JOB PREDICTION ANDROID APPLICATION

A Mini - Project Report submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Sravya Soma	**121710306050**
Divyesh Perla	**121710306038**
Revanth Mogarampalli	**121710306033**
Saikowshik Ananthula	**121710306004**

Under the esteemed guidance of

SRINIVASA RAO GIDUTURI

Associate Professor, GIT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM

(Deemed to be

University)

VISAKHAPATNAM

December 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

We, hereby declare that the Project review entitled "**Job Prediction Android Application**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfilment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 20th January, 2021.

Registration	Name	Signature
121710306050	Sravya Soma	
121710306038	Divyesh Perla	
121710306033	Revanth Mogarampalli	
121710306004	Saikowshik Ananthula	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM (Deemed to be University)

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Job Prediction Android Application"** is a bonafide record of work carried out by **121710306004, 121710306033, 121710306038** and **121710306050** submitted in partial fulfilment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

PROJECT GUIDE	HEAD OF THE DEPARTMENT
Dr.Srinivasa Rao Giduturi	**Dr. Sireesha Rodda**
(Associate Professor)	**(Professor)**
CSE, GIT	**CSE, GIT**
GITAM	**GITAM**

ACKNOWLEDGEMENT

We would like to thank our project guide **Dr.Srinivasa Rao Giduturi**, Associate Professor, Department of CSE for her stimulating guidance and profuse assistance. We shall always cherish our association with her for her guidance, encouragement and valuable suggestions throughout the progress of this work. We consider it a great privilege to work under her guidance and constant support

We also express our thanks to the project's reviewers **Dr. Lakshmeeswari Gondi**, Associate Professor, and **Dr. Sita Mahalakshmi Tummala**, Professor, Department of CSE, GITAM (Deemed to be University) for their valuable suggestions and guidance for doing our project.

We consider it is a privilege to express our deepest gratitude to **Dr.R.Sireesha,** Head of the Department, Computer Science Engineering for his valuable suggestions and constant motivation that greatly helped us to successfully complete this project.

Our sincere thanks to **Dr. C. Dharma Raj**, Principal, GITAM Institute of Technology, GITAM (Deemed to be University) for inspiring us to learn new technologies and tools.

Finally, we deem it a great pleasure to thank one and all that helped us directly and indirectly throughout this project.

Sravya Soma	**121710306050**
Divyesh Perla	**121710306038**
Revanth Mogarampalli	**121710306033**
Saikowshik Ananthula	**121710306004**

TABLE OF CONTENTS

ABSTRACT

It is hard to identify a suitable job for a student or a job-seeker who requires jobs based on their skill set; it is also hard for employers to find a way to select candidates that best suits the job they require. In this project, we mainly focus on predicting the jobs utilizing deep neural networks, which includes CNN and machine learning algorithms like Logistic Regression with various word embeddings on IT datasets pre-trained. Also, we thought of a simple yet effective model by CNN. Our experimental results illustrated that our proposed CNN model achieved the highest result with an F1-score of 92.5%. Additionally, we scrutinize these results to discover better solutions in the future.

KEYWORDS USED

N-Grams: An N-gram is a contiguous sequence of n items from a given sample of text or speech. The items may contain list of phonemes, syllables, letters, words or base pairs based on the application.

Stemming: It is the way of minimizing inflection in words to their root forms such as mapping alike words to the same stem even if the stem itself is not a valid word in the Language.

Lemmatization: It is the way of grouping the different inflected forms of a word so they can be analyzed as a single item.

Logistic regression: This is a mathematical model that utilizes a logistic function to predict output based on a group of binary dependent variable.

Precision: In information retrieval, pattern recognition and classification in machine learning, precision is the fraction of relevant instances among the retrieved instances.

Recall: It is the fraction of the total amount of similar instances that were actually recalled. Precision and recall depend on an understanding and measure of similarity.

F-Score: It is simply a ratio of precision to the recall values of a model. It is one of the major evaluation metrics for a model

Confusion Matrix: It is also called as an error matrix. It is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.

CNN: A convolution neural network in short CNN is a type of artificial neural network used in image recognition and processing which is specifically designed to process pixel data.

Label binarizer: Fit label binarizer and transform multi-class labels to binary labels.

Tokenization: It is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences.

INTRODUCTION

Recently, the sudden development of Information Technology causes a plethora of job positions and particular IT job requirements. Many jobs emerged, which made it difficult for students and job seekers to identify a job that suits their knowledge and skills. In addition to this, recruitment companies have to manually filter the vast bundles of applicants' profiles to select the best-suited person for the job, causing a significant time waste if the applicants are enormous in numbers. So, we wanted to address and solve these problems with the knowledge we possess right now.

Our project is a classification system using various machine learning and deep learning, and Natural language processing techniques, which will work on descriptions of jobs that include skills, knowledge, and interests. We've fetched 25 types of jobs with their descriptions, consisting of 700 samples a type into a single dataset from a well-known website 'indeed.com' by web-scraping techniques. Later, we've analyzed the data and experimented with this data using various Machine learning techniques, which will be discussed further in this report. Finally, we've created a tflite model that will help the android application to predict the jobs.

Input: We will provide a small paragraph which is a description of skill set, knowledge, and interests.

Output: The application will provide a Job that best fits the provided description.

LITERATURE SURVEY

This project serves as a way to increase the existing model, which helps in predicting jobs based on their descriptions. A paper which was published by Vietnam National University students can yield only 72.40%. In the reported paper, they have used BiGRU LSTM with the Glove model to achieve this accuracy. The dataset which they have used in this model was collected by Dr.G. Papachristou includes 25 types of IT jobs. So, we've identified the problem. It lies in the dataset itself. The entire dataset has much redundancy. So, we have collected a whole new dataset with almost the same IT jobs.

Later on, we have created a simple logistic regression model to identify how far we came. Ironically, we got an accuracy of 83%, which gave us the confidence to go further deep. After experimenting the data with the CNN model, we achieved an accuracy of approximately 93%.

Finally, we have referenced a Hate Speech detection research paper, which helped us to get a glimpse of how lstm and other classifiers work. Also, it helped us to get a better understanding of how the Text CNN model works. This research presents a glimpse of the survey on hate speech detection. Provided the steadily growing body of social media content, the amount of online hate speech is also growing. Due to the huge scale of the web, ways that automatically identify hate speech are required. Their survey describes key areas that have been explored to automatically recognize these types of utterances using natural language processing.

We've referenced many YouTube videos to get an idea of the basic concepts of CNN and its advantages and disadvantages. Also, we've researched a lot of GitHub projects to get the ideology of collecting data through web scraping.

In the future, we will provide an android application platform and release this application in the play store. So that many students and employers will get benefited by our application. We will add error handling techniques such that they can handle real life usage scenarios.

PROBLEM IDENTIFICATION AND OBJECTIVES

In this ever-changing world, many opportunities emerge with the people's needs, but the problem here is it gives rise to several confusions. People like students and job seekers were unable to identify the job that best suits them. Many people spoil their lives by compromising to work in fields they are not interested in and do not fit well. Also, it is a hectic process for recruiters to sort out thousands of applicants to find a person that suits the vacancy well.

OUR SOLUTION TO ADDRESS THIS

Every student or applicant possesses some skill set and a curriculum vitae, are interested and comfortable in a particular domain. So, with this help of skill set and their C. V's description we will possess the required data to predict a class which suits that data. This process will help the people to achieve their dream jobs and helps recruiters in saving their time for selecting applicants by using a basic android application that will be readily available in the play store. We labialized each word in the description of the job and then we will identify the similarity of each class.

OBJECTIVES

- Discovery and Creation of Dataset
- Pre-processing of Fetched data
- Data Visualisation
- Identifying a correct algorithm for a better model metrics
- Optimizing the training speed and storage
- Developing a User Interface to interact with the user(Android Application).

Discovery and Creation of Dataset

The dataset here we require should contain description and skills that are required for a specific job. We have searched for the dataset with the above stated requirements in sites like kaggle. But they did not satisfy our needs. So, we decided to create our own dataset using web scraping by using a python module named "BeautifulSoup". We have created a python program named scraper and improved it upto three versions and finally collected the data with the job name as a label name and its description. All this data is stored into a ".csv" file using pandas' library. The scraper will go to Indeed.com website and fetch the list of required jobs from that website into a list. From this list the scraper again visits the individual website and fetches the job name and descriptions from that website. This process will continue till we get 700 samples of data for each type of class label. There will be 25 types of IT jobs we will be focusing on. So, 25*700=17500 data points will be collected. Also, the website data will be automatically encoded to UTF-8 format. This is done because website data when converted to plain text contains noisy data. So, to omit this we have used UTF-8 encoding.

Pre-Processing of Fetched Data

After web scraping, we've got ample amounts of data for training the model. But this data contains redundancy which will reduce the accuracy of the model. In doing so, we have researched many ways like stemming, lemmatization etc. Finally, we have decided to use lemmatization because of its useful features. All the punctuations marks are removed and every word is converted into its past form and capitalization was removed. We have created a list of stop words in which the lemmatizer checks and removes the redundancy. Each and every letter will be converted to lower case alphabets. We have replaced shorter forms with their longer forms like "isn't": "is not", "can't": "cannot ". To do so, we've imported a contractions list. Tf-Idf means term frequency-inverse document frequency. it computes the regularized frequency where every word count is divided by the total number of documents in which this word reappears.

Tf-idf (q, d) = Bow (q, d) * log (Total Number of Documents / (Number of documents in which word q appears))

If the word reappears regularly in a specific document, but appears less in other documents, it is most significant that the word describes a peculiar meaning for that document and will get awarded a larger frequency count because of its high Idf. Comparatively, if a word reappears in many documents, then its Idf is near to 1 and the logarithm updates 1 into 0 and lessens its result.

Data Visualization

In normal cases it is very easy to visualize the data. But in our project, we have text data, so methods are slightly different and the output yielded will be less compared to other visualizations. So, to overcome this we have chose to use word cloud data visualization technique to get a glimpse of the data. This will give an idea of how the frequency of each word is in the dataset.

Word cloud

In word clouds, each term size will get increased by its frequency of its repetition in the dataset. It is a square box figure in which words will be represented within the box randomly. Each word will be of different color and will be distinguished easily. The below provided is a word cloud of our project's dataset. It represents some of the important features that play major role in model training.

Logistic Regression Based Model

Introduction

Logistic Regression Based Model Introduction Logistic regression is a statistical model that uses a logistic function to portray a binary dependent variable, even though many more complex extensions are present. In regression analysis, logistic regression is approximating the variables of a logistic model. A binary logistic model has a relying variable with two possible values, such as pass/fail which is indicated by an indicator parameter, where the two parameters are labeled "0" and "1". In logistic model, the log-odds for the value labeled "1" is a linear combination of one or more independent variables; the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a sequential variable). The correlating probability of the value labeled "1" can lies between 0 and 1, hence the labeling; the function that changes from log-odds to probability is the logistic function, so the name. The unit of measurement to measure the log-odds scale is called a logit, from logistic units. Analogous models that possess a different sigmoid function that can be used instead of the logistic function, such as the probit model. It is the defining feature of the logistic model. It is the increasing one of the free variables multiplicatively laminates the odds of the given output at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

Usage

In this model, the dependent variable has two levels. Outputs which have greater than two values are shaped by multinomial logistic regression and, if the multiple groups are ordered, by ordinal logistic regression. The logistic regression model simply models probability of output into input and does not perform statistical classification, even though it can be utilized to make a classifier, by selecting a cutoff value and classifying inputs with probability higher than the cut-off as one of the class, lesser than the cutoff as the other; It is a general way to create a binary classifier. The coefficients are generally not calculated by a closed-form expression.

Implementation

At first, we will import libraries that are required for the model. Some of them are pandas, NumPy, sklearn etc.

```
In [1]: ##Libraries
        import pandas as pd
        import numpy as np
        from sklearn import preprocessing
        import matplotlib.pyplot as plt
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import train_test_split
        import os
```

After importing the libraries, we will load the data from the .csv file that we have collected and pre-processed for training.

```
In [2]: ##Load the file
        data = pd.read_csv("sample.csv", header = 0, names = ['Query', 'Description'],encoding="utf-8-sig")
```

Then we will split the dataset into 90% as training and 10% as test data. This is the initial step of the training model.

```
In [4]: ##Split the data into training and test sets
        X = data2.iloc[:,1:]
        y = data2.iloc[:,0]
        X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

Later on, we will create a classifier object using the Logistic Regression function that was imported from sklearn. we will fit this object on to the splitted data. With this step we have finished the training of the model.

```
In [6]: ##Fit logistic regression to the training set
        classifier = LogisticRegression(random_state=0)
        classifier.fit(X_train, y_train)
```

Now for predicting how far our training model is, we have used classifier.predict() function on the test data.

```
In [7]: ##Predicting the test set results and creating confusion matrix
        y_pred = classifier.predict(X_test)
        from sklearn.metrics import confusion_matrix
        confusion_matrix = confusion_matrix(y_test, y_pred)
        print(confusion_matrix)
```

```
[[158   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   1   0  23]
 [  0 136   0   0   0   0   0   0   0   0  15   0   0   0   0   0   0   0
    0   0   0   0   0   0  12]
 [  0   0 145   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0  29]
 [  0   0   0 130   2   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0  21]
 [  0   0   0   5 159   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0  18]
 [  0   0   1   0   0 141   0   0   0   0   0   0   0   0   2   0   0   0
    0   0   0   0   0   0  24]
 [  0   0   0   0   0   0 171   0   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0  28]
 [  1   0   0   0   0   0   0  58   0   0   0   0   0   0   0   0   0   0
    0   0   0   0   0   0 121]
 [  0   0   0   0   0   0   0   0 138   0   0   0   0   0   0   3   0   0
    0   0   0   1   0   0  29]
 [  0   0   0   0   0   0   0   0   0  41   0   0   0   0   0   0   0   0
```

Finally, we have printed a confusion matrix. This method gave us an utmost accuracy percentage of 74%. We have generated a classification report with the help of sklearn.metrics() to understand precision recall f1-score and support for each and every class label.

```
In [8]: ##Accuracy
        print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(classifier.score(X_test, y_test)))
```

```
Accuracy of logistic regression classifier on test set: 0.74
```

```
In [9]: ##Compute precision, recall, F-measure and support
        from sklearn.metrics import classification_report
        print(classification_report(y_test, y_pred))
```

```
                           precision    recall  f1-score   support

             .Net Developer       0.99      0.87      0.93       182
          Android developer       0.88      0.83      0.86       163
          Big Data Engineer       0.99      0.83      0.91       174
           Business Analyst       0.96      0.85      0.90       153
               Data Analyst       0.98      0.87      0.92       182
             Data Scientist       1.00      0.84      0.91       168
      Database Administrator       1.00      0.86      0.92       199
           DevOps Engineer       1.00      0.32      0.49       180
         Digital Marketing       1.00      0.81      0.89       171
         Financial Advisor       1.00      0.25      0.40       166
              IOS developer       0.91      0.83      0.87       206
                 IT Support       0.67      0.39      0.49       178
Information Security Analyst       1.00      0.91      0.95       188
            Java Developer       0.98      0.87      0.92       176
   Machine Learning Engineer       0.97      0.91      0.94       160
            Project Manager       0.96      0.91      0.93       173
               SAP Consultant       1.00      0.88      0.94       187
          Software Developer       0.87      0.88      0.88       166
          Software Engineer       0.95      0.78      0.86       170
           Software Tester       0.98      0.89      0.94       161
        Technical Support       0.46      0.13      0.20       183
              UI Designer       0.81      0.13      0.23       186
             Web Developer       0.94      0.80      0.86       162
          graphic designer       0.97      0.89      0.93       174
    human resource manager       0.15      1.00      0.26       167

                  accuracy                           0.74      4375
                 macro avg       0.90      0.74      0.77      4375
              weighted avg       0.90      0.74      0.77      4375
```

Convolution Neural Networks Based Model

Introduction

CNNs also called as convolution neural networks are a major class in deep learning era. They are majorly used in image classification problems. They have real time applications like image and video processing and recognition. In recent days are mainly used in medical fields to recognize tumors and infected cells. They are also used by voice assistants like “Ok Google” and “Alexa”.

CNNs are repetitive layers of neuron web in which each neuron is connected to successive neuron input. Because of this tightly packed neuron it may cause overfitting. So, to overcome this dropout regularization is used mostly.

CNNs evaluate problem in a different way, they make more sophisticated patterns into smaller layers with simpler patterns and evaluate each layer. Therefore, when coming to complexity and connectedness, CNNs are on lower side.

They were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex of the brain. Each individual cortical neuron responds to stimuli only in a specific restricted region of the visual field also called as receptive field. They partially overlap so they cover the whole visual field.

Implementation

At first, we will import libraries that are required for the model. Some of them are pandas, NumPy, sklearn, and keras etc.

In [1]:
```
from keras.models import Sequential
from keras.layers import Conv1D, GlobalMaxPooling1D, Embedding, LSTM
from keras.layers.core import Dense, Dropout, Activation
from keras.preprocessing.text import Tokenizer
from keras import metrics, regularizers
from keras.preprocessing import sequence
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
```

After importing the libraries, we will load the data from the .csv file that we have collected and pre-processed for training.

```
In [2]: ##Load the file
        data = pd.read_csv("sample.csv", header = 0, names = ['Query', 'Description'],encoding="utf-8-sig")
```

Then we will split the dataset into 90% as training and 10% as test data. This is the initial step of the training model. Also, here we have used. astype('str') function to encode integer values to strings.

```
In [3]: #Split the dataset to Training and Test subsets (90/10)
        train, test = train_test_split(data, test_size = 0.1, random_state = 17) #random_state = None

        train_descs = train['Description'].astype("str")
        train_labels = train['Query'].astype("str")

        test_descs = test['Description'].astype("str")
        test_labels = test['Query'].astype("str")
```

Model parameters like nb_epoch, batch_size, num_labels, kernel_size and hidden_dims etc which will be tuned according to the model. Here the number of epochs is 20 which represents the model will train for 20 times with the same data. Kernel size is 16. And max_features are 2000 which is the maximum a vocabulary set should contain. Batch size is chosen as 32.

```
In [4]: # Model Parameters
        vocab_size = 1000

        sequences_length = 1200

        embedding_dimensionality = 64
        max_features = 2000 #equal to vocab_size

        num_labels = len(train_labels.unique())
        batch_size = 32
        nb_epoch = 20

        nof_filters = 200
        kernel_size = 16

        hidden_dims = 512
```

Next step is to convert the sequences of text into numeric vector which will serve as an input to our model. A tokenizer is used to do this process. Later on, we have created a LabelBinarizer object to convert the text into vectors. Splitting of data is done here simultaneously.

```
In [5]: # Convert Texts to Numeric Vectors for Input
tokenizer = Tokenizer(num_words = vocab_size)
tokenizer.fit_on_texts(train_descs)

x_train = tokenizer.texts_to_sequences(train_descs)
x_test = tokenizer.texts_to_sequences(test_descs)

x_train = sequence.pad_sequences(x_train, maxlen = sequences_length, padding = 'post')
x_test = sequence.pad_sequences(x_test, maxlen = sequences_length, padding = 'post')

encoder = LabelBinarizer()
encoder.fit(train_labels)
y_train = encoder.transform(train_labels)
y_test = encoder.transform(test_labels)
```

We have selected our model as a sequential. To this we added a Conv1D layer with all the necessary hyper parameters. We have used a dropout layer which will reduce overfitting. Softmax and relu activation functions has yielded the best result. Softmax it's a function, not a loss. It squashes a vector in the range (0, 1) and all the resulting elements add up to 1. It is applied to the output scores ss. As elements represent a class, they can be interpreted as class probabilities.

The Softmax function cannot be applied independently to each sisi, since it depends on all elements of ss. For a given class sisi, the Softmax function can be computed as:

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

Where sjsj are the scores inferred by the net for each class in CC. Note that the Softmax activation for a class sisi depends on all the scores in ss.

Crossentropy will select the best output of two activation functions and adam optimizer reduced training time with best results.

In [6]:
```
model = Sequential()
model.add(Embedding(max_features, embedding_dimensionality, input_length = 1200))

model.add(Conv1D(nof_filters, kernel_size, padding='valid', activation='relu', strides = 1))
model.add(GlobalMaxPooling1D())

model.add(Dense(hidden_dims))
model.add(Dropout(0.3))
model.add(Activation('relu'))

model.add(Dense(num_labels))
model.add(Activation('softmax'))

model.summary()
model.compile(loss='categorical_crossentropy', optimizer='adam', #'sgd', 'adam', 'RMSprop', 'Adagrad'
                    metrics = [metrics.categorical_accuracy])
```

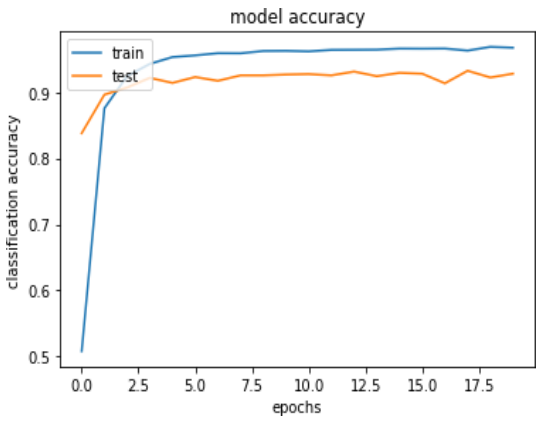
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 1200, 64)          128000
_________________________________________________________________
conv1d (Conv1D)              (None, 1185, 200)         205000
_________________________________________________________________
global_max_pooling1d (Global (None, 200)               0
_________________________________________________________________
dense (Dense)                (None, 512)               102912
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
activation (Activation)      (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 25)                12825
_________________________________________________________________
activation_1 (Activation)    (None, 25)                0
=================================================================
Total params: 448,737
Trainable params: 448,737
Non-trainable params: 0
```

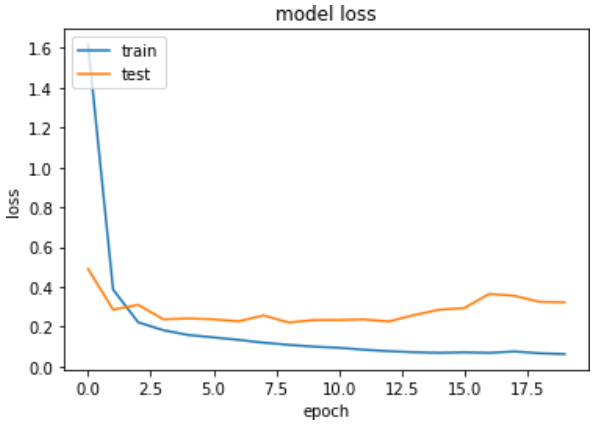
In the next step we will train the model with the divided data and train it for up to 20 epochs.

In [7]:
```
history = model.fit(x_train, y_train,
                    batch_size = batch_size,
                    epochs = nb_epoch,
                    verbose = True,
                    validation_split = 0.1)
```

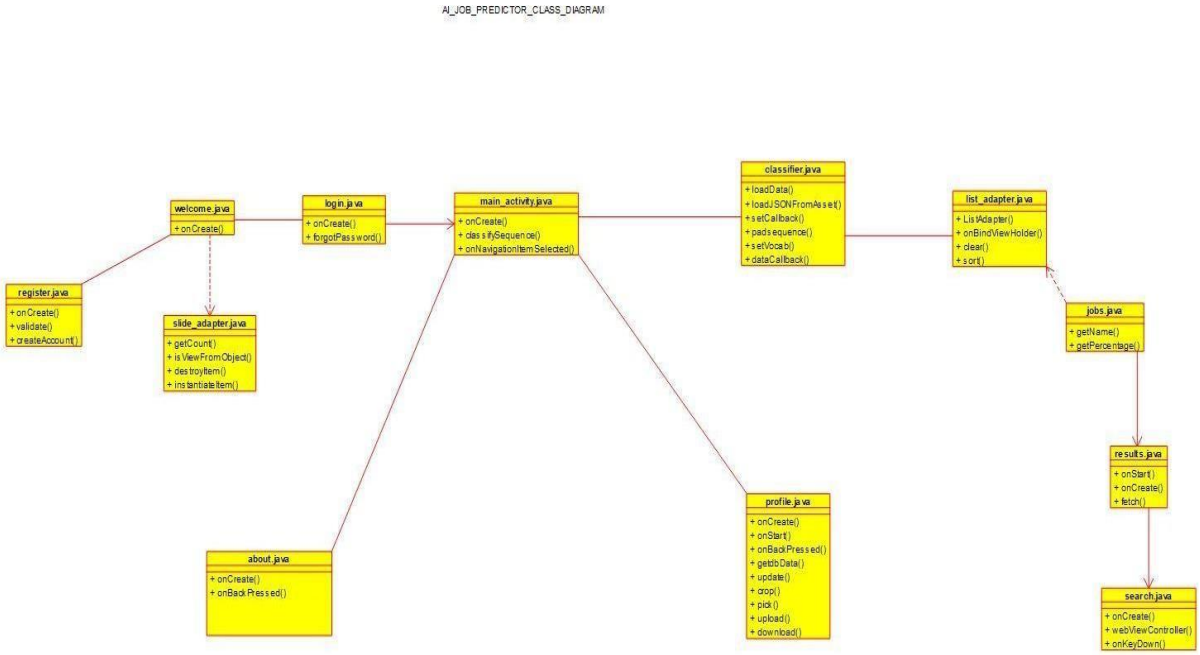
Our model has yielded 92.40% accuracy with the model accuracy and loss graphs provided below.

```
55/55 [==============================] - 2s 43ms/step - loss: 0.3492 - categorical_accuracy: 0.9240

Test categorical_crossentropy: 0.3491825759410858
Categorical accuracy: 0.9240000247955322
```


model accuracy
train
test
classification accuracy
epochs
0.5
0.6
0.7
0.8
0.9
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


model loss
train
test
loss
epoch
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

UML DIAGRAM


AI_JOB_PREDICTOR_CLASS_DIAGRAM
welcome.java
+ onCreate()
login.java
+ onCreate()
+ forgotPassword()
main_activity.java
+ onCreate()
+ classifySequence()
+ onNavigationItemSelected()
classifier.java
+ loadData()
+ loadJSONFromAsset()
+ setCallback()
+ padsequence()
+ setVocab()
+ dataCallback()
list_adapter.java
+ ListAdapter()
+ onBindViewHolder()
+ clear()
+ sort()
register.java
+ onCreate()
+ validate()
+ createAccount()
slide_adapter.java
+ getCount()
+ isViewFromObject()
+ destroyItem()
+ instantiateItem()
jobs.java
+ getName()
+ getPercentage()
results.java
+ onStart()
+ onCreate()
+ fetch()
about.java
+ onCreate()
+ onBackPressed()
profile.java
+ onCreate()
+ onStart()
+ onBackPressed()
+ getInfoData()
+ update()
+ crop()
+ pick()
+ upload()
+ download()
search.java
+ onCreate()
+ webViewController()
+ onKeyDown()

GUI Implementation

About.java

In this program we have used some packages which are used to build android applications and we had extended a class appcompatactivity from about class. We had kept image viewer ,image button to view content of facebook,instagram and also LinkedIn on clicking on it we can view the information related to it

.

```java
package com.example.ai_job_predictor;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.media.Image;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;

public class About extends AppCompatActivity {
ImageButton back;
ImageView facebook,insta,linkedin;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
        back=findViewById(R.id.back2);
        facebook=findViewById(R.id.imageView16);
        insta=findViewById(R.id.imageView14);
        linkedin=findViewById(R.id.imageView15);
        back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(About.this,MainActivity.class));
                overridePendingTransition(R.anim.push_right_in,R.anim.push_right_out);
                finish();
            }
```

About.xml

We have the version 1.0 .In this code is about the about page we had kept view, text View, Image Button, Scroll View, Linear Layout, to set heights, widths background, elevation we have used so constraints

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".About">

    <View
        android:id="@+id/view4"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/accent"
        android:elevation="5dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView14"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:elevation="10dp"
        android:text=" About Us"
        android:textAlignment="center"
        android:textColor="#B72ED6"
        android:textSize="36sp"
        app:layout_constraintBottom_toBottomOf="@+id/view4"
        app:layout_constraintEnd_toEndOf="@+id/view4"
        app:layout_constraintHorizontal_bias="0.269"
        app:layout_constraintStart_toStartOf="@+id/view4"
        app:layout_constraintTop_toTopOf="@+id/view4"
```

```
<ImageButton
    android:id="@+id/back2"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:backgroundTint="@android:color/transparent"
    android:elevation="5dp"
    android:tint="@color/black"
    app:layout_constraintBottom_toBottomOf="@+id/view4"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/abc_vector_test" />

<ImageView
    android:id="@+id/imageView14"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:background="@color/accent"
    android:elevation="5dp"
    app:layout_constraintEnd_toEndOf="@+id/view4"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/insta" />

<ImageView
    android:id="@+id/imageView15"
    android:layout_width="48dp"
    android:layout_height="48dp"
    android:layout_marginBottom="8dp"
    android:background="@android:color/transparent"
    android:elevation="5dp"
    android:paddingLeft="2dp"
    android:paddingRight="2dp"
    app:layout_constraintBottom_toTopOf="@+id/scrollView3"
```

```
<ScrollView
    android:id="@+id/scrollView3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="?attr/actionBarSize"
    android:alpha="1.0"
    android:background="@android:color/transparent"
    android:backgroundTintMode="multiply"
    android:clipToPadding="true"
    android:fillViewport="true"
    app:layout_constraintTop_toTopOf="parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/textView13"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:alpha="1.0"
            android:padding="5dp"
            android:text="Recently, the sudden development of Information Technology causes a plethora of job positions and particular IT job requirements. Many jobs emerg
            android:textAlignment="center"
            android:textAllCaps="false"
            android:textColor="#000000"
            android:textSize="20dp" />
    </LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Classifier.java

To load the CVV model and to give the input to the model and extract the output form the model.

The output is then tokenized, padded and sent to the model.

```
1  package com.example.ai_job_predictor;
2
3  import android.content.Context;
4  import android.os.AsyncTask;
5  import android.util.Log;
6  import android.widget.Toast;
7
8  import androidx.recyclerview.widget.AsyncListUtil;
9
10 import org.json.JSONException;
11 import org.json.JSONObject;
12
13 import java.io.IOException;
14 import java.io.InputStream;
15 import java.lang.reflect.Array;
16 import java.util.ArrayList;
17 import java.util.Arrays;
18 import java.util.Collection;
19 import java.util.Collections;
20 import java.util.HashMap;
21 import java.util.Iterator;
22 import java.util.List;
23
24 public class Classifier {
25     private Context context;
26     private String filename;
27     private static DataCallback callback;
28     private int maxlen;
29     private HashMap<String, Integer> vocabData;
30
31     public Classifier(Context context, String Jfilename) {
32         this.context = context;
33         this.filename = Jfilename;
34     }
35
36     public void loadData() {
37         LoadVocabularyTask loadVocabulary = new LoadVocabularyTask(callba
38         loadVocabulary.execute(loadJSONFromAsset(filename));
39     }
40
41     private String loadJSONFromAsset(String filename) {
42         String json;
```

```
        try {
            InputStream inputStream = context.getAssets().open(filename);
            int size = inputStream.available();
            byte[] buffer = new byte[size];
            inputStream.read(buffer);
            inputStream.close();
            json = new String(buffer);
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
        return json;
    }

    public void setCallback(DataCallback callback) {
        this.callback = callback;
    }

    ArrayList<Integer> tokenize(String message) {
        String[] parts = message.split(" ");
        ArrayList<Integer> tokenizedMessage = new ArrayList();
        int index = 0;
        for (String part : parts) {
            if (part.trim() != "") {
                if (vocabData.get(part) == null) {
                    index = 0;
                } else {
                  index = vocabData.get(part);
                }
            }
            tokenizedMessage.add(index);


        }
        //Toast.makeText(context,"",Toast.LENGTH_LONG).show();
        return tokenizedMessage;
    }
    public ArrayList<Integer> padSequence(List<Integer>sequence){
        sequence.removeAll(Collections.singleton(0));
        if (sequence.size()>maxlen){
            ArrayList<Integer> array=new ArrayList<>(sequence.subList(0,maxlen));
            //Toast.makeText(context,"if"+array.size(),Toast.LENGTH_LONG).show();
            return array;
```

```
        }
        else if (sequence.size()<maxlen){
            for (int i=sequence.size();i<maxlen;i++){
                sequence.add(0);
            }
           // Toast.makeText(context,"else"+sequence.size(),Toast.LENGTH_LONG).show();
            return (ArrayList<Integer>) sequence;
        }
        else{return (ArrayList<Integer>) sequence;}


    }
    public void setVocab(HashMap<String,Integer>data){
        this.vocabData=data;
    }
    public void setMaxlength(int maxlen){this.maxlen=maxlen;}


    interface DataCallback{
        void onDataProcessed(HashMap<String, Integer> result); }
    private class LoadVocabularyTask extends AsyncTask<String, Void, HashMap<String, Integer>> {
        private DataCallback callback;
        public LoadVocabularyTask(DataCallback callback) {
            this.callback= (DataCallback) callback;
        }

        @Override
        protected HashMap<String, Integer> doInBackground(String... strings) {
            JSONObject jsonObject= null;
            try {
                jsonObject = new JSONObject(strings[0]);
            } catch (JSONException e) {
                e.printStackTrace();
            }
            Iterator<String> iterator=jsonObject.keys();
            HashMap<String,Integer>data=new HashMap<>();
            while(iterator.hasNext()){
                String key = iterator.next();
                try {
                    data.put(key,(Integer)jsonObject.get(key));
                } catch (JSONException e) {
                    e.printStackTrace();
                }
```

```
                try {
                    data.put(key,(Integer)jsonObject.get(key));
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
            return data;
        }
        @Override
        protected void onPostExecute(HashMap<String,Integer>result){
            super.onPostExecute(result);
            callback.onDataProcessed(result);
        }
    }

}
```

Jobs.java

This is to show the percentage gained by the users resume for the respective jobs. Based on the percentage level the user can apply to the jobs which have a higher percentage.

```java
package com.example.ai_job_predictor;

public class Jobs {
    String name;
    float percentage;
    public Jobs(String name,float percentage){
        this.name=name;
        this.percentage=percentage;
    }
    public String getName() {
        return name;
    }
    public float getPercentage() {
        return percentage;
    }
}
```

ListAdapter.java

After attaining the output to hold the list of jobs the listAdapter is used. The recycler view is controlled by the List Adapter and showing the jobs and addition of the jobs or removal and their descriptions is controlled using the ListAdapter

```java
package com.example.ai_job_predictor;

import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.net.Uri;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.common.collect.ImmutableList;

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Random;

public class ListAdapter extends RecyclerView.Adapter{
    private Context mcontext;
    private List<Jobs> JobList;
    TextView title,percentage;
    ImageButton button;
    String[] colors={"#FFCB7D","#DBA9A9","#C1E950","#CFE2CF","#B3EBCC","#9BF8FF","#82BEF3","#AFAFCF","#C1A3FF","#FFE2FF","#EB8484","#20d2bb"};
    Random generator = new Random();
```

```
43  public ListAdapter(Context context,List<Jobs> jobList){
44      JobList=jobList;
45      mcontext=context;
46  }
47      @Override
48      public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
49          View view;
50          view = LayoutInflater.from(parent.getContext())
51                  .inflate(R.layout.jobs_layout, parent, false);
52          return new jobHolder(view);
53      }
54      @Override
55      public int getItemViewType(int position) {
56          Jobs job = JobList.get(position);
57
58          return position;
59      }
60      @Override
61      public int getItemCount() {
62          return JobList.size();
63      }
64
65      @Override
66      public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
67      final Jobs details=JobList.get(position);
68          ((jobHolder)holder).bind(details);
69          button=holder.itemView.findViewById(R.id.imageButton);
70          if(position!=0) { button.setVisibility(View.INVISIBLE); }
71          int randomIndex = generator.nextInt(colors.length);
72          holder.itemView.findViewById(R.id.cardView).setBackgroundColor(Color.parseColor(colors[randomIndex]));
73          button.setOnClickListener(new View.OnClickListener() {
74              @Override
75              public void onClick(View v) {
76                  String job = details.getName();
77                  mcontext.startActivity(new Intent(mcontext, Search.class).putExtra("job", job));
78              }
79          });
80      }
81      public class jobHolder extends RecyclerView.ViewHolder {
82          TextView title;
83          public jobHolder(View itemView) {
84              super(itemView);
```

```
            title = itemView.findViewById(R.id.tv10);
            percentage=itemView.findViewById(R.id.textView11);
        }

        void bind(Jobs message) {
            title.setText(message.getName());
            percentage.setText(String.valueOf(message.getPercentage()));
        }
    }

    public void clear() {
        JobList.clear();
    }
    public void msort(){
        Collections.sort(JobList, new Comparator<Jobs>() {
            @Override
            public int compare(Jobs o1, Jobs o2) {
                return new Float(o1.getPercentage()).compareTo(new Float(o2.getPercentage()));

            }
        }.reversed());

    }

}
```

Login.java

Using the firebase authentication class we created the login page which stores the users information in a database format.

```java
package com.example.ai_job_predictor;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class Login extends AppCompatActivity {
EditText email,password;
Button login;
FirebaseAuth fa;
FirebaseUser user;
TextView forgot,register;
ProgressBar pb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        email=findViewById(R.id.editText);
        password=findViewById(R.id.editText2);
        login=findViewById(R.id.button);
        fa=FirebaseAuth.getInstance();
        forgot=findViewById(R.id.reset);
        register=findViewById(R.id.link_signup);
        pb=findViewById(R.id.progressBar);
        user=FirebaseAuth.getInstance().getCurrentUser();
        if(user!=null){
            startActivity(new Intent(Login.this,MainActivity.class));
```

```
            finish();
        }
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                pb.setVisibility(View.VISIBLE);
                String email_text=email.getText().toString();
                String password_text=password.getText().toString();
                if(email_text.isEmpty() || password_text.isEmpty()){
                    Toast.makeText(Login.this,"Please check your Email and Password",Toast.LENGTH_LONG).show();
                    if(email_text.isEmpty()){email.requestFocus(); email.setError("Email Invalid  or Empty");}
                    else if(password_text.isEmpty()){password.requestFocus();password.setError("Password Empty or Invalid");}
                    login.setBackgroundResource(R.drawable.rounded_orange);
                    pb.setVisibility(View.INVISIBLE);

                }
                else if(!email_text.matches("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@" + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$")){
                    Toast.makeText(Login.this,"Invalid Email",Toast.LENGTH_LONG).show();
                    email.setError("Invalid Email");
                    email.requestFocus();
                    pb.setVisibility(View.INVISIBLE);
                }
                else{
                    login.setBackgroundResource(R.drawable.rounded);
                    fa.signInWithEmailAndPassword(email_text,password_text).addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            Toast.makeText(Login.this,"Login failed due to:"+e.getMessage(),Toast.LENGTH_LONG).show();
                        }
                    }).addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                        @Override
                        public void onSuccess(AuthResult authResult) {
                            startActivity(new Intent(Login.this,MainActivity.class));
                            overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
                            finish();
                        }
                    });
                    pb.setVisibility(View.INVISIBLE);
                }


            }
        });
```

```
        forgot.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
            forgotpassword();
            }
        });
        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(Login.this,Register.class));
                overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
                finish();
            }
        });
    }
    public void forgotpassword(){
        pb.setVisibility(View.VISIBLE);
        if (TextUtils.isEmpty(email.getText().toString())) {
            email.setError("Invalid Email");
            email.requestFocus();
            Toast.makeText(Login.this,"Enter a valid email to continue",Toast.LENGTH_LONG).show();
            pb.setVisibility(View.INVISIBLE);
        }
        else{
            pb.setVisibility(View.VISIBLE);
            fa.sendPasswordResetEmail(email.getText().toString().trim()).addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if(task.isSuccessful()){
                        Toast.makeText(Login.this,"Check your email to reset your account.",Toast.LENGTH_LONG).show();
                        //pb.setVisibility(View.INVISIBLE);

                    }
                    else{
                        Toast.makeText(Login.this,"Cannot generate reset link."+task.getException(),Toast.LENGTH_LONG).show();
                        //pb.setVisibility(View.INVISIBLE);
                    }

                }
            });
            pb.setVisibility(View.INVISIBLE);
        }
    }
}
```

Login.xml

This code is about the login page. To enter login details like username ,password we had kept a linear layout, text view, edit text, and included buttons and progress bar to logging into the account.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/expresso"
    android:fitsSystemWindows="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="24dp"
        android:paddingTop="56dp"
        android:paddingRight="24dp">


        <!-- Email Label -->

        <!-- Password Label -->

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Email"
            android:textColor="@color/accent" />

        <EditText
            android:id="@+id/editText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Password"
        android:textColor="@color/accent" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword"
        android:textColor="@color/accent" />

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:background="@drawable/profile_back_button"
        android:text="Login"
        android:textColor="@color/accent" />

    <TextView
        android:id="@+id/link_signup"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:layout_marginBottom="24dp"
        android:gravity="center"
        android:text="No account yet? Create one"
        android:textColor="@color/accent"
        android:textSize="16dip" />
```

```
        android:layout_marginTop="30dp"
        android:layout_marginBottom="24dp"
        android:gravity="center"
        android:text="No account yet? Create one"
        android:textColor="@color/accent"
        android:textSize="16dip" />

    <TextView
        android:id="@+id/reset"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Forgot password?"
        android:textColor="@color/accent"
        android:textSize="16sp" />

    <ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:visibility="invisible" />

</LinearLayout>
:rollView>
```

MainActivity.java

The major feature and the whole game changer for the app is the predicting the job by the information provided by the user in the resume is done in this section. This landing page once the user logins to the app. It controls all the java files and sends the intents to all the files

```java
package com.example.ai_job_predictor;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.content.res.AssetFileDescriptor;
import android.database.Cursor;
import android.graphics.pdf.PdfDocument;
import android.net.Uri;
import android.os.Bundle;
import android.os.Parcelable;
import android.provider.OpenableColumns;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.ml.common.FirebaseMLException;
import com.google.firebase.ml.common.modeldownload.FirebaseModelManager;
import com.google.firebase.ml.custom.FirebaseCustomRemoteModel;
import com.google.firebase.ml.custom.FirebaseModelInterpreter;
import com.google.firebase.ml.custom.FirebaseModelInterpreterOptions;

import org.tensorflow.lite.Interpreter;
import org.w3c.dom.Document;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.FloatBuffer;
```

```
import java.nio.MappedByteBuffer;
import java.nio.channels.FileChannel;
import java.nio.charset.Charset;
import java.sql.Array;
import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;

public class MainActivity extends AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener {
    private static final int PIC_CROP =1 ;
    EditText et;
    Button b,c;
    ProgressBar pb;
    FirebaseAuth fa;
    private NavigationView navigationView;
    Toolbar toolbar;
    DrawerLayout drawerLayout;
    boolean stop =true;
    Interpreter interpreter;
    @Override
    protected void onStart(){
        super.onStart();
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        drawerLayout = findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle actionBarDrawerToggle = new ActionBarDrawerToggle(MainActivity.this, drawerLayout, toolbar, R.string.navigation_drawer_open, R.string.navigation_
        actionBarDrawerToggle.syncState();
        drawerLayout.addDrawerListener(actionBarDrawerToggle);
        navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        et = findViewById(R.id.editTextTextMultiLine);
        Classifier classifier = new Classifier(this, "dict.json");
        classifier.setMaxlength(2000);
        b = (Button)findViewById(R.id.button);
        c = (Button)findViewById(R.id.button2);
        pb=findViewById(R.id.progressBar);
        fa=FirebaseAuth.getInstance();
        if(fa.getCurrentUser()==null) {startActivity(new Intent(MainActivity.this,Welcome.class));finish();}
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                b.setBackgroundDrawable(getResources().getDrawable(R.drawable.rounded_orange));
```

```
                pb.setVisibility(View.VISIBLE);
                final String message = et.getText().toString().trim().toLowerCase();
                // Toast.makeText(getBaseContext(),"message:"+message,Toast.LENGTH_LONG).show();
                if (message.length()<150)
                {Toast.makeText(getBaseContext(), "Please enter text length of >150",Toast.LENGTH_LONG).show(); pb.setVisibility(View.INVISIBLE);}
                else{
                    classifier.setCallback(new Classifier.DataCallback() {
                        @Override
                        public void onDataProcessed(HashMap<String, Integer> result) {
                            try {
                                classifier.setVocab(result);
                                ArrayList<Integer> tokenizedMessage = classifier.tokenize(message);
                                ArrayList<Integer> paddedMessage = classifier.padSequence(tokenizedMessage);
                                classifySequence(paddedMessage);
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                        }
                    });
                    classifier.loadData();

                }
            }
        });
        c.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                et.setText("");
                b.setBackgroundDrawable(getResources().getDrawable(R.drawable.rounded));
            }
        });

    }
    public void classifySequence(List<Integer>sequence) throws IOException {
        final String s ="";
        String p="";
        ByteBuffer input=ByteBuffer.allocateDirect(2000*4).order(ByteOrder.nativeOrder());
       float[][] outputs=new float[1][25];
        for (float value: sequence) { input.putFloat(value); }
       FirebaseCustomRemoteModel remoteModel = new FirebaseCustomRemoteModel.Builder("converted_model.tflite").build();
     FirebaseModelManager.getInstance().getLatestModelFile(remoteModel)
                .addOnCompleteListener(new OnCompleteListener<File>() {
                    @Override
                    public void onComplete(@NonNull Task<File> task) {
                        float[][] output = new float[1][25];
                        File modelFile = task.getResult();
                        if (modelFile != null) {
                            interpreter = new Interpreter(modelFile);
                        } else {
```

```
                            try {
                                InputStream inputStream = getAssets().open("converted_model.tflite");
                                byte[] model = new byte[inputStream.available()];
                                inputStream.read(model);
                                ByteBuffer buffer = ByteBuffer.allocateDirect(model.length)
                                        .order(ByteOrder.nativeOrder());
                                buffer.put(model);
                                interpreter = new Interpreter(buffer);
                                Log.d("loaded", "Loaded");
                            } catch (IOException e) {
                                // File not found?
                            }
                        }
                        interpreter.run(input, output);
                        startActivity(new Intent(MainActivity.this,Results.class).putExtra("list",output[0]));
                       pb.setVisibility(View.INVISIBLE);

                    }
                });

    }

    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {
        switch (menuItem.getItemId()) {
            case R.id.nav_profile:
                startActivity(new Intent(MainActivity.this,Profile.class));
                overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
                finish();
                break;
            case R.id.about_app:
                startActivity(new Intent(MainActivity.this, About.class));
                overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
                finish();
                break;

            case R.id.logout_user:
                fa.signOut();
                startActivity(new Intent(this,Welcome.class));
                overridePendingTransition(R.anim.push_right_in,R.anim.push_right_out);
                finish();
                break;

            case R.id.refresh:
                if (stop) {
                    pb.setVisibility(View.VISIBLE);
                    finish();
                    startActivity(getIntent());
                    getSupportActionBar().setTitle("Good to go..");
```

```
                stop = true;
                pb.setVisibility(View.INVISIBLE);
            } else {
                Toast.makeText(MainActivity.this, "No refresh required", Toast.LENGTH_LONG).show();

            }
            break;
        case R.id.nav_share:
            Intent shareIntent =   new Intent(android.content.Intent.ACTION_SEND);
            shareIntent.setType("text/plain");
            shareIntent.putExtra(Intent.EXTRA_SUBJECT,"Now get blood donations with ease at your location");
            String app_url = "https://drive.google.com/drive/folders/1FlfmK-cDQMoGbUzicx48tYpwHR6fanr1?usp=sharing";
            shareIntent.putExtra(android.content.Intent.EXTRA_TEXT,app_url);
            startActivity(Intent.createChooser(shareIntent, "Share via"));
            break;
        case R.id.rate:
            Uri uri = Uri.parse("market://details?id=" + this.getPackageName());
            Intent goToMarket = new Intent(Intent.ACTION_VIEW, uri);
            goToMarket.addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY |
                    Intent.FLAG_ACTIVITY_NEW_DOCUMENT |
                    Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
            try {
                startActivity(goToMarket);
            } catch (ActivityNotFoundException e) {
                startActivity(new Intent(Intent.ACTION_VIEW,
                        Uri.parse("http://play.google.com/store/apps/details?id=" + this.getPackageName())));
            }
            break;
    }

    // menuItem.setChecked(true);
    drawerLayout.closeDrawer(GravityCompat.START);
    return true;

}
```

Main.xml

In this code we are explaining the main function. We had kept a linear layout, text view, edit text, and included buttons and progress bar. To enter login password labels and to set height, width settings.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/expresso"
    android:fitsSystemWindows="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="24dp"
        android:paddingTop="56dp"
        android:paddingRight="24dp">


        <!-- Email Label -->

        <!-- Password Label -->

        <TextView
            android:id="@+id/textView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Email"
            android:textColor="@color/accent" />

        <EditText
            android:id="@+id/editText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
```

```
            android:text="Login"
            android:textColor="@color/accent" />

        <TextView
            android:id="@+id/link_signup"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="30dp"
            android:layout_marginBottom="24dp"
            android:gravity="center"
            android:text="No account yet? Create one"
            android:textColor="@color/accent"
            android:textSize="16dip" />

        <TextView
            android:id="@+id/reset"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Forgot password?"
            android:textColor="@color/accent"
            android:textSize="16sp" />

        <ProgressBar
            android:id="@+id/progressBar"
            style="?android:attr/progressBarStyle"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:visibility="invisible" />

    </LinearLayout>
/ScrollView>
```

Profile.java

To manage the user details stored in the firebase like name, age, phone, email id , profile picture these all can be changed according to the user's will.

```java
package com.example.ai_job_predictor;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.provider.OpenableColumns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.Toast;

import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.firestore.DocumentChange;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.EventListener;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.FirebaseFirestoreException;
import com.google.firebase.firestore.QuerySnapshot;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.StorageTask;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.util.Calendar;
import java.util.Date;

import static android.view.View.VISIBLE;

```

```
            @Override
            public void onClick(View view) {
                pick();
            }
        });

    }

    @Override
    protected void onStart() {
        getdbData();
        super.onStart();
    }

    @Override
    public void onBackPressed(){
        startActivity(new Intent(Profile.this, MainActivity.class));
        overridePendingTransition(R.anim.push_right_in,R.anim.push_right_out);
        finish();
        super.onBackPressed();
    }
    public void getdbData(){
        //pb.setVisibility(VISIBLE);
        fs.collection("Users").whereEqualTo("userId", userId).addSnapshotListener(new EventListener<QuerySnapshot>() {
            @Override
            public void onEvent(@Nullable QuerySnapshot queryDocumentSnapshots, @Nullable FirebaseFirestoreException e) {
                if(e!=null){}
                else {
                    for (DocumentChange doc : queryDocumentSnapshots.getDocumentChanges()) {
                        if (doc.getType() == DocumentChange.Type.ADDED) {
                            a.setText(doc.getDocument().getString("name"));
                            b.setText(doc.getDocument().getString("phone"));
                            c.setText(doc.getDocument().getString("email"));
                            d.setText(doc.getDocument().getString("age"));
                            String userId = doc.getDocument().getString("userId").trim();
                            downloadUrl=doc.getDocument().getString("url");
                            f.setText(doc.getDocument().getString("address").trim());
                            }


                        }
                    }
                }

        });
//pb.setVisibility(View.GONE);
```

Profile.xml

This code is about a profile page. we had kept the view, text View, Image Button, Scroll View, Linear Layout, to set heights, widths, background, elevation we have used so constraints. To set the profile perfectly. Building the profile.

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Profile">

    <View
        android:id="@+id/view3"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/accent"
        android:elevation="5dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/back"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:background="@drawable/button_background"
        android:backgroundTint="@android:color/transparent"
        android:elevation="5dp"
        app:layout_constraintBottom_toBottomOf="@+id/view3"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/back_to_home_button" />

    <ScrollView
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="?attr/actionBarSize">

        <com.mikhaellopez.circularimageview.CircularImageView
            android:id="@+id/imageView3"
            android:layout_width="200dp"
            android:layout_height="200dp"
            android:src="@drawable/ic_launcher_background"
            app:civ_border_color="#B53F3F"
            app:civ_border_width="4dp"
            app:civ_shadow="true"
            app:civ_shadow_color="#3f51b5"
            app:civ_shadow_radius="10dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

        <ImageView
            android:id="@+id/imageView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:layout_constraintBottom_toTopOf="@+id/constraintLayout"
            app:layout_constraintEnd_toEndOf="@+id/imageView3"
            app:srcCompat="@android:drawable/ic_menu_edit" />
```

```
android:layout_width="match_parent"
android:layout_height="0dp"
android:paddingTop="50dp"
app:layout_constraintTop_toBottomOf="@+id/imageView3">

<LinearLayout
    android:id="@+id/linearLayout5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
    android:paddingTop="10dp"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout6"
    tools:layout_editor_absoluteX="0dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="horizontal">

        <ImageView
            android:id="@+id/imageView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            app:srcCompat="@drawable/ic_dob" />

        <EditText
            android:id="@+id/editTextTextdate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
```

```
    tools:layout_editor_absoluteX="0dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="horizontal">

        <ImageView
            android:id="@+id/imageView7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            app:srcCompat="@drawable/ic_email" />

        <EditText
            android:id="@+id/editTextTextPersonName"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:ems="10"
            android:inputType="textPersonName" />
    </LinearLayout>

</LinearLayout>

<LinearLayout
    android:id="@+id/linearLayout3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="vertical"
```

```
                android:gravity="center"
                android:orientation="horizontal">

                <ImageView
                    android:id="@+id/imageView12"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    app:srcCompat="@drawable/ic_address" />

                <EditText
                    android:id="@+id/editTextAddress"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:ems="10"
                    android:gravity="start|top"
                    android:inputType="textMultiLine" />

            </LinearLayout>

        </LinearLayout>

    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
    </LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Register.java

To provide the data and the fetch the data from different text fields provided in the register page like name, email, phone, password, age, using the edit text and the buttons login and the register using the OnclickListener.

```java
package com.example.ai_job_predictor;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.gms.tasks.TaskExecutors;
import com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthOptions;
import com.google.firebase.auth.PhoneAuthProvider;
import com.google.firebase.auth.UserProfileChangeRequest;
import com.google.firebase.firestore.DocumentReference;
import com.google.firebase.firestore.FirebaseFirestore;
import com.rilixtech.widget.countrycodepicker.CountryCodePicker;

import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.TimeUnit;

public class Register extends AppCompatActivity {
CountryCodePicker ccp;
EditText namef,emailf,phonef,passwordf,agef,addressf;
TextView login;
Button register;
FirebaseAuth fa;
FirebaseFirestore fs;
    String userId, email, password, phone, name, address,age;
    public boolean validate(String name, String email, String password, String phone, int age) {
        if (TextUtils.isEmpty(name)) {
            namef.setError("Enter a valid name");
            namef.requestFocus();
            return false;
        }
```

```
        if (TextUtils.isEmpty(email) || !email.matches("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
                + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$")) {
            emailf.setError("Invalid Email");
            emailf.requestFocus();
            return false;
        }
        if (!password.matches("^(?=.*[0-9])(?=.*[a-z])(?=.*[@#$%^&+=])(?=\\S+$).{8,}$")) {
            passwordf.setError("password must contain special symbols,numbers and letters combination");
            passwordf.requestFocus();
            return false;
        }
        if (TextUtils.isEmpty(phone)) {
            phonef.setError("Enter a valid Phone");
            return false;
        }
        if(age<=17 ||age>=100){
            agef.setError("Age must be greater than 18");
            agef.requestFocus();
            return false;
        }else {
            return true;
        }
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        ccp=findViewById(R.id.ccp);
        namef=findViewById(R.id.editText3);
        emailf=findViewById(R.id.editText4);
        phonef=findViewById(R.id.editText6);
        passwordf=findViewById(R.id.editText5);
        agef=findViewById(R.id.age);
        addressf=findViewById(R.id.address);
        login=findViewById(R.id.link_login);
        register=findViewById(R.id.button2);
        fa=FirebaseAuth.getInstance();
        fs=FirebaseFirestore.getInstance();
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(Register.this,Login.class));
                overridePendingTransition(R.anim.push_right_in, R.anim.push_right_out);
                finish();
            }
        });
        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
                name = namef.getText().toString().trim();
                email = emailf.getText().toString().trim();
                password = passwordf.getText().toString().trim();
                phone = "+"+ccp.getSelectedCountryCode() + phonef.getText().toString().trim();
                age = agef.getText().toString().trim();
                address=addressf.getText().toString().trim();

                if (validate(name, email, password, phone,Integer.valueOf(age))) {
                    createaccount();
                }
            }
        });
    }
    public void createaccount(){
        fa.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {

                if (task.isSuccessful()) {
                    //pb.setVisibility(View.INVISIBLE);
                    userId = fa.getCurrentUser().getUid();
                    Map<String, Object> user = new HashMap<>();
                    user.put("name",name);
                    user.put("phone",phone);
                    user.put("email",email);
                    user.put("age",age);
                    user.put("userId",userId);
                    user.put("address",address);
                    Toast.makeText(Register.this, "Authentication passed.", Toast.LENGTH_SHORT).show();
                    DocumentReference cf = fs.collection("Users").document(fa.getCurrentUser().getUid());
                    cf.set(user);
                    finish();
                    UserProfileChangeRequest profileUpdates = new UserProfileChangeRequest.Builder()
                            .setDisplayName(name).build();

                    fa.getCurrentUser().updateProfile(profileUpdates);

                } else {

                    Toast.makeText(Register.this, "Authentication failed."+task.getException(), Toast.LENGTH_LONG).show();
                }
                startActivity(new Intent(Register.this, MainActivity.class));
                overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
                finish();
            }
        });

    }
```

Register page:

In this code we are discussing the required information for registering or creating an account in the app. This includes name, password, age, address, Phone,Email. We had kept view, text View, Image Button, Scroll View, Linear Layout, to set heights, widths background, elevation we have used so constraints

1 contributor

160 lines (138 sloc) | 5.48 KB

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/expresso"
    android:fitsSystemWindows="true"
    android:visibility="visible">

    <LinearLayout
```

om/Saikowshik007/AI_Job_Predictor/wiki ="match_parent"

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <com.rilixtech.widget.countrycodepicker.CountryCodePicker
        android:id="@+id/ccp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:gravity="bottom"
        app:ccp_textColor="#FFFFFF" />

    <EditText
        android:id="@+id/editText6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="phone"
        android:textColor="@color/accent" />

</LinearLayout>


<TextView
    android:id="@+id/textView6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Password"
    android:textColor="@color/accent" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="10dp"
    android:text="Name"
    android:textColor="@color/accent" />

<EditText
    android:id="@+id/editText3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="@color/accent"
    android:textColorHint="@color/accent" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Email"
    android:textColor="@color/accent" />

<EditText
    android:id="@+id/editText4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textEmailAddress"
    android:textColor="@color/accent" />
```

Results.java

To retrieve the data from the ListAdapter and to create the output given by the model in a structured manner to be handled easily.

```java
package com.example.ai_job_predictor;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.Toast;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Results extends AppCompatActivity {
    ImageButton back;
    private RecyclerView jobRecycler;
    public static ListAdapter listAdapter;
    public static List<Jobs>jobs=new ArrayList<>();
    @Override
    protected void onStart(){
        fetch();
        jobRecycler = findViewById(R.id.job_recycler);
        jobRecycler.setLayoutManager(new LinearLayoutManager(this));
        listAdapter = new ListAdapter(Results.this,jobs);
        listAdapter.msort();
        jobRecycler.setAdapter(listAdapter);
        jobRecycler.getLayoutManager();
        super.onStart();
    }
```

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_results);
    fetch();

    back=findViewById(R.id.back3);
    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            startActivity(new Intent(Results.this,MainActivity.class));
            jobs.clear();
           finish();
        }
    });
}
public void fetch(){
    jobs.clear();
    Intent i=getIntent();
    DecimalFormat df = new DecimalFormat("##.##");
    List<String>labels= Arrays.asList(".Net Developer","Android developer","Big Data Engineer","Business Analyst","Data Analyst","Data Scientist","Database Administrator",
    float[]results=i.getFloatArrayExtra("list");
    for(int j=0;j<results.length;j++) {
        jobs.add(new Jobs(labels.get(j),Float.parseFloat(df.format(results[j]*100))));
    }
}
}
```

Result page:

In this code we will get the result which means after providing the thecv of a person/student/ candidate we can obtain the result which is shown by this code .According to the resume/cv the result is obtained.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Results">

    <View
        android:id="@+id/view5"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/accent"
        android:elevation="5dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/back3"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:backgroundTint="@android:color/transparent"
        android:elevation="5dp"
        app:layout_constraintBottom_toBottomOf="@+id/view4"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/back_to_home_button" />
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/back3"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:backgroundTint="@android:color/transparent"
        android:elevation="5dp"
        app:layout_constraintBottom_toBottomOf="@+id/view4"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/back_to_home_button" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/job_recycler"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="?attr/actionBarSize"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/view5" />

</androidx.constraintlayout.widget.ConstraintLayout>
```


Search.java

To search the highest matched job profile from the model in indeed.com and to handle the view of the webpage url in the app using the webview component.

```java
package com.example.ai_job_predictor;

import androidx.appcompat.app.AppCompatActivity;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

import android.content.Intent;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.ImageButton;
import android.widget.ProgressBar;
import android.widget.TextView;

public class Search extends AppCompatActivity {
    String job;
    Intent i;
    WebView browser;
    ProgressBar pb;
    SwipeRefreshLayout swipeRefresh;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_search);
        i = getIntent();
        pb = findViewById(R.id.progressBar3);
        job = i.getStringExtra("job");
        browser = findViewById(R.id.webview);
        swipeRefresh = findViewById(R.id.swiperefresh);
        browser.getSettings().setJavaScriptEnabled(true);
        browser.setWebViewClient(new WebViewController());
        String url = "https://www.indeed.co.in/jobs?q=" + job.replace(" ", "%20");
        browser.loadUrl(url);
        browser.canGoBack();
        swipeRefresh.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
            @Override
            public void onRefresh() {
                browser.reload();
                swipeRefresh.setRefreshing(false);
            }
        });
```

```
48      public class WebViewController extends WebViewClient {
49
50          @Override
51          public boolean shouldOverrideUrlLoading(WebView view, String url) {
52              view.loadUrl(url);
53              return true;
54          }
55          @Override
56          public void onPageStarted(WebView view, String url, Bitmap favicon) {
57              pb.setVisibility(View.VISIBLE);
58              super.onPageStarted(view, url, favicon);
59          }
60          @Override
61          public void onPageFinished(WebView view, String url) {
62              super.onPageFinished(view, url);
63              pb.setVisibility(View.GONE);
64          }
65      }
66
67      @Override
68      public boolean onKeyDown(int keyCode, KeyEvent event) {
69          // Check if the key event was the Back button and if there's history
70          if ((keyCode == KeyEvent.KEYCODE_BACK) && browser.canGoBack()) {
71              browser.goBack();
72              return true;
73          }
74          // If it wasn't the Back key or there's no web page history, bubble up to the default
75          // system behavior (probably exit the activity)
76          return super.onKeyDown(keyCode, event);
77      }
78  }
```

Search .xml:

This code is about searching in this we have kept view, text View, Image Button, Scroll View, Linear Layout, to set heights, widths background, elevation we have used so constraints used for Searching propose.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/a
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Search">

    <TextView
        android:id="@+id/textView10"
        android:layout_width="266dp"
        android:layout_height="37dp"
        android:text="Search"
        app:layout_constraintBottom_toBottomOf="@+id/linearLayout2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/linearLayout2"
        app:layout_constraintTop_toTopOf="parent" />

    <ProgressBar
        android:id="@+id/progressBar3"
        style="@android:style/Widget.DeviceDefault.Light.ProgressBar.Large"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:indeterminate="true"
        android:indeterminateTint="@color/primary_dark"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/webview"
        tools:visibility="visible" />
```

```
        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="@android:color/transparent"
            android:elevation="4dp"
            android:minHeight="?attr/actionBarSize"
            android:theme="?attr/actionBarTheme" />
    </LinearLayout>

    <androidx.swiperefreshlayout.widget.SwipeRefreshLayout
        android:id="@+id/swiperefresh"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginTop="?attr/actionBarSize"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <WebView
            android:id="@+id/webview"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:scrollbars="horizontal|vertical">

        </WebView>

    </androidx.swiperefreshlayout.widget.SwipeRefreshLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

SliderAdapter.java

To control the slide animation in the starting page giving the instructions to handle the app and the logic is done using the SliderAdapter.

```java
package com.example.ai_job_predictor;

import android.content.Context;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.viewpager.widget.PagerAdapter;
import androidx.viewpager.widget.ViewPager;

public class SlideAdapter extends PagerAdapter {
    Context context;
    LayoutInflater inflater;public int[] imageArray={R.drawable.a,R.drawable.b,R.drawable.d};
    public String[] titleArray={"Image1","Image2","Image3"};
    public String[] descriptionArray={"A1","A2"};
    public int[] backgroundColorArray={R.color.jet,R.color.oil,R.color.accent,R.color.teal_700};
    public SlideAdapter(Context context){
        this.context=context;

    }
    @Override
    public int getCount() {
        return titleArray.length;
    }
     @Override
     public boolean isViewFromObject(@NonNull View view, @NonNull Object object) {
         return view==object;
     }

     @Override
     public void destroyItem(View container, int position, Object object) {
         ((ViewPager) container).removeView((View) object);
     }

     @NonNull
     @Override
     public Object instantiateItem(@NonNull ViewGroup container, int position) {
         inflater= (LayoutInflater)context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
         View view=inflater.inflate(R.layout.slide,container,false);
         //ImageView imageView=(ImageView)view.findViewById(R.id.);
         LinearLayout linearLayout=view.findViewById(R.id.linearLayout);
         //linearLayout.setBackgroundColor(backgroundColorArray[position]);
         linearLayout.setBackgroundResource(imageArray[position]);

         container.addView(view);
         return view;
     }

}
```

Welcome.java

This code is for the welcome page of the app having the buttons login and register, using the viewpager for the view and sliderAdapter for the scroll feature. On clicking on the login button, the the onClickLister class activates and override happens which takes us to the login in page where the user gets to login to the app if already existed or he has to register an account to use the app

```
package com.example.ai_job_predictor;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;
import androidx.viewpager2.widget.ViewPager2;

import com.google.firebase.auth.FirebaseAuth;
import com.tbuonomo.viewpagerdotsindicator.WormDotsIndicator;

public class Welcome extends AppCompatActivity {
    Button login,register;
    ViewPager viewpager;
    SlideAdapter slideAdapter;
    WormDotsIndicator wormDotsIndicator;
    FirebaseAuth fa;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.welcome);
        login=findViewById(R.id.button3);
        fa= FirebaseAuth.getInstance();
        register=findViewById(R.id.button4);
        slideAdapter=new SlideAdapter(this);
        viewpager=findViewById(R.id.viewPager);
        viewpager.setAdapter(slideAdapter);
        wormDotsIndicator = (WormDotsIndicator) findViewById(R.id.worm_dots_indicator);
        wormDotsIndicator.setViewPager(viewpager);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(Welcome.this,Login.class));
                overridePendingTransition(R.anim.push_right_in,R.anim.push_right_out);
            }
        });
        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(Welcome.this,Register.class));
                overridePendingTransition(R.anim.push_left_in, R.anim.push_left_out);
            }
        });
```

Welcome.xml:

This page about the welcome contains the details related to login ang signup page from the welcome page we had redirected login and sign up pages we had used here. Image viewer,edit view,linear Layout. And some constraints.

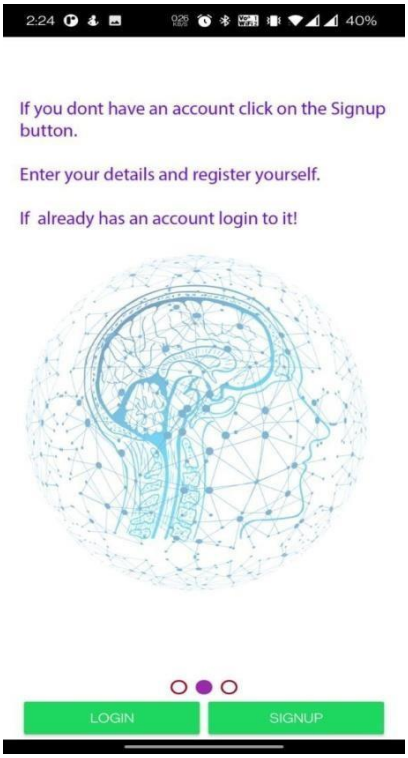
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/viewPager"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" >

    </androidx.viewpager.widget.ViewPager>

    <com.tbuonomo.viewpagerdotsindicator.WormDotsIndicator
        android:id="@+id/worm_dots_indicator"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="50dp"
        app:dotsColor="@color/purple_500"
        app:dotsCornerRadius="8dp"
        app:dotsSize="16dp"
        app:dotsSpacing="4dp"
        app:dotsStrokeColor="@color/design_default_color_error"
        app:dotsStrokeWidth="2dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
```

```
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:gravity="bottom"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:layout_marginStart="16dp"
        android:layout_weight="1"
        android:insetRight="2dp"
        android:text="Login"
        app:backgroundTint="#1ed760" />

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_marginEnd="16dp"
        android:layout_weight="1"
        android:insetLeft="2dp"
        android:text="Signup"
        app:backgroundTint="#1ed760" />
</LinearLayout>
```

```
        app:layout_constraintStart_toStartOf="parent" />

        <Button
            android:id="@+id/button3"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="center"
            android:layout_marginStart="16dp"
            android:layout_weight="1"
            android:insetRight="2dp"
            android:text="Login"
            app:backgroundTint="#1ed760" />

        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_marginEnd="16dp"
            android:layout_weight="1"
            android:insetLeft="2dp"
            android:text="Signup"
            app:backgroundTint="#1ed760" />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

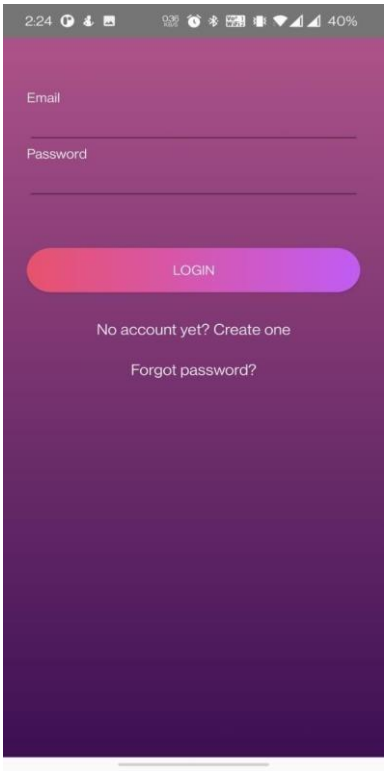
Application GUI

Welcome

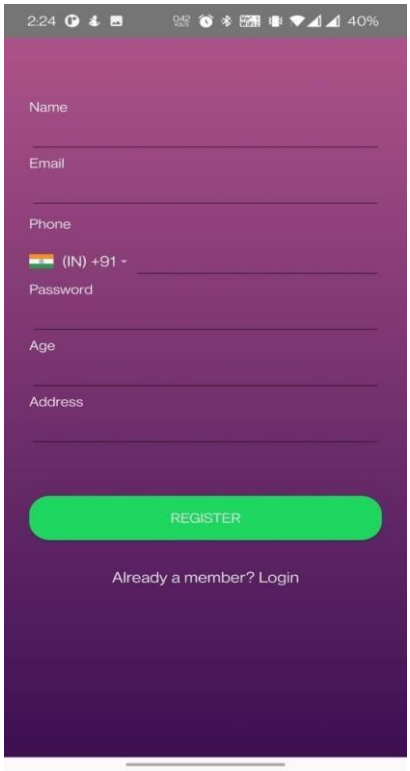


Tutorial screen1

Tutorial screen2

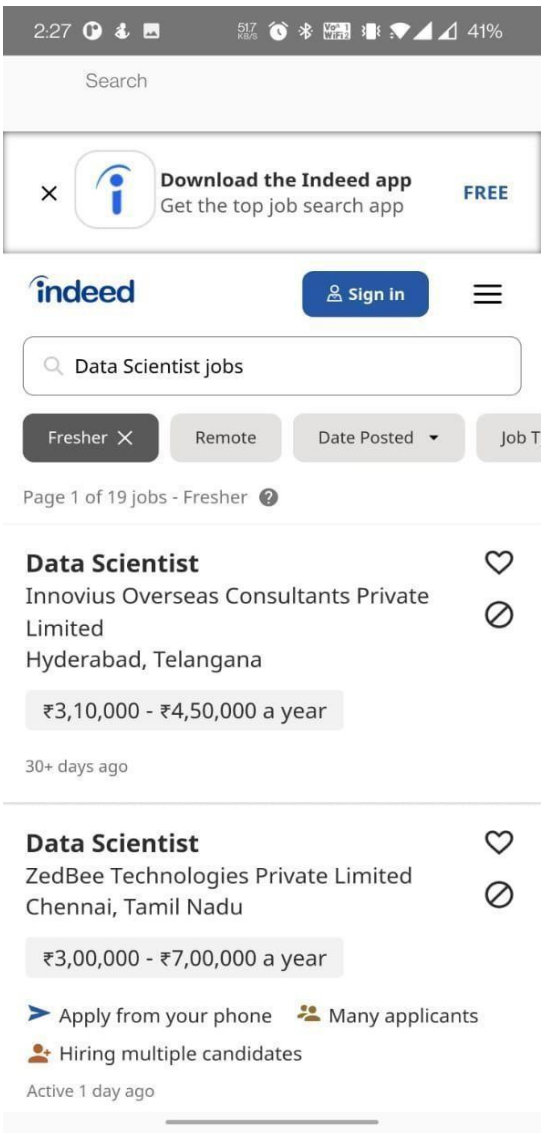


Login screen



Register screen

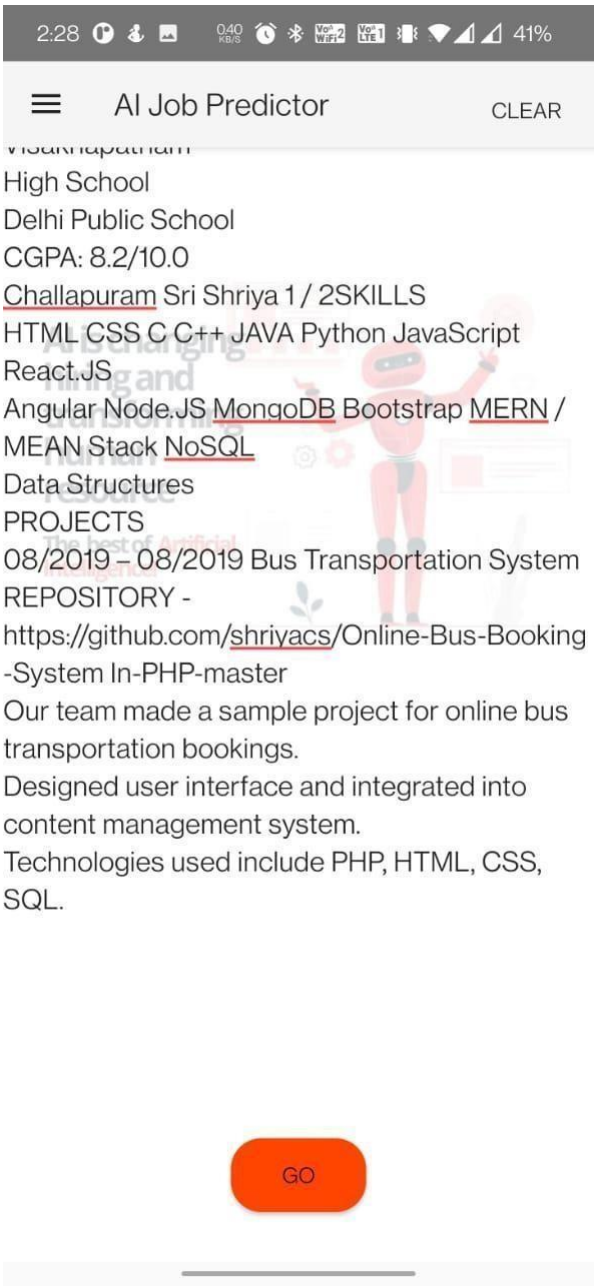
Main screen



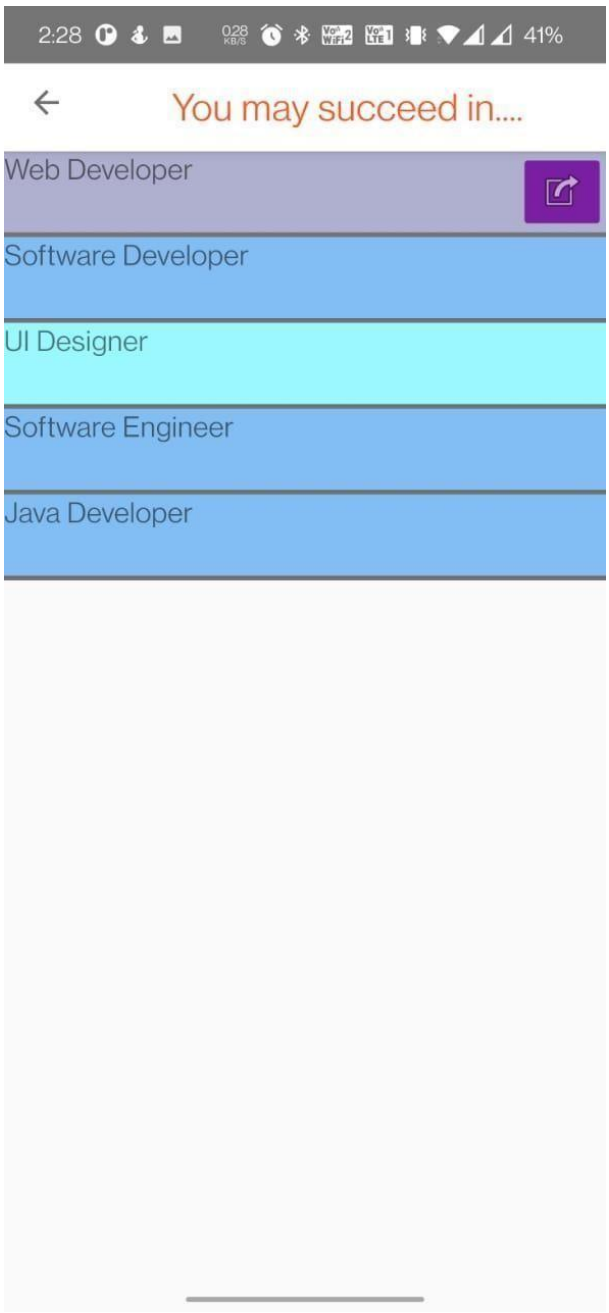
Search screen

Profile screen

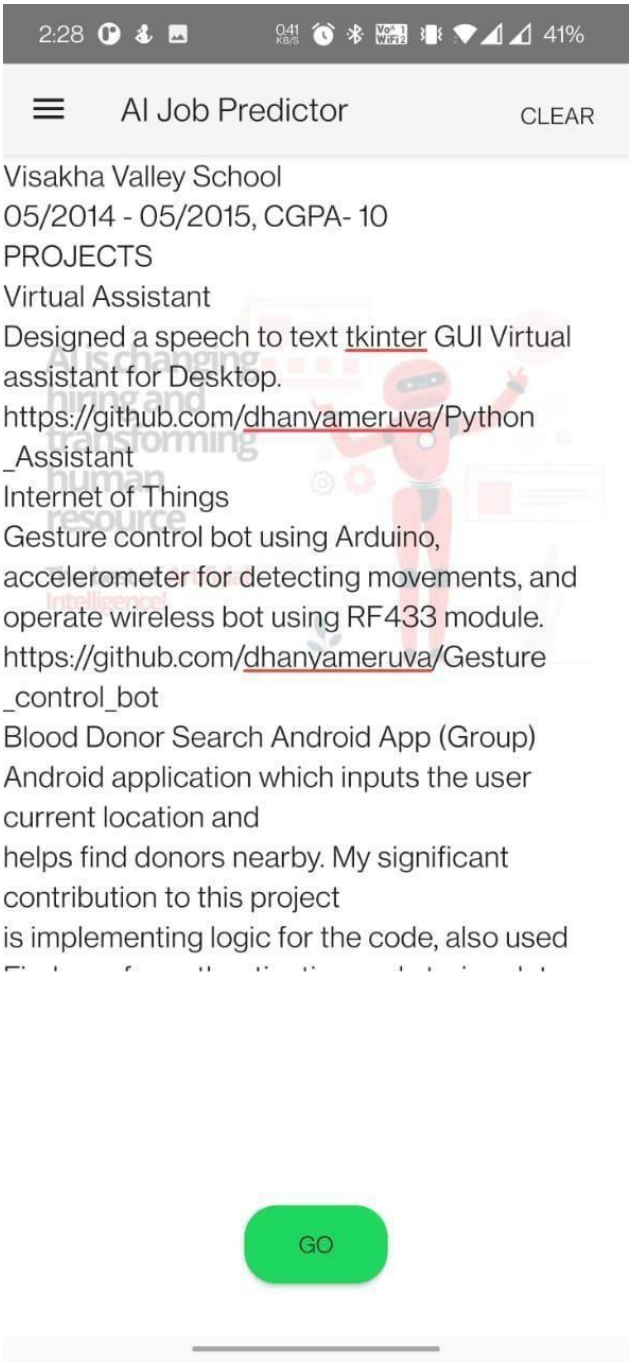
Test Case1:



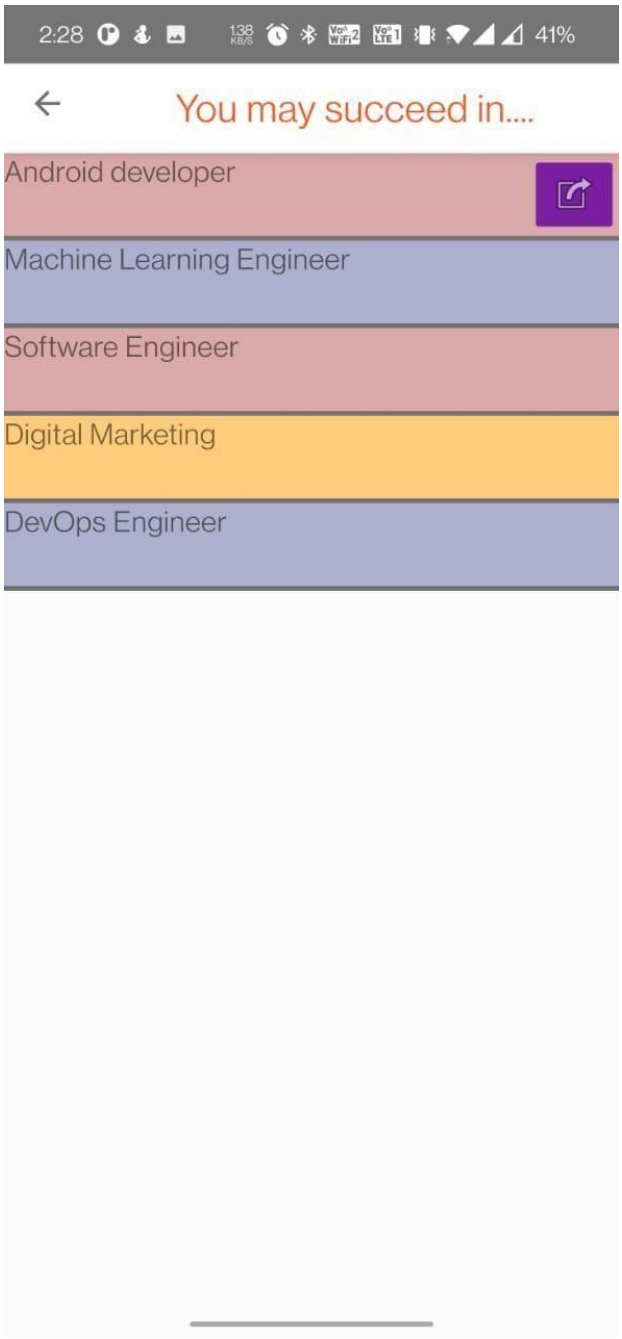
Result:



Test case 2:



Result:



REFERENCES

1.https://firebase.google.com/ for Backend of application

2.https://stackoverflow.com/ for debugging

3. https://developer.android.com/studio?gclid=Cj0KCQjw-LOEBhDCARIsABrC0Tm_gqjAXNzQzBPlMocDy6qdPUEo4pejOuaFwK5CelP1111O-muf0GkaApdZEALw_wcB&gclsrc=aw.ds for getting SDK tools and emulator and also to download android studio

4. https://github.com/smarteist/Android-Image-Slider a reference for slide adapter

5. https://arxiv.org/pdf/1912.12214 reference paper for the whole project idea

6. https://medium.com/voice-tech-podcast/text-classification-using-cnn-9ade8155dfb9 for understanding sample CNN

7.https://drive.google.com/file/d/1_28YPTmwVm6EKrhY1ZcRPjtWX4occRYL/view?usp=sharing custom dataset link prepared from indeed.com through web scraping

8. https://www.javatpoint.com/android-recyclerview-list-example a reference to implement recycler view in our application.

9. https://youtu.be/BbRpgDzgcj0 youtube video for demonstration of the application

10. https://github.com/Saikowshik007/AI_Job_Predictor Project git link