# TECH AGRI APP

*A project report submitted in partial fulfillment of the Academic requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION TECHNOLOGY**

**By**

**Yashaswini.K (2451-16-737-029)**

***Under the guidance of***

**Ms.J.Sowjanya**<br>
**Assistant Professor, Dept. of I.T**<br>
**MVSR Engineering College**<br>
**Nadergul, Hyderabad.**

**10/14**

**DEPARTMENT OF INFORMATION TECHNOLOGY**<br>
**MVSR ENGINEERING COLLEGE**<br>
**(Affiliated to Osmania University, Hyderabad. Recognized by AICTE)**<br>
**Nadergul, Saroornagar Mandal, Hyderabad-501510**<br>
**2018-2019**

# ACKNOWLEDGEMENT

We with extreme jubilance and deepest gratitude, would like to thank Ms. J.Sowjanya, Assistant professor, Department of Information Technology, MVSR Engineering College, for his/her constant encouragement and facilities provided to us to complete our project in time.

With immense pleasure, we record our deep sense of gratitude to our beloved Head of the department Dr. Ch. Samson, Department of Information Technology, MVSR Engineering College, for permitting us to carry out this project.

We would like to extend our gratitude to Ms.Srilaxmi, Assistant professor, Mr.Muninder, Assistant professor and Mr.Karunakar, Lab Assistant, Department of Information Technology, M V S R Engineering College, for his valuable suggestions and timely help during the course of the project.

We express, from the bottom of my heart, our deepest gratitude to our parents and family for the support, dedication, and love.

Finally, we express our heartfelt thanks to each and every one who directly and indirectly helped us in successful completion of this project work.

K.Yashaswini

(245-16-737-029)

# M.V.S.R ENGINEERING COLLEGE

Affiliated to Osmania University, Hyderabad. Recognized by AICTE)

Nadergul, Saroornagar Mandal, Hyderabad-501510

DEPARTMENT OF INFORMATION TECHNOLOGY

Certificate

Certified that this a bonafide Mini Project II work carried out by Mrs K.Yashaswini___ Bearing H.T. No. 2451-16-737-029_ in the course Mini Project prescribed for B.E ( 3/4 ) Sem: _VI_ by Osmania University, in the department during the academic year 2018 - 2019

Faculty in-Charge

Guide Signature

Head of the Department

## ABSTRACT

- This Application is a software system developed to provide an interface between the Farmers and the sellers/distributors.
- This app helps in avoiding intermediaries which is favorable for farmers.

**CONTENT**

# 1. INTRODUCTION

## 1.1 EXISTING SYSTEM

- Every time farmers have to sell their crops at market to authorized distributors for fixed prices set by market management which is loss for them (this system works manually).

### 1.1.2) DISADVANTAGES

- Marketplaces can **charge commission** on every sale and fees can vary from site to site. Before selling your products on a marketplace, make sure you understand the marketplace's pricing structure.
- The online stores may impose restrictive **terms and conditions** in terms of how you can communicate to customers.
- There may be marketplace limitations as to how your business can **brand** its online presence.

## 1.2 PROPOSED SYSTEM

- This Application is a software system developed to provide an interface between the Farmers and the sellers/distributors.
- This app helps in avoiding intermediaries which is favorable for farmers.
- The current application can be accessed by two types of users (farmers and sellers). As farmer is illiterate, he accesses this application with the help of another trusted party.
- Information about schemes given by government for farmers can be uploaded in this application.

### 1.2.1) ADVANTAGES

- Can access everything under one roof**.**
- It provides an **additional channel** to market and sell your products
- the platform offers **reduced marketing costs** compared to other sales channels
- It allows new **opportunities for overseas sales** - there has been noted online marketplace growth in the categories of homeware, pets, fashion and sporting goods
- They are popular with customers and offer a **convenient way to compare prices and products** from a single source
- Being part of an established online marketplace provides a **level of trust between farmer and the buyer**
- Impartial reviews of farmers products and service may give **new customers the confidence to buy**
- They offer opportunities to establish **new trading partnerships with traders and suppliers**, either within your supply chain or across supply chains
- They provide **greater transparency** - availability, prices and stock levels are accessible in an open environment.
- It is possible to **operate a round-the-clock** - time constraints and problems with varying international trading hours are removed

### 1.3 APPLICATIONS

- This technology enables access to databases and information  to carry out data entry in app based applications.
- For instance, the farmers can access information about dealers, and the distributors can view the farmers details for trading.

## 2. SOFTWARE REQUIREMENTS

**Software Requirements:**

## 2.1ANDROID STUDIO

Android Studio is the official integrated 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 primary IDE for native Android application development

## Features:

The following features are provided in the current stable version

Gradle-based build support
Android-specific refactoring and quick fixes
Lint tools to catch performance, usability, version compatibility and other problems
ProGuard integration and app-signing capabilities
Template-based wizards to create common Android designs and components
A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
Support for building Android Wear apps
Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
Android Virtual Device (Emulator) to run and debug apps in the Android studio.

## 2.2SQLITE:

SQLite is a relational database management system contained in a C programming library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

## Features:

SQLite implements most of the SQL-92 standard for SQL but it lacks some features. For example, it partially provides triggers, and it cannot write to views (however it provides INSTEAD OF triggers that provide this functionality). While it provides complex queries, it still has limited ALTER TABLE function, as it cannot modify or delete columns.

SQLite uses an unusual type system for an SQL-compatible DBMS; instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is dynamically typed. Moreover, it is weakly typed in some of the same ways that Perl is: one can insert a string into an integer column (although SQLite will try to convert the string to an integer first, if the column's preferred type is integer). This adds flexibility to columns, especially when bound to a dynamically typed scripting language. However, the technique is not portable to other SQL products. A common criticism is that SQLite's type system lacks the data integrity mechanism provided by statically typed columns in other products. The SQLite web site describes a "strict affinity" mode, but this feature has not yet been added. However, it can be implemented with constraints like CHECK. Tables normally include a hidden row id index column which gives faster access. If a database includes an Integer Primary Key column SQLite will typically optimize it by treating it as an alias for row id, causing the contents to be stored as a strictly typed 64-bit signed integer and changing its behaviour to be somewhat like an auto-incrementing column. Future versions of SQLite may include a command to introspect whether a column has behaviour like that of row id to differentiate these columns from weakly-typed, non-autoincrementing Integer Primary Keys.

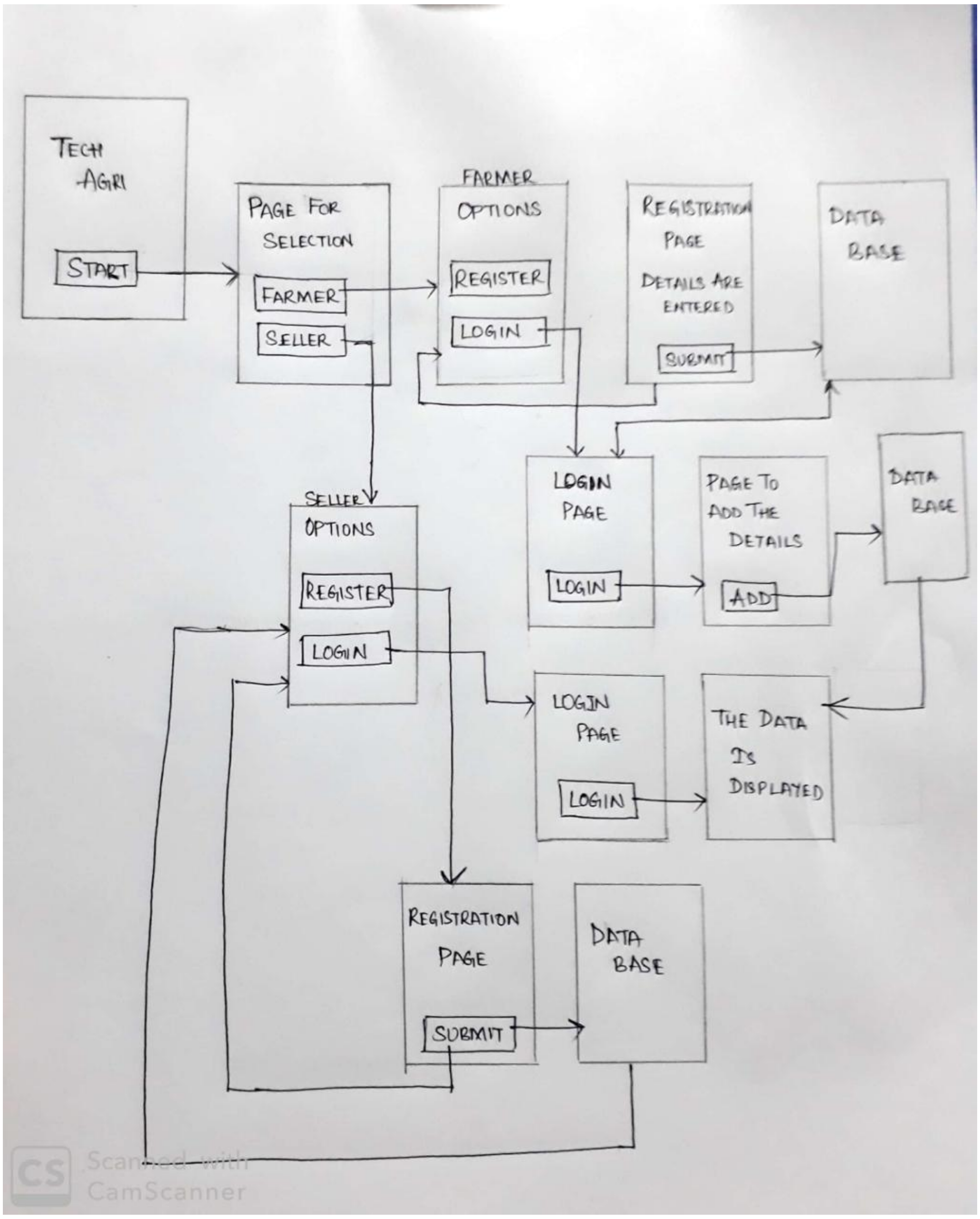
SQLite with full Unicode function is optional.

Several computer processes or threads may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an error code (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when write-ahead logging (WAL) is turned on enabling concurrent reads and writes.

SQLite version 3.7.4 first saw the addition of the FTS4 (full text search) module, which features enhancements over the older FTS3 module. FTS4 allows users to perform full text searches on documents similar to how search engines search webpages. Version 3.8.2 added support for creating tables without row id, which may provide space and performance improvements. Common table expressions support was added to SQLite in version 3.8.3.

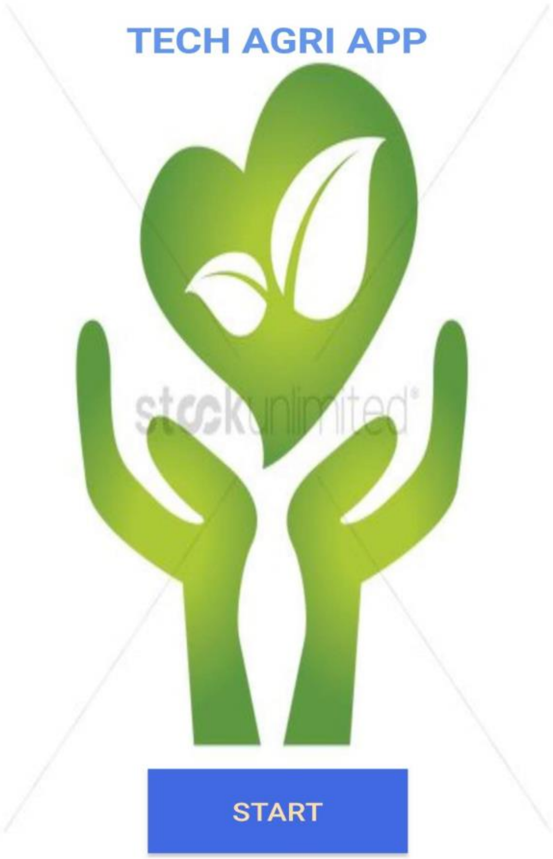
In 2015, with the json1 extension and new subtype interfaces, SQLite version 3.9 introduced JSON content managing**.**

## 3.DESIGN

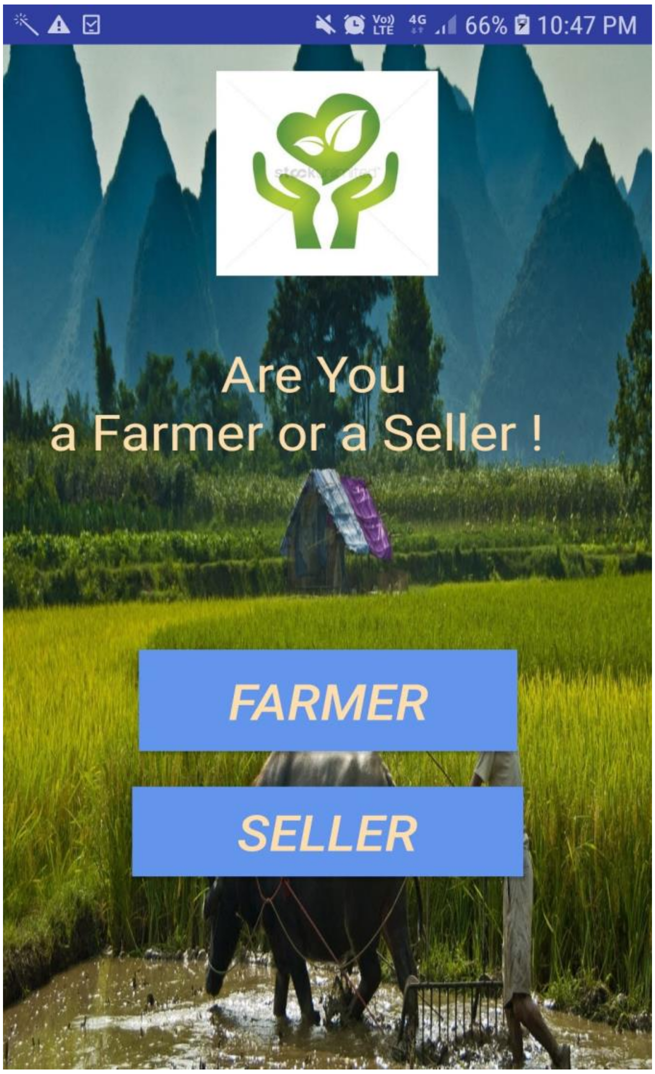

TECH AGRI
START
PAGE FOR SELECTION
FARMER
SELLER
FARMER OPTIONS
REGISTER
LOGIN
REGISTRATION PAGE
DETAILS ARE ENTERED
SUBMIT
DATA BASE
LOGIN PAGE
LOGIN
PAGE TO ADD THE DETAILS
ADD
DATA BASE
SELLER OPTIONS
REGISTER
LOGIN
LOGIN PAGE
LOGIN
THE DATA IS DISPLAYED
REGISTRATION PAGE
SUBMIT
DATA BASE

# 4. RESULT

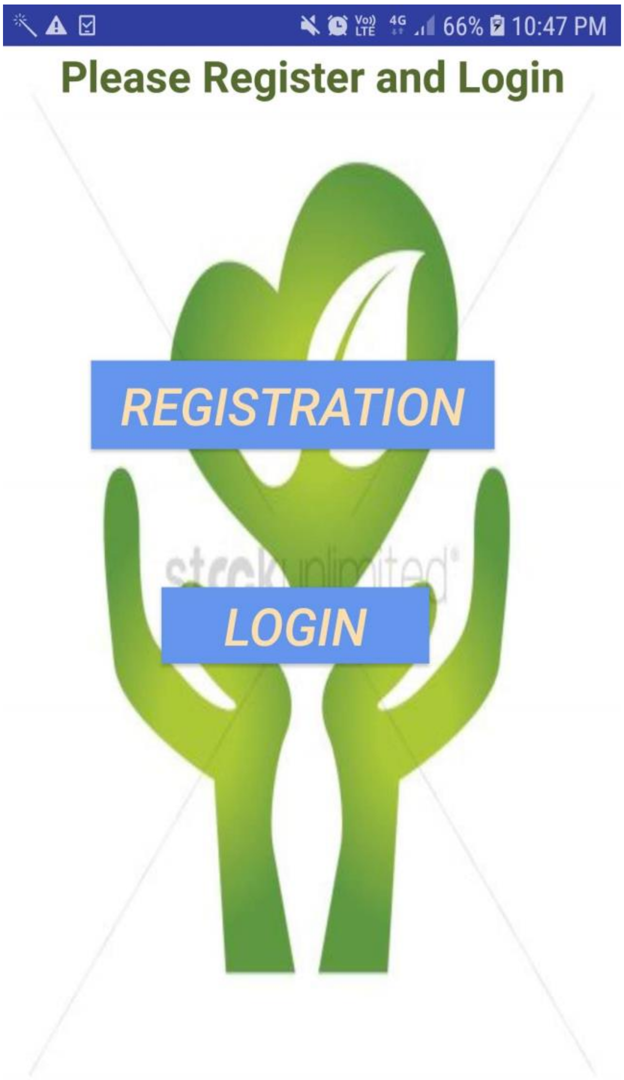
**HOME PAGE**



## Selection Page

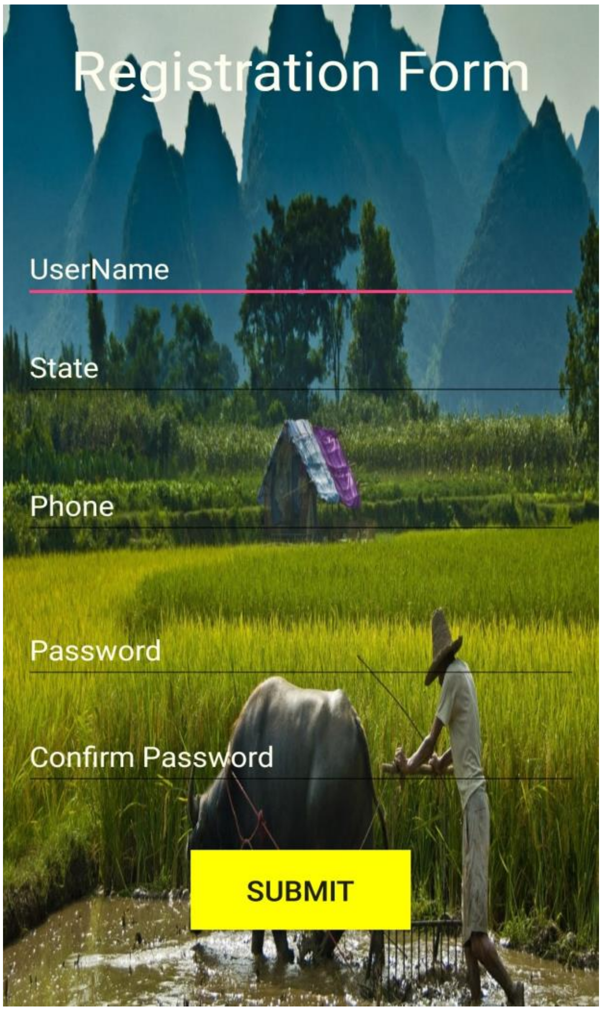


Are You
a Farmer or a Seller !

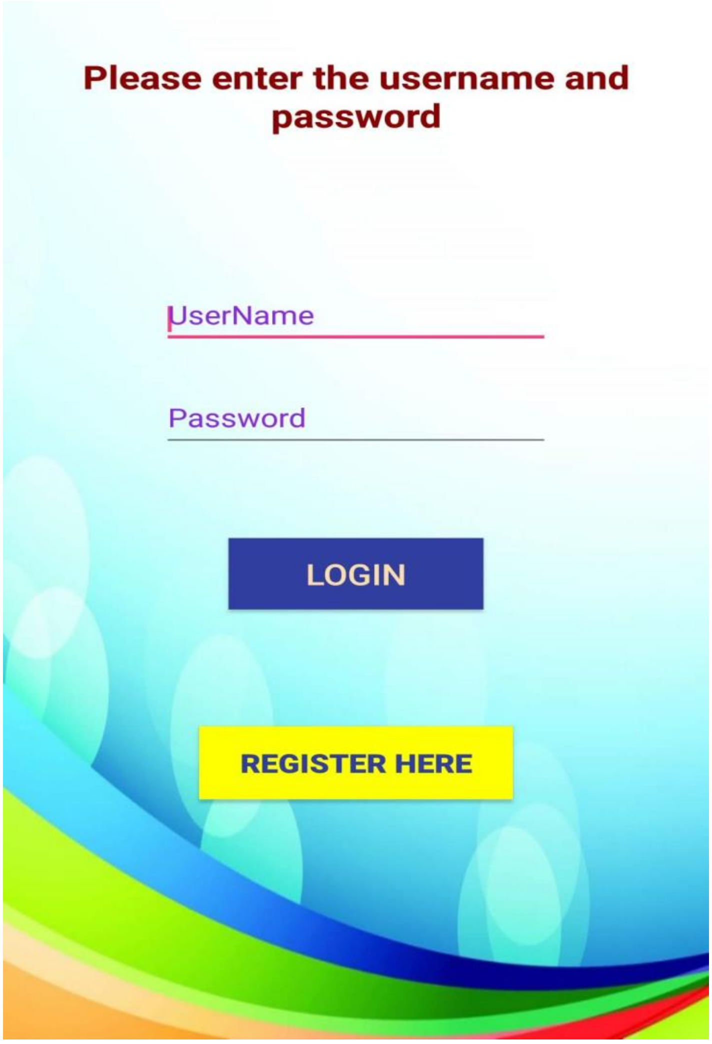


Please Register and Login

Registration Page

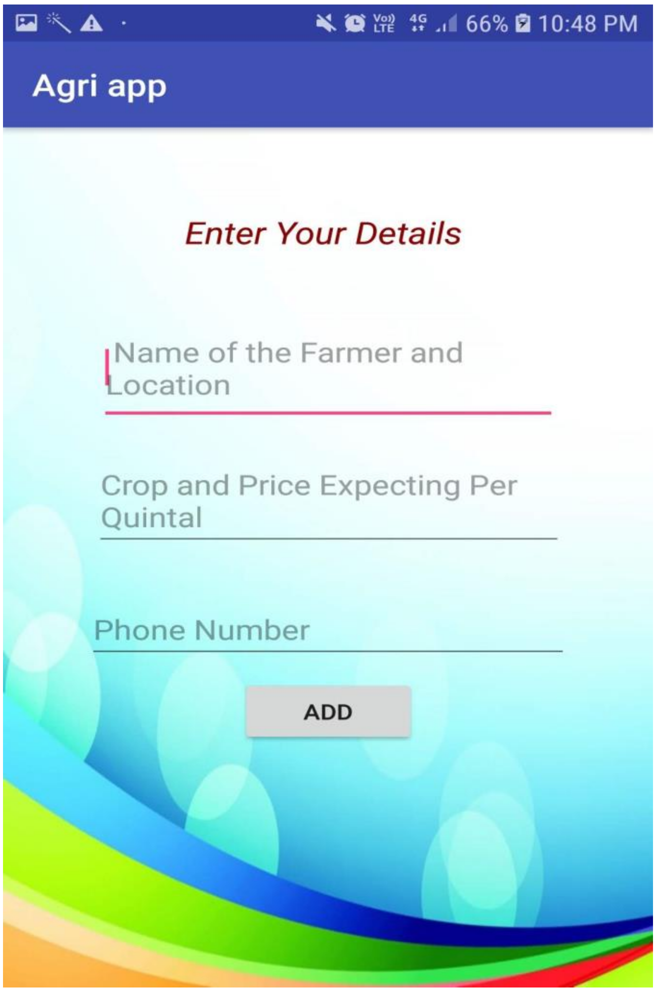


**Farmer’s Login Page**



Please enter the username and password

**Page to add details**



Agri app

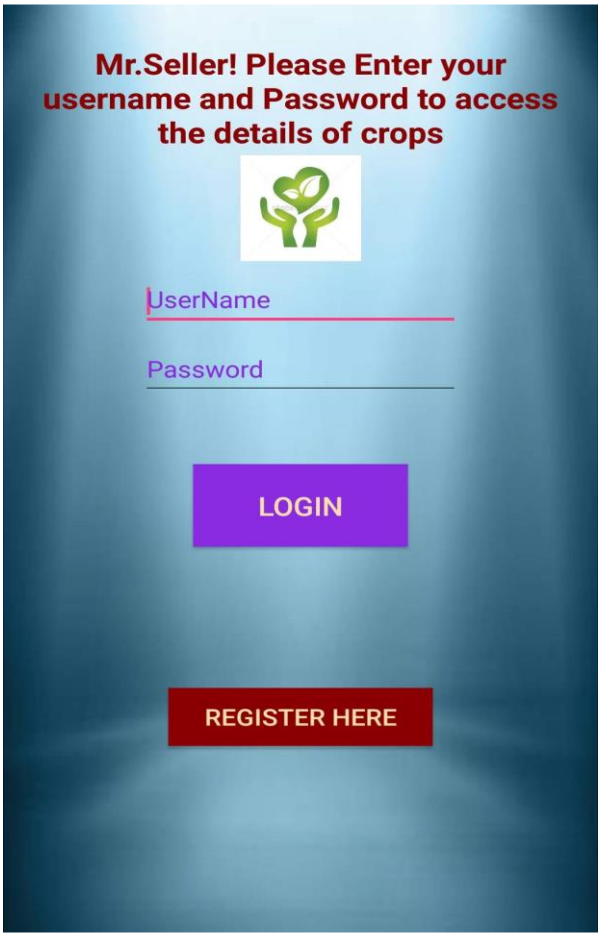
Enter Your Details

Name of the Farmer and Location

Crop and Price Expecting Per Quintal

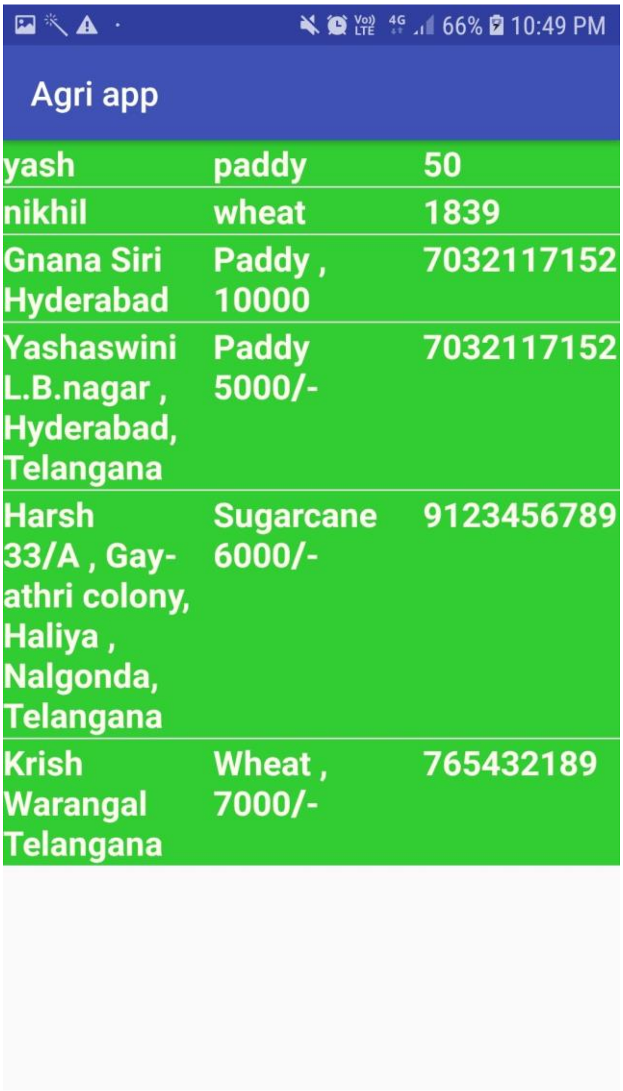
Phone Number

**Seller’s Login Page**



Mr.Seller! Please Enter your username and Password to access the details of crops

## Display page



| | | |
|---|---|---|
| yash | paddy | 50 |
| nikhil | wheat | 1839 |
| Gnana Siri<br>Hyderabad | Paddy ,<br>10000 | 7032117152 |
| Yashaswini<br>L.B.nagar ,<br>Hyderabad,<br>Telangana | Paddy<br>5000/- | 7032117152 |
| Harsh<br>33/A , Gay-athri colony,<br>Haliya ,<br>Nalgonda,<br>Telangana | Sugarcane<br>6000/- | 9123456789 |
| Krish<br>Warangal<br>Telangana | Wheat ,<br>7000/- | 765432189 |

## 8) FUTURE SCOPE

App can be designed such that the farmers can choose the distributors by there profile.

Can be linked up with the Government, such that it can serve as a platform to view the list of farmers in India.

## 9) CONCLUSION

Intermediaries can be avoided by this platform.

Farmers can sell crops on the prices that they fix.

Price fixation is only in the hands of farmers.

The farmer is benefited by this.

# APPENDICES

## A-CODE

Login page .xml code

```
<?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=".Login">



    <EditText
      android:id="@+id/editText"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="110dp"
      android:ems="10"
      android:hint="Email"
      android:inputType="textPersonName"
      />

    <EditText
      android:id="@+id/editText2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="199dp"
      android:ems="10"
      android:hint="Password"
      android:inputType="textPassword" />

    <Button
      android:id="@+id/login"
      android:layout_width="120dp"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_centerHorizontal="true"
      android:layout_marginBottom="197dp"
      android:text="LOGIN" />

    <Button
      android:id="@+id/registerhere"
      android:layout_width="192dp"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_marginBottom="67dp"
      android:text="REGISTER HERE" />
  </RelativeLayout>
```

Login page .java code

```
Packagecom.example.yashaswinikk.newitdept;

importandroid.database.Cursor;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import android.database.sqlite.SQLiteDatabase;


public class Login extends AppCompatActivity {
    EditText e1,e2;
    Button b1,b2;
    public DatabaseHelper db = new DatabaseHelper( this );


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate( savedInstanceState );
        setContentView( R.layout.activity_login );
        e1 = (EditText)findViewById( R.id.editText );
        b2=(Button)findViewById( R.id.registerhere ) ;
        b2.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent( Login.this,MainActivity.class );
                startActivity( intent );
            }
        } );
        e2 = (EditText)findViewById( R.id.editText2 );
        b1 = (Button)findViewById( R.id.login );
        b1.setOnClickListener( new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String email=e1.getText().toString();
                String password=e2.getText().toString();
                Boolean Chkemailpass=db.emailpassword( email,password );
                if(Chkemailpass==true)
                    Toast.makeText( getApplicationContext(),"successfull login" ,Toast.LENGTH_SHORT).show();
                else
                    Toast.makeText(getApplicationContext(),"wrong email or password",Toast.LENGTH_SHORT).show();
            }
        } );
    }
    public Boolean emailpassword(String email, String password){

        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery( "select * from user where email=? and password=?",new String[]{email,password}
);
        if (cursor.getCount() > 0) return true;
        else return false;
    }


    public SQLiteDatabase getReadableDatabase() {

        return null;
    }
```

}

## Registration page .xml code

```
<?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=".RegisterActivity"
    android:background="@drawable/img19">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Register"
      android:textSize="35sp"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="45dp"
      android:textColor="#145A32"/>

    <EditText
      android:id="@+id/name"
      android:layout_width="336dp"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="146dp"
      android:hint="UserName"
      android:inputType="textPersonName"
      android:textColor="@color/colorPrimaryDark"
      android:textColorHint="@color/colorPrimaryDark" />

    <EditText
      android:id="@+id/email"
      android:layout_width="336dp"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="214dp"
      android:hint="Email"
      android:inputType="textEmailAddress"
      android:textColor="@color/colorPrimaryDark"
      android:textColorHint="@color/colorPrimaryDark" />

    <EditText
      android:id="@+id/phone"
      android:layout_width="336dp"
      android:layout_height="wrap_content"
      android:layout_centerInParent="true"
      android:hint="Phone"
```

```
        android:inputType="phone"
        android:textColor="@color/colorPrimaryDark"
        android:textColorHint="@color/colorPrimaryDark" />

    <EditText
        android:id="@+id/password"
        android:layout_width="336dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="213dp"
        android:hint="Password"
        android:inputType="textPassword|numberPassword"
        android:textColor="@color/colorPrimaryDark"
        android:textColorHint="@color/colorPrimaryDark" />

    <EditText
        android:id="@+id/cpassword"
        android:layout_width="336dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="132dp"
        android:hint="Confirm Password"
        android:inputType="textPassword|numberPassword"
        android:textColor="@color/colorPrimaryDark"
        android:textColorHint="@color/colorPrimaryDark" />

    <Button
        android:id="@+id/regbtn"
        android:layout_width="133dp"
        android:layout_height="51dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="49dp"
        android:background="@color/Yellow"
        android:text="SUBMIT"
        android:textSize="18dp"/>



</RelativeLayout>
```

## Registration page .java code

```
package com.example.yashaswinikk.newitdept;

import android.content.Intent;
import android.database.Cursor;
import android.support.v7.app.AppCompatActivity;
        import android.os.Bundle;
        import android.util.Patterns;
        import android.view.View;
        import android.view.WindowManager;
        import android.widget.Button;
        import android.widget.EditText;
        import android.widget.Toast;
import android.database.sqlite.SQLiteDatabase;




public class RegisterActivity extends AppCompatActivity {
    private EditText et_name, et_email, et_phone, et_password, et_cpassword;
    private String name, email, phone, password, cpassword;
    Button regbtn;
    DatabaseHelper db;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
        try {
            this.getSupportActionBar().hide();

        } catch (Exception e) {
            Toast.makeText(getApplicationContext(), "" + e, Toast.LENGTH_LONG).show();
        }
        db = new DatabaseHelper(this);
        et_name = (EditText) findViewById(R.id.name);
        et_email = (EditText) findViewById(R.id.email);
        et_phone = (EditText) findViewById(R.id.phone);
        et_password = (EditText) findViewById(R.id.password);
        et_cpassword = (EditText) findViewById(R.id.cpassword);
        regbtn = (Button) findViewById(R.id.regbtn);


        regbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                String s4 = et_name.getText().toString();
                String s5 = et_phone.getText().toString();
                String s1 = et_email.getText().toString();
                String s2 = et_password.getText().toString();
                String s3 = et_cpassword.getText().toString();
                if (s1.equals("") || s2.equals("") || s3.equals("") || s4.equals("") || s5.equals("")) {

                    Toast.makeText(getApplicationContext(), "Fields are empty", Toast.LENGTH_SHORT).show();
                    register();
                } else {
                    if (s2.equals(s3)) {
                        Boolean chkusername = db.chkemail(s4);

                        if (chkusername == true) {
                            Boolean insert = db.insert(s4, s2);
                            if (insert) {

                                Toast.makeText(getApplicationContext(), "registered successfully", Toast.LENGTH_SHORT).show();
                                Intent intent = new Intent(RegisterActivity.this, activity2.class);
                                startActivity(intent);

                            }
                        } else {
                            Toast.makeText(getApplicationContext(), "name already exists ", Toast.LENGTH_SHORT).show();

                        }

                    } else {
                        Toast.makeText(getApplicationContext(), "password does not match", Toast.LENGTH_SHORT).show();
                    }
                }


            }

        });
    }

    @Override
    public void onBackPressed() {
        Intent intent = new Intent(this, activity2.class);
        startActivity(intent);
        return;
    }

    public void register() {
        initialze();
        if (!validate()) {
            Toast.makeText(this, "Signup has Failed", Toast.LENGTH_SHORT).show();
        } else {
            onSignupSuccess();
        }
    }

    public void onSignupSuccess() {
        Intent intent = new Intent(this, activity2.class);
        startActivity(intent);

    }

    public boolean validate() {
        boolean valid = true;
        if (name.isEmpty() || name.length() > 32) {
            et_name.setError("please enter valid name");
            valid = false;
        }
        if (email.isEmpty() || !Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            et_email.setError("please enter valid email address");
            valid = false;
        }

        if (phone.isEmpty()) {
            et_phone.setError("please enter phone number");
            valid = false;
        }
        if (password.isEmpty()) {
            et_password.setError("please enter password");
            valid = false;
        }
        if (cpassword.isEmpty()) {
            et_cpassword.setError("please enter password");
            valid = false;
        }
        return valid;
    }

    public void initialze() {
        name = et_name.getText().toString().trim();
        email = et_email.getText().toString().trim();
        phone = et_phone.getText().toString().trim();
        password = et_password.getText().toString().trim();
        cpassword = et_cpassword.getText().toString().trim();

    }

    public boolean chkusername(String email) {
        SQLiteDatabase db;
        db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("Select * from user where email=?", new String[]{email});
        if (cursor.getCount() > 0) return false;
        else return true;

    }

    private SQLiteDatabase getReadableDatabase() {

        return null;
    }


}
```

Database Helper code

```
package com.example.yashaswinikk.newitdept;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;




public class DatabaseHelper extends SQLiteOpenHelper {
    public DatabaseHelper(Context context) {
        super( context, "Login.db", null, 1 );
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL( "Create table user (email text primary key ,password text)" );

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL( "drop table if exists user" );

    }

    public boolean insert(String email, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put( "email", email );
        contentValues.put( "password", password );
        long ins = db.insert( "user", null, contentValues );
        if (ins == -1) return false;
        else return true;


    }

    public boolean chkemail(String email) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery( "Select * from user where email=?", new String[]{email} );
        if (cursor.getCount() > 0) return false;
        else return true;

    }

    public Boolean emailpassword(String email, String password){

        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery( "select * from user where email=? and password=?",new String[]{email,password}
);
        if (cursor.getCount() > 0) return true;
        else return false;
    }

}
```

## Activity_far.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
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/img1"
    tools:context=".far">


    <EditText
        android:id="@+id/etFirstName"
        android:layout_width="255dp"
        android:layout_height="74dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="120dp"
        android:hint=" Name of the Farmer and Location"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="MissingConstraints" />

    <EditText
        android:id="@+id/etLastName"
        android:layout_width="261dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/etFirstName"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:hint="Crop and Price Expecting Per Quintal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.504"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/etFirstName"
        tools:ignore="MissingConstraints" />

    <EditText
        android:id="@+id/etFavFood"
        android:layout_width="268dp"
        android:layout_height="wrap_content"
        android:layout_below="@+id/etLastName"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="28dp"
        android:hint="Phone Number"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/etLastName"
        tools:ignore="MissingConstraints" />

    <Button
        android:id="@+id/btnAdd"
        android:layout_width="99dp"
        android:layout_height="45dp"

        android:layout_alignParentTop="true"

        android:layout_marginBottom="188dp"
        android:layout_marginTop="8dp"
        android:text="Add"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/etFavFood"
        app:layout_constraintVertical_bias="0.0"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textView"

        android:layout_width="297dp"
        android:layout_height="39dp"
        android:layout_marginBottom="8dp"
        android:layout_marginTop="36dp"
        android:textColor="@color/DarkRed"
android:textSize="20sp"
        android:text="Enter Your Details "
        android:textAlignment="center"
        android:textStyle="italic"
        app:layout_constraintBottom_toTopOf="@+id/etFirstName"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.505"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />


</android.support.constraint.ConstraintLayout>
```

## Far.java

```
package com.example.yashaswinikk.agriapp;

import android.app.ActionBar;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;




    public class far extends AppCompatActivity {

        EditText etFirstName,etLastName,etFavFood;
        Button btnAdd,btnView;
        DatabaseHelper1 myDB;


        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_far);
            etFavFood = (EditText) findViewById(R.id.etFavFood);
            etFirstName = (EditText) findViewById(R.id.etFirstName);
            etLastName = (EditText) findViewById(R.id.etLastName);
            btnAdd = (Button) findViewById(R.id.btnAdd);

            myDB = new DatabaseHelper1(this);



            btnAdd.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    String fName = etFirstName.getText().toString();
                    String lName = etLastName.getText().toString();
                    String fFood = etFavFood.getText().toString();
                    if(fName.length() != 0 && lName.length() != 0 && fFood.length()
!= 0){

                        AddData(fName,lName, fFood);
                        etFavFood.setText("");
                        etLastName.setText("");
                        etFirstName.setText("");
                    }else{
                        Toast.makeText(far.this,"You must put something in the text
field!",Toast.LENGTH_LONG).show();
                    }
                }
            });

        }

        public void AddData(String firstName,String lastName, String favFood ){
            boolean insertData = myDB.addData(firstName,lastName,favFood);

            if(insertData==true){
                Toast.makeText(far.this,"Successfully Entered
Data!",Toast.LENGTH_LONG).show();
            }else{
                Toast.makeText(far.this,"Something went wrong
:(.",Toast.LENGTH_LONG).show();
            }
        }
    }
```