Appendix-6

CHAPTER – 1

Introduction

1.1 Introduction

Here we are introducing a mobile application using machine learning and Android Studio can be employed to predict the likelihood of a candidate's placement success based on their profile and qualifications. Furthermore, with the widespread use of smartphones, developing an Android application can provide an accessible and user-friendly platform for placement prediction.This project aims to combine the power of machine learning algorithms with the convenience of an Android application to assist students and job seekers in predicting their chances of getting placed. By analyzing historical placement data and leveraging the capabilities of machine learning models, this system can provide valuable insights into a candidate's placement potential.

The project will be implemented using the Android Studio development environment, which offers a rich set of tools and libraries for Android app development. Android Studio provides an intuitive interface for designing user interfaces and enables seamless integration of machine learning models. The project will be implemented using the Android Studio development environment, which offers a rich set of tools and libraries for Android app development. Android Studio provides an intuitive interface for designing user interfaces and enables seamless integration of machine learning models.

Key Features of the Placement Prediction System:

User Profile Input: The Android application will allow users to input their relevant details such as educational qualifications, skills, projects, internships, and other pertinent information necessary for placement prediction.

Data Collection and Preprocessing: The system will gather and preprocess historical placement data from various sources, including previous placement records, student profiles, and industry trends. This data will serve as the training dataset for the machine learning model.

Machine Learning Model Training: Various machine learning algorithms such as decision trees, random forests, or support vector machines will be trained using the collected data. These models will learn from the past placement patterns and predict the likelihood of a successful placement for a given candidate.

Prediction and Visualization: Once the machine learning models are trained, the Android application will utilize these models to provide real-time placement predictions based on user input. The results will be presented to the user in a user-friendly and visually appealing manner, allowing them to interpret their chances of placement.

Performance Evaluation: The accuracy and effectiveness of the machine learning models will be evaluated using appropriate metrics, such as precision, recall, and F1 score. This evaluation will ensure the reliability of the predictions and provide insights into model improvements.

By combining machine learning and Android app development, this project aims to empower students and job seekers with a tool that can assist them in making informed decisions regarding their career choices. The application will provide valuable insights into the likelihood of placement success, enabling users to focus their efforts on the most promising opportunities and enhance their overall career prospects.

1.2 Problem Statement

The problem we aim to solve is the uncertainty and lack of information faced by students and job seekers regarding their placement prospects. With numerous factors influencing the placement process, including qualifications, skills, and industry trends, it becomes challenging for individuals to assess their chances of getting placed accurately.

1.3 Objective

The objective of this project is to develop a machine learning-based placement prediction system integrated into an Android application. The system will leverage historical placement data and user profiles to predict the likelihood of a candidate's success in securing a job placement. The Android application will provide an intuitive interface for users to input their information and receive real-time placement predictions.

Data Collection and Preprocessing: Gather relevant historical placement data from various sources, including previous placement records, student profiles, and industry trends. Preprocess the data to ensure its quality, consistency, and suitability for training machine learning models.\

Machine Learning Model Training: Select appropriate machine learning algorithms, such as decision trees, random forests, or support vector machines, to train the placement prediction models. Utilize the preprocessed data to train these models, allowing them to learn from historical placement patterns.

Android Application Development: Utilize Android Studio to develop an intuitive and user-friendly Android application. Design an interface that allows users to input their relevant details and receive real-time placement predictions based on the trained machine learning models.

1.4 Goal of Project

The goal of this project is to develop a robust and accurate placement prediction system integrated into an Android application. The project aims to provide students and job seekers with a reliable tool that can predict their placement success based on their qualifications and profile.

Accurate Placement Predictions: Develop machine learning models that can accurately predict the likelihood of a candidate's success in securing a job placement. The goal is to create models that capture the complex relationships between various factors and provide reliable predictions to the users.

User-Friendly Interface: Design an intuitive and user-friendly interface within the Android application to allow users to input their information easily. The goal is to create an application that is accessible and straightforward, even for individuals with limited technical expertise.

By achieving these goals, the project aims to contribute to the career development of students and job seekers by providing them with a reliable tool that can guide their placement decisions. The project seeks to leverage the capabilities of machine learning and the convenience of an Android application to create a valuable resource for individuals navigating the competitive job market.

Chapter 2

Problem Identification

2.1 Existing System

The existing system for placement prediction using machine learning and Android Studio aims to provide individuals with valuable insights into their job placement prospects. The system utilizes machine learning algorithms and leverages the capabilities of Android Studio to develop a mobile application that assists individuals in making informed decisions regarding their career paths.In the existing system, relevant data about individuals, such as academic performance, technical skills, internships, and extracurricular activities, is collected. This data is preprocessed to ensure its quality and compatibility with machine learning algorithms. Steps like handling missing values, encoding categorical variables, and normalizing numerical features are performed during the preprocessing phase.

Limitations of the existing system:

- High false positives: There is no interactive tool for users to check for their attrition.
- Technology Complexity: Most of the system is complex to understand, and not user-friendly as compared to our proposed system.
- Time-Consuming Feature: In the existing system, the performance is low and most of the time system gets hung due to load.
- Not Easy to Understand: Systems are complex to understand and they were not user-friendly.

2.2 Proposed System

The proposed system for placement prediction using machine learning and Android Studio aims to enhance the accuracy and usability of job placement predictions for individuals. By leveraging machine learning algorithms and the capabilities of Android Studio, the system seeks to provide individuals with a comprehensive and user-friendly mobile application to assess their placement prospects. In the proposed system, relevant data about individuals, such as academic performance, technical skills, internships, and extracurricular activities, will be collected. This data will undergo preprocessing to ensure its quality and compatibility with machine learning algorithms. Steps like handling missing values, encoding categorical variables, and normalizing numerical features will be performed during the preprocessing phase.

To integrate the model into an Android application, it will be incorporated into an Android Studio project. The application will provide a user-friendly interface where individuals can input their relevant information. The model will process this data to generate a prediction regarding their likelihood of securing a job placement. The application will also provide personalized insights, recommendations, and resources to help individuals improve their chances of getting placed. Thorough testing will be conducted to ensure the functionality, usability, and accuracy of the Android application.

Chapter 3

Requirements

3.1 Software Requirements

Android Studio: The integrated development environment (IDE) for Android application development.

Java: Programming languages used for Android app development.

Android Software Development Kit (SDK): Provides tools and libraries for building, testing, and debugging Android applications.

Jupyter Notebook: This is used for building machine learning models.

AWS(EC2 Server): It is used for deploying the project in the server which builds FastAPI using pickle and python files.

3.2 Hardware Requirements

Ethernet connection (LAN) OR a wireless adapter (Wi-Fi).

Recommended 64 GB.

Memory (RAM): Minimum 1 GB; Recommended 4 GB

Chapter 4

Design and Implementation

4.1 Design

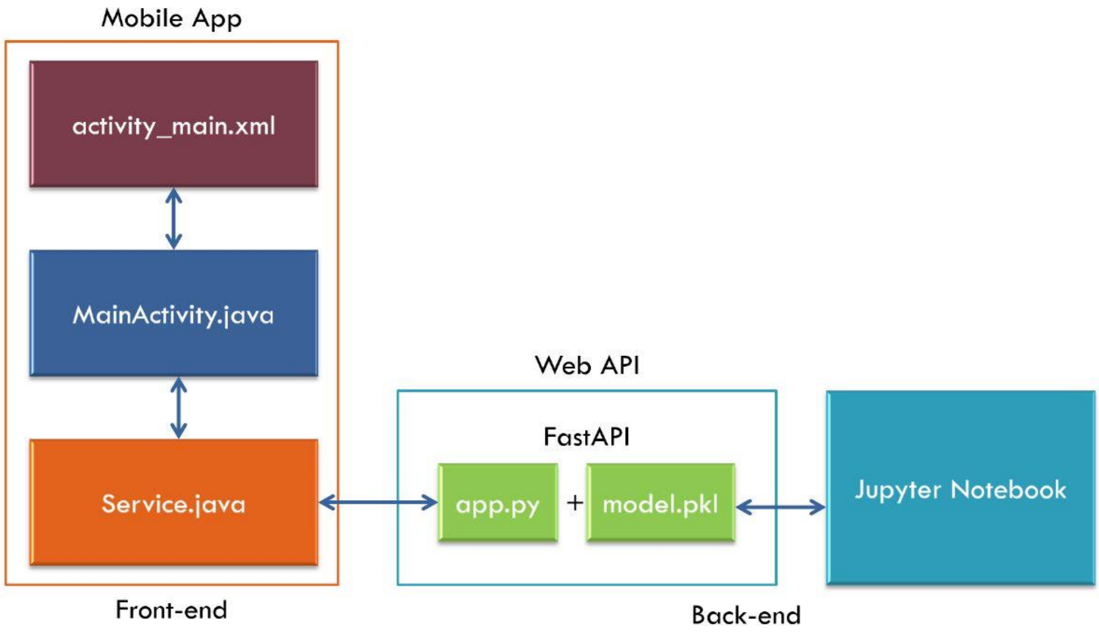


Fig 4.1.1

4.2 Implementation

1. **Identify Data Requirements**: Determine the types of employee data needed for prediction, such as demographics, performance metrics, job satisfaction, and turnover history.

2. **Data Collection and Preprocessing**: Gather relevant data from various sources and preprocess it to handle missing values, outliers, and inconsistencies. Perform feature engineering, such as creating new variables or transforming existing ones, to enhance the predictive power of the data.

3. **Model Selection**: Choose an appropriate machine learning algorithm or ensemble of algorithms based on the problem requirements. Common algorithms used for turnover prediction include logistic regression, decision trees, random forests, and support vector machines.

4. **Model Training and Evaluation**: Split the pre-processed data into training and testing sets. Train the selected machine learning model on the training data and evaluate its performance using suitable metrics like accuracy.

5. **FastAPI**

a) **Pickle File:** The file which is used to dump the whole machine learning code.

b) **Python File(app.py):** Python files are used for routes and models.

6**.EC2 Server:** We are deploying the FastAPI in the server to get FastAPI URL.

7. **User Interface Design**: Design an intuitive user interface in Android Studio to capture relevant employee data for prediction. Create input fields and user interface elements for data entry.

8. **Java**: Programming languages used for Android app development.

9. **Mobile view**: Real-time view of the whole application.

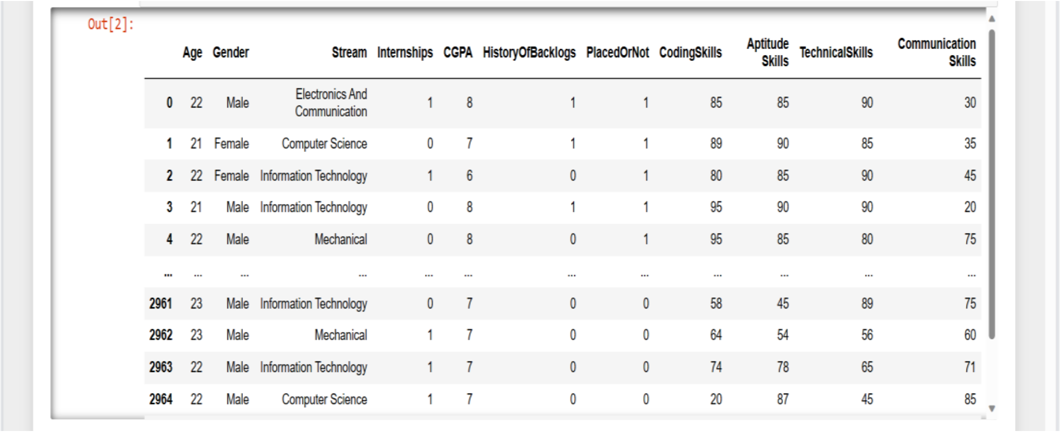
Chapter 5

Code

5.1 Source Code

A. Machine Learning code in Jupyter.

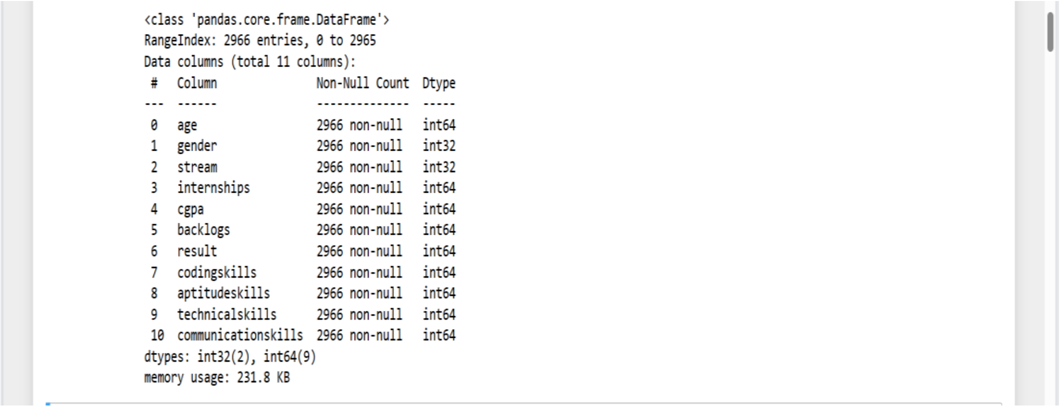
```
df = pd.read_csv("https://placement-
analysis.s3.amazonaws.com/placementprediction.csv")
df
```



Out[2]:

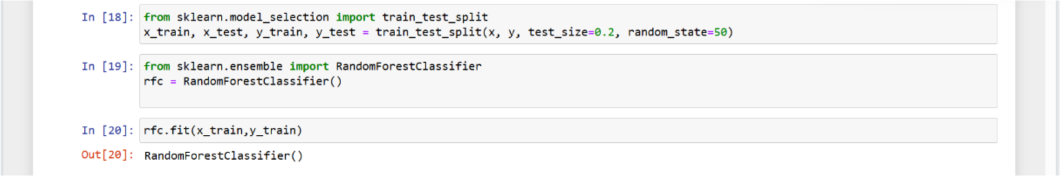
	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot	CodingSkills	Aptitude Skills	TechnicalSkills	Communication Skills
0	22	Male	Electronics And Communication	1	8	1	1	85	85	90	30
1	21	Female	Computer Science	0	7	1	1	89	90	85	35
2	22	Female	Information Technology	1	6	0	1	80	85	90	45
3	21	Male	Information Technology	0	8	1	1	95	90	90	20
4	22	Male	Mechanical	0	8	0	1	95	85	80	75
...	...	...	...	...	...	...	...	...	...	...	...
2961	23	Male	Information Technology	0	7	0	0	58	45	89	75
2962	23	Male	Mechanical	1	7	0	0	64	54	56	60
2963	22	Male	Information Technology	1	7	0	0	74	78	65	71
2964	22	Male	Computer Science	1	7	0	0	20	87	45	85

```
df.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 11 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   age                  2966 non-null   int64
 1   gender               2966 non-null   int32
 2   stream               2966 non-null   int32
 3   internships          2966 non-null   int64
 4   cgpa                 2966 non-null   int64
 5   backlogs             2966 non-null   int64
 6   result               2966 non-null   int64
 7   codingskills         2966 non-null   int64
 8   aptitudeskills       2966 non-null   int64
 9   technicalskills      2966 non-null   int64
 10  communicationskills  2966 non-null   int64
dtypes: int32(2), int64(9)
memory usage: 231.8 KB
```

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=50)
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
rfc.fit(x_train,y_train)
```

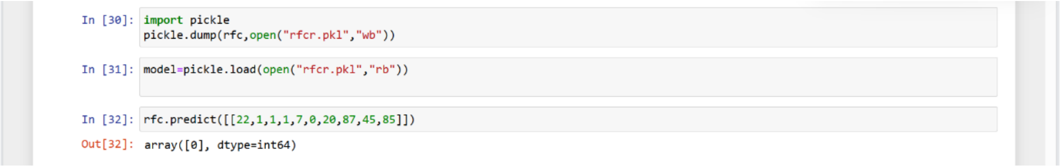


```
In [18]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=50)

In [19]: from sklearn.ensemble import RandomForestClassifier
         rfc = RandomForestClassifier()

In [20]: rfc.fit(x_train,y_train)
Out[20]: RandomForestClassifier()
```

```
import pickle
pickle.dump(rfc,open("rfcr.pkl","wb"))
```

```
model=pickle.load(open("rfcr.pkl","rb"))
```

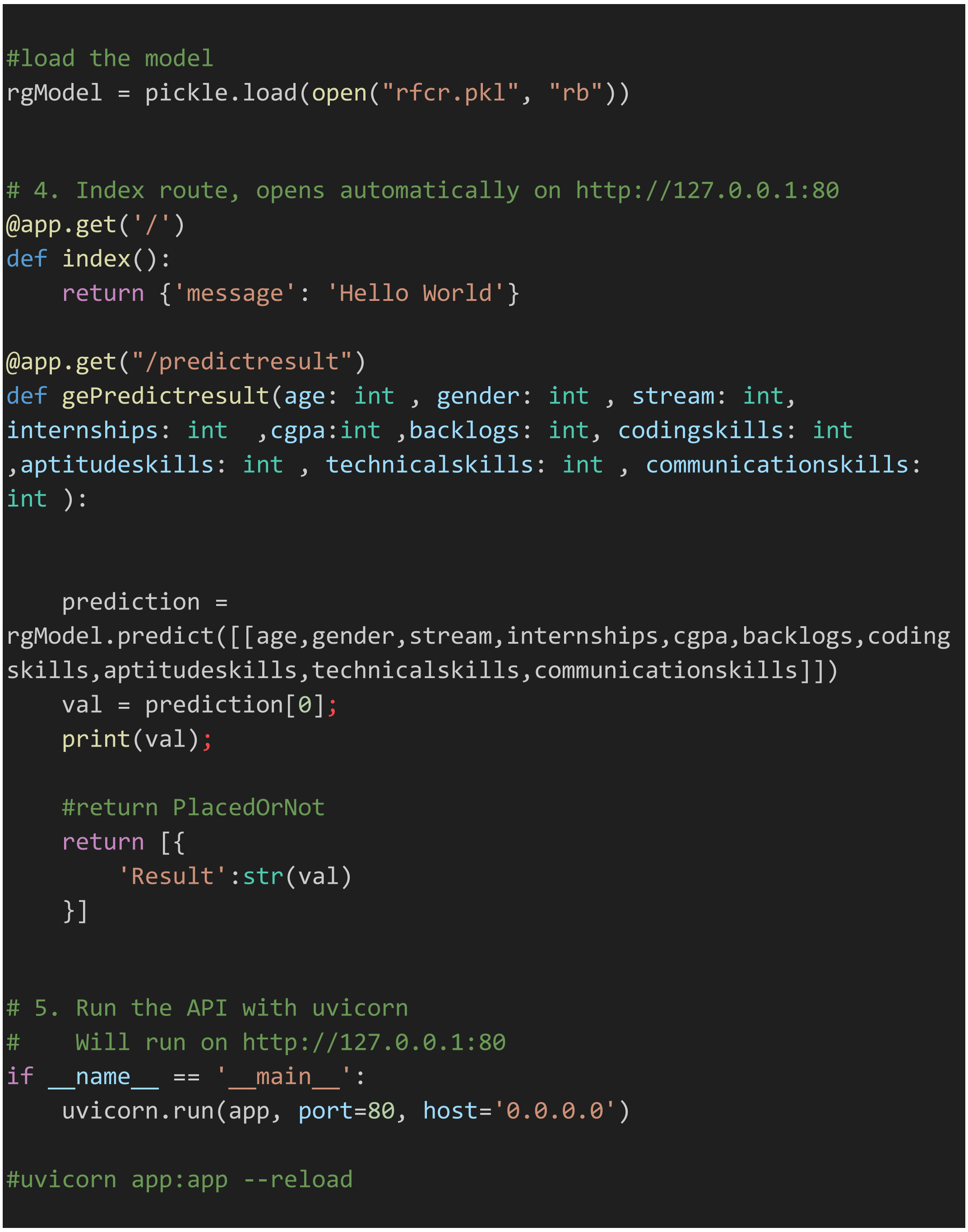
```
rfc.predict([[22,1,1,1,7,0,20,87,45,85]])
```



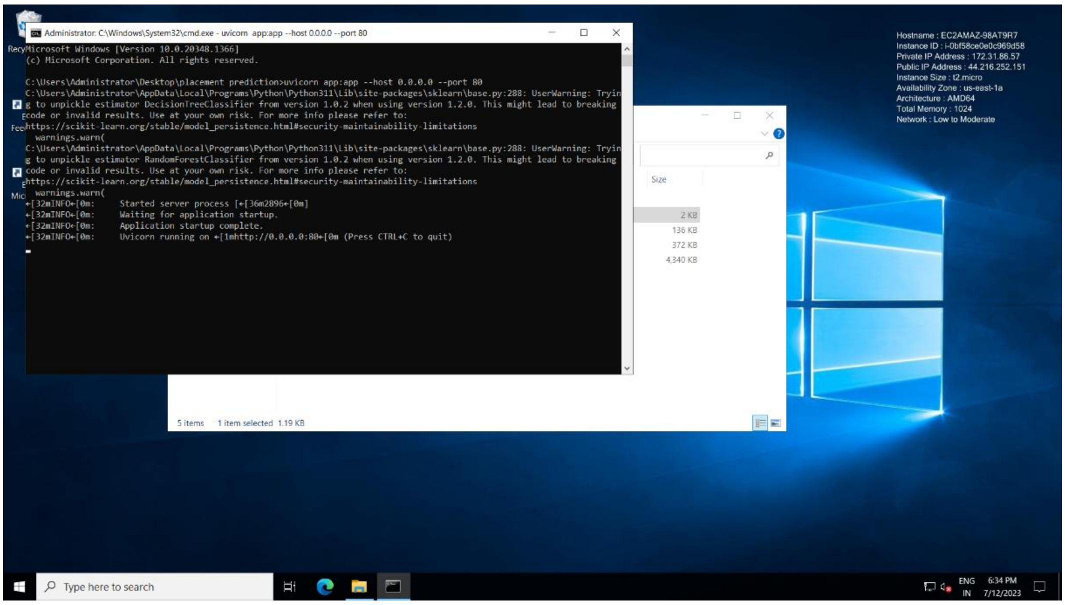
```
In [30]: import pickle
         pickle.dump(rfc,open("rfcr.pkl","wb"))

In [31]: model=pickle.load(open("rfcr.pkl","rb"))

In [32]: rfc.predict([[22,1,1,1,7,0,20,87,45,85]])
Out[32]: array([0], dtype=int64)
```

B. FastAPI (app.py)

```
# 1. Library imports
import uvicorn
from fastapi import FastAPI


from fastapi.middleware.cors import CORSMiddleware
import pickle


# 2. Create the app object
app = FastAPI()

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```



```
#load the model
rgModel = pickle.load(open("rfcr.pkl", "rb"))


# 4. Index route, opens automatically on http://127.0.0.1:80
@app.get('/')
def index():
    return {'message': 'Hello World'}

@app.get("/predictresult")
def gePredictresult(age: int , gender: int , stream: int,
internships: int  ,cgpa:int ,backlogs: int, codingskills: int
,aptitudeskills: int , technicalskills: int , communicationskills:
int ):


    prediction =
rgModel.predict([[age,gender,stream,internships,cgpa,backlogs,coding
skills,aptitudeskills,technicalskills,communicationskills]])
    val = prediction[0];
    print(val);

    #return PlacedOrNot
    return [{
        'Result':str(val)
    }]


# 5. Run the API with uvicorn
#    Will run on http://127.0.0.1:80
if __name__ == '__main__':
    uvicorn.run(app, port=80, host='0.0.0.0')

#uvicorn app:app --reload
```


Administrator: C:\Windows\System32\cmd.exe - uvicorn app:app --host 0.0.0.0 --port 80
Size
2 KB
136 KB
372 KB
4,340 KB
5 items 1 item selected 1.19 KB
Type here to search
ENG
IN
6:34 PM
7/12/2023

```
Microsoft Windows [Version 10.0.20348.1366]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\placement prediction>uvicorn app:app --host 0.0.0.0 --port 80
C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:288: UserWarning: Tryin
g to unpickle estimator DecisionTreeClassifier from version 1.0.2 when using version 1.2.0. This might lead to breaking
code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:288: UserWarning: Tryin
g to unpickle estimator RandomForestClassifier from version 1.0.2 when using version 1.2.0. This might lead to breaking
code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
←[32mINFO←[0m:     Started server process [←[36m2896←[0m]
←[32mINFO←[0m:     Waiting for application startup.
←[32mINFO←[0m:     Application startup complete.
←[32mINFO←[0m:     Uvicorn running on ←[1mhttp://0.0.0.0:80←[0m (Press CTRL+C to quit)
```

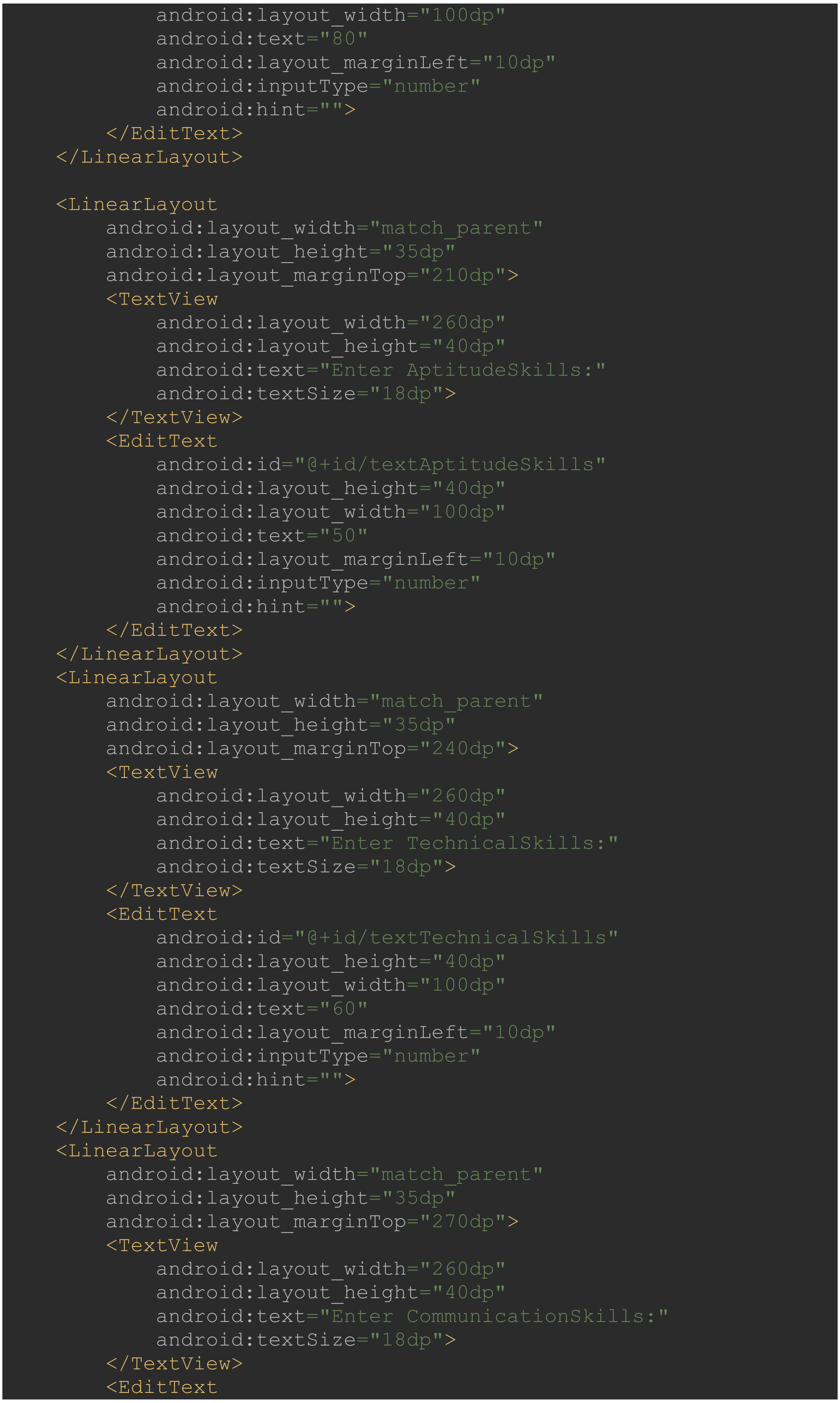
Hostname : EC2AMAZ-98AT9R7
Instance ID : i-0bf58ce0e0c969d58
Private IP Address : 172.31.86.57
Public IP Address : 44.216.252.151
Instance Size : t2.micro
Availability Zone : us-east-1a
Architecture : AMD64
Total Memory : 1024
Network : Low to Moderate

Fig:5.1.1

C. Frontend (Android Studio)

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    tools:context=".HomeFragment">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="20dp"
            android:text="Enter Age:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textAge"
            android:layout_height="40dp"
            android:text="22"
            android:layout_width="100dp"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="30dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Gender:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textGender"
            android:layout_height="40dp"
            android:text="1"
            android:layout_width="100dp"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="60dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Stream:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textStream"
            android:layout_height="40dp"
            android:text="2"
            android:layout_width="100dp"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="90dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Internships:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textInternships"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="1"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="120dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter CGPA:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textCGPA"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="8"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="150dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Backlogs:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textBacklogs"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="1"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="180dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter CodingSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textCodingSkills"
            android:layout_height="40dp"
```

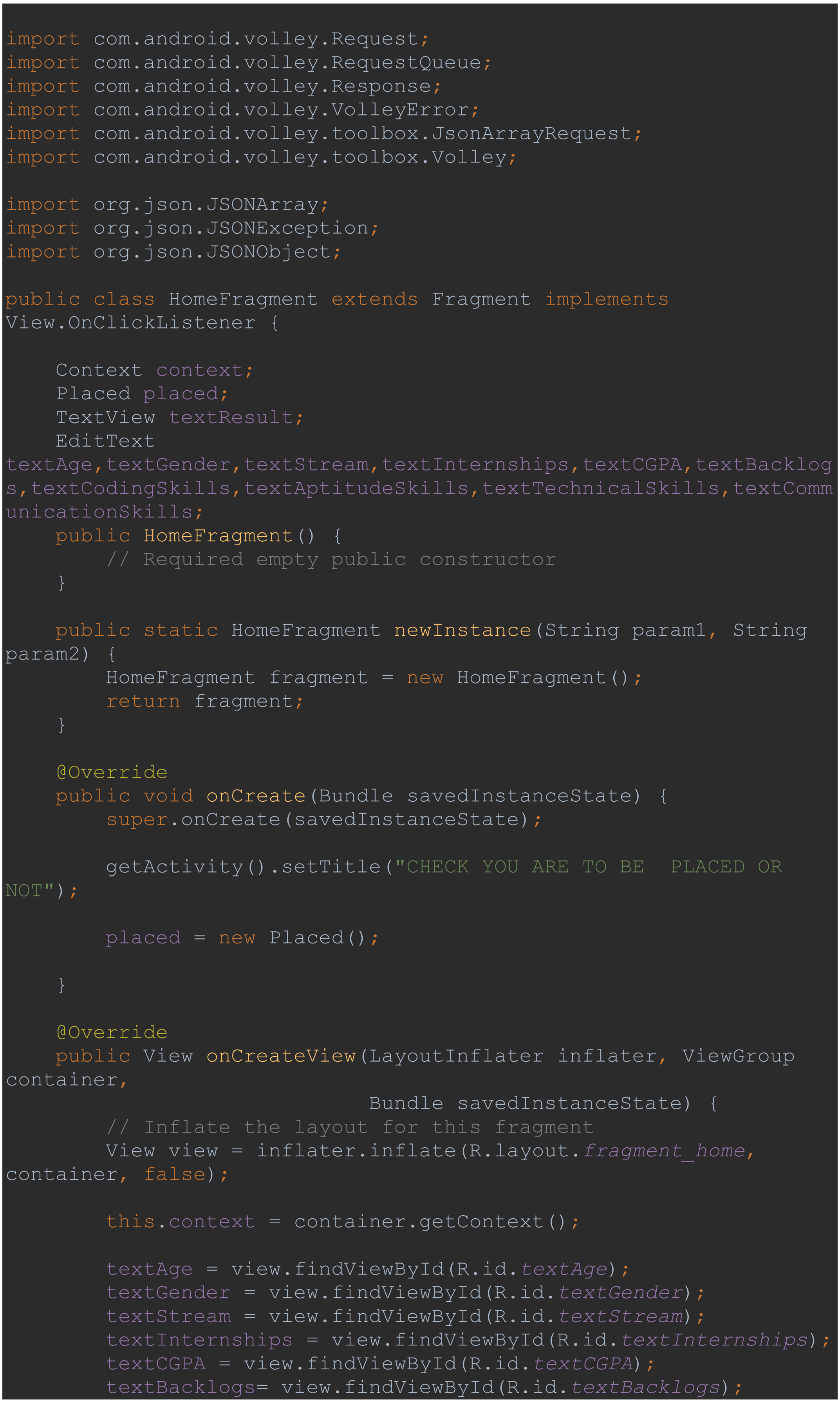


```
            android:layout_width="100dp"
            android:text="80"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="210dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter AptitudeSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textAptitudeSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="50"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="240dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter TechnicalSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textTechnicalSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="60"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="270dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter CommunicationSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
```

```
            android:id="@+id/textCommunicationSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="75"
            android:layout_marginLeft="10dp"
            android:inputType="numberDecimal"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="310dp">
        <Button
            android:id="@+id/btnResult"
            android:layout_width="match_parent"
            android:layout_height="50sp"
            android:layout_marginTop="2dp"
            android:onClick="predictResult"
            android:text="PREDICT">
        </Button>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="340dp">
        <TextView
            android:id="@+id/textResult"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=""
            android:textSize="20sp"
            android:layout_marginTop="20dp">
        </TextView>
    </LinearLayout>

</FrameLayout>
```

MainActivity.java

```
package com.example.placement_predict;
import android.content.Context;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
```



```
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class HomeFragment extends Fragment implements
View.OnClickListener {

    Context context;
    Placed placed;
    TextView textResult;
    EditText
textAge,textGender,textStream,textInternships,textCGPA,textBacklog
s,textCodingSkills,textAptitudeSkills,textTechnicalSkills,textComm
unicationSkills;
    public HomeFragment() {
        // Required empty public constructor
    }

    public static HomeFragment newInstance(String param1, String
param2) {
        HomeFragment fragment = new HomeFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getActivity().setTitle("CHECK YOU ARE TO BE  PLACED OR
NOT");

        placed = new Placed();

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_home,
container, false);

        this.context = container.getContext();

        textAge = view.findViewById(R.id.textAge);
        textGender = view.findViewById(R.id.textGender);
        textStream = view.findViewById(R.id.textStream);
        textInternships = view.findViewById(R.id.textInternships);
        textCGPA = view.findViewById(R.id.textCGPA);
        textBacklogs= view.findViewById(R.id.textBacklogs);
        textCodingSkills =
view.findViewById(R.id.textCodingSkills);
        textAptitudeSkills =
view.findViewById(R.id.textAptitudeSkills);
        textTechnicalSkills =
view.findViewById(R.id.textTechnicalSkills);
        textCommunicationSkills =
view.findViewById(R.id.textCommunicationSkills);

        textResult = view.findViewById(R.id.textResult);

        Button btnResult = view.findViewById(R.id.btnResult);
        btnResult.setOnClickListener(this);

        return view;
    }

    private void GetResult() {

        String url = "http://44.216.252.151/predictresult?";
        url += "age=" + placed.Age + "&";
        url += "gender=" + placed.Gender + "&";
        url += "stream=" + placed.Stream + "&";
        url += "internships=" + placed.Internships + "&";
        url += "cgpa=" + placed.CGPA + "&";
        url += "backlogs=" + placed.Backlogs + "&";
        url += "codingskills=" + placed.CodingSkills+ "&";
        url += "aptitudeskills=" + placed.AptitudeSkills+ "&";
        url += "technicalskills=" + placed.TechnicalSkills + "&";
        url += "communicationskills=" +
placed.CommunicationSkills;
        Log.d("url",url);

        // creating a new variable for our request queue
        RequestQueue queue = Volley.newRequestQueue(context);

        // make json array request and then extracting data from
each json object.
        JsonArrayRequest jsonArrayRequest = new
JsonArrayRequest(Request.Method.GET, url, null, new
Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                try {
                    JSONObject responseObj =
response.getJSONObject(0);

                    String returnValue =
responseObj.getString("Result");
//                    textResult.setText("Result: " +
returnValue);
                    if(returnValue.equals("1")){
                        textResult.setText("You are Placed");
                    }else{
                        textResult.setText("You are not to be
placed. Improve your skills");
                    }

                    Toast.makeText(context, returnValue,
Toast.LENGTH_SHORT).show();

                } catch (JSONException e) {
                    throw new RuntimeException(e);
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(context, error.toString(),
Toast.LENGTH_SHORT).show();
                Log.d("Error", error.toString());

            }
        });
        queue.add(jsonArrayRequest);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btnResult:
                placed.Age =  textAge.getText().toString();
                placed.Gender =  textGender.getText().toString();
                placed.Stream =  textStream.getText().toString();
                placed.Internships =
textInternships.getText().toString();
                placed.Backlogs =
textBacklogs.getText().toString();
                placed.CGPA =  textCGPA.getText().toString();
                placed.CodingSkills =
textCodingSkills.getText().toString();
                placed.AptitudeSkills =
textAptitudeSkills.getText().toString();
                placed.TechnicalSkills =
textTechnicalSkills.getText().toString();
                placed.CommunicationSkills =
textCommunicationSkills.getText().toString();



                GetResult();
                textResult.setText("wait..");
                break;
        }
    }
}
```

activity_splash.xml

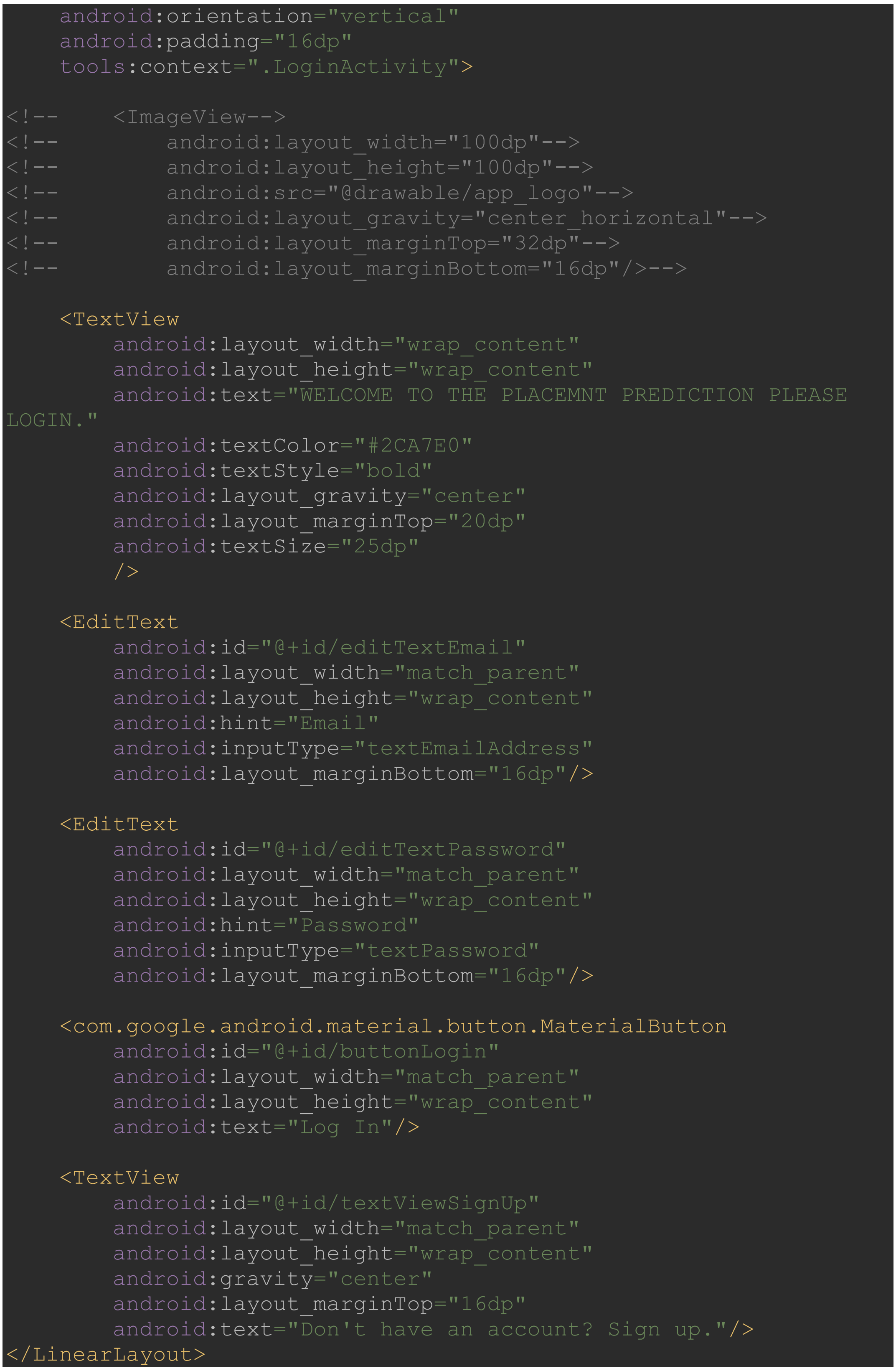
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SplashActivity"
    android:background="@drawable/logo5">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"

        android:layout_marginLeft="60dp"
        android:layout_marginTop="50dp"
        android:layout_marginRight="30dp"
        android:text=""
        android:textColor="@color/black"
        android:textSize="40dp">

    </TextView>


</RelativeLayout>
```

SplashActivity.java

```
package com.example.placement_predict;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;

public class SplachActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splach);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(SplachActivity.this ,
LoginActivity.class);
                startActivity(intent);
                finish();

            }
        },4000);
    }
}
```

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

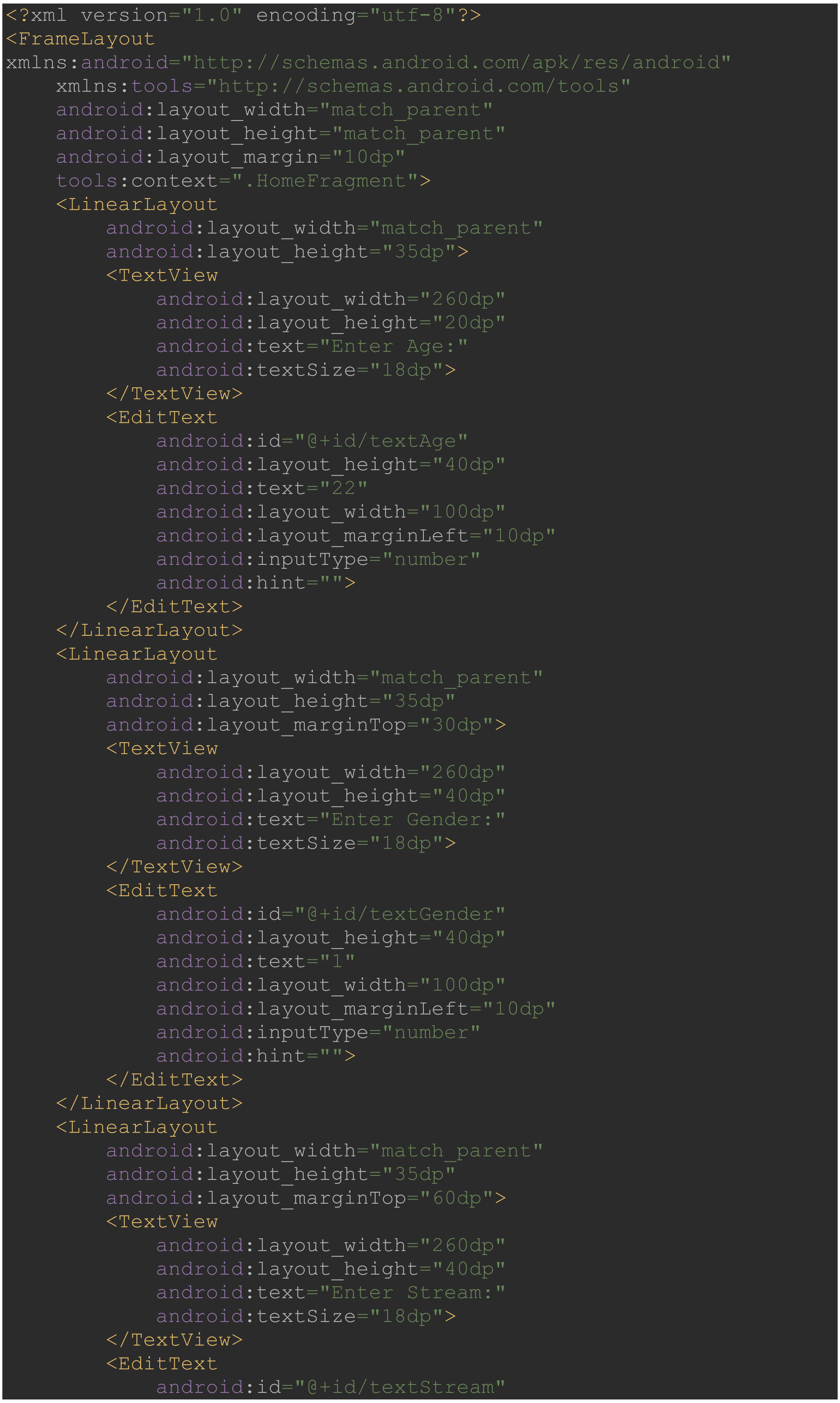


```
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".LoginActivity">

<!--    <ImageView-->
<!--        android:layout_width="100dp"-->
<!--        android:layout_height="100dp"-->
<!--        android:src="@drawable/app_logo"-->
<!--        android:layout_gravity="center_horizontal"-->
<!--        android:layout_marginTop="32dp"-->
<!--        android:layout_marginBottom="16dp"/>-->

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WELCOME TO THE PLACEMNT PREDICTION PLEASE
LOGIN."
        android:textColor="#2CA7E0"
        android:textStyle="bold"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:textSize="25dp"
        />

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="16dp"/>

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>

    <com.google.android.material.button.MaterialButton
        android:id="@+id/buttonLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Log In"/>

    <TextView
        android:id="@+id/textViewSignUp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_marginTop="16dp"
        android:text="Don't have an account? Sign up."/>
</LinearLayout>
```

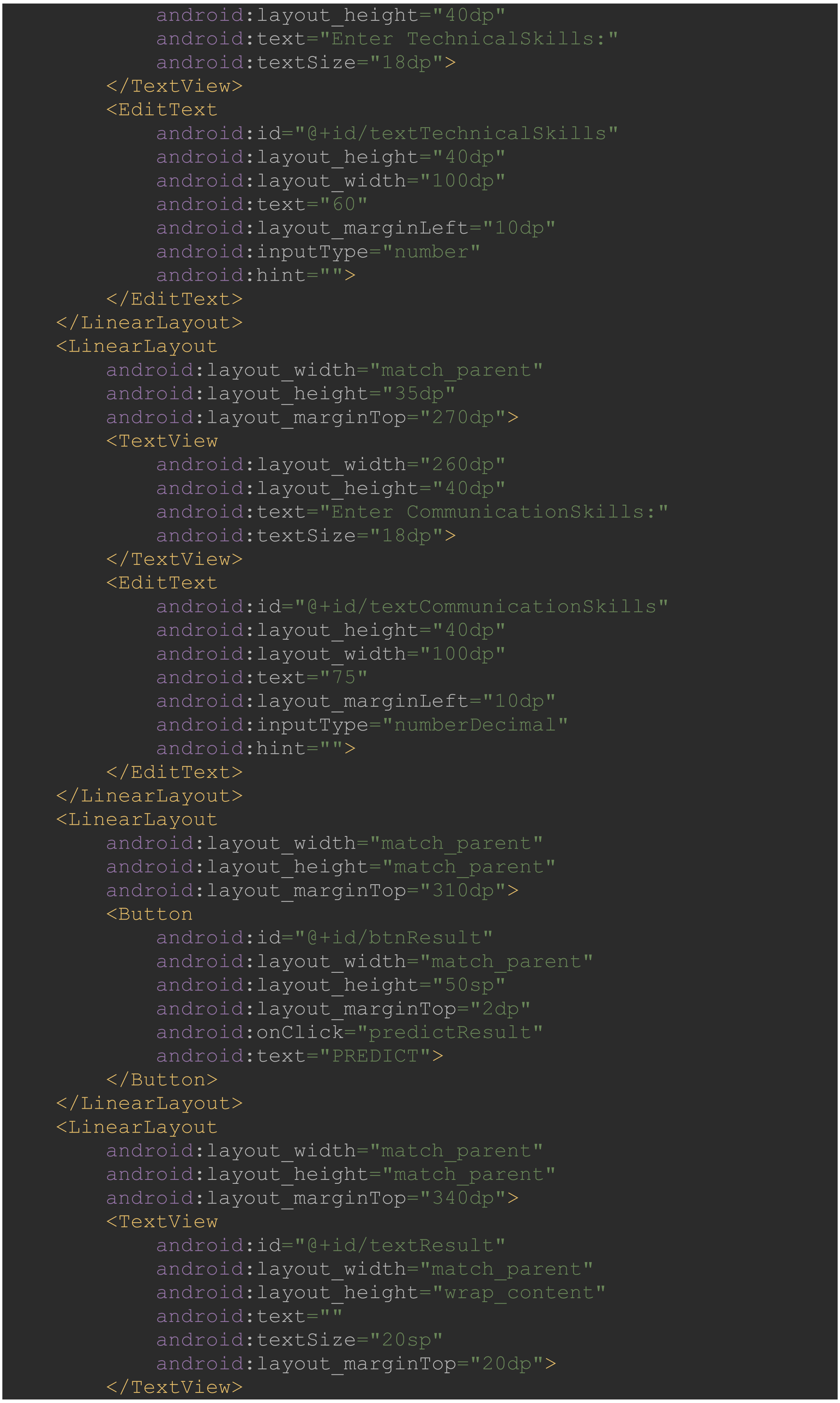
LoginActivity.java

```
package com.example.placement_predict;

import androidx.appcompat.app.AppCompatActivity;

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

import com.google.android.material.button.MaterialButton;

public class LoginActivity extends AppCompatActivity {



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        TextView userName =(TextView)
findViewById(R.id.editTextEmail);
        TextView password =(TextView)
findViewById(R.id.editTextPassword);

        MaterialButton btnSubmit = (MaterialButton)
findViewById(R.id.buttonLogin);

        //admin and admin

        btnSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if
(userName.getText().toString().equals("2011cs010266@mallareddyuniv
ersity.ac.in") &&
password.getText().toString().equals("lavan@143")) {
                    //correct
                    Intent intent = new Intent(LoginActivity.this,
MainActivity.class);

                    startActivity(intent);
                } else
                    //incorrect
                    Toast.makeText(LoginActivity.this, "LOGIN
FAILED !!!", Toast.LENGTH_SHORT).show();
            }


        });
    }
    }
```

fragment_home.xml



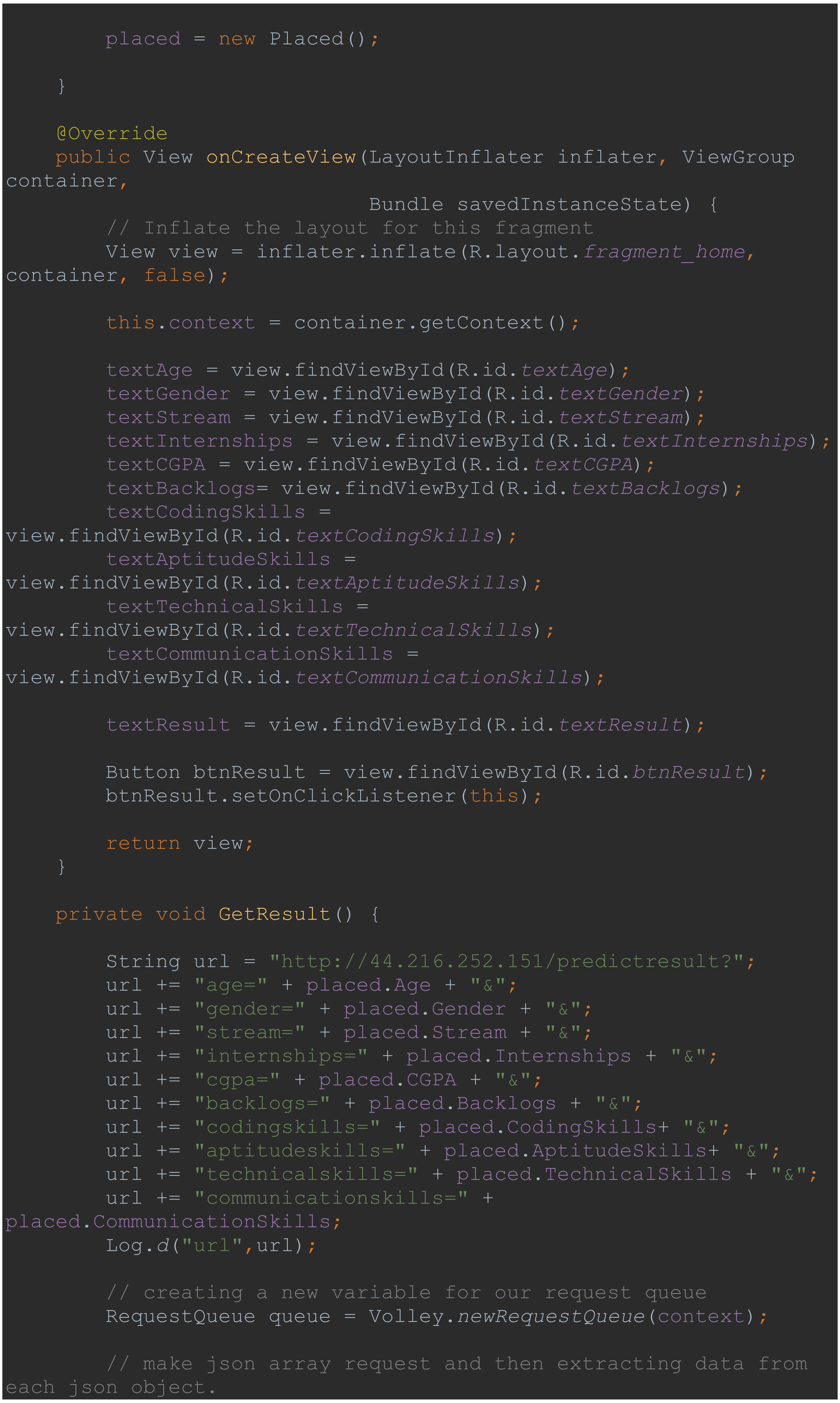
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    tools:context=".HomeFragment">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="20dp"
            android:text="Enter Age:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textAge"
            android:layout_height="40dp"
            android:text="22"
            android:layout_width="100dp"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="30dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Gender:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textGender"
            android:layout_height="40dp"
            android:text="1"
            android:layout_width="100dp"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="60dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Stream:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textStream"
            android:layout_height="40dp"
            android:text="2"
            android:layout_width="100dp"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="90dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Internships:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textInternships"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="1"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="120dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter CGPA:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textCGPA"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="8"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="150dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter Backlogs:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textBacklogs"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="1"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="180dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter CodingSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textCodingSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="80"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="210dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter AptitudeSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textAptitudeSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="50"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="240dp">
        <TextView
            android:layout_width="260dp"
```



```
            android:layout_height="40dp"
            android:text="Enter TechnicalSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textTechnicalSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="60"
            android:layout_marginLeft="10dp"
            android:inputType="number"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="35dp"
        android:layout_marginTop="270dp">
        <TextView
            android:layout_width="260dp"
            android:layout_height="40dp"
            android:text="Enter CommunicationSkills:"
            android:textSize="18dp">
        </TextView>
        <EditText
            android:id="@+id/textCommunicationSkills"
            android:layout_height="40dp"
            android:layout_width="100dp"
            android:text="75"
            android:layout_marginLeft="10dp"
            android:inputType="numberDecimal"
            android:hint="">
        </EditText>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="310dp">
        <Button
            android:id="@+id/btnResult"
            android:layout_width="match_parent"
            android:layout_height="50sp"
            android:layout_marginTop="2dp"
            android:onClick="predictResult"
            android:text="PREDICT">
        </Button>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="340dp">
        <TextView
            android:id="@+id/textResult"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text=""
            android:textSize="20sp"
            android:layout_marginTop="20dp">
        </TextView>
    </LinearLayout>

</FrameLayout>
```

HomeFragment.java

```
package com.example.placement_predict;
import android.content.Context;
import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class HomeFragment extends Fragment implements
View.OnClickListener {

    Context context;
    Placed placed;
    TextView textResult;
    EditText
textAge,textGender,textStream,textInternships,textCGPA,textBacklog
s,textCodingSkills,textAptitudeSkills,textTechnicalSkills,textComm
unicationSkills;
    public HomeFragment() {
        // Required empty public constructor
    }

    public static HomeFragment newInstance(String param1, String
param2) {
        HomeFragment fragment = new HomeFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getActivity().setTitle("CHECK YOU ARE TO BE  PLACED OR
NOT");
```



```
        placed = new Placed();

    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_home,
container, false);

        this.context = container.getContext();

        textAge = view.findViewById(R.id.textAge);
        textGender = view.findViewById(R.id.textGender);
        textStream = view.findViewById(R.id.textStream);
        textInternships = view.findViewById(R.id.textInternships);
        textCGPA = view.findViewById(R.id.textCGPA);
        textBacklogs= view.findViewById(R.id.textBacklogs);
        textCodingSkills =
view.findViewById(R.id.textCodingSkills);
        textAptitudeSkills =
view.findViewById(R.id.textAptitudeSkills);
        textTechnicalSkills =
view.findViewById(R.id.textTechnicalSkills);
        textCommunicationSkills =
view.findViewById(R.id.textCommunicationSkills);

        textResult = view.findViewById(R.id.textResult);

        Button btnResult = view.findViewById(R.id.btnResult);
        btnResult.setOnClickListener(this);

        return view;
    }

    private void GetResult() {

        String url = "http://44.216.252.151/predictresult?";
        url += "age=" + placed.Age + "&";
        url += "gender=" + placed.Gender + "&";
        url += "stream=" + placed.Stream + "&";
        url += "internships=" + placed.Internships + "&";
        url += "cgpa=" + placed.CGPA + "&";
        url += "backlogs=" + placed.Backlogs + "&";
        url += "codingskills=" + placed.CodingSkills+ "&";
        url += "aptitudeskills=" + placed.AptitudeSkills+ "&";
        url += "technicalskills=" + placed.TechnicalSkills + "&";
        url += "communicationskills=" +
placed.CommunicationSkills;
        Log.d("url",url);

        // creating a new variable for our request queue
        RequestQueue queue = Volley.newRequestQueue(context);

        // make json array request and then extracting data from
each json object.
```

```
        JsonArrayRequest jsonArrayRequest = new
JsonArrayRequest(Request.Method.GET, url, null, new
Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                try {
                    JSONObject responseObj =
response.getJSONObject(0);

                    String returnValue =
responseObj.getString("Result");
//                    textResult.setText("Result: " +
returnValue);
                    if(returnValue.equals("1")){
                        textResult.setText("You are Placed");
                    }else{
                        textResult.setText("You are not to be
placed. Improve your skills");
                    }

                    Toast.makeText(context, returnValue,
Toast.LENGTH_SHORT).show();

                } catch (JSONException e) {
                    throw new RuntimeException(e);
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(context, error.toString(),
Toast.LENGTH_SHORT).show();
                Log.d("Error", error.toString());

            }
        });
        queue.add(jsonArrayRequest);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.btnResult:
                placed.Age =  textAge.getText().toString();
                placed.Gender =  textGender.getText().toString();
                placed.Stream =  textStream.getText().toString();
                placed.Internships =
textInternships.getText().toString();
                placed.Backlogs =
textBacklogs.getText().toString();
                placed.CGPA =  textCGPA.getText().toString();
                placed.CodingSkills =
textCodingSkills.getText().toString();
                placed.AptitudeSkills =
textAptitudeSkills.getText().toString();
                placed.TechnicalSkills =
textTechnicalSkills.getText().toString();
                placed.CommunicationSkills =
textCommunicationSkills.getText().toString();
```

```
                GetResult();
                textResult.setText("wait..");
                break;
        }
    }
}
```

fragment_tests.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".TestsFragment">

    <!-- TODO: Update blank fragment layout -->
    <WebView
        android:id="@+id/webView"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </WebView>


</FrameLayout>
```

TestsFragment.java

```
package com.example.placement_predict;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;


public class TestsFragment extends Fragment {

    public TestsFragment() {
        // Required empty public constructor
    }

    public static TestsFragment newInstance(String param1, String
param2) {
        TestsFragment fragment = new TestsFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getActivity().setTitle("Take Tests");
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_details,
container, false);

        // Find the WebView by its unique ID
        WebView webView = view.findViewById(R.id.webView);

        webView.loadUrl("https://www.indiabix.com/online-
test/categories/");
        webView.getSettings().setJavaScriptEnabled(true);
        webView.setWebViewClient(new WebViewClient());

        return view;
    }
}
```

fragment_details.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    tools:context=".DetailsFragment">


    <WebView
        android:id="@+id/webView"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </WebView>

</FrameLayout>
```

DetailsFragment.java

```
package com.example.placement_predict;

import android.os.Bundle;
import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;


public class DetailsFragment extends Fragment {

    public DetailsFragment() {
        // Required empty public constructor
    }

    public static DetailsFragment newInstance(String param1,
String param2) {
        DetailsFragment fragment = new DetailsFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getActivity().setTitle("Team Details");
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_details,
container, false);

        // Find the WebView by its unique ID
        WebView webView = view.findViewById(R.id.webView);

        webView.loadUrl("http://placement-analysis.s3-website-us-
east-1.amazonaws.com/");
        webView.getSettings().setJavaScriptEnabled(true);
        webView.setWebViewClient(new WebViewClient());

        return view;
    }
}
```

fragment_about.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".AboutFragment">
    <!-- TODO: Update blank fragment layout -->
    <WebView
        android:id="@+id/webView"
        android:layout_margin="10dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </WebView>


</FrameLayout>
```

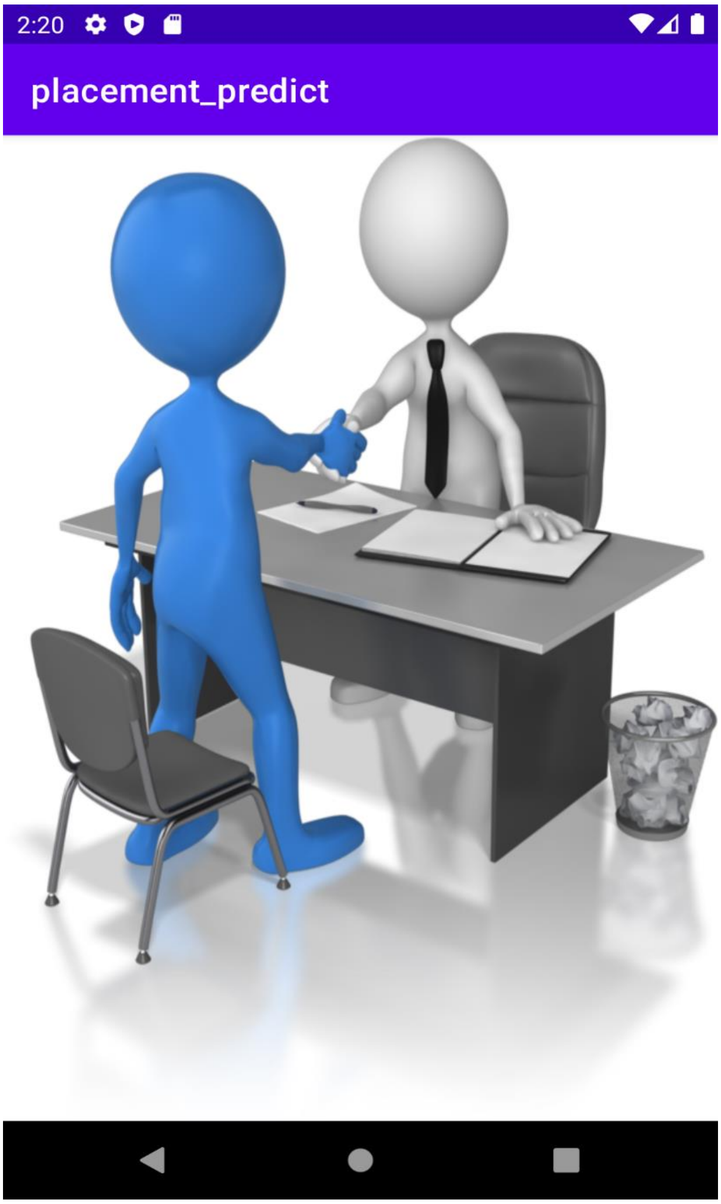
AboutFragment.java

```
package com.example.placement_predict;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebView;
import android.webkit.WebViewClient;


public class AboutFragment extends Fragment {

    public AboutFragment() {
        // Required empty public constructor
    }

    public static AboutFragment newInstance(String param1, String
param2) {
        AboutFragment fragment = new AboutFragment();
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getActivity().setTitle("About the project");
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_details,
container, false);

        // Find the WebView by its unique ID
        WebView webView = view.findViewById(R.id.webView);

        webView.loadUrl("http://aboutproject.s3-website-us-east-
1.amazonaws.com/");
```

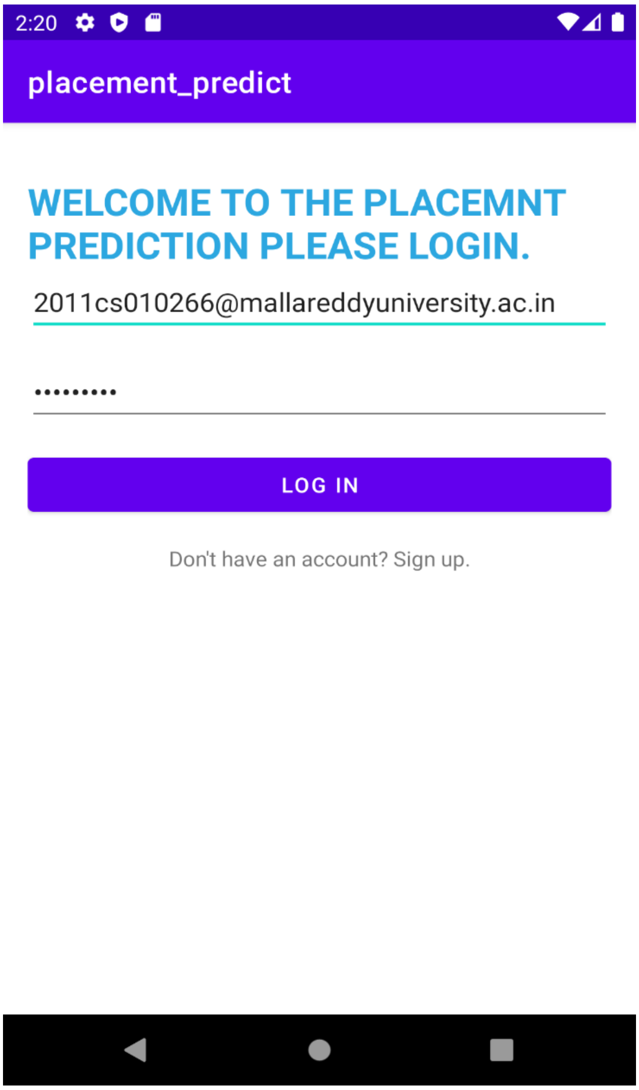
```
        webView.getSettings().setJavaScriptEnabled(true);
        webView.setWebViewClient(new WebViewClient());

        return view;
    }
}
```

5.2. Screenshot of Application

5.2.1. Splash Screen for the application.



5.2.2. Login Screen



2:20

placement_predict

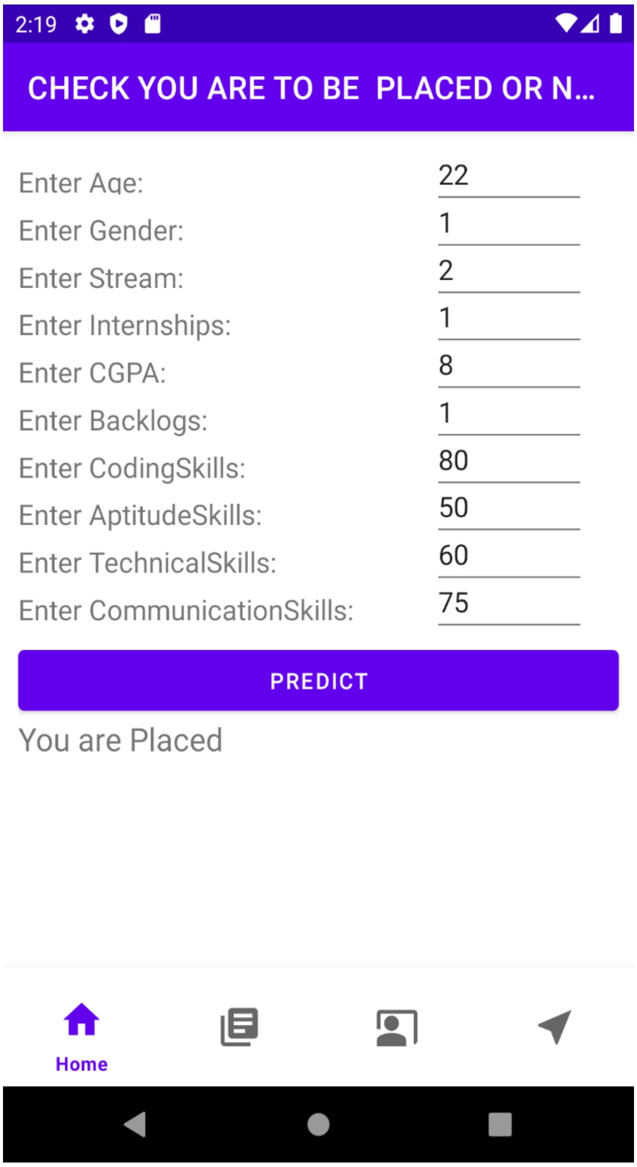
WELCOME TO THE PLACEMNT PREDICTION PLEASE LOGIN.

2011cs010266@mallareddyuniversity.ac.in

•••••••••

Don't have an account? Sign up.

5.2.3. Home page screen



CHECK YOU ARE TO BE PLACED OR N...

Enter Age: 22

Enter Gender: 1

Enter Stream: 2

Enter Internships: 1

Enter CGPA: 8

Enter Backlogs: 1

Enter CodingSkills: 80

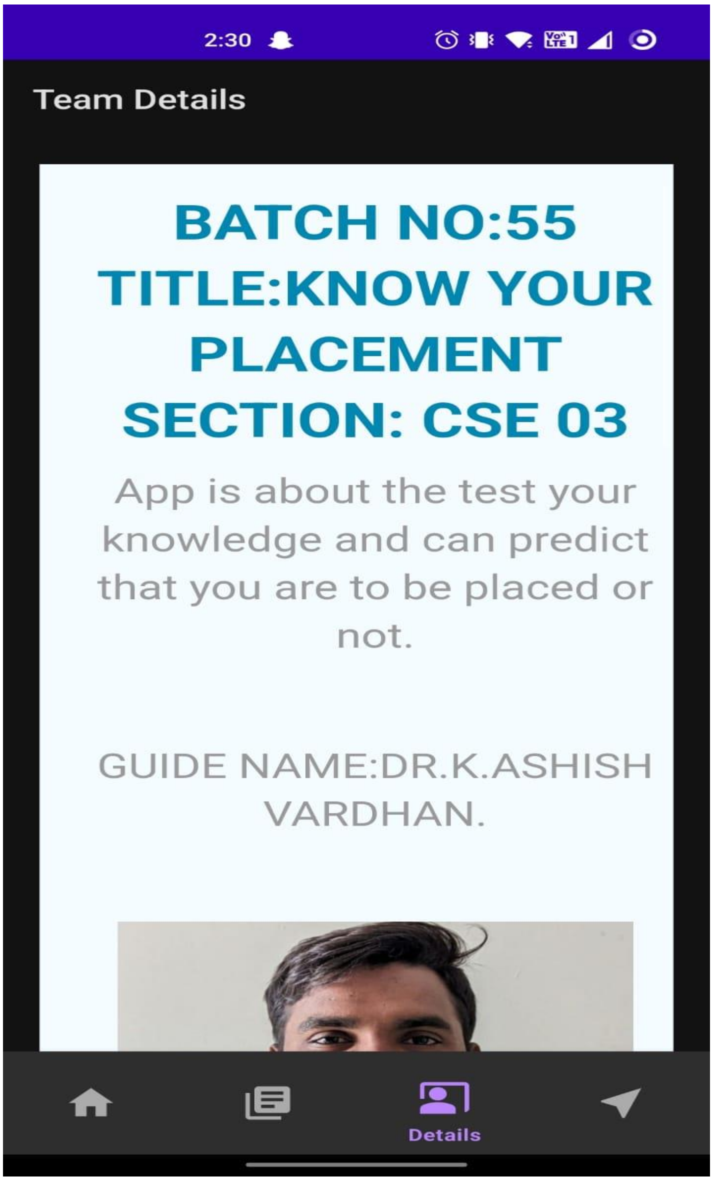
Enter AptitudeSkills: 50

Enter TechnicalSkills: 60

Enter CommunicationSkills: 75

You are Placed

5.2.4. Details page



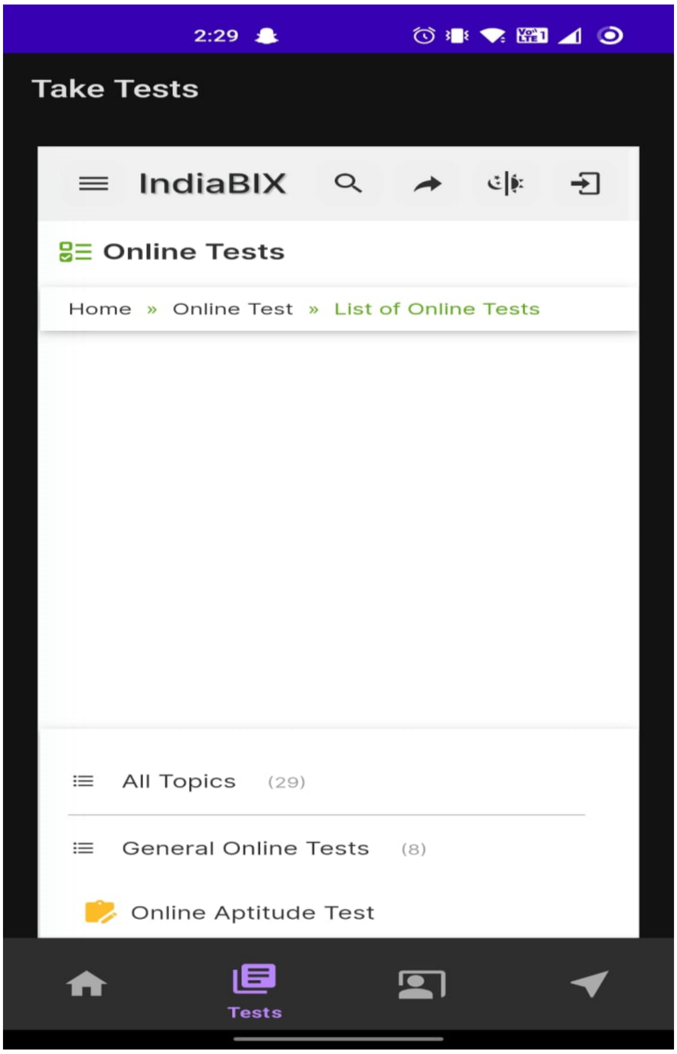
Team Details

BATCH NO:55
TITLE:KNOW YOUR PLACEMENT
SECTION: CSE 03

App is about the test your knowledge and can predict that you are to be placed or not.

GUIDE NAME:DR.K.ASHISH VARDHAN.

5.2.5. Tests Page



2:29

Take Tests

IndiaBIX

Online Tests

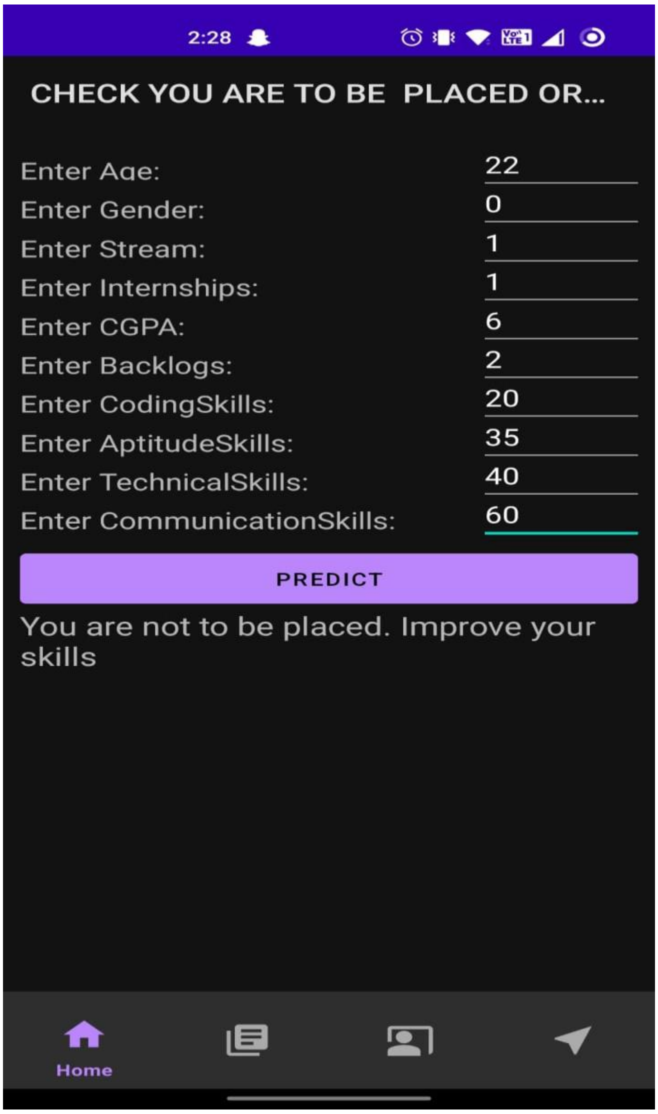
Home » Online Test » List of Online Tests

All Topics (29)

General Online Tests (8)

Online Aptitude Test

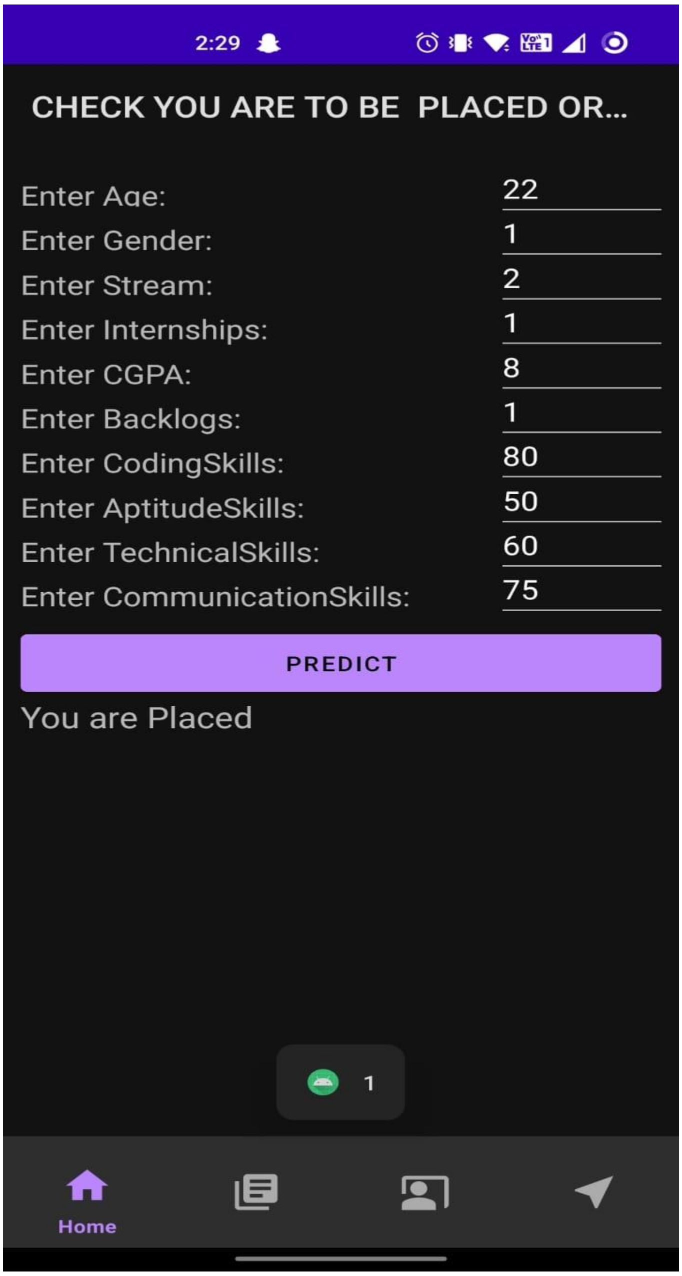
5.2.6. output predictions



CHECK YOU ARE TO BE PLACED OR...

Enter Age:	22
Enter Gender:	0
Enter Stream:	1
Enter Internships:	1
Enter CGPA:	6
Enter Backlogs:	2
Enter CodingSkills:	20
Enter AptitudeSkills:	35
Enter TechnicalSkills:	40
Enter CommunicationSkills:	60

You are not to be placed. Improve your skills



CHECK YOU ARE TO BE PLACED OR...

Enter Age: 22

Enter Gender: 1

Enter Stream: 2

Enter Internships: 1

Enter CGPA: 8

Enter Backlogs: 1

Enter CodingSkills: 80

Enter AptitudeSkills: 50

Enter TechnicalSkills: 60

Enter CommunicationSkills: 75

You are Placed

Chapter 6

Results & Conclusion

6.1 Results

We have created a machine learning model that can learn from historical data to understand the relationships between the features and placement prediction. This model should then be able to generalize and make accurate predictions on new, unseen data. The objective is to create a machine learning model that can learn from historical data to understand the relationships between these features and to find the placement of an student. After successful prediction, the accuracy was almost 86% where it is good to predicate the data and these are integrated into Android for user-friendly purposes so that the higher authorities of an organization can find the particular student placement to be placed or not.

6.2 Conclusion

The conclusion of placement prediction using Android will depend on the accuracy and performance of the underlying machine learning model integrated into the Android application. Here are some expected outcomes:

1) Real-Time Predictions

2) Prediction Accuracy

3) User Interface

4) User Feedback and Satisfaction

References

1. Pandey, A., & Singh, V. K. (2019). Job Placement Prediction Using Machine Learning Techniques. 2019 International Conference on Communication and Electronics Systems (ICCES). DOI: 10.1109/CESYS.2019.8706309

2. Palaniappan, B., & Mohan, S. (2020). A Machine Learning Approach for Job Placement Prediction. 2020 IEEE International Conference on Inventive Computation Technologies (ICICT). DOI: 10.1109/ICICT48040.2020.9198416

3. Kim, J., & Choi, K. (2019). Job Placement Prediction Model Based on Machine Learning. Proceedings of the 52nd Hawaii International Conference on System Sciences. DOI: 10.24251/hicss.2019.704.