



# End-to-End Analysis of LinkedIn Job Postings Using Python and Power BI

**Naru Lavanya**

Reg. No. 24Q71F0037

[krishnanlavanya945@gmail.com](mailto:krishnanlavanya945@gmail.com)

Department of Master of Computer Applications

Avanthi Institute of Engineering and Technology (Autonomous)

Vizianagaram, Andhra Pradesh, India

*Under the guidance of Mr. Ch. Bhupathi Raju, MCA, Assistant Professor*

[challa.bhupati@gmail.com](mailto:challa.bhupati@gmail.com)

**Abstract**—In today's competitive job market, analysing job trends is essential for both job seekers and organisations. This project focuses on extracting meaningful insights from LinkedIn job data. The system collects job-posting data including job titles, locations, required skills, company details, and salary information. Python is used for data collection, cleaning, preprocessing, and analysis, with libraries such as Pandas, NumPy, and Matplotlib applied to identify patterns and trends in job postings. The processed data is then visualised using Power BI to create interactive dashboards that provide clear insights into job demand, skill requirements, location-based opportunities, and industry trends. The project helps users understand current job-market trends, identify in-demand skills, and make informed career decisions, and it assists organisations in analysing hiring patterns and improving recruitment strategies. Unlike platforms that present raw listings without structured analytics, the proposed system provides a complete end-to-end pipeline—from data extraction through preprocessing and analysis to interactive visualisation—within a unified workflow. The system was validated through eight functional test cases covering dataset loading, data cleaning, skill extraction, dashboard loading, filtering, skill search, data download, and error handling, all of which passed. Overall, the system provides a user-friendly and efficient solution for job-data analysis and smarter decision-making in the evolving job market.

**Keywords**—Job Market Analysis; Data Analytics; LinkedIn Job Postings; Python; Power BI; Data Visualisation; Web Scraping; Interactive Dashboard.

## I. INTRODUCTION

In the modern digital era, the job market has become highly competitive and dynamic, with thousands of opportunities posted online every day. Platforms such as LinkedIn play a crucial role in connecting job seekers with employers by providing detailed job postings across various industries and locations. However, due to the vast amount of available data, it becomes difficult for individuals and organisations to manually analyse job trends, required skills, and hiring patterns.

This project aims to address this challenge by leveraging data-analysis and visualisation techniques. The system collects job-related data such as job titles, company names, locations, required skills, and salary information, and processes it to extract meaningful insights. By using Python for data preprocessing and



analysis, and Power BI for visualisation, the project transforms raw data into interactive and easy-to-understand dashboards.

The main objective is to help job seekers identify in-demand skills, explore job opportunities across different regions, and make informed career decisions, while also assisting organisations in understanding recruitment trends and improving hiring strategies. Overall, the project demonstrates how data analytics can be effectively applied to a real-world problem, providing valuable insights into the evolving job market and enabling smarter decision-making.

## II. LITERATURE SURVEY

The rapid growth of online recruitment platforms such as LinkedIn has generated a vast amount of job-related data, making it an important source for analysing employment trends. Several researchers and industry practitioners have explored techniques to extract meaningful insights from job postings using data analytics, machine learning, and visualisation tools. Studies in job-market analysis highlight the importance of using programming tools such as Python for data collection, preprocessing, and analysis; libraries such as Pandas and NumPy are widely used for handling structured datasets, while visualisation tools such as Power BI and Tableau help present insights through dashboards. Research has shown that analysing job postings can reveal trends such as in-demand skills, the geographic distribution of jobs, and industry-specific requirements.

Previous works have also focused on web-scraping techniques using Selenium and BeautifulSoup to collect job data from online platforms, and some studies applied machine-learning algorithms to predict job trends and recommend career paths. However, many existing systems focus only on partial aspects such as data collection or visualisation, lacking a complete end-to-end solution that integrates all stages from data extraction to interactive visualisation. This project builds on these approaches by providing a comprehensive system that performs data collection, preprocessing, analysis, and visualisation in a unified workflow, emphasising user-friendly dashboards and skill-based analysis for both job seekers and organisations.

**TABLE I. REPRESENTATIVE PRIOR WORK AND TOOLS**

S.No	Area	Tools / Techniques	Note
1	Data collection	Selenium, BeautifulSoup	Web scraping of job postings
2	Data handling	Pandas, NumPy	Structured dataset processing
3	Visualisation	Power BI, Tableau	Dashboard-based insight delivery
4	Job-trend prediction	Machine-learning algorithms	Career-path recommendation
5	Job-market studies	Data-mining techniques	Trend and skill analysis
6	This project	Python + Power BI + Streamlit	Unified end-to-end pipeline



### III. EXISTING SYSTEM AND PROPOSED SYSTEM

#### *A. Existing System*

Currently, job seekers rely on platforms such as LinkedIn to search for job opportunities. While these platforms provide large amounts of job data, they lack structured analytical insights, leaving users to interpret raw listings manually. There is no centralised analysis of job trends, and identifying in-demand skills or comparing opportunities across regions is difficult and time-consuming without integrated tooling.

#### **Limitations of the existing system:**

- No centralised analysis of job trends.
- Difficult to identify in-demand skills.
- Time-consuming manual search and comparison.
- Lack of interactive visualisation for decision-making.
- No integrated system combining data extraction, analysis, and visualisation.

#### *B. Proposed System*

The proposed system provides an end-to-end solution that collects job-posting data, processes it, and generates meaningful insights through visualisation. It automates data collection and preprocessing, applies analytical techniques to identify trends, and presents the results through interactive dashboards built with Power BI and Streamlit, giving both job seekers and organisations a clear and practical view of the job market.

#### **Advantages of the proposed system:**

- Automated data collection and preprocessing.
- Centralised analysis of job trends and in-demand skills.
- Interactive dashboards for fast, informed decision-making.
- Unified pipeline from extraction to visualisation.
- Built with widely available, low-cost tools (Python, Pandas, Power BI).
- Useful for both job seekers and recruiting organisations.

### IV. SYSTEM ANALYSIS AND DESIGN

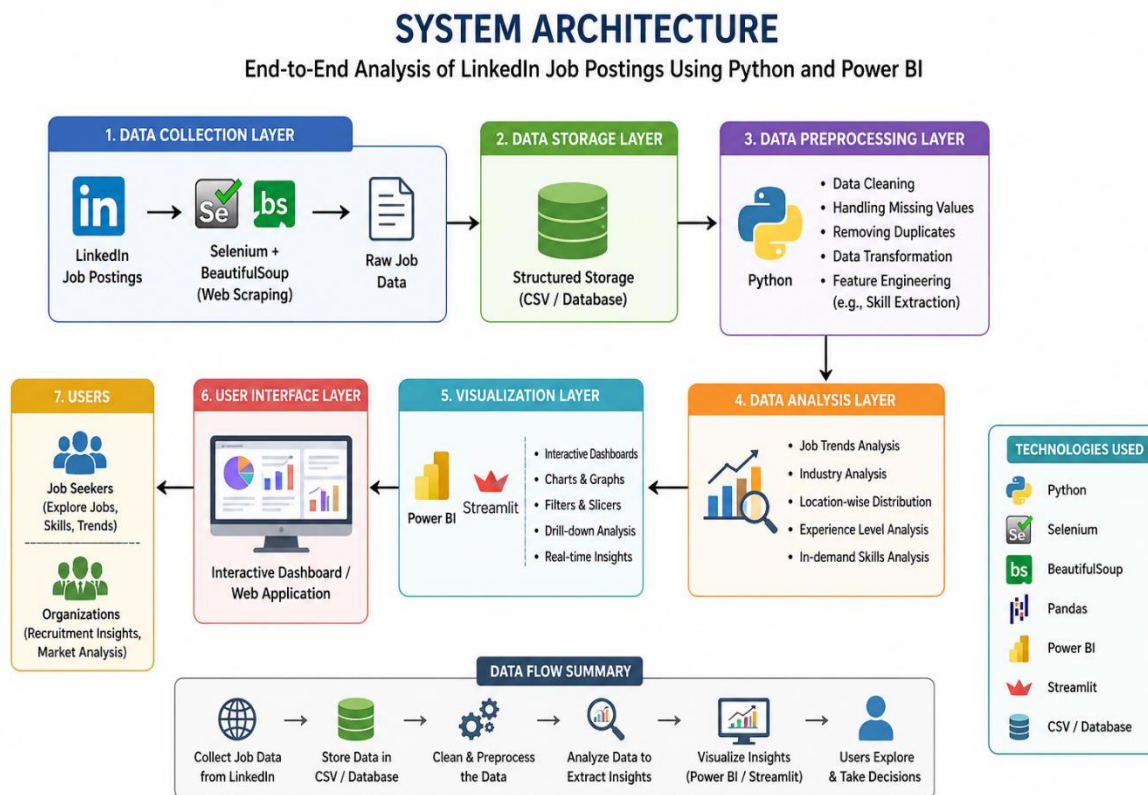
#### *A. Requirements and Feasibility*

System analysis focuses on identifying how job data is collected, processed, analysed, and presented efficiently. Functionally, the system must collect job-posting data, clean and preprocess it (handling missing values, removing duplicates, formatting), perform analytical processing to identify trends, and present interactive dashboards with filtering and search. The project is feasible because tools such as Python, Pandas, and Power BI are widely available and efficient and require no expensive infrastructure; the prototype targets Python 3.10 or above with a standard code editor.

#### *B. System Architecture*



The architecture describes how data flows through the components to form a complete analytical pipeline. Raw job data is collected and then cleaned with Python—handling missing values, removing duplicates, and formatting. Analytical techniques are applied using Pandas and NumPy to identify trends such as in-demand skills, location-based opportunities, and industry patterns. The processed data is visualised with Power BI and Streamlit, where interactive dashboards present the insights, supporting filtering, skill-based search, and data download.



### C. Workflow

The end-to-end workflow is: collect job-posting data; clean and preprocess it with Python; analyse it using Pandas and NumPy to extract trends; and visualise the results through interactive Power BI and Streamlit dashboards. Users can filter results, search by skill, and download the underlying data, while invalid inputs are handled with clear error messages.

## V. SYSTEM IMPLEMENTATION

### A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python (3.10 or above)



Component	Technology / Tool
Data Handling / Analysis	Pandas, NumPy
Visualisation Libraries	Matplotlib, Plotly
Web Scraping	Selenium, BeautifulSoup
Dashboards	Power BI, Streamlit
Data Format	CSV job-postings dataset
Development Tool	Code editor / IDE

### B. Implementation Details

The implementation follows the analytical pipeline. Job-posting data is collected (with web-scraping tools such as Selenium and BeautifulSoup where applicable) and loaded from a CSV dataset. Python with Pandas and NumPy is used to clean the data—removing null values and duplicates and formatting fields—and to perform the analysis that identifies trends such as in-demand skills, location-based opportunities, and industry-specific requirements. Matplotlib and Plotly support analytical charts, and the results are surfaced through Power BI and a Streamlit application that provides interactive dashboards.

### C. Dashboards and Interaction

The Power BI and Streamlit dashboards present job demand, skill requirements, location-based opportunities, and industry trends. Users can apply filters to narrow results, search by a specific skill (for example, jobs requiring Python), and download the filtered dataset as a CSV file. The dashboards turn raw, unstructured job data into clear, actionable insights for both job seekers and organisations.

## VI. SYSTEM TESTING AND RESULTS

The system was validated through eight functional test cases covering dataset loading, data cleaning, skill extraction, dashboard loading, filter functionality, skill search, data download, and error handling. All test cases passed and behaved as expected.

TABLE III. FUNCTIONAL TEST CASES

ID	Description	Input	Expected Output	Result
TC01	Load dataset	CSV file	Data loaded successfully	Pass
TC02	Data cleaning	Raw dataset	Clean data, no null/duplicates	Pass
TC03	Skill extraction	Job description	Skills identified correctly	Pass
TC04	Dashboard loading	User opens app	Dashboard displays charts	Pass
TC05	Filter functionality	Select filters	Updated results by filter	Pass

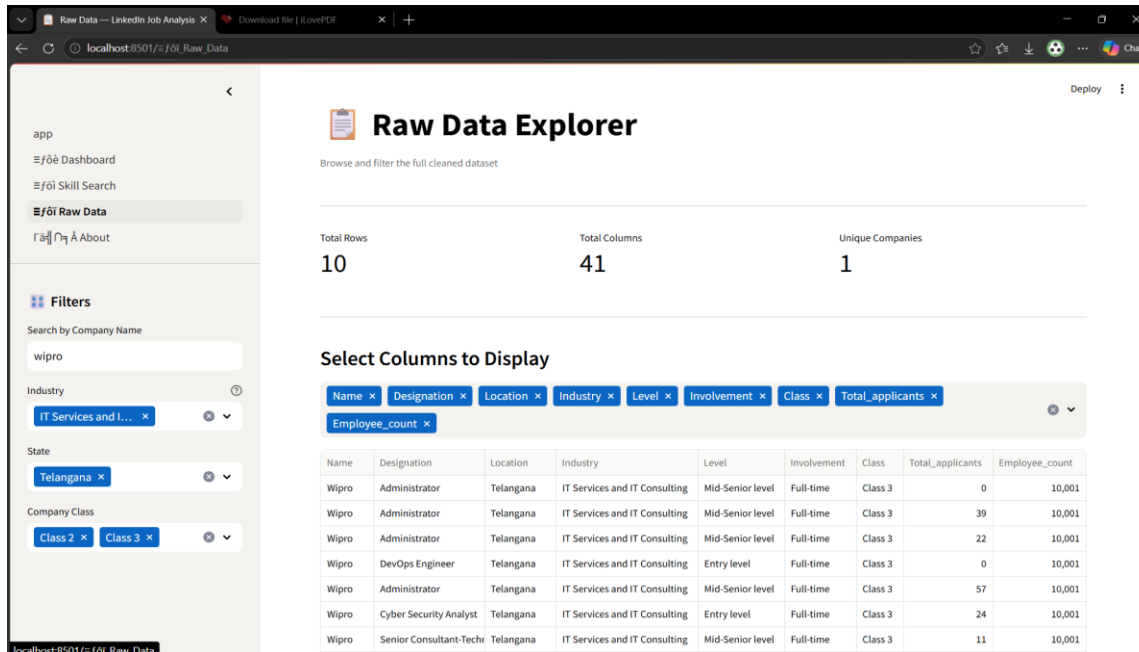


ID	Description	Input	Expected Output	Result
TC06	Skill search	Select “Python”	Jobs with Python displayed	Pass
TC08	Error handling	Invalid input	Proper error message	Pass

### A. Result and Analysis

The implemented system loads and cleans the job-postings dataset, extracts skills from job descriptions, and presents interactive dashboards that allow filtering, skill-based search, and data download. By integrating data collection, preprocessing, analysis, and visualisation in one workflow, the system reveals job-market trends such as in-demand skills, geographic distribution of opportunities, and industry-specific requirements. The source reports these outcomes qualitatively; no specific numeric metrics are claimed here, and the depth of insight depends on the size and freshness of the collected job data.

*Representative screenshots from the prototype implementation:*



*Fig. 1. Dataset loading and cleaning.*

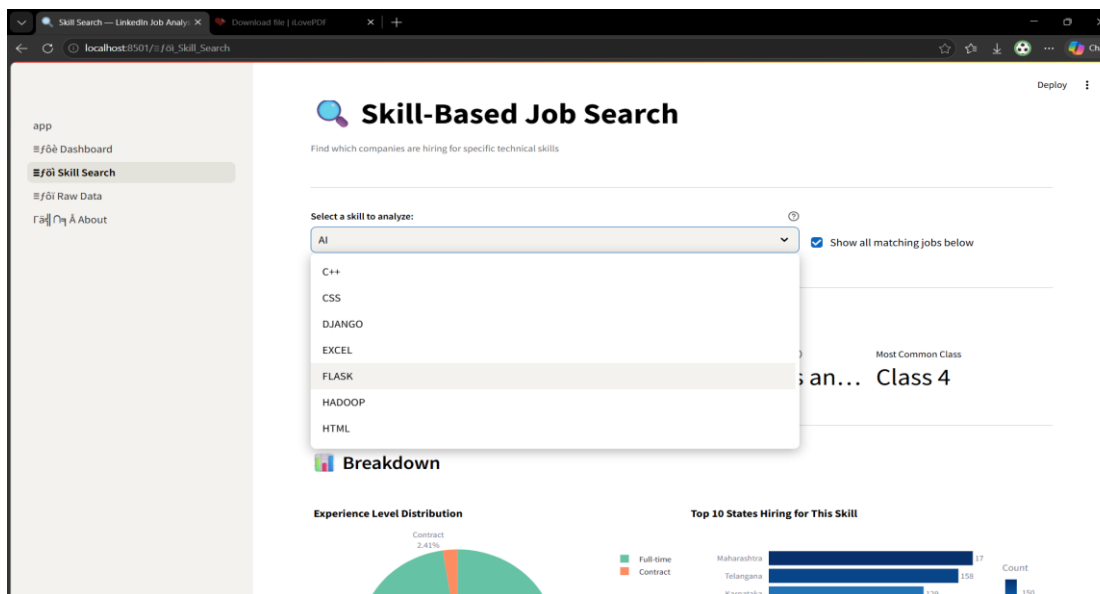


Fig. 4. Skill-based filtering and data download.

## VII. CONCLUSION AND FUTURE SCOPE

This project demonstrates how data analytics can be effectively applied to analyse LinkedIn job postings using Python and Power BI. By collecting job-posting data and processing it through a unified pipeline of cleaning, preprocessing, analysis, and visualisation, the system transforms raw, unstructured listings into clear and actionable insights. The interactive dashboards help job seekers identify in-demand skills and explore opportunities across regions, while also assisting organisations in understanding recruitment trends and improving hiring strategies. Compared with manual searching on job platforms, the system provides a centralised, efficient, and user-friendly solution for job-data analysis, and functional testing confirmed that all modules behave as expected.

Future work can extend the system in several directions: automating continuous data collection so dashboards stay current, expanding the dataset across more platforms and regions, applying machine-learning models to predict job trends and recommend career paths, adding richer skill-extraction using natural-language processing, and deploying the dashboards on the cloud for broader accessibility. These enhancements would further improve the system's practical value for both job seekers and organisations.

## REFERENCES

- [1] A. Kumar and S. Reddy, "Analysis of Job Market Trends Using Data Mining Techniques," *International Journal of Computer Applications*, vol. 178, no. 7, pp. 15–20, 2021.
- [2] M. Sharma et al., "Job Trend Prediction Using Machine Learning," *International Journal of Data Science*, vol. 5, no. 2, pp. 45–52, 2023.
- [3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2016.
- [4] Power BI Documentation, Microsoft. [Online]. Available: <https://powerbi.microsoft.com>
- [5] Python Documentation. [Online]. Available: <https://docs.python.org/3/>



- [6] Pandas Documentation. [Online]. Available: <https://pandas.pydata.org>
- [7] NumPy Documentation. [Online]. Available: <https://numpy.org>
- [8] Streamlit Documentation. [Online]. Available: <https://streamlit.io>
- [9] Matplotlib Documentation. [Online]. Available: <https://matplotlib.org>
- [10] Plotly Documentation. [Online]. Available: <https://plotly.com>