

**SIX WEEK SUMMER TRAINING**

**REPORT**

ON

**ANDROID**

Submitted By:

**AMAN SINHA**

**Reg. No.: 11711845**

**Program: Computer Science Engineering**

Under the guidance of

**Mr. ANUP RANA**

**School of Computer Science and Engineering**

**Lovely Professional University, Phagwara**

**(5 June - 16 July 2019)**

**Declaration**

I hereby declare that I have completed my six weeks training conducted at CETPA Infotech Private Limited, Dehradun from 5 June to 16 July under the guidance of Anup Rana. I further declare that I have worked with full dedication during these six weeks of training and my learning outcome will fulfil the requirements of training for the award of degree of Computer Science and Engineering, Lovely Professional University

(Signature of student)

Name of student: AMAN SINHA

Reg. No.: 11711845

Date:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgement**

I wish to express my gratitude to, my tutor at CETPA Infotech Private Limited, Mr. Anup Rana for providing me an opportunity to do my Summer Training and project work in “Android App Development”

Under his guidance I have completed my project and tried my best to implement what I had learnt till now.

I sincerely thank my Faculty for their guidance and encouragement to do my internship. They also help me by updating us about the information of what to do and not to do during our internship and help us with all. I also thanks my friend for helping me with my problem that I face in my project.

**CERTIFICATE**



**CONTENT**

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **TOPIC** | **Page Number** |
| **1.** | **Introduction** | **6-8** |
| **2.** | **Profile of Problem** | **9** |
| **3.** | **Existing System** | **9** |
| **4.** | **Software Requirement Analysis** | **10** |
| **5.** | **Design** | **11-12** |
| **6.** | **Code** | **13-19** |
| **7.** | **Features Used in Application** | **20-27** |
| **8.** | **Testing** | **28-29** |
| **9.** | **GANTT Chart** | **30** |
| **10.** | **Project Legacy** | **30** |
| **11.** | **Bibliography** | **31** |

**INTRODUCTION**

**What is an app or an application?**

An application is a software program that is designed to a specific function directly for the user which can be access easily.

**What is Android?**

Android is an open source operating System for mobile devices such as smart-phones and tablet, computers. Android offers a unified approach to application development for mobile devices which means developers need to develop only for Android, and their applications should be able to run on different devices powered by android.

Android was developed by the Open Handset Alliance (OHA), led by Google, and other companies. Android is mainly based on direct manipulation, using touch gestures that loses correspond to real world actions. It provides us with the manipulate on-screen object, along with a virtual keyboard for text input.

**History of Android:**

Android was initially developed by Android Inc., the code names of android ranges from A to N. Founded by Andy Rubin in Palo Alto, California, United States in Oct 2003.

Android word is actually referred to as robot.

Android was named after the nickname of Andy Rubin by their co-workers for his love for robot.

Google acquired android Incorporation on 17th Aug,2005.

Android was developed by the Open Handset Alliance (OHA) for camera phone, led by Google, and other companies. But shifted to smart-phones due to low market of cameras at that time.

HTC launches the first android mobile.

In 2010, Google launched its Nexus series of devices in smart-phone world with android OS.

**Android Version:**

1. Android 1.0, 1.1(Base, Base\_1\_1)

2. Android 1.5(Cupcake)

3. Android 1.6(Donut)

4. Android 2.0(Eclair)

5. Android 2.0.1(Eclair\_0\_1)

6. Android 2.1.x(Eclair\_MR1)

7. Android 2.2.x(Froyo)

8. Android 2.3 - 2.3.2(Gingerbread)

9. Android 2.3.3 - 2.3.4(Gingerbread\_MR1)

10. Android 3.0.x, 3.1.x, 3.2(Honeycomb, HC \_MR1, HC\_Mr2)

11. Android 4.0 - 4.0.2, 4.0.3 - 4.0.4((Ice\_Cream\_Sandwich, ICS\_MR1)

12. Android 4.1 - 4.1.1, 4.2 - 4.2.2,4.3(Jellybean, JB\_MR1, JB\_MR2)

13. Android 4.4, 4.4W (KitKat, K\_Watch)

14. Android 5.0, 5.1(Lollipop, L\_MR1)

15. Android 6.0 (Marshmallow)

16. Android 7.0 (Nougat)

17. Android 8.0-8.1(Oreo)

18. Android 9.0(Pie)

**Android App Development**

Android app is a combination of different source code in a single place whose action can be performed just by a single touch.

Android programming is based on java programming language so if we have basic understanding on Java programming then it will be a fun to study Android app development.

**Java in Android App Development**

Java is a programming language that doesn’t compile to native processor code but rather it refers to virtual machine which understands an intermediate format i.e.; java byte-code. Each platform that uses java to run needs a virtual device.

An android app uses a android application that runs on android platform. It build on custom virtual machine that gives its user the addition usage and application power and a user friendly environment. Android actual virtual machine is called Dalvik.

**Android Software Development Kit (Android SDK)** :

Apps that extend the functionality of device is written using Android SDK and often using java programming language. The SDK includes a set of development tools, including a debugger, software libraries, a handset emulator, sample code, etc.

Initially Google supported Integrated Development Environment (IDE) i.e.; Eclipse using the Android Development Tools (ADT) plugins. Other development tools are also available such as Native Development Kit (NDK). Android Studio that is based on IntelliJ IDEA developed by Google as its primary IDE for android app development. Android is a selection of third-party application which can be acquired by users by downloading and installing the Android Application Package (APK) file.

**What is API Level?**

API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.

**Features of Android**:

1. It is an open source user friendly software
2. It has beautiful user interface
3. It reduced cost of development
4. It has rich development environment
5. Inter application Integration
6. It supports single and bi-directional text

**Android Application:**

They are generally developed in the java language using the Android SDK

Android applications can be packaged easily and sold out either through a store such as Google play, etc. There are many android applications that we already know and uses them few are music, news, weather, etc

Many android applications are also available for free most of them are already available in play store.

This apps are compatible with almost every platform but few are also available for the specific platform.

**Profile of the Problem:**

My app GoEazy is an application which can be used to store the data of various students related to the choices given by the application.

The Whole project has been created on Android Studio Application by importing various libraries and various different dependencies in the application.

The application can be run in the mobile phone which has the android above 2.3 - 2.3.2(Gingerbread).

The app also has been added with Realtime database of google firebase which provides various cloud and authentication features.

The Various features of the app are:

* Store the data of Student in Student Management Option.
* Store the data of the Queries which the student has in Queries Management Option.
* Store the marks of the Student in Marks Management Option.

**Existing System:**

The system which is used nowadays has some drawbacks which need to be improved for better performance. The system through which the feedback is taken is not good enough. The views of each and every student are not expressed through these systems. As the technology is developed day by day, we need to use this technology so we can get an efficient result in adequate time. This is a very time-consuming process.

In the present system the result is viewed on the notice board. It requires lot of paperwork and is time consuming. Moreover, there is no system still present through which students can take advice from senior students. College cannot even provide urgent notifications to students in case of emergency.

Demerits of Existing System: -.

* It difficult to generate insights from data.
* It is not easy to predict data manually in existing system.
* Not possible to train model.
* Time consuming.

**Software Requirement Analysis**

Android Studio is requried to make an Android Project.

**Android Studio** is the officialintegrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.  The first stable build was released in December 2014, starting from version 1.0.

Since 7 May 2019, Kotlin is Google’s preferred language for Android app development. Still, other languages are supported, including by Android Studio.



**Hardware Requirement of Android Studio**:

1. 64-bit distribution capable of running 32-bit.
2. 3GB RAM minimum,8GB RAM recommended; plus, 1GB for the Android Emulator.
3. 2GB of Available disk space minimum,4Gb Recommended (500MB for IDE + 1.5GB for Android SDK and emulator system image)
4. 1280 x 800 minimum screen resolution.

**Software Requirement of Android Studio:**

1-Windows, Linux or Mac

**Design**:

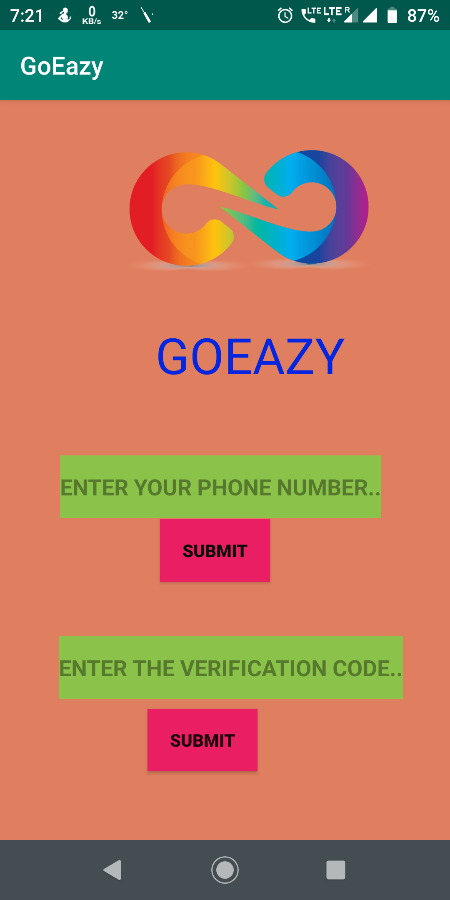
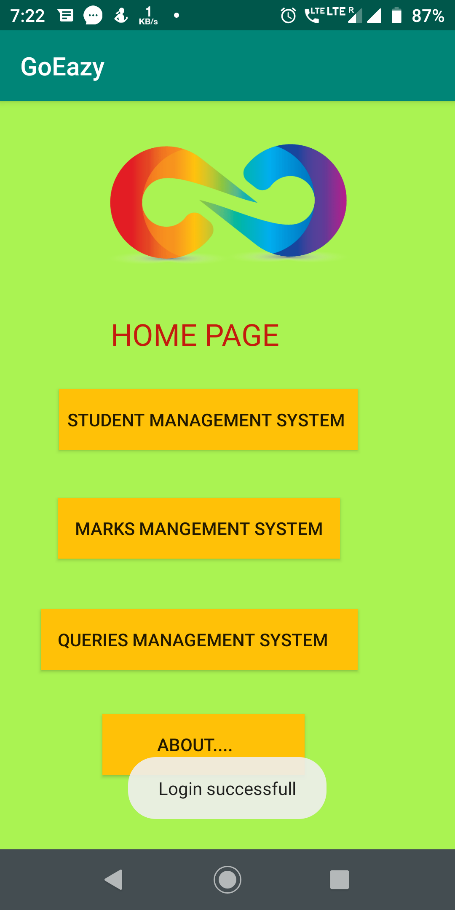
Various elements of the app are:

1-LOGIN PAGE

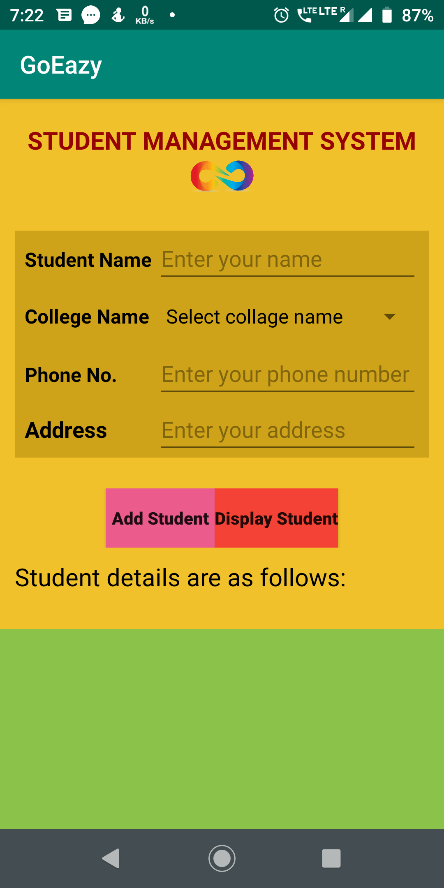
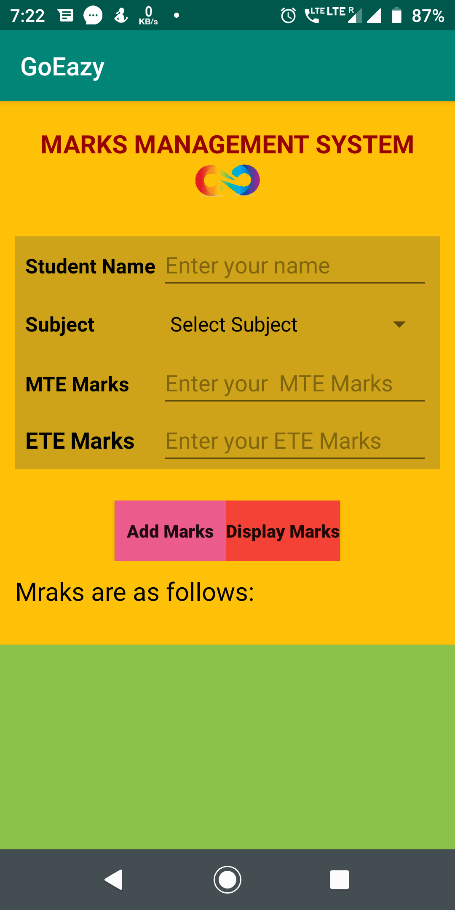
2-HOME PAGE

1. STUDENT MANAGEMENT SYSTEM
2. MARKS MANAGEMENT SYSTEM
3. QUERIES MANAGEMENT SYSTEM
4. ABOUT

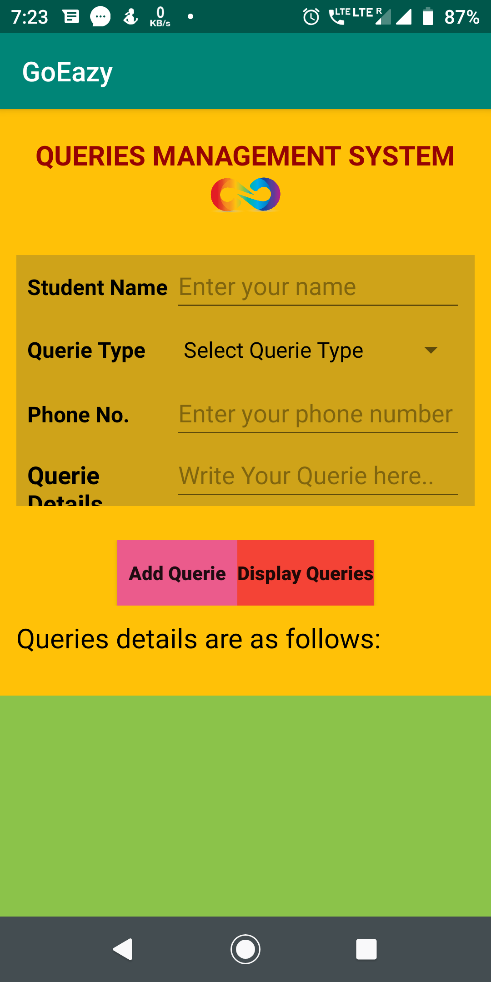
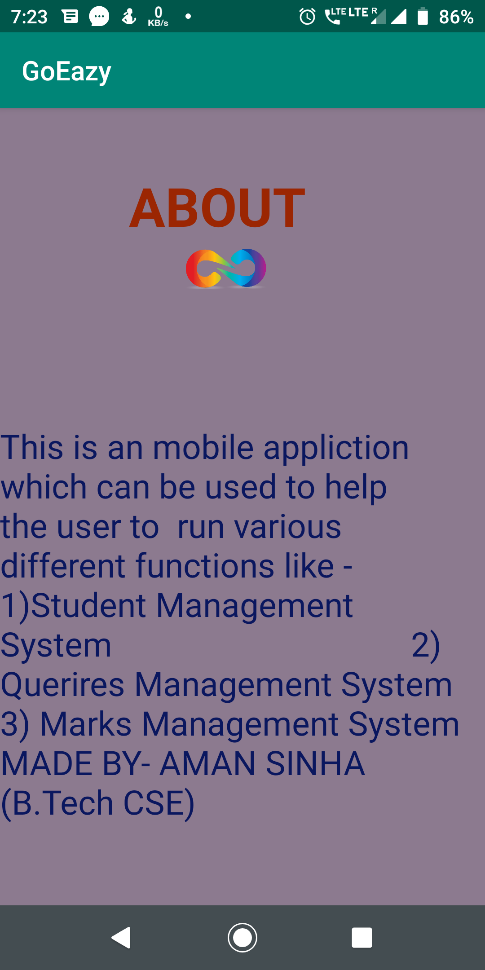
**LOGIN PAGE HOME PAGE**

**STUDENT MANAGEMENT SYSTEM MARKS MANAGEMENT SYSTEM**

**QUERIES MANAGEMENT SYSTEM ABOUT**

**CODE**

**LOGIN PAGE**

public class MainActivity extends AppCompatActivity {

EditText et1,et2;

Button b1,b2;

String codeSent;

FirebaseAuth mAuth;

@SuppressLint("WrongViewCast")

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mAuth=FirebaseAuth.getInstance();

et1 = findViewById(R.id.editTextPhone);

et2 = findViewById(R.id.editTextxCode);

b1=findViewById(R.id.buttongetverificationcode);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sendVerificationCode();

}

});

b2=findViewById(R.id.buttonSignIn);

b2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

veryfySignInCode();

}

}); }

private void veryfySignInCode(){

String code=et2.getText().toString();

PhoneAuthCredential credential = PhoneAuthProvider.getCredential(codeSent, code);

signInWithPhoneAuthCredential(credential);

}

private void signInWithPhoneAuthCredential(PhoneAuthCredential credential) {

mAuth.signInWithCredential(credential)

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

Toast.makeText(getApplicationContext(),"Login successfull",Toast.LENGTH\_SHORT).show();

Intent intent=new Intent(MainActivity.this,homepage.class);

startActivity(intent);

} else {

if(task.getException()instanceof FirebaseAuthInvalidCredentialsException) {

Toast.makeText(getApplicationContext(),"Login invalid",Toast.LENGTH\_SHORT).show();

}}} });}

public void sendVerificationCode() {

String phone = et1.getText().toString();

if (phone.isEmpty()) {

et1.setError("phone no is empty ");

et1.requestFocus();

return;

}

if (phone.length() < 10) {

et1.setError(" enter a valid phone no");

et1.requestFocus();

return;

}

PhoneAuthProvider.getInstance().verifyPhoneNumber(

phone, // Phone number to verify

60, // Timeout duration

TimeUnit.SECONDS, // Unit of timeout

this, // Activity (for callback binding)

mCallbacks); // OnVerificationStateChangedCallbacksPhoneAuthActivity.java

}

PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks = new PhoneAuthProvider.OnVerificationStateChangedCallbacks() {

@Override

public void onVerificationCompleted(PhoneAuthCredential phoneAuthCredential) {

}

@Override

public void onCodeSent(String s, PhoneAuthProvider.ForceResendingToken forceResendingToken) {

super.onCodeSent(s, forceResendingToken);

codeSent=s;

};}

**HOME PAGE**

package com.example.goeazy;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

public class homepage extends AppCompatActivity {

Button b1,b2,b3,b4;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.homepage);

b1=findViewById(R.id.b1);

b2=findViewById(R.id.b2);

b3=findViewById(R.id.b3);

b4=findViewById(R.id.b4);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent=new Intent(homepage.this,StudentMang.class);

startActivity(intent);

}

});

b3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent=new Intent(homepage.this,QuereiesMang.class);

startActivity(intent);

}

});

b2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent=new Intent(homepage.this,marksMang.class);

startActivity(intent) }

});

b4.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent=new Intent(homepage.this,about.class);

startActivity(intent); });

**STUDENT MANAGEMENT SYSTEM**

package com.example.goeazy;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

import java.util.ArrayList;

public class StudentMang extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

static final String TAG=StudentMang.class.getName();

TextView tv7;

EditText studentNameTF,studentPhoneTF,studentAddressTF;

Button addstudent,viewstudent;

Spinner;

String collageName="";

ArrayList<StudentModel> studentArrayList=new ArrayList<>();

String collageNames[]={"Select collage name","DIT","Graphic Era","LPU"};

DataBaseHelper1 dataBaseHelper;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.studentmang);

tv7=findViewById(R.id.tv7);

studentNameTF=findViewById(R.id.et1);

studentAddressTF=findViewById(R.id.et3);

studentPhoneTF=findViewById(R.id.et2);

addstudent=findViewById(R.id.b1);

viewstudent=findViewById(R.id.b2);

spinner=findViewById(R.id.spinner);

dataBaseHelper= new DataBaseHelper1(this);

spinner.setOnItemSelectedListener(this);

ArrayAdapter arrayAdapter=new ArrayAdapter(this,android.R.layout.simple\_spinner\_dropdown\_item,collageNames);

spinner.setAdapter(arrayAdapter);

spinner.setPrompt(collageNames[0]);

addstudent.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String name=studentNameTF.getText().toString();

int phone=Integer.parseInt(studentPhoneTF.getText().toString());

String address=studentAddressTF.getText().toString();

String collageName=spinner.getSelectedItem().toString();

dataBaseHelper.addNewStudent(new StudentModel(name,collageName,address,phone));

Toast.makeText(getApplicationContext(),"Student data saved",Toast.LENGTH\_SHORT).show();

}

});

viewstudent.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

studentArrayList.addAll(dataBaseHelper.allstudentDetails());

for (int i=0;i<studentArrayList.size();i++){

tv7.setText(tv7.getText() + "Student ID is : "+studentArrayList.get(i).getId() +"\n");

tv7.setText(tv7.getText() + "Student Name is : "+studentArrayList.get(i).getName() +"\n");

tv7.setText(tv7.getText() + "Student Collage is : "+studentArrayList.get(i).getCollageName() +"\n");

tv7.setText(tv7.getText() + "Student Phone No. is : "+studentArrayList.get(i).getPhone() +"\n");

tv7.setText(tv7.getText() + "Student Address is : "+studentArrayList.get(i).getAddress() +"\n");

tv7.setText(tv7.getText() + "\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n"); }}});}

@Override

public void onItemSelected(AdapterView<?> parent, View, int position, long id) {

}

@Override

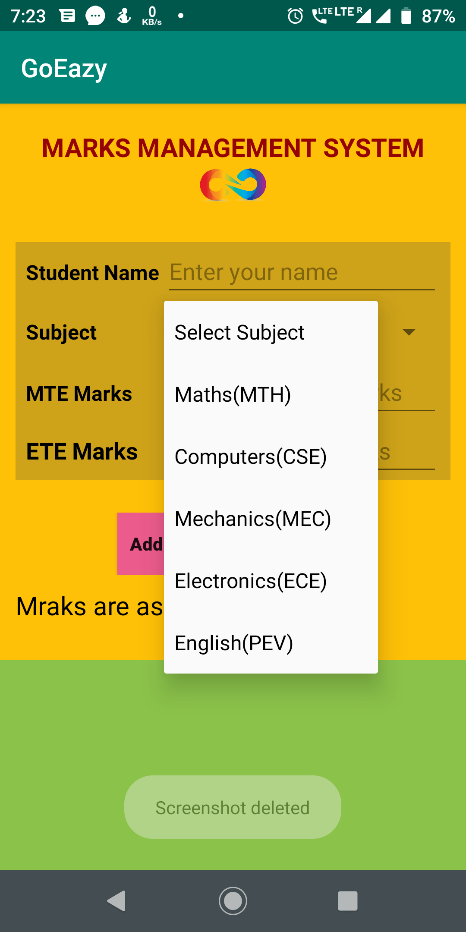
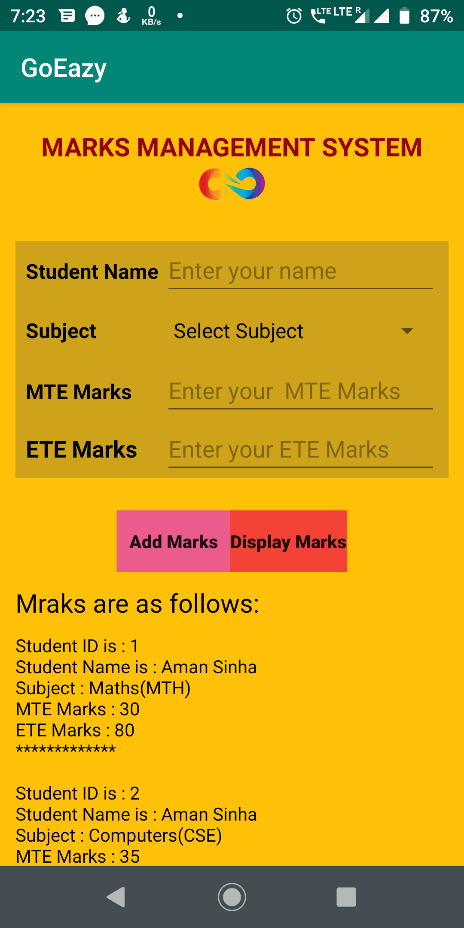
public void onNothingSelected(AdapterView<?> parent) {

}

}

**Various Features used in the application:**

Widgets used:

Scroll View

Edit Text

Text View

Spinner

Button

ImageView

<Scroll View xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#8BC34A">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#F1C12B"  
 android:orientation="vertical"  
 android:paddingLeft="12dp"  
 android:paddingRight="12dp">  
  
 <TextView  
 android:id="@+id/tv1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp"  
 android:text="Student Management System"  
 android:textAllCaps="true”  
 android:textColor="#960303"  
 android:textSize="20sp"  
 android:textStyle="bold" />  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="71dp"  
 android:layout\_height="30dp"  
 android:layout\_gravity="center"  
 android:foregroundGravity="center"  
 app:srcCompat="@drawable/img1" />

<Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="2" />

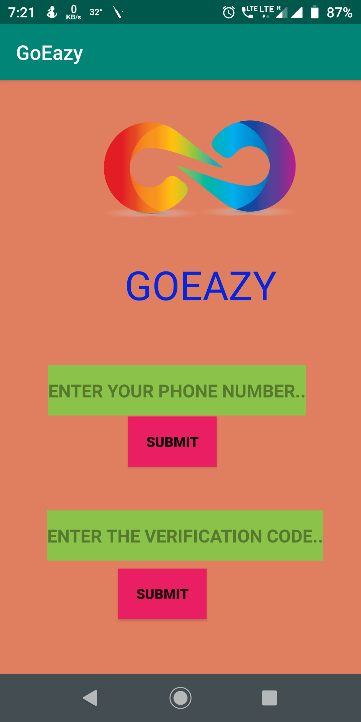
<Button  
 android:id="@+id/b1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:background="#EB5B8C"  
 android:text="Add Student"  
 android:textAllCaps="false"  
 android:textStyle="bold" />

</LinearLayout>  
  
</ScrollView>

Splash Activity:

Splash Screen is most commonly the first startup screen which appears when App is opened. In other words, it is a simple constant screen for a fixed amount of time which is used to display the company logo, name, advertising content etc.

Normally it shows when app is first time launched on android device or it may be some kind of process that is used to show screen to user just before the app loads completely.

package com.example.goeazy;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import android.support.v7.app.AppCompatActivity;  
  
public class SplashActivity extends AppCompatActivity {  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_splash*);  
  
 getSupportActionBar().hide();  
  
 CountDownTimer = new CountDownTimer(3000,100) {  
 @Override  
 public void onTick(long millisUntilFinished)}  
 @Override  
 public void onFinish() {  
 Intent =new Intent(SplashActivity.this,MainActivity.class);  
 startActivity(intent);  
 finish();  
} };countDownTimer.start();}}

Intent:

Android uses Intent for communicating between the components of an Application and also from one application to another application.

Intent are the objects which is used in android for passing the information among Activities in an Application and from one app to another also. Intent are used for communicating between the Application components and it also provides the connectivity between two apps.

For example: Intent facilitate you to redirect your activity to another activity on occurrence of any event. By calling, startActivity() you can perform this task.

b4.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent=new Intent(homepage.this,about.class);  
 startActivity(intent);  
  
 }  
});

SQL

SQLite is a Structure query base database, open source, light weight, no network access and standalone database. It support embedded relational database features.

Whenever an application needs to store large amount of data then using sqlite is more preferable than other repository system like Shared Preferences or saving data in files.

Android has built in SQLite database implementation. It is available locally over the device(mobile & tablet) and contain data in text format. It carry light weight data and suitable with many languages. So, it doesn’t required any administration or setup procedure of the database.

CODE

package com.example.goeazy;

import android.content.ContentValues;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import java.util.ArrayList;

import java.util.List;

public class DataBaseHelper1 extends SQLiteOpenHelper {

private static final int DATABASE\_VERSION=1;

private static final String DATABASE\_NAME ="student\_db";

private static final String TABLE\_NAME="student\_record";

private static final String STUDENT\_NAME="student\_name";

private static final String STUDENT\_ID="student\_id";

private static final String STUDENT\_COLLAGE="student\_collage";

private static final String STUDENT\_ADDRESS="student\_address";

private static final String STUDENT\_PHONE\_NUMBER="student\_phone";

private static final String CREATE\_TABLE = "CREATE TABLE " + TABLE\_NAME + " ( " +STUDENT\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + STUDENT\_NAME + " TEXT, " +STUDENT\_COLLAGE + " TEXT, " + STUDENT\_ADDRESS+ " TEXT, " + STUDENT\_PHONE\_NUMBER + " INTEGER ); "

public DataBaseHelper1(Context context) {

super(context,DATABASE\_NAME,null,DATABASE\_VERSION);

}

public long addNewStudent(StudentModel studentModel) {

SQLiteDatabase sqLiteDatabase=this.getWritableDatabase();

ContentValues = new ContentValues();

contentValues.put(STUDENT\_NAME,studentModel.name);

contentValues.put(STUDENT\_COLLAGE,studentModel.collageName);

contentValues.put(STUDENT\_ADDRESS,studentModel.address);

contentValues.put(STUDENT\_PHONE\_NUMBER,studentModel.phone);

long id=sqLiteDatabase.insert(TABLE\_NAME,null,contentValues);

sqLiteDatabase.close();

return id; }

public List<StudentModel> allstudentDetails() {

List<StudentModel> studentsList = new ArrayList<>();

String selectQuery = " SELECT \* FROM " + TABLE\_NAME ;

SQLiteDatabase = this.getWritableDatabase();

Cursor = sqLiteDatabase.rawQuery(selectQuery, null);

if (cursor.moveToFirst()){

do { StudentModel = new StudentModel();

studentModel.setId(cursor.getInt(cursor.getColumnIndex(STUDENT\_ID))); studentModel.setName(cursor.getString(cursor.getColumnIndex(STUDENT\_NAME))); studentModel.setCollageName(cursor.getString(cursor.getColumnIndex(STUDENT\_COLLAGE))); studentModel.setAddress(cursor.getString(cursor.getColumnIndex(STUDENT\_ADDRESS))); studentModel.setPhone(cursor.getLong(cursor.getColumnIndex(STUDENT\_PHONE\_NUMBER)));

studentsList.add(studentModel);

} while (cursor.moveToNext()); }

sqLiteDatabase.close(); return studentsList }public StudentModel getSingleStudentDetails(long id) {

SQLiteDatabase = this.getReadableDatabase();

Cursor = sqLiteDatabase.query(TABLE\_NAME, new String[]{STUDENT\_ID, STUDENT\_NAME, STUDENT\_COLLAGE, STUDENT\_ADDRESS,

STUDENT\_PHONE\_NUMBER}, STUDENT\_ID + "=?", new String[]{String.valueOf(id)}, null, null ,null, null);

if (cursor != null) { cursor.moveToFirst(); }

StudentModel = new StudentModel(cursor.getInt(cursor.getColumnIndex(STUDENT\_ID)),

cursor.getString(cursor.getColumnIndex(STUDENT\_NAME)), cursor.getString(cursor.getColumnIndex(STUDENT\_COLLAGE)),

cursor.getString(cursor.getColumnIndex(STUDENT\_ADDRESS)), cursor.getLong(cursor.getColumnIndex(STUDENT\_PHONE\_NUMBER)));

cursor.close(); return studentModel; }

public int getStudentsCount() {

String query = "SELECT \* FROM " + TABLE\_NAME;

SQLiteDatabase = this.getReadableDatabase();

Cursor = sqLiteDatabase.rawQuery(query, null);

int totalStudentsCount = cursor.getCount();

cursor.close(); return totalStudentsCount; }

public int updateIndividualStudentDetails(StudentModel studentModel) {

SQLiteDatabase = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(STUDENT\_NAME, studentModel.getName());

values.put(STUDENT\_COLLAGE, studentModel.getCollageName());

values.put(STUDENT\_ADDRESS, studentModel.getAddress());

values.put(STUDENT\_PHONE\_NUMBER, studentModel.getPhone());

return sqLiteDatabase.update(TABLE\_NAME, values, STUDENT\_ID + " = ?",

new String[]{String.valueOf(studentModel.getId())});

} @Override public void onCreate(SQLiteDatabase db) {

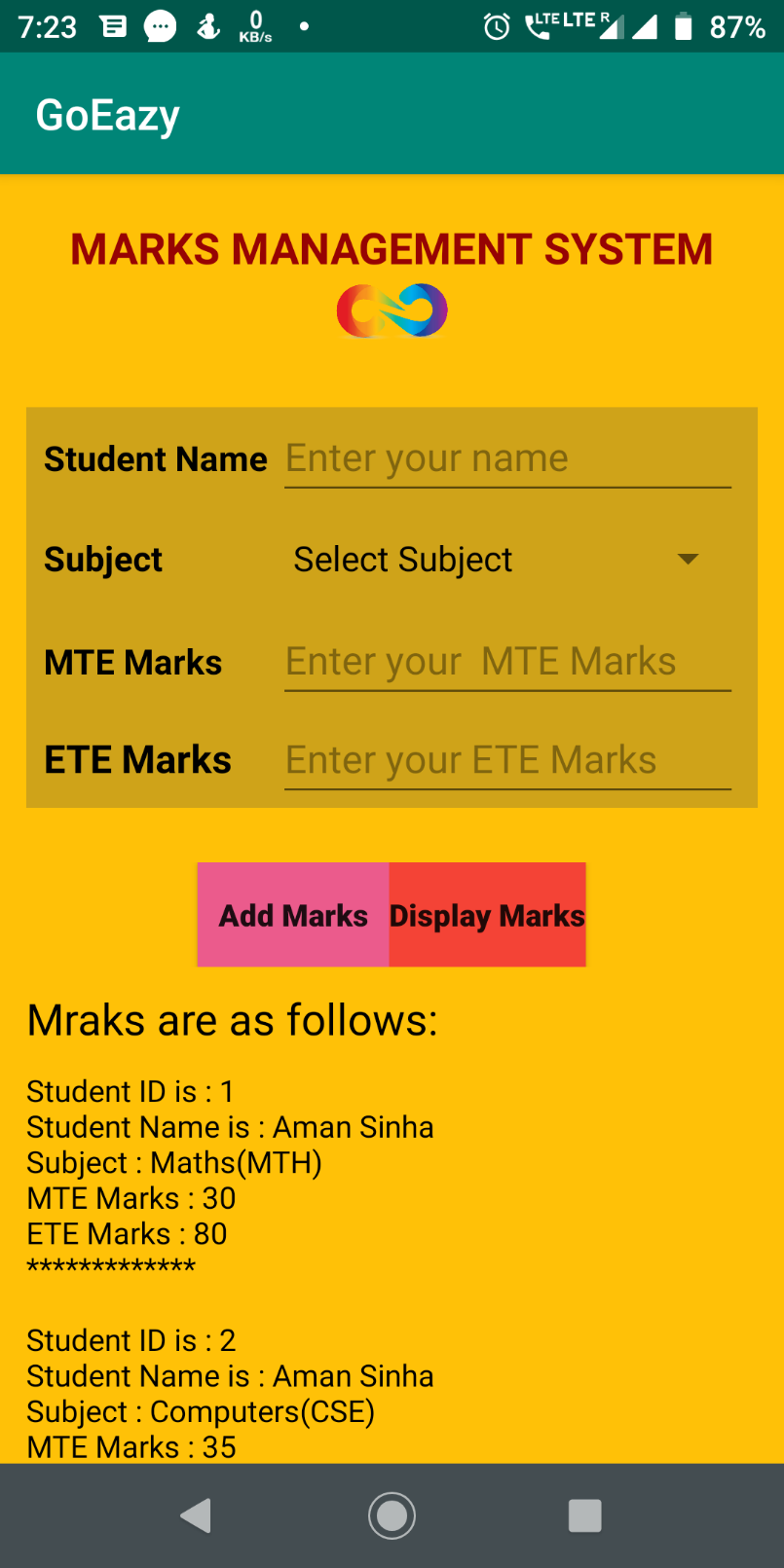
db.execSQL(CREATE\_TABLE);

}@Override

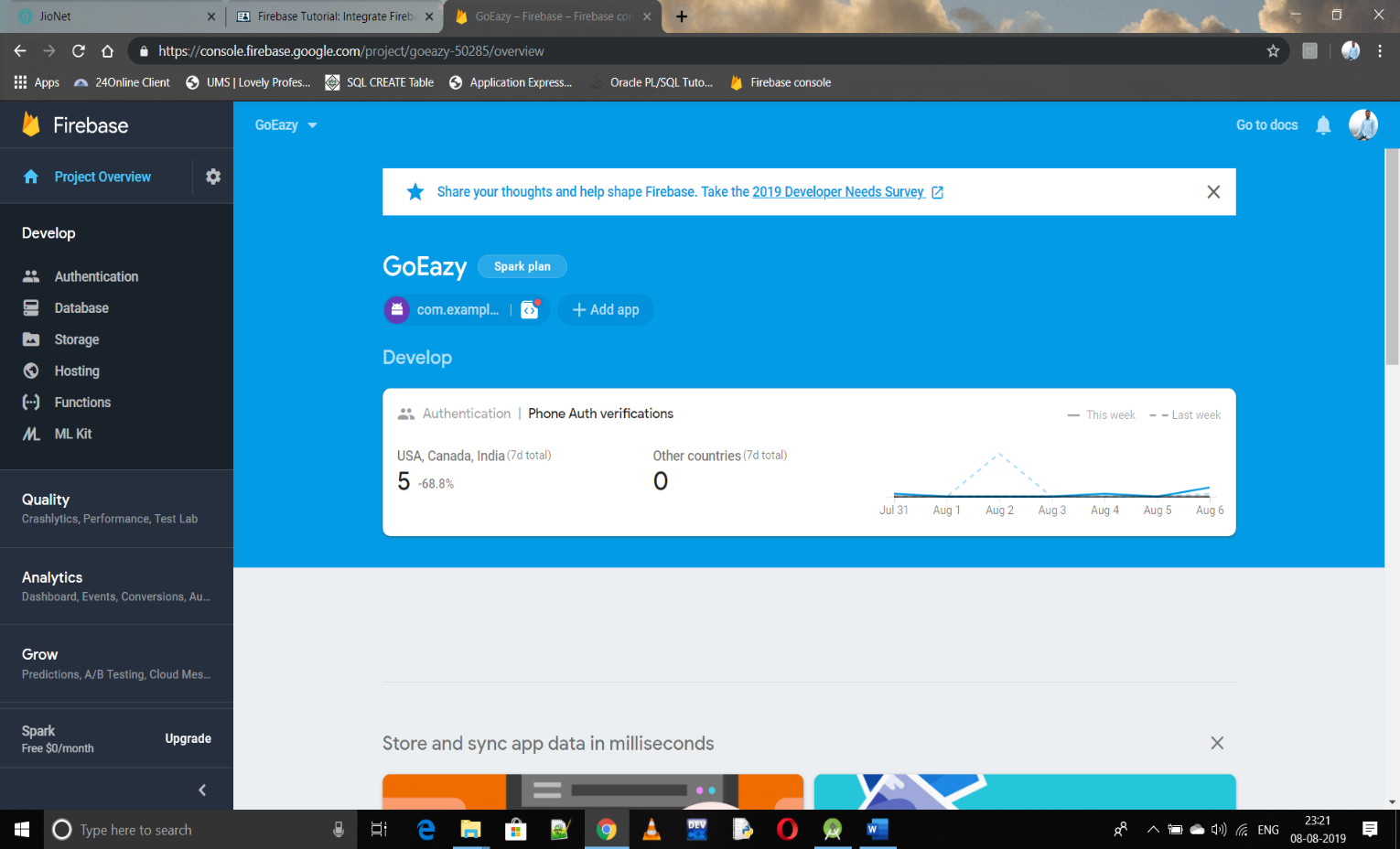
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);}}

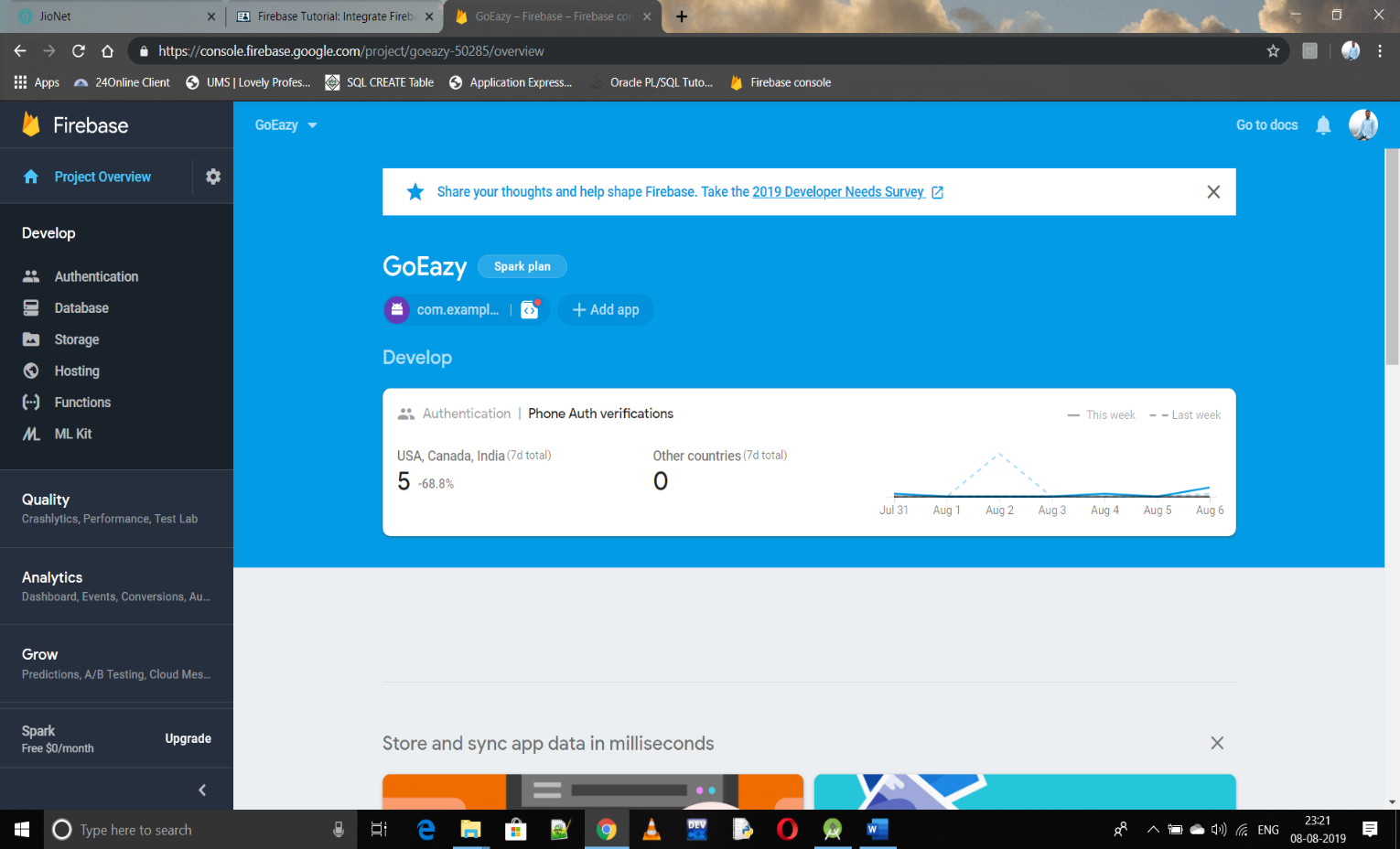


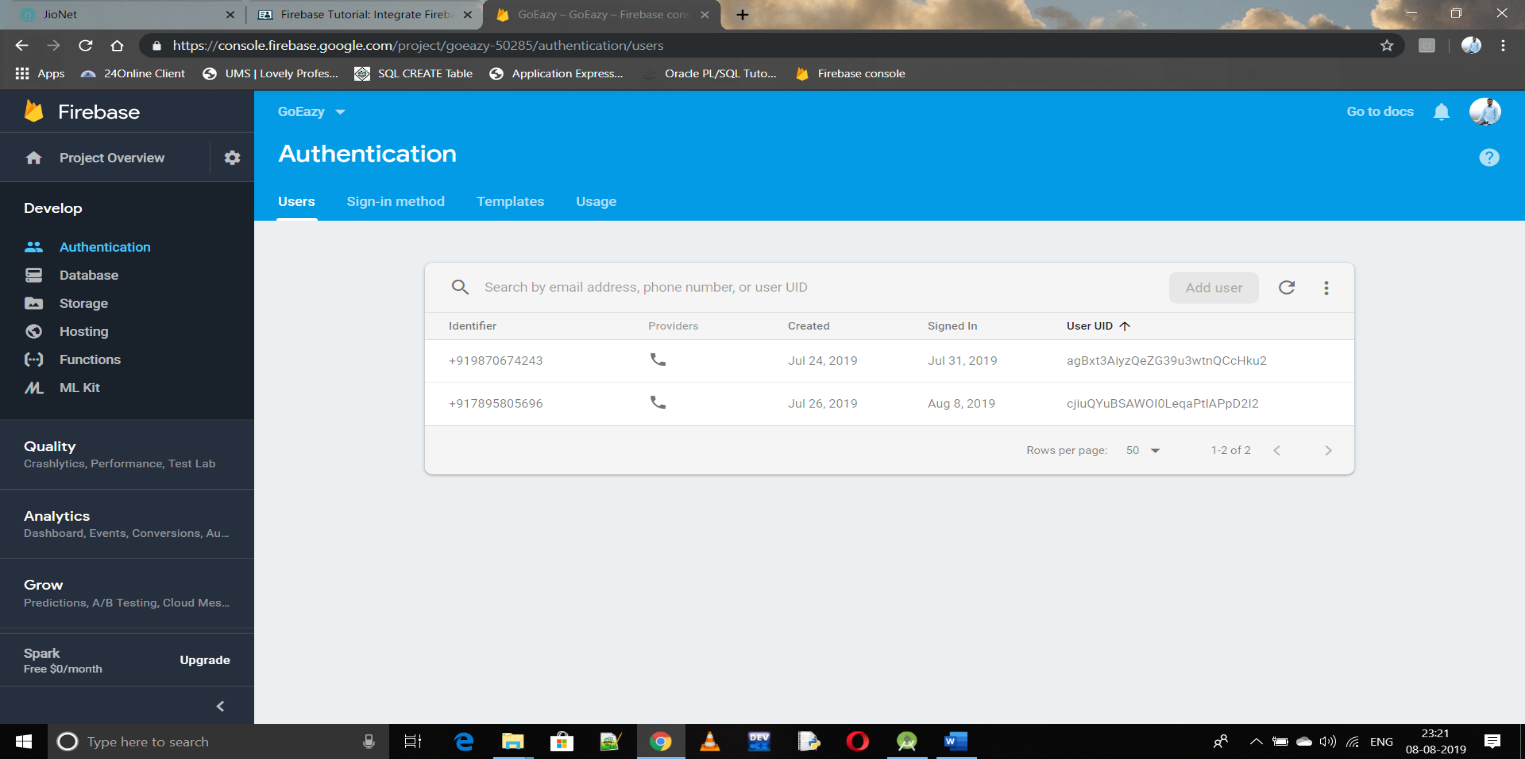
Firebase:

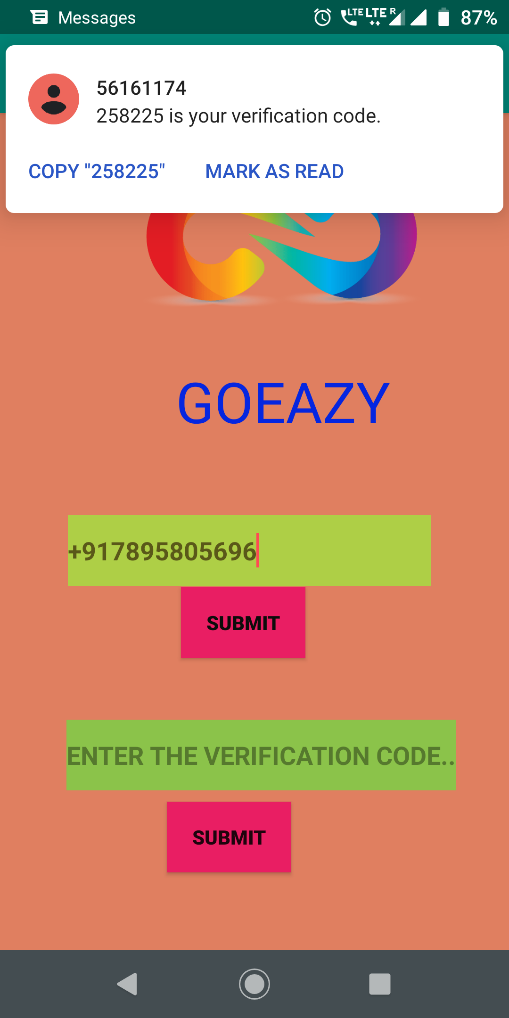


The Assistant tool window in Android Studio. Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. Firebase is made up of complementary features that you can mix-and-match to fit your needs, with Google Analytics for Firebase at the core.

Firebase provides a Realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.







After connecting with firebase

for authentication you we will get an OTP

for the authentication of the mobile

for a secure login process.

**Testing:**

Testing is the process carried out on software to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications.

Testing is advantageous in several ways. Firstly, the defects found help in the process of making the software reliable. Secondly, even if the defects found are not corrected, testing gives an idea as to how reliable the software is. Thirdly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

**Testing plan**

The testing sub-process includes the following activities in a phase dependent manner:

* Create Test Plans.
* Create Test Specifications.
* Review Test Plans and Test Specifications.
* Conduct tests according to the Test Specifications, and log the defects.
* Fix defects, if any.
* When defects are fixed continue from activity.

**Testing strategy**

The development process repeats this testing sub-process a number of times for the following phases.

* Unit Testing.
* Integration Testing
* System Testing.
* Acceptance Testing.

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

**Black Box and White Box Testing**

In black box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the structural system coverage in white box testing. For this purpose tools or compilers that can insert test probes into the programs can be used.

**Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No | Test Case Description | Expected Result | Actual Result | Remarks |
| 1. | Click on the application | App should open with splash activity | Same as expected. |  |
| 2. | Incorrect Phone Number | The error message telling invalid mobile number should be displayed. | Same as expected. |  |
| 3. | Valid OTP | Home page should open | Same as expected. |  |
| 4. | Click on Student Management System | Details page should open | Same as expected. |  |
| 5. | Click on Marks Management System | Marks Entering page should open | Same as expected. |  |
| 6. | Value Entering | Values should be entered | Same as expected. |  |
| 7. | Click on Add Data | Data added successfully toast should appear | Same as expected. |  |
| 8 | Click on View Data | Entered data Should appear in the scroll view | Same as expected. |  |

**GANTT CHART**

ANDROID

16 July

5 June

Introduction to Android……………….

Android Studio Introduction……………..

Various Widgets……………………………..

Button,ImageView,TextView,etc

AlertBox……………………………………….

Intent Activity……………………………………….

Fragments……………………………………………….

Layouts……………………………………………………….

Linear Layout,Relative Layout

SQLite………………………………………………………………..

Firebase…………………………………………………………………….

Project of GOEazy………………………………………………………………

**Project legacy:-**

**(a)Technical and Managerial Lesson Learnt:-**

The technology that I have learnt is Android. Android is more important and most popular language in today’s era. We can perform programming, built applications, can perform better GUI programming, etc

An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS.

Large parts of Android are written in Java and its APIs are designed to be called primarily from Java. It is possible to develop C and C++ app using the Android Native Development Kit (NDK), however it isn't something that Google promotes. According to Google, “the NDK will not benefit most apps.

**Bibliography**

1. https://developer.android.com/
2. https://console.firebase.google.com/
3. https://developer.android.com/studio
4. https://www.tutorialspoint.com/android/
5. https://www.geeksforgeeks.org/android-app-development-fundamentals/