

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**Jnana Sangama, Belgaum-590018**



**A Mobile Application Development Laboratory Mini Project Report on**  
**“E-VOTING SYSTEM”**

**Submitted in Partial fulfillment of the Requirements for VI Semester of the Degree of**

**Bachelor of Engineering In**  
**Computer Science & Engineering**

**By**

**DADIREDDY SAI KUMAR REDDY**  
**(1CR19CS035)**

**DARSHAN R**  
**(1CR19CS037)**

**Under the Guidance of**

**Mr. Kartheek GCR**  
**Asst Professor, Dept. of CSE**  
**And**

**Mrs. Poonam Vijay Tijare**  
**Asst Professor, Dept. of CSE**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**CMR INSTITUTE OF TECHNOLOGY**

**#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,**  
**BANGALORE-560037**

# CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,  
BANGALORE-560037

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



### CERTIFICATE

This is to certify that the Mobile Application Development project work entitled “**E-Voting System**” has been carried out by **Dadireddy Sai Kumar Reddy (1CR19CS035)** and **Darshan R (1CR19CS037)** bonafide students of CMR Institute of Technology in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2021-2022**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. This MAD project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

-----  
**Signature of Guide**

**Mr. Kartheek GCR**  
Asst Professor  
Dept. of CSE, CMRIT

-----  
**Signature of Guide**

**Mrs. Poonam Vijay Tijare**  
Asst Professor  
Dept. of CSE, CMRIT

-----  
**Signature of HOD**

**Dr. Shreekanth Mooroor Prabhu**  
Professor & Head  
Dept. of CSE, CMRIT

External Viva

Name of the examiners

Signature with date

- 1.
- 2.

# ABSTRACT

The Project is developed for the threat free and user oriented Online Voting System. The Online Voting system is made for the people of the country residing around the world and wants to vote for their representative. The election can be conducted in two ways the paper ballot election and the automated ballot elections.

The automated ballot elections are called the electronic voting. The online voting system is highly developed and the online polling system can be replaced by accurately and directly voting online and immediate results. The online voting system is done by the internet so it can be called the Internet Voting or E-Voting.

The online voting system consists of the data and information. The database of the Candidate's information and details, Voter's Id, Calculation of total votes, Checking information by the voter, Remove wrong information, Creating an election type, Adding candidates to the election, Voting for a Candidate, Election information can be immediately transferred to Election Commission.

## ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them. We take the privilege to express our sincere thanks to **Dr. Sanjay Jain**, Principal for providing the encouragement and much support throughout our work. We are deeply indebted to Prof. **Dr. Shreekanth Mooroor Prabhu** (Head of Department of CSE) and the entire team in the Computer Department. They supported us with scientific guidance, advice and encouragement, they were always helpful and enthusiastic and this inspired us in our work. We would like to extend our sincere esteems towards our guides, **Mr. Kartheek GCR and Mrs. Poonam Vijay Tijare** for the support, guidance and encouragement, they provided during the BE Project. This work would have not been possible without their valuable time, patience and motivation. We thank them for making our stint thoroughly pleasant and enriching.

D Sai Kumar Reddy(1CR19CS035)

Darshan R (1CR19CS037)

# Table of Contents

1. Introduction.....	1
1.1 Objectives	
1.2 Justification	
2. System Requirements .....	2
2.1 Introduction	
2.2 Functional Requirements	
2.3 Non-Functional Requirements	
2.4 Details of Software	
2.5 Hardware Requirements	
2.6 Software Requirements	
3. Design.....	3
3.1 XML File	
3.2 Front End Activity Design	
4. Implementation.....	21
4.1 Java File	
4.2 Manifests File	
5. Discussion and Screenshots.....	33
5.1 Login Page	
5.2 Registration and Reset Password	
5.3 Admin Home Page	
5.4 User Home Page	
6. Conclusion and Future Scope .....	38
7. Bibliography .....	39

## **List of Figures**

3.2.1 Login Page Design.....	18
3.2.2 Admin Home Page Design .....	19
3.2.3 User Home Page Design .....	20
5.1.1 Login Page .....	33
5.1.2 Biometric Authentication.....	33
5.2.1 Registration Page .....	34
5.2.2 Password Reset .....	34
5.3.1 Admin Home Page.....	35
5.3.2 Candidate Operation .....	35
5.3.3 Create Election.....	36
5.3.4 Select Election.....	36
5.3.5 Result Sheet.....	36
5.4.1 User Home Page.....	37
5.4.2 Voting Page.....	37

## Chapter 1

# INTRODUCTION

The Online voting system (OVS) also known as E-Voting is a term encompassing several different types of voting embracing both electronic means of counting votes. Electronic voting technology can include punched cards, optical scan voting systems and specialized voting kiosks (including self contained direct-recording electronic voting systems or DRE). It can also involve transmission of ballots and votes via telephones, private computer networks, or the internet.

Online voting is an electronic way of choosing leaders via a mobile application. The advantage of online voting over the common "queue method" is that the voters have the choice of voting at their own free time and there is reduced congestion. It also minimizes on errors of vote counting. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes.

## 1.1 Objectives

1. To develop an Android application “E-VOTING SYSTEM” (Online Voting System) by using Android Studio 4.2.2 .
2. To have an effective and error free elections through online.

## 1.2 Justification

1. The ONLINE VOTING SYSTEM shall reduce the time spend making long queues at the polling stations during voting.
2. It shall also enable the voters to vote from any part of the globe as explained since this is an online application available on the internet.
3. Cases of vote miscounts shall also be solved.

## Chapter 2

# SYSTEM REQUIREMENTS

## 2.1 INTRODUCTION

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system.

## 2.2 Functional Requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases.

The various methods used in this project are as follows:-

1. Emulator: To perform and display the functionality of the project.
2. Android studio: To create, design, test, debug and run the android project.
3. Mouse: To navigate through the emulator.

## 2.3 Non-functional requirements:

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements Often apply to the system as a whole.

Non-functional Requirements are as follows:

### 2.3.1 Dependability:

The dependability of a computer system is a property of the system that equates to its trustworthiness.

### 2.3.2 Availability:

The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.



### **2.3.3Reliability:**

The ability of the system to deliver services as specified. The program is compatible with all types Of operating system without any failure.

### **2.3.4Safety:**

The ability of the system to Operate without catastrophic failure. This program is user friendly and it will never effects the system.

### **2.3.5Security:**

The ability of the system to protect itself against accidental or deliberate intrusion.

## **2.4 Details Of Software:**

Here, the coding, creating, designing, testing, debugging and running of our project is done in Android Studio.

### **2.4.1 Android 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 or as a subscriptionbased since in 2020. It is a replacement for the Eclipse Android Development Tools (EADT as the primary IDE for native Android application development.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features.

While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

## 2.5 Hardware Requirements

1. x86\_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
2. 8 GB RAM or more
3. 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
4. 1280 x 800 minimum screen resolution

## 2.6 Software Requirements

1. 64-bit Microsoft® Windows® 8/10
2. Android Studio version 4.2.2 or higher
3. An emulator or mobile device of Software Development Kit (SDK) 26 or higher 4 JDK 14.0.2
4. Firebase

## Chapter 3

# DESIGN

### 3.1 XML File

#### 3.1.1 Login Page

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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"
    tools:context=".LoginPage">

    <TextView
        android:id="@+id/banner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="E-Voting"
        android:textSize="60sp"
        android:textAlignment="center"
        android:textStyle="bold"
        android:fontFamily="sans-serif-black"
        android:layout_marginTop="50dp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:gravity="center_horizontal" />

    <TextView
        android:id="@+id/bannerDescription"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

    android:text="Easy Voting"
    android:textStyle="italic"
    android:textSize="18sp"
    android:textAlignment="center"
    app:layout_constraintTop_toBottomOf="@id/banner"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    android:gravity="center_horizontal" />

```

<EditText

```

    android:layout_width="380dp"
    android:layout_height="wrap_content"
    android:id="@+id/Mailadress"
    android:ems="10"
    android:inputType="textEmailAddress"
    android:textSize="24sp"
    android:hint="E-mail Adress"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="60dp"
    app:layout_constraintTop_toBottomOf="@id/bannerDescription"
    />

```

<EditText

```

    android:layout_width="380dp"
    android:layout_height="wrap_content"
    android:id="@+id/password"
    android:ems="10"
    android:inputType="textPassword"
    android:textSize="24sp"
    android:hint="Password"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    android:layout_marginTop="10dp"
    app:layout_constraintTop_toBottomOf="@id/Mailadress"
    />

```

<Button

```

    android:id="@+id/signIN"
    android:layout_width="300dp"
    android:layout_height="70dp"
    android:layout_marginTop="24dp"
    android:backgroundTint="#0ED689"
    android:text="LOGIN"
    android:textColor="@color/white"
    android:textSize="26sp"
    android:textStyle="bold"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/password"
/>

```

<TextView

```

    android:id="@+id/forgotpassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Forgot Password?"
    android:textStyle="bold"
    android:textSize="16sp"
    app:layout_constraintTop_toBottomOf="@id/signIN"
    app:layout_constraintLeft_toLeftOf="parent"
    android:layout_marginTop="25dp"
    android:layout_marginLeft="50dp"
/>

```

<TextView

```

    android:id="@+id/register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Register"
    android:textStyle="bold"
    android:textSize="16sp"
    app:layout_constraintLeft_toRightOf="@id/forgotpassword"

```

```
app:layout_constraintTop_toTopOf="@id/forgotpassword"
android:layout_marginLeft="80dp"
/>
```

```
<TextView
```

```
    android:id="@+id/fpText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="500dp"
    android:text="FingerPrint Option"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<ProgressBar
```

```
    android:id="@+id/progressBar"
    style="?android:attr/progressBarStyleLarge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:visibility="gone"
    tools:ignore="MissingConstraints"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

### 3.1.2 Admin Home Page

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    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=".HomePageAdmin"
```

```
    android:orientation="vertical"
```

```
    android:padding="10dp"
```

```
    android:background="#f0f0f0"
```

```
    android:gravity="center"
```

```
    android:id="@+id/ll">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="ADMIN PANEL"
```

```
    android:layout_marginBottom="20dp"
```

```
    android:textColor="@color/black"
```

```
    android:textStyle="bold"
```

```
    android:textSize="30dp"
```

```
<LinearLayout
```

```
    android:clipToPadding="false"
```

```
    android:gravity="center"
```

```
    android:orientation="horizontal"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
>
```

```
<androidx.cardview.widget.CardView // used to display any sort of data by providing a  
                                rounded corner layout along with a specific elevation
```

```

android:foreground="?android:attr/selectableItemBackground"
android:clickable="true"
android:layout_width="160dp"
android:layout_height="190dp"
android:layout_margin="10dp">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:id="@+id/candidateAdmin"
    android:gravity="center">
    <ImageView
        android:layout_width="64dp"
        android:layout_height="64dp"
        android:background="@drawable/cerclebackgroundpurple"
        android:src="@drawable/ic_baseline_account_circle_24"
        android:padding="10dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:layout_marginTop="10dp"
        android:text="Candidates"/>
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/lightgray"
        android:layout_margin="10dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="See the active candidates"
        android:padding="5dp"

```



```
        android:textColor="@android:color/darker_gray"/>
    </LinearLayout>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:foreground="?android:attr/selectableItemBackground"
    android:clickable="true"
    android:layout_width="160dp"
    android:layout_height="190dp"
    android:layout_margin="10dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:id="@+id/createElectionAdmin"
        android:gravity="center">
        <ImageView
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:background="@drawable/cerclebackgroundpink"
            android:src="@drawable/ic_baseline_create_24"
            android:padding="10dp"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:layout_marginTop="10dp"
            android:text="Create Election"/>
        <View
            android:layout_width="match_parent"
            android:layout_height="1dp"
            android:background="@color/lightgray"
            android:layout_margin="10dp"/>
        <TextView
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Creating new election"
        android:padding="5dp"
        android:textColor="@android:color/darker_gray"/>
    </LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
<LinearLayout
    android:clipToPadding="false"
    android:gravity="center"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <androidx.cardview.widget.CardView
        android:foreground="?android:attr/selectableItemBackground"
        android:clickable="true"
        android:layout_width="160dp"
        android:layout_height="190dp"
        android:layout_margin="10dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:id="@+id/results"
            android:gravity="center">
            <ImageView
                android:id="@+id/imageView2"
                android:layout_width="64dp"
                android:layout_height="64dp"
                app:srcCompat="@drawable/viewresults" /> // To add an image go to project -> res ->
                                                         drawable -> paste the image
            <TextView
```

```
        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:layout_marginTop="10dp"
        android:text="Results"/>
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@color/lightgray"
        android:layout_margin="10dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="View Results Here"
        android:padding="5dp"
        android:textColor="@android:color/darker_gray"/>

    </LinearLayout>
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:foreground="?android:attr/selectableItemBackground" // To add background image
    android:clickable="true"
    android:layout_width="160dp"
    android:layout_height="190dp"
    android:layout_margin="10dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:id="@+id/SettingsAdmin"
        android:orientation="vertical">
```

```
<ImageView
    android:layout_width="64dp"
    android:layout_height="64dp"
    android:background="@drawable/cerclebackgroundyellow"
    android:padding="10dp"
    android:src="@drawable/ic_baseline_settings_24" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="Settings"
    android:textStyle="bold" />
```

```
<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="10dp"
    android:background="@color/lightgray" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:padding="5dp"
    android:text="Profile Settings"
    android:textColor="@android:color/darker_gray" />
```

```
</LinearLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
</LinearLayout>
```

```
<LinearLayout
```

---

```

    android:clipToPadding="false"

    android:gravity="center"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <androidx.cardview.widget.CardView
        android:foreground="?android:attr/selectableItemBackground"
        android:clickable="true"
        android:layout_width="160dp"
        android:layout_height="190dp"
        android:layout_margin="10dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:id="@+id/logoutAdmin"
            android:gravity="center">
            <ImageView
                android:layout_width="64dp"
                android:layout_height="64dp"
                android:background="@drawable/cerclebackgroundpurplegreen"
                android:src="@drawable/ic_baseline_psychology_24"
                android:padding="10dp"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:textStyle="bold"
                android:layout_marginTop="10dp"
                android:text="Logout"/>
            <View
                android:layout_width="match_parent"
                android:layout_height="1dp"

```

---

```
        android:background="@color/lightgray"
        android:layout_margin="10dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Logout of the system"
        android:padding="5dp"
        android:textColor="@android:color/darker_gray"/>

    </LinearLayout>
</androidx.cardview.widget.CardView>
```

### 3.1.3 Voting Page

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"

    app:cardElevation="8dp"
    app:cardCornerRadius="8dp"
    android:layout_margin="16dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="16dp"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_vertical"
```

```
android:orientation="horizontal">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:text="Name : "
    android:textColor="@color/black"
    android:textSize="26sp"
    android:textStyle="bold"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/nameSelection"
    android:layout_gravity="center_vertical"
    android:text="Null"
    android:paddingLeft="15dp"
    android:textSize="26sp"
    android:textColor="@color/black"/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"

    android:orientation="horizontal">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:text="Nationality : "

        android:textColor="@color/black"
```

```

        android:textSize="26sp"
        android:textStyle="bold"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/nationalitySelection"
        android:layout_gravity="center_vertical"
        android:text="Null"
        android:paddingLeft="15dp"
        android:textSize="26sp"
        android:textColor="@color/black"/>

</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_vertical"

    android:orientation="horizontal">
    <TextView

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:text="Age : "

        android:textColor="@color/black"
        android:textSize="26sp"
        android:textStyle="bold"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/ageSelection"
        android:layout_gravity="center_vertical"

```



```
        android:text="Null"
        android:paddingLeft="15dp"
        android:textSize="26sp"
        android:textColor="@color/black"/>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="horizontal"
```

```
>
```

```
<Button
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:backgroundTint="@color/green"
    android:id="@+id/voteButton"
```

```
    android:text="Vote"
    android:textSize="18dp" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
</androidx.cardview.widget.CardView>
```

## 3.2 Front End Activity Design

### 3.2.1 Login Page

In Login Page we have used Text View to label App Name, Forget Password, Register and Finger Print Option. Edit Text has been used for Mail Address and Password fields. Also a Button is used for Login.

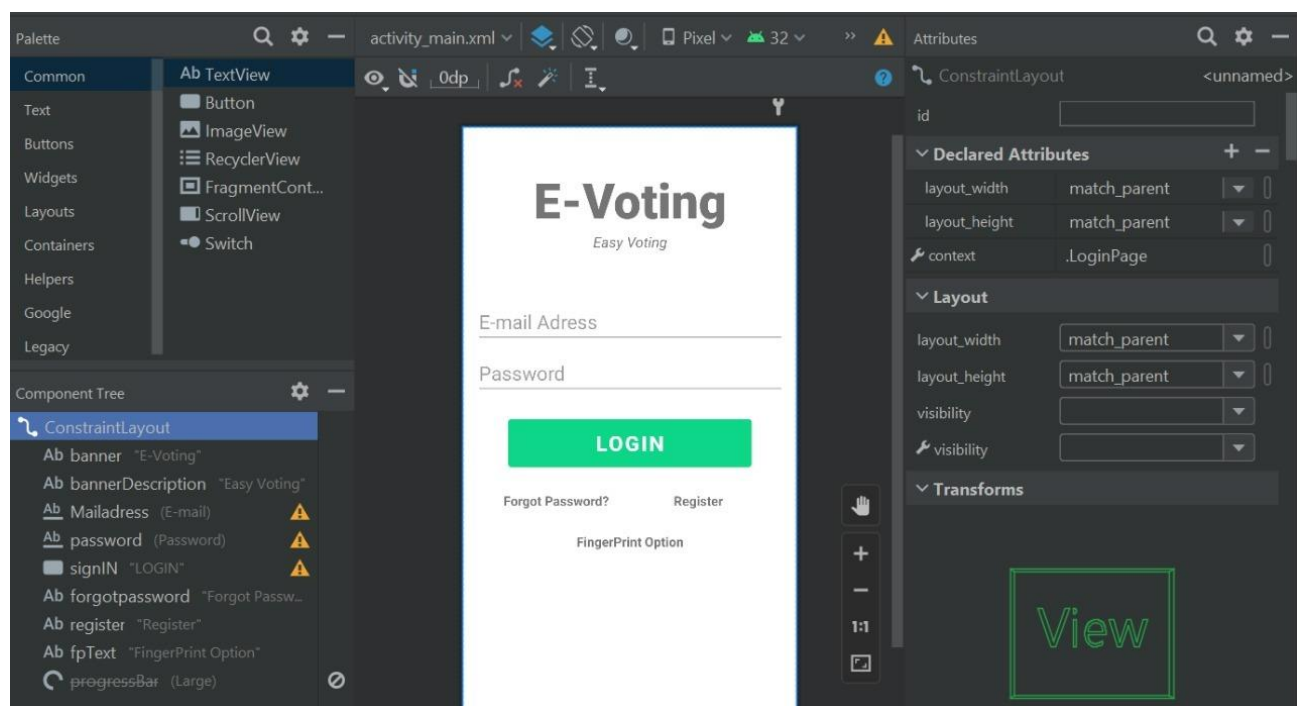


Fig 3.2.1: Login Page

### 3.2.2 Admin Home Page

In Admin Home Page we have created a Card View in a Linear Layout for the display of different options in the page. Image View has been used to place images for icons. View feature is used to keep a short description of the options available.

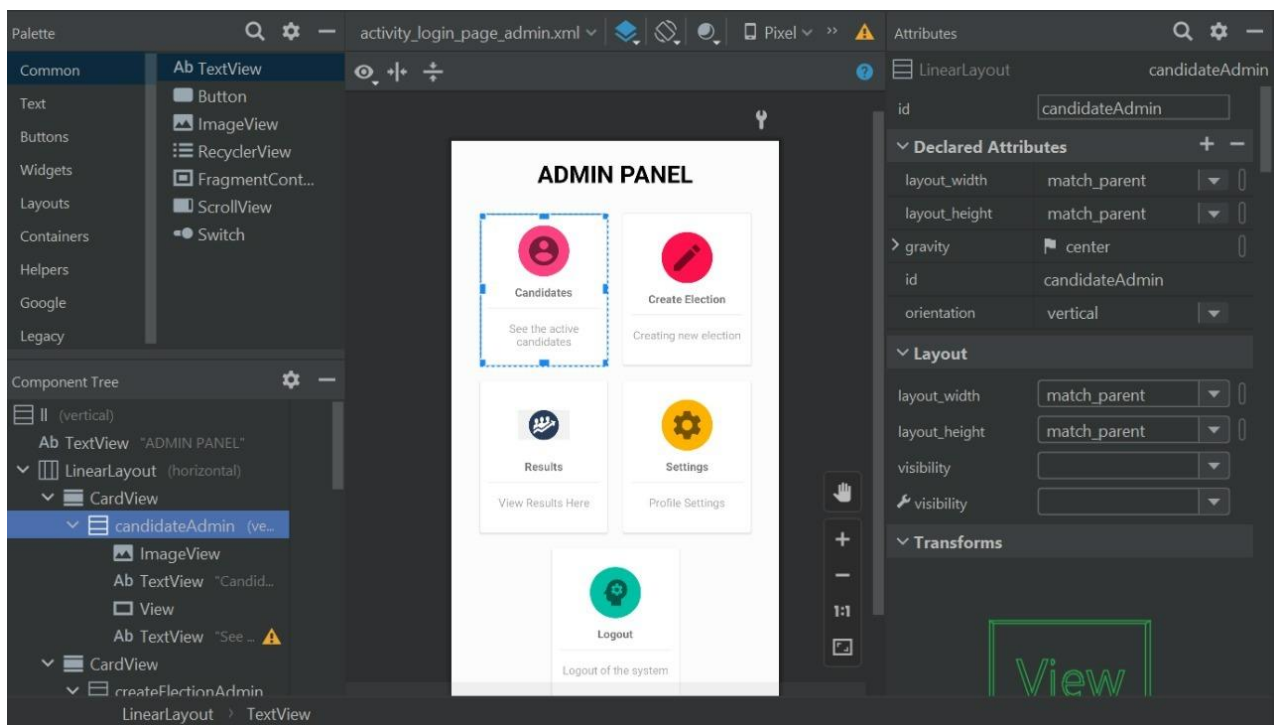


Fig 3.2.2 : Admin Home Page

### 3.2.3 User Home Page

In User Home Page we have created a Card View in a Linear Layout for the display of different options in the page. Image View has been used to place images for icons. View feature is used to keep a short description of the options available.

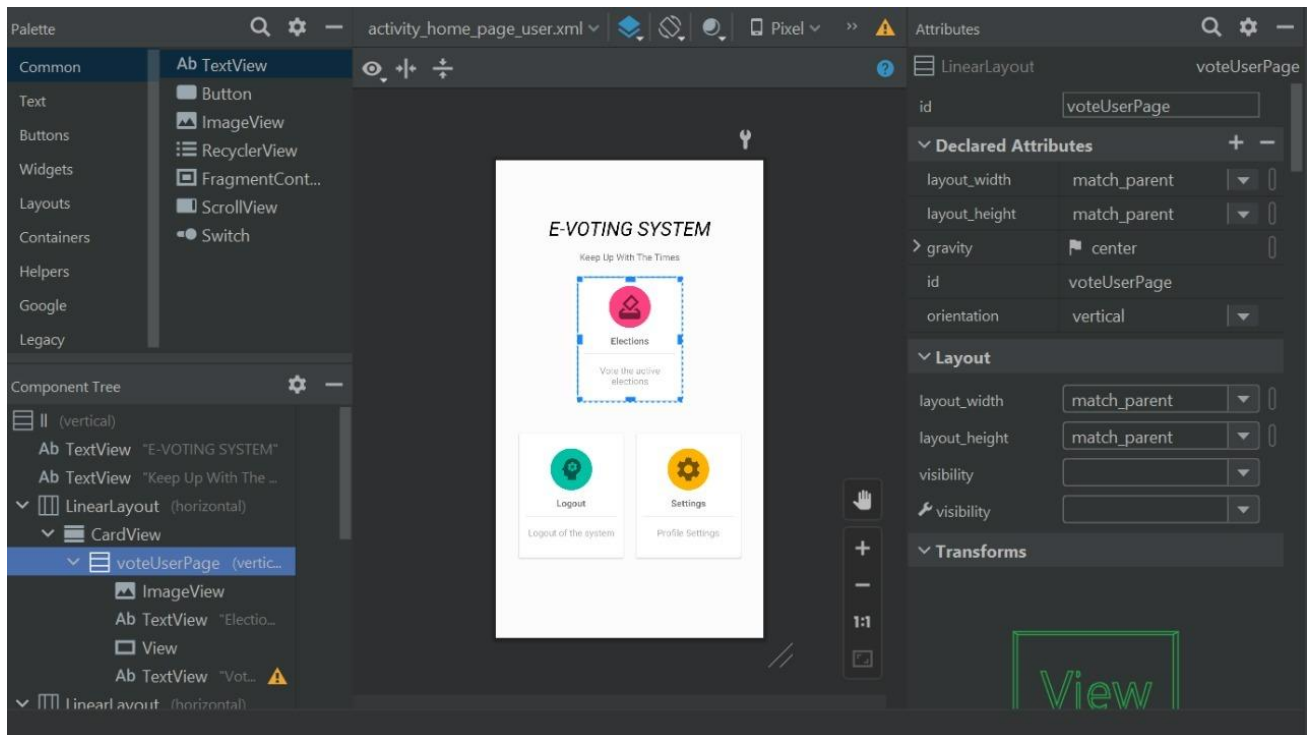


Fig 3.2.3 : User Home Page

## Chapter 4

# IMPLEMENTATION

### 4.1 Java File

#### 4.1.1 LoginPage.java

```
package com.example.e_votingsystem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;      // Import Firebase Modules for Database backup
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
```

```
public class LoginPage extends AppCompatActivity implements View.OnClickListener{

    private TextView register,forgotpassword, fingerprint;
    private EditText editTextEmail, editTextPassword;
    private Button signIn;
    private FirebaseUser user;
    private FirebaseAuth mAuth;
    private ProgressBar progressBar;
    int temp;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        register= findViewById(R.id.register);
        register.setOnClickListener(this);

        fingerprint= findViewById(R.id.fpText);
        fingerprint.setOnClickListener(this); // OnClickListener() interface has an onClick(View v)
                                                method that is called when the view (component) is
                                                clicked

        forgotpassword=findViewById(R.id.forgotpassword);
        forgotpassword.setOnClickListener(this);

        signIn=findViewById(R.id.signIn);
        signIn.setOnClickListener(this);

        editTextEmail=findViewById(R.id.Mailadress);
        editTextPassword=findViewById(R.id.password);

        progressBar=findViewById(R.id.progressBar);
```

## E-Voting App

---

```

    mAuth=FirebaseAuth.getInstance();

}

@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.register:
            startActivity(new Intent(this,RegisterUser.class));
            break;
        case R.id.forgotpassword:
            startActivity(new Intent(this,ResetPassword.class));
            break;
        case R.id.signIN:
            userLogin();
            break;
        case R.id.fpText:
            startActivity(new Intent(this,FingerPrintAuth.class));
            break;
    }
}

private void userLogin() {
    String email=editTextEmail.getText().toString().trim();
    String password=editTextPassword.getText().toString().trim();

    if (email.isEmpty()){
        editTextEmail.setError("Email is required");
        editTextEmail.requestFocus();
        return;
    }
    if(!Patterns.EMAIL_ADDRESS.matcher(email).matches()){
        editTextPassword.setError("Password is required"); // Checks for Email & Password Match
    }
}

```

---

```
        editTextPassword.requestFocus();
        return;
    }
    if (password.isEmpty()){
        editTextPassword.setError("Password is required!");
        editTextPassword.requestFocus();
        return;
    }
    if (password.length()<6){
        editTextPassword.setError("Min password length is 6 characters");
        editTextPassword.requestFocus();
        return;
    }
    progressBar.setVisibility(View.VISIBLE);
    mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new
    OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isComplete()){
                user=FirebaseAuth.getInstance().getCurrentUser(); //Getting instance from database
                if(user.isEmailVerified()){                                through Firebase to validate mail
                    if(task.isSuccessful()){                                and password
                        progressBar.setVisibility(View.GONE);
                        Toast.makeText(LoginPage.this,"LoginSuccess",Toast.LENGTH_SHORT).show();
                        checkedTypeOfUser(user.getUid());
                    }
                }
                else{
                    Toast.makeText(LoginPage.this,"Error ! : Mail or password is invalid! "
                    ,Toast.LENGTH_SHORT).show();
                    progressBar.setVisibility(View.GONE);
                }
            }
        }
    });
```



```
    }  
    else{  
        user.sendEmailVerification();           // Sending Email Verification to Login  
        Toast.makeText(LoginPage.this,"Check your email to verify your account.",  
        Toast.LENGTH_LONG).show();  
    }  
  
    }  
    else{  
        Toast.makeText(LoginPage.this,"Failed to login! Please check your credentials",  
        Toast.LENGTH_LONG ).show();  
    }  
    }  
    }  
    });  
}
```

```
public void checkedTypeOfUser(String Uid ) {  
    DatabaseReference                                     df=  
    FirebaseDatabase.getInstance().getReference().child("Users").child(Uid);  
    df.addValueEventListener(new ValueEventListener() {  
        @Override  
        public void onDataChange(@NonNull DataSnapshot snapshot) {  
  
            String typeAccountCheck=snapshot.child("typeOfAccount").getValue().toString().trim();  
  
            if(typeAccountCheck.equals("Admin")){  
  
                temp = 1;  
                startActivity(new Intent(LoginPage.this,FingerPrintAuthAdmin.class));  
  
            }  
        }  
    }  
}
```

```

else if(typeAccountCheck.equals("Users")){
    temp= 0;
    startActivity(new Intent(LoginPage.this, FingerPrintAuth.class)); // Checks for Finger
                                                                    Print
}
else{
    temp = -1;
    Toast.makeText(LoginPage.this,"Invalidaccount type",Toast.LENGTH_LONG).show();

}
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    Toast.makeText(LoginPage.this,"Something wrong happened.",Toast.LENGTH_SHORT)
        .show();
}
});
}
}

```

## 4.1.2 Create Candidate

```

package com.example.e_votingsystem;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;

```

## E-Voting App

---

```
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Spinner;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;

public class CreateCandidatePage2 extends AppCompatActivity {
    private EditText nameCandidate,numberofIdentity,nationality,age;
    private Spinner spinner;
    private Button save;
    private ProgressBar progressBar;
    private DatabaseReference dbref,dbrefElectionName;
    ValueEventListener listener;
    private ArrayList<String> list;
    private ArrayAdapter<String> adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create_candidate_page2);
        save=findViewById(R.id.createCandidateButton);
        numberofIdentity=findViewById(R.id.IdentityNumber);
        spinner=findViewById(R.id.spinner);
        nameCandidate=findViewById(R.id.CandidateName);
        nationality=findViewById(R.id.nationalityofCandidate);
        age=findViewById(R.id.ageOfCandidate);
        progressBar=findViewById(R.id.progressBar3);

        dbref=FirebaseDatabase.getInstance().getReference("Candidates"); //Getting data of candidates
```

---

```

dbrefElectionName =FirebaseDatabase.getInstance().getReference("Election");
list=new ArrayList<String>();
adapter=new
ArrayAdapter<String>(this,android.R.layout.simple_spinner_dropdown_item,list);
spinner.setAdapter(adapter); //spinner class for drop down view

save.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        insertData();
    }
});
fetchData();

}

private void insertData() {
    String name=nameCandidate.getText().toString().trim(); //To get input for the fields
    String age1=age.getText().toString().trim();
    String nation=nationality.getText().toString().trim();
    String Id=numberofIdentity.getText().toString().trim();
    String spinnerData=spinner.getSelectedItem().toString().trim();
    int voteCount=0;
    if(name.isEmpty()){
        nameCandidate.setError("Full name is required");
        nameCandidate.requestFocus();
        return;
    }
    if(age1.isEmpty()){
        age.setError("Age is required"); //Displaying Error Statements if field is empty
        age.requestFocus();
        return;
    }
}

```

```
if(nation.isEmpty()){
    nationality.setError("Nationality is required");
    nationality.requestFocus();
    return;
}
if(Id.isEmpty()){
    numberOfIdentity.setError("Id is required");
    numberOfIdentity.requestFocus();
    return;
}
if(spinnerData.equals("")) {
    Toast.makeText(this,"Please select which election",Toast.LENGTH_LONG).show();
    spinner.requestFocus();
    return;
}

Candidate candidate=new Candidate(name,voteCount,age1,nation,Id,spinnerData);
dbref.child(candidate.fullNameCandidate).setValue(candidate);
if(!candidate.equals("")) {
    Toast.makeText(CreateCandidatePage2.this,"Successfully!",Toast.LENGTH_LONG).show();
    startActivity(new Intent(CreateCandidatePage2.this, HomePageAdmin.class));
} //Intent is used to to navigate from one page to another page
else{
    Toast.makeText(CreateCandidatePage2.this, "Failed!",Toast.LENGTH_LONG).show();
}
progressBar.setVisibility(View.GONE);
}

public void fetchData(){
    listener=dbrefElectionName.addValueEventListener(new ValueEventListener() {
        //To receive event about data changes at a location

        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            for(DataSnapshot dataSnapshot:snapshot.getChildren())
```

```
{
    Election election=dataSnapshot.getValue(Election.class);
    list.add(election.name);
}
adapter.notifyDataSetChanged();
}

@Override
public void onCancelled(@NonNull DatabaseError error) {

}
});
}
}
```

#### 4.1.3 Voting Page:

```
package com.example.e_votingsystem;

import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;

public class MyAdapter2 extends RecyclerView.Adapter<MyAdapter2.MyViewHolder> {
    Context context; //RecyclerView displays large sets of data
    ArrayList<Candidate> list;
```

```
private OnItemClickListener onItemClickListener;

public MyAdapter2(Context context, ArrayList<Candidate> list ) {
    this.context = context;
    this.list = list;
}

@NonNull
@Override
public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View v= LayoutInflater.from(context).inflate(R.layout.candidatelist,parent,false);
    MyViewHolder mv=new MyViewHolder(v,onItemClickListener);
    return mv;
}

@Override // To update RecyclerView
public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
    Candidate candidate= list.get(position);
    holder.candidateTextName.setText(candidate.getFullNameCandidate());
    holder.ageText.setText(candidate.getAge());
    holder.nationText.setText(candidate.getNationality());
}

@Override
public int getItemCount() {
    return list.size();
}

public interface OnItemClickListener{
    void onVoteClick(int position);
}

public void setOnItemClickListener(OnItemClickListener listener){
    onItemClickListener=listener;
}

public class MyViewHolder extends RecyclerView.ViewHolder{
    TextView candidateTextName,ageText,nationText;
```

```
public MyViewHolder(@NonNull View itemView, final OnItemClickListener listener) {
    super(itemView);

    candidateTextName=itemView.findViewById(R.id.nameSelection);
    ageText=itemView.findViewById(R.id.ageSelection);
    nationText=itemView.findViewById(R.id.nationalitySelection);
    itemView.findViewById(R.id.voteButton).setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v) {
            if(listener!=null){
                int position = getAdapterPosition();
                if (position != RecyclerView.NO_POSITION) {
                    listener.onVoteClick(position);
                }
            }
        }
    });
}
```

## 4.2 Manifests File

To set Login Page as the launcher page for the application, following changes has to be done in Manifests File.

```
<activity android:name=".LoginPage">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```



## Chapter 5

### DISCUSSION AND SCREENSHOTS

The following snapshots define the results and outputs that we get from the application in step by step process.

#### 5.1 Login Page

1. Figure 5.1.1 is the launcher page of the app where the user enters his credentials to login to his account. If the user has not registered an account, he/she can register through Register option. Password can also be changed via Forgot Password option.
2. Figure 5.1.2 is the authentication page that arises when the user press login button. The user should unlock it via his fingerprint.

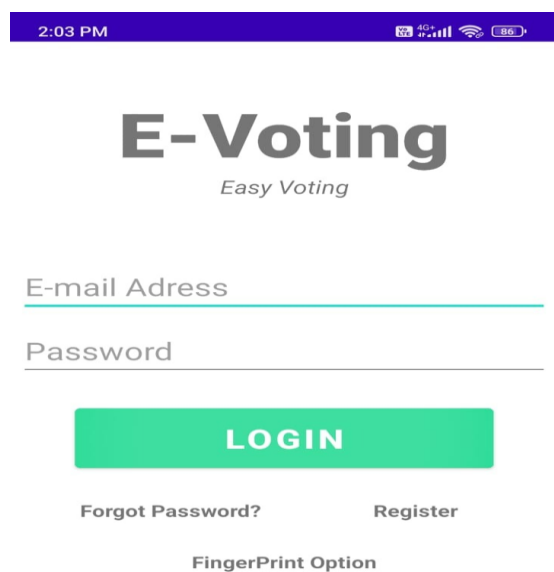


Fig 5.1.1: Login Page

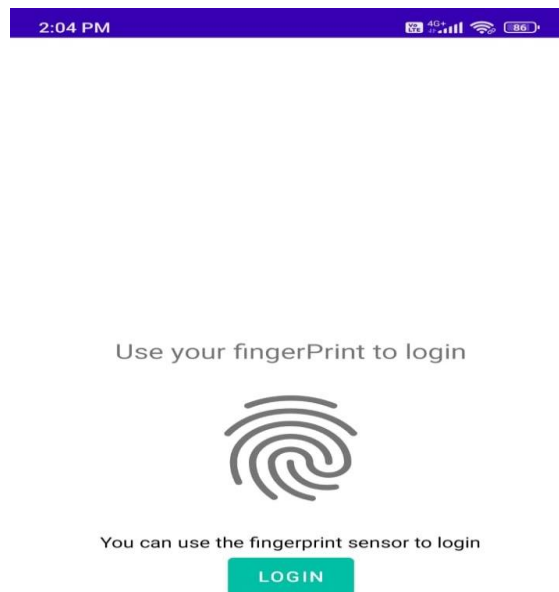
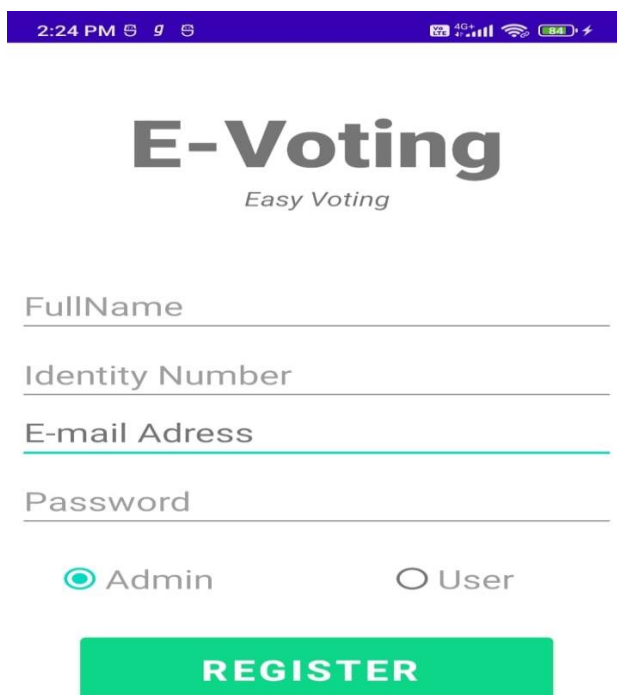


Fig 5.1.2: Biometric Authentication

## 5.2 Registration and Reset Password

1. Figure 5.2.1 is the Registration Page where a new user should enter his details and sets a password for his account.
2. Figure 5.2.2 is the Password Resetting, where the user enters the mail he used to register his account and submits, later a link will be shared to his mail through which the user can change the password.



2:24 PM 9 5 4G+ 54

# E-Voting

Easy Voting

FullName

Identity Number

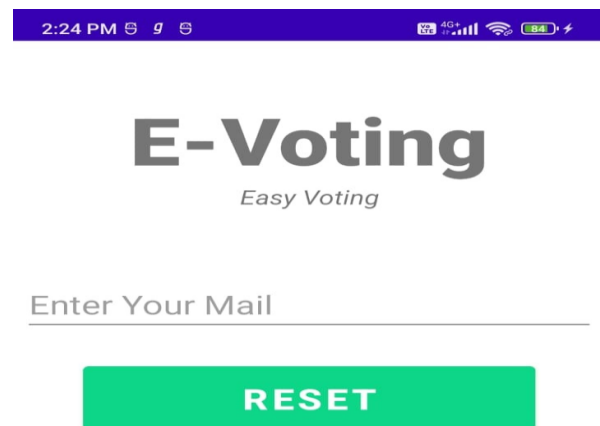
E-mail Address

Password

☒ Admin ☐ User

**REGISTER**

Fig 5.2.1 Registration Page



2:24 PM 9 5 4G+ 54

# E-Voting

Easy Voting

Enter Your Mail

**RESET**

Fig 5.2.2 Password Reset

## 5.3 Admin Home Page

1. Figure 5.3.1 is the Admin Home Page where we have different options like Candidates, Create Elections, Results, Settings, Logut.
2. Figure 5.3.2 is the Candidate Operation Option where we can add a candidate for particular election through Create Candidate or we can view the active candidates of a particular election in Candidates.
3. Figure 5.3.3 is the Creating an Election page which is intended on pressing Create Election Option in home page. Here we can add the type of election and select start and end date for the election.
4. Figure 5.3.4 is the different elections list that we get inside Results option. On selecting a particular election it intends to the result page of it.
5. Figure 5.3.5 is the Result Sheet where we have different candidates contested in the selected election with the amount of votes each received.

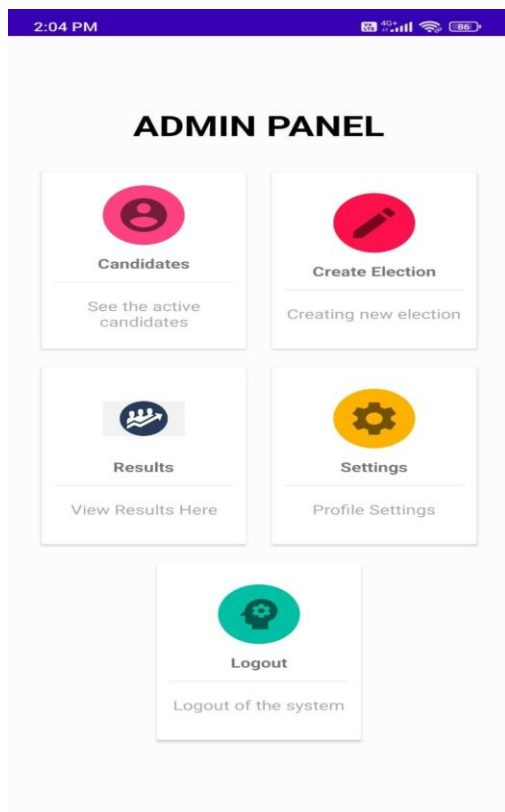


Fig 5.3.1 : Admin Home Page

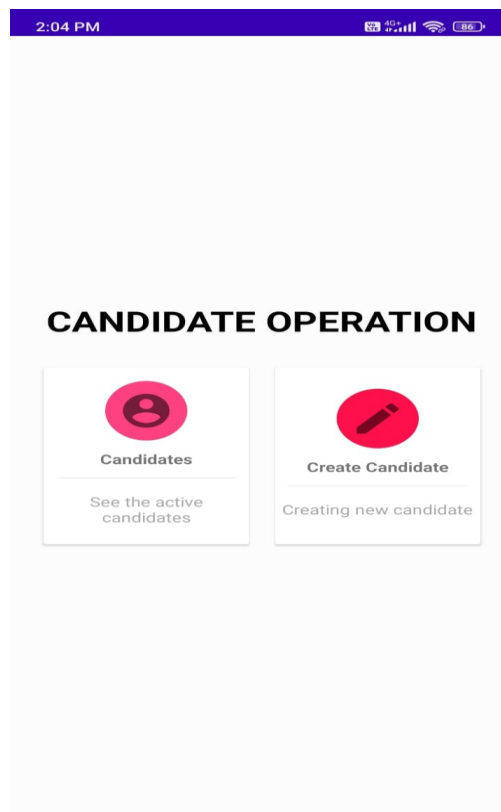


Fig5.3.2 : Candidate Operation

2:04 PM

## Create an Election

*Which type of election ?*

**Title**  
Name of Election

Start Date      End Date  
JUL 14 2022    JUL 14 2022

**CREATE ELECTION**

Fig 5.3.3 : Create Election

2:04 PM

## Select Election

**Due Date :** MAY 22 2022

**Election :** MP  
**Due Date :** JUL 14 2022

**Election :** MP  
**Due Date :** JUL 14 2022

**Election :** mla  
**Due Date :** JUL 15 2022

**Election :** President  
**Due Date :** MAY 28 2021

Fig 5.3.4 : Select Election

2:04 PM

## Candidate List

**Name :** a  
**Nationality :** india  
**Age :** 34  
7 Votes

**Name :** b  
**Nationality :** india  
**Age :** 32  
0 Votes

Fig 5.3.5 : Result Sheet

## 5.4 User Home Page

1. Figure 5.4.1 is the User Home Page where we have options like Elections , Settings and Logout.
2. Figure 5.4.2 is the Voting Page where the user gets the list of candidates contesting in the election and a vote button to caste their vote.

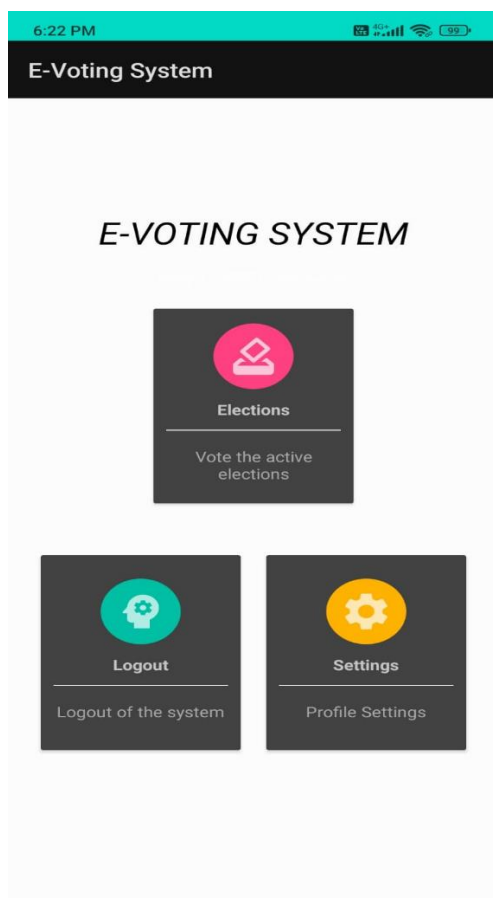


Fig 5.4.1 : User Home Page

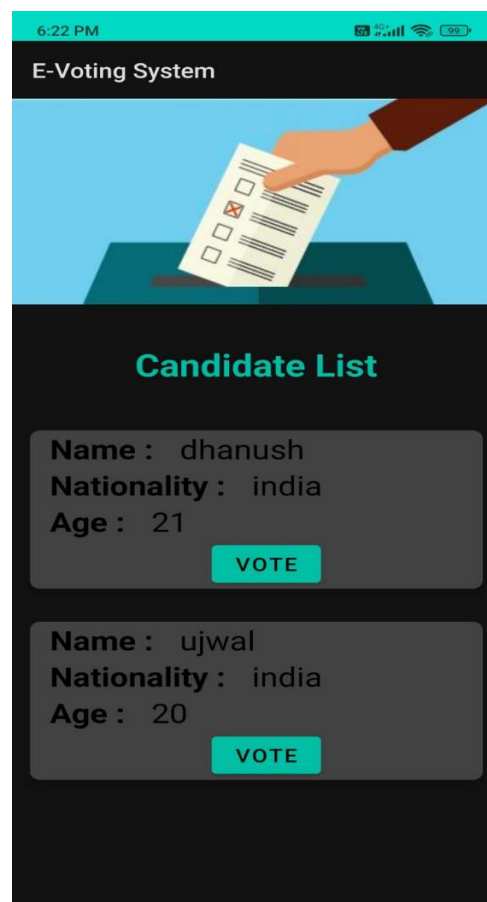


Fig 5.4.2 : Voting Page

## Chapter 6

# CONCLUSION & FUTURE SCOPE

Online Voting Systems have many advantages over the traditional voting system. Some of these advantages are less cost, faster generation results, easy accessibility, accuracy, and low risk of human and mechanical errors.

The voters have the choice of voting at their own free time and there is reduced congestion. It also minimizes on errors of vote counting. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes. It is very difficult to develop online voting system which can allow security and privacy on the high level.

Future development of this project focuses to design a system which can be easy to use and will provide security and privacy of votes on acceptable level by proper authentication and processing section. We can also add cornea detection authentication technique for more security. Further developing the contents for Settings option will also be our priority for future progress of this project.

## BIBLIOGRAPHY

- [1] <https://youtu.be/YrPDbxkfXoI>
- [2] [https://youtu.be/dRYnm\\_k3w1w](https://youtu.be/dRYnm_k3w1w)
- [3] [https://youtu.be/Jt-F7OSb\\_LU](https://youtu.be/Jt-F7OSb_LU)
- [4] <https://github.com/topics/voting-app>
- [5] [https://en.wikipedia.org/wiki/Electronic\\_voting](https://en.wikipedia.org/wiki/Electronic_voting)