**Architecture Design**

**Buddy- Software Architecture Document**

**By-  
Shubhi Shrotriya**

**Contents**

Abstract 3

1. Introduction 4 .

1.1 What is Architecture Design? 4

1.2 Scope 4

1.3 Constraints 4

2. Technical Specification 5

2.1 Dataset 5

2.2 DataBase 7

2.3 Deployment 8

3. Technology Stack 8

4. Proposed Solution 9

5. Architecture Description 10

**Abstract**

The job search process can be daunting and time-consuming for both job seekers and recruiters. Buddy aims to address this issue by providing a comprehensive job search and recruitment platform. This app offers a user-friendly interface that allows job seekers to easily browse and search for job opportunities based on their preferences and qualifications, while recruiters can use it to post job openings, search for candidates, and communicate with applicants. Buddy's three main products are job recommendations, resume analyzer, and candidate recommendations. The job recommendation feature provides personalized job recommendations based on a user's qualifications and experience, while the resume analyzer feature offers career recommendations, courses recommendations, and skills recommendations based on a user's work experience and education. The candidate recommendation feature helps recruiters find the most qualified candidates for their job openings. In addition, the app also offers advanced technologies such as Optical Character Recognition (OCR) and Natural Language Processing (NLP) to process resumes and job postings to provide the best match for both job seekers and recruiters. The future scope of this project includes integration with social media, AI and machine learning, and expansion to new markets. Buddy is an excellent tool for both job seekers and recruiters, making the job search and hiring process more efficient and effective.

**1. Introduction**

**1.1 What is Architecture Design?**

The goal of Architecture Design (AD) or a low-level design document is to give the internal design of the actual program code for the Job Recommendation. AD describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

**1.2 Scope**

Architecture Design(AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

**1.3 Constraints**

1. For recommending the jobs to the candidate :-

We are currently exploring several job domains that have promising opportunities and potential for growth. These fields include:

1. Human Resources
2. Blockchain Developer
3. Cloud Engineer
4. Cyber Security
5. Data Analyst/Business Analyst
6. Data Scientist
7. Database Engineer
8. Frontend Developer
9. Machine Learning Engineer
10. Product Manager
11. Software Engineer

2. For resume analyzer we are recommending skills and courses belonging to following domains :-

1. Data Science
2. Web Development
3. Android Development
4. IOS Development
5. UI-UX Development

3. For Resume recommender , as we are having very limited number of resumes , and most of them are from data science domain , so by increasing number of resumes we can improve the resume recommendation

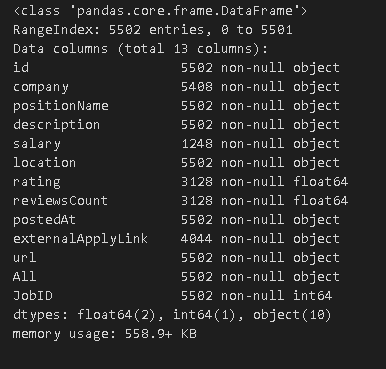
**2. Technical Specification**

**2.1 Dataset**

**For recommending the jobs to the candidate**

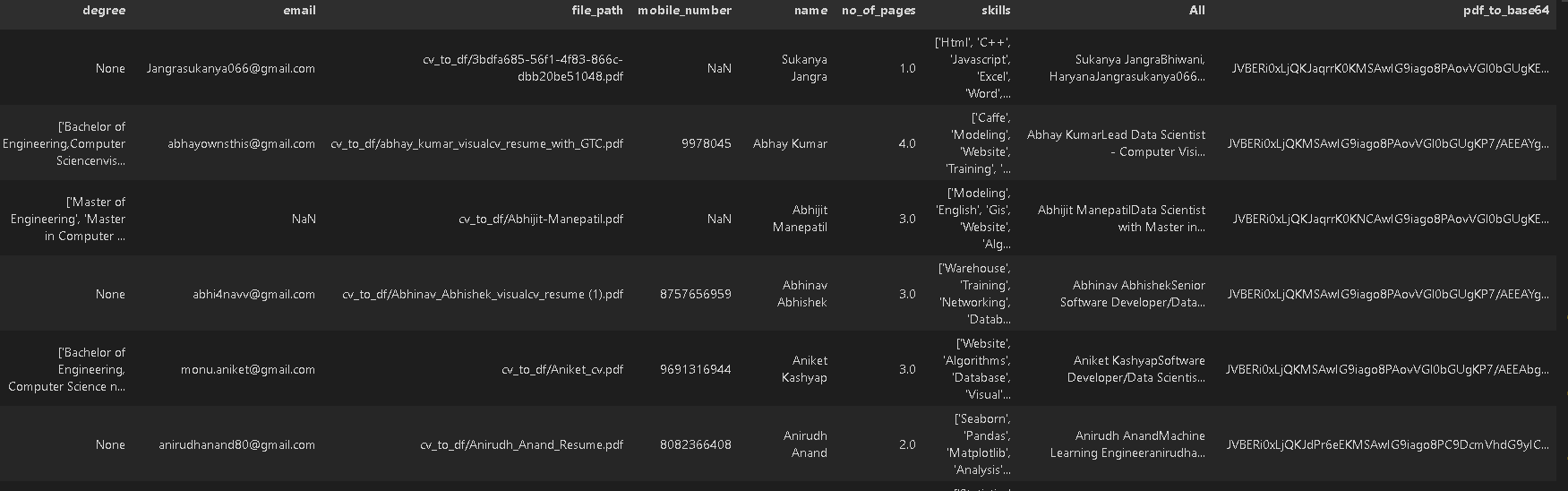
****

Due to web scraping restrictions imposed by job search engines such as Indeed, we sought out alternative means to collect pertinent job-related information. We turned to Apify, an advanced web scraping and automation tool, to gather the data we needed. With Apify's powerful capabilities, we were able to extract vast amounts of information in an efficient and accurate manner. This enabled us to provide up-to-date and precise job postings and candidate recommendations to both job seekers and recruiters, ensuring that they have access to the most relevant information available.



**For Resume recommender**

After extracting information from resume and converting the resume into base64 format , we have created this dataframe



**2.2 DataBase**

**MongoDB**

MongoDB is a cross-platform document-oriented database program that uses JSON-like documents with optional schemas. It is a NoSQL database that stores data in flexible, JSON-like documents, which makes it easier to store and process complex data types, such as nested arrays and objects. MongoDB is designed for scalability, high availability, and high performance.



**2.3 Deployment**

We used Streamlit, an open-source framework for creating data apps, to deploy our system. Streamlit was chosen for its ease of use, flexibility with different programming languages, and ability to handle various data visualizations. We set up a Streamlit account, developed the app code, tested it locally, and then deployed it for public access.



**3. Technology Stack**

|  |  |
| --- | --- |
| **Front End** | Python |
| **Backend** | Python |
| **Deployment** | Streamlit |

**4. Proposed Solution**

**JOB SEEKERS**

To assist job seekers, the process begins with uploading their CV to Buddy. The CV is then processed by Optical Character Recognition (OCR) technology and undergoes Natural Language Processing (NLP). The NLP process compares the uploaded CV with various job postings to determine the best match based on similarity. Finally, the system provides a list of recommended jobs that match the user's qualifications and experience.

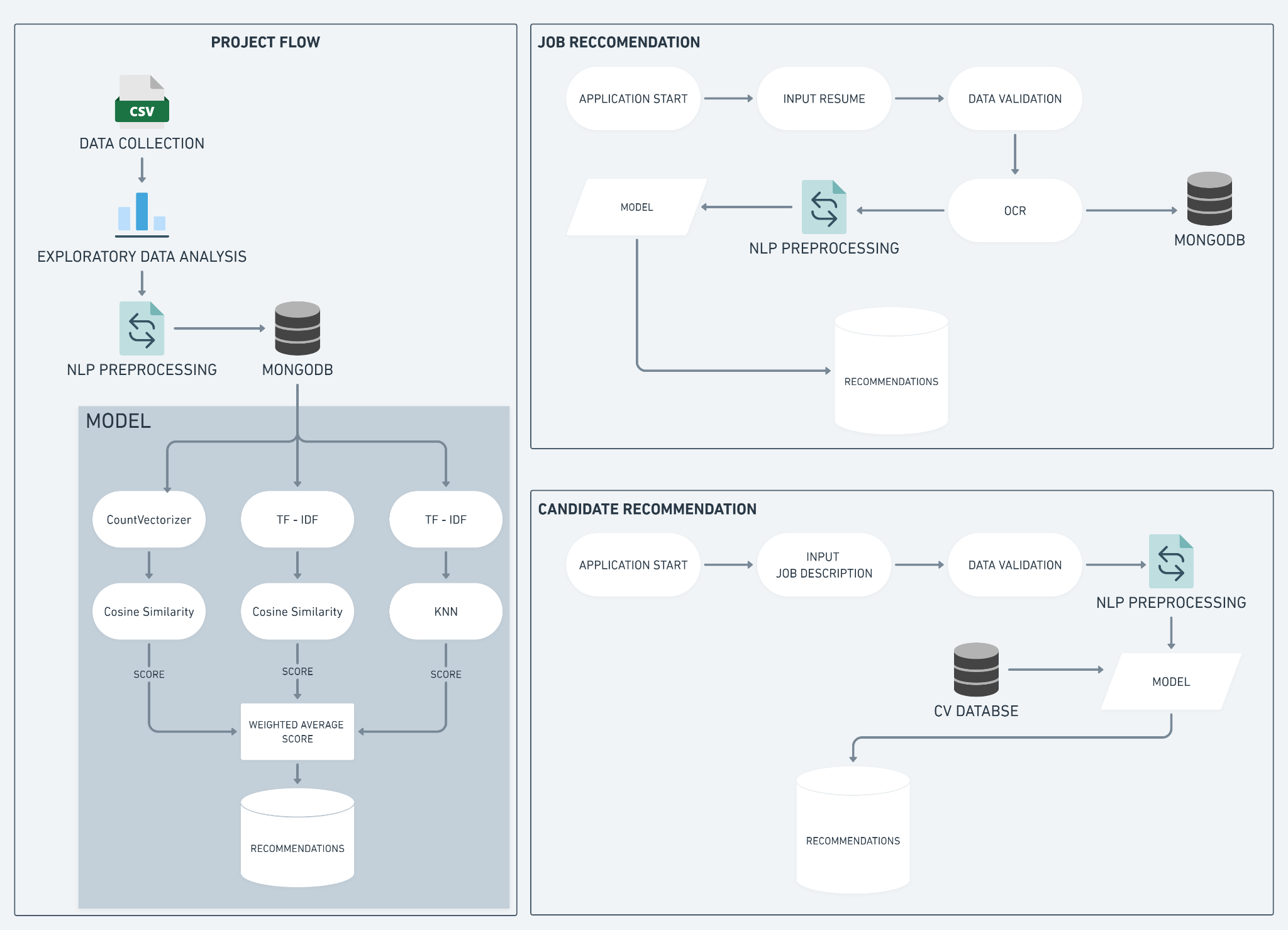
**RECRUITERS**

For recruiters, the process begins with inputting the job post into Buddy. The job post undergoes NLP analysis alongside the CVs in the database. Using various comparison methods, Buddy identifies the best-matching candidates for the job posting. The system then provides a list of recommended candidates to the recruiter based on their qualifications and experience.

**RESUME ANALYZER**

To extract user information from a resume, various features are analyzed. Firstly, the resume score is calculated based on the completeness of the information provided, the relevance to the job, and the overall layout. Career recommendations are then suggested based on the user's experience and education, which can help them target the right job opportunities. Additionally, resume writing tips are provided to improve the resume's overall effectiveness. Courses recommendations are also suggested based on the user's education and job preferences, which can help them acquire new skills and knowledge. Finally, skills recommendations are suggested based on the user's work experience and industry trends, which can help them stay competitive in the job market.

**5** **Architecture detail**

****

**5.1Data Gathering**

Data source for job details :https://console.apify.com/r/default-page

Train and Test data are stored in .csv format.

Data source for resume : random Internet search

**5.2 Data Pre-Processing**

Data preprocessing is an essential step in building a Machine Learning model and depending on how well the data has been preprocessed; the results are seen.

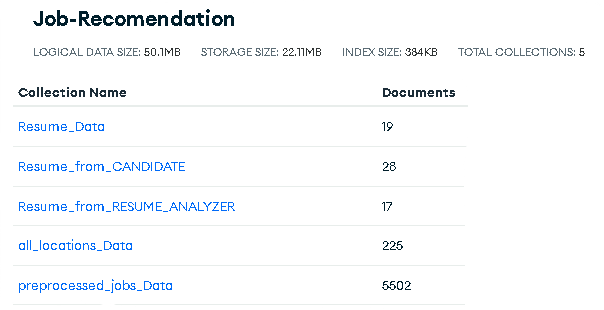
In NLP, text preprocessing is the first step in the process of building a model.

* Word tokenization:
* Stop words removal
* Lemmatization
* Bigram Collection Finder

**5.3 Data Insertion into Database**

We are inserting the data in MongoDB database job recommendation, and in that Database we are storing the the data in three distinct collections: Resume\_Data, all\_locations\_Data, and preprocessed\_jobs\_Data.

Where as there are two more collections inside the same database which are Resume\_from\_CANDIDATE and Resume\_from\_RESUME\_ANALYZER in order to store resumes entered by users on the job recommendation and resume analyzer pages. These collections will serve as a repository for user-provided resumes, allowing for more personalized and accurate job recommendations and analysis.



**5.4 Export Data from Database**

Data Export from Database - The data in a stored database is exported as a CSV file to be used for Model Training.

**5.5 Model Building**

|  |  |  |
| --- | --- | --- |
| **Model 1** | **Model 2** | **Model 3** |
| count vectorizer | TF - IDF | TF - IDF |
| cosine similarity | cosine similarity | KNN |
| SCORE - 1 | SCORE - 2 | SCORE - 3 |

We will scale and add all the three scores to make recommendations

**5.6 Data from User**

Here we will collecting the data from user :-

* **JOB SEEKERS** - The user inputs their resume, selects desired location and number of recommendations.
* **RECRUITERS** - The user inputs the Job Description and the number of candidates he wants to shortlist.
* **RESUME ANALYZER** - The user inputs their resume

**5.7 Data Validation**

To ensure the accuracy and validity of user-entered data, we check whether the input data is valid or not.

**5.8 Text Pre-processing**

We used OCR to extract text from the PDF, and then apply NLP preprocessing techniques such as stop word removal, tokenization, lemmatization, and bigram collection. These techniques help to clean and prepare the text data for further analysis.

**5.9 User Data Inserting into Database**

We insert user data and resume to augment our dataset, improving our model's performance. This approach expands our data, allowing for more accurate predictions.

**5.10 Model Call**

Based on the user input, our model will be loaded and will be used to predict/recommend.

**5.11 Job Recommendation**

The model generates Jobs/Output recommendations based on input data. Recommendations are presented in a list or ranking and may be refined based on user preferences or constraints. The process facilitates efficient decision-making.

**5.12 Deployment**

We used Streamlit, an open-source framework for creating data apps, to deploy our system. Streamlit was chosen for its ease of use, flexibility with different programming languages, and ability to handle various data visualizations. We set up a Streamlit account, developed the app code, tested it locally, and then deployed it for public access..

****