**Chapter 1**

**INTRODUCTION**

The "Login System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Login System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Password, Username, Register, Change Password, Programmer. Every Login System has different Username needs; therefore, we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

**Chapter 2**

**TECHNOLOGIES USED**

**2.1 Firebase Platform**

Firebase is a backend service provided by Google that offers many useful features for mobile and web apps. Some of the features provided by Firebase are as followed: Realtime Database – allows data to be stored and synced between users and devices in a NoSQL database. The updated data is synced with all connected devices very they too quickly, and the data remains after the app goes offline. Authentication – provides a simple and secure way to manage users using the app. It provides multiple methods of authentication such as, email and password, third-party providers like Google or Facebook, and using your existing account system directly. Cloud Functions – allows the developer to extend the app using custom backend code without the need to manage one’s own servers. The functions can be triggered by events, which are produced by Firebase, Google Cloud services, and other third-party services. Cloud Storage – can be used to store user-generated content like images, audio, and video with object storage built for Google scale. The Firebase SDKs for Cloud Storage add Google security.

**2.2 Android Studio**

Android Studio is an integrated development environment (IDE) for Google Android Operating System. It is built based on JetBrains’ IntelliJ IDEA Community Edition, and it specifically designed for creating applications on Android devices. Some of the key features of Android Studio are as follows: Instant Run – a feature that pushes code and resource changes to the running app. It allows changes to be made to the app without the need to restart the app, or rebuilding the APK, so that the effects can be seen instantly. An Emulator – a virtual android device that can simulate variety of hardware features such as fire base authentication, GPS location, network latency, motion sensors, and multi touch input that can be used to run and install the app. It can then be used for testing purposes. Testing Tools and Frameworks – extensive testing tools such as, JUnit 4 and functional UI test frameworks are included with Android Studio. Espresso Test Recorder can generate UI test code by recording the developer’s interactions with the app on a device or emulator.

The tests can be run on a device, an emulator, in Firebase Test Lab, or on a continuous integration environment.

**2.3 XML**

XML or Extensible Markup Language is a text language that can be used to describe the behaviour of programming languages that process them. XML was developed XML working group in 1996. According to World Wide Web Consortium there are ten design goals for XML. These design goals are: XML shall be straightforwardly usable over the Internet. XML shall support a wide variety of applications. XML shall be compatible with SGML. It shall be easy to write programs which process XML documents. The number of optional features in XML is to be kept to the absolute minimum, ideally zero. XML documents should be human-legible and reasonably clear. The XML design should be prepared quickly. The design of XML shall be formal and concise. XML documents shall be easy to create. 10.Terseness in XML markup is of minimal importance. XML is used when transferring data from the database to the client, and in designing the visual aspect of Android applications. When data is sent from the database, it is sent using XML. This allows the data to be processed by any programming language the same way, since the data is always sent using XML. As mentioned, XML is also used to design the user interface of Android applications. This means that all the visual aspects such as, the layout of the page, the position of all button and text fields, as well as the colour of anything on the page is specified using XML. Since XML is human-legible, it makes the process of designing a page in the app relatively easy and intuitive.

**2.4 Java Programming Language**

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991. In the paper The Java Language Specification Java SE 8 Edition James Gosling states, “Java programming language is a general- purpose, concurrent, class based, object-oriented language. It is designed to be simple enough that many programmers can achieve fluency in the language. The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included. It is intended to be a production language, not a research language.” Java is a very flexible programming language which is used to create many different types of applications for

many different operating systems. This is possible because Java can be run on any operating system, as long a s the Java Runtime Environment is available. The application created for Android devices must be coded using Java programming language. This allows these apps to work on variety of different devices, no matter the company that has manufactured device.

**2.5 NoSQL**

There are two types of databases. Relational database and non-relational database. SQL is the best-known language used in creating relational databases, so any nonrelational database is referred to as NoSQL, also known as “NoSQL” or “Not Only SQL”. As mentioned, one of the services provided by the Firebase platform is a database. This database is a NoSQL database, which means it is non-relational and no tables exist within the database; instead, all data is saved in a tree-like structure. Any storage and retrieval of data is done by using specific functions designed for the database. These functions can be called using Java, or other programming languages depending on where the database is being used.

This means that there is no need for another programming language like SQL to be used to retrieve and store data within the database.

**2.6 Hardware Specification**

RAM : 8 GB RAM

Storage : 100 GB

**2.7 Software Specifications**

User interface design : XML

Programming Language : Java

Server : Android Studio

**Chapter 3**

**FEASIBILITY STUDY**

After doing the project Login Authentication System, study and analysing all the

existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite

time.

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

**3.1 Economical Feasibility**

This is a very important aspect to be considered while developing a project. We

decided the technology based on minimum possible cost factor.

* All hardware and software cost has to be borne by the organization.
* Overall, we have estimated that the benefits the organization is going to receive

from the proposed system will surely overcome the initial costs and the later on

running cost for system.

**3.2 Technical Feasibility**

This included the study of function, performance and constraints that may affect

the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

**3.3 Operational Feasibility**

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

**Chapter 4**

**IMPLEMENTATION**

**4.1 Firebase Authentication**

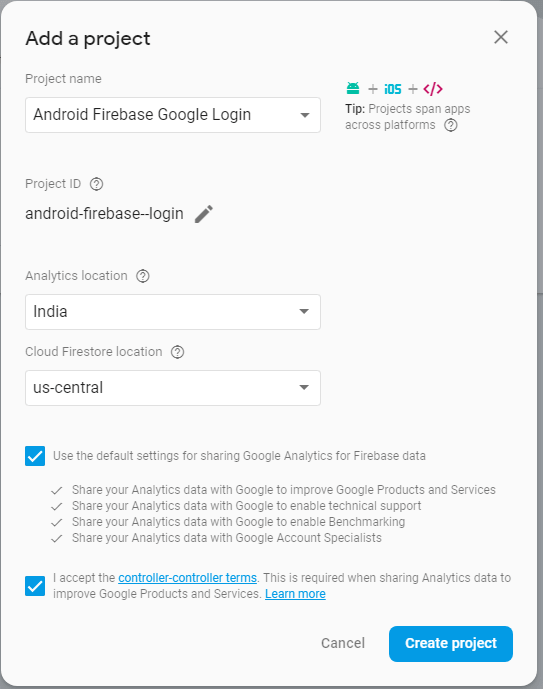
**Firebase Authentication** provides backend services for easy use the SDKs and in-built UI libraries to authenticate the user in the application. Most of the apps need the identity of the user, and after knowing their status, the app saves the user's data securely in the cloud. It aims to build a secure authentication system.

Using Firebase Authentication, we will authenticate the log-in of Google, Facebook, GitHub, Twitter and more.

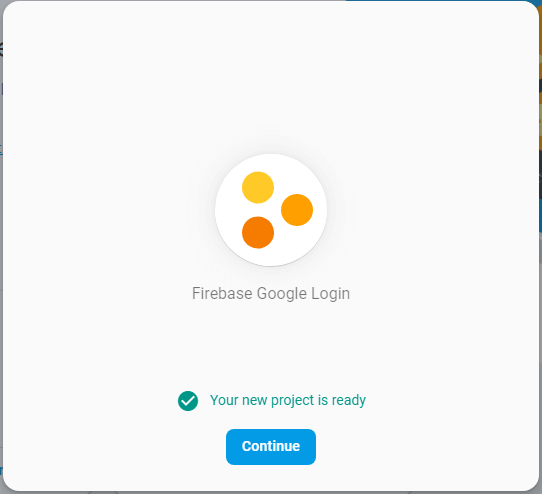
In this tutorial, we will integrate the Firebase Authentication for Google Sign-In functionality in our Android application using **Google** and **Firebase APIs.**

**4.2 Steps to create and configure Android App on Google Firebase Account**

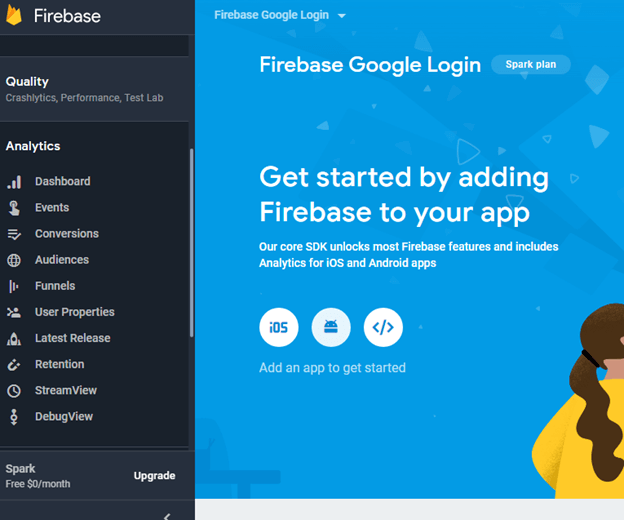
1. Create a Firebase developer account at <https://firebase.google.com/> and click on 'GO TO CONSOLE'.
2. Click on 'Add project'.
3. Fill the project name and select the analytics location, cloud Firestorm location, accept the controller terms and click 'Create project'.



1. 4. When your new project is successfully ready click on **'Continue'**.



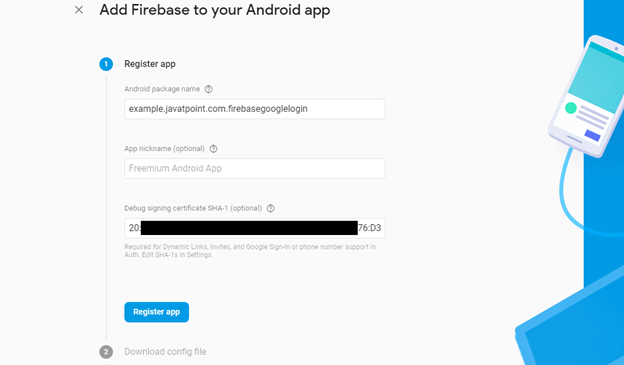
5. Select the **'Android'** platform SDK.



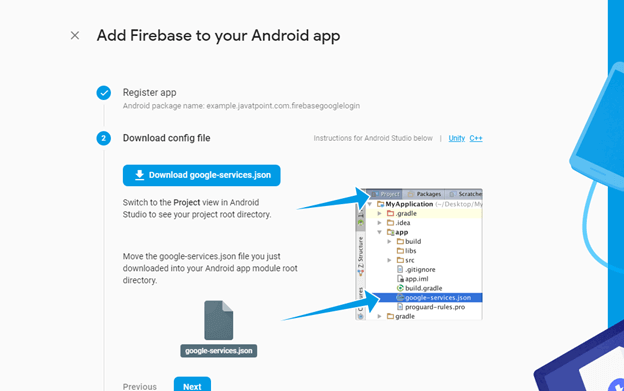
6. Register your app to Firebase by providing required app information and click on **'Register app'**.

We can get the app certificate SHA-1 key through following steps:

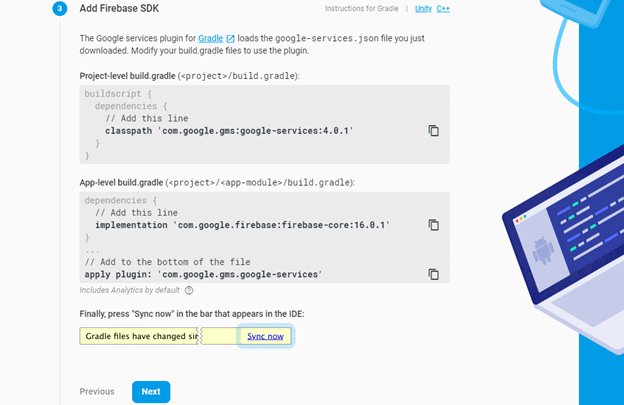
1. Open Android project.
2. Open the Gradle tab from a right-side panel.
3. Double click on 'signing Report'.
4. We will found our app SHA-1 key on 'Gradle Console'.



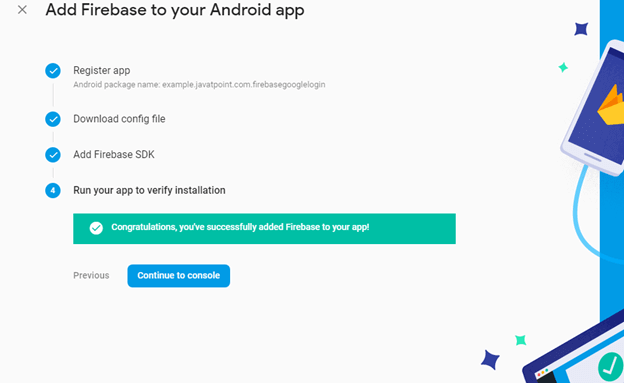
7. Now download the **'google-services.json'** file to integrate it into the Android application and click on **'Next'**.



8. Add the firebase SDK dependencies in. Gradle files of your application and click 'Sync now' in the IDE and click on **'Next'**.



1. Run your app to verify the installation configuration, if everything is fine it displays a success message and then click on **'Continue to console'**.



10. At console page select **Authentication -**gt**; Sign-in method** -gt; **Google**- gt**; Enable** and click on**'Save'**

**Chapter 5**

**SOURCE CODE**

**5.1 SIGNUP**

**5.1.1 Java Code For SIGNUP**

package com.example.anu;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.renderscript.ScriptGroup;

import android.view.View;

import android.widget.Toast;

import com.example.anu.databinding.ActivityMainBinding;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.FirebaseFirestore;

public class MainActivity extends AppCompatActivity

{

ActivityMainBinding binding;

ProgressDialog progressDialog;

FirebaseAuth firebaseAuth;

FirebaseFirestore firebaseFirestore;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

binding=ActivityMainBinding.inflate(getLayoutInflater());

setContentView(binding.getRoot());

firebaseAuth=FirebaseAuth.getInstance();

firebaseFirestore=FirebaseFirestore.getInstance();

progressDialog=new ProgressDialog(this);

binding.signup.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String name=binding.fullname.getText().toString();

String number=binding.mobilenumber.getText().toString();

String email=binding.emailaddress.getText().toString().trim();

String password=binding.password.getText().toString();

progressDialog.show();

firebaseAuth.createUserWithEmailAndPassword(email,password)

.addOnSuccessListener(new OnSuccessListener<AuthResult>()

{

@Override

public void onSuccess(AuthResult authResult) {

startActivity(new Intent(MainActivity.this,LoginActivity.class));

progressDialog.cancel();

firebaseFirestore.collection("User")

.document(FirebaseAuth.getInstance().getUid())

.set(new UserModel(name,number,email));

}

})

.addOnFailureListener(new OnFailureListener()

{

@Override

public void onFailure(@NonNull Exception e)

{

Toast.makeText(MainActivity.this,e.getMessage(),Toast.LENGTH\_SHORT).show();

progressDialog.cancel();

}

});

}

});

binding.goToLoginActivity.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

startActivity(new Intent(MainActivity.this,LoginActivity.class));

}

});

}

}

**5.1.2 XML Code For SIGNUP**

<?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"

android:orientation="vertical"

android:padding="20dp"

tools:context=".MainActivity">

<EditText

android:id="@+id/fullname"

android:layout\_margin="10dp"

android:padding="10dp"

android:background="@drawable/editext\_background"

android:hint="Full Name"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

<EditText

android:id="@+id/mobilenumber"

android:layout\_margin="10dp"

android:padding="10dp"

android:background="@drawable/editext\_background"

android:hint="Mobile Number"

android:maxLength="11"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

<EditText

android:id="@+id/emailaddress"

android:layout\_margin="10dp"

android:padding="10dp"

android:background="@drawable/editext\_background"

android:hint="Email Address"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

<EditText

android:id="@+id/password"

android:layout\_margin="10dp"

android:padding="10dp"

android:background="@drawable/editext\_background"

android:hint="Password"

android:inputType="textPassword"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

<Button

android:id="@+id/signup"

android:layout\_marginTop="10dp"

android:layout\_gravity="center"

android:text="Sign Up"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

**5.2 LOGIN**

**5.2.1 Java Code For LOGIN**

package com.example.anu;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Toast;

import com.example.anu.databinding.ActivityLoginBinding;

import com.example.anu.databinding.ActivityMainBinding;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.ktx.Firebase;

public class LoginActivity extends AppCompatActivity

{

ActivityLoginBinding binding;

FirebaseAuth firebaseAuth;

ProgressDialog progressDialog;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

binding=ActivityLoginBinding.inflate(getLayoutInflater());

setContentView(binding.getRoot());

firebaseAuth=FirebaseAuth.getInstance();

progressDialog=new ProgressDialog(this);

binding.login.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View view)

{

String email=binding.emailaddress.getText().toString().trim();

String password=binding.password.getText().toString().trim();

progressDialog.show();

firebaseAuth.signInWithEmailAndPassword(email,password)

.addOnSuccessListener(new OnSuccessListener<AuthResult>()

{

@Override

public void onSuccess(AuthResult authResult)

{

progressDialog.cancel();

Toast.makeText(LoginActivity.this,"Login Succesfull",Toast.LENGTH\_SHORT).show();

}

})

.addOnFailureListener(new OnFailureListener()

{

@Override

public void onFailure(@NonNull Exception e) {

progressDialog.cancel();

Toast.makeText(LoginActivity.this,e.getMessage(),Toast.LENGTH\_SHORT).show();

}

});

}

});

binding.resetpassword.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

{

String email=binding.emailaddress.getText().toString();

progressDialog.setTitle("Sending Mail");

progressDialog.show();

firebaseAuth.sendPasswordResetEmail(email)

.addOnSuccessListener(new OnSuccessListener<Void>()

{

@Override

public void onSuccess(Void unused)

{

progressDialog.cancel();

Toast.makeText(LoginActivity.this,"Email sent",Toast.LENGTH\_SHORT).show();

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e)

{

progressDialog.cancel();

Toast.makeText(LoginActivity.this,e.getMessage(),Toast.LENGTH\_SHORT).show();

}

});

}

});

binding.goToSignUpActivity.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v) {

startActivity(new Intent(LoginActivity.this,MainActivity.class));

}

});

}

}

**5.2.2 XML Code For LOGIN**

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

<Linearayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="20dp"

android:orientation="vertical"

tools:context=".LoginActivity">

<EditText

android:id="@+id/emailaddress"

android:layout\_margin="10dp"

android:padding="10dp"

android:background="@drawable/editext\_background"

android:hint="Email Address"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

<EditText

android:id="@+id/password"

android:layout\_margin="10dp"

android:padding="10dp"

android:background="@drawable/editext\_background"

android:hint="Password"

android:inputType="textPassword"

android:layout\_width="match\_parent"

android:layout\_height="50dp"/>

<Button

android:id="@+id/login"

android:layout\_marginTop="10dp"

android:layout\_gravity="center"

android:text="Login"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

**Chapter 6**

**SNAPS SHOTS**

Figure 6.1 shows the signup page for new users.

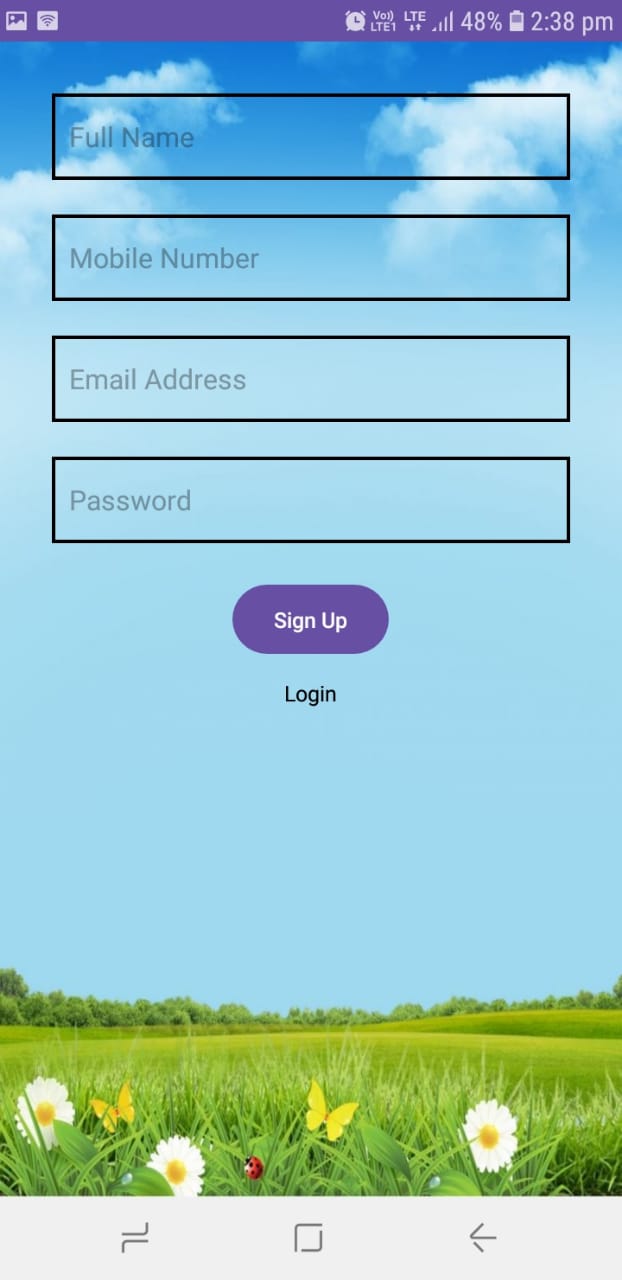


Fig. 6.1: Signup page

Figure 6.2 shows the fill signup page for new users.

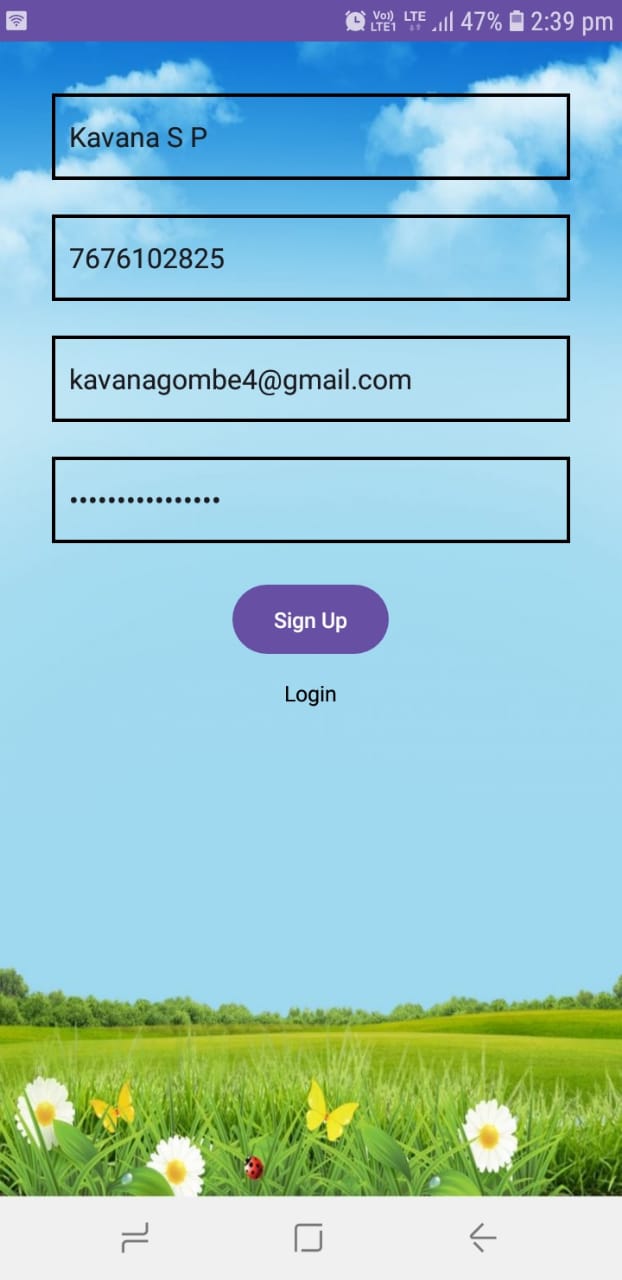
****

Fig. 6.2 : Fill the Signup Form

Figure 6.3 shows the login page.

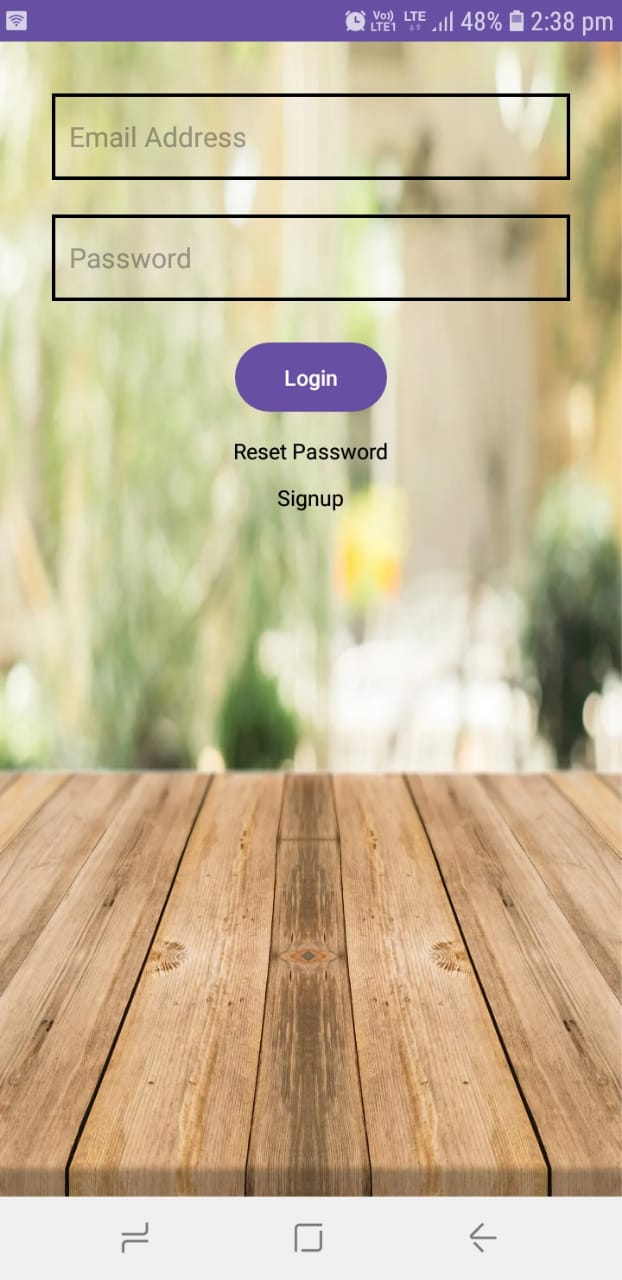


Fig. 6.3 : Login Form

Figure 6.4 shows the login page for already how have fill the signup page.

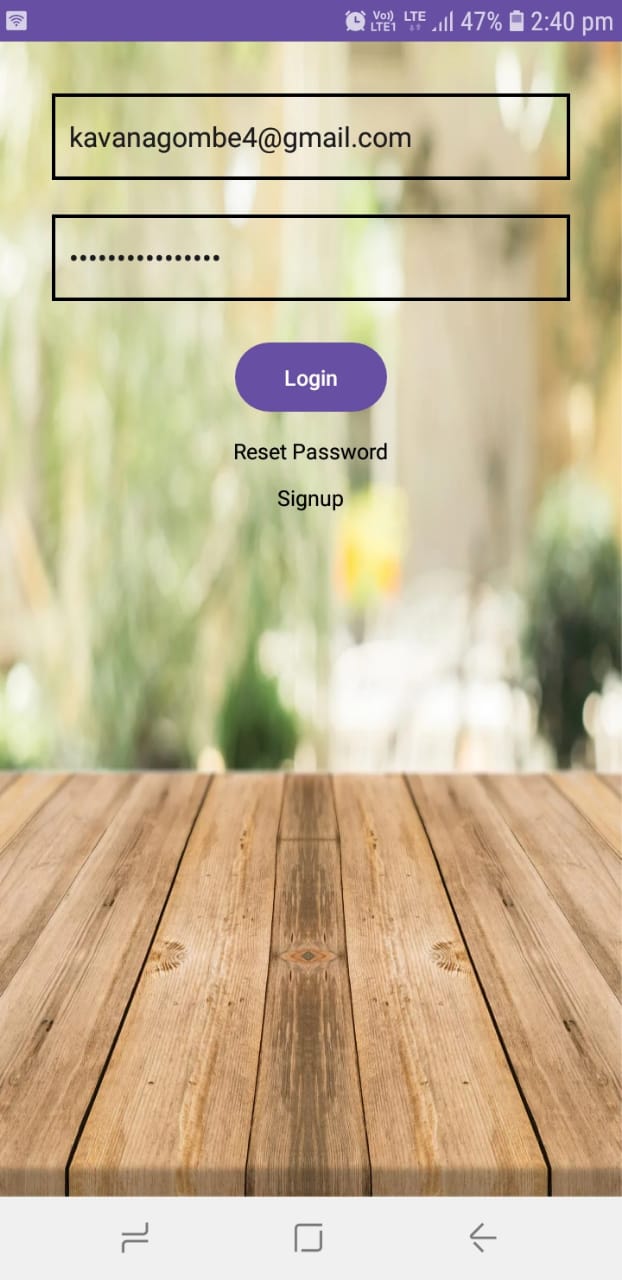


Fig. 6.4 : Login Form Fill

Figure 6.5 shows the send the mail for the user.

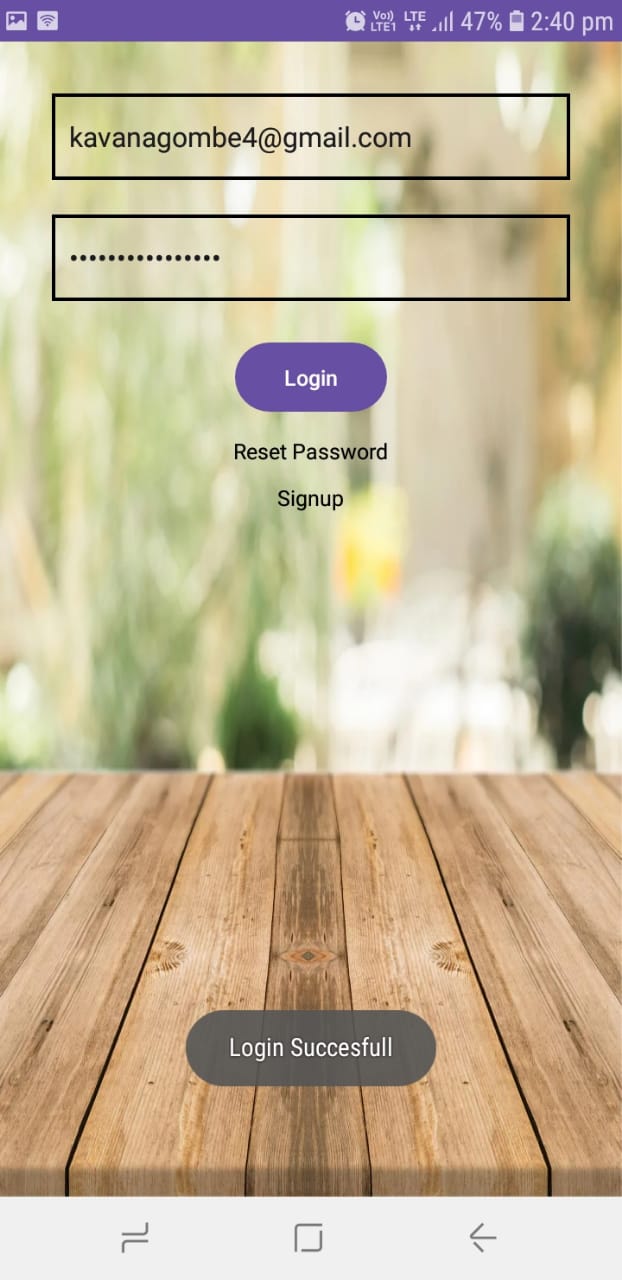


Fig. 6.5 : Mail Sent

Figure 6.6 shows the mail in users mail account.

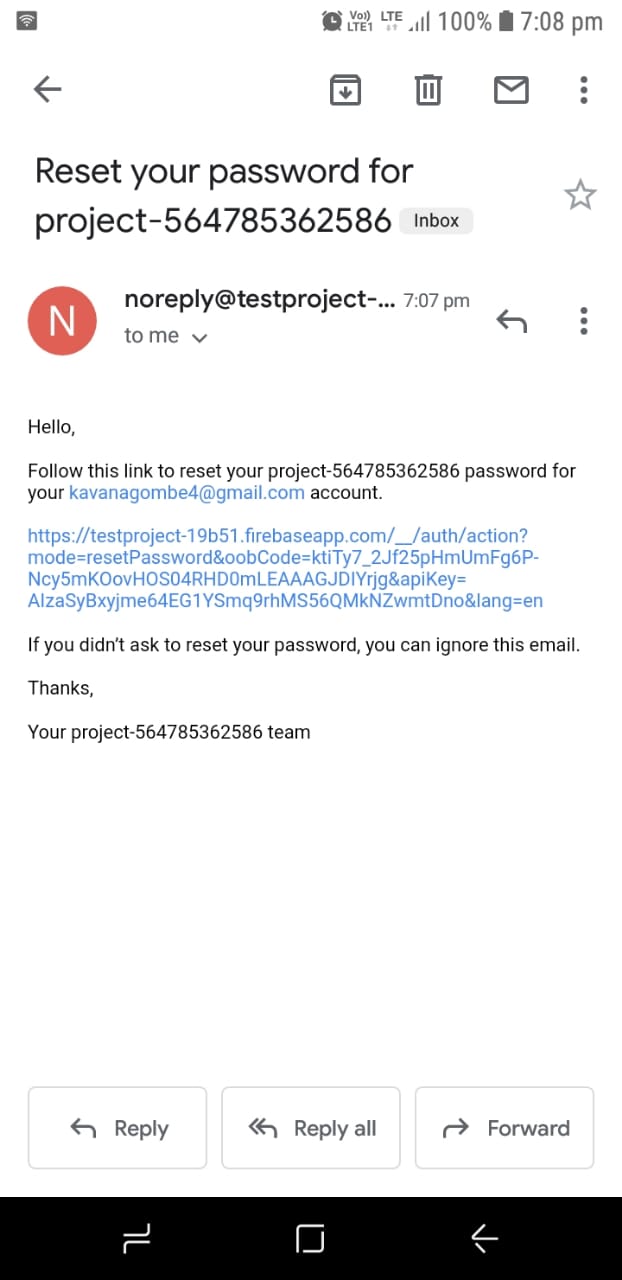


Fig. 6.6 : Email Link for Reset Password

Figure 6.7 show the change the mail password.

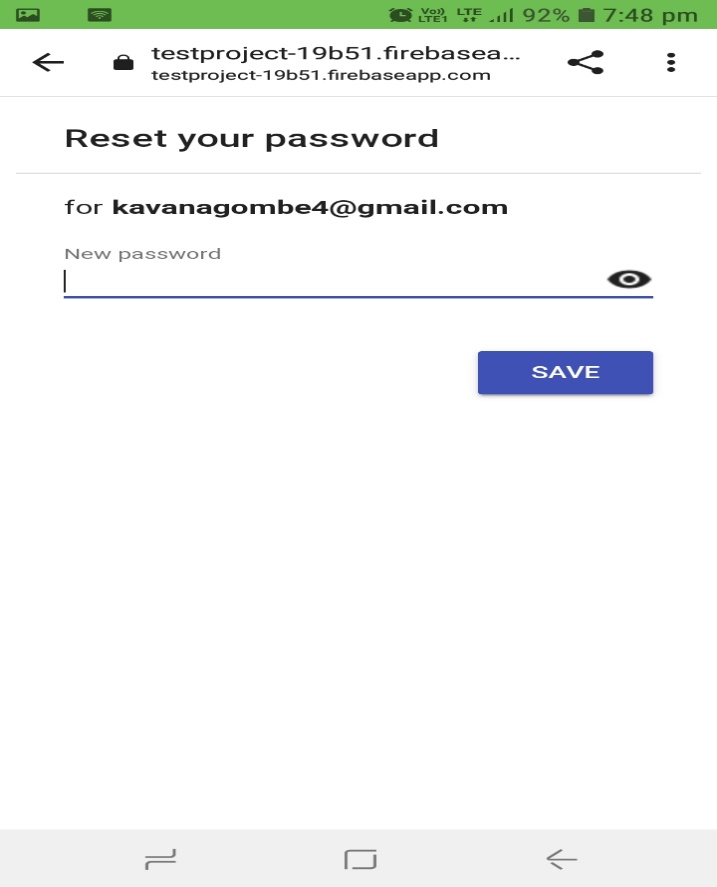


Fig. 6.7 : Reset Password

**Chapter 7**

**CONCLUSION AND FUTURE SCOPE**

Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

**Future scope of the project**

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

* We can add printer in future.
* We can give more advance software for Login System including more facilities.
* We will host the platform on online servers to make it accessible worldwide.
* Integrate multiple load balancers to distribute the loads of the system.
* Create the master and slave database structure to reduce the overload of the database queries.
* Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

The above-mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Username and Password. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the Login System.

Enhancements can be done to maintain all the Username, Password, Login, Register,

Programmer.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.

**REFERENCES**

1. Google for problem solving
2. http://www.javaworld.com/javaworld/jw-01-1998/jw-01-Credentialreview.html
3. Head First Java 2nd Edition
4. http://www.jdbc-tutorial.com/
5. Java and Software Design Concepts by Apress
6. https://www.tutorialspoint.com/java/
7. http://www.javatpoint.com/java-tutorial
8. https://docs.oracle.com/javase/tutorial/
9. http://www.JSP.net/
10. http://www.tutorialspoint.com/mysql/
11. httpd.apache.org/docs/2.0/misc/tutorials.html