

Android Voting App using Fingerprint Authentication

ABSTRACT

The voting system is a crucial part of democracy within which citizens get to choose their leaders through which they voice their opinions in the country. Voting systems have evolved massively from hand-counting votes to newer systems that include cards, biometric authentication-requiring machines, and levers. The conventional system is a tedious process due to which the ever busy citizens are unable to vote.

Fabrication and alteration of received data is also another very crucial drawback of these systems. Corrupt candidates also set up /bribe groups that either try to vote multiple times or force/ manipulate others to change their votes. The people in-charge of the entire system can also be untrustworthy and malicious and fidget with votes which gives out inaccurate results.

The underlying reason behind an online and computer based system is an attempt to overcome these shortcomings of the older system. The tedious task of paperwork and time along with unavoidable human made errors makes voting and election an unnecessarily extensive process. The main aim of this project is to create a system that can help motivate voters to remotely participate to help acquire a more rounded result.

INTRODUCTION

Due to the advanced technology, olden techniques of voting can be transformed to newer innovations. The computer based system provides a commodious, practical, and systematic way of voting which assures a secure and safe experience and result. The existing voting system requires voters to physically be available at a voting location from where they can cast their votes. Countries spend millions in an attempt to keep the event peaceful and efficient.

For centuries these voting booths have contained an EVM which is an Electronic Voting Machine that is a huge investment for the country and due to declining voting participants, it's becoming a failing one.

Some of the different types of Voting systems that have evolved through the centuries are:

(A) Paper-Based Voting: Voters mark their votes on a piece of paper which are then totaled and counted by hand.

(B) Lever Voting Machine: This method uses a levering machine to allow users to vote.

(C) Direct-recording Electronic (DRE) voting systems: This method uses touch screens or keyboards for users to submit their votes.

(D) Punch Card: Voters are given metallic hole-punchers to select their candidate by punching a hole in the ballot.

These olden techniques are faced with various problems such as:

(A) Requires extensive manual labor

(B) Rigid Locations

(C) Prone to human errors

- (D)Time Consuming
- (E)Cost inefficient

However, with recent technological upgrades, the voting process has become more advanced. It has evolved to be a secure online legal system through which any voter can vote from anywhere in the country. In this system, a voter can use his right online with no difficulty. He needs to be registered first for him to vote.

The main scope of this project is as follows:

- (A)Improved and efficient system
- (B)Security of the voters
- (C)Maximum participation

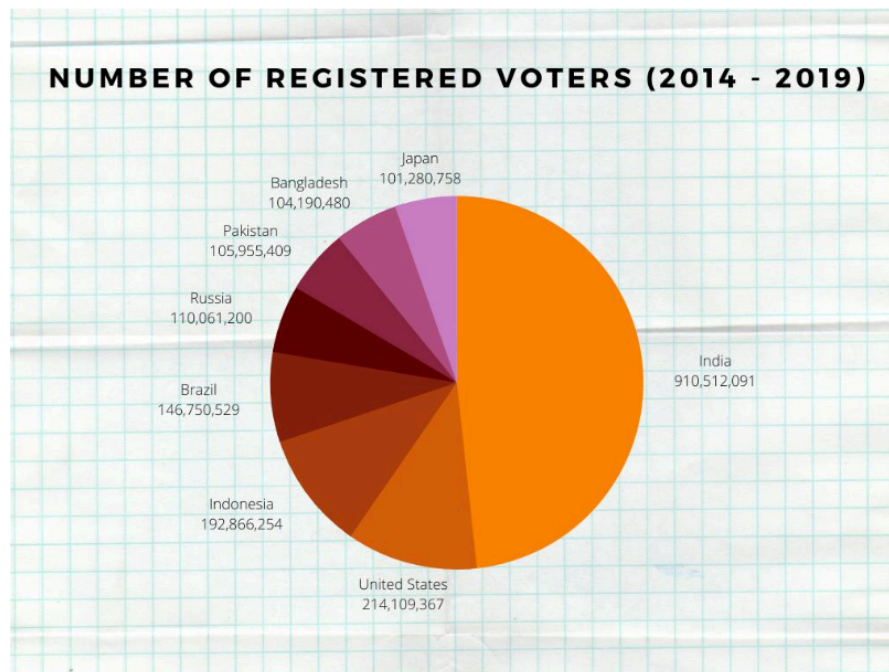
Voting Systems require three crucial and non-negotiable characteristics that make them transparent and comprehensible.

- (A) Anonymity of the voter to ensure their safety and confidence
- (B)Security of the votes and results
- (C)Accuracy of the results

This project ensures that the application follows these characteristics and aims to be an upgrade in the system.

LITERATURE REVIEW

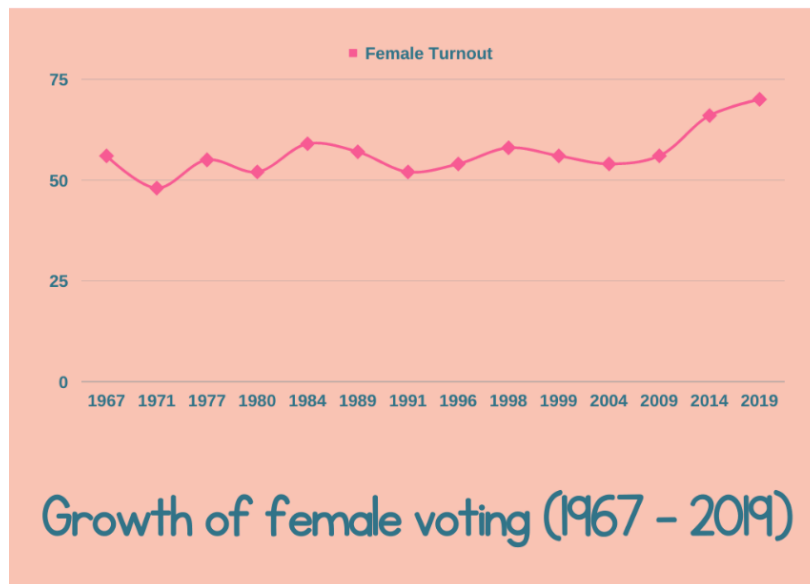
Figure 1. Comparison of numbers of registered voters in different nations [1]



In the above figure, we can see the comparison between different nations from around the globe ranging the countries from Asia and South America, in this graph, we can see that in terms of registered voters Indians are dominating with a total of 910,512,091 million registered voters in the year 2019, the United States of America is at the second place with a total of 21,109,367 million voters in the year 2016. Japan on the other hand a very small country compared to both India and the United States of America has the least amount of registered voters with a total of 101,280,758 million registered voters.

Although both the Indian and Pakistan nations are very political, the numbers of registered voters in India are far greater than in Pakistan and that is due to the comparison in their populations where Pakistan's population stands at 229 million human beings and which makes almost 50 % of Pakistan registered voters but when it comes to India with its massive population of 1.4 billion that only leaves about less than 25% of population unregistered. There are other countries such as Bangladesh and Russia whose populations hold a large amount of registered voters in their nation [2] [3].

Figure 2 Growth of female voting in the year 1967 – 2019 [4]



In figure 2 we can see that female voting has come a long way from when women got their rights to vote which was in 1947 at the time of India's independence but before the early 50s women voting rights were as close as a joke to the government but as time passed women voting rights were forming its roots to become a tall tree, as we can see in the graph that in 1967 the number of female voters was only 56% whereas the male turnout was close to 70%, after that throughout the years there were up and down in both male turnouts and female turnouts but in the majority, the male turnout was at least 25 % higher than the female turnout, it was during the time of 1998 where women started voting more often and from 2004 the charts have been at a constant growth till today. [5]

In 2004 the female turnout was 56 % whereas the male turnout was 65% but by the second phase of 2020 both the male turnout and the female turnout were 70 % which is a tremendous amount of growth from when the constitution was formed

The journey for female voting didn't just start after the independence but it was first started back in 1921 when the women of madras won the first legislature in British India and it was the first time it was won by a considerable majority. This only meant only one thing women were here first recognized as people of the state who have every right to vote as the men of the country do [6].

As mentioned earlier various voting techniques prevailed in earlier times which the following authors discuss in detail

1. The author in [7] talks about the oldest architecture of the paper-based voting system in which voting is done using cards that contain the complete details of the voter. This ensures verifiability but lacked efficiency.
2. Ankit Anand [8] proposes a website-based system that allows remote voting. The system eradicates human error chances but increases the risk of network and cyber attacks.

Various other authors [9] [10] discuss the creation of an android based application that uses either biometric authentication or encryption or both

The author in [11] proposes two systems. The first is an SMS voting system that will contain the certificate number, form number, and candidate number. A predefined format will be conveyed to the audience through which they will be able to cast their vote.

The second method proposed in this research is Internet voting which uses HTML CSS JScript and SWF for the front end, PHP for the middle, and MYSQL as the backend database.

Failure of electronics and network threats turned out to be the biggest drawbacks while risks of ambiguity were completely resolved.

Author Swaminathan and Dinesh proposed a system [12] where the internet is the means for voting. Candidates are given a User PIN and secret key using steganography which is a form of encryption. Fingerprints are taken as keys for encryption. The voter will be required to log in to the system using his/her personal identification number and secret key to cast his vote.

This system has immense security and authentication but is very complex and requires various government guarded information.

Authors in [13] a backend database to store voters and candidate details along with usernames and passwords and collected votes. The model proposes an online system that is basic and user-friendly with three phases: a registration phase, authentication phase, and voting phase. It is a two-way system that used OTP for user identification and password sent to registered emails. This system provides high transparency but is also highly complex.

Authors in [14] & [15] shed immense light on the benefits of the online voting system while also discussing the drawbacks which they believe with the right approach can be completely eradicated.

These projects are what inspired the start behind creating an android based application as everyone from kids to older generations has access and can devices with minimal assistance.

THEORETICAL BACKGROUND

A. ANDROID STUDIO

Android is an operating system for mobile devices that is created by Google and is an upgraded version of the Linux OS.

Android studio is a free unified environment where applications can be created for phones, tablets, watches, tv, etc.

This project uses all the basic android imports but also a few specific and crucial ones.

- i. `import android.database.sqlite.SQLiteDatabase` and `import android.database.sqlite.SQLiteOpenHelper` for SQLite connectivity and
- ii. `import android.hardware.biometrics.BiometricManager` and `import androidx.biometric.BiometricPrompt` for fingerprint usage

B. SQLite

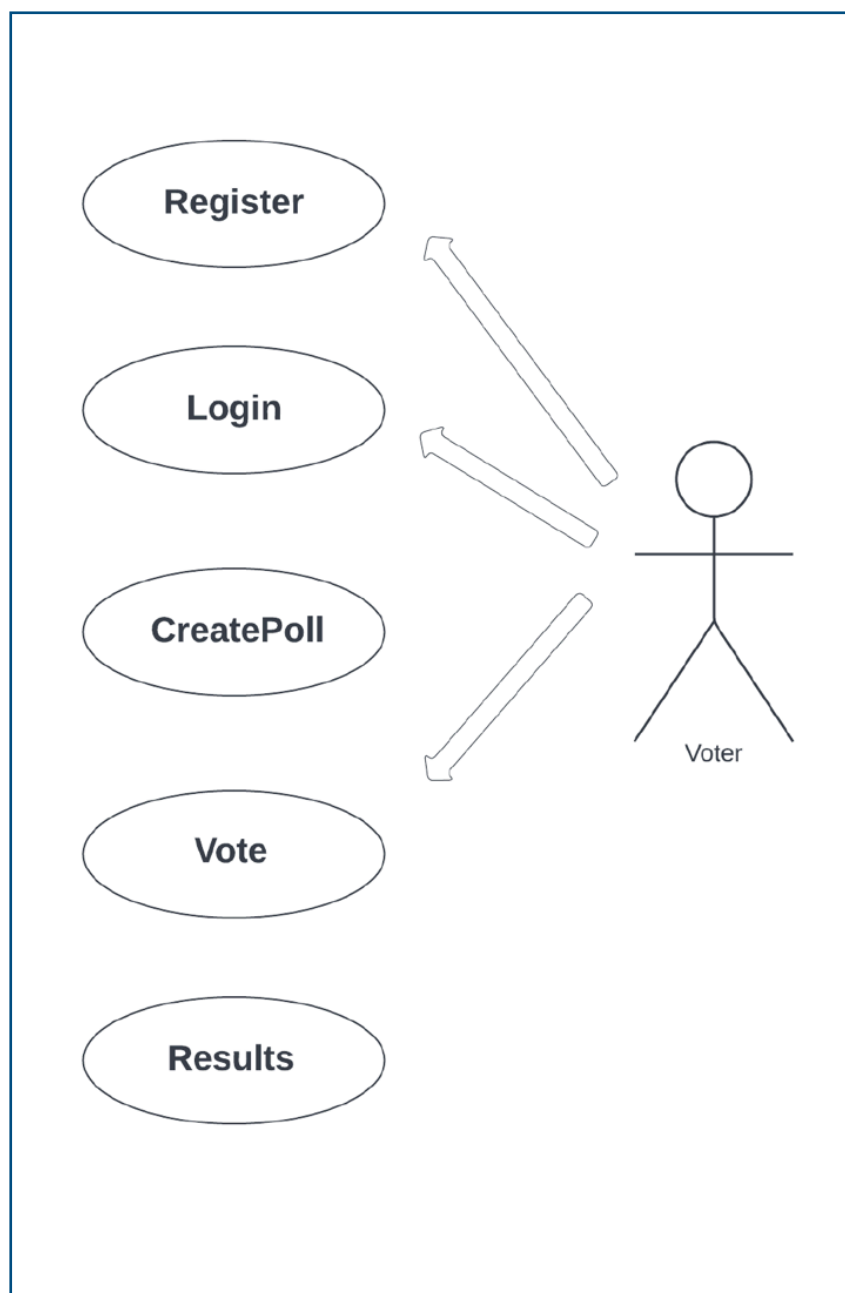
It is a database engine written in C programming language and acts as a library that can be embedded in applications.

This project uses SQLite for the creation of relations and databases that can store voters' data and also the votes casted.

FRAMEWORK

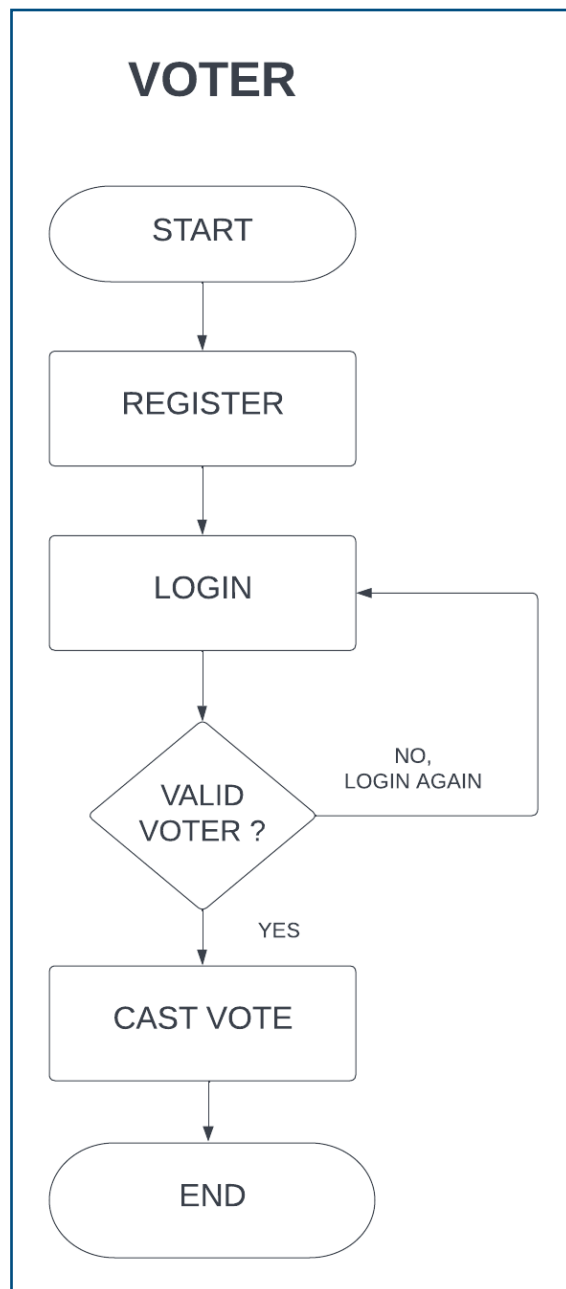
(A)Use Case Diagram

It defines the interactions an entity will have with the system.



(B)Activity Diagram

This shows the sequence of the program from start to finish.



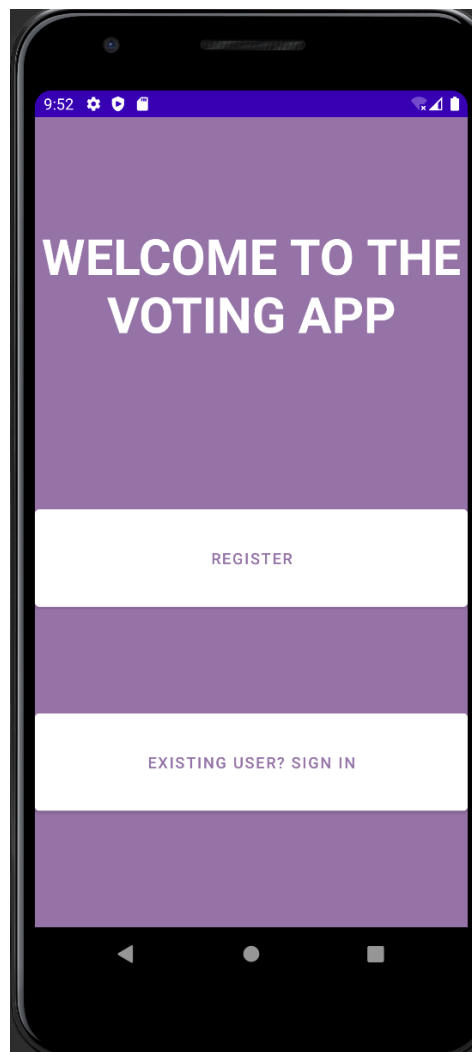
HARDWARE AND SOFTWARE REQUIRED

- Operating System: Android
- Storage: 1GB
- RAM: 512 MB
- Sensors: Fingerprint Scanner
- Wifi Connectivity at all times

WORKING OF THE PROJECT

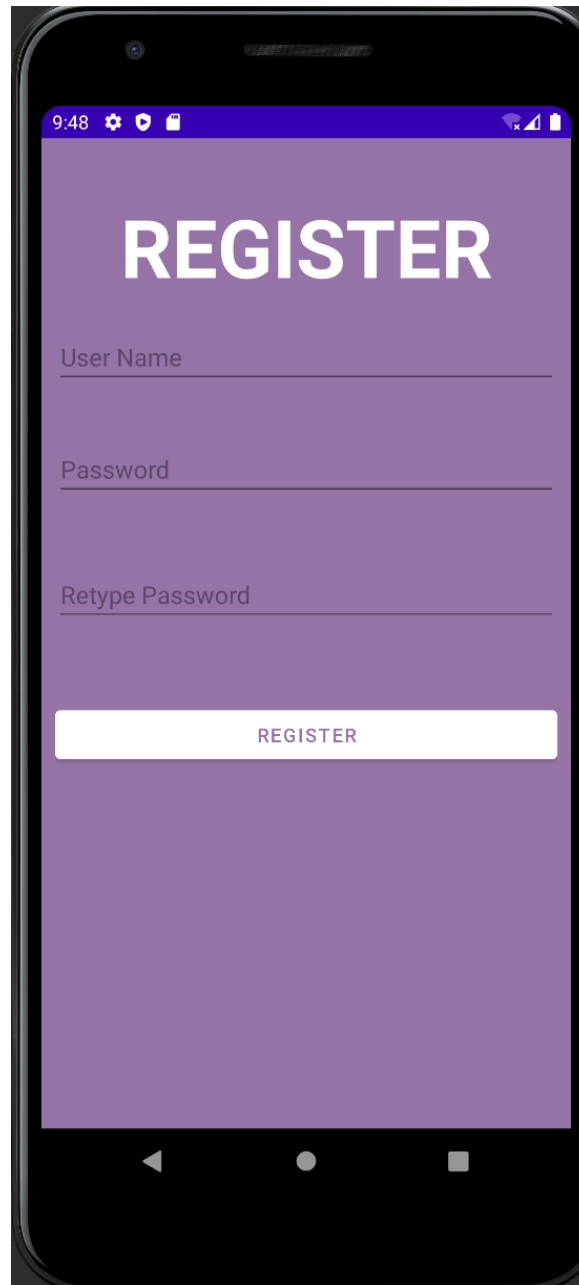
The flow of the project is as follows:

FRAME 1: This is the first page a user sees when the app launches. On this page the user can select to either Register (which is a one time option per device) if they are new or login if they have existing accounts.



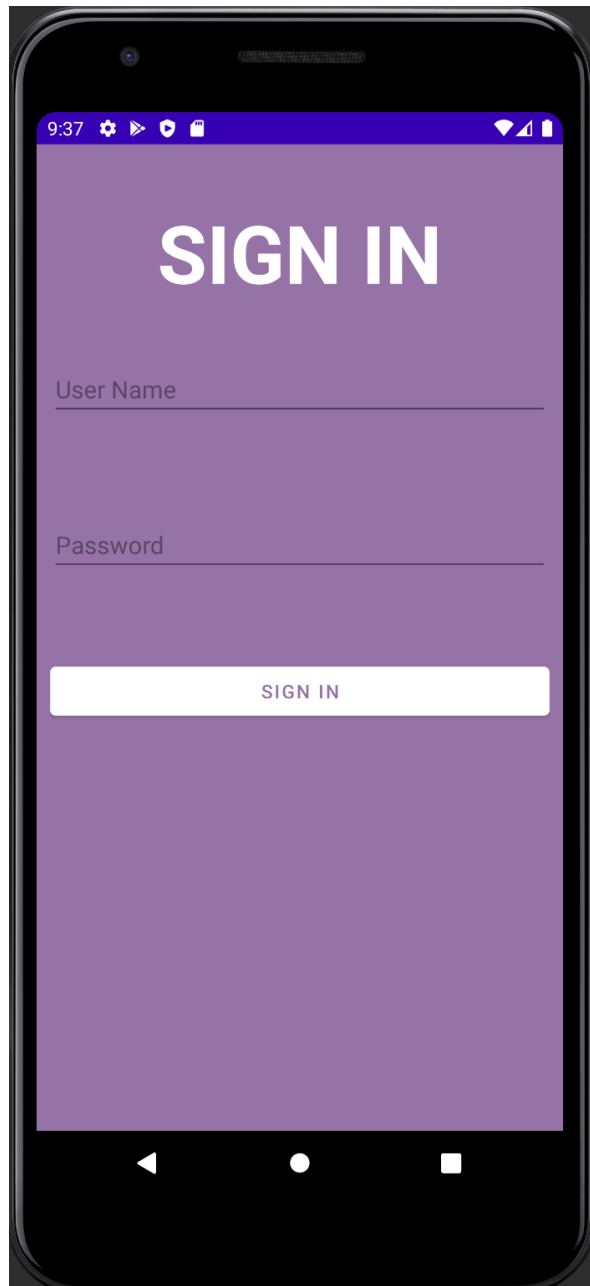
FRAME 2: If the user selects the one time option of Register the following page launches. Here the user is required to fill out 3 fields:

- a. Username which is the name they will also use for signing in
- b. Password which is the authentication required for login
- c. Repassword which is duplication of the password for certainty

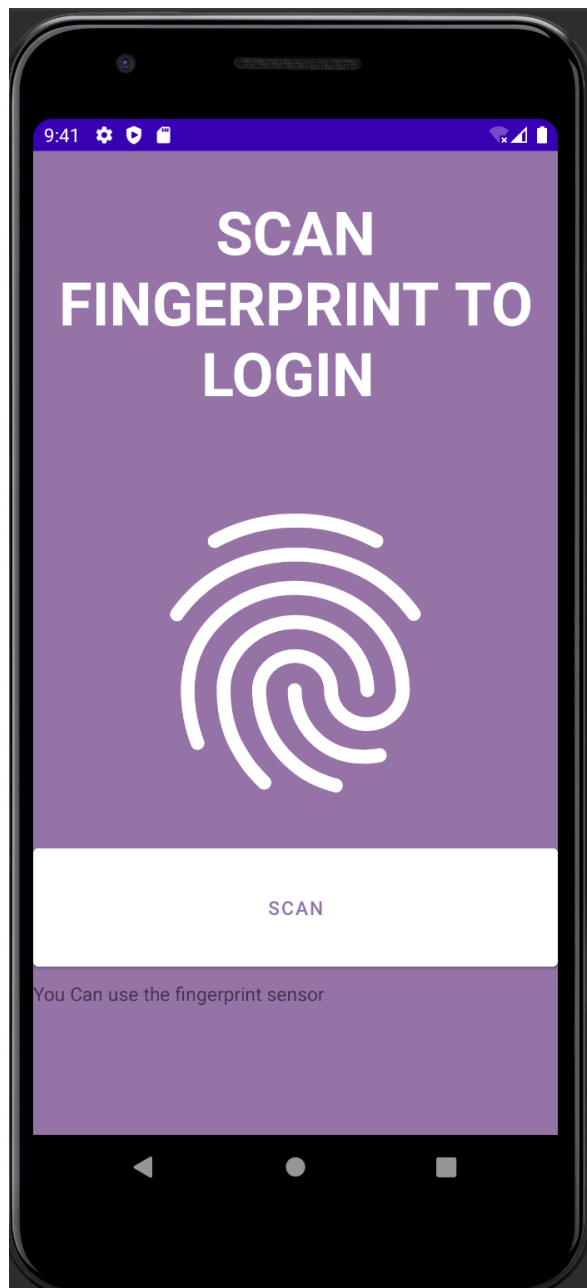


Once the Register button is clicked the user is taken to the Sign in page which will now become the default page anytime the user launches the application. This is done to ensure that one user can only create one account using one device hence following the one man one vote narrative.

FRAME 3: Here the user uses their created username and password credentials to log in into their account.



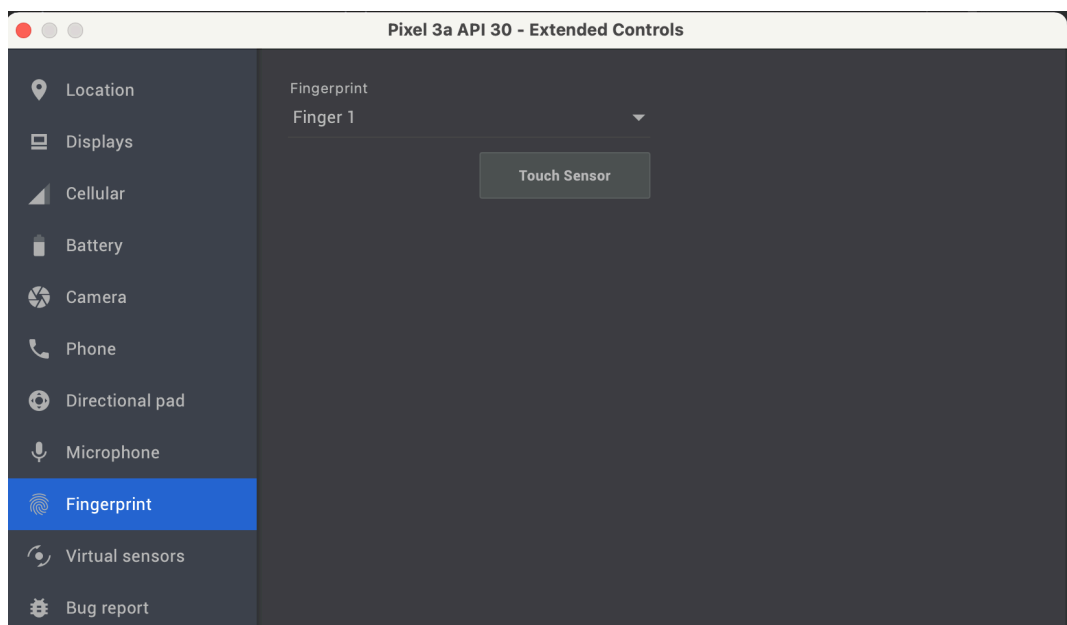
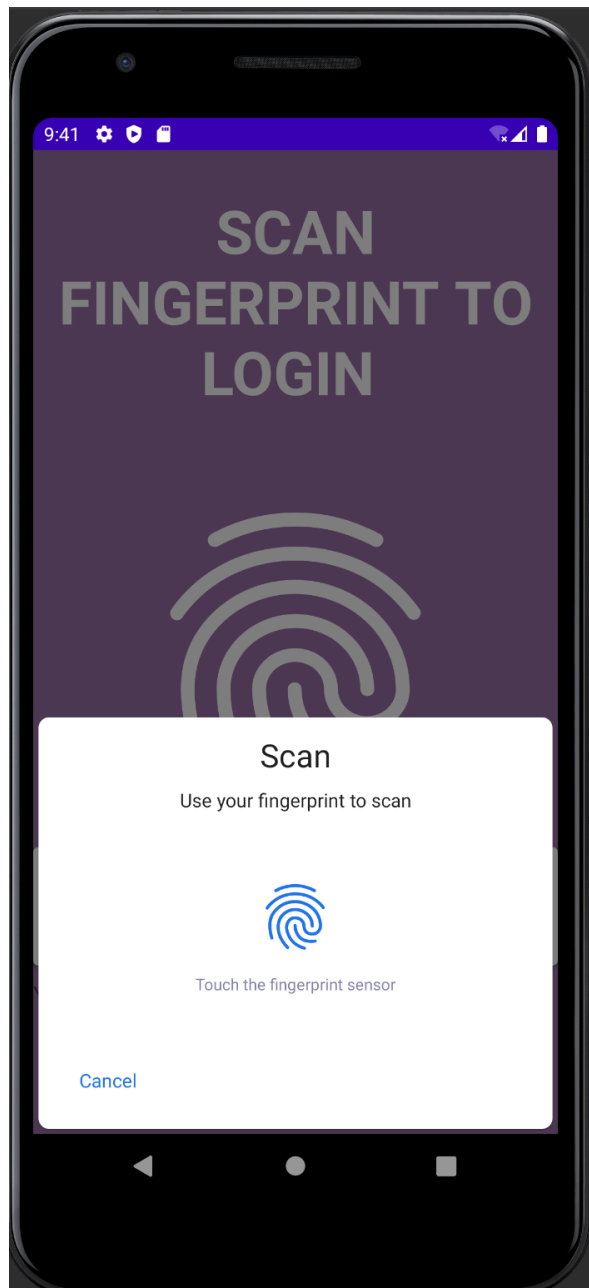
Frame 4: After every sign in the user is required to authenticate themselves by using fingerprint. Hence frame 4 launches each time a user wants to sign in.



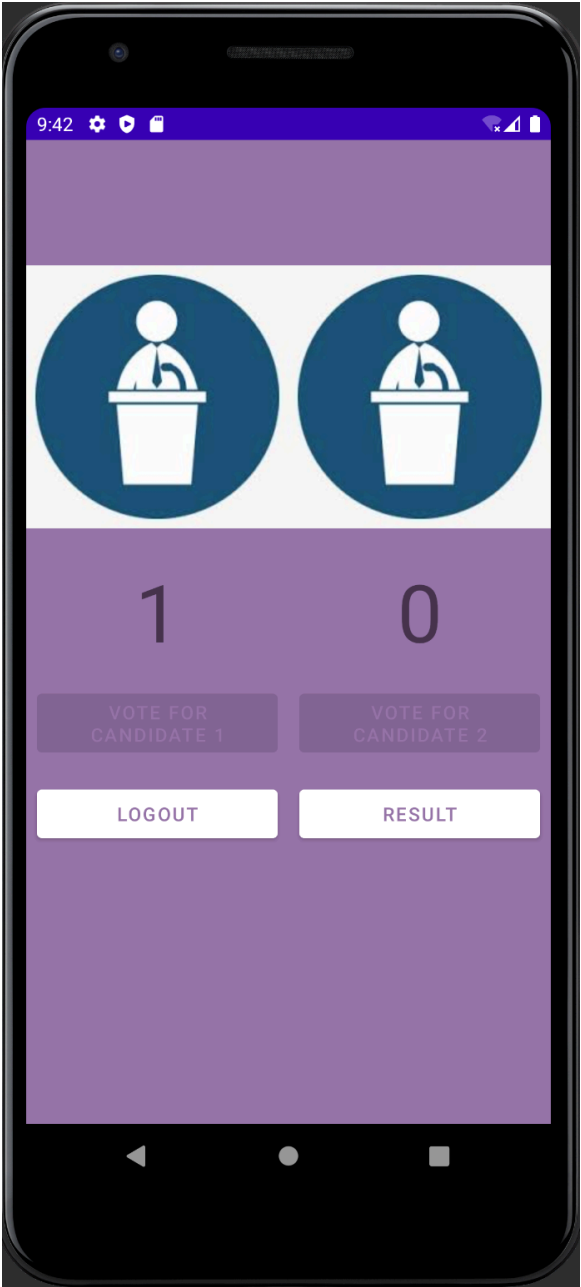
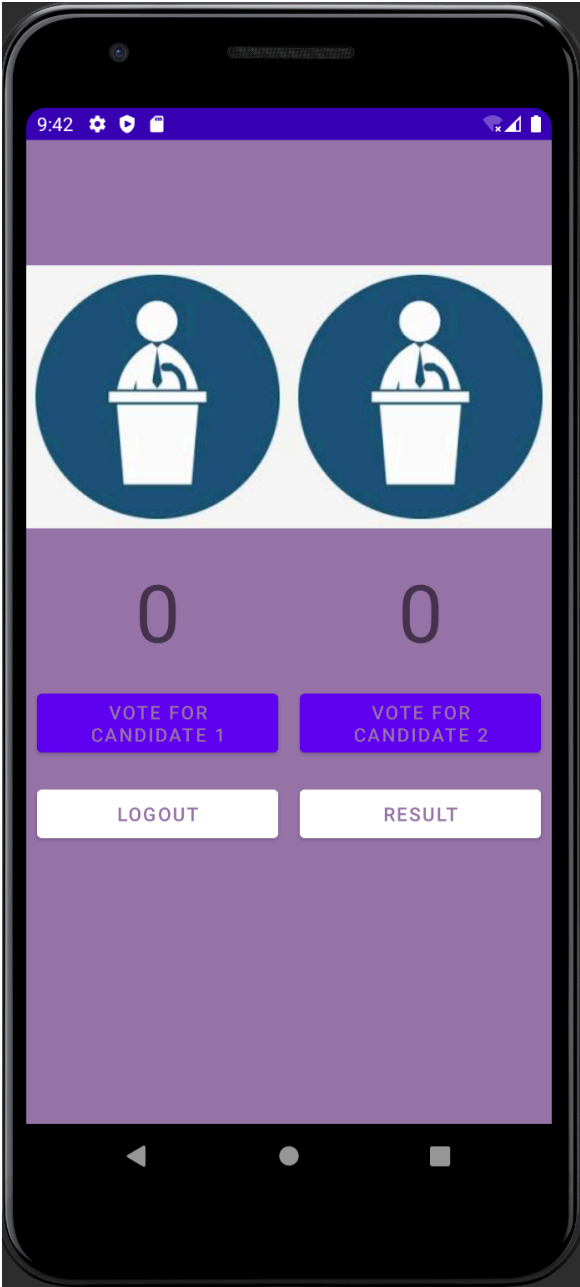
The user is required to click scan which then shows the following pop-up dialog box if following three conditions match:

- a. The device has a fingerprint Scanner
- b. The fingerprint scanner has no hardware issues
- c. There is a saved fingerprint on the device

Frame 5: The user then touches the scanner to store their fingerprint. For this test run a virtual finger has been used to test the system as the application is running on an emulator and not a physical device.

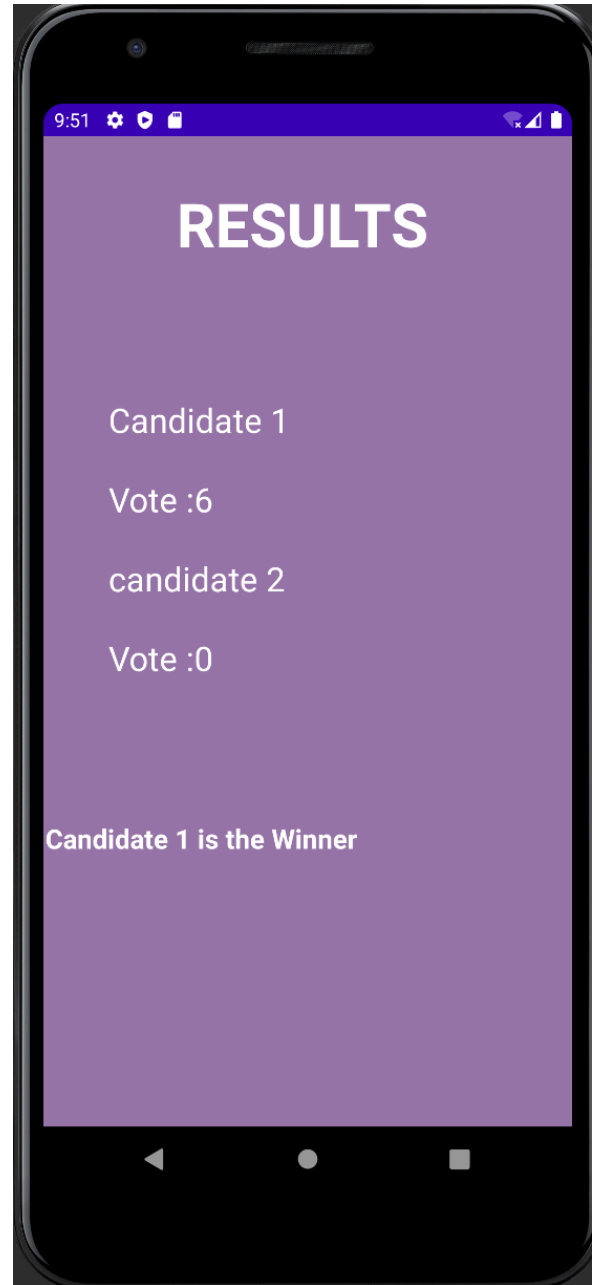
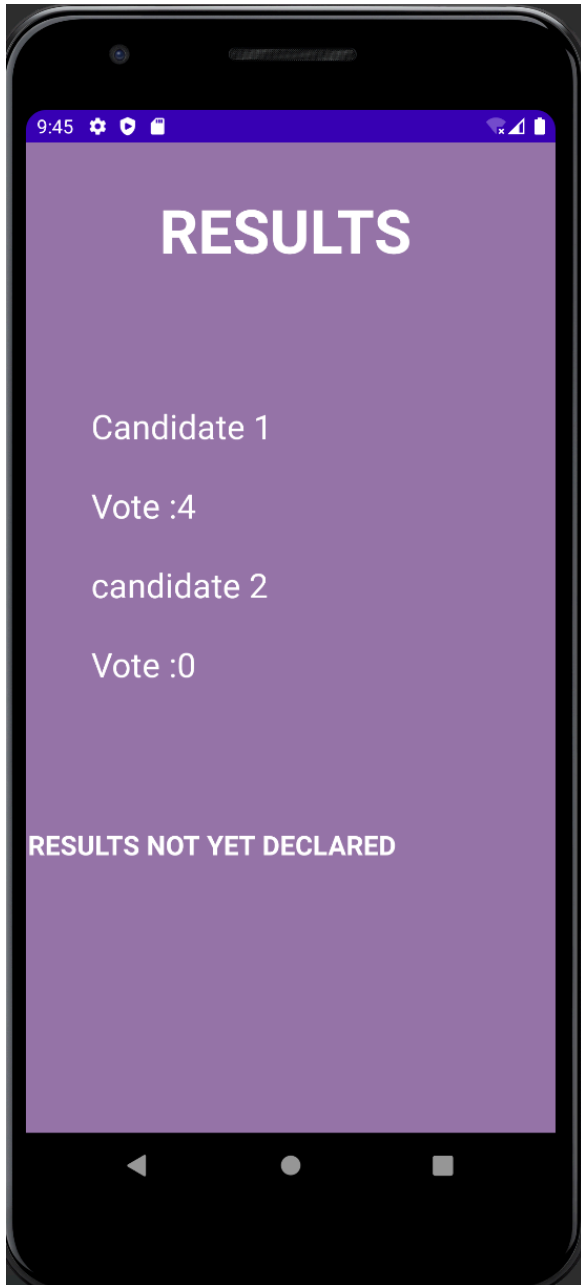


Frame 6: Once the user has signed in successfully they can cast their vote. This frame contains two buttons for the two candidates, These buttons are enabled for only one click to ensure no voter tries to vote twice. The data of the voter is also saved once they vote for a certain candidate to avoid multiple votings.



Frame 7: The user can then select to logout of their account which will terminate the application or view the results. The results are only published after a set number of votes have been casted.

If the user clicks results button before a set number of votes are casted the following message will appear or else the second frame containing the name of the winner will be displayed.



COMPARISION TABULAR

BASIS	THIS PROJECT	PAPER BASED VOTING	WEB BASED VOTING
AUTHENTICATION OF THE VOTER	High due to fingerprint and credentials	Lower	Fairly high due to credentials
RELIABILITY OF THE SYSTEM	Can face crashes	Always available	Can face network crashes
TRANSPARENCY OF RESULTS	Each vote can be viewed	Shows only final result	Shows only final result
ANONYMITY OF THE VOTER	Provided	Not provided	Provided
ACCURACY OF RESULTS	Computer operated hence high	Prone to human error	Computer operated hence high
EFFICENCY	High because system operated	Lower due to human efforts	High because system operated

CONCLUSION

People are evolving more and more with the passage of time which means they wish to be involved and be informed about anything and everything that affects them, this largely includes selecting the people that will be ruling them. But time has also made it impossible for people to always be available for such things hence remote innovations are helping carry out these demands. This project gives voters the ability to select their leaders at their own convenience with utmost security and reliability. It fulfills the anonymity, security, and accuracy characteristics. However, as this is the initial project with greater innovations and trials it can be perfected even further and minor loopholes can be avoided. This project is also extremely open to future additions like facial recognition for better security or device serial number storage for better accuracy. This project is the first step toward something bigger, better, and safer.

REFERENCES

1. <https://www.statista.com/chart/17690/registered-voters-in-the-most-recent-national-election/>
2. <https://www.worldometers.info/world-population/pakistan-population/>
3. <https://scroll.in/article/921543/the-election-fix-women-are-voting-in-record-numbers-but-are-parties-listening-to-them>
4. <https://worldpopulationreview.com/countries/india-population>
5. International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015 “An Implementation of Secure Online Voting System”
6. An Efficient Online Voting System, ISSN 2249-6645, Volume-2, Issue, July-Aug-2012, IJMER.
7. L. Foresti, C. S. Regazzoni, and R. Visvanathan, Scanning the issue technology Special issue on video communications, processing and understanding for third generation surveillance systems, Proc. IEEE, vol. 89, no. 10, pp. 1355-1367, Oct. 2016.
8. S. Misra, M. Reisslein, and G. Xue, A survey of multimedia streaming in wireless sensor networks, IEEE Communications Surveys and Tutorials, Fourth Quarter, vol. 10, no. 4, pp. 1839, 2015.
9. Rezwan, Rahil, et al. "Biometrically secured electronic voting machine." 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2017
10. Sarker, M. Mesbahuddin, and Tajim Md Niamat Ullah Akhund. "The roadmap to the electronic voting system development: a literature review." International Journal of Advanced Engineering, Management and Science 2.5 (2016)
11. Efficient E-voting Android Based System, Volume-3, Issue-11, Nov-2013, IJARCSSE.
12. Highly Secure Online Voting System with Multi Security using Biometric and Steganography. ISSN: 2249-9954 ISSUE 2, VOLUME 2 (APRIL 2012)
13. Design a Secure Electronic Voting System Using Fingerprint Technique, ISSN 1694-0784, Volume-10, Issue -4, and IJCSI.
14. International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015 “An Implementation of Secure Online Voting System”
15. Gentles, D and Suresh, S (2011). “Biometric Secured Mobile Voting”, Proceedings of Second IEEE/IFIP Asian Himalayas International Conference on Internet, Kathmandu, Nepal.

CODE:

1. DB HELPER JAVA CLASS

```
package com.learnandroid.loginsqlite;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;
```

```
public class DBHelper extends SQLiteOpenHelper {
    public static final String DBNAME = "Login.db";
    public DBHelper(Context context) {
        super(context, "Login.db", null, 1);
    }
```

```
    @Override
    public void onCreate(SQLiteDatabase MyDB) {
        MyDB.execSQL("create Table users(username TEXT
primary key, password TEXT)");
    }
```

```
    @Override
    public void onUpgrade(SQLiteDatabase MyDB, int i, int i1)
    {
        MyDB.execSQL("drop Table if exists users");
    }
```

```
    public Boolean insertData(String username, String
password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
```

```

        ContentValues contentValues= new ContentValues();
        contentValues.put("username", username);
        contentValues.put("password", password);
        long result = MyDB.insert("users", null,
contentValues);
        if(result==-1) return false;
        else
            return true;
    }

```

```

    public Boolean checkusername(String username) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from users
where username = ?", new String[]{username});
        if (cursor.getCount() > 0)
            return true;
        else
            return false;
    }

```

```

    public Boolean checkusernamepassword(String username,
String password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from users
where username = ? and password = ?", new String[]
{username,password});
        if(cursor.getCount(>)>0)
            return true;
        else
            return false;
    }
}

```

2. DB HELPER2 JAVA CLASS

```
package com.learnandroid.loginsqlite;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;
```

```
public class DBHelper2 extends SQLiteOpenHelper {
    public DBHelper2(Context context) {
```

```
        super(context, "Candidate.db", null, 1);
    }
```

```
    @Override
    public void onCreate(SQLiteDatabase DB) {
        DB.execSQL("create Table candidate(candidate TEXT
primary key, vote INTEGER default 0, username TEXT );");
    }
```

```
    @Override
    public void onUpgrade(SQLiteDatabase DB, int i, int i1) {
        DB.execSQL("drop Table if exists candidate");
    }
```

```
    public Boolean updateCanData(String candidate, Integer
vote,String username) {
        SQLiteDatabase DB = this.getReadableDatabase();
```

```

        DB.execSQL("UPDATE candidate SET vote= vote + 1,
username='"+username+"' WHERE
candidate='"+candidate+"'");
        return true;
    }

```

```

    public Boolean insertcanData(String username){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        ContentValues contentValues= new ContentValues();
        contentValues.put("username", username);
        long result = MyDB.insert("candidate", null,
contentValues);
        if(result==-1) return false;
        else
            return true;
    }

```

```

    public Boolean checkusername2(String username) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from
candidate where username = ?", new String[]{username});
        if (cursor.getCount() > 0)
            return true;
        else
            return false;
    }

```

```

    public Cursor ViewData(){
        SQLiteDatabase DB=this.getReadableDatabase();
        Cursor cursor=DB.rawQuery("SELECT * FROM candidate",
null);
        return cursor;
    }

```

```

    }

    public Cursor ViewData2(){
        SQLiteDatabase DB=this.getReadableDatabase();
        Cursor cursor2=DB.rawQuery("select * from candidate
where ( SELECT sum(vote)=5 from candidate) and VOTE =
(SELECT MAX(VOTE) FROM CANDIDATE)", null);

        return cursor2; }
}

```

3. FINGERPRINT JAVA CLASS

```

package com.learnandroid.loginsqlite;

```

```

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.biometric.BiometricPrompt;
import androidx.core.content.ContextCompat;

```

```

import android.content.Intent;
import android.graphics.Color;
import android.hardware.biometrics.BiometricManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

```

```

import java.util.concurrent.Executor;

```

```

public class fingerprint extends AppCompatActivity {

```

```

@RequiresApi(api = Build.VERSION_CODES.Q)
@Override
public void onBackPressed(){}
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_fingerprint);
    Button scan_btn=findViewById(R.id.scan_btn);
    TextView msg_txt=findViewById(R.id.txt_msg);
    androidx.biometric.BiometricManager biometricManager=
    androidx.biometric.BiometricManager.from(this);

```

```

        switch(biometricManager.canAuthenticate())
        {
            case BiometricManager.BIOMETRIC_SUCCESS:
                msg_txt.setText("You Can use the fingerprint
sensor");
                break;
            case BiometricManager.BIOMETRIC_ERROR_NO_HARDWARE:
                msg_txt.setText("The device does not have a
fingerprint sensor");
                scan_btn.setEnabled(false);
                break;
            case BiometricManager.BIOMETRIC_ERROR_HW_UNAVAILABLE:
                msg_txt.setText("The sensors are currently
unavailable");
                scan_btn.setEnabled(false);
                break;
            case BiometricManager.BIOMETRIC_ERROR_NONE_ENROLLED:

```

```
msg_txt.setText("The device has no saved  
fingerprints");  
scan_btn.setEnabled(false);  
break;
```

```
}  
Executor executor=  
ContextCompat.getMainExecutor( this);  
androidx.biometric.BiometricPrompt  
biometricPrompt=new  
androidx.biometric.BiometricPrompt(fingerprint.this,  
executor, new  
androidx.biometric.BiometricPrompt.AuthenticationCallback() {  
    @Override  
    public void onAuthenticationError(int errorCode,  
@NonNull CharSequence errString) {  
        super.onAuthenticationError(errorCode,  
errString);  
    }  
}
```

```
    @Override  
    public void onAuthenticationSucceeded(@NonNull  
androidx.biometric.BiometricPrompt.AuthenticationResult  
result) {  
        super.onAuthenticationSucceeded(result);  
        Toast.makeText(fingerprint.this, "LOGIN  
SUCESS", Toast.LENGTH_LONG).show();  
        Intent intent = new  
Intent(getApplicationContext(), HomeActivity.class);  
        startActivity(intent);  
    }  
}
```

```
    @Override  
    public void onAuthenticationFailed() {
```

```
        super.onAuthenticationFailed();  
    }  
    });
```

```
        BiometricPrompt.PromptInfo promptInfo= new  
BiometricPrompt.PromptInfo.Builder()  
        .setTitle("Scan")  
        .setDescription("Use your fingerprint to  
scan")  
        .setNegativeButtonText("Cancel")  
        .build();  
        scan_btn.setOnClickListener(new  
View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                biometricPrompt.authenticate(promptInfo);  
            }  
        });  
    }  
}
```

4. FIRST SCREEN JAVA CLASS

```
package com.learnandroid.loginsqlite;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;
```



```
public class first extends AppCompatActivity {  
    Button register, signin;
```

```
    @Override  
    public void onBackPressed() {}  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_first);
```

```
        register = (Button) findViewById(R.id.btnsignup);  
        signin = (Button) findViewById(R.id.btsignin);
```

```
        signin.setOnClickListener(new View.OnClickListener() {  
            public void onClick(View view) {  
                Intent intent = new Intent(getApplicationContext(),  
LoginActivity.class);  
                startActivity(intent); }  
        });
```

```
        register.setOnClickListener(new  
View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Intent intent = new  
Intent(getApplicationContext(), MainActivity.class);  
                startActivity(intent); }  
        });
```

```
} }
```

5. HOME ACTIVITY JAVA CLASS

```
package com.learnandroid.loginsqlite;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;
```

```
public class HomeActivity extends AppCompatActivity {  
    Button cand1_btn, cand2_btn, result_btn, logout_btn;  
    TextView cand1, cand2;  
    DBHelper2 DB;
```

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

```
        cand1 = findViewById(R.id.cand1_count);  
        cand2 = findViewById(R.id.cand2_count);
```

```
        cand1_btn = findViewById(R.id.btncand1);  
        cand2_btn = findViewById(R.id.btncand2);
```

```
        result_btn=findViewById(R.id.result_btn);
        logout_btn=findViewById(R.id.logout_btn);
        DB= new DBHelper2(this);
```

```

        cand1_btn.setOnClickListener(new
View.OnClickListener() {
            public void onClick(View view) {
                SharedPreferences sharedPreferences =
getSharedPreferences("myKey", MODE_PRIVATE);
                String username =
sharedPreferences.getString("value", "");
                Boolean checkuser =
DB.checkusername2(username);
                if(checkuser==false){
                    cand1_btn.setEnabled(false);
                    cand2_btn.setEnabled(false);
                    int count = 1;
                    cand1.setText(String.valueOf(count));
                    String candidate = "Candidate 1";
                    Integer vote = 1;
                    Boolean
checkupdatedata=DB.updateCanData(candidate,vote,username);
                    if(checkupdatedata==true){

```

```

        Toast.makeText(HomeActivity.this, "Vote
Registered", Toast.LENGTH_SHORT).show();
    }
    else {

```

```
Toast.makeText(HomeActivity.this,
"Unsuccessful", Toast.LENGTH_SHORT).show();
    }
}
```

```

        else {
            Toast.makeText(HomeActivity.this, "You
have already voted once", Toast.LENGTH_SHORT).show();
        }
    }
});

```

```

        cand2_btn.setOnClickListener(new
View.OnClickListener() {
            public void onClick(View view) {
                SharedPreferences sharedPreferences =
getSharedPreferences("myKey", MODE_PRIVATE);
                String username =
sharedPreferences.getString("value", "");
                Boolean checkuser =
DB.checkusername2(username);
                if(checkuser==false){
                    cand1_btn.setEnabled(false);
                    cand2_btn.setEnabled(false);
                    int count = 1;
                    cand2.setText(String.valueOf(count));
                    String candidate = "Candidate 2";
                    Integer vote = 1;
                    Boolean
checkupdatedata=DB.updateCanData(candidate,vote,username);
                    if(checkupdatedata==true){
                        Toast.makeText(HomeActivity.this, "Vote
Registered", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        }
    }
}
else {

```

```

        Toast.makeText(HomeActivity.this,
"Unsuccessful", Toast.LENGTH_SHORT).show();
    }
}

```

```

    }
}
else {
    Toast.makeText(HomeActivity.this, "You
have already voted once", Toast.LENGTH_SHORT).show();
}
}
});

```

```

        logout_btn.setOnClickListener(new
View.OnClickListener() {
            public void onClick(View view) {
                Intent intent = new
Intent(getApplicationContext(), LoginActivity.class);
                startActivity(intent);
            }
        });

        result_btn.setOnClickListener(new
View.OnClickListener() {
            public void onClick(View view) {
                Intent intent = new
Intent(getApplicationContext(), ResultActivity.class);
                startActivity(intent);
            }
        });
    }
}
}

```

6. LOGIN JAVA CLASS

```

package com.learnandroid.loginsqlite;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;

```

```

import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {
    EditText username, password;
    Button btnlogin;
    DBHelper DB;
    @Override
    public void onBackPressed() {}
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        username = (EditText) findViewById(R.id.username1);
        password = (EditText) findViewById(R.id.password1);
        btnlogin = (Button) findViewById(R.id.btnsignin1);
        DB = new DBHelper(this);
        btnlogin.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

                String user = username.getText().toString();
                SharedPreferences sharedPref =
getSharedPreferences("myKey", MODE_PRIVATE);
                SharedPreferences.Editor editor =
sharedPref.edit();
                editor.putString("value", user);
                editor.apply();

```

```
String pass = password.getText().toString();
```

```
        if(user.equals("") || pass.equals(""))
            Toast.makeText(LoginActivity.this,
"Please enter all the fields", Toast.LENGTH_SHORT).show();
        else{
            Boolean checkuserpass =
DB.checkusernamepassword(user, pass);
            if(checkuserpass==true){
                Toast.makeText(LoginActivity.this,
"Sign in successfull", Toast.LENGTH_SHORT).show();
                Intent intent = new
Intent(getApplicationContext(), fingerprint.class);
                startActivity(intent);
            }else{
                Toast.makeText(LoginActivity.this,
"Invalid Credentials", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
});
}
```

7. MAIN ACTIVITY JAVA CLASS

```
package com.learnandroid.loginsqlite;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;

import android.content.Intent;
```

```
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    EditText username, password, repassword;
```

```
    Button signup;
```

```
    DBHelper DB;
```

```
    @Override
```

```
    public void onBackPressed() {}
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        username = (EditText) findViewById(R.id.username);
```

```
        password = (EditText) findViewById(R.id.password);
```

```
        repassword = (EditText)
findViewById(R.id.repassword);
```

```
        signup = (Button) findViewById(R.id.btnsignup);
```

```
        DB = new DBHelper(this);
```

```
        signup.setOnClickListener(new View.OnClickListener()
{
```

```
            @Override
```

```
            public void onClick(View view) {
```

```
                String user = username.getText().toString();
```

```
                String pass = password.getText().toString();
```



```

        String repass =
repassword.getText().toString();
        if(user.equals("") || pass.equals("") ||
repass.equals(""))
            Toast.makeText(MainActivity.this, "Please
enter all the fields", Toast.LENGTH_LONG).show();
        else{
            if(pass.equals(repass)){
                Boolean checkuser =
DB.checkusername(user);
                if(checkuser==false){
                    Boolean insert =
DB.insertData(user, pass);
                    if(insert==true){

```

```

Toast.makeText(MainActivity.this, "Registered successfully",
Toast.LENGTH_SHORT).show();
                Intent intent = new
Intent(getApplicationContext(), LoginActivity.class);
                startActivity(intent);
            }else{

```

```

Toast.makeText(MainActivity.this, "Registration failed",
Toast.LENGTH_SHORT).show();
            }
        }
        else{
            Toast toast =
Toast.makeText(MainActivity.this, "User Already exists",
Toast.LENGTH_LONG);
            toast.setGravity(Gravity.TOP, 0,
0);
            toast.show();

```

```

    }

```

```

        }else{
            Toast.makeText(MainActivity.this,
"Passwords not matching", Toast.LENGTH_SHORT).show();
        }
    } }
});

```

```

    }
}

```

8. RESULT ACTIVITY JAVA CLASS

```

package com.learnandroid.loginsqlite;

```

```

import androidx.appcompat.app.AppCompatActivity;

```

```

import android.database.Cursor;

```

```

import android.os.Bundle;

```

```

import android.widget.TextView;

```

```

public class ResultActivity extends AppCompatActivity {
    DBHelper2 DB;
    @Override
    public void onBackPressed(){}
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result);
        DB = new DBHelper2(this);
        TextView textview = findViewById(R.id.textView2);
        Cursor cursor = DB.ViewData();
        StringBuilder stringBuilder = new StringBuilder();
        while (cursor.moveToNext()) {

```

```
        stringBuilder.append("\n \n " +
cursor.getString(0)
        + "\n \n Vote :" + cursor.getInt(1));
    }
    textview.setText(stringBuilder);
```

```
TextView textview2 = findViewById(R.id.textView3);
Cursor cursor2 = DB.ViewData2();
StringBuilder stringBuilder2 = new StringBuilder();
if (cursor != null) {
    textview2.setText("Result not yet declared");
} else {
    while (cursor2.moveToNext()) {
        stringBuilder2.append("\n \n " +
cursor2.getString(0));
    }
    textview2.setText(stringBuilder2 + " is the
Winner");
}
}
```