**A MINI PROJECT REPORT**

**On**

**Ishtehaar**

**Submitted by**

**Nishant Kumar Singh**

**Roll No:161500363**

**Rishabh Singh**

**Roll No:161500452**

**Mukesh Kumar Maurya**

**Roll No:161500330**

**To**

**Mr. Neeraj Khanna**

**(Technical Trainer)**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**

****

**GLA University**

**Mathura- 281406, INDIA**

**April, 2019**



**Department of Computer Engineering and Applications GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**

***Declaration***

*We hereby declare that the work which is being presented in the Mini Project* ***“ISHTEHAAR”,****in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of our own work carried under the supervision of* ***Mr. Neeraj Khanna,Technical Trainer( Asstt.proff), GLA University, Mathura****.*

**Name of Students with signature**

**Nishant Kumar Singh**

**Rishabh Singh**

**Mukesh Kumar Maurya**

Department of Computer Engineering & Applications I



**Department of Computer Engineering and Applications GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**

**CERTIFICATE**

*This is to certify that the project entitled* ***“ISHTEHAAR”*** *carried out in Mini*

*Project – II Lab is a bonafide work done by* ***Nishant Kumar Singh(161500363),Rishabh***

***Singh(161500452),Mukesh Kumar Maurya(161500330)*** *and is submitted in partial*

*Fulfillment of the requirements for the award of the degree Bachelor*

*Technology(ComputerScience& Engineering).*

**Signature of Supervisor:**

**Name of Supervisor: Mr.Neeraj Khanna(Technical Trainer)**

**Date:**

Department of Computer Engineering & Applications II

**ACKNOWLEDGEMENT**

*It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.*

*Our heartiest thanks to* ***Dr. (Prof). Anand Singh Jalal,*** *Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.*

*We owe special debt of gratitude to* ***Mr. Neeraj Khanna ,Technical Trainer(****Assistant Professor) ,Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.*

Nishant Kumar Singh

Rishabh Singh

Mukesh Kumar Maurya

Department of Computer Engineering & Applications III

**Abstract**

The purpose of this software is to find the different ability of the students in various fields of technology & cultural activity. Also helps in finding the student of the semester & coder of the semester on the basis of their abilities.

Several, time it happens, there are problems arises that when nominations for student and coder of the semester are going on but the students are not aware of it. That’s why number if participants are is low and to find the students with the different abilities which is not possible manually, so we need a platform where every student can participate & provide their skills. So, we are providing an online platform where students can provide their skills and abilities. Therefore, it is going to be a very simple and automatic process.

Department of Computer Engineering & Applications IV

**Table of Contents**

|  |  |
| --- | --- |
| Declaration | ii |
| Certificate | iii |
| Acknowledgments | iv |
| Abstract | v |
|  |
| Table of Contents | vi |
|  |
| **1.Introduction** | **7** |
| 1.1.Purpose | 9 |
| 1.2.Scope | 12 |
| 1.3.Some Important Definitions | 16 |
| 1.4.Tools Used | 22 |
| **2.Software Requirement Analysis** | **23** |
| 2.1.Problem | 25 |
| 2.2.Modules And Their Fuctions | 28 |
| **3.Software Design** | **30** |
| 3.1.Zero Level DFD | 31 |
| 3.2. One Level DFD | 32 |
| 3.3. Use Case Diagram | 33 |
| 3.4.Sequence Diagram | 34 |
| **4.Implementation and User Interface** | **42** |
| **5.References** | **44** |
| **6.Appendices** | **46** |

Department of Computer Engineering & Applications V

**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.

Ishtehaar is a portal between Job Seekers and Recruiters. It is helpful for fresher as well as young graduates, to get stated with their career. Using Ishtehaar, 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.

Ishtehaar 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, Ishtehaar focuses on the entire career growth of the candidate.

**(a) Objective:**

* The primary objective of Ishtehaar 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.
* Ishtehaar 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, Ishtehaar 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. Ishtehaar 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

**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 bytecode.
* **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, thebrowser, 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 groupIDs 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.

**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 Post greSQL 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.
* Filter 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 sers.
* 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.

**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

**2.Software Requirement Analysis**

2.1.Problem

Finding a job and getting person required for a work in affordable salary is not everyone’s cup of tea. As you see most of the websites or apps helping people find jobs are for middle level of people that required minimum graduation of Intermediate but there is no place where people like these can find. Also now days everything is getting digital so our old way of seeking for jobs into the newspaper must be changed. There are a lot of jobs for such people but the problem is the information is not conveyed properly as there is a gap between them so, we are trying to give a solution to these problem.

2.2.Modules And Their Functions

**SignUp**

The 'Sign up' activity allows the user to register himself in Ishtehaar. 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.

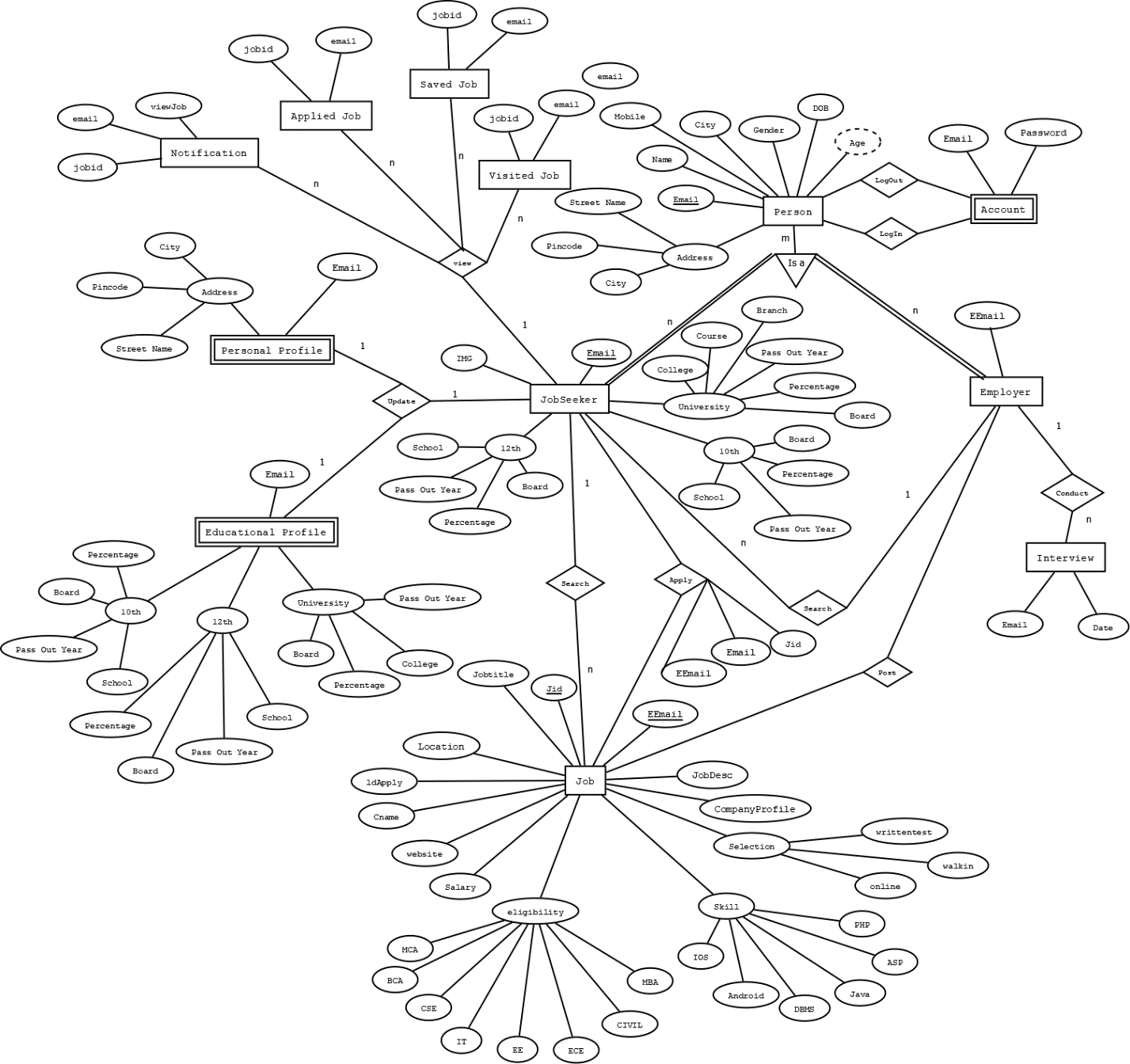
**DataFlow Diagram**



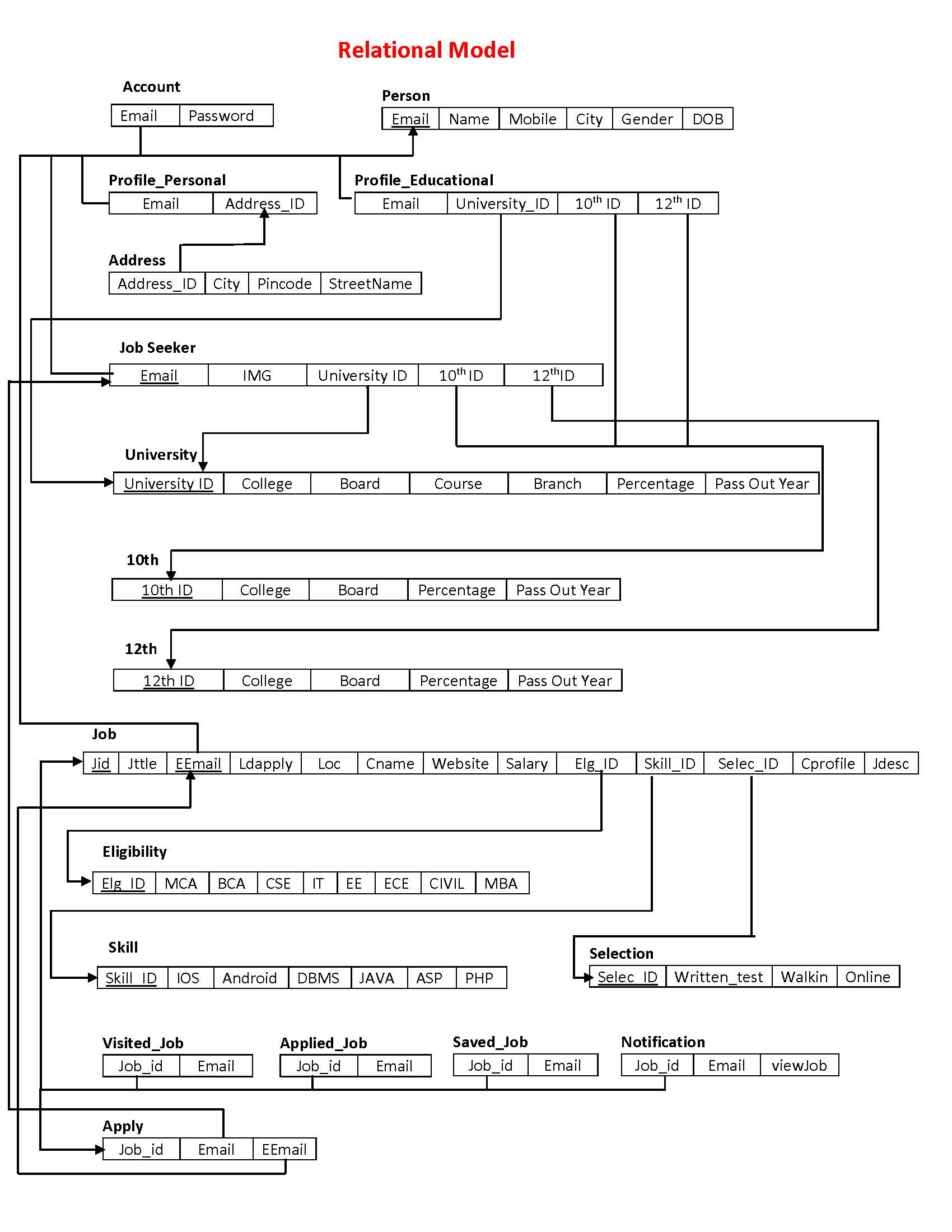




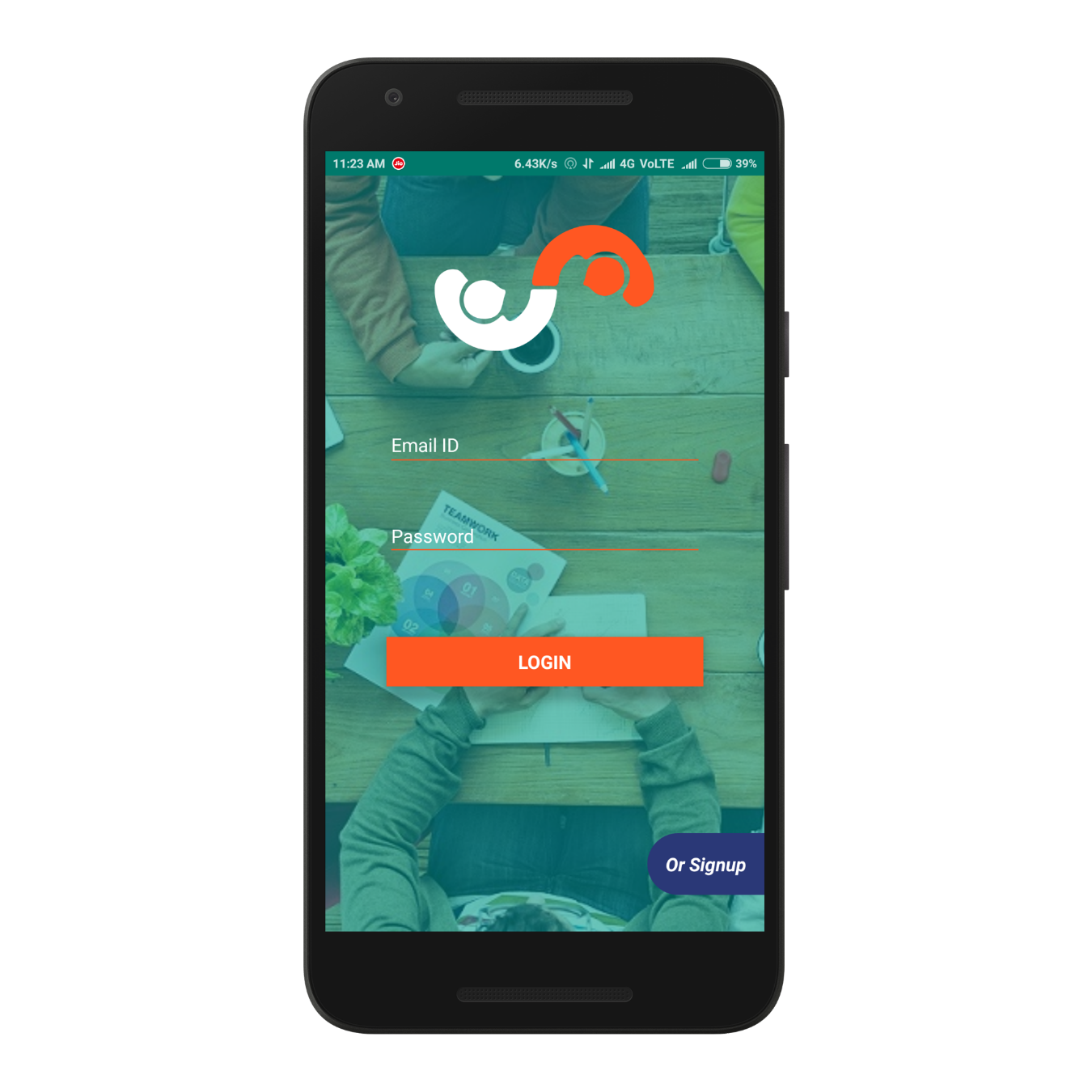
**Entity Relationship Diagram**



**Relational Model**



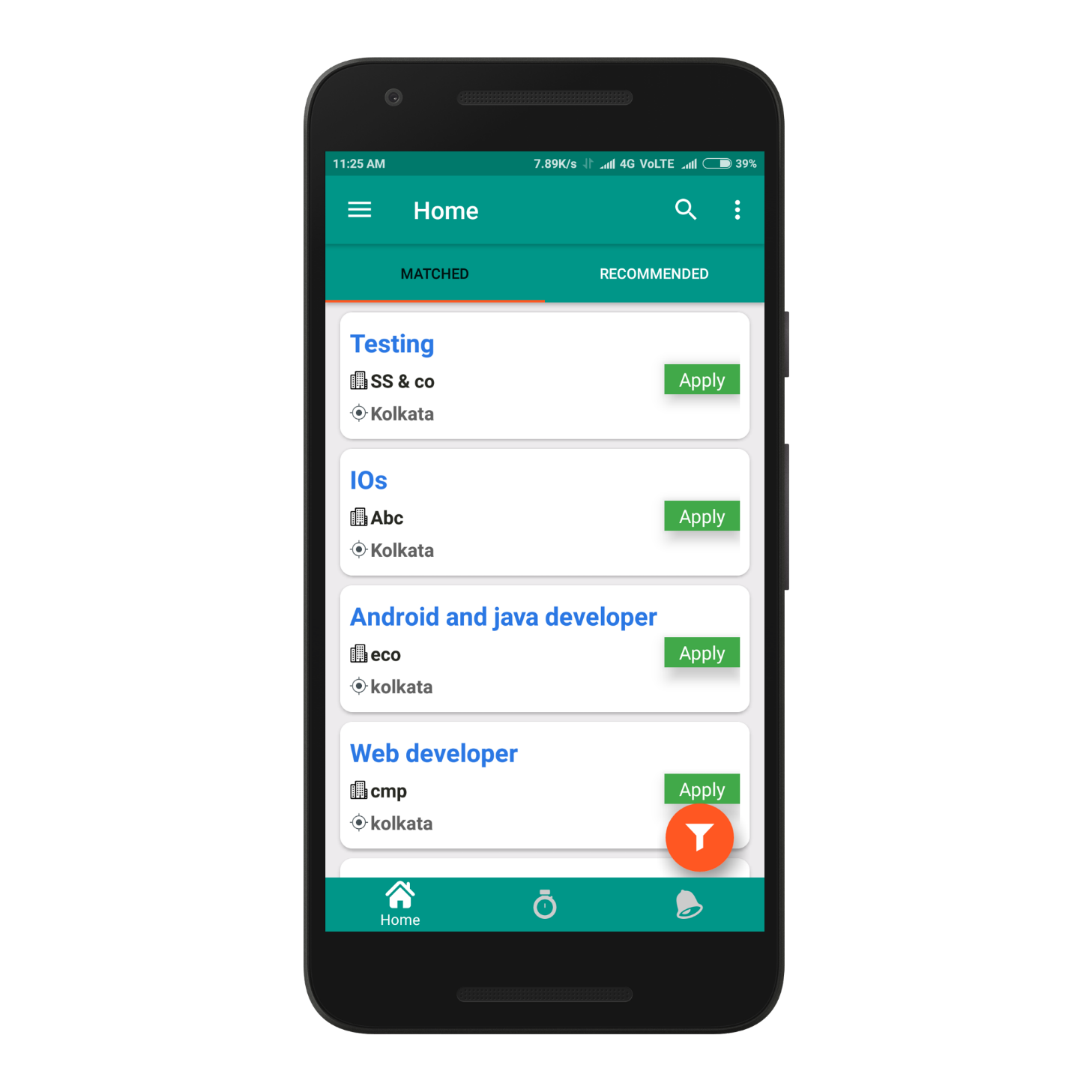
**Screenshots: Login**

****

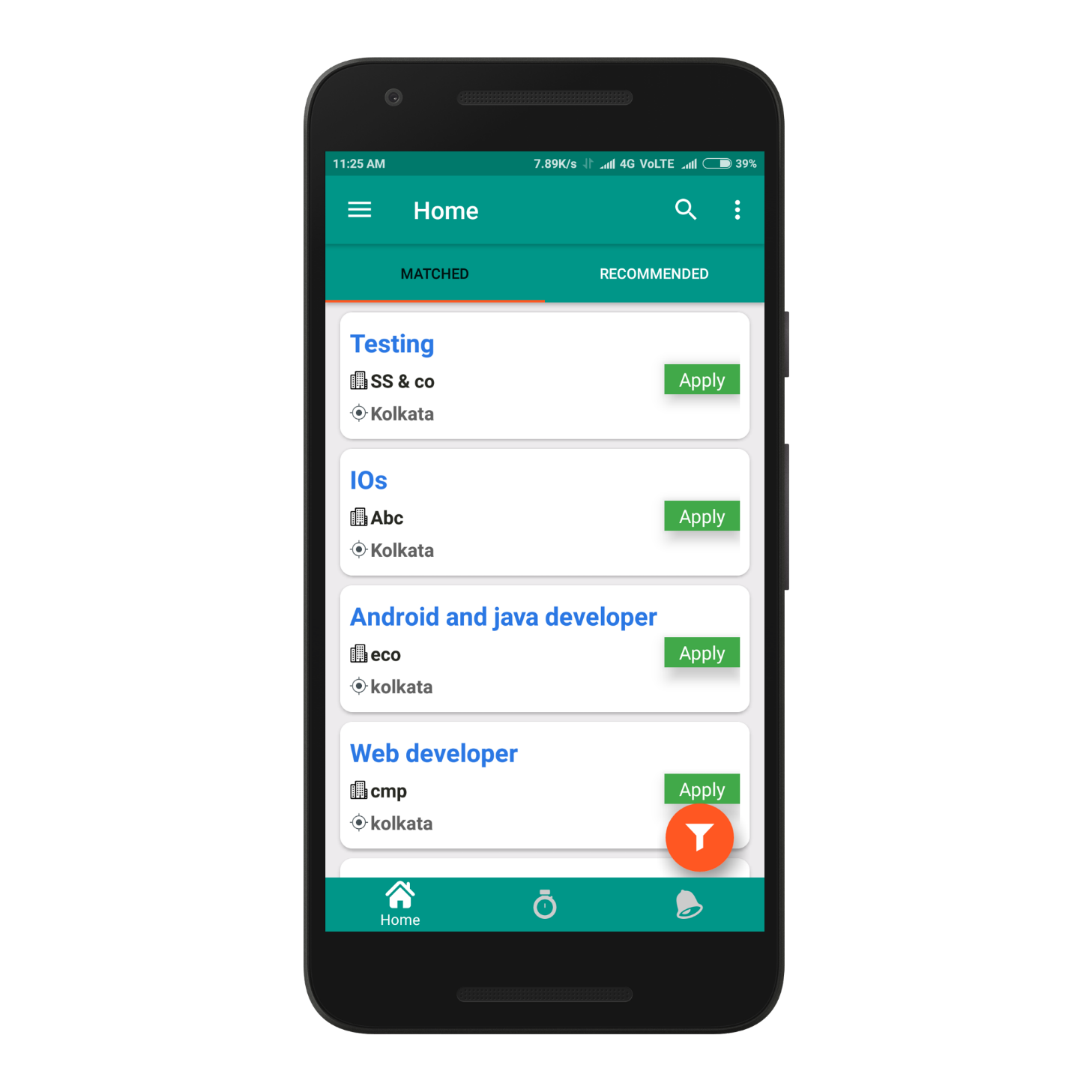
**Sign up:**

****

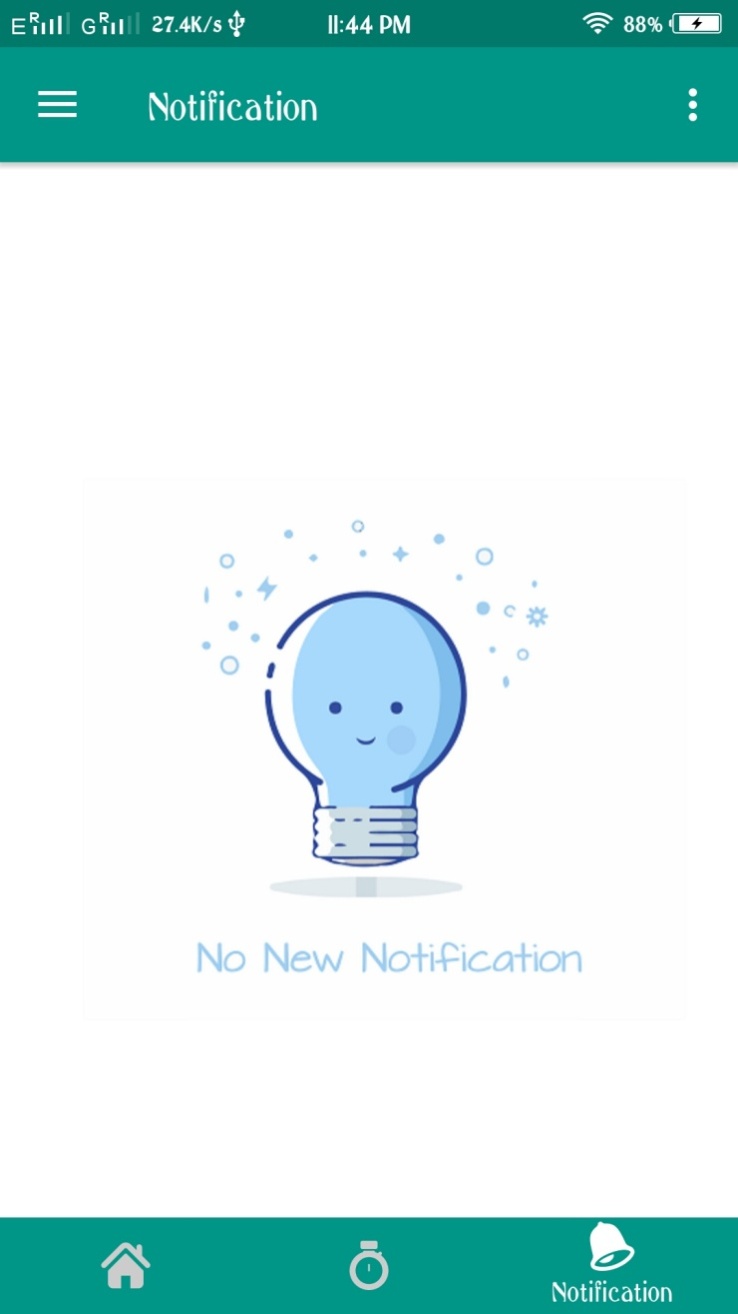
**Home:**



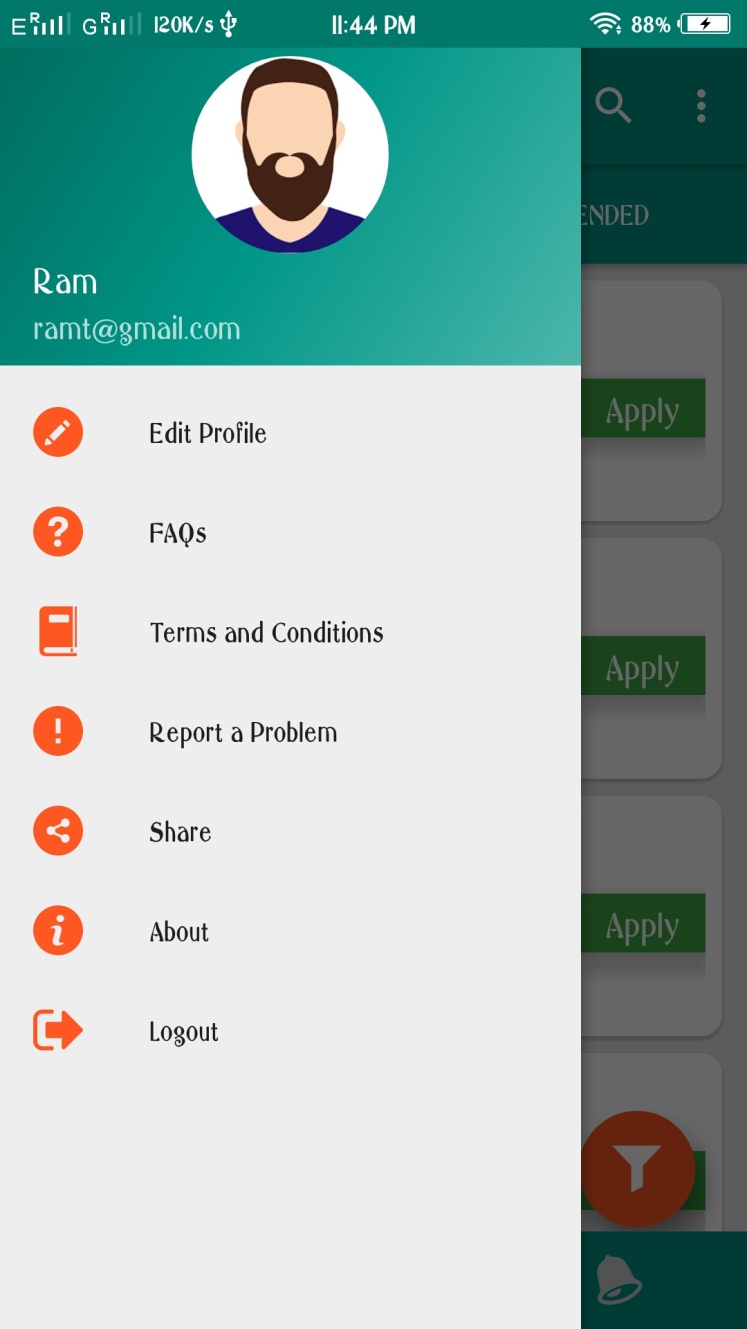
**Activity:**



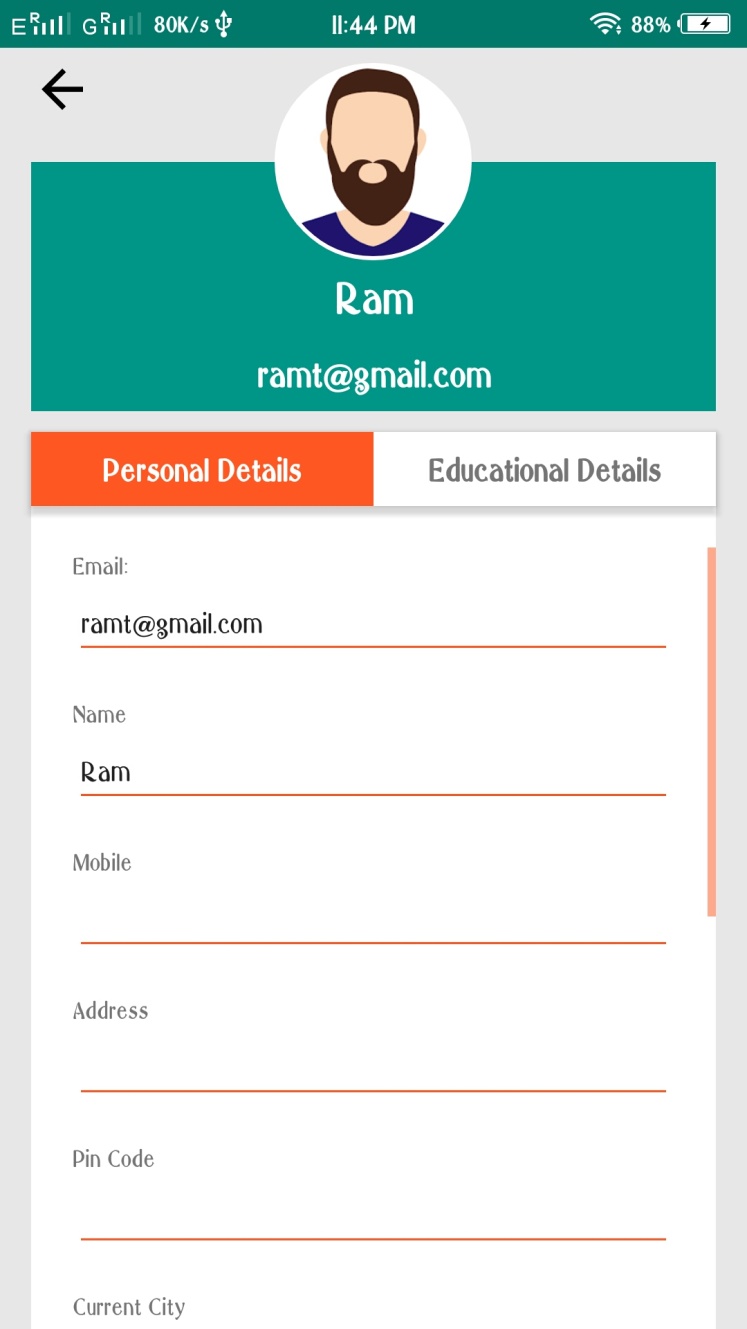
**Notification:**

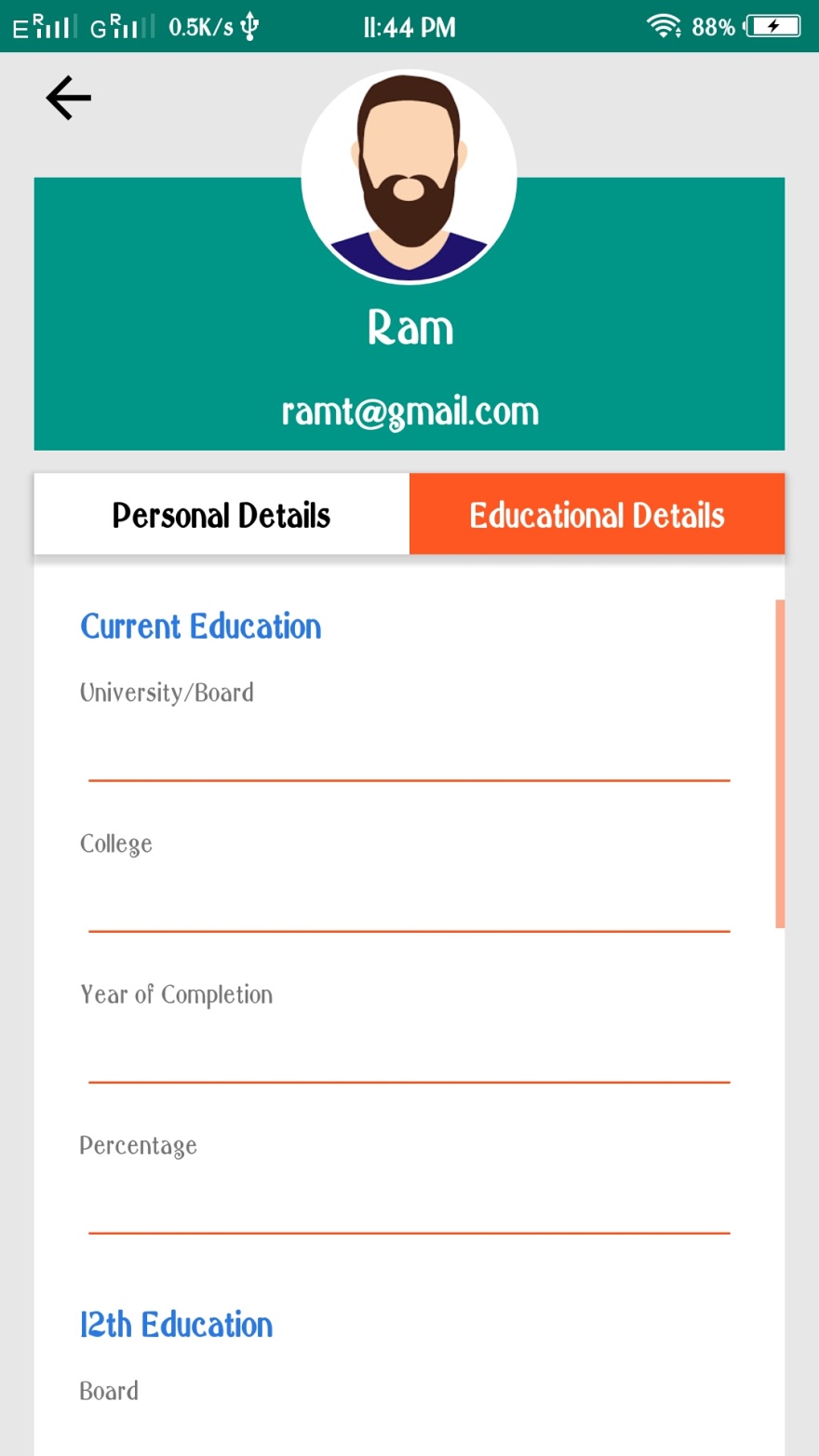


**Menu:**

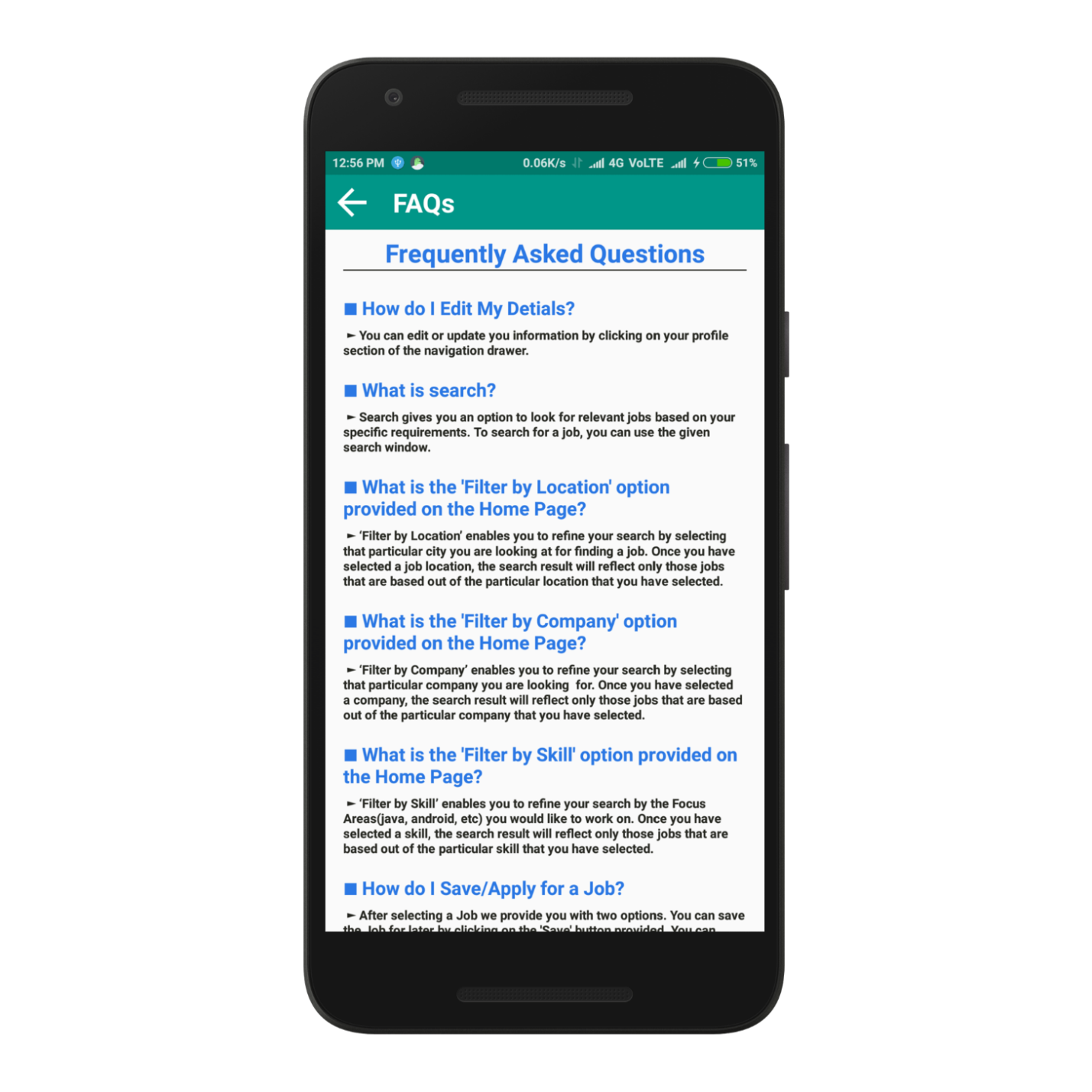


**Profile:**





**FAQ:**



**Coding:**

**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** brdevelopers.com.ishtehaar;  
  
**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\_empname**,**tv\_empemail**;  
**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) {  
  
 ViewAdapter viewAdapter=**new** ViewAdapter(getSupportFragmentManager());  
 viewAdapter.addFragment(**new** MatchedFragment(),**"Matched"**);  
 viewAdapter.addFragment(**new** Recommended(),**"Recommended"**);  
 viewPager.setAdapter(viewAdapter);  
 }  
  
**private void** setterViewPagerActivity(ViewPager viewPager) {  
  
 ViewAdapter viewAdapter=**new** ViewAdapter(getSupportFragmentManager());  
 viewAdapter.addFragment(**new** Viewed\_Fragment(),**"Viewed"**);  
 viewAdapter.addFragment(**new** Saved\_Fragment(),**"Saved"**);  
 viewAdapter.addFragment(**new** Applied\_Fragment(),**"Applied"**);  
 viewPager.setAdapter(viewAdapter);  
 }  
  
**private void** setterViewPagerNotification(ViewPager viewPager) {  
  
 ViewAdapter viewAdapter=**new** ViewAdapter(getSupportFragmentManager());  
 viewAdapter.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\_9l52LilIhlLOE48\_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**);  
 setterViewPager(**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**);  
 setterViewPagerActivity(**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***);  
 setterViewPagerNotification(**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.

5.3.1Unit Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testcase Ref No** | | **TCT-001** | | |
|  | |  |  | |
| Functionality | | : | Login to the System | |
|  | |  |  | |
| Expectedoutcome | | : | Theuser should only login with valid credentials  failing which some error message follows | |
|  | | | | |
| StepNo. | DataUsed | | | ActualOutcome |
| 1. | Click on the log in button  withoutenteringusernameor password | | | Analert messagecametoenter  Username |
| 2. | Click on the log in button  afterenteringsomeusername leaving password field blank | | | Analert message came to enter  Password |
| 3. | Click on the log in button  afterenteringsomepassword but leaving username field blank | | | Analert message came to enter  Username |
| 4. | Click on the log in button  after entering some wrong username but correct password | | | Amessage displayed on Log in  pageabout this |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testcase Ref No** | | **TCT-002** | | |
|  | |  |  | |
| Functionality | | : | Enter valid Data for member signup | |
|  | |  |  | |
| Expectedoutcome | | : | 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. | |
|  | | | | |
| StepNo. | DataUsed | | | ActualOutcome |
| 1. | Click onthesignupbutton  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. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testcase Ref No** | | **TCT-003** | | |
|  | |  |  | |
| Functionality | | : | Applying for a job | |
|  | |  |  | |
| Expectedoutcome | | : | Theuserwill be alerted if the job he/she wants to apply for has already been applied. | |
|  | | | | |
| StepNo. | DataUsed | | | ActualOutcome |
| 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**

**6.1TestReports**

|  |  |  |
| --- | --- | --- |
| TestCase No | Date | Pass/ Fail |
| TCT-001 | 01/04/2019 | Pass |
| TCT-002 | 05/04/2019 | Pass |
| TCT-003 | 05/04/2019 | 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 stated 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