

**A PROJECT REPORT ON**

**“ONLINE JOB PORTAL”**

**Jorhat Institute Of Science And  
Technology**



**Submitted by: -**

Bijoy Dutta (07) Reg.no: -

Rocktim Rajkumar (30) Reg.no: -

**Guided by:**

Mr. Rajiv Kalita (Assistant Professor)

Department of Computer Science and Information Technology

# **CERTIFICATE**

---

This is to certify that the project entitles “JOB VIBE: ONLINE JOB PORTAL APP” submitted to Jorhat Institute of Science and Technology, Assam in partial fulfillment of the requirement for the completion of the major project of B.SC in Information Technology Degree. It is an original work carried out by Bijoy Dutta and Rajkumar Rocktim Narayan Singha, B.SC IT 6<sup>th</sup> Semester, under the guidance of Mr. Rajiv Kalita (Assistant Professor), Department of Computer Science and Information Technology, JIST.

The matter embodied in this project is a genuine work done by the students. They have also submitted the project with the report. I have been supervising them throughout the activity and they completed the work to my ultimate satisfaction. This work reported here in has not been presented for other academic purpose elsewhere. This as per requirement for a live project under some organization and it would probably meet the entire essential requisite according to the user needs. Their effort is being appreciated by me and I wish them a bright future.

**Date: -**

**Signature of Internal/Guide**

CS&IT Department

JIST, Jorhat, Assam

# Acknowledgments

---

I take this opportunity to express my profound gratitude and deep regards to my guide Mr. Rajiv Kalita (Assistant Professor, CS & IT dept., JIST) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Ms. Jameson Mushahary (Head of Dept., CS & IT dept., JIST) for his cordial support, valuable information and guidance which helped me in completing this task through various stages.

I am obliged to staff members of AISAT College, for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Lastly, I thank almighty, my parents and my classmates for their constant encouragement without which this assignment would not have been possible.

Bijoy Dutta

Rajkumar Rocktim Narayan Singha

BSc-IT (6<sup>th</sup> Semester)

# Index

---

1. Introduction .....	1
▪ Purpose	
▪ Objectives	
▪ Scope	
▪ Achievements	
2. Survey of Technology .....	4
▪ Android	
▪ JSON	
▪ SQLite	
3. System Analysis .....	13
▪ Existing System	
▪ Proposed System	
▪ Constrains	
▪ Assumptions	
4. Feasibility Study .....	15
▪ Operational Feasibility	
▪ Technical Feasibility	
▪ Economical Feasibility	
5. System Requirements & Specifications (SRS) .....	16
▪ Functional Requirements	
▪ Non-Functional Requirements	
▪ External Interface Requirement	
6. Project Modules .....	19
7. Data Flow Diagram .....	20
8. Entity Relationship Diagram .....	23
9. Relational Model .....	24
10.Screenshots .....	25
11.Codes .....	29
12.Testing Approaches .....	40
13.Conclusion .....	43
14.Bibliography .....	44

---

# Introduction

---

The era of mobile technology opens the windows to the android app. The websites are vanishing and the mobile phones are emerging. It's the time to change from conventional websites to apps, which has become the part of our daily routine. In the scenario of the assignment, we were required to develop an android application on Online Job Portal. So, we introduce 'Job Vibe' the android application.

Job Vibe acts as a portal between Job Seekers and Recruiters. It is helpful for fresher as well as young graduates, to get started with their career. Using Job Vibe, one can search job matching his or her qualification and appear for the selection process once the application is being registered and accepted. It can be very helpful as it allows users of different profile to upload their educational details and search matching job according to their qualification. Also, every user can apply for multiple job at a time.

Job Vibe connects jobseekers and recruiters by accurately matching candidate profile to the relevant job openings through an advanced 2-way matching technology. While most online job portals focus on getting candidates the next job, Job Vibe focuses on the entire career growth of the candidate.

## **(a) Objective:**

The primary objective of Job Vibe is to provide a platform for young graduates and fresher to find the right and satisfactory job according to their qualification, to provide job opportunities to the job seekers, to meet manpower requirements of Industries and Projects in corporate sectors and to provide live vacancies.

Job Vibe is designed with a view to bring the right job for the talented and bright candidates. It is developed with an eye to bridge the gap between talent and opportunities and offers end-to-end recruitment solutions.

For individuals thinking about new job, new career, or new direction, Job Vibe helps you explore the possibilities and find the opportunities that are right for you.

## **(b) Purpose:**

The primary purpose of the online job portal is to works closely to bridge the gap between talent and opportunities and offers end-to-end recruitment solutions and to bring candidates and top employers under one roof.

This online job portal is prepared for providing all categories of job and help to get various type of job. It enables applicants to search for jobs in a conventional manner and easily apply for them. Job Vibe intends to bring the right job to the right candidate.



### **(c)Scope:**

- Jobseekers will have an easy access to the different job opportunities.
- Jobseekers can find job according to their qualification, location or as per their field of interest.
- Jobseekers can apply for multiple jobs at a time.
- Job seekers can directly contact the organization regarding any query about the job.

### **(d) Achievements:**

By successfully implementing the project: -

- A substantial knowledge has been acquired on the implementation of Android technologies.
- This knowledge will be useful in the future in creating any type of Android application.
- Team Management Skills: Roles and Responsibility.
- Time Bound: Specific planned dates for when results will be achieved

# Survey of Technology

---

In an Android application like Job Portal, there is a scope for a large number of platforms, languages and frameworks to choose from. Before selecting from this large array of technologies, the following aspects, which are characteristic to android based application like this one, have been kept in mind:

- Data validation
- Performance
- Reliability
- Scalability
- Security
- Portability
- Performance
- Time constraint
- Cost constraint

The various technologies available for consideration are as follows:

**1.Android**

**2.JSON**

**3.SQLite**



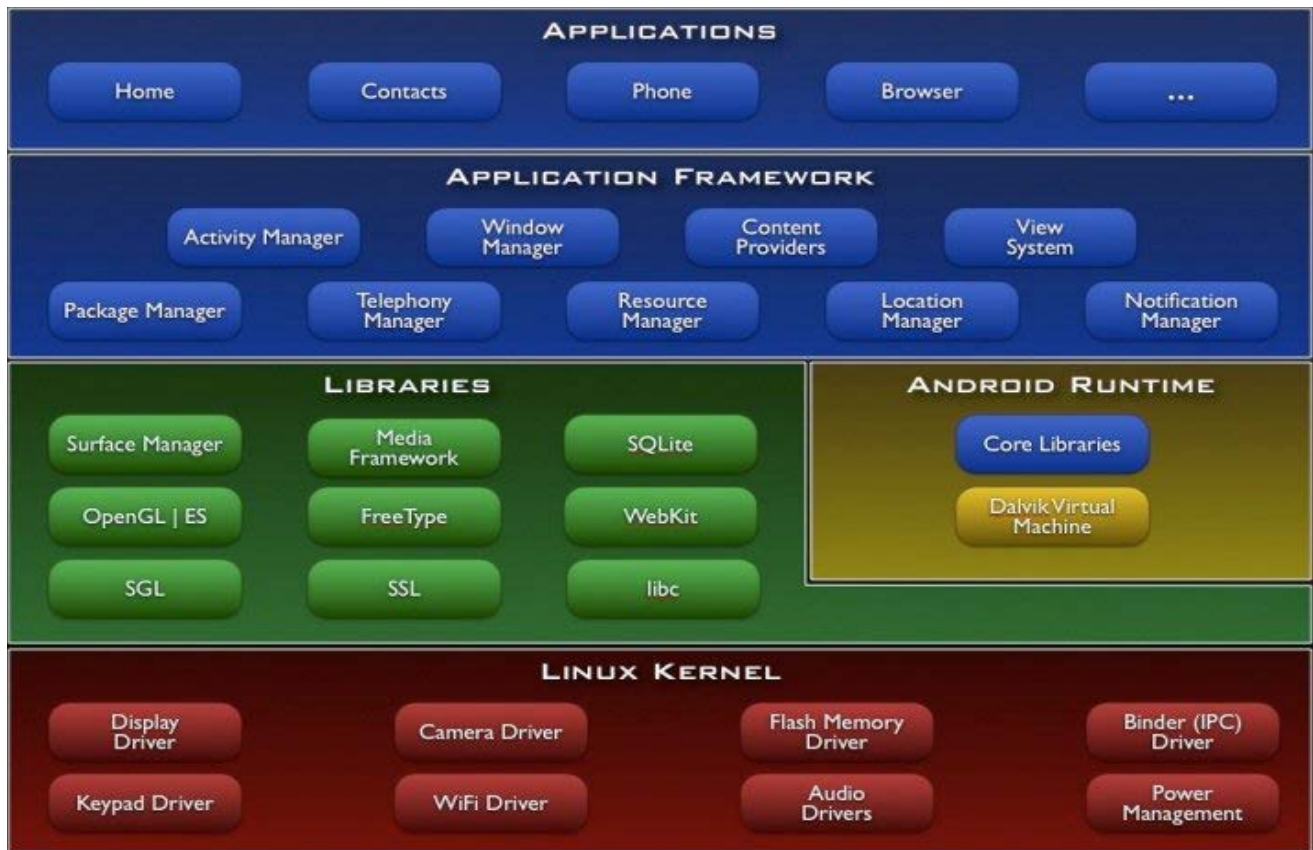
## **WHAT IS ANDROID: -**

- It is an open source software platform and operating system for mobile devices
- Based on the Linux kernel
- Developed by Google and later the Open Handset Alliance (OHA)
- Allows writing managed code in the Java language
- Android has its own virtual machine i.e. DVM (Dalvik Virtual Machine), which is used for executing the android application.
- Android is a free downloadable open source software stack for mobile devices that include an Operating system.
- Android OS is developed under a code name based on dessert items.

## **ANDROID ARCHITECTURE: -**

The software stack is split into Four Layers: -

- The application layer
- The application framework
- The libraries and runtime
- The kernel



## **LINUX KERNEL: -**

- The architecture is based on the Linux2.6 kernel.
- This layer is core of android architecture. It provides service like power management, memory management, security etc.
- It helps in software or hardware binding for better communication.

## **NATIVE LIBRARIES: -**

- Android has its own libraries, which is written in C/C++. These libraries cannot be accessed directly. With the help of application framework, we can access these libraries. There are many libraries like web libraries to access web browsers, libraries for android and video formats etc.

## **Android Run Time: -**

- **Dalvik virtual machine-** The Android Runtime was designed specifically for Android to meet the needs of running in an embedded environment where you have limited battery, limited memory, limited CPU.
- Dalvik is the process virtual machine in Google's android operating system. It is the software that runs the apps on android devices. Dalvik is thus an integral part of android, which is typically used on mobile devices such as mobile phones and tablet computers.
- Programs are commonly written in java and compiled to byte code.
- **Core libraries-** This is in blue, meaning that it's written in the Java programming language.
- The core library contains all of the collection classes, utilities, IO, all the utilities and tools that you've come to expected to use.

## **Application Framework: -**

- This is all written in a Java programming language and the application framework is the toolkit that all applications use.
- These applications include the ones that come with a phone like the home applications, or the phone application.
- It includes applications written by Google, and it includes apps that will be written by you.
- So, all apps use the same framework and the same APIs.

These are as follows: -

- **Activity manager:**-It manages the lifecycle of applications. It enables proper management of all the activities. All the activities are controlled by activity manager.
- **Resource manager:**-It provides access to non-code resources such as graphics etc.
- **Notification manager:**-It enables all applications to display custom alerts in status bar.
- **Location manager:** - It fires alerts when user enters or leaves a specified geographical location.
- **Package manager:**-It is use to retrieve the data about installed packages on device.
- **Window manager:**-It is use to create views and layouts.
- **Telephony manager:**-It is use to handle settings of network connection and all information about services on device.

**APPLICATION LAYER:** -The final layer on top is Applications.

- It includes the home application, the contacts application, the browser, and apps.
- It is the most upper layer in android architecture.
- All the applications like camera, Google maps, browser, sms, calendars, contacts are native applications. These applications work with end user with the help of application framework to operate.

## **SECURITY**

- Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications.

- Android is designed having multi-layer security which provides flexibility for this platform. When attackers attempt attack on device, android platform helps to reduce the portability of the attack.

There are key components of android security which are described as follows: -

- Design review: -when a security model is designed then it will be reviewed by the developers so that risk level will be very less while using the model.
- Code review and penetrating testing: -the goal of this code review is that in which it will be checked that how the system will become strong?
- Open source and community review: -android uses open source technologies that have significant external review such as Linux kernel.
- Incident response: -android team enables the rapid mitigation of vulnerabilities to ensure that potential risks to all android users are minimized.

## **JSON (JavaScript Object Notation)**

JavaScript Object Notation (JSON) is an open standard data exchange format based on a JavaScript syntax subset. JSON is text-based, lightweight, and generally considered easily readable/writeable.

Although closely connected to JavaScript, JSON is language-independent. Though independent, JSON uses conventions similar to other languages (e.g., C, C++, Java, Perl and Python), making JSON an ideal data-exchange language.

JSON characteristics include the following:

- Flexibility, allowing the programmer to define keys.
- Less overhead, as content is mostly data.
- Portable data.
- Non-proprietary.
- Common and convenient format for Web services.

Commonly utilized in Web application development, JSON may be used as a data format for any application where information is stored as text.

JSON is preferred by some over XML as a data exchange format because it is less verbose, works quickly, reduces data size and simplifies document processing. It is used extensively in web development, especially because it seamlessly transfers information between potentially incompatible technologies. For example, it could involve a Java application running on a UNIX box or a C# application running on Windows.

## SQLite

In the simplest terms, SQLite is a public-domain software package that provides a *relational database management system*, or RDBMS. Relational database systems are used to store user-defined records in large tables. In addition to data storage and management, a database engine can process complex query commands that combine data from multiple tables to generate reports and data summaries. Other popular RDBMS products include Oracle Database, IBM's DB2, and Microsoft's SQL Server on the commercial side, with MySQL and PostgreSQL being popular open source products.

The “Lite” in SQLite does not refer to its capabilities. Rather, SQLite is lightweight when it comes to setup complexity, administrative overhead, and resource usage. SQLite is defined by the following features:

### **Server less**

SQLite does not require a separate server process or system to operate. The SQLite library accesses its storage files directly.

### **Zero Configuration**

No server means no setup. Creating an SQLite database instance is as easy as opening a file.

### **Cross-Platform**

The entire database instance resides in a single cross-platform file, requiring no administration.

### **Self-Contained**

A single library contains the entire database system, which integrates directly into a host application.



## **Small Runtime Footprint**

The default build is less than a megabyte of code and requires only a few megabytes of memory. With some adjustments, both the library size and memory use can be significantly reduced.

## **Transactional**

SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.

## **Full-Featured**

SQLite supports most of the query language features found in the SQL92 (SQL2) standard.

## **Highly Reliable**

The SQLite development team takes code testing and verification very seriously.

Overall, SQLite provides a very functional and flexible relational database environment that consumes minimal resources and creates minimal hassle for developers and users.

# System Analysis

---

## Existing System:

- The existing system requires applicants to search through print and visual media for job opportunities.
- Applicants need to apply for jobs using conventional methods and wait for further details for a long period of time.
- Employers need to advertise the vacancies and sort all applicant details, conduct selection procedures and complete the formalities.
- This approach is tedious and requires much effort and resources.

## Proposed System:

- Simple and professional GUI for all qualification groups.
- All vacancies are available on a single interface.
- Job Seekers can save jobs according to their requirement.
- Reduce paper work and extra cost.
- Filer and search facility for job seekers according to their requirement.
- Applicant gets instant response for any query through notification and other media.
- Easy posting of job vacancies by the employers
- Applicants can update their personal and educational details which helps in better filtering of matching jobs.

### **Constraints:**

- The interface is only provided in English hence the user must know English in order to use the application.
- Login and password is used for identification of the user and there is no facility for guest users.
- Only registered users have the rights to access the facilities provided by the system.

### **Assumptions:**

- The user has intermittent knowledge of android devices and its UI.
- As the system is an online system, the user must have an active internet connection on the device.
- The user knows English as the GUI is provided in English.
- The Job seeker has a degree/diploma in some recognized field of study.

# Feasibility Study

---

It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time. There may be different ways of checking whether a system is feasible or not. The following feasibility studies were performed to gauge the feasibility of the system.

## **Operational Feasibility:**

In this test, the operational scope of the system is checked. The system under consideration should have enough operational reach. It is observed that the proposed system is very user friendly and since the system is built with enough help, even persons with little knowledge of android can find the system very easy.

## **Technical Feasibility:**

This test includes a study of function, performance and constraints that may affect the ability to achieve an acceptable system. This test begins with an assessment of the technical viability of the proposed system. One of the main fusers to be accessed is the need of various kinds of resources for the successful implementation for the proposed system.

## **Economic Feasibility:**

An evaluation of development cost weighed against the ultimate income or benefit derived from the development of the proposed system is made. Care must be taken that incurred in the development of the proposed of the system should not exceed from the system. The income can be in terms of money or goodwill, since the software brings in both, the system is highly viable.

# **System Requirements & Specifications(SRS)**

System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

The software specification document satisfies the following: -

- It specifies the external system behaviors.
- It specifies constraints on the implementation.
- It is easy to change.
- It serves as reference tool for system maintainers.
- It records forethought about the life cycle of the system.
- It characterizes acceptable response to undesired events.

## **Functional Requirements:**

- The System must provide following functionalities—
- Keeping records of registration of candidates.
- Keeping the record of jobs.
- Keeping the employer details.
- Storing the record for applied job by the candidates.
- Storing the job saved by the candidates.

## **Non Functional Requirements:**

Following Non-functional requirements will be there in the online shopping portal.

- Secure access of confidential data (candidate's details).
- 24 X 7 availability.
- Better component design to get better performance at peak time.

Flexible service based architecture will be highly desirable for future extension.

Nonfunctional requirements define system properties and constraints. It arises through user needs, because of budget constraints or organizational policies, or due to the external factors such as safety regulations, privacy registration and so on.

Various other Non-functional requirements are:

1. Security
2. Reliability
3. Maintainability
4. Portability
5. Extensibility
6. Reusability
7. Application Affinity/Compatibility
8. Resource Utilization

## **External Interface Requirements:**

### **User Interface:**

User of the system will be provided with the Graphical user interface, there is no command line interface for any functions of the product.

### **Hardware Interface:**

The application will run on an Android emulator or an Android mobile device with the following specifications: -

Processor: - 1.2GHz or above.

RAM: - 2GB or above.

Internal Storage: - 4GB or above.

### **Software Interface: -**

The software will run on the Android operating system, specifically version 5.0 (Lollipop) and above.

## **Tools and Languages Used**

### **Language Used: -**

- Java
- XML
- SQL

### **Software Used: -**

- Android Studio
- GitHub
- Balsamiq Mockups
- Adobe Photoshop



# Project Module

---

## Sign Up

---

The 'Sign up' activity allows the user to register himself in Job Vibe. It is to be noted that there is no option for 'Guest Login' in the application. Hence, any user willing to access the facilities provided by the application must register using proper information. During the signup process, the user must first enter a valid email address followed by a password of his/her choice (minimum 8 characters). After this process is completed, the user will be asked to fill some additional information regarding his/her personal and education details. Once this process is done, the user is directed to the 'Home' activity.

## Login

---

The 'Login' facility is added for user authentication purpose. It helps to verify the credentials of an authorized user. In this activity the user needs to enter the proper credentials, matching the credentials entered during 'Signup' to access the rest of the functionalities.

## Home

---

This is first layout the user interacts with after his/her registration is completed. Here he/she will find two tabs headed as 'Matched' and 'Recommended'. In 'Matched' the user will see jobs according to his qualification whereas in 'Recommended' he/she will see all the available jobs. The user can also filter the jobs according to his/her Skill, Location and preferred Company. He/she can also search job according to desired field of work, say for example Mobile App Development, Web Developer, etc.

## Activity

---

The 'Activity' layout keeps track of user's activity within the app. It is divided into three tabs headed as 'Viewed', 'Saved' and 'Applied'. In 'Viewed', the user can find all the jobs he/she has been viewing recently. In 'Saved', the user can find the job he/she has saved for later. In 'Applied', the user can find the jobs he/she has applied for.

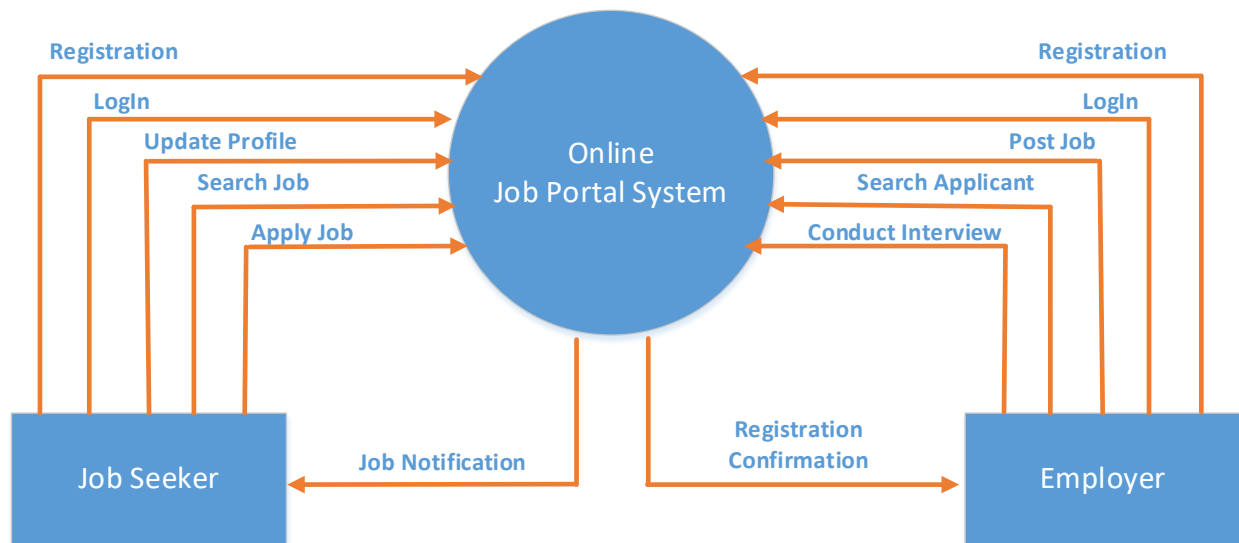
## Notification

---

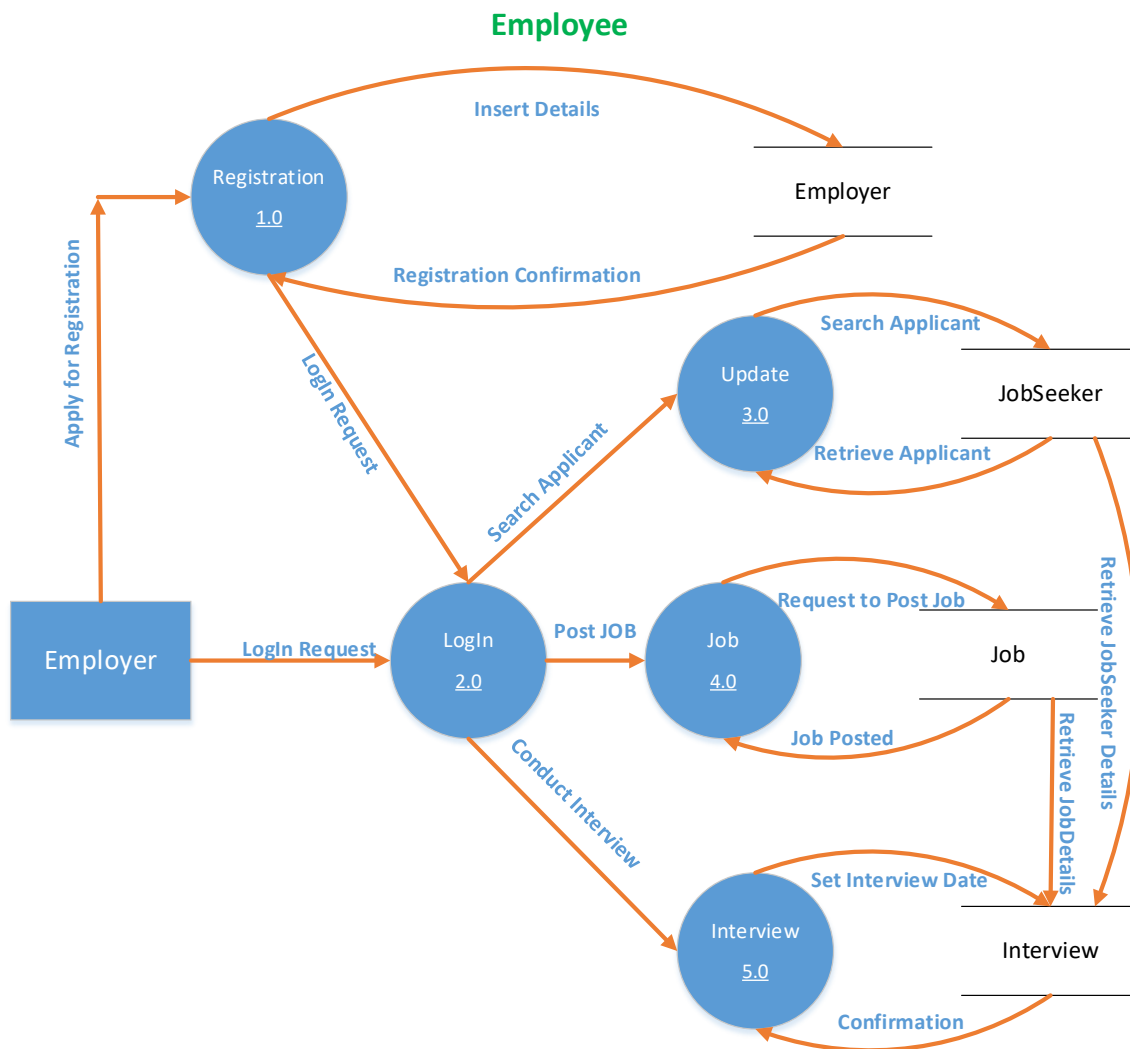
The 'Notification' layout notifies the user once a job is successfully applied.

# Data Flow Diagram

---

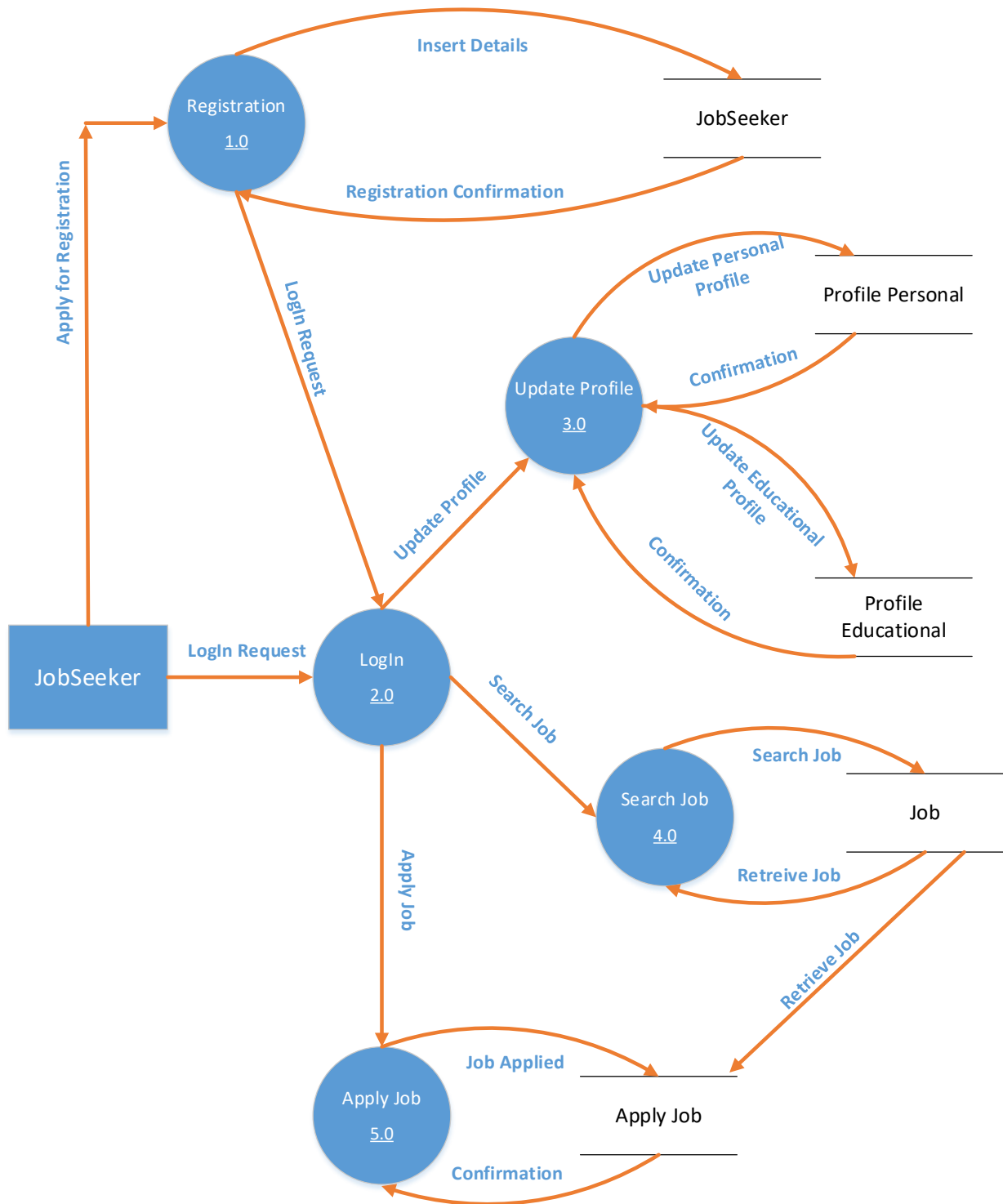


Context Level



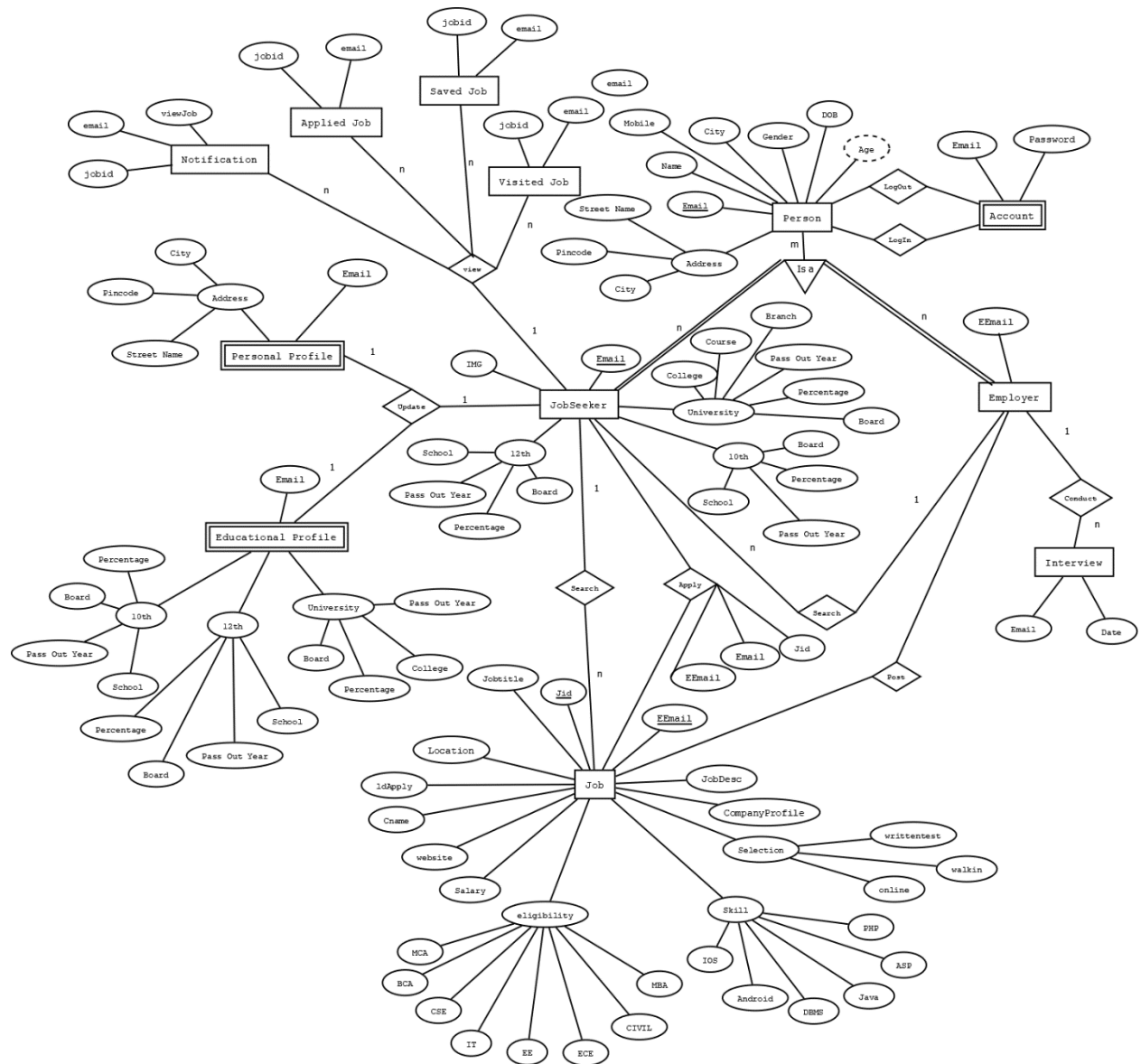
**Level 1 DFD**

## Job Seeker



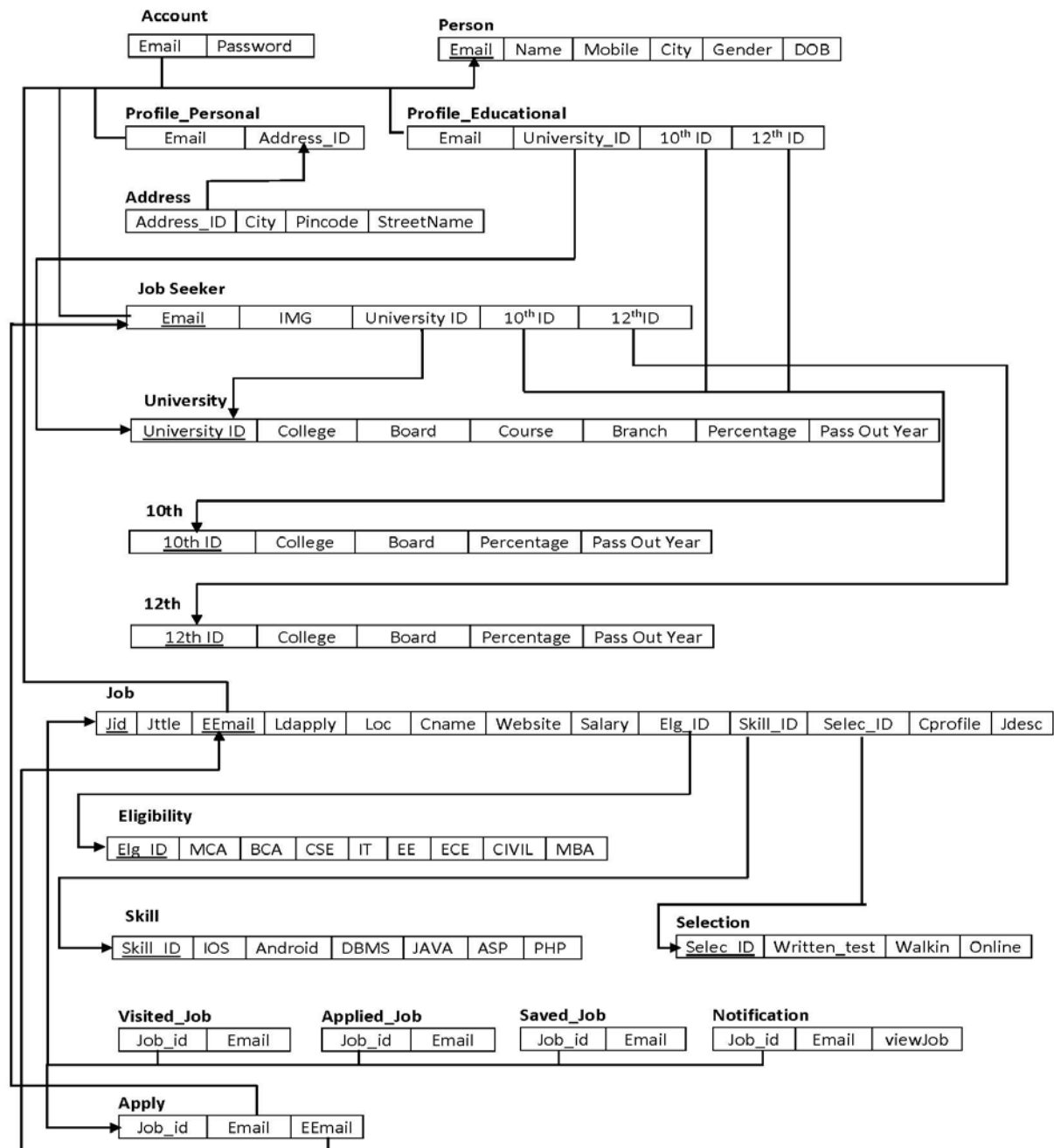
## Level 1 DFD

# Entity Relationship Diagram



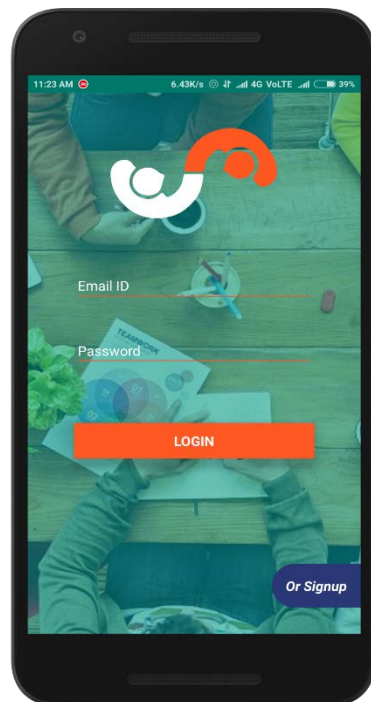
# Relational Model

## Relational Model



## Screenshots

---





11:23 AM 1.38K/s 4G VoLTE 39%

Email ID

Password

Confirm Password

SIGNUP

Or Login

5:19 PM 1.88K/s 4G VoLTE 39%

### Advance Information

**Personal Details** Educational Details

Email:  
arjun@gmail.com

Name

Mobile

Address

Pin Code

Current City

Gender  
☒ Male ☐ Female

Date of Birth

Personal Details

5:19 PM 0.26K/s 4G VoLTE 39%

### Advance Information

Personal Details **Educational Details**

**Current Education**

University/Board

College

Year of Completion

Percentage

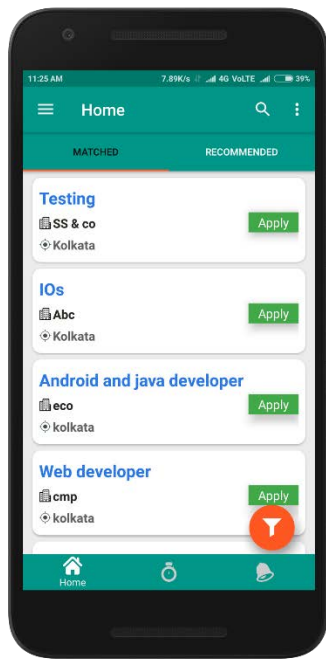
**12th Education**

Board

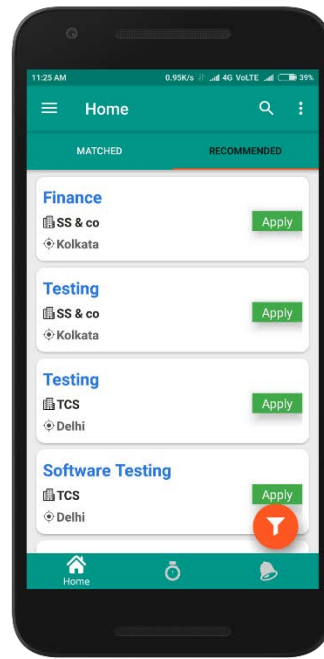
School

Year of Completion

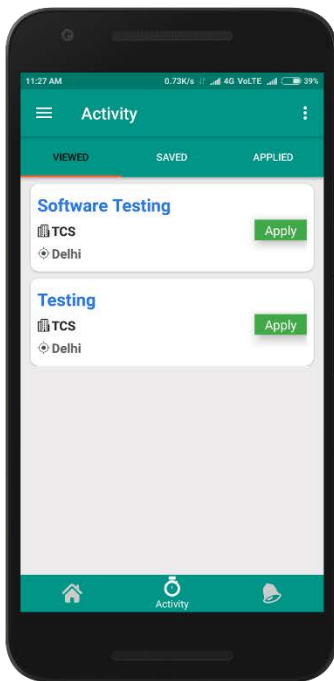
Educational Details



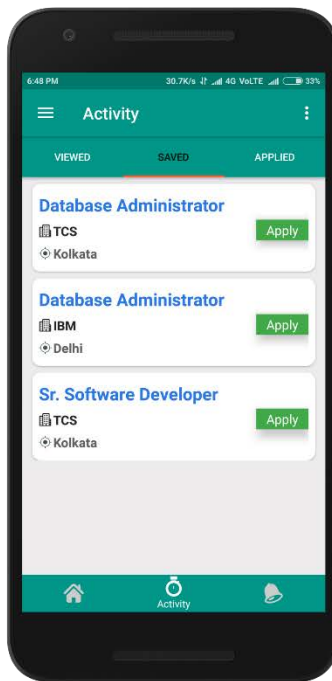
Matched Job



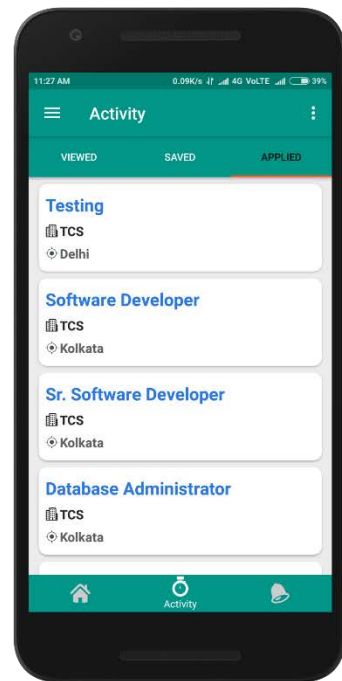
Recommended Job



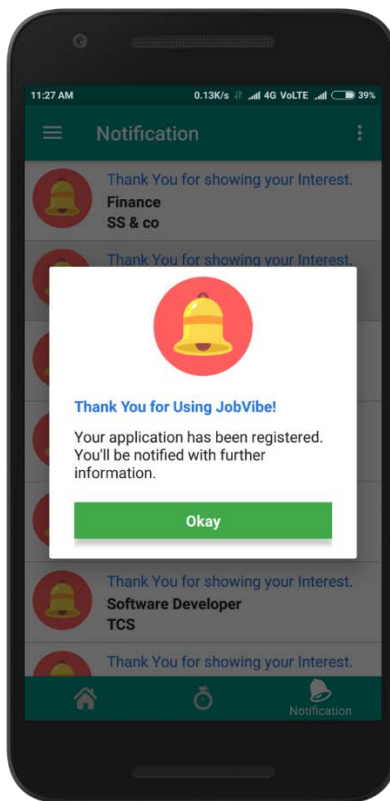
Viewed Job



Saved Job



Applied Job



# Codes

---

## Some XML Codes

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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.support.v4.widget.DrawerLayout
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:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

        <include
            layout="@layout/app_bar_home"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <android.support.design.widget.NavigationView
            android:id="@+id/nav_view"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:fitsSystemWindows="true"
            app:headerLayout="@layout/nav_header_home"
            app:itemIconTint="@color/orange"
            app:menu="@menu/activity_home_drawer">

            </android.support.design.widget.NavigationView>

        </android.support.v4.widget.DrawerLayout>

    </android.support.design.widget.CoordinatorLayout>

<?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"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".Home"
    tools:showIn="@layout/app_bar_home">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```

        android:id="@+id/AL_appbar">

        <android.support.design.widget.TabLayout
            android:id="@+id/TL_tab"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:tabTextColor="@color/white">

        </android.support.design.widget.TabLayout>

    </android.support.design.widget.AppBarLayout>

    <android.support.v4.view.ViewPager
        android:id="@+id/VP_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/AL_appbar"
    >

    </android.support.v4.view.ViewPager>

    <LinearLayout
        android:id="@+id/RL_buttons"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:background="@color/colorPrimary"
        android:padding="2dp">

        <LinearLayout
            android:id="@+id/layoutHome"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:orientation="vertical"
            android:gravity="center"
            android:layout_gravity="center">

            <ImageView
                android:id="@+id/IV_home"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:src="@drawable/ic_home"/>

            <TextView
                android:id="@+id/TV_home"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Home"
                android:visibility="gone"
                android:textColor="@color/white"/>

        </LinearLayout>

    <LinearLayout
        android:id="@+id/layoutActivity"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical"
        android:gravity="center"

```

```

        android:layout_gravity="center">

        <ImageView
            android:id="@+id/IV_activity"
            android:layout_toRightOf="@+id/IV_home"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_activity"/>

        <TextView
            android:id="@+id/TV_activity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Activity"
            android:visibility="gone"
            android:textColor="@color/white"/>

    </LinearLayout>

    <LinearLayout
        android:id="@+id/layoutNotification"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical"
        android:gravity="center"
        android:layout_gravity="center">

        <ImageView
            android:id="@+id/IV_notification"
            android:layout_toRightOf="@+id/IV_Activity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_notification"/>

        <TextView
            android:id="@+id/TV_notification"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Notification"
            android:visibility="gone"
            android:textColor="@color/white"/>
    </LinearLayout>

</LinearLayout>

</RelativeLayout>

```

## Some Java Codes

```
package br.developers.com.jobvibe;

import android.Manifest;
import android.annotation.SuppressLint;

import android.app.AlertDialog;
import android.content.DialogInterface;

import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.net.Uri;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.support.annotation.NonNull;
import android.support.design.widget.AppBarLayout;
import android.support.design.widget.Snackbar;
import android.support.design.widget.TabLayout;

import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentStatePagerAdapter;
import android.support.v4.view.ViewPager;
import android.text.Html;
import android.util.Log;
import android.view.View;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import java.io.ByteArrayOutputStream;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class Home extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener,
    View.OnClickListener {
```



```

private TextView
matched, recommended, viewed, saved, applied, tv_home, tv_activity, tv_notification, tv_emphnam
e, tv_emppemail;
private ImageView iv_home, iv_activity, iv_notification, iv_profileImage;
public static String canemail;
public static String name, getdegree, getfos;
private boolean onbackpressed=false;
private TabLayout tabLayout;
private ViewPager viewPager;
private final int REQUEST_CODE_GALLERY=999;
private Toolbar toolbar;
private LinearLayout layoutHome, layoutActivity, layoutNotify;
public static LinearLayout layoutbottom;

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

    iv_home=findViewById(R.id.IV_home);
    iv_activity=findViewById(R.id.IV_activity);
    iv_notification=findViewById(R.id.IV_notification);
    tv_home=findViewById(R.id.TV_home);
    tv_activity=findViewById(R.id.TV_activity);
    tv_notification=findViewById(R.id.TV_notification);
    tabLayout=findViewById(R.id.TL_tab);
    viewPager=findViewById(R.id.VP_view);
    layoutHome=findViewById(R.id.layoutHome);
    layoutActivity=findViewById(R.id.layoutActivity);
    layoutNotify=findViewById(R.id.layoutNotification);
    layoutbottom=findViewById(R.id.RL_buttons);

    tabLayout.setupWithViewPager(viewPager);
    setterViewPager(viewPager);

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

    iv_home.setOnClickListener(this);
    iv_activity.setOnClickListener(this);
    iv_notification.setOnClickListener(this);
    tv_home.setOnClickListener(this);
    tv_activity.setOnClickListener(this);
    tv_notification.setOnClickListener(this);
    layoutHome.setOnClickListener(this);
    layoutActivity.setOnClickListener(this);
    layoutNotify.setOnClickListener(this);

    canemail=getIntent().getStringExtra("emailid");
    name=getIntent().getStringExtra("name");
    getdegree= getIntent().getStringExtra("getdegree");
    getfos=getIntent().getStringExtra("getfos");

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
        this, drawer, toolbar, R.string.navigation_drawer_open,
        R.string.navigation_drawer_close);
    drawer.addDrawerListener(toggle);
    toggle.syncState();
}

```

```

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);

View headerView = navigationView.getHeaderView(0);
tv_empname = (TextView) headerView.findViewById(R.id.TV_profileName);
tv_empemail=(TextView)headerView.findViewById(R.id.TV_profileEmail);
iv_profileImage=headerView.findViewById(R.id.imageView);

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

        ActivityCompat.requestPermissions(Home.this,new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},REQUEST_CODE_GALLERY);

    }
});

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

    }
});

tv_empname.setText(name);
tv_empemail.setText(canemail);

navigationView.setNavigationItemSelectedListener(this);

iv_home.setImageResource(R.drawable.ic_onhome);
tv_home.setVisibility(View.VISIBLE);
iv_activity.setImageResource(R.drawable.ic_activity);
iv_notification.setImageResource(R.drawable.ic_notification);

loadProfilePic();
}

private void loadProfilePic() {

    DBManager db=new DBManager(this);
    byte[] byteimg=db.getImage(Home.canemail);
    if(byteimg!=null){
        Bitmap bitimg=BitmapFactory.decodeByteArray(byteimg, 0, byteimg.length);
        try{
            iv_profileImage.setImageBitmap(bitimg);
        }
        catch (Exception ex)
        {
            Log.d("logcheck","exception "+ex);
        }
    }
}

//Permission menu for access gallery or camera
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {

    if(requestCode==REQUEST_CODE_GALLERY){
        if(grantResults.length>0 && grantResults[0]==
PackageManager.PERMISSION_GRANTED){

```

```

        Intent intent=new Intent(Intent.ACTION_PICK);
        intent.setType("image/*");
        startActivityForResult(intent,REQUEST_CODE_GALLERY);
    }
    else{
        Toast.makeText(this, "You don't have permission to access file!",
Toast.LENGTH_SHORT).show();
    }
}

super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}

//Setting image to image View after permission granted
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {

    if(requestCode==REQUEST_CODE_GALLERY && resultCode==RESULT_OK && data!=null) {

        try {
            Uri uri = data.getData();
            InputStream inputStream = getContentResolver().openInputStream(uri);
            Bitmap bitmap = BitmapFactory.decodeStream(inputStream);
            iv_profileImage.setImageBitmap(bitmap);
            addImgToDb(bitmap);
        } catch (FileNotFoundException ex) {
            ex.printStackTrace();
        }
    }
    super.onActivityResult(requestCode, resultCode, data);
}

//Adding image to database
private void addImgToDb(Bitmap bitmap) {

    byte[] profieimg=imageViewtoByte(iv_profileImage);
    DBManager db=new DBManager(this);
    boolean bol=db.isImgExists(Home.canemail);
    if(!bol){
        db.insertImage(Home.canemail,profieimg);
    }
    else if(bol){
        db.updateImage(Home.canemail,profieimg);
    }
}

//Converting image to byte
private byte[] imageViewtoByte(ImageView iv_profileImage) {

    Bitmap bitmap=((BitmapDrawable)iv_profileImage.getDrawable()).getBitmap();
    ByteArrayOutputStream byteArrayOutputStream=new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG,80,byteArrayOutputStream);
    byte[] bytearray=byteArrayOutputStream.toByteArray();
    return bytearray;
}

private void setterViewPager(ViewPager viewPager) {

```

```

        ViewPager viewPager=new ViewPager(getSupportFragmentManager());
        viewPager.addFragment(new MatchedFragment(),"Matched");
        viewPager.addFragment(new Recommended(),"Recommended");
        viewPager.setAdapter(viewAdapter);
    }

    private void setterViewPagerActivity(ViewPager viewPager) {

        ViewPager viewPager=new ViewPager(getSupportFragmentManager());
        viewPager.addFragment(new Viewed_Fragment(),"Viewed");
        viewPager.addFragment(new Saved_Fragment(),"Saved");
        viewPager.addFragment(new Applied_Fragment(),"Applied");
        viewPager.setAdapter(viewAdapter);
    }

    private void setterViewPagerNotification(ViewPager viewPager) {

        ViewPager viewPager=new ViewPager(getSupportFragmentManager());
        viewPager.addFragment(new NotificationFragment(),"Notification");
        viewPager.setAdapter(viewAdapter);
    }

    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            Log.d("logcheck", "drawer");
            drawer.closeDrawer(GravityCompat.START);
        }
        else if(!onBackPressed){

            View v=findViewById(android.R.id.content);
            Snackbar.make(v, "Press back again to exit
            "+Html.fromHtml("&#9995;"), Snackbar.LENGTH_SHORT).show();

            onBackPressed=true;
        }
        else{
            super.onBackPressed();
        }
        new CountDownTimer(3000,1000){

            @Override
            public void onTick(long millisUntilFinished) {

            }

            @Override
            public void onFinish() {
                onBackPressed=false;
            }
        }.start();
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        if (id == R.id.nav_editProfile) {
            Intent editProfile = new Intent(Home.this, EditActivity.class);

```

```

        startActivity(editProfile);
        overridePendingTransition(R.anim.topttobottom,R.anim.bottomtotop);
        finish();
    } else if (id == R.id.nav_faq) {
        Intent intent=new Intent(Home.this,FAQ.class);
        startActivity(intent);
        overridePendingTransition(R.anim.topttobottom,R.anim.bottomtotop);
        finish();

    } else if (id == R.id.nav_terms) {
        Intent intent=new Intent(Home.this,Terms.class);
        startActivity(intent);
        overridePendingTransition(R.anim.topttobottom,R.anim.bottomtotop);
        finish();

    } else if (id == R.id.nav_report) {
        AlertDialog.Builder report=new AlertDialog.Builder(Home.this);
        View reportView=getLayoutInflater().inflate(R.layout.problem_layout,null);
        EditText problem=reportView.findViewById(R.id.ET_subject);

        report.setPositiveButton("Submit", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast.makeText(Home.this,"Report Successfully
Sent",Toast.LENGTH_LONG).show();

            }
        });
        report.setNegativeButton("Abort", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                dialogInterface.cancel();
            }
        });
        report.setView(reportView);
        report.show();

    } else if (id == R.id.nav_share) {

        Intent share=new Intent(Intent.ACTION_SEND);
        share.setType("text/plain");
        String body="Download Job Vibe!";
        String sub="The best Job Portal app in India.";
        share.putExtra(Intent.EXTRA_TEXT,body);
        share.putExtra(Intent.EXTRA_SUBJECT,sub);

        share.putExtra(Intent.EXTRA_TEXT,"https://drive.google.com/file/d/1bpu0_9l52LilIhLLOE4
8_xTYQEdHBYvC/view?usp=sharing");
        startActivity(Intent.createChooser(share,"Share Using"));

    }
    else if (id == R.id.nav_about) {
        Intent intent=new Intent(Home.this,AboutUs.class);
        startActivity(intent);
        overridePendingTransition(R.anim.topttobottom,R.anim.bottomtotop);
        finish();

    }
    else if(id==R.id.nav_logout){

        SharedPreferences

```

```

sharedPreferences=getSharedPreferences("Data",MODE_PRIVATE);
    SharedPreferences.Editor editor=sharedPreferences.edit();
    editor.putString("email","");
    editor.putString("password","");
    editor.commit();

    Intent login=new Intent(Home.this,Login.class);
    startActivity(login);
    finish();

}

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
}

@SuppressLint("ResourceAsColor")
@Override
public void onClick(View v) {

    if (v.getId()==R.id.IV_home || v.getId()==R.id.TV_home ||
v.getId()==R.id.layoutHome) {
        iv_home.setImageResource(R.drawable.ic_onhome);
        tv_home.setVisibility(View.VISIBLE);
        tv_activity.setVisibility(View.GONE);
        tv_notification.setVisibility(View.GONE);
        iv_activity.setImageResource(R.drawable.ic_activity);
        iv_notification.setImageResource(R.drawable.ic_notification);

        toolbar.setTitle(R.string.title_activity_home);

        tabLayout.setVisibility(View.VISIBLE);
        tabLayout.setupWithViewPager(viewPager);
        viewPager.setViewPager(viewPager);

    } else if (v.getId()==R.id.IV_activity || v.getId()==R.id.TV_activity ||
v.getId()==R.id.layoutActivity) {
        iv_activity.setImageResource(R.drawable.ic_onactivity);
        tv_activity.setVisibility(View.VISIBLE);
        tv_notification.setVisibility(View.GONE);
        tv_home.setVisibility(View.GONE);
        iv_home.setImageResource(R.drawable.ic_home);
        iv_notification.setImageResource(R.drawable.ic_notification);

        toolbar.setTitle(R.string.title_activity_activity);

        tabLayout.setVisibility(View.VISIBLE);
        tabLayout.setupWithViewPager(viewPager);
        viewPager.setViewPagerActivity(viewPager);

    } else if (v.getId()==R.id.IV_notification || v.getId()==R.id.TV_notification
|| v.getId()==R.id.layoutNotification) {
        iv_notification.setImageResource(R.drawable.ic_onnotification);
        tv_notification.setVisibility(View.VISIBLE);
        tv_home.setVisibility(View.GONE);
        tv_activity.setVisibility(View.GONE);
        iv_home.setImageResource(R.drawable.ic_home);
        iv_activity.setImageResource(R.drawable.ic_activity);

        toolbar.setTitle(R.string.title_activity_notification);
    }
}

```

```

        tabLayout.setVisibility(View.GONE);
        setContentView(viewPager);
    }
}

public class ViewAdapter extends FragmentStatePagerAdapter{

    private List<Fragment> toplist=new ArrayList<>();
    private List<String>titlelist=new ArrayList<>();

    public ViewAdapter(android.support.v4.app.FragmentManager fm) {
        super(fm);
    }

    @Override
    public android.support.v4.app.Fragment getItem(int position) {
        return toplist.get(position);
    }

    @Override
    public int getCount() {
        return toplist.size();
    }

    @Override
    public CharSequence getPageTitle(int position) {
        return titlelist.get(position);
    }

    public void addFragment(Fragment fragment, String string){

        toplist.add(fragment);
        titlelist.add(string);
    }
}
}

```

# Testing Approaches

---

## TESTING PROCEDURES

- **Unit Testing:** A Unit corresponds to a form/class in the package. Unit testing focuses on verification of the corresponding form or class. In this level we have tested all our forms/classes individually. This testing includes testing of control paths, interfaces, local data structures, logical decisions, boundary conditions, and error handling. From this testing we were able to save, retrieve, update, delete and the search records on a table.
- **Integration Testing:** Integration testing is used to verify the combination of the software modules. In this level, we have tested by combining all unit tested forms into a subsystem. Here we found that the subsystems are performing well.
- **System Testing:** System testing is used to verify, whether the developed system meets the requirements.
- **Acceptance Testing:** Acceptance is the part of the project by which the customer accepts the product. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

We hope that after the acceptance testing the system will perform the best result for the organization. When modification will be made, we will use regression testing during the maintenance of the system. The Application delivered to the customer may undergo changes. Changes may be due to addition of new functional modules or performance enhancement. For this purpose, proper maintenance of the system is must.

### 5.3.1 Unit Testing

Test case Ref No	TCT-001	
Functionality	:	Log in to the System
Expected outcome	:	The user should only login with valid credentials failing which some error message follows



Step No.	Data Used	Actual Outcome
1.	Click on the log in button without entering username or password	An alert message came to enter Username
2.	Click on the log in button after entering some username leaving password field blank	An alert message came to enter Password
3.	Click on the log in button after entering some password but leaving username field blank	An alert message came to enter Username
4.	Click on the log in button after entering some wrong username but correct password	A message displayed on Log in page about this

Test case Ref No	TCT-002	
Functionality	:	Enter valid Data for member signup
Expected outcome	:	The user will not be allowed to register with an email that already exists. Also the password entered by the user must match the confirm password. An alert message will be given to the user if the above cases are not fulfilled.
Step No.	Data Used	Actual Outcome
1.	Click on the signup button With existing email address	An alert message appeared saying that the email id already exists.
2.	Click on the signup button with unmatching passwords	An alert message appeared saying that the entered passwords does not match.

<b>Test case Ref No</b>		<b>TCT-003</b>
	:	
Functionality	:	Applying for a job
	:	
Expected outcome	:	The user will be alerted if the job he/she wants to apply for has already been applied.
Step No.	Data Used	Actual Outcome
1.	Applied for a job that has already been applied before.	An message was displayed saying the user has already applied for the job.

## 6. RESULTS AND DISCUSSION

### 6.1 Test Reports

Test Case No	Date	Pass / Fail
TCT-001	11/5/2018	Pass
TCT-002	02/6/2018	Pass
TCT-003	19/6/2018	Pass

# Conclusion

---

Android is a truly open, free development platform based on Linux and open source. Handset makers can use and customize the platform without paying a royalty. A component-based architecture inspired by Internet mash-ups. Parts of one application can be used in another in ways not originally envisioned by the developer and can even replace built-in components with own improved versions. This will unleash a new round of creativity in the mobile space.

- Android is open to all: industry, developers and users.
- Participating in many of the successful open source projects.
- Aims to be as easy to build for as the web.
- Google Android is stepping into the next level of Mobile Internet.

This Job Portal App acts as a portal between Job Seekers and Recruiters. It is helpful for fresher as well as young graduates, to get started with their career. Using Job Vibe, one can search job matching his or her qualification and appear for the selection process once the application is being registered and accepted.

# Bibliography

---

## 1. YouTube

<https://www.youtube.com/playlist?list=PLonJJ3BVjZW6hYgvtkaWvwAVvOFB7fkLa&pbjreload=10>

## 2. Developer Guides

<https://developer.android.com/guide/>

## 3. Javatpoint

<https://www.javatpoint.com/android-tutorial>

## 4. Stack Overflow

<https://stackoverflow.com/>

# **CERTIFICATE**

---

This is to certify that the project entitles “JOB VIBE: ONLINE JOB PORTAL APP” submitted to Jorhat Institute of Science and Technology, Assam in partial fulfillment of the requirement for the completion of the major project of B.SC in Information Technology Degree. It is an original work carried out by Bijoy Dutta and Rajkumar Rocktim Narayan Singha, B.SC IT 6<sup>th</sup> Semester, under the guidance of Mr. Rajiv Kalita (Assistant Professor), Department of Computer Science and Information Technology, JIST.

The matter embodied in this project is a genuine work done by the students. They have also submitted the project with the report. I have been supervising them throughout the activity and they completed the work to my ultimate satisfaction. This work reported here in has not been presented for other academic purpose elsewhere. This as per requirement for a live project under some organization and it would probably meet the entire essential requisite according to the user needs. Their effort is being appreciated by me and I wish them a bright future.

**Date: -**

**Signature of HOD**

CS&IT Department

JIST, Jorhat, Assam

# **CERTIFICATE**

---

This is to certify that the project entitles “JOB VIBE: ONLINE JOB PORTAL APP” submitted to Jorhat Institute of Science and Technology, Assam in partial fulfillment of the requirement for the completion of the major project of B.SC in Information Technology Degree. It is an original work carried out by Bijoy Dutta and Rajkumar Rocktim Narayan Singha, B.SC IT 6<sup>th</sup> Semester, under the guidance of Mr. Rajiv Kalita (Assistant Professor), Department of Computer Science and Information Technology, JIST.

The matter embodied in this project is a genuine work done by the students. They have also submitted the project with the report. I have been supervising them throughout the activity and they completed the work to my ultimate satisfaction. This work reported here in has not been presented for other academic purpose elsewhere. This as per requirement for a live project under some organization and it would probably meet the entire essential requisite according to the user needs. Their effort is being appreciated by me and I wish them a bright future.

**Date: -**

**Signature of External**