**A PROJECT REPORT ON**

**COLLEGE MANAGEMENT SYSTEM**

**Under the Guidance of**

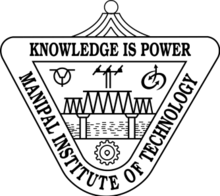
**Mr.Shailesh Kumar Shrivastava**, **(Sr. Technical Director, NIC & Head of DGRC,Patna)**

**Submitted by**

**Pranav Shekhar**

**Reg No 180905296**

**B.tech (2018 – 22)**



**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl*.* No*.*** | **Topic** | **PageNo*.*** |
|  | Declaration | 4 |
|  | Acknowledgement | 5 |
|  | **PROJECT REPORT** |  |
| ***1.*** | ABOUT THE ORGANIZATION | 6 |
| ***3.*** | ABOUT THE PROJECT | 7 |
| ***4.*** | TOOLS and PLATFORM | 9 |
| ***5.*** | SYSYEM REQUIREMENT SPECIFICATIONS   * SOFTWARE REQUIREMENTS * HARDWARE REQUIREMENTS | 10  10  11 |
| ***6.*** | FRONT END (About Android Studio) | 12 |
| ***7.*** | BACK END (About SQLite Database) | 17 |
| ***8.*** | DATABASE | 18 |
| ***9.*** | DATA FLOW DIAGRAM | 19 |
| ***10.*** | ENTITY RELATION DIAGRAM | 24 |
| ***11.*** | DATA STRUCTURE | 26 |
| ***12.*** | CODE DOCUMENTATION   * Splash Screen Activity * Login Screen Activity * Sign Up Screen Activity * Home Screen Activity | 28  28  29  31 |
| ***13.*** | SCREENSHOTS | 39 |
| ***14.*** | FUTURE SCOPE OF THE PROJECT | 43 |
| ***15.*** | CONCLUSION | 44 |
| ***16.*** | BIBLIOGRAPHY | 45 |

**Declaration**

I hereby declare that project title ”**COLLEGE MANAGEMENT SYSTEM**” Submitted by me to **National Informatics Centre Services Inc., Patna** is original in nature this project is outcome of my own work is bonafide, correct to the best of my knowledge and this work has been carried out taking care of Computer Application Ethics.

**Pranav Shekhar**

**Reg No: 180905296**

**Session:2018-2022**

**Acknowledgement**

**First of all I would like to pay humble respect in the lotus feet of the almighty God for his grace and mercy by which I am able to complete this project.**

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I respect and thank **Mr.Shailesh Kumar Shrivastava**, **(Senior Technical Director, NIC & Head of Digital Government Research Centre, Patna)** for providing me an opportunity to do the project work in **National Informatics Centre, Bihar** and giving us all support and guidance which made me complete the project duly. I am extremely thankful to **Mr. Navneet Sir and Mr. Abhinav Sir** for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

I owe my deep gratitude to our project guide, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

**Pranav Shekhar**

**Reg No: 180905296**

**Session:2018-2022**

**About National Informatics Centre(NIC)**

The National Informatics Centre is a part of the Indian Ministry

of Electronics and Information Technology. It has its headquarters in

New Delhi. It has offices in all 29 state capitals and 7 union-territory

headquarters and almost all districts. It is the premier science & technology

organization of the Government of India in Informatics Services

and Information & Communication Technology (ICT) applications.

It plays a pivotal role in steering e-governance applications in the

governmental departments at national, state and district levels. It enables

the improvement of government services and also maintains a

transparency in these services. Almost all Indian-government websites

are developed and managed by NIC. The NIC assists in implementing

information-technology projects, in collaboration with central

and state governments, in the areas of communication & information

technology. It also offers telecommunication networking services

including wireless metropolitan-area networks (MANs) and local-area

networks (LAN's) with gateways for Internet and Intranet resource

sharing.

NIC computer cells are located in almost all the Ministry buildings

of the Central Government and apex offices including the Indian

Prime Minister’s office, the Indian Presidential Palace (Rashtrapati

Bhavan) and India’s Parliament House (Sansad Bhavan). It also

provide support to grass root level administration. NIC provides the

network infrastructure and e-governance support to India’s central

government and state governments, union-territory administrations,

administrative divisions and other government bodies.

**1.INTRODUCTION & OBJECTIVE**

* **Title Of The Project**

**COLLEGE MANAGEMENT SYSTEM**

* **Introduction & Objective of the Project**

A live project in an organization is an essential part of our B.tech curricular. To fulfilling this requirement. I applied at **National Informatics Centre Patna**, which is a government organization for all types of work related to information processing, working for Govt. of India. The organization is based at Patna and they kindly agreed and provided a project of the development of an Android Application for the College Database Management Developed using **SQLite Database** With **Android Studio.**

The main objective of the project is to provide an offline solution for the Data Handling process for The College Management system which simplify the whole process of data pooling of student information in colleges and provides a robust and efficient platform to allocate student data offline.

**OBJECTIVES**

* Provide an innovative and effective management system..
* Offer fast and efficient usability of the data collected through the app.
* Dramatically reduce paper work and administrative work.

**TOOLS & PLATFORM**

* **JAVA**

Java is a widely used programming language expressly designed for use in the distributed environment of the internet. ... Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network.

* **ANDROID STUDIO**

Android Studio is the official integrated development environment(IDE) for Google’s Android Operating System, built on JetBrain’s IntelliJ IDEA software and designed specifically for Android Development. It is available for download on Windows,mac-OS and Linux based OS.

**SYSTEM REQUIREMENT SPECIFICATIONS**

**Software Requirements**

* **JAVA**

Java is a widely used programming language expressly designed for use in the distributed environment of the internet. Java can be used to create complete applications that may run on a single computer or be distributed among servers and clients in a network.

* **ANDROID STUDIO**

Android Studio is the official integrated development environment(IDE) for Google’s Android Operating System, built on JetBrain’s IntelliJ IDEA software and designed specifically for Android Development. It is available for download on Windows,mac-OS and Linux based OS.

* **SQLite Database**

SQLite is an in-process library that implements a self-contained, zero-configuration, serverless, transactional SQL database engine. The source code for SQLite exists in the public domain and is free for both private and commercial purposes.

**Hardware Specification**

Processor : Intel core i5 8th gen

RAM : 8 GB

Graphics : 2 GB

Hard disk : 256 GB

Monitor : 13 inch

Mouse : Touch-pad

Keyboard : 108 keys

***FRONTEND***

**ANDROID STUDIO**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

* A flexible Gradle-based build system
* A fast and feature-rich emulator
* A unified environment where you can develop for all Android devices
* Instant Run to push changes to your running app without building a new APK
* Code templates and GitHub integration to help you build common app features and import sample code
* Extensive testing tools and frameworks
* Lint tools to catch performance, usability, version compatibility, and other problems
* C++ and NDK support
* Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

This page provides an introduction to basic Android Studio features. For a summary of the latest changes, see Android Studio release notes.

1. **PROJECT STRUCTURE**

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

* Android app modules
* Library modules
* Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

* manifests: Contains the AndroidManifest.xml file.
* java: Contains the Java source code files, including JUnit test code.
* res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project from the Project dropdown (in figure 1, it's showing as Android).

You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

1. **USER INTERFACE**

* The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
* The **navigation ba**r helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
* The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
* The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
* The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
* The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

You can organize the main window to give yourself more screen space by hiding or moving toolbar and tool windows. You can also use keyboard shortcuts to access most IDE features.

At any time, you can search across your source code, databases, actions, elements of the user interface, and so on, by double-pressing the Shift key, or clicking the magnifying glass in the upper right-hand corner of the Android Studio window. This can be very useful if, for example, you are trying to locate a particular IDE action that you have forgotten how to trigger.

1. **GRADLE BUILD SYSTEM**

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

* Customize, configure, and extend the build process.
* Create multiple APKs for your app, with different features using the same project and modules.
* Reuse code and resources across source-sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build.gradle. They are plain text files that use Groovy syntax to

configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

1. **MULTIPLE APK SUPPORT**

Multiple APK support allows you to efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and allowing them to share test APK, javac, dx, and ProGuard settings.

For more information about multiple APK support, read Build multiple APKs.

1. **INLINE DEBUGGING**

Use inline debugging to enhance your code walk-throughs in the debugger view with inline verification of references, expressions, and variable values. Inline debug information includes:

* Inline variable values
* Referring objects that reference a selected object
* Method return values
* Lambda and operator expressions
* Tool-tip values

1. **DATA FILE ACCESS**

The Android SDK tools, such as Systrace, and logcat, generate performance and debugging data for detailed app analysis.

To view the available generated data files, open the Captures tool window. In the list of the generated files, double-click a file to view the data. Right-click any .hprof files to convert them to the standard Investigate your RAM usage file format.

1. **CODE INSPECTION**

Whenever you compile your program, Android Studio automatically runs configured Lint and other IDE inspections to help you easily identify and correct problems with the structural quality of your code.

The Lint tool checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.

In addition to Lint checks, Android Studio also performs IntelliJ code inspections and validates annotations to streamline your coding workflow.

For more information, see Improve your code with lint checks.

***BACKEND***

* **SQLite Database**

**SQLite** is a programming library which implements a relational database management system. The SQLite database concept is, in contrast to other client-server systems, to be linked into the applications code, instead of providing a standalone daemon with which an application can communicate to request or write data. Because of the small size of the library itself, and the ease of use, it is esepcially interesting for [embedded systems](http://en.wikipedia.org/wiki/embedded_systems).

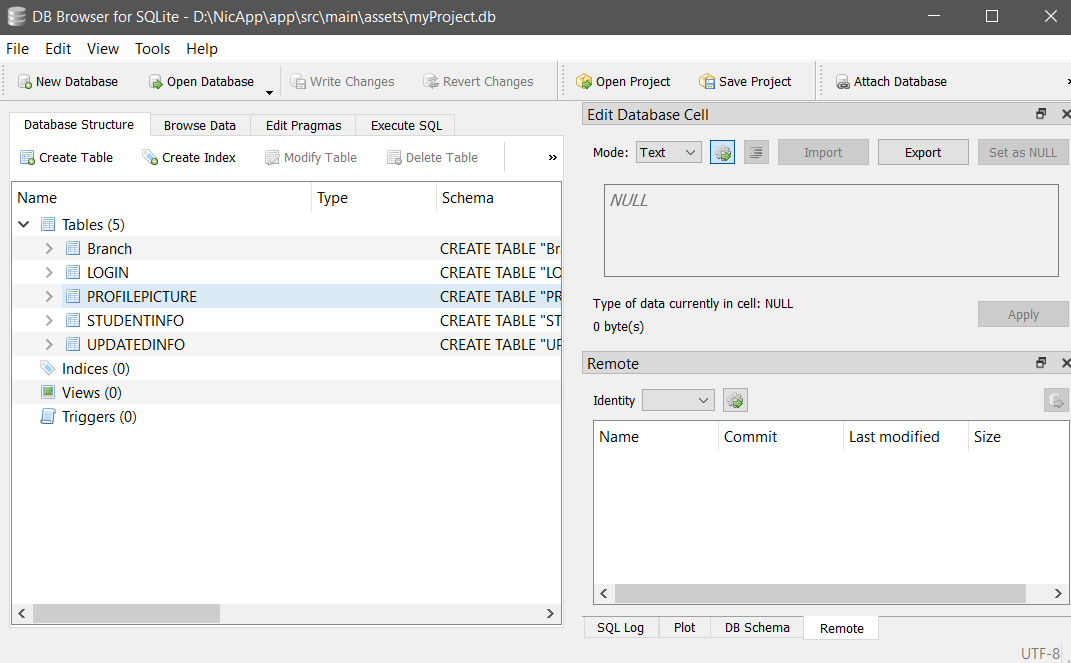
SQLite supports a variety of SQL-  (**S**tructured **Q**uery **L**anguage) commands with some exceptions and does not provide any access or user-management. That means, that everyone, who can access the database file, can access the data as well as write (change, delete, add) data, if he can write the database file. It therefore inherits the access permissions of the filesystem.

SQLite implements the SQL-92 standard for SQL and uses an unusual system for SQL compatible database management systems. Types are assigned to individual values, adding flexibility to columns when bound to dynamic scripting languages. Full unicode support in SQLIte is optional.

***DATABASE***

**My SQLite Database Interface**

We created a database named myProject for our project that contains various tables including Login, StudentInfo, etc

****

**DATA FLOW DIAGRAM**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](http://en.wikipedia.org/wiki/Information_system), modeling its *process* aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the [visualization](http://en.wikipedia.org/wiki/Data_visualization) of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).the basic notation used to create a DFD’s are as follows:

**1. Dataflow**: data moves in a specific from an origin to a destination.

C:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BAA.tmp.png

C:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BAB.tmp.pngC:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BBB.tmp.pngC:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BBC.tmp.pngC:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BBD.tmp.pngC:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BBE.tmp.png

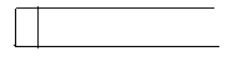
|  |
| --- |
|  |
|  | C:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BBF.tmp.png |

**2. Process: P**eople ,procedures or device that use or produce data. The physical components not identified.

**3**. **Source**: external sorce or destination of data, which may be people programs, organizations or other entities.



**4 .Data source:** here data are store and referenced by a process in the system.

C:\Users\Pranav\AppData\Local\Temp\ksohtml\wps2BD1.tmp.png 

The systematic representation of information system in a typical environment can be produced by a Data Flow Diagram which shows the system \ module or \ and sub – systems .The graphical description of a systems data and how the processes transform data is known as the Data Flow Diagram or the DFD

* **COLLEGE MANAGEMENT SYSTEM**
* **SPLASH SCREEN**
* **LOGIN**
* **REGISTRATION**
* **VIEW PROFILE**
* **UPDATE DETAILS**
* **EDIT DETAILS**
* **LOG OUT**

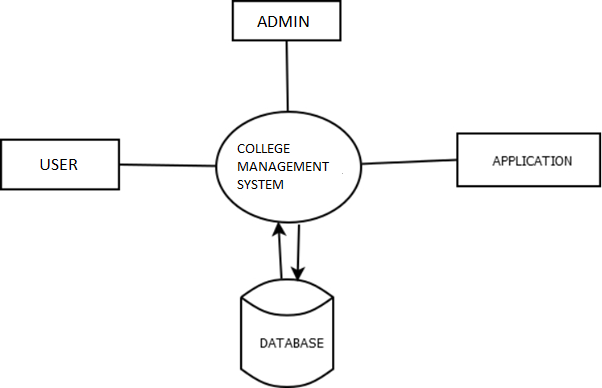
The client as admin can manipulate all the contents, , etc. Without going into any detail or complex structure of the database, but can do it all on the run i.e, in the live running website through the GUI available.

The modular structure is given below ,

* **COLLEGE MANAGEMENT SYSTEM (ADMIN)**
* **LOGIN**
* **VIEW USER DETAILS**
* **EDIT USER DETAILS**
* **DELETE USER DETAILS**
* **LOG OUT**

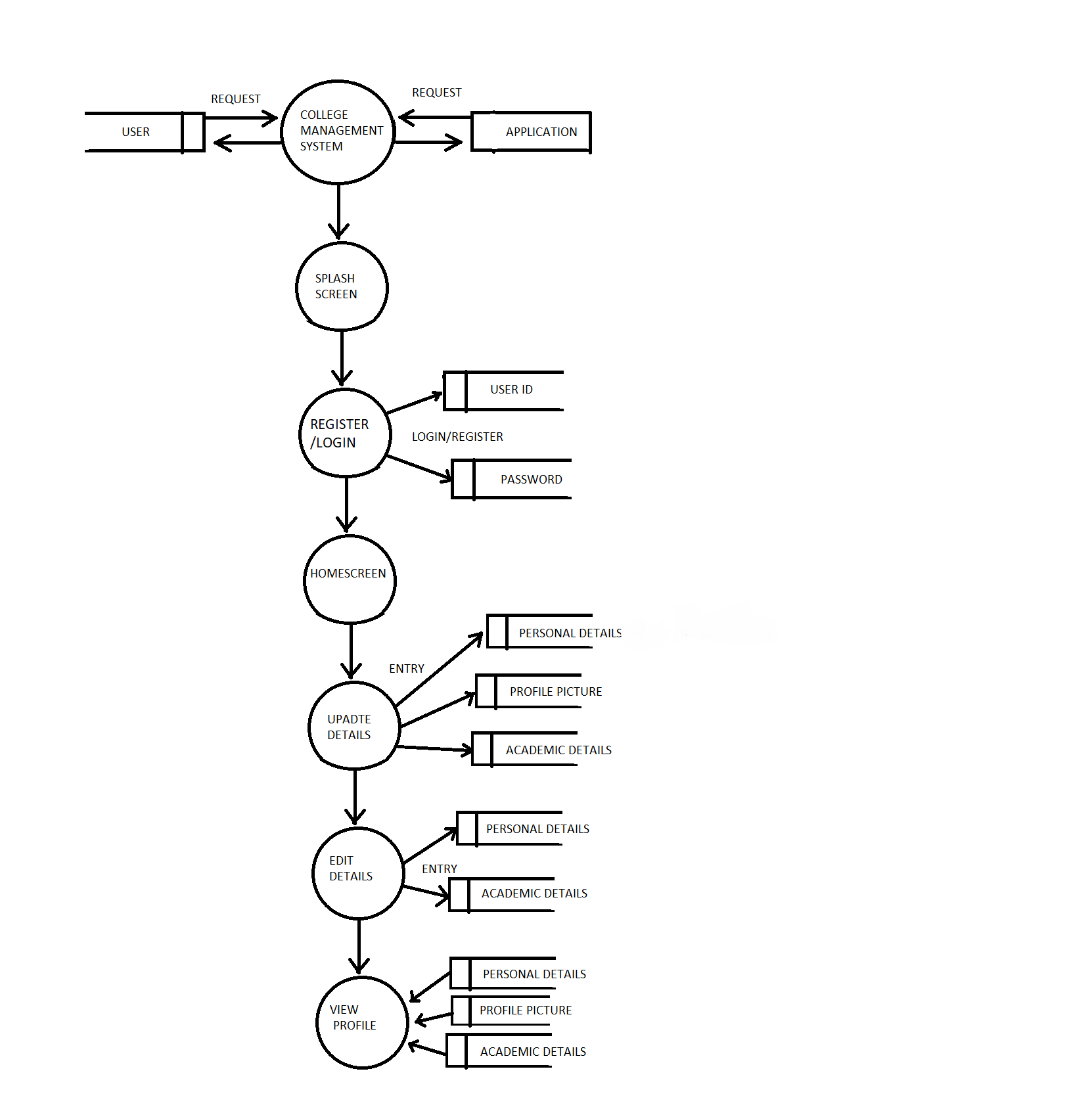
**PHYSICAL DFD for ONLINE RECRUITMENT SYSTEM**

**0 LEVEL**



For Viewer / 3rd party user

**PHYSICAL DFD for ONLINE RECRUITMENT SYSTEM**



**ENTITY RELATIONSHIP DIAGRAM**

Entity Relationship model is a high level conceptual Data model. It Allows us to describe the data involved in real world ENTITIES in terms of their object relationship. It is widely used to develop an initial design of a database. It describes data as a collection of entities, relationships and attributes. The ERD provide us a graphical description of relationships between entities & attributes.

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

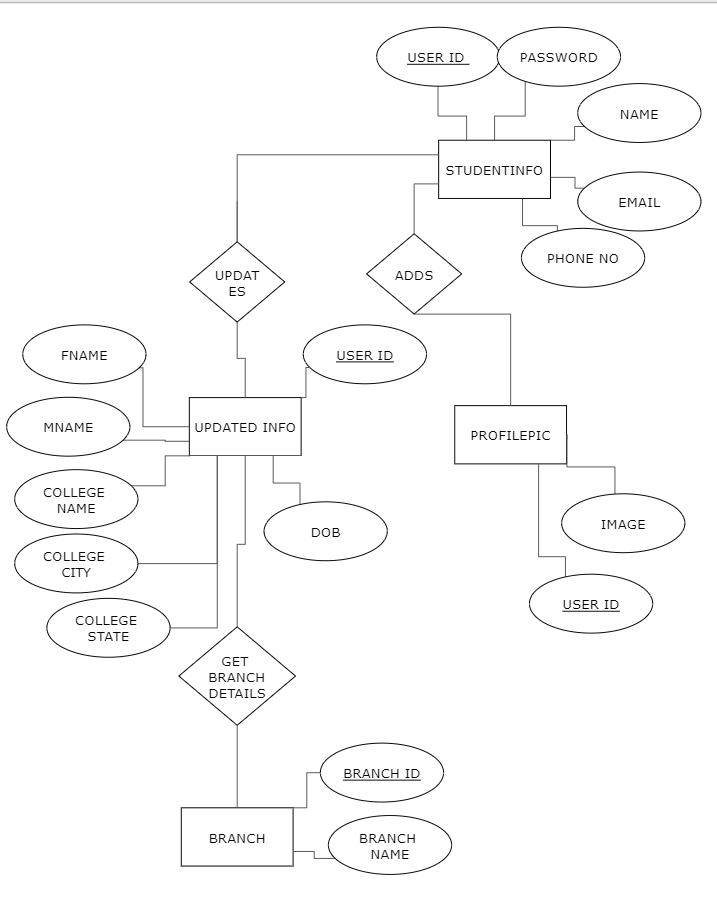
* Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information.



A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



* Actions, which are represented by diamond shapes, show how two entities share information in the database.   
  C:\Users\Pranav\AppData\Local\Temp\ksohtml\wpsE375.tmp.png 
* Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.

****

***DATA STRUCTURE***

**DATA STRUCTURES OF TABLES**

Here are the tables available in my database.

1. **myProject.db**

* **Table Name: LOGIN**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Key** |
| User\_ID | TEXT | Primary key |
| Password | TEXT | **-** |

* **Table Name: STUDENTINFO**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Key** |
| User\_ID | TEXT | - |
| Password | TEXT | **-** |
| Name | TEXT | **-** |
| Email\_ID | TEXT | **-** |
| PhoneNo | TEXT | **-** |

* **Table Name: UPDATEDINFO**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Key** |
| User\_ID | TEXT | - |
| Fname | TEXT | **-** |
| Mname | TEXT | **-** |
| ColName | TEXT | **-** |
| ColCity | TEXT | **-** |
| ColState | TEXT | **-** |
| Branch | TEXT | **-** |
| DOB | TEXT | **-** |

* **Table Name: PROFILE PIC**

|  |  |  |
| --- | --- | --- |
| **Field** | **Data type** | **Key** |
| User\_ID | TEXT | - |
| PROFIEPIC | BLOB | **-** |

**CODE DOCUMENTATION**

**SplashScreen.java**

**package** nicapp.nic.bihar.nicapp;  
  
**import** android.content.Intent;  
**import** android.os.StrictMode;  
**import** android.preference.PreferenceManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.view.Window;  
**import** android.view.WindowManager;  
**import** android.view.animation.Animation;  
**import** android.view.animation.AnimationUtils;  
**import** android.widget.ImageView;  
**import** android.widget.TextView;  
  
**public class** SplashScreen **extends** AppCompatActivity {  
 TextView **start1**;  
 Intent **i**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 */\*requestWindowFeature(Window.FEATURE\_NO\_TITLE);  
 getSupportActionBar().hide();  
\*/* **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_splash\_screen***);  
  
  
 **this**.getWindow().setFlags(WindowManager.LayoutParams.***FLAG\_FULLSCREEN***,  
 WindowManager.LayoutParams.***FLAG\_FULLSCREEN***);  
 **if** (android.os.Build.VERSION.***SDK\_INT*** > 9)  
 {  
 StrictMode.ThreadPolicy policy = **new** StrictMode.ThreadPolicy.Builder().permitAll().build();  
 StrictMode.*setThreadPolicy*(policy);  
 }  
 **start1** = findViewById(R.id.***main***);  
 Animation animob = AnimationUtils.*loadAnimation*(**this**, R.anim.***alpha***);  
 **start1**.startAnimation(animob);  
 String uname = PreferenceManager.*getDefaultSharedPreferences*(SplashScreen.**this**).getString(**"User\_ID"**, **""**);  
 String pass = PreferenceManager.*getDefaultSharedPreferences*(SplashScreen.**this**).getString(**"Password"**, **""**);  
  
 **if**(uname!=**""** && pass!=**""**){  
 **i** = **new** Intent(SplashScreen.**this**,HomeScreen.**class**);  
 }  
 **else**{  
 **i** = **new** Intent(getApplicationContext(),MainActivity.**class**);  
 }  
  
 Thread t1 = **new** Thread() {  
 **public void** run() {  
  
 **try** {  
 *sleep*(4000);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 } **finally** {  
 startActivity(**i**);  
 finish();  
 }  
 }  
 };  
 t1.start();  
  
  
 }  
}

**SplashScreen.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".SplashScreen"**>  
 <**ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/splash4"  
 android:id="@+id/pic1"**/>  
  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:textAlignment="center"  
 android:layout\_height="wrap\_content"  
 android:text="Welcome to"  
 android:textSize="50sp"  
 android:layout\_marginTop="40sp"  
 android:id="@+id/main2"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:textAlignment="center"  
 android:textSize="30sp"  
 android:layout\_height="wrap\_content"  
 android:textStyle="bold"  
 android:text="College Management App"  
 android:id="@+id/main3"  
 android:layout\_below="@id/main2"**/>  
 <**ProgressBar  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="550dp"  
 android:id="@+id/pbar"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="20sp"  
 android:textAlignment="center"  
 android:text="Hello Student!"  
 android:id="@+id/main"  
 android:layout\_below="@+id/pbar"**/>  
  
</**RelativeLayout**>

**MainActivity.java (Login page)**

**package** nicapp.nic.bihar.nicapp;  
  
**import** android.annotation.SuppressLint;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.preference.PreferenceManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.view.Window;  
**import** android.view.WindowManager;  
**import** android.widget.AutoCompleteTextView;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.io.IOException;  
  
**import** nicapp.nic.bihar.nicapp.Model.RegistrationModel;  
**import** nicapp.nic.bihar.nicapp.database.DatabaseHelper;  
**import** nicapp.nic.bihar.nicapp.database.OnlineDatabase;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 EditText **name\_L**;  
 EditText **password\_L**;  
 DatabaseHelper **mDatabase**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 */\*requestWindowFeature(Window.FEATURE\_NO\_TITLE);\*/* Context context;  
 **mDatabase** = **new** DatabaseHelper(MainActivity.**this**);  
context = **this**;  
 **try** {  
 **mDatabase**.createDataBase();  
 } **catch** (IOException ioe) {  
  
 **throw new** Error(**"Unable to create database"**);  
 }  
  
 **mDatabase**.openDataBase();  
  
 */\*getSupportActionBar().hide();\*/* TextView signup;  
 Button b2;  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 signup = findViewById(R.id.***tv1***);  
 signup.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent hello = **new** Intent(MainActivity.**this**, SignupPage.**class**);  
 startActivity(hello);  
 }  
 });  
 b2 = findViewById(R.id.***button1***);  
 b2.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **name\_L** = findViewById(R.id.***useridlogin***);  
 **password\_L** = findViewById(R.id.***passwordlogin***);  
 **boolean** validate=**false**;  
 validate=validate1();  
 **if**(validate==**true**) {  
 String username = **name\_L**.getText().toString();  
 String password = **password\_L**.getText().toString();  
 String method =**"login"**;  
 OnlineDatabase ob1 = **new** OnlineDatabase(MainActivity.**this**);  
 ob1.execute(method,username,password);  
 RegistrationModel pm = **mDatabase**.profilelogin(username, password);  
 **if**(pm==**null**) {  
 Toast.*makeText*(MainActivity.**this**, **"Wrong Credentials"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 **else**{  
PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext()).edit().putString(**"User\_ID"**,username).commit();  
 PreferenceManager.*getDefaultSharedPreferences*(getApplicationContext()).edit().putString(**"Password"**,password).commit();  
  
 Intent i = **new** Intent(MainActivity.**this**, HomeScreen.**class**);  
 startActivity(i);  
 finish();  
  
  
 }  
 }  
 }  
 });  
 }  
 **private boolean** validate1() {  
 **boolean** validname = **true**;  
 String uname = **name\_L**.getText().toString();  
 String pass = **password\_L**.getText().toString();  
 **if**(uname.isEmpty()){  
 **name\_L**.setError(**"Please fill your User ID "**);  
 validname=**false**;  
 }**else**{  
 **name\_L**.setError(**null**);  
  
 }  
 **if**(pass.isEmpty()){  
 **password\_L**.setError(**"Please Enter Your Password"**);  
 validname=**false**;  
 }**else**{  
 **password\_L**.setError(**null**);  
  
 }  
 **return** validname;  
 }  
}

**MainActivity.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/road"  
 android:orientation="vertical"  
 tools:context=".MainActivity"  
 android:padding="9dp"**>  
  
 <**ImageView android:alpha="0.7"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:src="@drawable/back"  
 android:id="@+id/back"  
 android:padding="60sp"**/>  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_centerInParent="true"  
 android:gravity="center"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="User ID:"  
 android:textStyle="bold"**/>  
  
 <**EditText  
 android:inputType="number"  
 android:layout\_width="150sp"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/useridlogin"** />  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Password:"  
 android:textStyle="bold"**/>  
  
 <**EditText  
 android:layout\_width="150sp"  
 android:inputType="textPassword"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/passwordlogin"**/>  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Log IN"  
 android:textStyle="bold"  
 android:id="@+id/button1"**/>  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/tv1"  
 android:layout\_marginTop="10sp"  
 android:textStyle="bold"  
 android:text="New User? Click here"**/>  
 </**LinearLayout**>  
  
 </**RelativeLayout**>

**Sign-up.java**

**package** nicapp.nic.bihar.nicapp;  
  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
**import** java.io.File;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.OutputStream;  
  
**import** nicapp.nic.bihar.nicapp.Model.RegistrationModel;  
**import** nicapp.nic.bihar.nicapp.database.DatabaseHelper;  
**import** nicapp.nic.bihar.nicapp.database.OnlineDatabase;  
  
  
  
**public class** SignupPage **extends** AppCompatActivity {  
 EditText **name**, **email**, **password**, **user**, **phone**, **cPassword**;  
 String **\_name**, **\_email**, **\_password**, **\_user**, **\_phone**, **\_cPassword**;  
 DatabaseHelper **mDatabase**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 Button b1;  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_signup\_page***);  
 **mDatabase** = **new** DatabaseHelper(**this**);  
 b1 = findViewById(R.id.***buttonSubmit***);  
 **name** = findViewById(R.id.nameS);  
 user = findViewById(R.id.idS);  
 email = findViewById(R.id.emailS);  
 password = findViewById(R.id.passwordS);  
 phone = findViewById(R.id.phoneS);  
 cPassword= findViewById(R.id.password\_confS);  
 DatabaseHelper localDBHelper;  
 Context context;  
 localDBHelper = **new** DatabaseHelper(SignupPage.**this**);  
  
*// prefs = PreferenceManager.getDefaultSharedPreferences(this);* context = **this**;  
 **try** {  
 localDBHelper.createDataBase();  
 } **catch** (IOException ioe) {  
  
 **throw new** Error(**"Unable to create database"**);  
 }  
  
 localDBHelper.openDataBase();  
  
 b1.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **boolean** valid = **false**;  
 valid = Validate();  
 **if** (valid == **true**) {  
 setValue();  
 *//uploadOnline();* String method =**"register"**;  
 OnlineDatabase ob1 = **new** OnlineDatabase(SignupPage.**this**);  
 ob1.execute(method,\_user,\_password,\_name,\_email,\_phone);  
 RegistrationModel regmodel = **new** RegistrationModel();  
 regmodel.setName(\_name);  
 regmodel.setEmail(**\_email**);  
 regmodel.setPassword(**\_password**);  
 regmodel.setPhone\_no(**\_phone**);  
 regmodel.setUserID(**\_user**);  
 **long** result = **mDatabase**.InsertNewEntry(SignupPage.**this**, regmodel);  
 **if** (result > 0) {  
 Toast.*makeText*(SignupPage.**this**, **"Data Inserted"**, Toast.***LENGTH\_SHORT***).show();  
  
 Thread t1 = **new** Thread() {  
 **public void** run() {  
  
 **try** {  
 *sleep*(500);  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 } **finally** {  
 Intent m1 = **new** Intent(SignupPage.**this**, MainActivity.**class**);  
 startActivity(m1);  
 finish();  
 }  
 }  
 };  
 t1.start();  
  
  
 } **else** Toast.*makeText*(SignupPage.**this**, **"Data Not Inserted"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });}  
  
 **private void** setValue() {  
 \_name = name.getText().toString().trim();  
 \_email = email.getText().toString().trim();  
 \_password = password.getText().toString().trim();  
 \_user = user.getText().toString().trim();  
 \_phone = phone.getText().toString().trim();  
  
 }  
  
 **public boolean** Validate() {  
 \_name = name.getText().toString().trim();  
 \_email = email.getText().toString().trim();  
 \_password = password.getText().toString().trim();  
 \_user = user.getText().toString().trim();  
 \_phone = phone.getText().toString().trim();  
 \_cPassword = cPassword.getText().toString().trim();  
 String emailPattern = **"[a-zA-Z0-9.\_-]+@[a-z]+.+[a-z]+"**;  
 **boolean** validName = **true**;  
  
 **if** (\_name.isEmpty()) {  
 name.setError(**"Please Fill you name"**);  
 name.requestFocus();  
 validName = **false**;  
 } **else** {  
 name.setError(**null**);  
  
 }  
 **if** (\_email.length()>0) {  
 **if** (!\_email.matches(emailPattern)) {  
 email.setError(**"Please fill an appropiate email"**);  
 email.requestFocus();  
 validName = **false**;  
  
 }  
 }  
 **else**{  
 email.setError(**null**);  
 }  
 **if** (\_password.isEmpty()) {  
 password.setError(**"Please fill an appropiate password"**);  
 password.requestFocus();  
 validName = **false**;  
 } **else** {  
 password.setError(**null**);  
  
 }  
  
 String str\_pass=password.getText().toString().trim();  
 String str\_conf\_pass=cPassword.getText().toString().trim();  
  
 **if**(!str\_pass.equals(str\_conf\_pass)){  
 cPassword.setError(**"Password does not match"**);  
 validName = **false**;  
  
 }  
 **else**{  
 cPassword.setError(**null**);  
 }  
 **if** (\_phone.isEmpty() || \_phone.length()!=10) {  
 phone.setError(**"Please Fill your Phone no"**);  
 phone.requestFocus();  
 validName = **false**;  
 } **else** {  
 phone.setError(**null**);  
  
 }  
 **if** (\_user.isEmpty()) {  
 user.setError(**"Please Fill an appropiate user Id"**);  
 user.requestFocus();  
 validName = **false**;  
 } **else** {  
 user.setError(**null**);  
  
 }  
 **return** validName;  
 }  
  
  
 **private boolean** copyDatabase(Context context) {  
 **try** {  
  
 InputStream inputStream = context.getAssets().open(DatabaseHelper.DB\_NAME);  
 String outFileName = DatabaseHelper.DB\_PATH + DatabaseHelper.DB\_NAME;  
 OutputStream outputStream = **new** FileOutputStream(outFileName);  
 **byte**[] buff = **new byte**[1024];  
 **int** length = 0;  
 **while** ((length = inputStream.read(buff)) > 0) {  
 outputStream.write(buff, 0, length);  
 }  
 outputStream.flush();  
 outputStream.close();  
 Log.w(**"signup"**, **"DB copied"**);  
 **return true**;  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 **return false**;  
 }  
 }  
 **public void** uploadOnline(){  
 String method =**"register"**;  
 OnlineDatabase ob1 = **new** OnlineDatabase(**this**);  
 ob1.execute(method,\_user,\_password,\_name,\_email,\_phone);  
  
 }  
}

**Sign-up.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/road"  
 android:orientation="vertical"  
 tools:context=".SignupPage"**>  
 <**ImageView android:alpha="0.7"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:src="@drawable/signup"  
 android:id="@+id/back"  
 android:padding="60sp"**/>  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_centerInParent="true"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Name:"  
 android:layout\_gravity="center"  
 android:textStyle="bold"**/>  
  
 <**EditText  
 android:layout\_width="150sp"  
 android:id="@+id/nameS"  
 android:layout\_height="wrap\_content"** />  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Email ID:"  
 android:textStyle="bold"**/>  
  
 <**EditText  
 android:layout\_width="150sp"  
 android:inputType="textEmailAddress"  
 android:id="@+id/emailS"  
 android:layout\_height="wrap\_content"** />  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Password:"  
 android:textStyle="bold"**/>  
  
 <**EditText  
 android:id="@+id/passwordS"  
 android:inputType="textPassword"  
 android:layout\_width="150sp"  
 android:layout\_height="wrap\_content"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Confirm Password:"  
 android:textStyle="bold"**/>  
 <**EditText  
 android:id="@+id/password\_confS"  
 android:inputType="textPassword"  
 android:layout\_width="150sp"  
 android:layout\_height="wrap\_content"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Phone number:"  
 android:layout\_gravity="center"  
 android:textStyle="bold"**/>  
  
 <**EditText  
 android:id="@+id/phoneS"  
 android:layout\_width="150sp"  
 android:inputType="phone"  
 android:layout\_height="wrap\_content"** />  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="User ID:"  
 android:textStyle="bold"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text=" (This will be used to login)"  
 android:layout\_gravity="center"  
 android:textSize="10sp"**/>  
  
 <**EditText  
 android:layout\_width="150sp"  
 android:id="@+id/idS"  
 android:layout\_height="wrap\_content"  
 android:inputType="number"**/>  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Sign Up"  
 android:onClick="btn\_submit"  
 android:textStyle="bold"  
 android:textAlignment="center"  
 android:layout\_gravity="center"  
 android:id="@+id/buttonSubmit"**/>  
 </**LinearLayout**>  
</**RelativeLayout**>

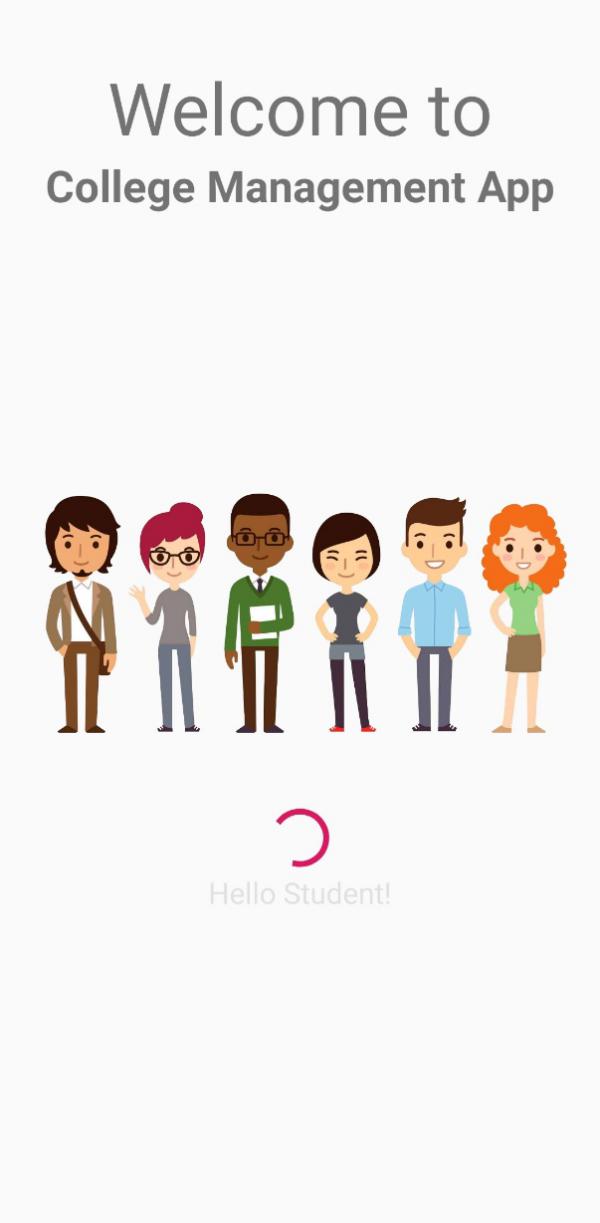
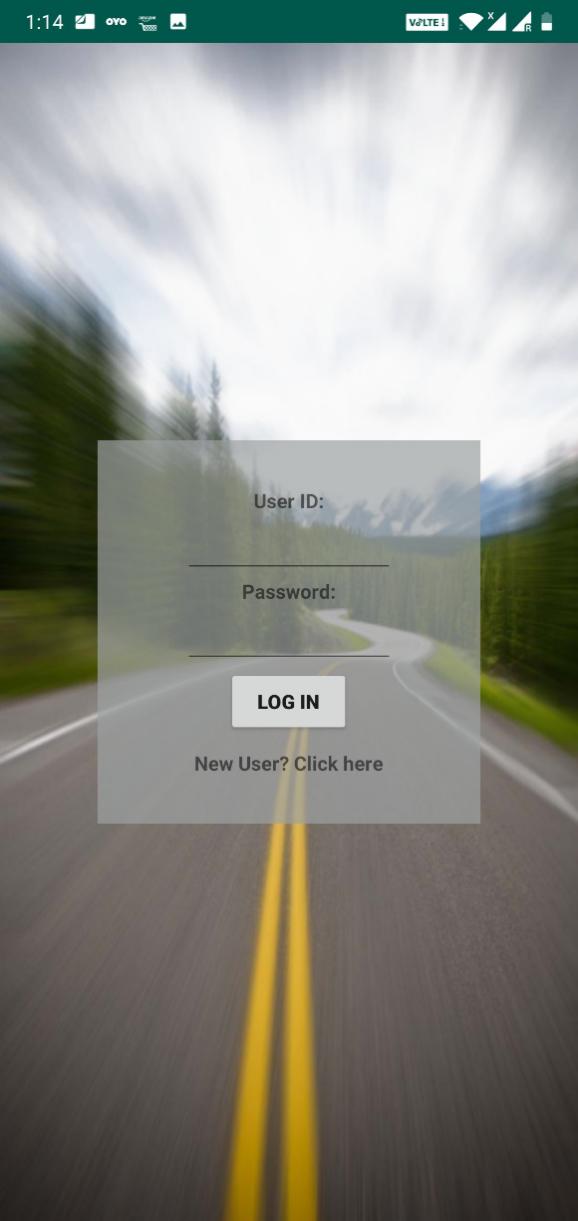
**HomeScreen.java**

**package** nicapp.nic.bihar.nicapp;  
  
**import** android.app.AlertDialog;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.SharedPreferences;  
**import** android.preference.PreferenceManager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.support.v7.widget.CardView;  
**import** android.view.View;  
**import** android.widget.Toast;  
  
**import** java.io.IOException;  
  
**import** nicapp.nic.bihar.nicapp.Model.RegistrationModel;  
**import** nicapp.nic.bihar.nicapp.Model.UpdateModel;  
**import** nicapp.nic.bihar.nicapp.database.DatabaseHelper;  
  
**public class** HomeScreen **extends** AppCompatActivity {  
 CardView c1, c2, c3, c4, c5, c6;  
 **int** ch=1;  
 DatabaseHelper mDatabase;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_home\_screen);  
 mDatabase = **new** DatabaseHelper(HomeScreen.**this**);  
 */\*DatabaseHelper mDatabase = null;  
 try {  
 mDatabase.createDataBase();  
 } catch (IOException ioe) {  
  
 throw new Error("Unable to create database");  
 }  
  
 mDatabase.openDataBase();  
  
\*/* c1 = findViewById(R.id.card1);  
 c1.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent i1 = **new** Intent(HomeScreen.**this**, Profile.**class**);  
 startActivity(i1);  
 }  
  
 });  
 c2 = findViewById(R.id.card2);  
 String username =PreferenceManager.getDefaultSharedPreferences(HomeScreen.**this**).getString(**"User\_ID"**, **""**);  
 ch = mDatabase.CheckInfo(username);  
 c2.setOnClickListener(**new** View.OnClickListener() {  
 **public void** onClick (View v){  
 **if**(ch==0) {  
 Toast.makeText(HomeScreen.**this**, **"Go To Edit Tab"**, Toast.LENGTH\_SHORT).show();  
 AlertDialog.Builder alertDialogBuilder = **new** AlertDialog.Builder(HomeScreen.**this**);  
 alertDialogBuilder.setMessage(**"You Can Only Update Your Profile once!"** +  
 **"Go to Edit Tab to make any changes"**);  
 AlertDialog alertDialog = alertDialogBuilder.create();  
 alertDialog.show();  
  
 }  
 **else**{  
 Intent i = **new** Intent(HomeScreen.**this**, Update.**class**);  
 startActivity(i);  
 finish();  
  
  
 }  
  
  
 }  
 });  
 c3 = findViewById(R.id.card3);  
 c3.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if**(ch==0){  
 Intent i3 = **new** Intent(HomeScreen.**this**, Edit.**class**);  
 startActivity(i3);}  
 **else** Toast.makeText(HomeScreen.**this**, **"Frist Enter Data in Update"**, Toast.LENGTH\_SHORT).show();  
 }  
 });  
 c6 = findViewById(R.id.card6);  
 c6.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent i1 = **new** Intent(HomeScreen.**this**, MainActivity.**class**);  
 PreferenceManager.getDefaultSharedPreferences(getApplicationContext()).edit().putString(**"User\_ID"**, **""**).commit();  
  
 PreferenceManager.getDefaultSharedPreferences(getApplicationContext()).edit().putString(**"Password"**, **""**).commit();  
 finish();  
 startActivity(i1);  
 }  
  
 });  
 }  
  
}

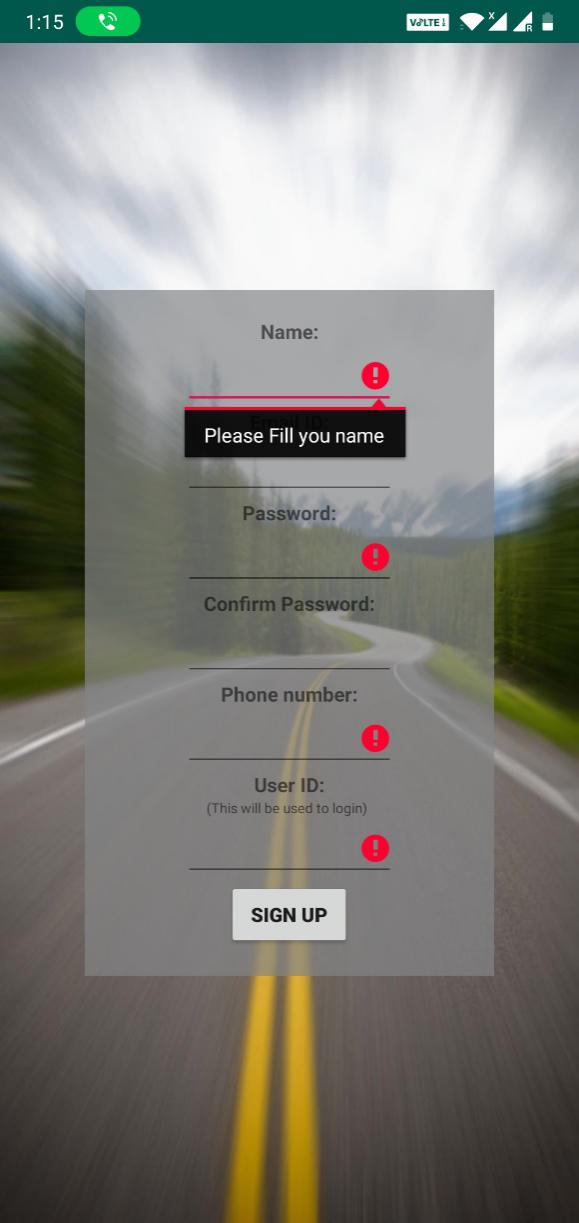
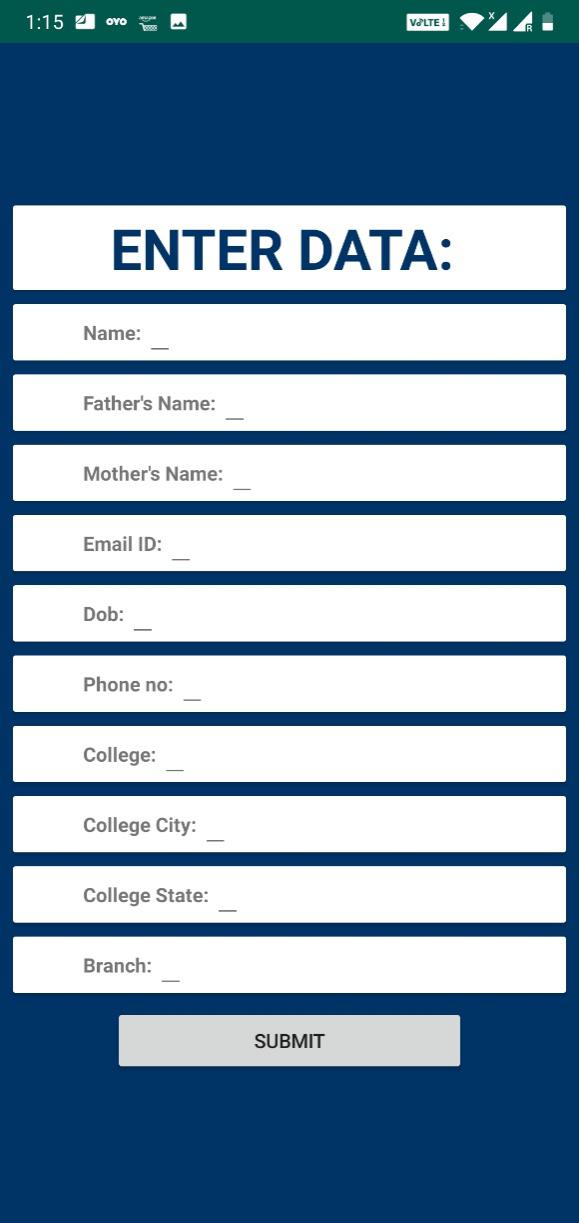
**HomeScreen.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".HomeScreen"  
 android:padding="35sp"  
 android:background="@drawable/wallp"  
 android:layout\_gravity="center"**>  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_centerInParent="true"  
 android:alpha="0.7"**>  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**android.support.v7.widget.CardView  
 android:layout\_height="200dp"  
 android:layout\_width="150dp"  
 app:cardCornerRadius="10sp"  
 android:id="@+id/card1"**>  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center"**>  
 <**ImageView  
 android:layout\_width="80sp"  
 android:layout\_height="80sp"  
 android:src="@drawable/profilepic"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="View Profile"**/>  
 </**LinearLayout**>  
 </**android.support.v7.widget.CardView**>  
 <**android.support.v7.widget.CardView  
 android:layout\_height="200dp"  
 android:layout\_width="150dp"  
 app:cardCornerRadius="10sp"  
 android:id="@+id/card2"  
 android:layout\_marginTop="20sp"  
 android:layout\_marginLeft="20sp"**>  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center"**>  
 <**ImageView  
 android:layout\_width="80sp"  
 android:layout\_height="80sp"  
 android:src="@drawable/update"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Update Profile"  
 android:textAlignment="center"**/>  
 </**LinearLayout**>  
 </**android.support.v7.widget.CardView**>  
 </**LinearLayout**>  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"**>  
 <**android.support.v7.widget.CardView  
 android:layout\_height="200dp"  
 app:cardCornerRadius="10sp"  
 android:id="@+id/card3"  
 android:layout\_width="150dp"**>  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center"**>  
 <**ImageView  
 android:layout\_width="80sp"  
 android:layout\_height="80sp"  
 android:src="@drawable/editlog"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Edit Profile"  
 android:textAlignment="center"**/>  
 </**LinearLayout**>  
 </**android.support.v7.widget.CardView**>  
 <**android.support.v7.widget.CardView  
 android:layout\_height="200dp"  
 android:layout\_width="150dp"  
 android:id="@+id/card6"  
 android:layout\_marginTop="20sp"  
 app:cardCornerRadius="10sp"  
 android:layout\_marginLeft="20sp"**>  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center"**>  
 <**ImageView  
 android:layout\_width="80sp"  
 android:layout\_height="100sp"  
 android:src="@drawable/logout"**/>  
 <**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textAlignment="center"  
 android:text="Logout"**/>  
 </**LinearLayout**>  
 </**android.support.v7.widget.CardView**>  
 </**LinearLayout**>  
  
 </**LinearLayout**>  
</**RelativeLayout**>

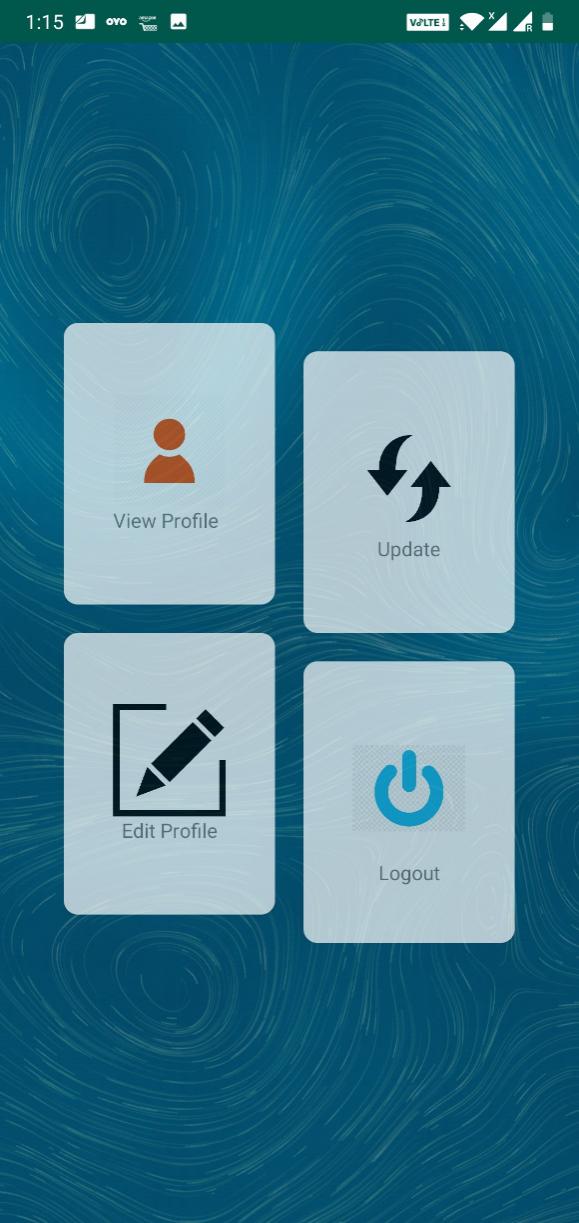
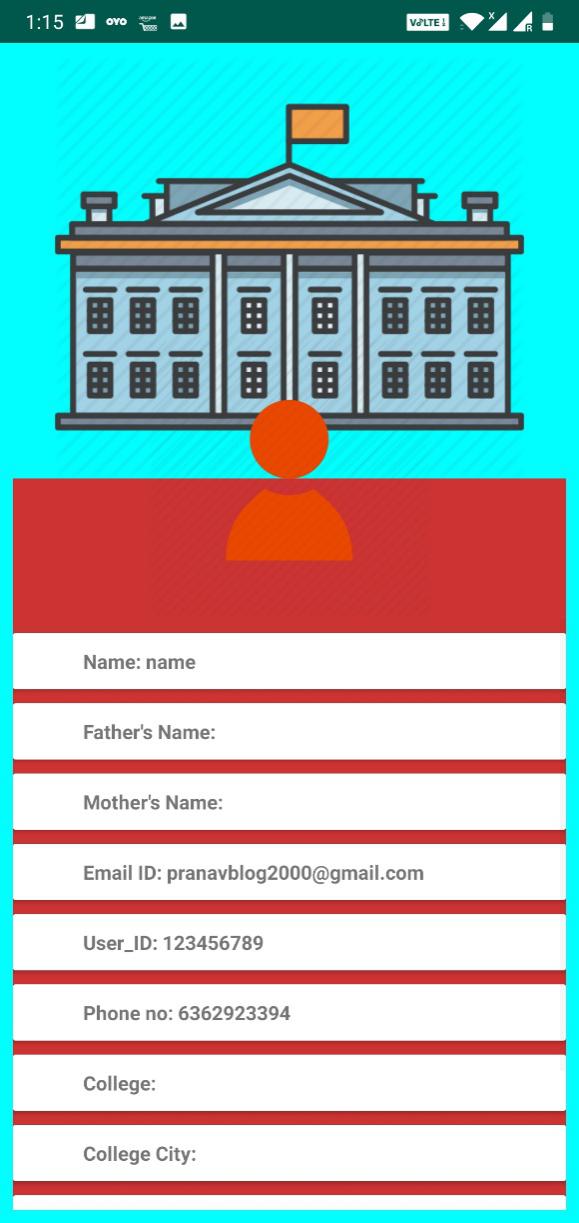
***SCREENSHOTS FROM MY APP***

**** ****

**SPLASH SCREEN LOGIN PAGE**

** **

**SIGNUP PAGE EDIT PAGE**

** **

**HOME SCREEN PROFILE PAGE**

****

**UPDATE PAGE**

**FUTURE SCOPE OF THE PROJECT**

The College Management System is saving time and manpower as well as providng a transparency throughtout the process. Work can be done further and simplified and integrate current system with online Database options or online server for global usages. The project has a great future for serving more number of candidates in an organized an efficient manner. The project College Management system exhibits pre possibilities for further development.

CONCLUSION

**I have tried my best to complete this project successfully, amidst a pleasures environment of constant cooperation and remarkable guidance.**

BIBLIOGRAPHY

For the development of this project, I have taken reference from several books and internet and shared those experiences during the designing, feasibility and actual coding of the project , here are the few :-

* Learning Java Basics (www.javatpoint.com)
* Tutorials from Youtube. (Telusko Learning)
* Google for reference.

THANK YOU