## Indian Institute of Information Technology, Allahabad

## Database Management System

## PROJECT

**Instructor:** Dr. Ranjana Vyas, Dr. Anjali Gautam, Dr. Soumyadev Maity

---

# VoteIt


## An Online Voting Mobile Application

GitHub Repo Link
https://github.com/VarunT11/OnlineVotingSystem

## GROUP DETAILS

Group Number - 8
Section - B


| | | |
|---|---|---|
| IIT2019127 | Mridul Mittal | iit2019127@iiita.ac.in |
| IIT2019146 | Kshitij Kumar | iit2019146@iiita.ac.in |
| IIT2019148 | Piyush Gurjar | iit2019148@iiita.ac.in |
| IIT2019154 | Varun Tiwari | iit2019154@iiita.ac.in |

# TABLE OF CONTENTS

## ➔ SCHEMA OF THE TABLE

- Admin Table
- Voter Table
- Candidate Table
- Poll Table
- Poll Address Table
- Officer Table
- Officer PollNO

## ➔ USER INTERFACE

- Splash Screen
- Home Screen
- View Result Screen
- Poll List Screen
- Voter Login Screen
- Admin/Officer Login Screen
- Register Screen
- Voter Dashboard
- Officer Dashboard
- Admin Dashboard
- Cast Vote Screen
- OTP Verification Screen
- Add Voter Screen
- Add Admin Screen
- Add Candidate Screen
- Add Officer Screen
- Update Profile Screen

## ➔ ENTITY-RELATIONSHIP DIAGRAM (ER-DIAGRAM)

## ➔ CONVERT TABLE TO 3rd NORMAL FORM (3NF)

## ➔ PROJECT SETUP

# INTRODUCTION

## Details of the Project

**“VoteIt”** is an online mobile application that aims to provide users with a smooth and efficient way to cast their votes, manage and guide election processes effectively. It enables voters to choose their Candidate using their Mobile Phones thus reducing the load to travel and fulfill their right to vote. Also, it helps election Officers to manage and conduct elections smoothly. Admins manage the entire process of both Voters and Officers. The system is fully secured as it always asks for the *OTP* from the voter before casting their vote ensuring that no user can vote multiple times. Also, while logging in voters have to enter their VOTER_ID which is unique for every user. The Result Screen displays the result of the election once it's over.

## Tools and Technology Used

Tools that have been used in this project:

➔ **Android Studio**

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

➔ **MySQL Workbench**

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation, and maintenance into a single integrated development environment for the MySQL database system.

Technology that has been used in this project:

➔ **Android**

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

➔ **Java**

Java is a High Level, a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere.

➔ **Firebase**

Firebase is a toolset developed by Google to “build, improve, and grow your app”. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud and scale with little to no effort on the part of the developer.

➔ **XML(Extensible Markup Language)**

XML stands for eXtensible Markup Language. It is a software- and hardware-independent tool for storing and transporting data.

➔ **MySQL:** MySQL is a Relational Database Management System (RDBMS) that uses Structured Query Language (SQL) which is used for adding, accessing, and managing content in tables corresponding to its database.

# USE CASES

## Admin

- ➔ **Login_Admin**: This allows the verified user to log in by entering the required data.
- ➔ **Add Voter:** Adds voters by entering the required credentials who are eligible for mobile voting.
- ➔ **Add Admin:** Admins can add other admins.
- ➔ **Add Officer:** Allows Admins to add officers to manage polls.
- ➔ **Add Poll:** Admins can add polls by giving details about the location and time of election.
- ➔ **Declare Result/Change Election Status:** After the election, admins can declare Results in the VIEW RESULT SCREEN.
- ➔ **View Candidate**: Admins can see the list of participating candidates.
- ➔ **View Voter**: Admins can see the list of all the voters eligible for mobile voting.
- ➔ **View Officer:** Admins can see the list of all the Officers and the polls in which they are entitled to work for.
- ➔ **Remove Voter:** Admin has the power to remove voters in case of threats.
- ➔ **Remove Officer:** Admin has the power to remove officers in case of failure to perform poll duty.
- ➔ **Remove Poll:** Admin can remove poll which is not having elections.
- ➔ **Update Profile:** Allows admin to update their profile.
- ➔ **Forgot Password:** This allows the user to change the password by entering the required credentials.

## Officer

- ➔ **Login_Officer:** Allows the Officer to log in to avail the further functions.

- ➔ **Add Candidate:** Officer can add a Candidate who can contest the election.
- ➔ **View Candidate:** Officer can view the list of candidates who are contesting an election in their poll.
- ➔ **Remove Candidate:** Officers have the power to remove the candidate from their poll.
- ➔ **Edit Profile:** This allows the officer to update candidate profile info.
- ➔ **View Poll:** Officers can view details of the poll.
- ➔ **VIew Voter:** Officer can see voters who are eligible to vote in their poll.
- ➔ **Update profile:** Allows Officer to update their profile details.
- ➔ **Forgot Password:** This allows the user to change the password by entering the required credentials.

### Voter

- ➔ **Login_Voter:** Voters need to login to cast their vote.
- ➔ **Cast Vote:** Voters can choose the candidate from their poll and cast their vote.
- ➔ **View Candidate:** Can see the candidate details.
- ➔ **Update Profile:** Allows Voter to update their profile info.
- ➔ **See Result:** Can see the election result.
- ➔ **Forgot Password:** Allows the user to change the password by entering the required credentials.

## SCHEMA OF THE TABLES

### Admin Table

Field:

- Username(PRIMARY KEY): String (Not NULL)
- Name : String (Not NULL)
- Password : String (Not NULL)
- Photo_Url : String

- Phone No.: String (Not NULL)

### Voter Table

Field:

- VoterID(PRIMARY KEY) : String (Not NULL)
- Name : String (Not NULL)
- Phone No.: String (Not NULL)
- DOB : String (Not NULL)
- Poll No. : Integer (Not NULL)
- Is Mobile Registered : Boolean(Not NULL)
- Password : String (Default NULL)
- Registration Time : String (Default NULL)
- Photo_Url : String (Default NULL)
- Has Voted : Boolean (Default false)

### Candidate Table

Field:

- Candidate_ID(PRIMARY KEY): String (Not NULL)
- Name : String (Not NULL)
- Phone No.: String (Not NULL)
- DOB : String (Not NULL)
- Photo_Url : String
- Poll No. : Integer (Not NULL)
- Election Symbol Name : String (Not NULL)
- Election Symbol : String (Not NULL)
- No. of Votes : Integer (Not NULL)

### Poll Table

Field:

- Poll No.(PRIMARY KEY): Integer (Not NULL)
- No. of Candidates : Integer (Default 0)
- No. of Voters : Integer (Default 0)
- Election Start Time : String (Not NULL)
- Election End Time : String (Not NULL)

- No. of Votes Casted : Integer (Default 0)

### Poll Address Table

Fields:

- Poll No.(PRIMARY KEY): Integer (Not NULL)
- Address : String (Not NULL)

### Officer Table

Field:

- Username(PRIMARY KEY): String (Not NULL)
- Name : String (Not NULL)
- Password : String (Not NULL)
- Photo_Url : String
- Phone No.: String (Not NULL)

### OfficerPollNo. Table

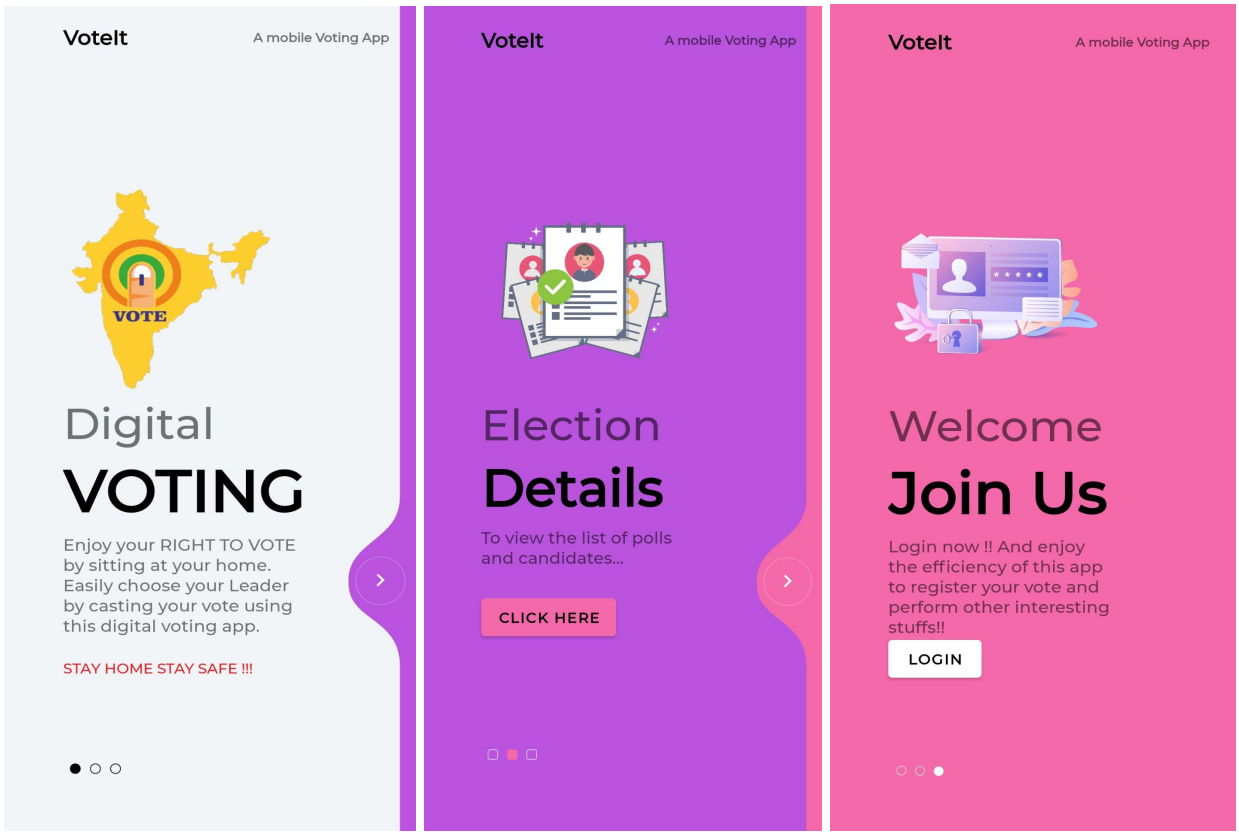
Field:

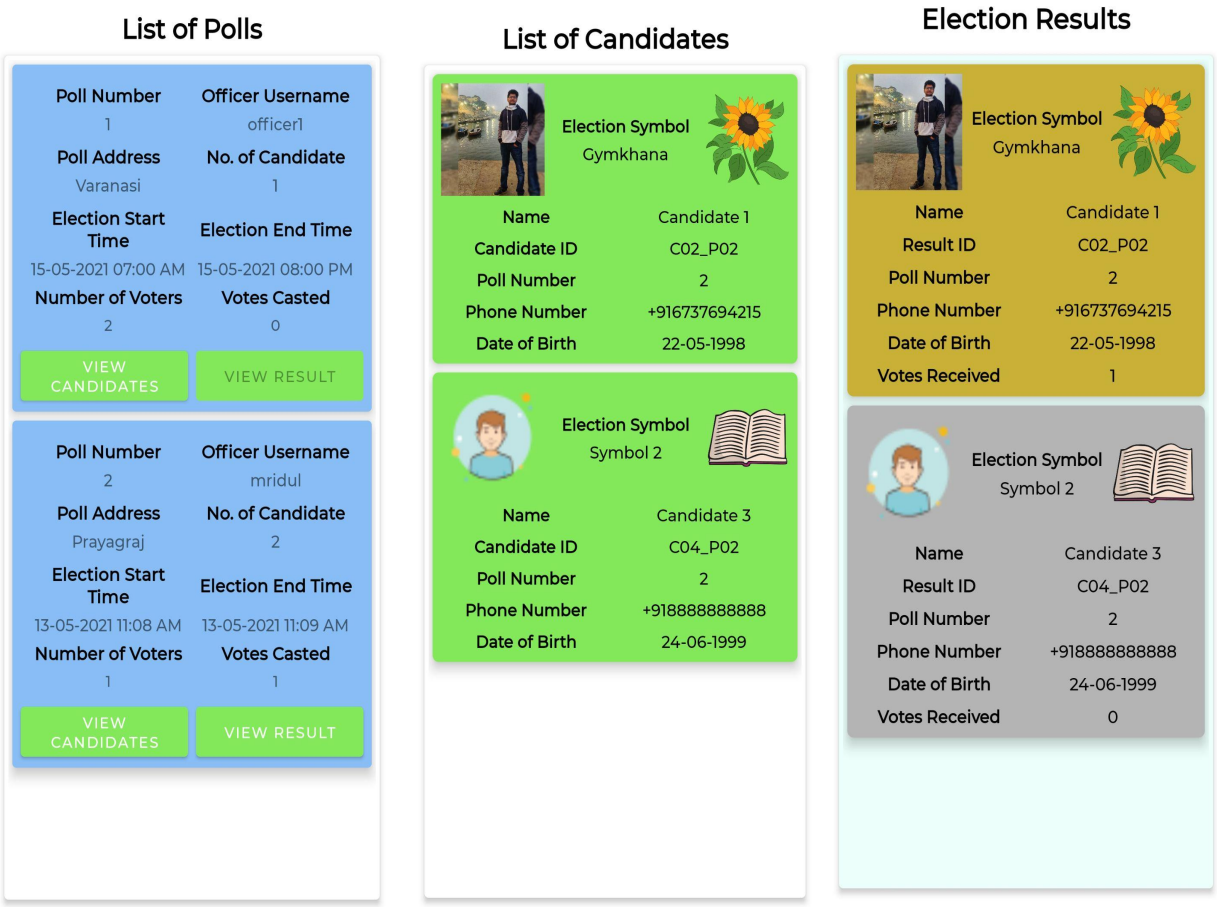
- Poll No.(PRIMARY KEY): Integer (Not NULL)
- Username : String (Not NULL)

## USER INTERFACE


**Splash Screen:** This is the first screen that comes up when the app runs. It contains the Animation with the logo and the app name.

**Home Screen** *(down)***:** This screen contains 3 fragments which have fluid animation. First fragment describes about app, second one gives Election Details and to login move to the third fragment.

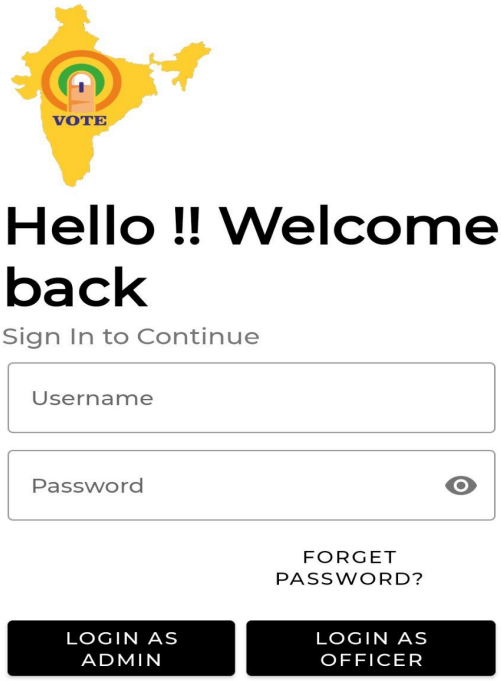





**Poll List:** It will show the list of poll with election details and have 2 buttons.

**View Candidate:** This button will take you to the screen where you can find the list of candidates who are contesting for the election in that particular poll.

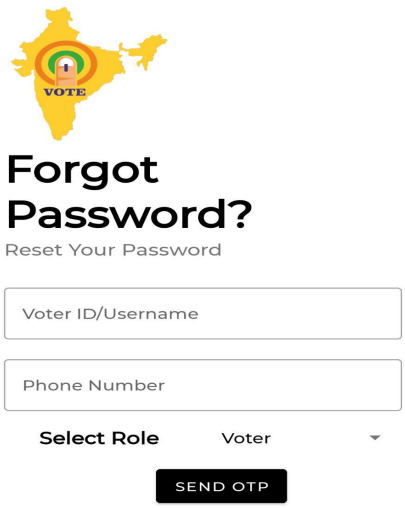
**View Result:** This button will show the result (once the election time is over). It will show the result in card view form in which the top most candidate will have the most number of votes.


**Login Screens:** We have designed two login screens.

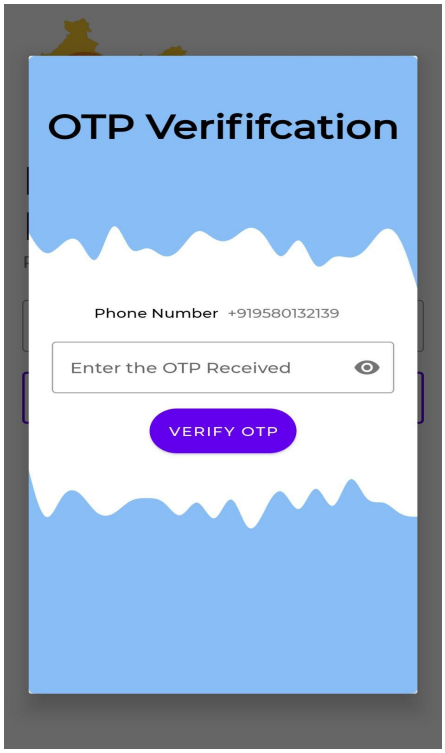
Login as Voter Screen: This screen serves the login purpose of voter and have multiple buttons like ***Forget Password?, New User, Voter list and Login(Admin/Officer)***.
*Voter List Button: This button will help the voter to see their voterID in order to login.*
*Login(Admin/Officer): This button will take you to the login screen for Admins or Officers.*

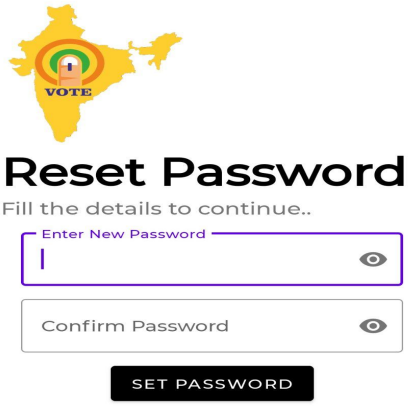
Login as Admin/Officer Screen: This screen serves the login purpose of Admins and election officers.

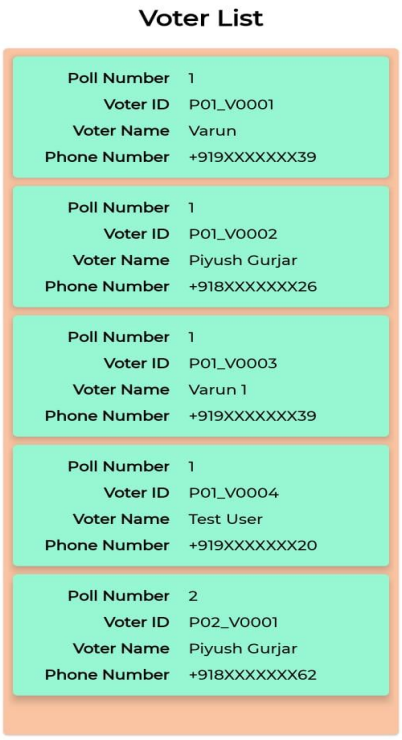



***Forget Password?:*** This screen allows the user to change the password by entering the required credentials.
Then, OTP will be sent to the registered phone for confirmation of the user. After entering the OTP one can change the password and set the new one.

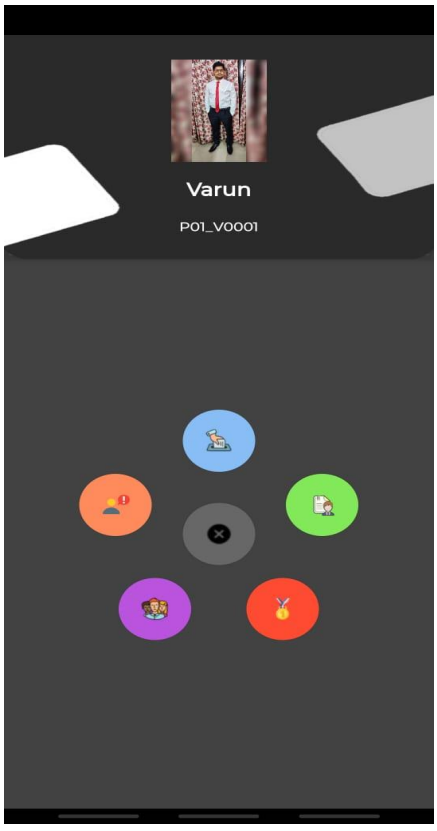
*Screens for OTP and New Password are shown below.*



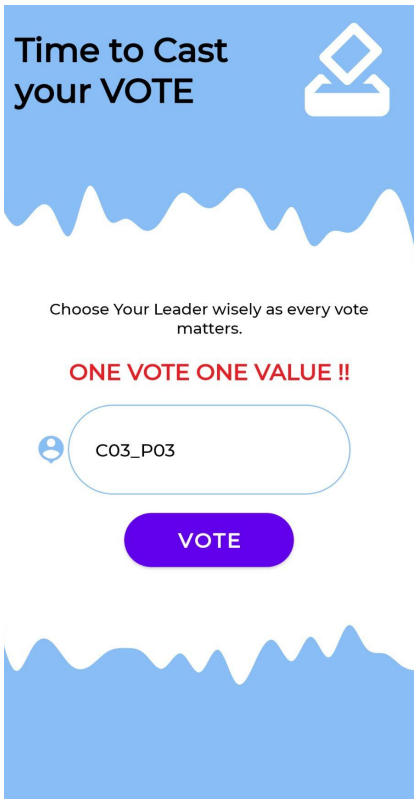



**Voter List Screen:** As mentioned above, this screen gives information about Voters. It shows voterID, their poll number, Phone Number (partially) and name.




**Voter Dashboard:** This screen shows all the functions a voter can perform through this app.
It shows voter img along with name and voterID.

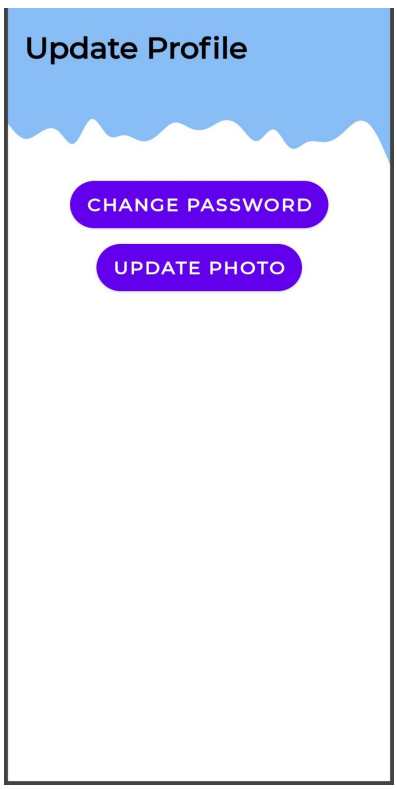
It contains circular animated menu which have 5 buttons each having a specific purpose.



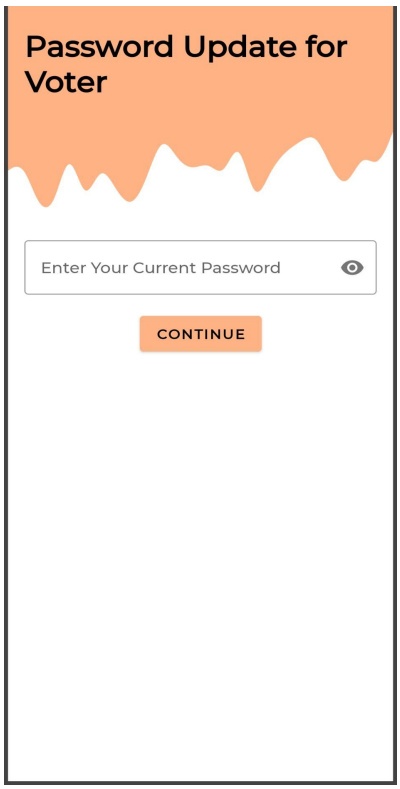

**VOTE screen:** This is the main screen where the voter can exercise their right to vote and choose the deserving candidate.

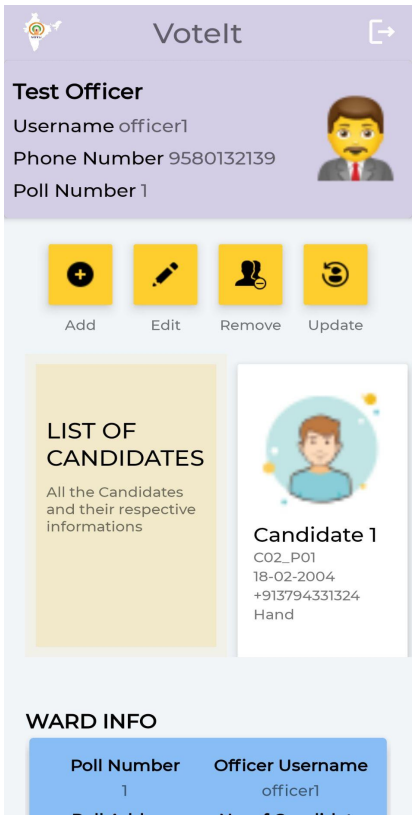
**Update Profile**: For user to change their profile details.

**Poll Details:** View details of the poll

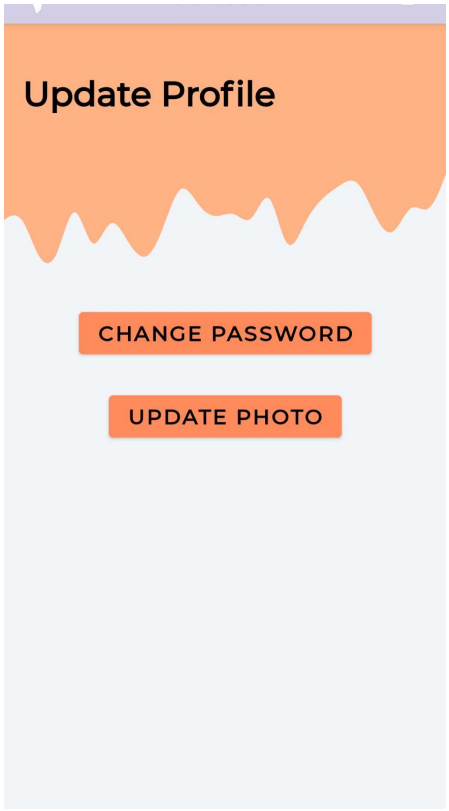




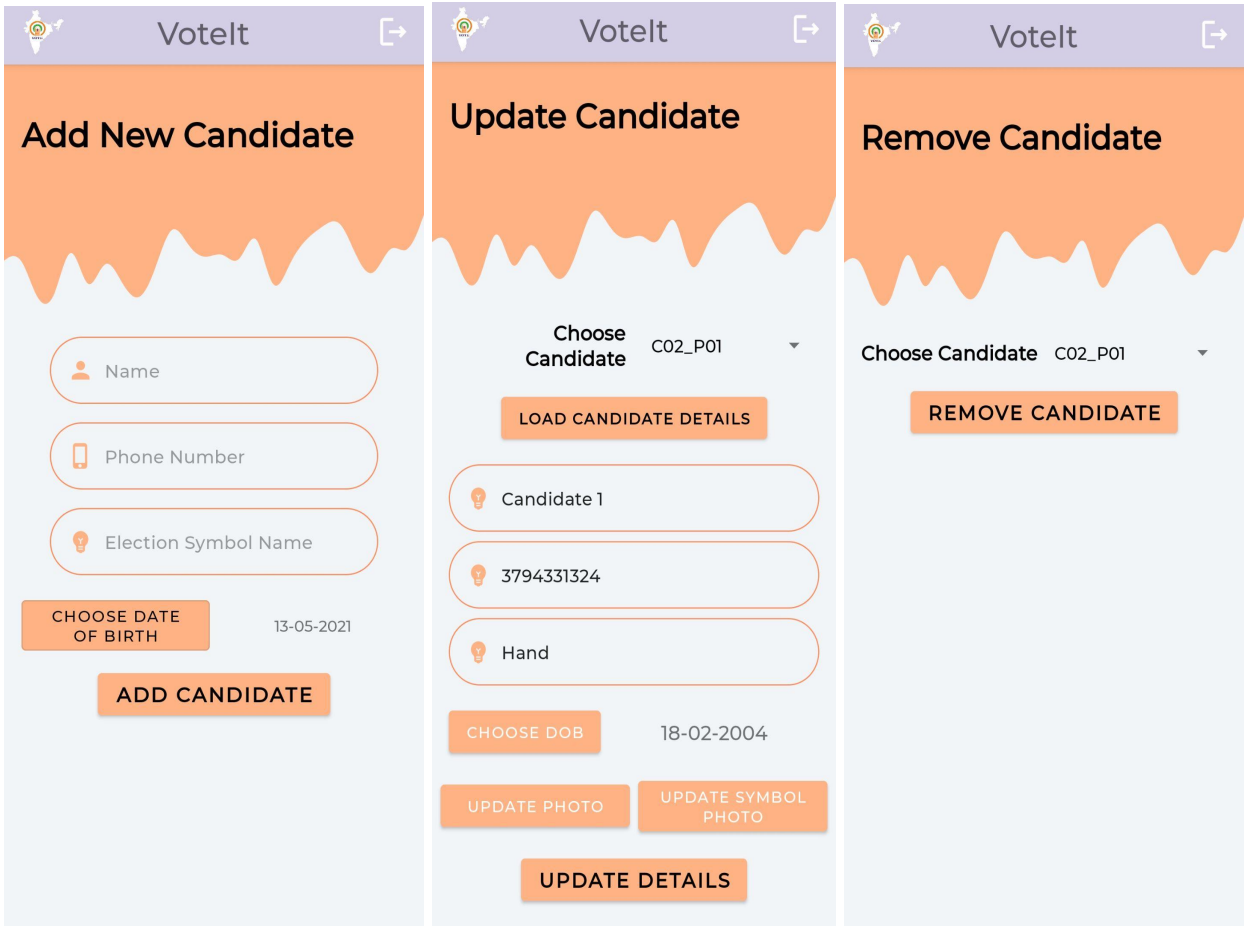





**Officer Dashboard:** This screen is the main screen for a officer where he can monitor the ongoing election.
It contains buttons each serving a purpose for officers *(explained in use cases above)*.
It has card layout design which briefly describes various informations regarding candidate, voter and ward.




**Update Profile Screen:**
Officer can update their own profile details.

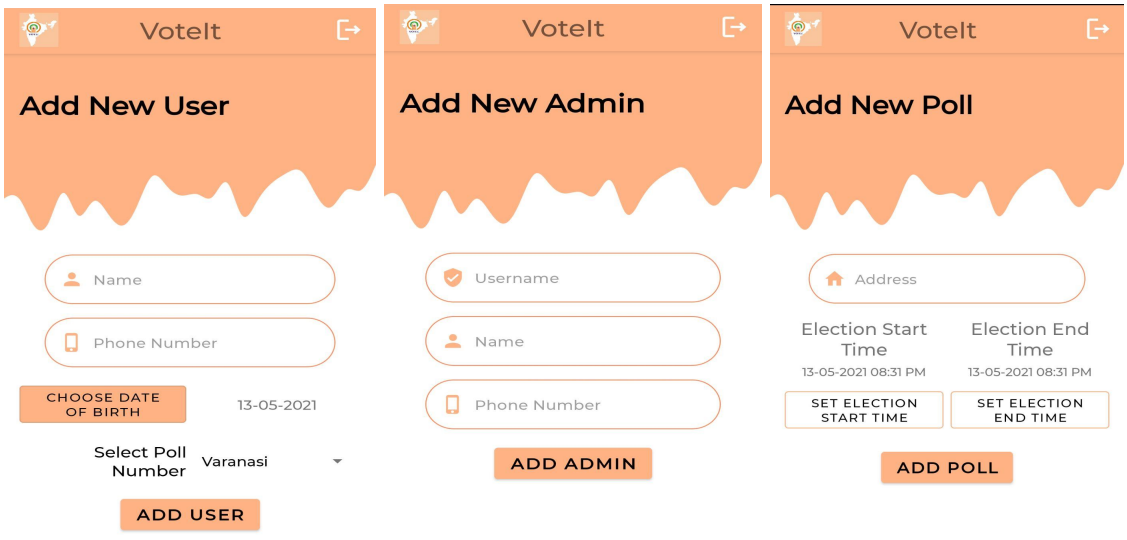


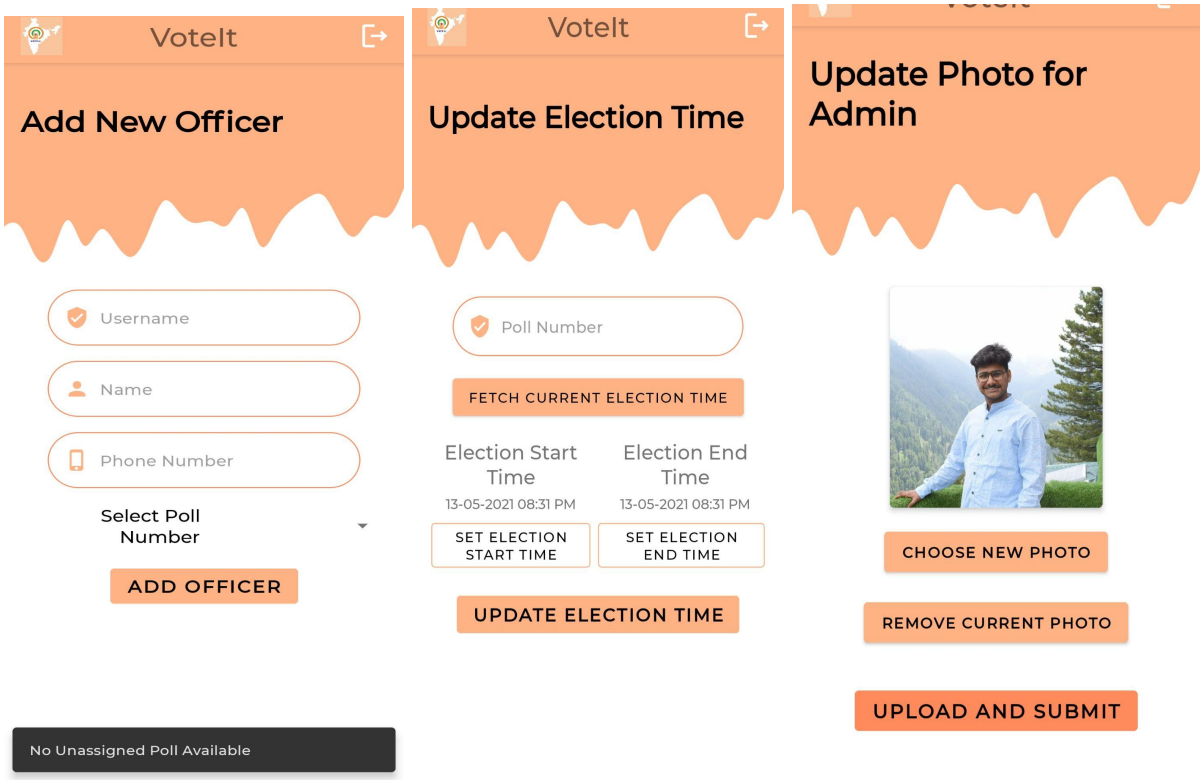

**Add New Candidate Screen:** Officer can add new candidates who are contesting election from their poll.

**Update Candidate Screen:** Officer can update the profile information of any candidate from their poll.

**Remove Candidate Screen:** Officer has the power to remove any candidate from the contesting the election.


**Admin Dashboard:** This screen is the main screen for the admin where he can monitor the ongoing election.
It has card layout design which briefly describes various informations regarding officer and voter.
It contains circular animated menu which have several buttons each having a specific purpose with some common functions.

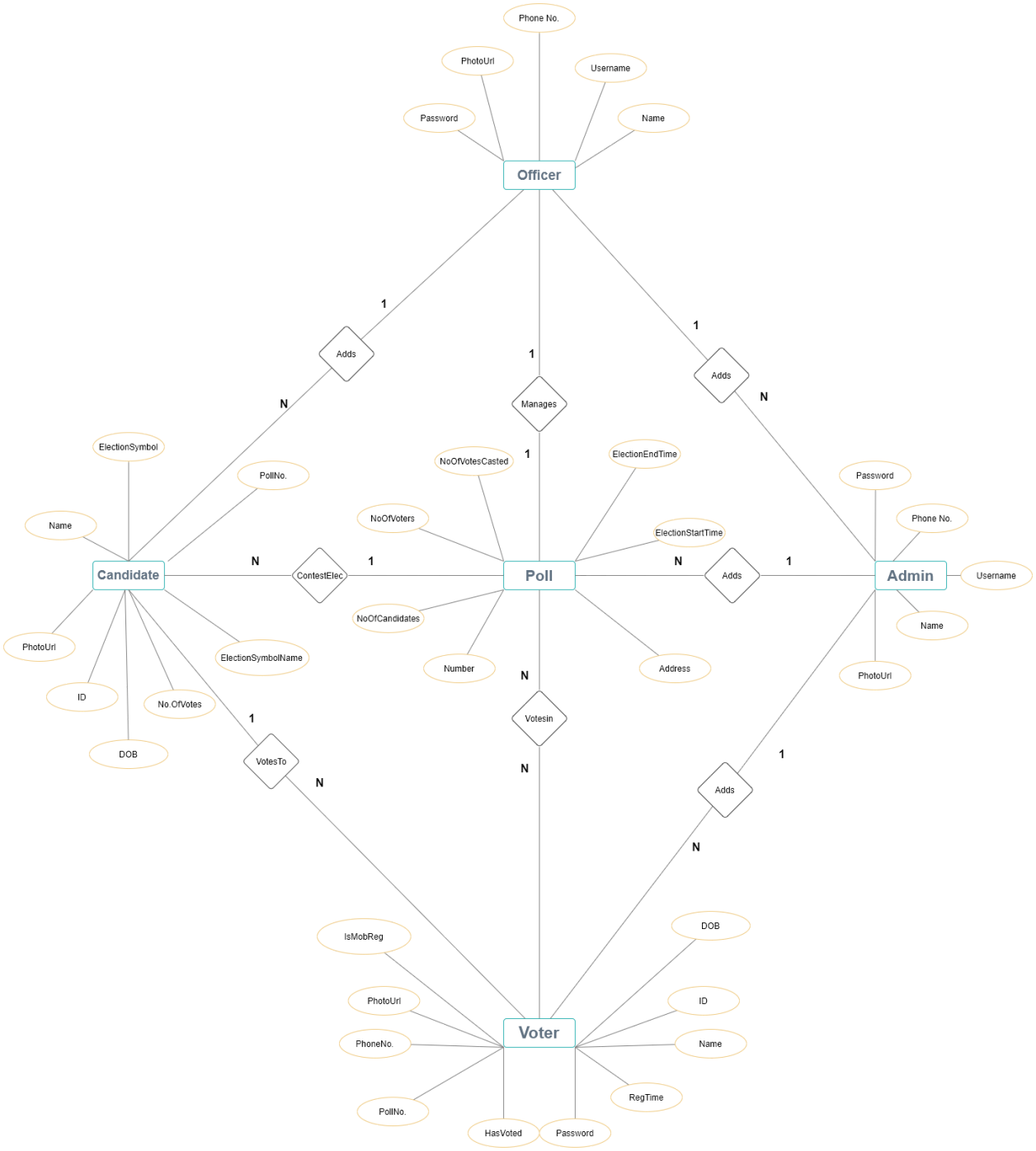





**Functions:**

Following are the functions which admin can perform *(explained in details in the use case section).*

# ER DIAGRAM




*(The ER diagram is also attached to the zip file in .png format)*

## 3rd NORMAL FORM

1. 1NF :-
   - There are no multi-valued attributes in any of the used tables.
   - All the records are unique
     So the given tables are already in 1NF

2. 2NF :-
   - The table is in 1NF.
   - No partial dependency or single column primary key

   There is no partial dependency in the table as we have single column primary key and all columns of the table are dependent on primary key not a part of primary key

   ➔ Table Admin

   Dependencies:

   Username → Name
   Username → Password
   Username → Photo_Url
   Username → Phone No.

   No partial Dependencies

   ➔ Table User

   Dependencies:

   VoterID → Name
   VoterID → Phone No.
   VoterID → DOB
   VoterID → Poll No.
   VoterID → Is Mobile Registered
   VoterID → Password

VoterID → Registration Time
VoterID → Photo_Url
VoterID → Has_Voted

No partial Dependencies

➔ Table Candidate

Dependencies:

Candidate_ID → Name
Candidate_ID → Phone No.
Candidate_ID → DOB
Candidate_ID → Photo_Url
Candidate_ID → Poll No.
Candidate_ID → Election Symbol Name
Candidate_ID → Election Symbol
Candidate_ID → No. of Votes

No partial Dependencies

➔ Table Poll

Dependencies:

Poll_No. → No. of Candidates
Poll_No. → No. of Voters
Poll_No. → Election Start Time
Poll_No. → Election End Time
Poll_No. → No. of Votes Casted

No partial Dependencies

➔ Table Poll Address

Dependencies:

Poll No. → Address

No partial Dependencies

➔ Table Officer

Dependencies:

Username → Name

Username → Password

Username → Photo_Url

Username → Phone No.

No partial Dependencies

➔ Table OfficerPollNo.

Dependencies:

Poll No. → Username

No partial Dependencies

So we see that there are no partial dependencies in our schema. So all the attributes are fully functionally dependent on the candidate keys. Hence the table is in 2nd Normal Form.

3. 3NF :-
   - The table is in 2NF.
   - Has no transitive functional dependencies

   It means that there should not exist any column other than primary key which can uniquely map the complete table

   For this we decomposed two tables
   1) Table Officer: Each officer had unique poll number so we decomposed this table in seperate table
   2) Table Poll : Each Poll had unique address which

can be used to map the table so we decomposed it in two tables (poll_address)

➔ Table Admin

- Username : Primary Key (Unique)
- Name : Any two admin may have same name (Not Unique)
- Password: Any two admin may have same password (Not Unique)
- Photo_Url: Optional Field so many values null (Not Unique)
- Phone_No: Two admins may have same phone no (Not Unique)

➔ Table Officer

- Username : Primary Key (Unique)
- Name : Any two officer may have same name (Not Unique)
- Password: Any two officer may have same password (Not Unique)
- Photo_Url: Optional Field so many values null (Not Unique)
- Phone_No: Two officers may have same phone no (Not Unique)

➔ Table Officer_PollNo ( Decomposed from Table Officer)

- Username : Primary Key (Unique)
- PollNo : Unique for each username

➔ Table Candidate

- Cand_Id : Primary Key for each candidate (Unique)
- Name : Any two candidates may have same name (Not Unique)
- Phone_No: Two candidates may have same phone no (Not Unique)

- DOB: Two candidates may have same dob (Not Unique)
- Photo_Url: Optional Field so many values null (Not Unique)
- Election Symbol Name : Many candidates from same party (Not Unique)
- Election Symbol Photo : Many candidates from same party (Not Unique)
- Number of Votes: Many candidates have same number of votes(Not Unique)

➔ Table Poll
- PollNo : Primary Key (Unique)
- Number of candidates : Same number possible for many polls (Not Unique)
- Number of voters : Same number possible for many polls (Not Unique)
- Election Start Time : Same time for all polls (Not Unique)
- Election End Time : Same time for all polls (Not Unique)
- Number of votes casted : Can be same for many polls (Not Unique)

➔ Table Poll_Address ( Decomposed from Table Poll)
- PollNo : Primary Key (Unique)
- Address : Unique for each poll

➔ Table Voters
- Voter Id : Primary key (Unique)
- Name : Any two voters may have same name (Not Unique)
- Phone Number : Same for families (Not Unique)
- DOB: Two voters may have same dob (Not Unique)
- Poll Number : Many voters from one poll (Not

Unique)
- Is Mobile Registered : Boolean 0/1 (Not Unique)
- Password: Any two voters may have same password (Not  Unique)
- Registration Time : Many voters register at the same time (Not Unique)
- Photo_Url: Optional Field so many values null (Not Unique)
- Has Voted : Boolean 0/1 (Not Unique)

- We can see that there exists no column in the table except the primary key which has all unique values and can be used as the primary key to map the table.

  Hence all the tables are 3NF normalised.

## PROJECT SETUP

**Method 1:** Directly Install the apk from the folder in your mobile device/emulator.

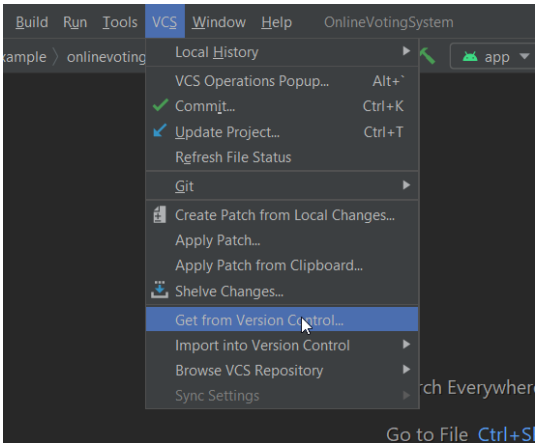
**Method 2:** Set Up Project Locally from GitHub

- Download latest version of Android Studio
  https://developer.android.com/studio

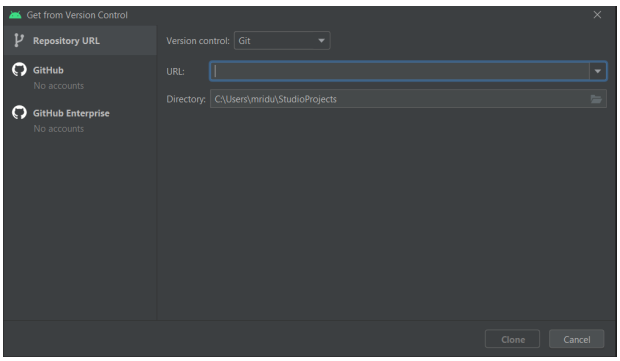
- Make sure git is installed
  https://git-scm.com/download/win

- Make sure latest version of JAVA SE SDK is installed
  https://www.oracle.com/java/technologies/javase-jdk16-downloads.html

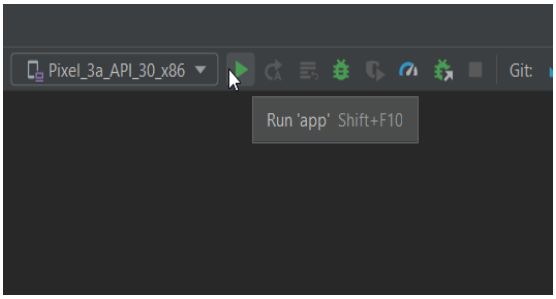
- Click on Get From Version Control




- Add the github project repo link in the URL field and click on Clone
  https://github.com/VarunT11/OnlineVotingSystem.git




- Wait for gradle to build and then click on run app to run the project on the emulator/device.



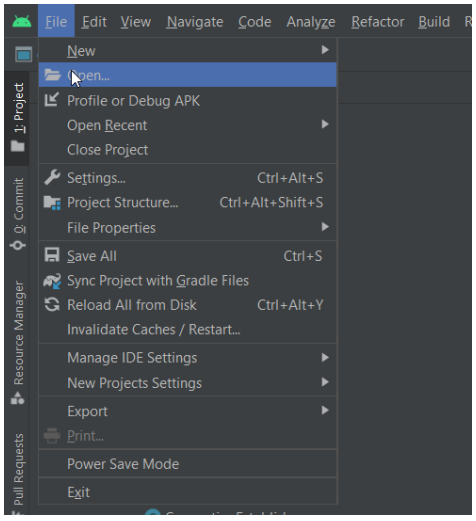

**Method 3:** Set Up Project Locally from folder

- Download latest version of Android Studio https://developer.android.com/studio

- Make sure git is installed https://git-scm.com/download/win

- Make sure latest version of JAVA SE SDK is installed https://www.oracle.com/java/technologies/javase-jdk16-downloads.html

- Click on Open Project and navigate the to the project folder




- Wait for gradle to build and then click on run app to run the project on the emulator/device.