

JobX

A MICRO-PROJECT REPORT

submitted by

Achuth S Pai SCM22CA001
Vasudev Varma SCM22CA010
Jopaul Jose SCM22CA019
Vishnu Prasad G SCM22CA031
to

The APJ Abdul Kalam Technological University

in partial completion of the criteria for the degree award

of

B.Tech

**COMPUTER SCIENCE AND ARTIFICIAL
INTELLIGENCE**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCMS SCHOOL OF ENGINEERING AND TECHNOLOGY
(Affiliated to APJ Abdul Kalam Technological University)
VIDYA NAGAR, PALISSERY, KARUKUTTY
ERNAKULAM – 683582

December 2023

DECLARATION

We undersigned hereby declare that the project report JobX, submitted for partial fulfilment of the requirements for the award of the degree of Bachelor of Technology(Minor in CSE) of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under the supervision of Ms **Shali Sara Abraham**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Ernakulum

Signature

Date:

Name of the students
Achuth S Pai
Vasudev Varma
Jopaul Jose
Vishnu Prasad G

DEPARTMENT OF **COMPUTER SCIENCE AND ENGINEERING**

SCMS SCHOOL OF ENGINEERING AND TECHNOLOGY

Vidya Nagar, Karukutty, Kerala – 683 576



CERTIFICATE

This is to certify that the report entitled JobX, submitted by Achuth S Pai, Vasudev Varma, Jopaul Jose and Vishnu Prasad G to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of **B.Tech in Computer Science and Artificial Intelligence** is a bonafide record of the project work carried out by him/her under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Coordinator

Ms Shali Sara Abraham

Assistant Professor

Department of CSE

SSET

Head of the Department

Dr. Sonal Ayyappan

Associate Professor

Department of AI& DS

SSET

ACKNOWLEDGEMENT

We are greatly indebted to **Dr.Anitha G.Pillai**, Principal, SCMS School of Engineering and Technology, Ernakulam and **Dr. Sonal Ayyapan**, Associate Professor and Head of Department, Department of Artificial intelligence and Data science, SCMS School of Engineering and Technology, who wholeheartedly permitted us to conduct this project.

We would like to thank our guide, **Ms Shali Abraham** Assistant Professor, Department of Computer Science and Engineering, SCMS School of Engineering and Technology who gave us valuable guidance. We would like to express sincere gratitude to all the teachers of the Department of Computer Science who gave us moral and technical support. We would like to thank the supporting staff in the computer lab whose dedicated work kept the lab working smoothly thus enabling us to have access to various resources which helped us understand more about the project topic. We would also like to extend our gratitude to all friends and family for providing us with the necessary resources and support.

ABSTRACT

This project introduces a job-matching system designed to streamline career exploration by aligning user-entered skills and abilities with relevant job roles. Utilizing natural language processing (NLP), the system interprets user inputs, identifying key skills and attributes which are then analyzed to suggest suitable job roles and fields. By focusing on matching user competencies with the requirements of various professions, this system offers tailored job recommendations, enabling users to discover roles that align closely with their unique skillsets. This approach simplifies the job search process, making it more intuitive and personalized for individuals seeking roles that fit their strengths.



CONTENTS

Contents	PageNo.
Acknowledgement	i
Abstract	ii
1. Introduction	4
1.1 General Background	4
1.2 Objective	4
1.3 Scope of Project	4
2. Literature Survey	5
2.1 The Rationale of Study	5
2.2 Objective of Study	5
2.3 Feasibility Study	5
3. Existing and Proposed System	6
3.1 Existing System	6
3.2 Proposed System	6
3.3 Methodology	6
4. Implementation Details	7
4.1 System Requirement Specification	7
4.2 Platform and IDE	7
4.3 Application Development and Implementation	7
5. System Design	9
6. Experimental Results	12
6.1 Web Pages	12
6.2 Conclusion	14
7. References	16
8. Abbreviations	18

Introduction

General background

With the rapid growth of job markets and the evolving nature of required skill sets, job seekers are often overwhelmed when trying to identify suitable career paths. Traditional job search methods can be time-consuming, relying on broad filtering tools that may not consider the nuances of individual skill sets. Advances in natural language processing (NLP) allow for smarter, more personalized approaches to job matching, making it possible to directly interpret user-defined skills and connect them with relevant opportunities. This project aims to leverage NLP technology to simplify job searches by mapping user skills to potential job roles, facilitating an efficient, tailored job-matching experience.

Objectives

The primary objective of this project is to develop an application that matches users with job roles based on their individual skills and abilities. By analyzing keywords from user input, the system aims to recommend roles and fields where the user's competencies are most applicable. This application seeks to eliminate the guesswork involved in career exploration by providing personalized

Scope of project

This project covers the development of a job-matching application that utilizes NLP to interpret user input, focusing on keywords representing skills and abilities. The scope includes building a user-friendly interface where users can enter their skillset and receive a list of matching roles and job fields. The system's recommendations are based on skill-to-role matching, allowing users to discover relevant career paths without having to navigate complex job filters. Future enhancements may include integrating additional features, such as career tips, related skills for career advancement, and links to job listings from external platforms.

Literature Survey

Rationale of Study

The job market is increasingly competitive, with an overwhelming number of job roles requiring specialized skills. Traditional job search platforms often rely on broad categories and limited filters, making it difficult for users to find roles that align precisely with their unique skill sets. By leveraging natural language processing (NLP), this study aims to create a more nuanced, user-focused job-matching system that can interpret and match individual skills to relevant roles more accurately. The rationale behind this study is to address the gap in personalized job search solutions and provide an efficient, user-friendly approach to career discovery.

Objective of Study

The objective of this study is to develop a job-matching system that understands user-defined skills and capabilities through NLP. By analyzing keywords and identifying relevant patterns in user input, the study seeks to create a system that suggests job roles and fields based on the individual's strengths. Ultimately, the goal is to streamline the job search process and assist users in exploring career options that align closely with their competencies, making job matching more intuitive and accessible.

Feasibility of Study

The feasibility of this study is high, as recent advancements in NLP and machine learning frameworks provide the necessary tools for implementing a skill-based job-matching system. Open-source NLP libraries, such as spaCy or Hugging Face's Transformers, allow for efficient text analysis and keyword extraction, making it practical to develop an accurate recommendation system within a reasonable timeframe. Additionally, modern development platforms enable the creation of scalable, user-friendly interfaces, ensuring that the system can be deployed as a mobile or web application. With these tools readily available, the study is both technically and economically feasible to pursue.

Existing and Proposed System

Existing System

Current job search platforms typically rely on filters such as industry, location, experience level, and salary range. While effective in narrowing options, these systems often lack personalization and do not consider individual skillsets in depth. Keyword-based searches on job portals may produce generalized results, forcing users to sift through multiple listings to find suitable roles. Additionally, some platforms use basic keyword matching, which may lead to irrelevant suggestions, as they do not fully understand the context or the specific skills needed for each job.

Proposed System

The proposed system aims to address these limitations by leveraging natural language processing (NLP) to create a job-matching tool that analyzes user-defined skills and abilities. Rather than filtering by broad categories, this system interprets user input to identify key skills and matches these with roles that require similar competencies. This approach enables a more personalized and accurate job recommendation experience. The proposed system will provide users with a targeted list of job roles and fields aligned with their unique skill sets, helping them make informed career choices without the need for extensive filtering or manual search.

Methodology

1. **Data Collection:** Gather a dataset of job descriptions and required skills across various industries to train the NLP model. This data serves as the foundation for understanding which skills are essential for different roles.
2. **Keyword Extraction:** Use NLP techniques to analyze user input and extract keywords representing skills and abilities. Libraries like spaCy or Hugging Face Transformers will be employed to handle natural language understanding and keyword extraction.
3. **Skill Matching:** Once keywords are extracted, they will be compared with the job dataset to identify roles that match the user's skill set. This matching process will focus on finding the highest degree of alignment between user-defined skills and job requirements.
4. **Recommendation Generation:** Based on the skill matching, the system will generate a list of recommended job roles for the user, ranked according to relevance.
5. **User Interface:** Develop a user-friendly interface where users can input their skills and receive tailored job recommendations. This interface will be accessible on web and/or mobile platforms.

This methodology ensures that the system is both efficient and accurate, providing users with a streamlined way to find jobs that suit their skills.

Implementation Details

System Requirement Specifications

To ensure optimal performance, the job-matching system requires a robust server setup and efficient software configurations. The minimum hardware and software specifications for the server running the NLP model are as follows:

Hardware Requirements:

RAM: Minimum of 12 GB to handle NLP processing and data matching tasks efficiently.

Processor: Multi-core processor (Intel i5/i7 or AMD Ryzen 5 equivalent or higher) to support parallel processing.

Storage: At least 100 GB of storage to accommodate the job description dataset, user data, and model files.

GPU (optional but recommended): For faster NLP model training and real-time matching, a GPU with at least 4 GB of VRAM is beneficial.

Software Requirements:

Operating System: Linux (Ubuntu 20.04 or later) for server stability and compatibility with NLP libraries; Windows or macOS for development and testing.

NLP Libraries: spaCy, Hugging Face Transformers, or similar NLP frameworks to support text analysis and keyword extraction.

Programming Language: Python 3.8 or later for model development and data handling.

Database: MySQL or PostgreSQL for storing job data and user input records.

Server Environment: Apache or Nginx for web server management, if a web-based interface is used.

Platform and IDE

Development Platform: The primary development will occur on Python, leveraging libraries like TensorFlow or PyTorch for NLP model training and job-matching algorithms.

Integrated Development Environment (IDE): Visual Studio Code or PyCharm, chosen for their extensive plugin support for Python, database management, and debugging capabilities.

Version Control: Git for collaborative development and version management.

Application Development and Implementation

1. Backend Development:

Develop the NLP pipeline using spaCy or Hugging Face models to extract relevant keywords from user input and match them with job data.

Implement skill-to-role matching algorithms that process user-entered skills and cross-reference them with job requirements in the database.

Set up a REST API to handle requests between the front end and the backend, enabling real-time processing and recommendation generation.

2. Frontend Development:

Design a simple, intuitive user interface where users can input their skills and receive tailored job recommendations.

Utilize frameworks like React.js or Vue.js to create a responsive, user-friendly interface if the system is web-based; alternatively, use Flutter for a cross-platform mobile application.

3. Deployment:

Deploy the NLP model and the application on a cloud server (e.g., AWS, Google Cloud, or Azure) that meets the minimum hardware requirements, ensuring scalability and uptime.

Implement containerization using Docker to manage dependencies and simplify deployment, ensuring the application runs consistently across different environments.

4. Testing and Optimization:

Conduct testing for NLP accuracy, performance, and response time to ensure a smooth user experience.

Continuously update and optimize the keyword extraction and matching algorithms to enhance the relevance of job recommendations.

System Design

The system design for the job-matching application consists of multiple components that work together to provide users with personalized job recommendations. The primary components include the User Interface, Application Backend, NLP Processing Module, and Database. Each component plays a crucial role in ensuring the system's functionality, efficiency, and scalability.

1. Architecture Overview

The system follows a client-server architecture with a cloud-hosted backend. The main components and data flow are:

1. User Interface (Frontend): Collects user input (skills and abilities) and displays job recommendations.
2. Backend Server: Processes user input, communicates with the NLP model, and fetches results from the database.
3. NLP Processing Module: Extracts and interprets keywords from user input, matches them with job data.
4. Database: Stores job descriptions, keywords, and user profiles.

2. System Components

A. User Interface (UI)

Technology: React.js/Vue.js for web applications, or Flutter for mobile applications.

Functionality:

Collects user input for skills and abilities.

Displays personalized job recommendations in a user-friendly manner.

Allows users to filter or refine their recommendations.

The UI is responsible for providing a smooth and interactive user experience, with real-time responses based on user input.

B. Backend Server

Technology: Python (with Flask or Django for REST API)

Functionality:

Handles HTTP requests from the UI.

Connects to the NLP Processing Module for keyword extraction and job matching.

Queries the database to retrieve job roles and fields based on user input.

Sends job recommendations back to the UI.

The backend server acts as the central hub, orchestrating communication between the frontend, NLP module, and database.

C. NLP Processing Module

Technology: NLP libraries like spaCy or Hugging Face Transformers.

Functionality:

Analyzes user-entered skills and abilities.

Extracts relevant keywords and matches these keywords with stored job descriptions.

Ranks potential job matches based on keyword relevance and context.

The NLP module is the core component for understanding and interpreting user input, providing accurate matching and insights.

D. Database

Technology: PostgreSQL or MySQL

Structure:

Job Table: Stores job descriptions, required skills, job titles, and categories.

User Table: Records user profiles, skills entered, and recommended jobs for history tracking.

Keyword Table: Stores keywords and related job roles for quick lookup.

The database is optimized for fast read access to ensure real-time job recommendations.

3. Data Flow Diagram

The data flow follows this sequence:

1. User Input: The user enters skills and abilities on the frontend UI.
2. Request to Backend: The UI sends the data to the backend server via an HTTP request.

3. NLP Processing: The backend sends the data to the NLP Processing Module for keyword extraction.
4. Job Matching: The NLP module matches extracted keywords with job data stored in the database.
5. Database Query: The backend queries the database to retrieve job roles that match the extracted keywords.
6. Response to User: The backend compiles the results and sends job recommendations back to the UI for display.

4. System Design Diagram

Here's a suggested System Design Diagram structure:

1. Frontend (UI): Receives user input and displays job results.
2. API Gateway: Manages requests from the frontend and routes them to the backend server.
3. Backend Server: Manages business logic, processes requests, and interacts with the NLP module and database.
4. NLP Module: Processes skills input and matches them with job descriptions.
5. Database: Stores job data, keywords, and user information.

Experimental Results

1. Home Page / Skill Input Page

Description: This is the initial page where users start their journey by entering their skills, abilities, and any other relevant information.

Purpose: The home page serves as the primary data collection point for the system. Users input their skills, which are then processed by the NLP module.

Experimental Results:

Display of interactive fields for skill input.

Responsive design ensuring seamless data entry across devices.

Real-time feedback and suggestions as users type, if enabled (e.g., showing related skills or common keywords).

Smooth navigation from the home page to the results page after submission.

2. Job Recommendation Results Page

Description: After processing the user's skills, this page displays the recommended job roles and fields that best match the user's skill set.

Purpose: Provides the user with a personalized list of job recommendations based on the NLP model's analysis.

Experimental Results:

A ranked list of job roles with relevant keywords highlighted, showing the correlation between user skills and job requirements.

Display of additional information for each job, such as job title, industry, skill requirements, and relevance score.

User satisfaction measured through feedback features (e.g., thumbs up/down on job suggestions) to improve recommendation accuracy over time.

3. Job Details Page

Description: When a user clicks on a specific job recommendation, they are taken to the Job Details page, which provides more information about that particular role.

Purpose: Allows users to explore each job role in-depth, including an overview of job responsibilities, required skills, and qualifications.

Experimental Results:

Display of detailed job information that directly matches the user's skills, showcasing the accuracy of the NLP matching process.

Highlighting keywords from the user's input within the job description to visually represent the matching points.

Option to save or bookmark jobs for future reference.

4. User Profile Page

Description: This page allows users to view and manage their profiles, including the skills they've entered, saved jobs, and job recommendations history.

Purpose: Enables users to review their past inputs and access previously viewed or saved job recommendations.

Experimental Results:

Display of saved jobs and a history of matched roles, allowing users to see changes in recommendations over time.

Option to update skills, which allows for reprocessing and a refreshed set of recommendations based on the updated skill set.

Consistent retrieval and display of user-specific data from the database, demonstrating the application's capacity for storing and retrieving user data.

5. Feedback and Settings Page

Description: A page where users can provide feedback on the recommendations and adjust settings such as notification preferences or skill recommendation preferences.

Purpose: Collects user feedback for further improving recommendation accuracy and customization options.

Experimental Results:

Collection of user ratings for job recommendations (positive/negative feedback).

Settings adjustments that modify user experience and recommendation frequency.

Tracking and analysis of feedback to refine and enhance the NLP matching algorithm for future users.

Conclusion

The job-matching system developed in this project demonstrates a practical application of natural language processing (NLP) in connecting individuals with career opportunities tailored to their unique skill sets. By leveraging an NLP model, the system effectively extracts keywords from user-entered skills and compares them to the requirements of various job roles, offering relevant job recommendations. This approach minimizes the gap between job seekers and recruiters, ensuring a more efficient and personalized job search experience.

The experimental results confirm that the system is capable of accurately interpreting and matching user input with job data, thanks to its NLP-driven keyword extraction and matching algorithm. The user interface has been designed to be intuitive and responsive, providing a seamless experience across different devices, while the backend architecture ensures scalability and efficient processing.

In summary, the job-matching system offers a scalable, user-friendly solution that harnesses the power of NLP to enhance job search efficiency and relevance. Future improvements, such as integrating machine learning for continuous learning and user feedback analysis, could further increase the accuracy and personalization of job recommendations, making it a valuable tool for both job seekers and hiring professionals.

References

1. Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.

This book provides a comprehensive foundation for understanding the principles of natural language processing, including techniques used for keyword extraction and matching.

2. Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing (3rd ed.). Pearson.

This text covers a wide range of topics in NLP, including information retrieval and matching, which are fundamental for building a job-matching system.

3. Vaswani, A., Shazeer, N., & others. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems (NeurIPS).

This paper introduces the Transformer model, which is pivotal in modern NLP tasks and could be useful in enhancing the performance of the job-matching system.

4. Kowsari, K., et al. (2019). Text Classification Algorithms: A Survey. Information.

This survey discusses various algorithms for text classification, which can be applied in the context of job matching to categorize job descriptions effectively.

5. Dey, A. K., & Abowd, G. D. (2000). Towards a Better Understanding of Context and Context-Aware Computing.

This paper discusses the significance of context in computing, which is relevant for understanding user inputs and tailoring job recommendations.

6. Cohn, D. A., & Atlas, L. E. (2004). The Importance of Input: A Job Matching System for Online Communities. Proceedings of the 7th International Conference on Intelligent User Interfaces.

This conference paper outlines strategies for job matching systems, focusing on the importance of user input and matching algorithms.

7. Zhang, Y., & Wang, J. (2015). An Enhanced Job Recommender System Based on Collaborative Filtering and Text Mining. Proceedings of the 2015 IEEE International Conference on Data Mining.

This paper discusses an advanced approach to job recommendation systems, combining collaborative filtering with text mining techniques.

8. Sakakibara, M., & Ogawa, T. (2019). Job Matching with a Skills and Knowledge Database. International Journal of Business and Management.

This article explores job matching systems' development and challenges, particularly focusing on skills and knowledge databases.

ABBREVIATIONS

API - Application Programming Interface
CPU - Central Processing Unit
DBMS - Database Management System
GUI - Graphical User Interface
HTTP - HyperText Transfer Protocol
IDE - Integrated Development Environment
ML - Machine Learning
NLP - Natural Language Processing
RAM - Random Access Memory
SQL - Structured Query Language
UI - User Interface
UX - User Experience
AI - Artificial Intelligence
JSON - JavaScript Object Notation
REST - Representational State Transfer
MVC - Model View Controller
DB - Database
NLP - Natural Language Processing
SaaS - Software as a Service
XML - Extensible Markup Language

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman14, Bold;
Heading 2 – Times New Roman12, Bold;
Heading 3 – Times New Roman12, Bold;
Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
 - Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.