat-India

***Abstract*— The Computer Science is a highly sought-after field, offering some of the best job opportunities in the Information Technology sector through various job portals. Students often find that their skills in computing can open many doors in the rapidly evolving tech industry. There's a significant gap between the skills taught in educational settings and those demanded by the industry, largely due to the rapid evolution of technology. To address this, educational institutions need to frequently update their curricula collaborate with industry leaders to ensure students acquire relevant, up-to-date skills for the workforce. Thus, it's essential to analyze the skills that are currently in demand within the IT industry. A job portal is an online platform where companies can post job listings, offering a quick, reliable, and precise way to connect with potential employees. In this Research Paper, the Web Scrapping procedures and methods have been developed in Python to scrape employment details of Information Technology field from renowned job portals like Naukri.com, Monster.com, TimesJobs.com, Internshala.com and Myamcat.com. After scrapping, EDA (Exploratory Data Analysis) has been performed on scrapped data to summarize main characteristics of data set.**

***Keywords*—component Web Content Mining; Web Scraping; Exploratory Data Analysis; Data Preprocessing; Employability; Key skills**

I. INTRODUCTION

The World Wide Web (WWW) is a huge collection of diverse documents comprising text, images, audio, and video data [1]. Extracting data from websites has become a vital process known as web scraping. It involves retrieving information from multiple web sources, parsing the HTML content, and structuring it in a database for further analysis. The applications of web scraping encompass market research, competitor analysis, content aggregation, price monitoring, and more [2].

Python has emerged as the programming language for web scraping, owing to its simplicity, usability, and the availability of powerful libraries and frameworks tailored for this purpose. Notably, Python offers an wide range of libraries, such as BeautifulSoup, Scrapy, Selenium, and others. These libraries provide comprehensive features, including HTML parsing and seamless interaction with web pages, resulting in streamlined and efficient web scraping processes [3].

In the context of this study, our primary focus is given to web scraping job portals, especially those that are related to the Information Technology (IT) sector. During Financial year 2022, The Indian Information Technology and Business Process Management Sector has gradually grown more than 30 percent of the global outsourced BPM market. In 2022, IT sector's contribution to India's GDP was 7.4%.[4] The IT industry is a significant source of employment and creates a wide range of job opportunities. However, there is a gap between the skills that job seekers possess and what the IT industry requires of them. Having access to pertinent and comprehensive datasets is necessary for identifying and closing this gap.

Despite the fact that a number of sources offer job data, none of them provide a comprehensive list of all IT job categories. Consequently, a specialized dataset becomes necessary. In order to specifically serve the IT industry, this paper addresses this need by utilizing web scraping techniques to gather job data from various job portals. Python's strong libraries make it possible to quickly find a variety of job listings.

Exploratory Data Analysis (EDA) is a crucial first step in data science that focuses on examining and visualizing datasets to uncover patterns, spot outliers, and understand relationships between variables. In this paper, different EDA techniques have been performed on scrapped data and provided valuable results to enhance the employability [5].

II. PREVIOUS WORK

Mirjana Pejic-Bacha et al. (July 2019) focused on identifying essential skills for Industry 4.0 jobs using text mining. They applied descriptive analysis, clustering, and Jaccard similarity to job ads, revealing key roles like Supply Chain Analyst and skills such as advanced analytics and digital transformation [6].

Francisco J. García-Peñalvoet al. (February 2018) explored predicting employability using machine learning. They used

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

descriptive statistics and visualization techniques, achieving moderate prediction success and highlighting data processing improvements for better outcomes [7].

Marianne P. Ang et al. (October 2017) evaluated the BS Information System curriculum in the Philippines against industry needs. Using qualitative methods, they found an 80% alignment and recommended enhancing data analysis components to close remaining gaps [8].

David Smith et al. (2014) analysed IT job trends via web data mining of job portals. The study emphasized Java's dominance, the decline of COBOL, and the importance of SQL, calling for curriculum updates aligned with market needs [9].

K.Thamarai Selvi et al.(January 2016) investigated the impact of e-recruitment on job seekers. Using descriptive statistics, ANOVA, and the KNIME tool, they concluded that online recruitment is cost-effective and widely accessible) [10].

P Ramyaet al. (2020) presented a machine learning-based recommendation system to improve student performance. Through dataset preparation, algorithm comparison, and predictive modelling, they showed how BI and data mining can accurately predict academic success [11].

Walid Shalaby et al. (2017) proposed a deep learning and graph-based job recommendation system. Addressing cold-start and sparsity issues, the hybrid model significantly improved recommendation accuracy and user-job matching [12].

III. METHODOLOGY

Following Steps are required to create a Dataset for Information Technology Jobs from Job portals using Web Scrapping.

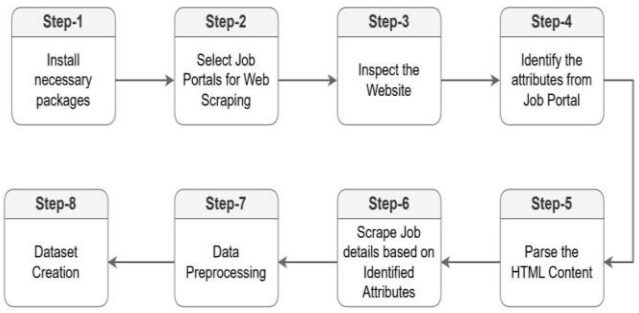

Fig. 1: Methodology to create a Dataset from Job Portals

Step 1 : Install the Essential Packages

The first step is to install the essential Python libraries and tools that will be used for web scraping. BeautifulSoup and Requests are the two most popular libraries used for web scraping in Python. While Requests is used to send HTTP requests to websites, BeautifulSoup is used to parse HTML and XML documents. The Python package manager pip can be used to install these libraries by executing the commands below in the terminal:

pip install beautifulsoup4
pip install requests

Step 2: Selecting Job Portals for Web Scraping

Numerous job portals such as Monster.com, Naukri.com, and Internshala.com provide platforms for employers to advertise job openings across various fields. The structure of a website can vary, and there are different techniques used to fetch data from it. Therefore, separate Python modules have been developed for each job portal, tailored to their specific website architecture.

Step 3: Inspect the website

In this step, the website's HTML structure must be examined to scrape the desired data. Some websites only provide static content, which consists of pre-rendered HTML pages that don't include any dynamic information. If the website has a static structure then select the Inspect Element menu from web browser then move the cursor to the desired data. After that step, copy the CSS selector and check the CSS selector by chrome extension CSS/XPATH selector. After selecting the precise elements for scrapping, web scraping tool can be used or source code can be written to extract the desired data from the website.

Some websites may implement dynamic content delivery, which involves data retrieval via APIs or other methods like WebSockets, server-sent events, or network requests. Python can be used to build modules that extract data from websites that implement dynamic content delivery. In Dynamic Content Delivery, a Python script is used to observe and replicate the API network request in order to retrieve data. This process involves monitoring the API request and simulating it within the script to access the needed information. Most of the data sent by API is in the form of JSON from which specific attributes can be extracted. Fig.2 shows the steps to inspect the website's structure.

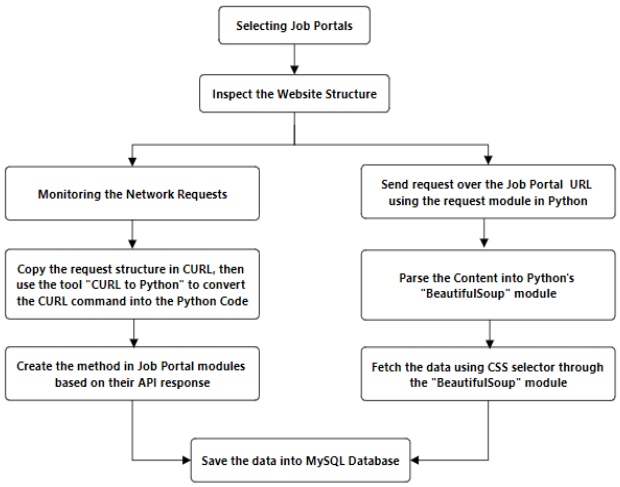

Fig. 2 Inspect the Website Structure

Step 4: Identify the attributes from Job Portals

At this stage, identify and select only those attributes from the job domain which are required. Identify the attributes from Job Portals like Job Title, Salary, Experience, Education, Key Skills, Location, JD, Source, JD_DATE, JB UNIQID.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

Step 5: Parse the HTML or JSON Data
If the website is providing static content, then a parser must be used like BeautifulSoup to parse the website's HTML data. To do this, make a BeautifulSoup object and pass the page's H TML content to it as follows:

```
import requests
from bs4 import BeautifulSoup
url = "https://jobportal.com"
response = requests.get(url)
soup=BeautifulSoup(response.content, "html.parser")
```

If the website is providing dynamic content then Python Request Library and parse the data in JSON form must be used.

```
import requests
url = "https://jobportal.com"
response = requests.get(url)
data=response.json()
```

Step 6: Identify attributes and scrape Job Details: Web scraping involves pulling information from online sources and organizing it into a usable format. This step must precede data preprocessing and analysis. For this purpose, a scraper script was created in Python, and 72 specific keywords were established, as displayed in Table 1, to extract Information Technology job listings from various job portals.

Table 1: 72 Keywords

C++	Software Developer	Application	Web Developer	J2EE	DBA
C	Information Security	Database Manager	Machine Learning	Blockchain	AWS
Web	Content developer	Security	Game Designer	Android	Cloud
PHP	Network Security Engineer	Agile	Data Scientist	Javascript	IOT
JAVA	Computer System Analyst	Data Analyst	Graphics Designer	Angular	SAP
Flutter	Mobile Application	Programmer	Software Tester	.NET	Node JS
QA	Cyber Security Engineer	Content Writer	Content Writer	SQL	Back end
Testing	Database Administrator	Automation	IOS developer	Azure	Unix
Oracle	Embedded Engineer	ASP.NET	Dot net Developer	AEM	R
Python	System Programmer	UX Designer	Quality Assurance	DevOps	Laravel
Linux	React JS Developer	Ecommerce	Data Entry Jobs	Full Stack	API
C#	Angular JS Developer	SEO executive	SMO Executive	Front End	Database

- Method creation for Job Portal Modules:
  Different modules and methods have been created for each job portal module to get the attributes because every job portal is differently structured. Separate methods have been for each job portal module to handle scraping tasks. Each method should be customized to the specific structure and requirements of the job portal.

6.1 Scrape individual Pages:
The "scrapeSinglePage" method have been implemented scrape all the job links in a specific job page.

6.2 Navigate all Pages:
The "scrapeAllPages" method have been developed to navigate through different pages within the job portal. Implement the method to jump from one page to another, following the navigation links or pagination.

6.3 Scrape Job Pages:
The "scrapeJobPage" method is used to extract attribute data from individual job pages. The required attributes have been defined to scrape, such as job titles, company names, locations, or descriptions. Implement the logic to extract the desired data from the job page HTML structure or using their internal API.

6.4 Implementing Python Threading for Concurrent Execution:
Separate threads have been created for each job portal module. And as the scraping tasks for different job portal modules run concurrently, this can improve efficiency by allowing multiple job portals to be scraped simultaneously.

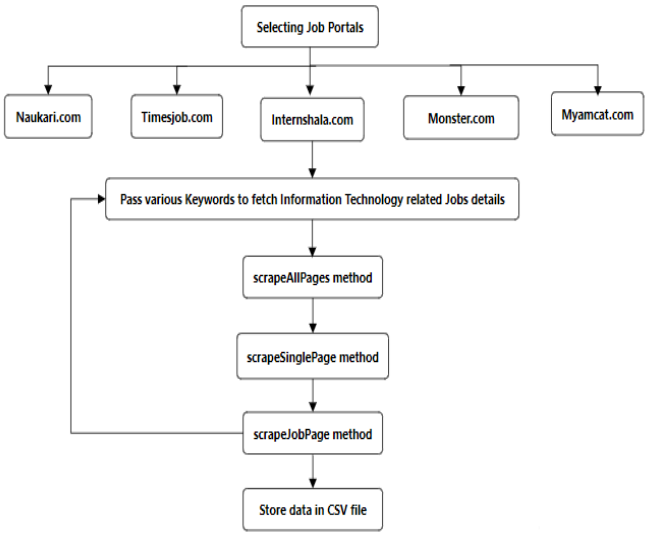

Fig. 3: Web Scraper

By utilizing separate methods for different scraping tasks, the scraping process can be modularized and make it easier to handle variations in website structure across different job portals. Implementing threading allows for concurrent execution, enabling efficient scraping of multiple job portals simultaneously, as shown in below Figure 3.

Step 7: Data Preprocessing
Data Preprocessing involves following steps:

7.1 Interpolation: The data extracted might lack cleanliness and could contain duplicates or missing values, necessitating the application of diverse techniques to rectify these issues and identify any missing data.

7.2 Feature Extraction: Not all the extracted data is pertinent for mining purposes, thus necessitating the selection of only the relevant attributes. Data preprocessing was then

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

carried out on the obtained dataset, incorporating User-Defined Functions (UDFs) to clean the data. This involved removing missing values, duplicates, as well as any extraneous characters such as new lines and tabs from the strings, ensuring the dataset was free from anomalies.

IV. RESULT

Multiple relevant websites have been explored using their URL addresses. Scrapper was generated and implemented using Python. The scrapped data set is retrieved in an CSV format. From the above methodology, total 10217 job data were scrapped.

From Scraped Data, key skills are also separated and stored in CSV and SQL databases.

On this data, EDA approach makes it simple to analyze and identify the skills and job domain that are in demand on the IT Sector, which can help job seekers and educators decide which skills to focus on acquiring.

1. Job Titles wise Word Cloud Plot:

A word cloud, or tag cloud, visually displays text data by varying the size of each word based on its frequency or significance within a text or dataset. Words that appear more often are shown in larger font, highlighting their prominence in the content. Conversely, less common words are displayed smaller, though they may still hold importance, especially if they add unique context to the subject. Job Titles like Software Developer, Graphic Designer, Content Writer, Data Entry Operator, etc. are highly demanding as shown in Figure 4.

Fig. 4 : Job Titles wise Word Cloud Plot

2. Job Titles wise Bar Plot :

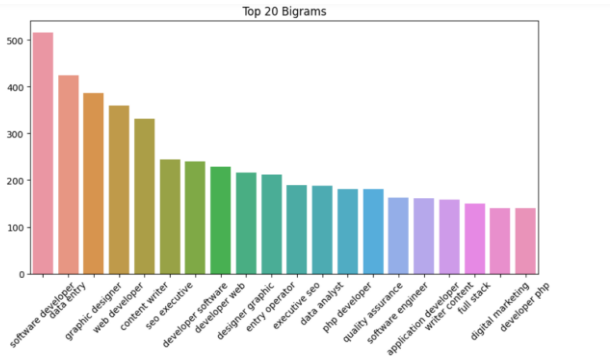

Fig. 5 : Top 20 Job Titles wise Bar Plot

Bar plot is a simple plot in EDA that is used to plot categorical variable on the x-axis and numerical variable on y-axis and also discover the relationship between both variables. Top 20 Job Titles like Software Developer, Data Entry Operator, Graphic Designer, Web Developer ,Content Writer, etc. are highly demanding with number of occurrences on y-axis as shown in Figure 5.

3. Key skills wise Bar Plot :

Bar Plot of Highly required key skills is also generated as shown in Figure 6. "CI" is the most frequent word: This could indicate a focus on continuous integration or continuous improvement in the data source. Other high-frequency words: "environment," "sql," "testing," "http," "data," "security," "web," "java," and "skills" suggest a technical focus. The presence of "marketing" and "communication" suggests there might also be a business or marketing component to the data.

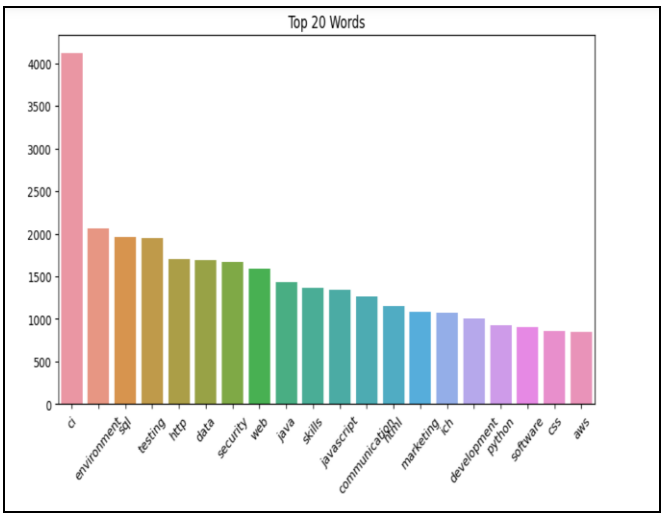

Fig 6 : Top 20 Key skills wise Bar Plot

4. Top 20 Bigram of Key Skills:

A "bigram" refers to a sequence of two adjacent words in a text corpus. It is useful for identifying collocations (words that frequently occur together) and extracting meaningful phrases from text. The Figure 7 shows the top 20 bigrams found in a dataset. Bigrams are pairs of consecutive words. The height of each bar represents the frequency of that bigram in the dataset. Most frequent bigrams: The most frequent bigrams are "ci environment", "communication skills" and "testing ci". This suggests that the dataset might be related to technical topics, particularly in the area of software development and testing.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 07, July - 2025

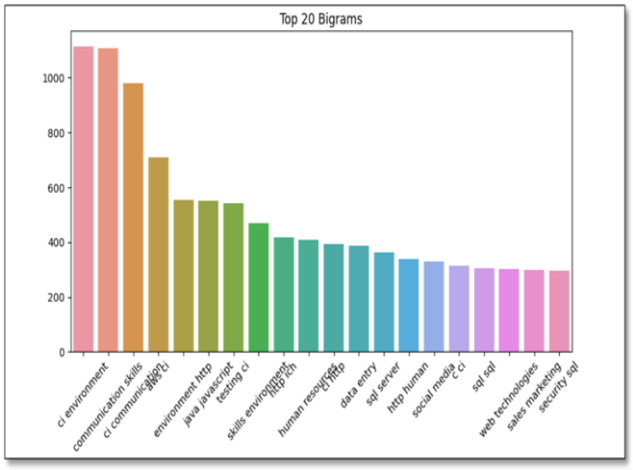

Fig 7 : Top 20 Key skills wise Bigram

V. CONCLUSION

In conclusion, the dataset generated through this web scraping method, focusing on job data from various job portals, holds significant potential for bridging the gap between academia and industry requirements. This study aims to use Python's advantages and its strong libraries to scrape data from job portals, with a focus on the IT sector. We aim to close the skill set gap between industry requirements and available skill sets by building a comprehensive dataset. By analyzing this dataset, researchers and educators can gain valuable insights into the skills that are in high demand in the job market. The availability of such a dataset enables a deeper understanding of the skills that are repeatedly emphasized in job listings, providing valuable information for educational institutions and students. By aligning their curriculum and skill development programs with the identified in-demand skills, educational institutions can better prepare students for industry-level job opportunities. The skills are also identified that employers are required the most through this analysis. The abilities are also determined that are in great demand for industry-level job opportunities by identifying the skills that are frequent in the scraped data. In Future enhancement, the knowledge provided by this web scraping method has the potential to support research, guide curriculum development, and improve student skill development.

REFERENCES

[1] A. Richlin, S. Jebakumari, and N. J. Goldena, “A Survey on Web Content Mining Methods and Applications for Perfect Catch Responses,” Int. Res. J. Eng. Technol., p. 407, 2008.

[2] Fatmasari, Y. N. Kunang, and S. D. Purnamasari, “Web Scraping Techniques to Collect Weather Data in South Sumatera,” Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018, vol. 17, pp. 385–390, 2019, doi: 10.1109/ICECOS.2018.8605202.

[3] P. Thota and E. Ramez, “Web Scraping of COVID-19 News Stories to Create Datasets for Sentiment and Emotion Analysis,” ACM Int. Conf. Proceeding Ser., pp. 306–314, 2021, doi: 10.1145/3453892.3461333.

[4] M. A. Khder, “Web scraping or web crawling: State of art, techniques, approaches and application,” Int. J. Adv. Soft Comput. its Appl., vol. 13, no. 3, pp. 144–168, 2021, doi: 10.15849/ijasca.211128.11.

[5] M. Komorowski, D. C. Marshall, J. D. Salciccioli and Y. Crutain, Chapter 15- Exploratory Data Analysis, MIT Critical Data, Secondary Analysis of Electronic Health Records, pp. 185-203, DOI 10.1007/978-3-319-43742-2_15

[6] M. Pejic-bach, T. Bertoncel, M. Meško, and Ž. Krstić, “International Journal of Information Management Text mining of industry 4 . 0 job advertisements,” Int. J. Inf. Manage., vol. 50, no. July, pp. 416–431, 2020.

[7] F. García-Peñalvo, J. Cruz-Benito, M. Martín-González, A. Vázquez-Ingelmo, J. C. Sánchez-Prieto, and R. Therón, “Proposing a Machine Learning Approach to Analyze and Predict Employment and its Factors,” Int. J. Interact. Multimed. Artif. Intell., vol. 5, no. 2, p. 39, 2018, doi: 10.9781/ijimai.2018.02.002.

[8] M. P. Ang, “EVALUATING THE RESPONSIVENESS OF BS IS CURRICULUM TO INDUSTRY SKILL REQUIREMENTS Aileen L . Rillon , MBA , MIM A Research Report Submitted to the Ateneo De Naga University Research Council October 2017.

[9] D. T Smith and A. Ali, “Analyzing Computer Programming Job Trend Using Web Data Mining,” Issues Informing Sci. Inf. Technol., vol. 11, pp. 203–214, 2014, doi: 10.28945/1989.

[10] K. T. Selvi and R. Ramya, “an Impact on Electronic-Recruitment and Its Perception Towards Job Portal Function Through Search Engines Among Job Seekers Using Knime Data Mining Tool,” Int. J. Res. Dev. Technol., no. 5, pp. 10–18, 2016.

[11] P. Ramya, S. G. Balakrishnan, and M. Kannan, “Recommendation system to improve students performance using machine learning,” IOP Conf. Ser. Mater. Sci. Eng., vol. 872, no. 1, 2020, doi: 10.1088/1757-899X/872/1/012038.

[12] W. Shalaby et al., “Help me find a job: A graph-based approach for job recommendation at scale,” Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017, vol. 2018-Janua, pp. 1544–1553, 2017, doi: 10.1109/BigData.2017.8258088.