

Visvesvaraya Technological University Belagavi, Karnataka-590 018



A MINI PROJECT REPORT

On

JOB PORTAL APPLICATION

Submitted

in partial fulfilment requirements for the credit of the Course

on

MOBILE APPLICATION DEVELOPMENT

18CSMP68

by

**SHREELAXMI S GANGOLLI
SWATI SAWANT**

**4CB19CS099
4CB19CS111**

Under the Guidance of
Mr. Janardhana Bhat K

Assistant Professor



**Department of Computer Science and Engineering
Canara Engineering College, Benjanapadavu
2021-2022**

CANARA ENGINEERING COLLEGE

BENJANAPADAVU, BANTWAL- 574219 – KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that **SHREELAXMI S GANGOLLI(4CB19CS099)** and **SWATI SAWANT(4CB19CS111)** have successfully completed the mini project work on '**Job Portal Application**' and submitted in partial fulfillment of the requirements of the Course on **Mobile Application Development (18CSMP68)** prescribed by the **Visvesvaraya Technological University** during the academic year **2021-2022**. It is verified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed by Bachelor of Engineering Degree.

Course Instructor
Mr. Janardhana Bhat K
Assistant Professor

H.O.D
Dr. Demian Antony D' Mello
Professor

Name of the Examiners

Signature with Date

1.

2.

ACKNOWLEDGEMENT

We are indebted to our **Principal, Dr. Ganesh V. Bhat** and management of Canara Engineering College for providing an environment with all facilities that helped us in completing our mini project.

We are extremely grateful to **Dr. Demian Antony D' Mello, Head of Computer Science & Engineering Department** for his moral support and encouragement.

We wish to express our sincere gratitude to our course instructor, **Mr. Janardhana Bhat K, Assistant Professor, Computer Science & Engineering Department**, for the guidance and suggestions.

We thank all the teaching and non-teaching staff of the department for their kind help in completing this mini project.

Last but not the least; we would like to add some personal note. If there is a driving force that kept us going, and what has not changed it is the constant support and blessing of our parents, family and friends. There is no doubt, in spite of my strenuous efforts error might remain in mini project. Naturally, we alone take full responsibility for any lack of clarity, occasional erratum or inexactness that may occur.

SHREELAXMI S GANGOLLI 4CB19CS099

SWATI SAWANT 4CB19CS111

ABSTRACT

The purpose of this study is to help the applicants know about the available job opportunities. Due to the advancement in technology, businesses no longer advertise job openings in newspapers or on leaflets. Rather than that, they focus on digital tools such as work hunt smartphone apps or online portals to attract the best candidates. On the other side, work hunters are freed of the burden of racing from one employer to the next in search of a place to drop their application. One of the most significant benefits of utilizing work portals is that they have access to an infinite number of job openings. Scope includes, the applicants can quickly browse through a variety of job openings to locate one that meets their criteria.

TABLE OF CONTENTS

CONTENTS	PAGE NO.
1. INTRODUCTION	
1.1 Introduction to Android Application	01
1.2 Introduction to Android Studio	02
1.3 Mini Project Description	03
2. REQUIREMENT SPECIFICATION	
2.1 Hardware Requirements	04
2.2 Software Requirements	04
3. DESIGN	
3.1 Architectural Diagram/Flowchart	05
3.2 XML Code	06
3.3 Database Schema diagram	29
4. IMPLEMENTATION DETAILS	
4.1 List of classes imported.	30
4.2 List of important library functions used.	32
4.3 Java Code	34
4.4 Manifest file code	60
5. RESULTS	
5.1 Sample Output (Screenshots)	62
6. CONCLUSION & FUTURE WORK	71
REFERENCES	72

CHAPTER 1

INTRODUCTION

1.1 Introduction to Android Application:

The Android Application is the largest installed base among various mobile platforms across the globe. Hundreds of millions of mobile devices are powered by Android in more than 190 countries of the world. Android Application is a software designed to run on an Android device or emulator. The term also refers to an APK file which stands for Android package. Android apps can be written in Kotlin, Java, and C++ and are run inside Virtual Machine. The official development environment is Android Studio. The compiled code along with any data and resource files required by the application is bundled by the apt tool into an Android package, an archive file marked by an .apk suffix. This file is the vehicle for distributing the application and installing it on mobile devices; it's the file users download to their devices. All the code in a single .apk file is considered to be one application. By default, every application runs in its own Linux process, with a unique Linux user id. Each process has its own virtual machine (VM).

1.1.1 Android Application Components:

- **Activities:** The application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that displays information and respond to user actions. In terms of desktop development, an Activity is equivalent to a Form.
- **Services:** Services are the invisible workers of your application. Service components run invisibly, updating your data sources and visible Activities and triggering Notifications.
- **Content Providers:** Are a shareable data store. Content Providers are used to manage and share application databases. Content Providers are the preferred way of sharing data across application boundaries
- **Intents:** They are simple message-passing framework. Using Intents, you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed.
- **Broadcast Receivers:** By creating and registering a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter criteria.

- **Notifications:** They are user notification framework. Notifications let you signal users without stealing focus or interrupting their current Activities. They are the preferred technique for getting a user's attention from within a Service or Broadcast Receiver

1.2 Introduction to Android Studio:

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools. To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and GitHub integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

1.2.1 Features:

Android Studio is having many cool features which can foster and could help you to increase development productivity. Here are the cool features:

- Powerful code editing (smart editing, code re-factoring)
- Rich layout Editor (As soon as you drag and drop views on the layout, it shows you preview in all the screens including Nexus 4, Nexus 7, Nexus 10 and many other resolutions. Layout designing can be done much faster way as compared to eclipse.)
- Gradle-based build support
- Maven Support
- Templated based wizards
- Lint tool analysis (The Android lint tool is a static code analysis tool that checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization).
- It provides ability to capture directly generate a screenshot of your application. Yes it was already included in the SDK but Android Studio provides something more:

- Device frame (As frames for many Nexus devices are available, you can capture screenshot in whichever frame you like most)
 - Drop shadow
 - Screen glare

1.3 Miniproject Description:

The Internet provides applicants with a new medium of obtaining useful information and for searching jobs meeting their criteria. The main purpose of job portal app is to build a platform in favor of applicants to know the available jobs. And for the company's to promote jobs. Job portal app provides standard job alerts whenever there are job openings. Job portals offer a full collection of job choices from top companies. It means one can have a superior chance of searching the job they want anywhere and however; they want it.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

- Processor: Intel core i3 and i5
- RAM: 8GB (Minimum 4GB)

2.2 Software Requirements

- Operating System: Windows 10
- Tool/IDE (with version) used: Android Studio 4.1.3
- Emulator Used: Pixel 3a API 30
- Language used: Java
- Database used: Firebase

CHAPTER 3

DESIGN

3.1 Architectural Diagram of the Application

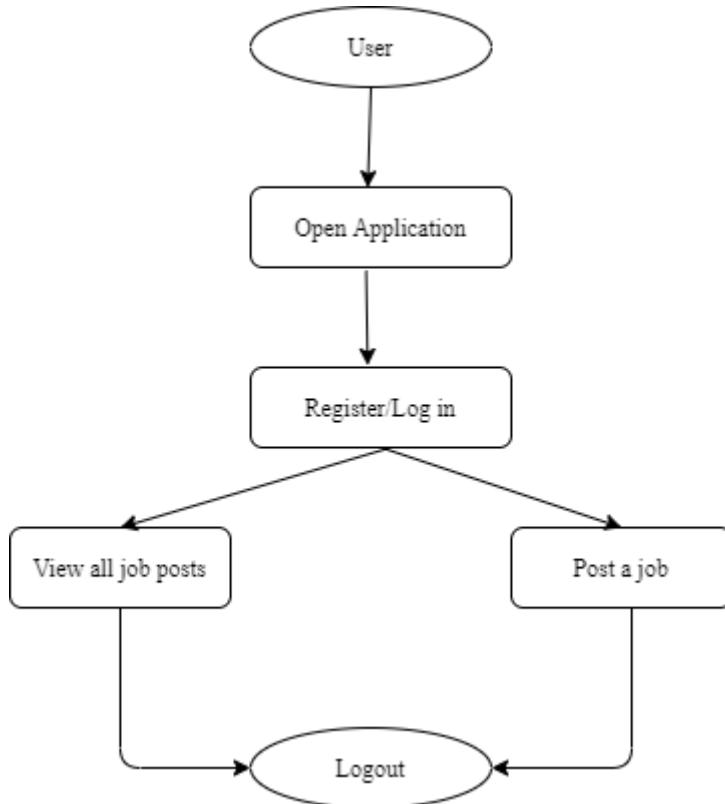


Fig (3.1): Architectural diagram

Explanation:

- In our application, the user needs to register himself first by entering his/her email and password. If he/she has an account, can directly log in.
- Then the user has to login by entering the email and password.
- The user can opt to either view available jobs or promote jobs available.
- If user chooses to promote job(s), then he/she can post available jobs under post a job button.

- If user chooses to view available jobs the he can do so under view all posts button.
- Then the user can logout otherwise.

3.2 XML Code

3.2.1 activity_registration.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/logojob"
    android:layout_height="match_parent"
    tools:context=".RegistrationActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

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

            <TextView
                android:id="@+id/textView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
```

```
    android:text="Registration"  
    android:textAppearance="?android:textAppearanceLarge"  
    android:textColor="@android:color/white" />
```

```
<EditText  
    android:id="@+id/email_registration"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:background="@drawable/edit_design"  
    android:hint="Email:"  
    android:padding="10dp"  
    android:textColor="@android:color/white"  
    android:textColorHint="@android:color/white" />
```

```
<EditText  
    android:id="@+id/registration_password"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"  
    android:background="@drawable/edit_design"  
    android:hint="Password:"  
    android:inputType="textPassword"  
    android:padding="10dp"  
    android:textColor="@android:color/white"  
    android:textColorHint="@android:color/white" />
```

```
<Button  
    android:id="@+id/btn_registration"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="10dp"
```

```
    android:background="@drawable/button_design"
    android:padding="15dp"
    android:textColor="@android:color/white"
    android:text="Registration"
    android:textAppearance="?android:textAppearanceMedium" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:padding="10dp"
    android:text="OR"
    android:textAppearance="?android:textAppearanceLarge"
    android:textColor="@android:color/white" />

<Button
    android:id="@+id	btn_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/button_design"
    android:padding="15dp"
    android:text="Log In"
    android:textColor="@android:color/white"
    android:textAppearance="?android:textAppearanceMedium" />

</LinearLayout>

</ScrollView>

</LinearLayout>
```

3.2.2 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:layout_gravity="center"
    android:gravity="center"
    android:background="@drawable/logojob"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

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

            <TextView
                android:id="@+id/textView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="Login"
                android:textAppearance="?android:textAppearanceLarge"
                android:textColor="@android:color/white" />

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:imeOptions="actionNext"
                android:inputType="textPersonName"
                android:padding="10dp" />
    
```

```
    android:id="@+id/email_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/edit_design"
    android:hint="Email:"
    android:padding="10dp"
    android:textColor="@android:color/white"
    android:textColorHint="@android:color/white" />
```

```
<EditText
    android:id="@+id/login_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/edit_design"
    android:hint="Password:"
    android:padding="10dp"
    android:textColor="@android:color/white"
    android:textColorHint="@android:color/white" />
```

```
<Button
    android:id="@+id	btn_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/button_design"
    android:padding="15dp"
    android:textColor="@android:color/white"
    android:text="Log In"
    android:textAppearance="?android:textAppearanceMedium" />
```

```
<TextView
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:padding="10dp"
    android:text="OR"
    android:textAppearance="?android:textAppearanceLarge"
    android:textColor="@android:color/white" />

<Button
    android:id="@+id	btn_reg"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/button_design"
    android:padding="15dp"
    android:text="Registration"
    android:textColor="@android:color/white"
    android:textAppearance="?android:textAppearanceMedium" />

</LinearLayout>

</ScrollView>
</LinearLayout>
```

3.2.3 activity_home.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:background="@drawable/logojob"
    android:layout_height="match_parent"
```

```
tools:context=".HomeActivity">

<include layout="@layout/toolbar"
    android:id="@+id/toolbar_home"/>

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

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="View all Posts"
        android:background="@drawable/button_design"
        android:textAppearance="?android:textAppearanceMedium"
        android:textColor="@android:color/white"
        android:id="@+id	btn_allJob"
    />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Post A Job"
        android:background="@drawable/button_design"
        android:textAppearance="?android:textAppearanceMedium"
        android:textColor="@android:color/white"
        android:id="@+id	btn_postJob"
    />

</LinearLayout>
```

</LinearLayout>

3.2.4 activity_post_job.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".PostJobActivity">

    <include layout="@layout/toolbar"
        android:id="@+id/toolbar_post_job" />

    <androidx.coordinatorlayout.widget.CoordinatorLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

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

            <androidx.recyclerview.widget.RecyclerView
                android:layout_width="match_parent"
                android:id="@+id/recycler_job_post_id"

                android:layout_height="match_parent">

            </androidx.recyclerview.widget.RecyclerView>
    
```

```
</LinearLayout>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:layout_width="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="20dp"
    android:layout_marginBottom="25dp"
    android:layout_marginRight="25dp"
    app:fabSize="normal"
    android:src="@drawable/add"
    android:id="@+id/fab_add"
    android:backgroundTint="@color/fabcolor"
    android:layout_height="wrap_content"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
</LinearLayout>
```

3.2.5 job_post_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="5dp"
    android:elevation="5dp"
    android:layout_margin="10dp"
    android:layout_height="wrap_content">

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

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Job Title"  
    android:paddingStart="12dp"  
    android:paddingEnd="12dp"  
    android:maxLines="1"  
    android:textAppearance="?android:textAppearanceLarge"  
    android:padding="10dp"  
    android:textColor="@android:color/black"  
    android:id="@+id/job_title"  
/>
```

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="1 November, 2020"  
    android:paddingStart="12dp"  
    android:paddingEnd="12dp"  
    android:textAppearance="?android:textAppearanceSmall"  
    android:padding="10dp"  
    android:textColor="@android:color/black"  
    android:id="@+id/job_date"/>
```

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Description"  
    android:paddingStart="12dp"  
    android:paddingEnd="12dp"  
    android:textAppearance="?android:textAppearanceMedium"
```

```
    android:padding="10dp"
    android:textColor="@android:color/black"
    android:id="@+id/job_description"
    android:maxLines="3"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Skills"
    android:paddingStart="12dp"
    android:paddingEnd="12dp"
    android:padding="10dp"
    android:textAppearance="?android:textAppearanceMedium"
    android:maxLines="1"
    android:textColor="@android:color/black"
    android:id="@+id/job_skills"
    />

<TextView
    android:layout_width="match_parent"
    android:text="Salary"
    android:paddingStart="12dp"
    android:paddingEnd="12dp"
    android:textAppearance="?android:textAppearanceMedium"
    android:textColor="@android:color/black"
    android:padding="10dp"
    android:id="@+id/job_salary"
    android:layout_height="wrap_content"/>

</LinearLayout>
</androidx.cardview.widget.CardView>
```

3.2.6 alljobpost.xml

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

    app:cardElevation="5dp"
    android:elevation="5dp"
    android:layout_margin="10dp"
    android:layout_height="wrap_content">

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

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Job Title"
            android:textAppearance="?android:textAppearanceLarge"
            android:padding="5dp"
            android:layout_margin="5dp"
            android:maxLines="1"
            android:textColor="@android:color/black"
            android:id="@+id/all_job_post_title"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="26 July 2021"
```

```
    android:textAppearance="?android:textAppearanceSmall"
    android:padding="5dp"
    android:layout_margin="5dp"
    android:textColor="#0a51bc"
    android:id="@+id/all_job_post_date"/>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Description"
    android:textAppearance="?android:textAppearanceLarge"
    android:padding="5dp"
    android:layout_margin="5dp"
    android:textColor="@android:color/black"
    android:maxLines="3"
    android:id="@+id/all_job_post_description"/>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Skills"
    android:textAppearance="?android:textAppearanceLarge"
    android:padding="5dp"
    android:layout_margin="5dp"
    android:textColor="@android:color/black"
    android:maxLines="1"
    android:id="@+id/all_job_post_skills"/>
```

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Salary"
    android:textAppearance="?android:textAppearanceLarge"
```

```
    android:padding="5dp"
    android:layout_margin="5dp"
    android:textColor="@android:color/black"
    android:maxLines="3"
    android:id="@+id/all_job_post_salary"/>
</LinearLayout>

</androidx.cardview.widget.CardView>
```

3.2.7 activity_insert_job_post.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    android:layout_height="match_parent"
    tools:context=".InsertJobPostActivity">

    <include layout="@layout/toolbar"
        android:id="@+id/insert_job_toolbar"/>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

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

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

```
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RelativeLayout
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="@android:color/black"
            android:text="Job Title:"
            android:padding="7dp"
            android:textAppearance="?android:textAppearanceMedium"/>

        </RelativeLayout>

        <RelativeLayout
            android:layout_width="0dp"
            android:layout_weight="3"
            android:layout_height="wrap_content">

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/job_title"/>

        </RelativeLayout>

    </LinearLayout>

    <LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
<RelativeLayout
    android:layout_width="0dp"
    android:layout_weight="2"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Job Description:"
        android:textColor="@android:color/black"
        android:padding="7dp"
        android:textAppearance="?android:textAppearanceMedium"/>

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="0dp"
        android:layout_weight="3"
        android:layout_height="wrap_content">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/job_description"/>

    </RelativeLayout>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RelativeLayout
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Skills:"
            android:textColor="@android:color/black"
            android:padding="7dp"
            android:textAppearance="?android:textAppearanceMedium"/>

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="0dp"
        android:layout_weight="3"
        android:layout_height="wrap_content">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/job_skill"/>

    </RelativeLayout>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"

        android:orientation="horizontal">

    <RelativeLayout
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Salary:"
            android:textColor="@android:color/black"
            android:padding="7dp"
            android:textAppearance="?android:textAppearanceMedium"/>

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="0dp"
        android:layout_weight="3"
        android:layout_height="wrap_content">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/job_salary"/>

    </RelativeLayout>

</LinearLayout>
<Button
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:text="Post job"
    android:textColor="@android:color/white"
    android:textAppearance="?android:textAppearanceMedium"
    android:id="@+id(btn_job_post"
    android:background="@drawable/job_post_button"/>

</LinearLayout>
</ScrollView>

</LinearLayout>
```

3.2.8 alljobpost.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:cardElevation="5dp"
    android:elevation="5dp"
    android:layout_margin="10dp"
    android:layout_height="wrap_content">

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

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

```
    android:text="Job Title"  
        android:paddingStart="12dp"  
        android:paddingEnd="12dp"  
        android:maxLines="1"  
        android:textAppearance="?android:textAppearanceLarge"  
        android:padding="10dp"  
        android:textColor="@android:color/black"  
        android:id="@+id/job_title"  
    />
```

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="1 November, 2020"  
    android:paddingStart="12dp"  
    android:paddingEnd="12dp"  
    android:textAppearance="?android:textAppearanceSmall"  
    android:padding="10dp"  
    android:textColor="@android:color/black"  
    android:id="@+id/job_date"/>
```

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Description"  
    android:paddingStart="12dp"  
    android:paddingEnd="12dp"  
    android:textAppearance="?android:textAppearanceMedium"  
    android:padding="10dp"  
    android:textColor="@android:color/black"  
    android:id="@+id/job_description"  
    android:maxLines="3"/>
```

```
<TextView
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Skills"
        android:paddingStart="12dp"
        android:paddingEnd="12dp"
        android:padding="10dp"
        android:textAppearance="?android:textAppearanceMedium"
        android:maxLines="1"
        android:textColor="@android:color/black"
        android:id="@+id/job_skills"
    />

<TextView
    android:layout_width="match_parent"
    android:text="Salary"
    android:paddingStart="12dp"
    android:paddingEnd="12dp"
    android:textAppearance="?android:textAppearanceMedium"
    android:textColor="@android:color/black"
    android:padding="10dp"
    android:id="@+id/job_salary"
    android:layout_height="wrap_content"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

```

3.2.9 activity_all_job.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"

```

```

    android:layout_height="match_parent"
    tools:context=".AllJobActivity">

<include layout="@layout/toolbar"
    android:id="@+id/all_job_post"/>

<androidx.recyclerview.widget.RecyclerView

    android:layout_width="match_parent"
    android:id="@+id/recycler_all_job"
    android:layout_height="match_parent
    "/>

</LinearLayout>

```

3.2.10 activity_job_details.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout

    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:orientation="vertical"
    android:layout_height="match_parent"
    tools:context=".JobDetailsActivity">
<include layout="@layout/toolbar"
    android:id="@+id/toolbar_job_de
tails"/>

<TextView

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:padding ="5dp"
    android:textAppearance ="?android:textAppearanceLarge"
    android:text ="Job Title..."
    android:id ="@+id/job_details_title"
    />
<TextView

```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:padding="5dp"
    android:text="2 August 2021"
    android:textAppearance="?android:textAppearanceSmall"
    android:textColor="#FF2F1842"
    android:id="@+id/job_details_date"/>

```

<TextView

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:text="Description"
        android:textAppearance="?android:textAppearanceMedium"
        android:textColor="@android:color/black"
        android:id="@+id/job_details_description"/>

```

<TextView

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:text="Skills"
        android:textAppearance="?android:textAppearanceMedium"
        android:textColor="@android:color/black"
        android:id="@+id/job_details_skills"
    />

```

<TextView

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:layout_margin="5dp"
        android:padding="5dp"
        android:text="Salary"
        android:textAppearance="?android:textAppearanceMedium"
        android:textColor="@android:color/black"
        android:id="@+id/job_details_salary"
    />

```

/>

</LinearLayout>

3.2.11 Toolbar.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.appcompat.widget.Toolbar

```

```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:theme="@style/ThemeOverlay.AppCompat.Dark"
    app:popupTheme="@style/ThemeOverlay.AppCompat.Dark"
    android:background="@color/toolbar_color"
    android:layout_height="wrap_content">>

</androidx.appcompat.widget.Toolbar
```

3.3 Database Schema diagram

Job Details

Job Title	Job Description	<u>id</u>	Date	Skills	Salary
-----------	-----------------	-----------	------	--------	--------

CHAPTER 4

IMPLEMENTATION

4.1 List of classes imported

- **NonNull :**
Denotes that a parameter, field or method return value can never be null
- **AppCompatActivity :**
Base class for activities that wish to use some of the newer platform features on older Android devices.
- **ProgressDialog :**
A dialog showing a progress indicator and an optional text message or view. Only a text message or a view can be used at the same time
- **Intent :**
An intent is an abstract description of an operation to be performed.
- **Bundle :**
A mapping from String keys to various Parcelable values.
- **View :**
This class represents the basic building block for user interface components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling
- **Button :**
A user interface element the user can tap or click to perform an action.
- **EditText :**
A user interface element for entering and modifying text.
- **ProgressBar :**
A user interface element that indicates the progress of an operation. Progress bar supports two modes to represent progress: determinate, and indeterminate.
- **Toast :**
A toast is a view containing a quick little message for the user. The toast class helps you create and show those.
- **OnCompleteListener :**

Listener called when Task completes.

- **Task :**

The task class represents a single operation that doesnot return a value and that usually executes asynchronously.

- **AuthResult :**

Result object obtained from operations that can affect the authentication state. Contains a method that returns the currently signed-in user after the operation has completed

- **FirebaseAuth :**

The entry point of the Firebase Authentication SDK.

- **SuppressLint :**

Indicates that Lint should ignore the specified warnings for the annotated element.

- **LayoutInflater :**

Instantiates a layout XML file into its corresponding View objects.

- **ViewGroup :**

A ViewGroup is a special view that can contain other views (called children.) The view group is the base class for layouts and views containers.

- **TintTypedArray :**

A class that wraps a TypedArray and provides the same public API surface. The purpose of this class is so that we can intercept calls to new APIs.

- **Toolbar :**

A standard toolbar for use within application content.

- **LinearLayoutManager :**

A RecyclerView.LayoutManager implementation which provides similar functionality to android.widget.ListView.

- **RecyclerView :**

A flexible view for providing a limited window into a large data set.

- **FirebaseRecyclerAdapter :**

This class is a generic way of backing a RecyclerView with a Firebase location. It handles all of the child events at the given Firebase location and marshals received data into the given class type

- **FirebaseRecyclerOptions :**

Options to configure a FirebaseRecyclerAdapter.

- **DatabaseReference :**

A Firebase reference represents a particular location in your Database and can be used for reading or writing data to that Database location.

- **FirebaseDatabase :**

The entry point for accessing a Firebase Database. You can get an instance by calling `getInstance()`. To access a location in the database and read or write data, use `getReference()`.

- **Menu :**

Interface for managing the items in a menu. By default, every Activity supports an options menu of actions or options.

- **MenuItem :**

Interface for direct access to a previously created menu item.

- **FloatingActionButton :**

Floating action buttons are used for a special type of promoted action. They are distinguished by a circled icon floating above the UI and have special motion behaviors related to morphing, launching, and the transferring anchor point.

4.2 List of imported Library Functions

- **onCreate() :**

Called when the activity is starting.

- **getSupportActionBar() :**

Set a Toolbar to act as the ActionBar for this Activity window.

- **setTitle() :**

Set the action bar's title. This will only be displayed if `DISPLAY_SHOW_TITLE` is set.

- **setHomeButtonEnabled() :**

Enable or disable the "home" button in the corner of the action bar.

- **getInstance() :**

Gets the default FirebaseDatabase instance.

- **getReference() :**

Gets a DatabaseReference for the database root node.

- **setHasFixedSize() :**

RecyclerView can perform several optimizations if it can know in advance that RecyclerView's size is not affected by the adapter contents.

- **setLayoutManager() :**
Set the RecyclerView.LayoutManager that this RecyclerView will use.

- **onBindViewHolder ():**
The new ViewHolder will be used to display items of the adapter using this.
- **setText() :**
Sets the text to be displayed.
- **onCreateViewHolder() :**
Creates new ViewHolder object whenever the RecyclerView needs a new one.
- **startListening() :**
Start listening for database changes and populate the adapter.
- **setAdapter() :**
Set a new adapter to provide child views on demand.
- **onCreateOptionsMenu() :**
Initialize the contents of the Activity's standard options menu. You should place your menu items in to menu.
- **onOptionsItemSelected() :**
This hook is called whenever an item in your options menu is selected. The default implementation simply returns false to have the normal processing happen
- **toString() :**
Returns a string representation of the object. In general, the `toString` method returns a string that "textually represents" this object.
- **trim() :**
Returns a string whose value is this string, with any leading and trailing whitespace removed.
- **setError() :**
Sets the right-hand compound drawable of the TextView to the "error" icon and sets an error message that will be displayed in a popup when the TextView has focus.
- **getDateInstance() :**
Gets the date formatter with the default formatting style for the default FORMAT locale.
- **makeText() :**
Make a standard toast that just contains text.
- **getStringExtra() :**

Retrieve extended data from the intent.

- **setStackFromEnd() :**

Compatibility support for android.widget.AbsListView.setStackFromBottom(boolean)

- **LinearLayoutManager () :**

Creates a vertical LinearLayoutManager.

4.3 Java Code

4.3.1 Mainactivity

```
package com.naazcodes.jobportal;

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

import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
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.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {

    private EditText email;
```

```
private EditText password;  
  
private Button btnLogin;  
private Button btnRegistration;  
  
//Firebase Auth  
  
private FirebaseAuth mAuth;  
  
//Progress dialog  
  
private ProgressDialog mDialog;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) { // Called when activity is  
starting  
  
super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_main);  
  
mAuth = FirebaseAuth.getInstance(); // Connecting to firebase authentication SDK  
if(mAuth.getCurrentUser() != null) {  
    startActivity(new Intent(getApplicationContext(), HomeActivity.class)); // After  
successful login redirecting to home page  
}  
  
mDialog = new ProgressDialog(this);  
  
LoginFunction();  
}  
  
private void LoginFunction(){
```

```
email=findViewById(R.id.email_login);
password=findViewById(R.id.login_password);

btnLogin = findViewById(R.id.btn_login);
btnRegistration=findViewById(R.id.btn_reg);

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String mEmail = email.getText().toString().trim(); // Converting email to string
        String pass = password.getText().toString().trim(); // Converting password to string

        if(TextUtils.isEmpty(mEmail)){ // Checking whether email is empty
            email.setError("Required field..");
            return;
        }

        if(TextUtils.isEmpty(pass)){ // Checking whether password is empty
            password.setError("Required field..");
        }

        mDialog.setMessage("Processing.."); // Proceesing data
        mDialog.show();

        mAuth.signInWithEmailAndPassword(mEmail,pass).addOnCompleteListener(new
        OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
```

```

if(task.isSuccessful()){
    Toast.makeText(getApplicationContext(),"Successful",
Toast.LENGTH_SHORT).show(); // showing successful toast message
    startActivity(new Intent(getApplicationContext(),HomeActivity.class));

    mDialog.dismiss();
} else{
    Toast.makeText(getApplicationContext(),"Login
Failed",Toast.LENGTH_SHORT).show(); // Showing Login failed message
}
}
);
}

btnRegistration.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // If user is not registered redirecting to registration page
        startActivity(new Intent(getApplicationContext(),RegistrationActivity.class));

    }
});
}
}

```

4.3.2 RegistrationActivity

```

package com.naazcodes.jobportal;

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

```

```
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

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

public class RegistrationActivity extends AppCompatActivity {

    private EditText emailReg;
    private EditText passReg;

    private Button btnReg;
    private Button btnLogin;

    //Firebase Authentication

    private FirebaseAuth mAuth;

    //Progress dialog

    private ProgressDialog mDialog;

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

```

setContentView(R.layout.activity_registration);

mAuth=FirebaseAuth.getInstance(); // Connecting to firebase authentication SDK

mDialog=new ProgressDialog(this);

Registration(); // Calling registration method
}

private void Registration(){

emailReg=findViewById(R.id.email_registration);
passReg=findViewById(R.id.registration_password);

btnReg=findViewById(R.id.btn_registration);
btnLogin=findViewById(R.id.btn_login);

btnReg.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String email=emailReg.getText().toString().trim(); // Converting email to string

        String pass=passReg.getText().toString().trim(); // Converting password to string

        if (TextUtils.isEmpty(email)){
            emailReg.setError("Requested field.."); // Showing error message
            return;
        }

        if (TextUtils.isEmpty(pass)){
            passReg.setError("Required field..");
        }
    }
}

```

```

return;
}

mDialog.setMessage("Processing.."); // Processing data
mDialog.show();

mAuth.createUserWithEmailAndPassword(email,pass).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {

        if(task.isSuccessful()){

            Toast.makeText(getApplicationContext(),"Successfull",
            Toast.LENGTH_SHORT).show(); // Showing successful message

            startActivity(new Intent(getApplicationContext(),HomeActivity.class));

            mDialog.dismiss();

        }else{
            Toast.makeText(getApplicationContext(),"Registration
            failed",Toast.LENGTH_SHORT).show(); // Toast message for failed registration
        }

    }
    });

}

});

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override

```

```
public void onClick(View v) {  
  
    startActivity(new Intent(getApplicationContext(), MainActivity.class)); // After  
    successful registration redirecting to login page  
  
}  
});  
}  
}
```

4.3.3 PostJobActivity

```
package com.naazcodes.jobportal;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.EditText;  
import android.widget.TextView;  
import androidx.appcompat.widget.Toolbar;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
//import com.example.jobportalapp.Model.Data;  
import com.firebaseio.ui.database.FirebaseRecyclerAdapter;  
import com.firebaseio.ui.database.FirebaseRecyclerOptions;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;
```

```
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.naazcodes.jobportal.Model.Data;

public class PostJobActivity extends AppCompatActivity {

    private FloatingActionButton fabBtn;
    private Toolbar toolbar;

    //Recycler View

    private RecyclerView recyclerView;

    //Firebase

    private FirebaseAuth mAuth;
    private DatabaseReference JobPostDatabase;

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

        toolbar = findViewById(R.id.toolbar_post_job);
        setSupportActionBar(toolbar);

        // Setting a toolbar to act as a actionBar
        getSupportActionBar().setHomeButtonEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        fabBtn = findViewById(R.id.fab_add)
    }
}
```

```
mAuth = FirebaseAuth.getInstance();

FirebaseUser mUser = mAuth.getCurrentUser();
String uId = mUser.getUid();

JobPostDatabase = FirebaseDatabase.getInstance().getReference().child("Job
Post").child(uId);

recyclerView = findViewById(R.id.recycler_job_post_id);
LinearLayoutManager layoutManager = new LinearLayoutManager(this);
layoutManager.setStackFromEnd(true);
layoutManager.setReverseLayout(true);

recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(layoutManager);

fabBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        startActivity(new Intent(getApplicationContext(), InsertJobPostActivity.class));

    }
});

}

@Override
protected void onStart(){
    super.onStart()
}
```

```
FirebaseRecyclerOptions<Data> firebaseRecyclerOptions = new  
FirebaseRecyclerOptions.Builder<Data>().setQuery(JobPostDatabase, Data.class).build();
```

```
        FirebaseRecyclerAdapter<Data, MyViewHolder> adapter = new  
        FirebaseRecyclerAdapter<Data, MyViewHolder>(firebaseRecyclerOptions) {  
  
            @NonNull  
            @Override  
            public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int  
viewType) {  
                View view =  
LayoutInflator.from(parent.getContext()).inflate(R.layout.job_post_item,parent,false);  
                return new MyViewHolder(view);  
            }  
  
            @Override  
            protected void onBindViewHolder(@NonNull MyViewHolder viewHolder, int i,  
@NonNull Data model) {  
// Setting values for the variable  
                viewHolder.setJobTitle(model.getTitle());  
                viewHolder.setJobDate(model.getDate());  
                viewHolder.setJobDescription(model.getDescription());  
                viewHolder.setJobSkills(model.getSkills());  
                viewHolder.setJobSalary(model.getSalary());  
            }  
        };  
        adapter.startListening();  
        recyclerView.setAdapter(adapter);  
    }  
  
    public static class MyViewHolder extends RecyclerView.ViewHolder{
```

View **myview**:

// Assigning the values

```
public MyViewHolder(@NonNull View itemView) {
    super(itemView);
    myview = itemView;
}
```

```
public void setJobTitle(String title){
```

```
    TextView mTitle = myview.findViewById(R.id.job_title);
    mTitle.setText(title);
```

```
}
```

```
public void setJobDate(String date){
```

```
    TextView mDate = myview.findViewById(R.id.job_date);
    mDate.setText(date);
```

```
}
```

```
public void setJobDescription(String description){
```

```
    TextView mDescription = myview.findViewById(R.id.job_description);
    mDescription.setText(description);
```

```
}
```

```
public void setJobSkills(String skills){
```

```
    TextView mSkills = myview.findViewById(R.id.job_skills);
    mSkills.setText.skills();
```

```
}
```

```
public void setJobSalary(String salary){
```

```
    TextView mSalary = myview.findViewById(R.id.job_salary);
    mSalary.setText(salary);
```

```
}
```

}

}

4.3.4 JobDetailsActivity

```
package com.naazcodes.jobportal;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

import androidx.appcompat.widget.Toolbar;

import com.firebaseio.ui.database.FirebaseIndexArray;

import org.w3c.dom.Text;

public class JobDetailsActivity extends AppCompatActivity {

    private Toolbar toolbar;

    private TextView mTitle;
    private TextView mDate;
    private TextView mDescription;
    private TextView mSkills;
    private TextView mSalary;

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_job_details);

    toolbar = findViewById(R.id.toolbar_job_details);
    setSupportActionBar(toolbar);
    getSupportActionBar().setTitle("Job Details"); // setting actionBar title

    getSupportActionBar().setHomeButtonEnabled(true);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    mTitle = findViewById(R.id.job_details_title);
    mDate = findViewById(R.id.job_details_date);
    mDescription = findViewById(R.id.job_details_description);
    mSkills = findViewById(R.id.job_details_skills);
    mSalary = findViewById(R.id.job_details_salary);

    //Receive Data from all job activity using intent..
}

Intent intent = getIntent();

String title = intent.getStringExtra("title");
String date = intent.getStringExtra("date");
String description = intent.getStringExtra("description");
String skills = intent.getStringExtra("skills");
String salary = intent.getStringExtra("salary");

mTitle.setText(title);
mDate.setText(date);
mDescription.setText(description);

```

```
    mSkills.setText.skills);
    mSalary.setText.salary);

}

}
```

4.3.5 InserJobPostActivity

```
package com.naazcodes.jobportal;

import androidx.appcompat.app.AppCompatActivity;

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

import androidx.appcompat.widget.Toolbar;

import com.naazcodes.jobportal.Model.Data;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;

import java.text.DateFormat;
import java.util.Date;

public class InsertJobPostActivity extends AppCompatActivity {
```

```
private Toolbar toolbar;

private EditText job_title;
private EditText job_description;
private EditText job_skills;
private EditText job_salary;
private Button btn_post_job;

//Firebase Auth

private FirebaseAuth mAuth;
private DatabaseReference mJobPost;

private DatabaseReference mPublicDatabase;

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

    toolbar = findViewById(R.id.insert_job_toolbar);
    setSupportActionBar(toolbar);
    getSupportActionBar().setTitle("Post job"); //setting Actionbar title

    getSupportActionBar().setHomeButtonEnabled(true);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    mAuth = FirebaseAuth.getInstance();

    FirebaseUser mUser = mAuth.getCurrentUser();
    String uId = mUser.getUid();
```

```

mJobPost = FirebaseDatabase.getInstance().getReference().child("Job Post").child(userId);
mPublicDatabase = FirebaseDatabase.getInstance().getReference().child("Public
database");

InsertJob(); // Calling insertjob method
}

private void InsertJob() {

job_title = findViewById(R.id.job_title);
job_description = findViewById(R.id.job_description);
job_skills = findViewById(R.id.job_skill);
job_salary = findViewById(R.id.job_salary);

btn_post_job = findViewById(R.id.btn_job_post);
// Method for clicking post job button
btn_post_job.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String title = job_title.getText().toString().trim();
        String description = job_description.getText().toString().trim();
        String skills = job_skills.getText().toString().trim();
        String salary = job_salary.getText().toString().trim();

        // Checking whether Required fields are filled if not showing error message
        if(TextUtils.isEmpty(title)){
            job_title.setError("Required Field..");
            return;
        }

        if(TextUtils.isEmpty(description)){
}

```

```
job_description.setError("Required Field..");
return;
}

if(TextUtils.isEmpty.skills)){
job_skills.setError("Required Field..");
return;
}

if(TextUtils.isEmpty(salary)){
job_salary.setError("Required Field..");
return;
}

String id = mJobPost.push().getKey();

String date = DateFormat.getDateInstance().format(new Date());

Data data = new Data(title, description, skills, salary, id, date);

mJobPost.child(id).setValue(data);

mPublicDatabase.child(id).setValue(data);
Toast.makeText(getApplicationContext(),"Successful",
Toast.LENGTH_SHORT).show(); // Showing toast message after successfully inserting job
// Redirecting to postJobActivity
startActivity(new Intent(getApplicationContext(), PostJobActivity.class));

}

});

}

}
```

4.3.6 HomeActivity

```
package com.naazcodes.jobportal;

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

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.widget.Toolbar;

import com.google.firebase.auth.FirebaseAuth;

public class HomeActivity extends AppCompatActivity {

    private Button btnAllJob;
    private Button btnPostJob;

    //Toolbar

    private Toolbar toolbar;
    private FirebaseAuth mAuth;

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

        toolbar = findViewById(R.id.toolbar_home);
        setSupportActionBar(toolbar);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_logout) {
            mAuth.signOut();
            Intent intent = new Intent(HomeActivity.this, LoginActivity.class);
            startActivity(intent);
        }
        return super.onOptionsItemSelected(item);
    }
}
```

```
getSupportActionBar().setTitle("Job Portal App"); // Setting title for actionBar

mAuth = FirebaseAuth.getInstance();

btnAllJob = findViewById(R.id.btn_allJob);
btnPostJob = findViewById(R.id.btn_postJob);

btnAllJob.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //On clicking allJob button redirection to AllJobActivity
        startActivity(new Intent(getApplicationContext(), AllJobActivity.class));
    }
});

btnPostJob.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //On clicking PostJob button redirection to PostJobActivity
        startActivity(new Intent(getApplicationContext(), PostJobActivity.class));
    }
});

@Override
public boolean onCreateOptionsMenu(Menu menu) {
```

```
getMenuInflater().inflate(R.menu.mainmenu,menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {

    switch (item.getItemId()){
        case R.id.logout: //After clicking logout button signing out from the home page
            mAuth.signOut();
            startActivity(new Intent(getApplicationContext(), MainActivity.class));
            break;

    }

    return super.onOptionsItemSelected(item);
}
```

4.3.7 AllJobActivity

```
package com.naazcodes.jobportal;

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

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import androidx.appcompat.widget.TintTypedArray;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.naazcodes.jobportal.Model.Data;
import com.firebaseio.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.naazcodes.jobportal.Model.Data;

import org.w3c.dom.Text;

public class AllJobActivity extends AppCompatActivity {

    private Toolbar toolbar;
    //Recycler
    private RecyclerView recyclerView;
    //Firebase
    private DatabaseReference mAllJobPost;
    private FirebaseRecyclerOptions<Data> options;
    private FirebaseRecyclerAdapter<Data,viewHolder> adapter;

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

```

toolbar = findViewById(R.id.all_job_post);
setSupportActionBar(toolbar);

getSupportActionBar().setTitle("All Job Post");
getSupportActionBar().setHomeButtonEnabled(true);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);

//Database

mAllJobPost = FirebaseDatabase.getInstance().getReference().child("Public database");

recyclerView = findViewById(R.id.recycler_all_job);
recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(new LinearLayoutManager(this));

options = new
FirebaseRecyclerOptions.Builder<Data>().setQuery(mAllJobPost,Data.class).build();

adapter = new FirebaseRecyclerAdapter<Data, viewHolder>(options) {
    @SuppressLint("SetTextI18n")
    @Override
    protected void onBindViewHolder(@NonNull viewHolder viewHolder, int i, @NonNull
Data data) {
    //The new viewholder will be used to display items of the adapter
        viewHolder.title.setText(" "+data.getTitle());
        viewHolder.date.setText(" "+data.getDate());
        viewHolder.desc.setText(" "+data.getDescription());
        viewHolder.salary.setText(" "+data.getSalary());
        viewHolder.skills.setText(" "+data.getSkills());
    }
    @NonNull
    @Override
    public viewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

```

```
View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.alljobpost,parent,false);
    return new viewHolder(v);
}
};

adapter.startListening(); // Start listening for database changes and populate the adapter
recyclerView.setAdapter(adapter); // Set a new adapter to provide child views

}

}
```

4.3.8 ViewHolder

```
package com.naazcodes.jobportal;

import android.view.View;
import android.widget.TextView;

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

public class viewHolder extends RecyclerView.ViewHolder {

    TextView title,date,desc,skills,salary;
    public viewHolder(@NonNull View itemView) {
        super(itemView);

        title = itemView.findViewById(R.id.all_job_post_title);
        date = itemView.findViewById(R.id.all_job_post_date);
        desc = itemView.findViewById(R.id.all_job_post_description);
        skills = itemView.findViewById(R.id.all_job_post_skills);
        salary = itemView.findViewById(R.id.all_job_post_salary);
    }
}
```

4.3.9 Model(Data.java)

```
package com.naazcodes.jobportal.Model;
```

```
public class Data {
```

```
    String title;
```

```
    String description;
```

```
    String skills;
```

```
    String salary;
```

```
    String id;
```

```
    public Data() {
```

```
    }
```

```
    String date;
```

```
    public Data(String title, String description, String skills, String salary, String id, String date) {
```

```
        this.title = title;
```

```
        this.description = description;
```

```
        this.skills = skills;
```

```
        this.salary = salary;
```

```
        this.id = id;
```

```
        this.date = date;
```

```
}
```

```
    public String getTitle() {
```

```
        return title;
```

```
}
```

```
    public void setTitle(String title) {
```

```
        this.title = title;
```

```
}
```

```
public String getDescription() {
```

```
    return description;
```

```
}
```

```
public void setDescription(String description) {
```

```
    this.description = description;
```

```
}
```

```
public String getSkills() {
```

```
    return skills;
```

```
}
```

```
public void setSkills(String skills) {
```

```
    this.skills = skills;
```

```
}
```

```
public String getSalary() {
```

```
    return salary;
```

```
}
```

```
public void setSalary(String salary) {
```

```
    this.salary = salary;
```

```
}
```

```
public String getId() {
```

```
    return id;
```

```
}
```

```
public void setId(String id) {
```

```
    this.id = id;
```

```
}
```

```
public String getDate() {  
    return date;  
}  
  
public void setDate(String date) {  
    this.date = date;  
}  
}
```

4.3 Manifest file code

4.3.1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.naazcodes.jobportal">  
  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportsRtl="true"  
        android:theme="@style/Theme.JobPortal">  
        <activity android:name=".JobDetailsActivity"  
            android:parentActivityName=".AllJobActivity"></activity>  
        <activity android:name=".InsertJobPostActivity"  
            android:parentActivityName=".HomeActivity"  
        />  
        <activity android:name=".PostJobActivity"  
            android:parentActivityName=".HomeActivity"  
        />  
        <activity android:name=".AllJobActivity"  
            android:parentActivityName=".HomeActivity"  
        />  
    </application>
```

```
/>  
<activity android:name=".HomeActivity" />  
<activity android:name=".RegistrationActivity" />  
<activity android:name=".MainActivity">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
    </activity>  
</application>  
  
</manifest>
```

CHAPTER 5

RESULTS

5.1 SCREENSHOTS

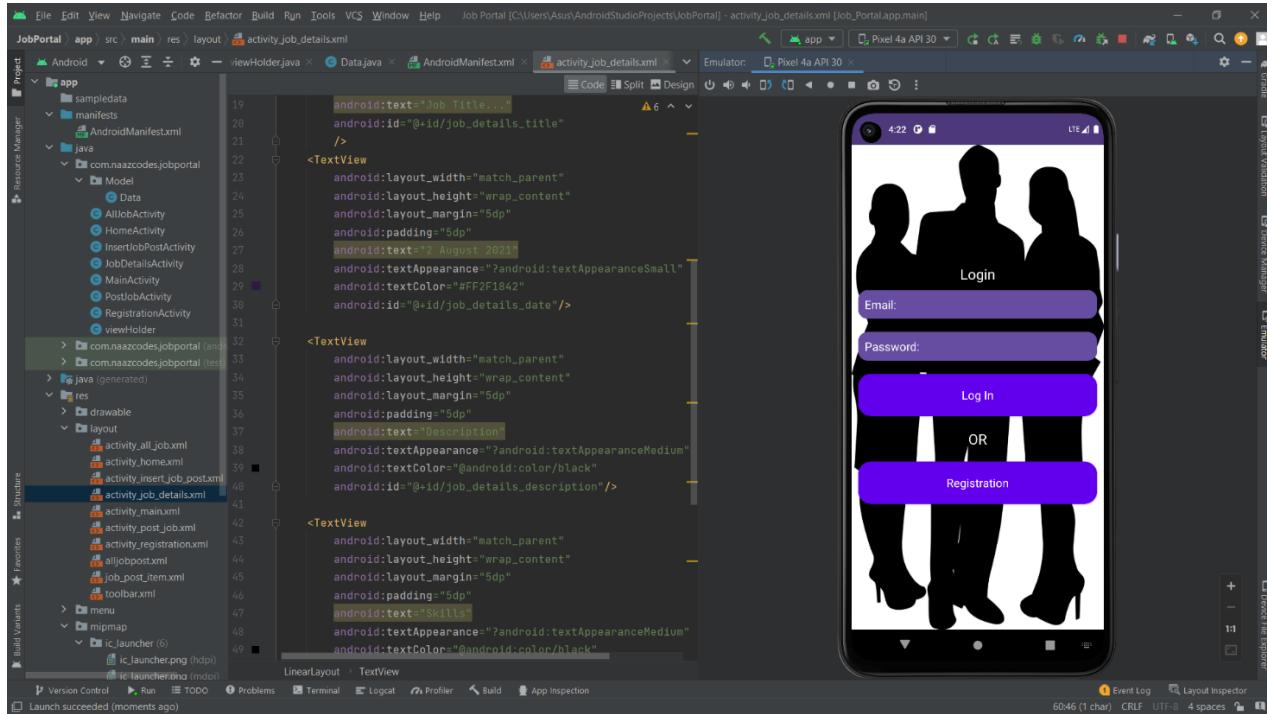


Figure 5.1.1: Front Page/Log In Page

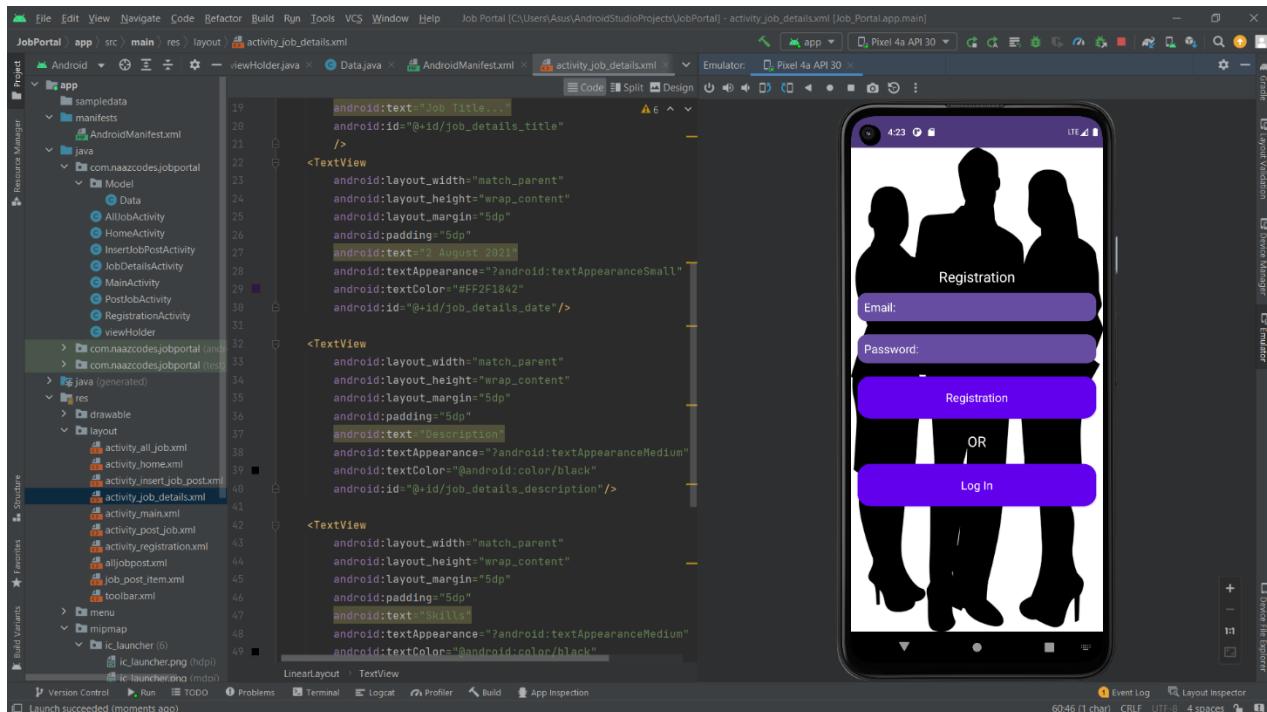


Figure 5.1.2: Registration Page

Job Portal Application

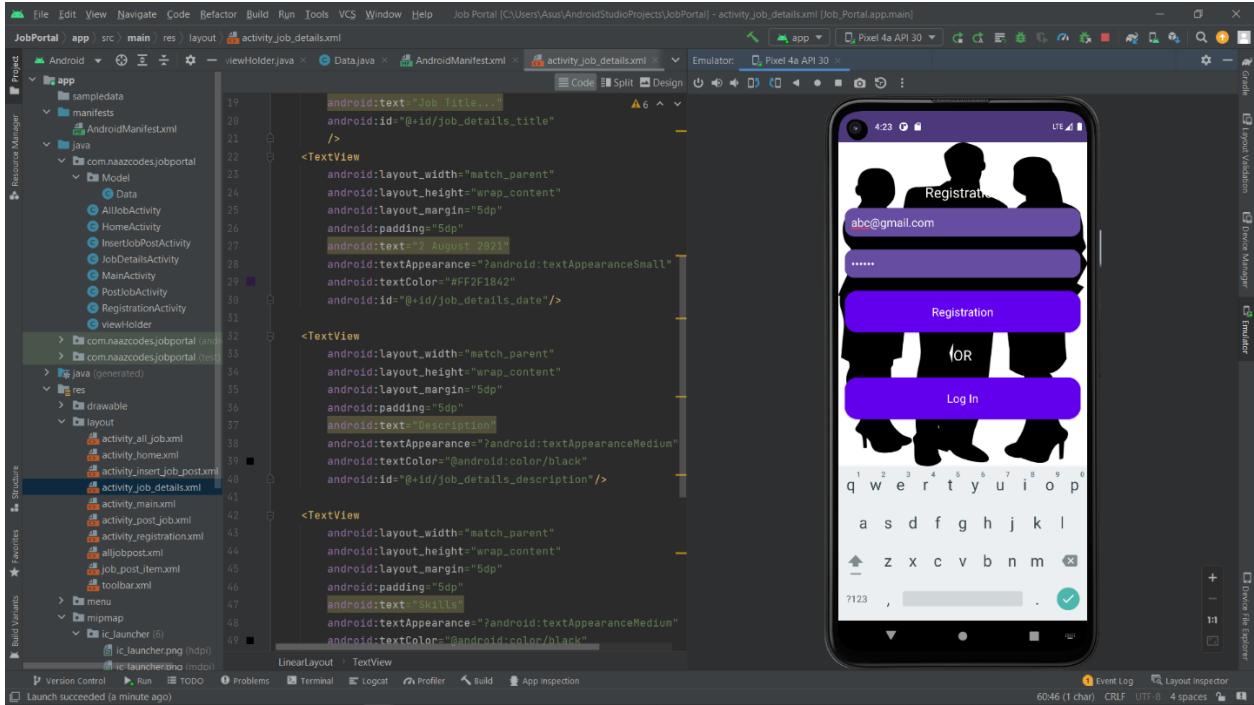


Figure 5.1.3: Registration Page(With Credentials)

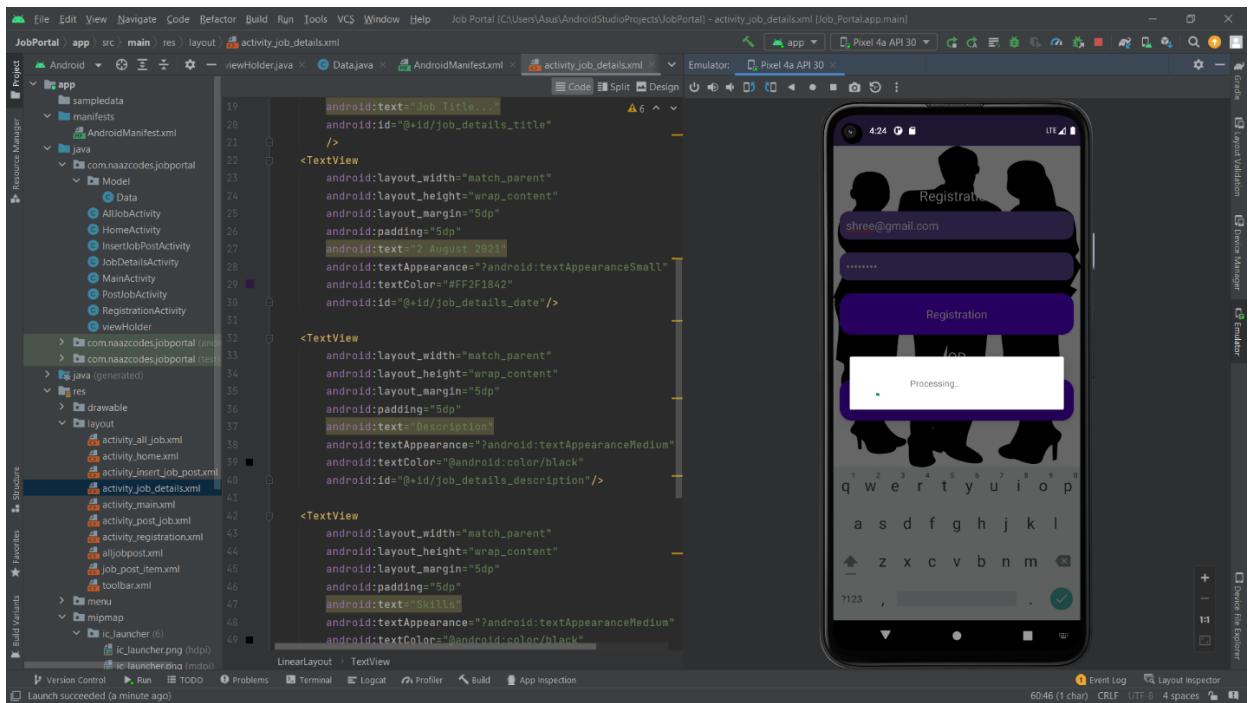


Figure 5.1.4: Registration Page(Processing Credentials)

Job Portal Application

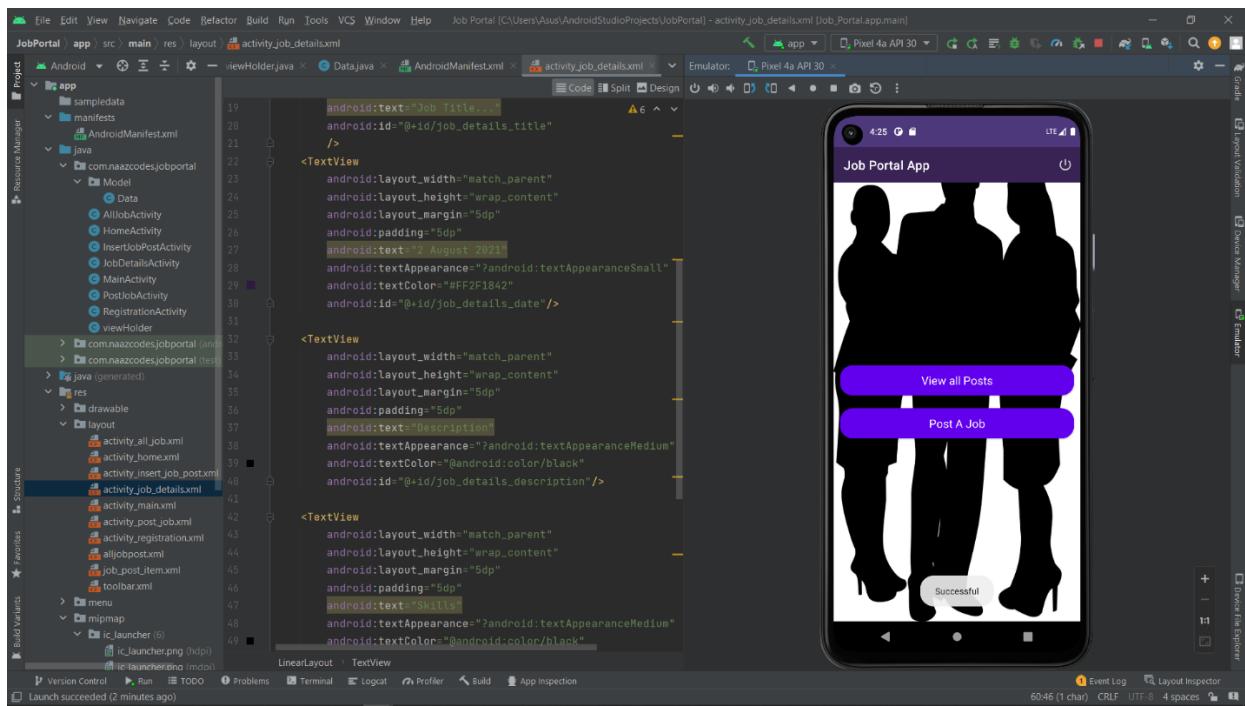


Figure 5.1.5: Page after Successful registration

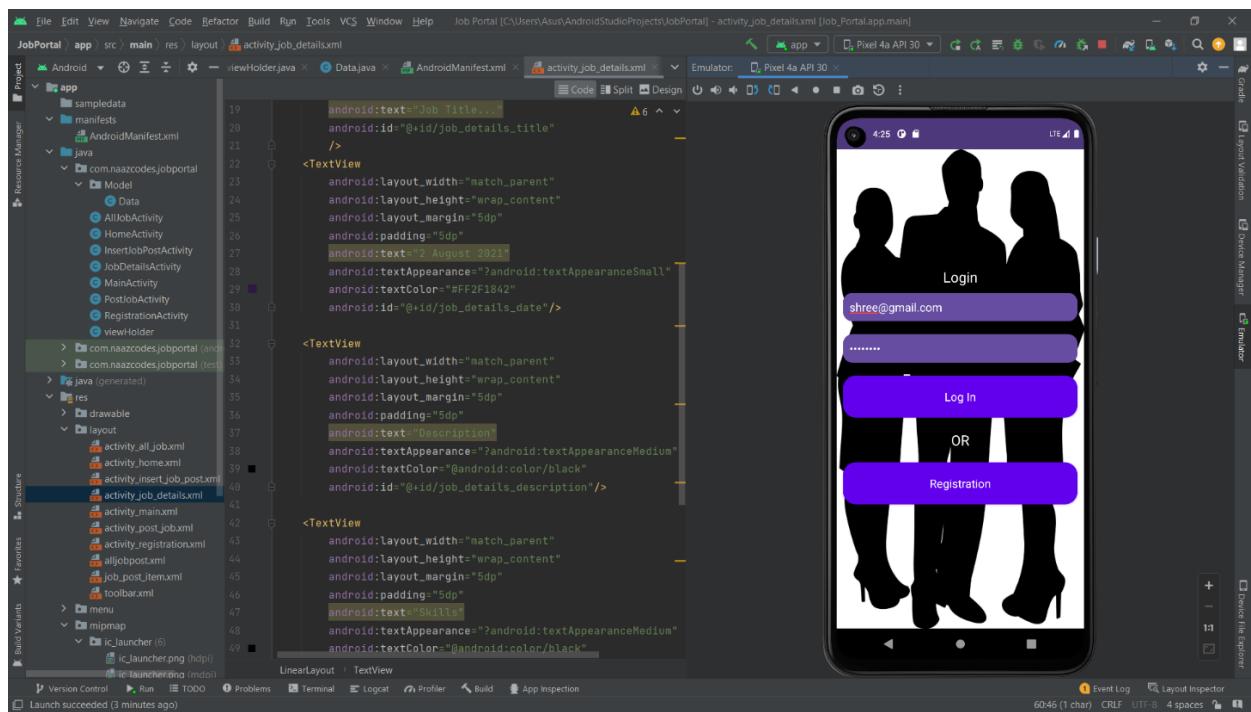


Figure 5.1.6: Login Page(with User Credentials)

Job Portal Application

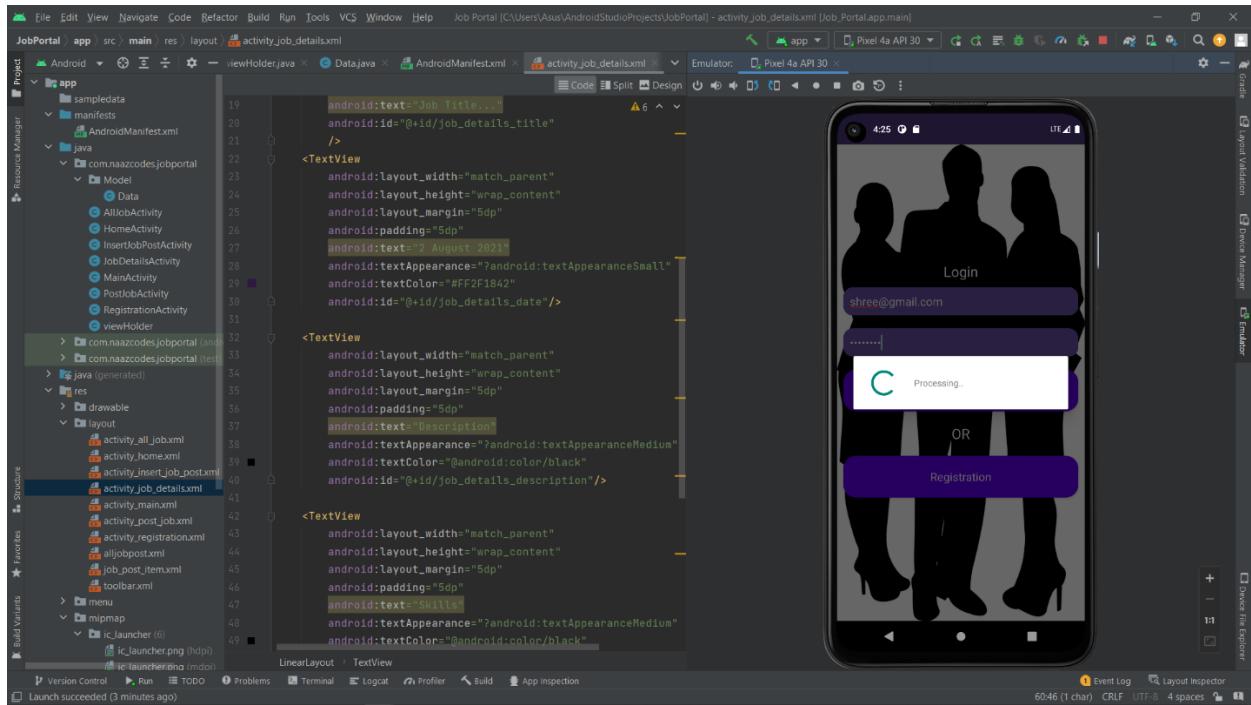


Figure 5.1.7: Login Page(Processing User Credentials)

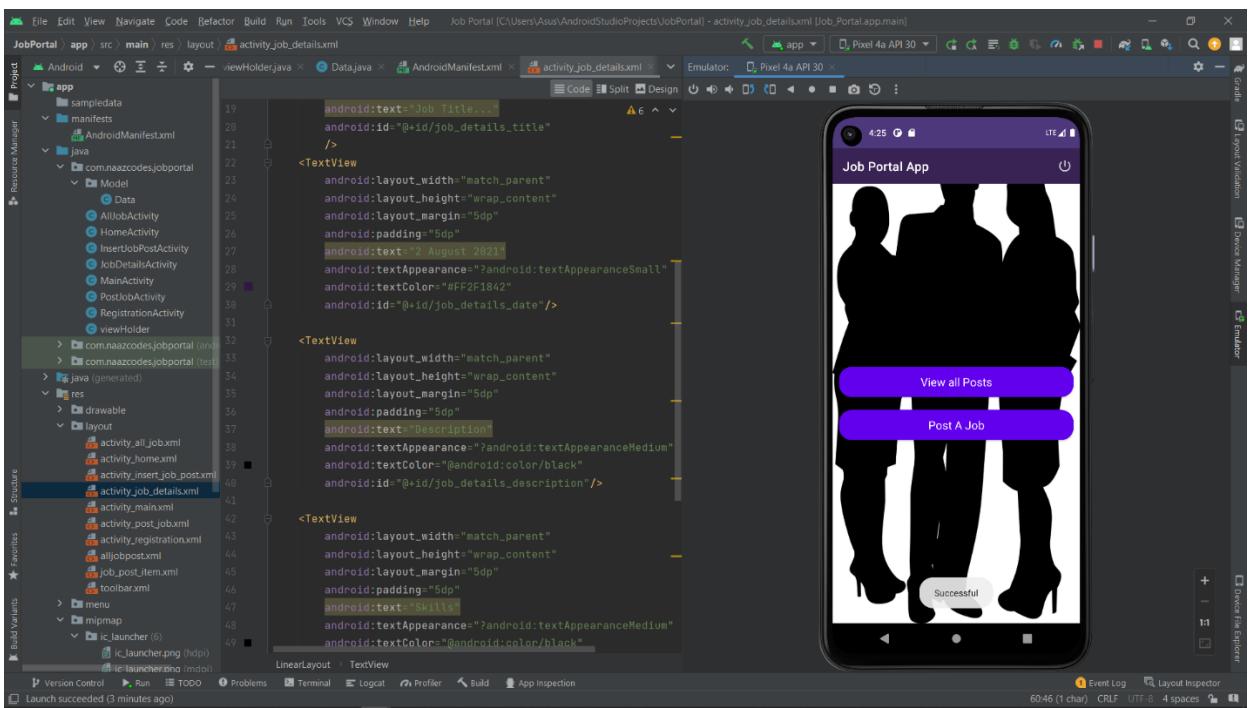


Figure 5.1.8: Page after Successful login

Job Portal Application

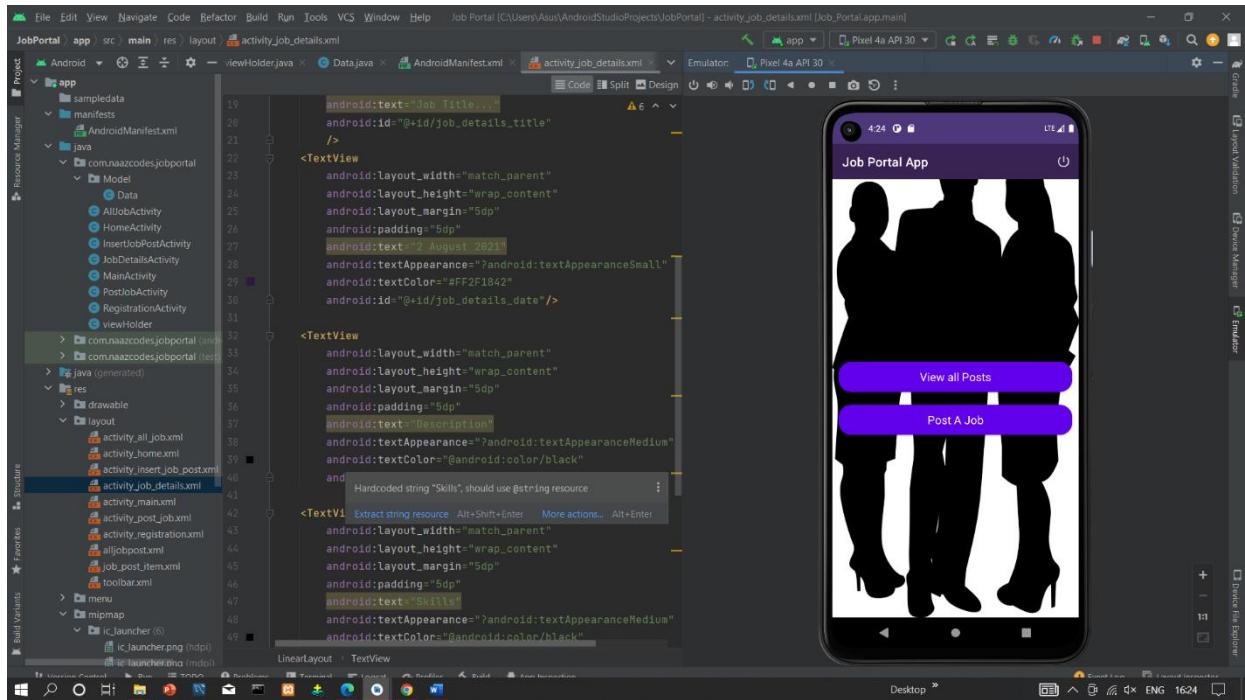


Figure 5.1.9: Page to Either Post or View job

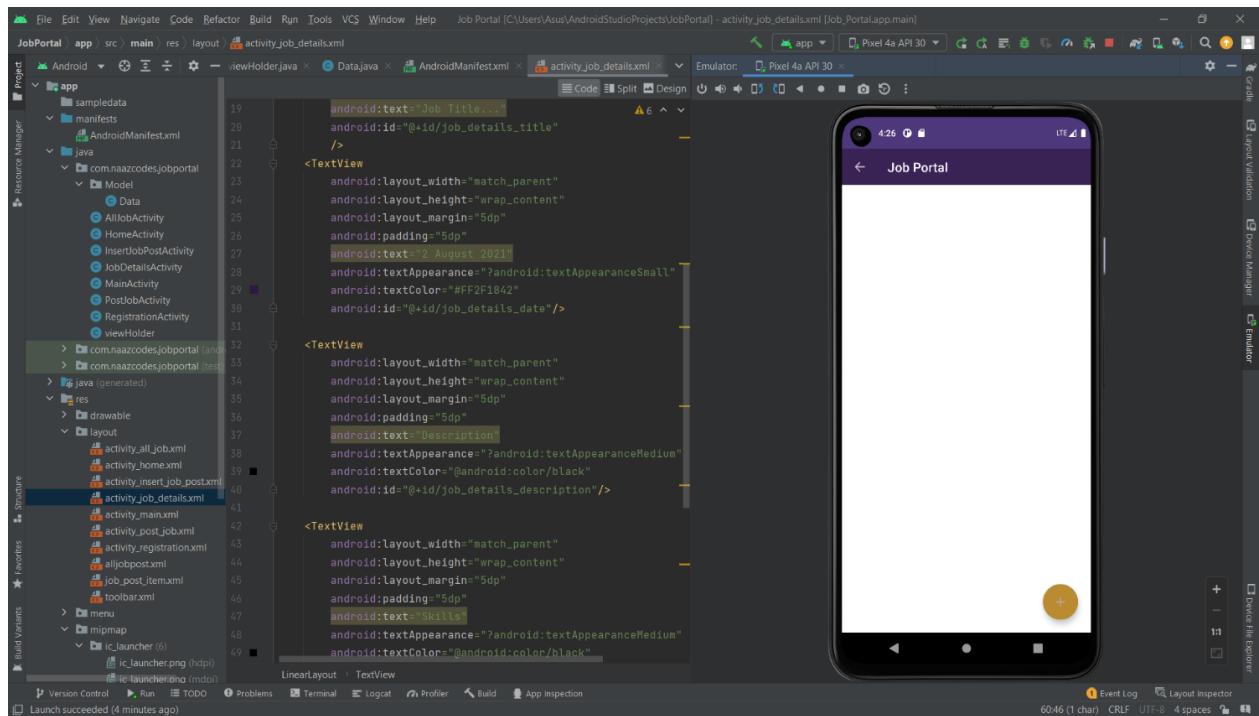


Figure 5.1.10: Post a Job Page (Before User posting a Job)

Job Portal Application

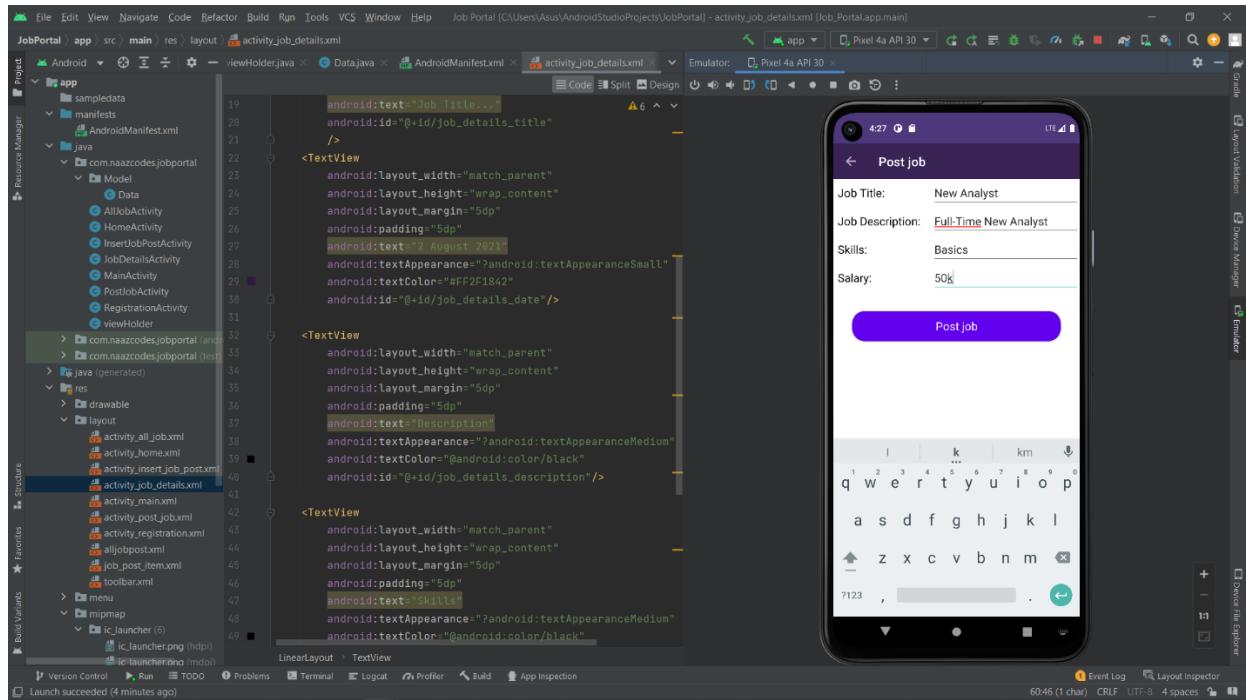


Figure 5.1.11: Page for User to post a Job

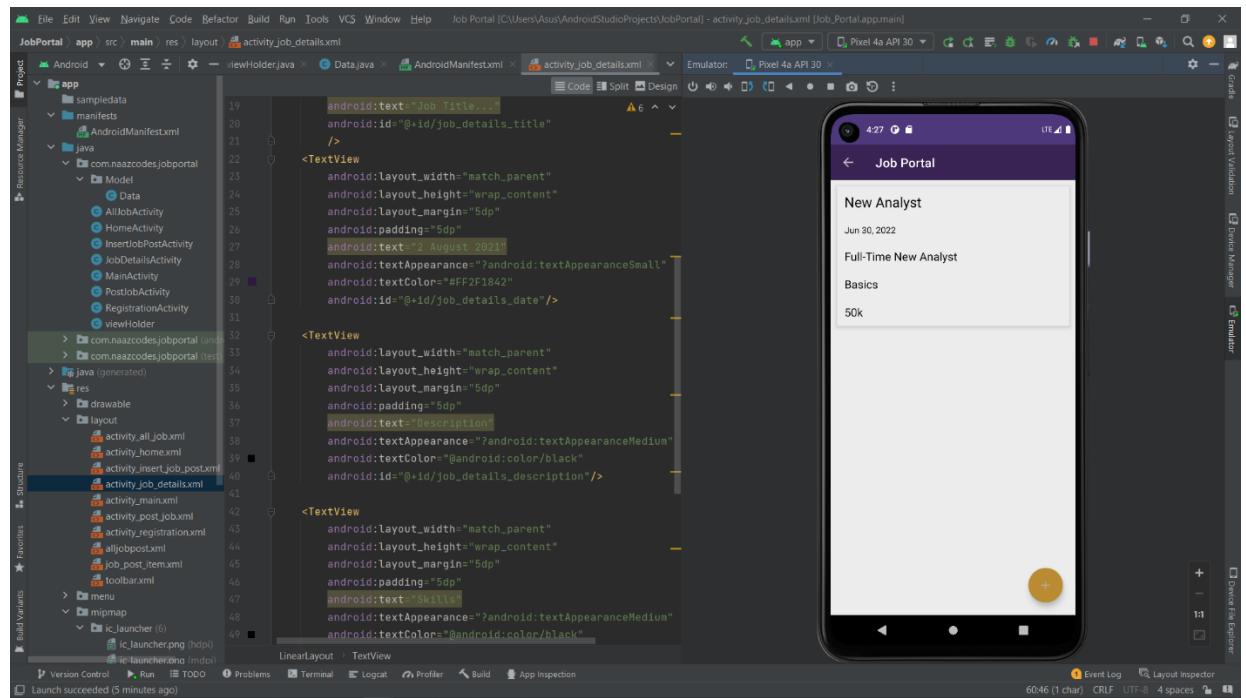


Figure 5.1.12: Post a job Page(After Posting a job)

Job Portal Application

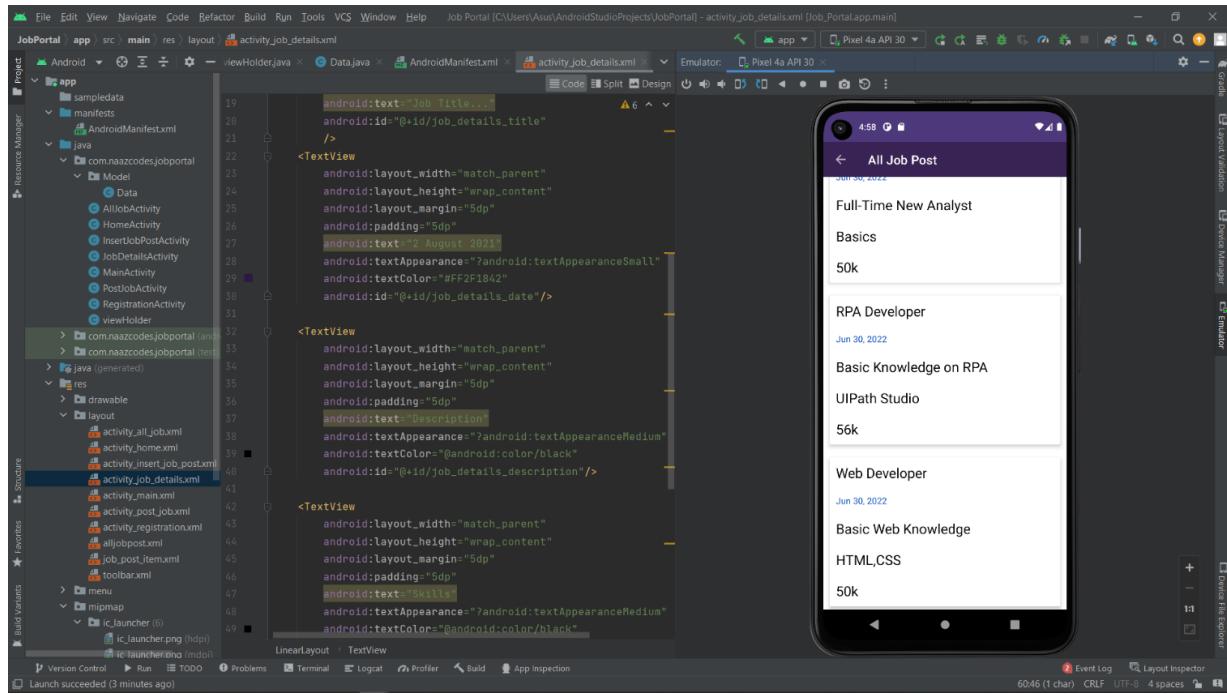


Figure 5.1.13: Page to view all jobs

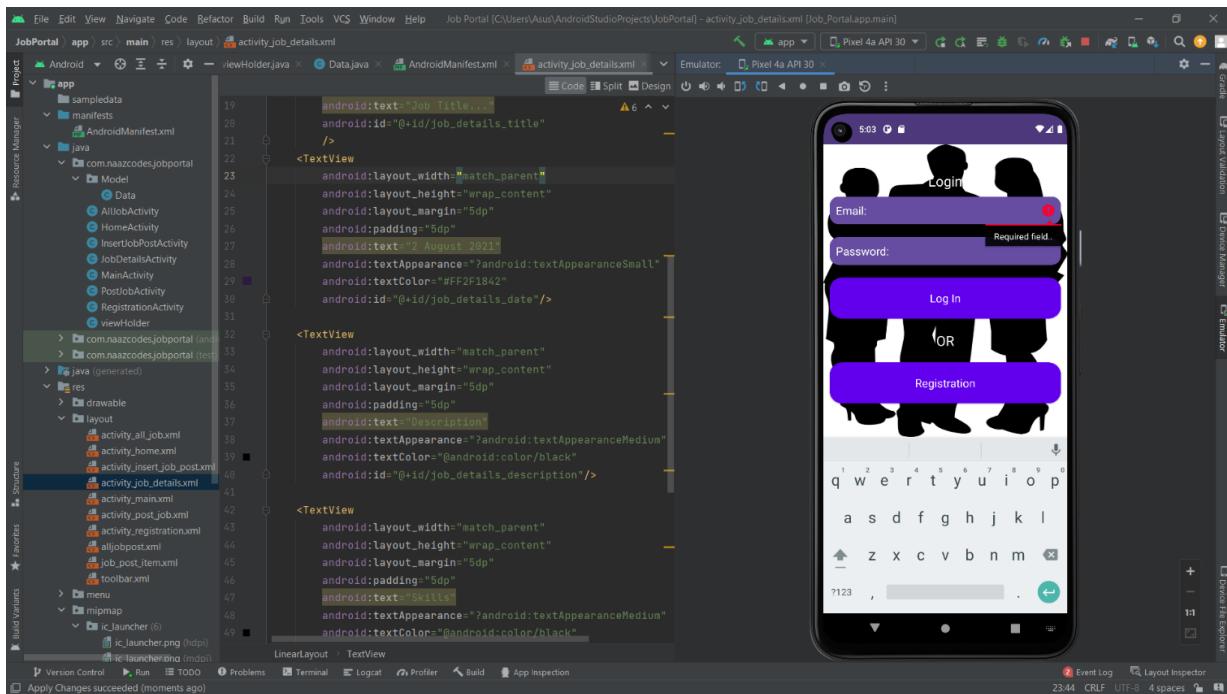


Figure 5.1.14: Failed Login Page

Job Portal Application

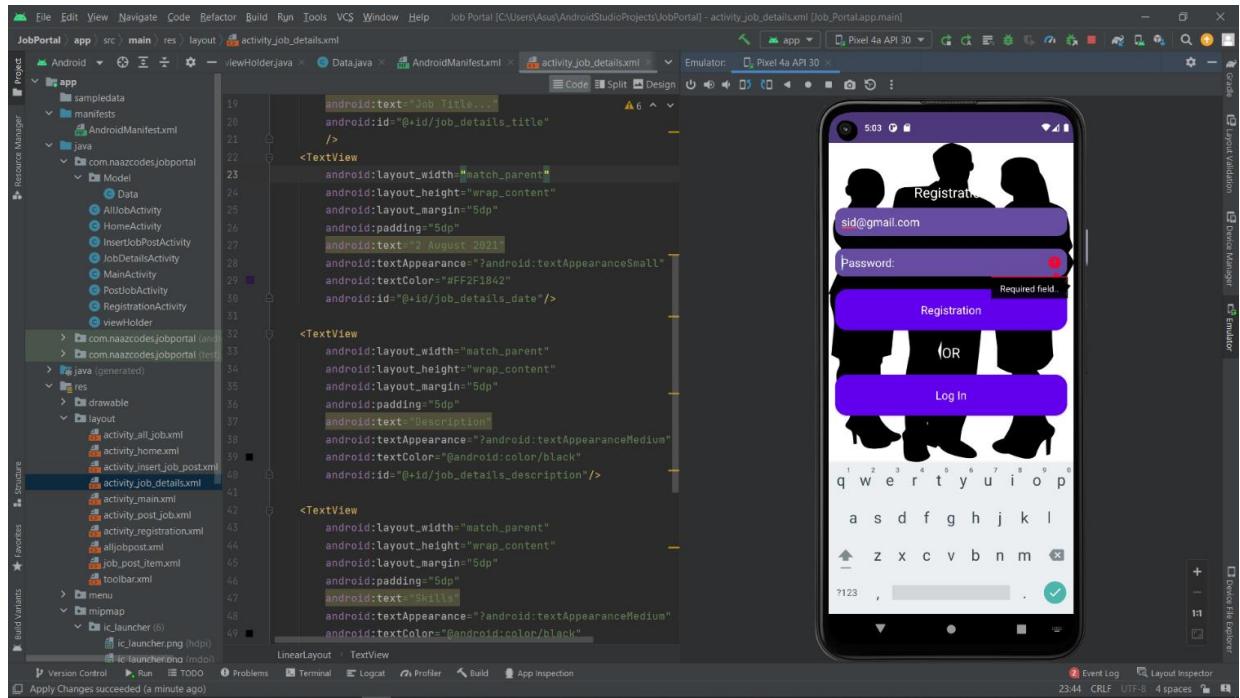


Figure 5.1.15: Failed Registration Page

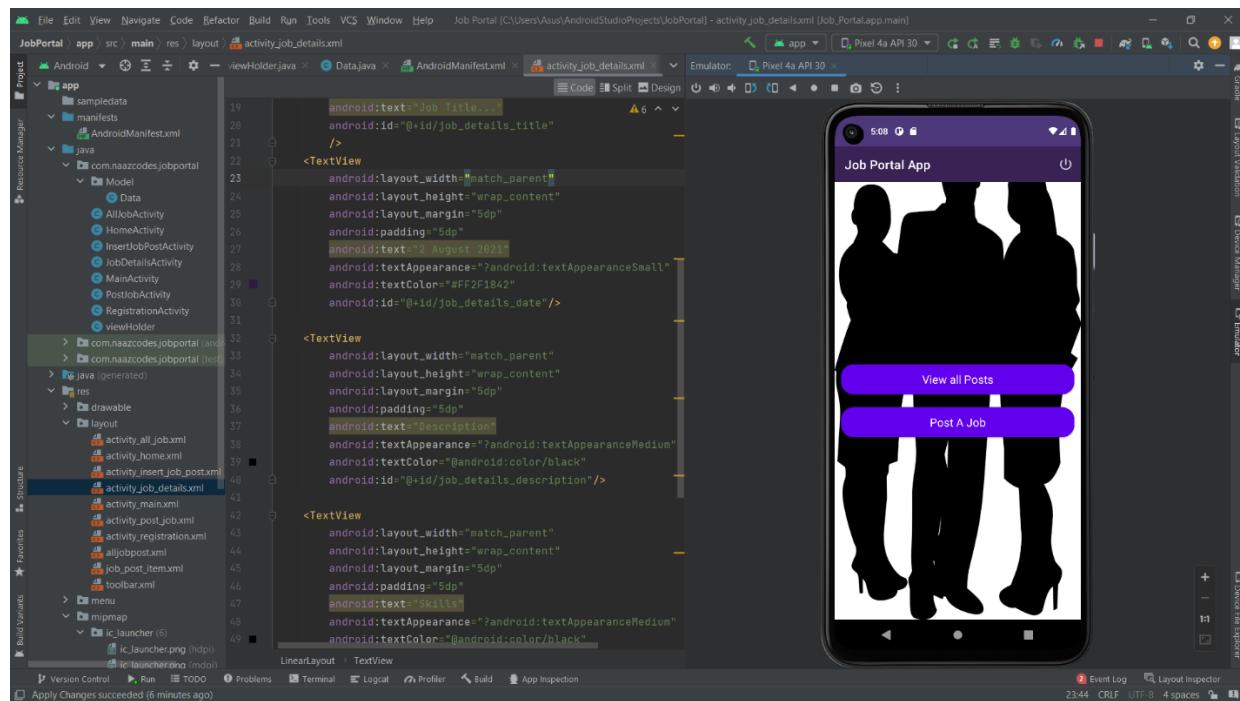


Figure 5.1.17: Logging out

Job Portal Application

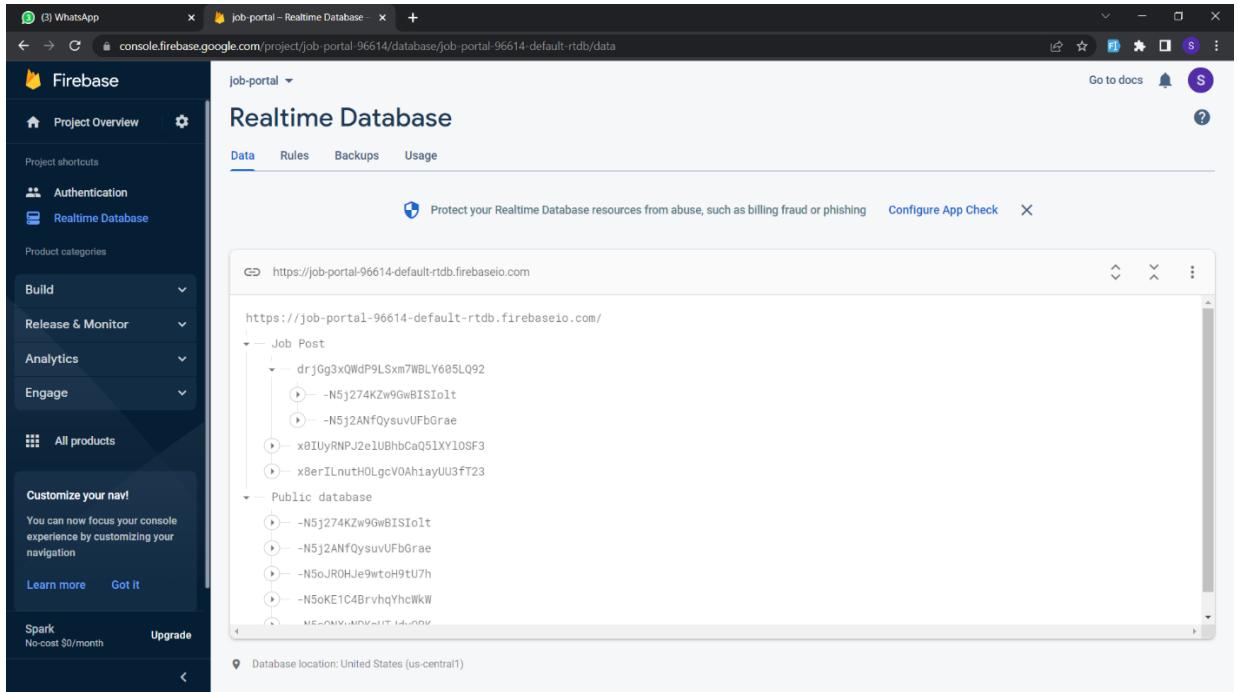


Figure 5.1.18(a):Firebase Console

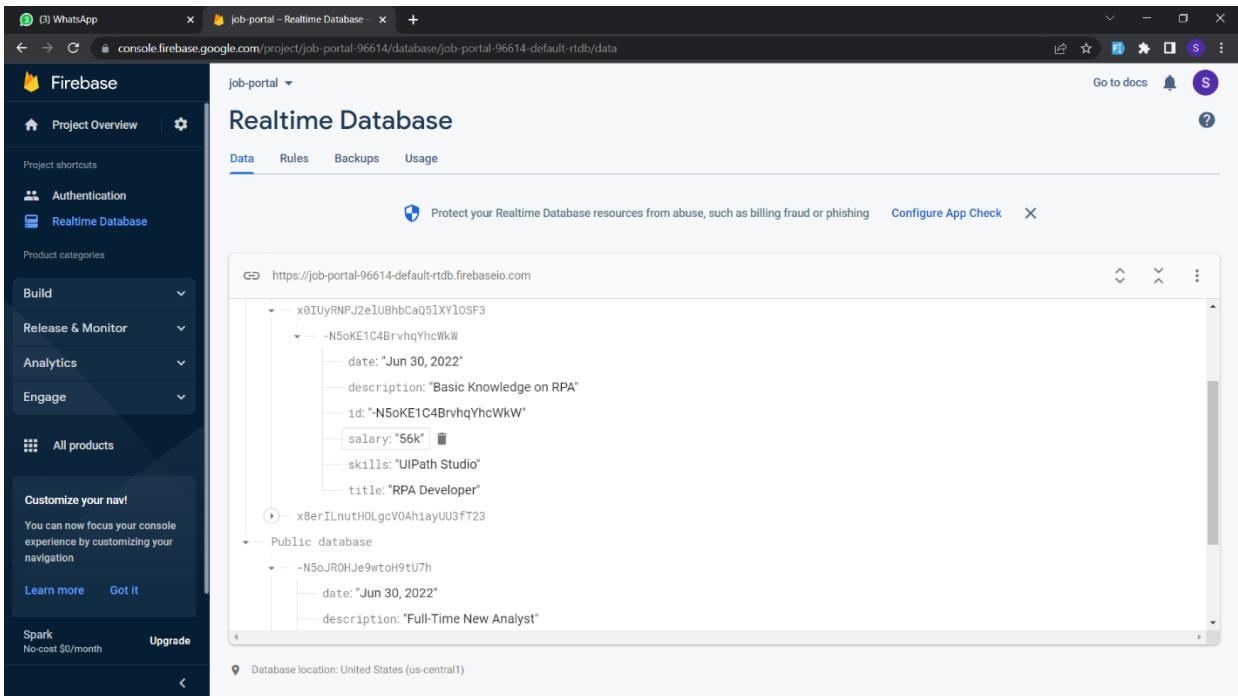


Figure 5.1.18(b):Firebase Console

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENTS

Over the years, things have changed in the recruitment space, with technology making all the difference when it comes to finding jobs as well as searching the right people for the jobs. This Job Portal App facilitates people for finding relevant candidate and jobs. At the same time, as more and more people prefer to use their smartphones for everyday activities, a majority of job services look to have mobile apps in addition to web portals.

Though the developed app can only get new job alerts in future it can be developed in order to facilitate a candidate to apply for the job meeting their criteria. The job portal can also be extended with advanced features such as reminder notifications, quick CV updates, and much more.

REFERENCES

The books and the other sources we referred while doing this project are as follows:

Referred Books

- Google Developer Training **Android Developer Fundamentals Course – Concept Reference** Google Developer Training Team,2017

Websites Referred

- ❖ <https://www.youtube.com>
- ❖ developer.android.com
- ❖ <https://stackoverflow.com/>
- ❖ www.javacodegeeks.com